ISBN 978-82-326-4272-4 (printed ver.)
ISBN 978-82-326-4273-1 (electronic ver.)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Vegard Edvardsen

Doctoral thesis

Doctoral theses at NTNU, 2019:338

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

NTNU

Doctoral theses at NTNU, 2019:338

Vegard Edvardsen

Navigating by decoding grid cells

Investigating the role of entorhinal grid cells through computational modeling at the intersection of neuroscience and artificial intelligence

NTNU
Norwegian University of
Science and Technology

Vegard Edvardsen

Navigating by decoding grid cells

Investigating the role of entorhinal grid cells through computational modeling at the intersection of neuroscience and artificial intelligence

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Computer Science

ISBN 978-82-326-4272-4 (printed ver.)
ISBN 978-82-326-4273-1 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:338

Printed by NTNU Grafisk senter

Abstract

Ever since grid cells were discovered in the mammalian entorhinal cortex over a decade ago, the striking representation of space generated by these neurons has offered a peek at the inner workings of navigational processes and possibly other high-level cognitive tasks in the brain. The hexagonal, grid-like patterns these neurons produce as the animal travels across space appear intriguingly algorithmic in their nature, raising the question of whether the same general principles could successfully be applied in artificial neural networks, potentially leading to new algorithms in artificial intelligence. Conversely, while grid cells are believed to play an important role in spatial computation and navigation, their specific role in the brain is not fully understood; building artificial navigational agents using the principles of grid cells could help illuminate their role in biology. In this thesis, we investigate the possible role of grid cells in "vector navigation", where agents navigate by calculating goal vectors using an internal coordinate system. We first develop a neural network able to perform vector navigation by reading out spatial coordinate information from grid cell populations. We then show that the proposed decoding mechanism can work over long distances, and that it can be integrated with other kinds of spatial information known from the hippocampal formation in order to enable the agent to traverse obstacles in complex environments. This demonstrates that grid cells can play the role of a coordinate system for vector navigation within larger navigational architectures, either in artificial (simulated or robotic) settings, or in the brain. We finally show that the proposed decoding mechanism remains functioning in the face of noisy and distorted grid cell signals, which is important for the model to be biologically plausible. Our results could inspire further neuroscientific investigation into grid cells' potential role in vector navigation. This shows that close interaction between neuroscience and artificial intelligence, e.g. on the topic of neural representations, might lead to valuable insights for both fields.

Preface

This thesis is submitted for partial fulfillment of the requirements for the degree of Philosophiae Doctor (Ph.D.) in Computer Science at the Faculty of Information Technology and Electrical Engineering (IE) at the Norwegian University of Science and Technology (NTNU). This work was funded by and undertaken under the auspices of the Department of Computer Science (IDI) at NTNU in Trondheim, Norway. Keith Downing was the main supervisor for this project, while Trygve Solstad and Pauline Haddow were co-supervisors.

The thesis takes the form of a paper collection, where the included papers have already either been published or submitted at conferences or in journals. For consistency and readability, the papers have been reformatted and typeset anew for inclusion in the thesis, and they therefore deviate visually and typographically from their published or submitted versions.

Acknowledgements

First and foremost, thanks to my supervisor Professor Keith Downing for introducing me to the world of research, for encouraging me to pursue a research degree and for all of his invaluable help during the course of this work. I am very grateful for his guidance and for all of our interesting discussions. Thanks to my co-supervisors Professor Trygve Solstad and Professor Pauline Haddow for helping supervise this work and for useful discussions and advice.

I am thankful to the Department of Computer Science for funding and supporting this work, and to my colleagues at the department for providing a friendly working environment. Thanks to Professor Neil Burgess for welcoming me to his group at the Institute of Cognitive Neuroscience (ICN) at University College London (UCL) for a research visit during the spring of 2017. Thanks to the group members there for their friendliness and for our many interesting conversations, and particularly thanks to Andrej Bicanski for our fruitful collaboration during and since my time there.

Last but not least, thanks to all my friends and family for being a constant source of encouragement and support during these years—you make it possible.

Contents

I Research Overview 1

1 Introduction 3
1.1 Understanding grid cells to benefit both neuroscience and AI . 3
1.2 Research questions 5
1.3 Overview of research conducted 8
1.4 Structure of the thesis 10

2 Background 11
2.1 Building neural networks to understand the brain, and vice versa 11
2.1.1 Neurons in brains and machines 11
2.1.2 Biological neurons 12
2.1.3 Artificial neurons 14
2.1.4 Artificial neural networks 14
2.1.5 Neural representations as a biology–AI bridge 16
2.2 Hippocampus essential to understanding natural intelligence . 18
2.2.1 The cerebral cortex—neocortex and hippocampus . . . 18
2.2.2 Role of hippocampus in navigation 21
2.3 Spatial neurons and computation in the hippocampal formation 22
2.3.1 Place cells, environmental context and remapping . . . 22
2.3.2 Grid cells and path integration 24
2.3.3 A wealth of diverse spatial information 25
2.4 Grid cells as a neural coordinate system 26
2.4.1 A metric for space and a GPS in the brain 26
2.4.2 Grid patterns and grid modules 26
2.4.3 Activity packets in twisted torus neural sheets 28
2.4.4 Multiple grid modules resolve the ambiguous grids . . 30
2.5 Grid formation and the role of grid cells 30

3 Research results **33**
3.1 Research Question 1: Decodability of grid cells 33
3.1.1 Reading out goal vectors directly from grid cells . . . 33
3.2 Research Question 2: Foundation for a navigational system . . 41
3.2.1 Navigating over arbitrarily long distances 41
3.2.2 Integrating grid cells with border cells and place cells . 45
3.3 Research Question 3: Plausibility of the decoding approach . . 48
3.3.1 Nested decoding of distorted grid modules 48

4 Discussion **51**
4.1 Summary of research contributions 51
4.2 Related work on the possible role of grid cells 54
4.3 Future directions and conclusion 56

II Publications **67**

A A passive mechanism for goal-directed navigation using grid cells (Edvardsen, 2015) **69**

B Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network (Edvardsen, 2016) **93**

C Long-range navigation by path integration and decoding of grid cells in a neural network (Edvardsen, 2017) **131**

D Navigating with grid and place cells in cluttered environments (Edvardsen et al., 2019) **157**

E Navigating with distorted grid cells (Edvardsen, 2018) **207**

Part I

Research Overview

What I cannot create,
I do not understand.

Richard Feynman

Chapter 1

Introduction

1.1 Understanding the role of grid cells for the benefit of both neuroscience and AI

The fields of Artificial Intelligence (AI) and neuroscience have had fruitful interactions throughout much of their histories [6–8]. Indeed, with a goal of building artificially intelligent computer systems, one should pay heed to the only known instance of real intelligence—the brain. Artificial neural networks were an early example of insights from neuroscience being applied for the pursuit of artificial intelligence, and this link between neuroscience and AI has now grown into a major field in its own right [9]. However, as neuroscience and AI over time have developed their separate traditions, goals, methods and vocabularies [8], the fields have diverged. State-of-the-art deep neural networks of today, although ostensibly based on principles from the brain, should more be thought of as based on mathematical optimization of cost functions than they are on specific architectures and algorithms grounded in biology. Yet as we continually learn more about the brain, there remains a large potential for putting those principles to use in artificial neural systems. How do neurons store and transmit information, i.e. what are the *neural representations* utilized by the brain? How does the collective activity of many neurons combine to support useful behavior? Might the contours of an algorithm be gleaned from the wealth of neuroscientific data [10]?

On the topic of neural representations, some of the most striking results from neuroscience are found in the hippocampal formation of mammals [11]. Signals recorded from neurons in this part of the brain—far removed from the raw sensory inputs to the brain and thus deep into the information processing hierarchy—offer a peek at the inner workings of high-level cognitive processes. The hippocampus in humans has long been known to be crucial for the formation of new long-term episodic memories [12]. Animal research on the hippocampus has focused more on navigation, also seemingly dependent on this area. It has even been argued that navigation and episodic memory could be two manifestations of the same underlying cognitive principles [13]. It is through this research that a variety of neurons with clear links to navigationally relevant variables have been discovered: place cells [14], border cells [15], head-direction cells [16], speed cells [17] and grid cells [18], all named for the various kinds of spatial information they convey. Of these, grid cells stand out as particularly captivating. These neurons activate in a clear relationship to the animal's location in space, yet the patterns generated are unlike any we would expect to see from the features of the external environment alone [19]. Specifically, these neurons are triggered whenever the animal crosses the vertices of an imaginary hexagonal lattice laid out in the two-dimensional plane—hence the name "grid cells". Such a hexagonal pattern might never have been experienced by the animal out in the real world, yet the brain has, through some process, seen fit to use the hexagonal grid structure to represent the animal's position in space [19].

Although the hexagonal grid pattern itself already appears intriguingly algorithmic, there is further evidence to stir our interest that these cells might reveal parts of the inner workings of a navigational algorithm in the brain. The grid pattern is always present, even in novel environments [18]. The pattern is maintained in the dark, which suggests that grid cells are supported by a *path integration* process that updates the representation based on internal self-motion velocity information [18]. Different grid cells exhibit different scalings and offsets of their grid patterns [20], which causes each individual location in the environment to have its own, unique combination of grid cell activity [21]. Taken together, these findings suggest that grid cells might implement a neural system for representing, updating and computing with two-dimensional coordinates in an entirely generic fashion [22]. In short, grid cells might embody a neural coordinate system for space.

Navigational circuits in the hippocampal formation are an active area of neuroscientific research, and there are vast numbers of questions yet unanswered about these spatial neurons in general and about grid cells in particular. This does not preclude, however, starting to attempt to apply what is already known about these neurons towards constructing artificial neural systems. The neuroscientific results so far have been sufficiently surprising to warrant an investigation into whether we can *build* something with these principles [23], if only from a high level of abstraction—such as neural representations—without any detailed biophysical simulations of the cells involved. Can we design artificial agents that navigate using principles derived from grid cells? Such an endeavor could bring new inspiration and insights into the design of artificial neural networks, and might also prove beneficial to neuroscience in return. By building a high-level model that utilizes the same spatial representations as grid cells, we can provide proofs of principle that grid representations are useful for solving certain tasks, and in such a way provide input to help guide and validate the trajectory of neuroscientific research [7].

1.2 Research questions

This section describes the overarching objectives for the research project. An overall Research Goal is stated first, whereafter this high-level goal is further concretized into three more specific Research Questions (RQs). These RQs are later used to organize the research content included in the thesis.

As outlined above, the starting point for this project was the neural representation known from grid cells, with its appealingly algorithmic nature. Are these neurons already sufficiently well-understood that we can start to build agents using these representations? Might we then learn new, useful principles for artificial intelligence and artificial neural networks through building such agents? Although an agent driven by such artificial neural networks will be a high-level model, necessarily somewhat disconnected from important biological details, it will provide a useful tool for exploring the possibilities enabled by this neural representation discovered in the brain.

From the neuroscientific standpoint, this systems-level perspective of how grid cells might be incorporated into a larger navigational circuit allows us to examine not just the mechanisms for *generating* these neural representations, but also to hypothesize how the information conveyed by grid cells might be interpreted and *used* by downstream neural networks (Fig. 1.1). Such models

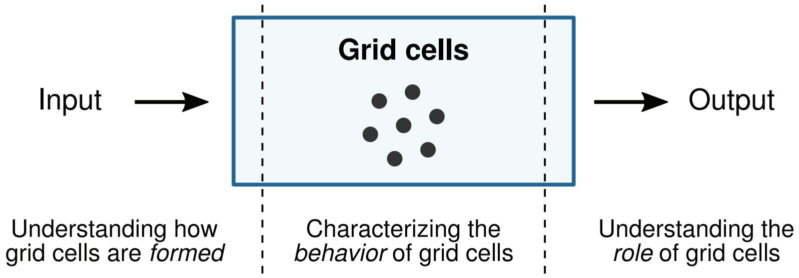

Input
Grid cells
Output
Understanding how grid cells are *formed*
Characterizing the *behavior* of grid cells
Understanding the *role* of grid cells

Figure 1.1: Grid cell research can proceed along many tracks. One simplified breakdown of the study of grid cells is to separate it into (a) characterizing the behavior of grid cells, e.g. through electrophysiological recordings of grid cells as animals face various navigational tasks and contingencies, (b) analyzing which inputs reach the grid cell population and attempting to understand how grid patterns form from these signals, and (c) exploring why grid cells should exist in the first place, by understanding which computations and behaviors are supported in downstream areas by their output signals—i.e. understanding which *role* grid cells play in the larger picture.

might better enable us to see where there are holes in our current understanding of the biological system. This motivates the following research goal and the several research questions that derive from it.

> **Research Goal** (Understanding the role of grid cells):
>
> Advancing our understanding of what grid cells
> might be useful for, and under which circumstances

As phrased above, this research goal is understood to encompass both the utility of grid cells to artificial agents and in the real brain—in general, what role grid cells might play in any larger information processing system. While this research goal encompasses and guides the whole research project, its scope is much greater than what can realistically be addressed by any individual project. However, it enables us to see how the project's research questions, more specific and focused, come about and fit into the larger picture.

The first research question investigated in this project was whether the coordinate information represented by grid cells can somehow be extracted for navigational purposes. If grid cells activate in a predictable, consistent fashion as a function of the animal's coordinates in the two-dimensional plane, might potentially the grid cell representation then also be “turned in reverse” to yield distances and directions between pairs of locations as encoded by populations of grid cells? Such a mechanism would then enable an agent to perform *vector navigation*, by heading along the straight-line path toward its goal.

> **Research Question 1** (Decodability of grid cells):
>
> Can grid cells be decoded by a downstream network, in order to make full use of the embodied coordinate system?

Provided a positive answer to this research question—as will be seen later—the remainder of the research project has been concerned with investigating further the possibilities enabled by grid cell decoding, as well its biological plausibility. After demonstrating that grid cells can indeed be decoded, the next research question addressed was the following.

> **Research Question 2** (Foundation for a navigational system):
>
> Can a grid cell decoder be the foundation for a larger agent architecture for navigational tasks?

If grid cells turn out to be decodable, this research question thus asks whether the decoded information would then be useful for navigation in practice. Even though the grid cell decoder would enable the agent to calculate directions and distances between locations, it remains to be shown that this information can then be put to valuable use in complex environments by a larger navigational architecture that can handle obstacles, etc. For this research question it becomes natural to look at the other cell types known to coexist alongside grid cells in the hippocampal formation, such as place cells and border cells.

The final research question addresses the biological plausibility of the work.

> **Research Question 3** (Plausibility of the decoding approach):
>
> Does grid cell decoding seem biologically plausible, given our current understanding of real grid cells?

For grid cell-based vector navigation to be considered a viable hypothesis as a possible role for biological grid cells, the decoding mechanism should not

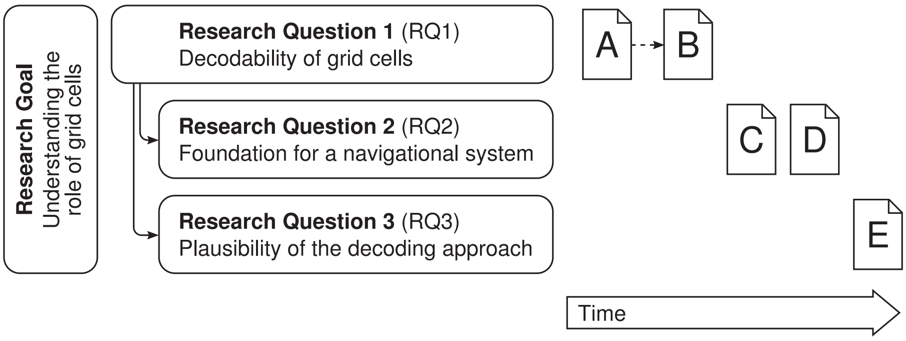

Research Goal
Understanding the role of grid cells
Research Question 1 (RQ1)
Decodability of grid cells
Research Question 2 (RQ2)
Foundation for a navigational system
Research Question 3 (RQ3)
Plausibility of the decoding approach
A
B
C
D
E
Time

Figure 1.2: Overview of the research conducted as part of this thesis. The individual research papers are ordered roughly chronologically (according to when the bulk of the research work was conducted), and shown grouped according to the main research questions they address. The arrows from RQ1 to RQ2 and RQ3 indicate that they build on the outcome from RQ1, while all of the questions have in common that they support the overarching Research Goal for the entire thesis.

make any unrealistic assumptions about the nature of the grid patterns. Biological grid patterns are imperfect versions of the idealized hexagonal grids often used in theoretical models. In particular, the grids experience various sorts of geometric distortions [24], and other perturbations (e.g. due to path integration-induced drift) should also be expected. Does grid cell decoding remain viable in face of these challenges?

1.3 Overview of research conducted

This section gives an overview of the research papers included as part of this thesis. Five papers, labeled Paper A–E, are included, all of which can be found in their entirety in the second part of the thesis. Fig. 1.2 lists the research questions and shows how the individual papers relate to them.

The first paper addresses Research Question 1, by describing and demonstrating a neural network that can perform vector navigation by decoding grid cells.

Paper A (Vegard Edvardsen, 2015):

Title:
A Passive Mechanism for Goal-Directed Navigation using Grid Cells
Published at conference:
2015 European Conference on Artificial Life (ECAL)

This conference publication, Paper A, was later invited for submission in extended form to a special issue of the journal *Natural Computing*. The extended version, Paper B, includes most of the material from the original publication as well as new background material on the ideas and principles behind the model, and underwent new peer review prior to publication. Both versions of the paper are included in the thesis for the sake of completeness, but for expediency the reader may choose to only read Paper B.

Paper B (Vegard Edvardsen, 2016):

Title:
Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network
Published in journal:
Natural Computing

The next paper addresses a shortcoming in the model presented in Papers A and B, by enabling the agent to navigate over arbitrarily long distances. This prepares the model for later integration into a larger navigational architecture, and as such supports Research Question 2.

Paper C (Vegard Edvardsen, 2017):

Title:
Long-Range Navigation by Path Integration and Decoding of Grid Cells in a Neural Network
Published at conference:
2017 International Joint Conference on Neural Networks (IJCNN)

Continuing the investigation under Research Question 2 (into whether a grid cell decoder can be the basis for a more fully-featured navigational system), the next paper presents a larger hippocampus-inspired navigation model with grid cells, place cells and border cells integrated into the same architecture. In contrast to the previous papers, this model is able to negotiate obstacles, enabling the agent to navigate in more complex environments.

> **Paper D** (Vegard Edvardsen, Andrej Bicanski and Neil Burgess, 2019):
>
> **Title:**
> Navigating with grid and place cells in cluttered environments
> **To appear in journal:**
> Hippocampus *(in press; accepted on July 19th, 2019)*

The final paper addresses Research Question 3, by investigating how the grid cell decoder fares with distorted and otherwise imperfect grid patterns. This is important to assert the biological plausibility of the model, and also relevant for potential robotic implementations that must handle noisy conditions.

> **Paper E** (Vegard Edvardsen, 2018):
>
> **Title:**
> Navigating with distorted grid cells
> **Published at conference:**
> 2018 Conference on Artificial Life (ALIFE)

1.4 Structure of the thesis

The remainder of this thesis is structured as follows. Chapter 2 presents useful background material on the motivation for this project and on the features of grid cells and other spatial neurons that are relevant to this research. Chapter 3 walks through each of the research questions and describes the main outcomes of the conducted research as it relates to those questions. Chapter 4 summarizes and discusses these contributions, before concluding the thesis.

Chapter 2

Background

2.1 Building neural networks to understand the brain, and vice versa

2.1.1 Neurons in brains and machines

Over a century ago, the idea took hold that the workings of the brain are the result of the collective action of cells known as neurons [25]. The human brain consists of on the order of one hundred billion neurons [25], which together implement the human capacity for intelligence. Ever since the endeavor of building artificial intelligence was initiated over half a century ago, *artificial neurons* have thus been leveraged as one of the means toward that end [9].

Just as there is a vast diversity in the family of neurons found in the brain, the pursuit of artificial neural networks has over the decades produced a great variety of artificial neurons and models. The recent resurgence of interest in artificial intelligence is underpinned by breakthroughs in the use of artificial neural networks within problem domains of such wide variety as image understanding [26], speech recognition [27], machine translation [28] and beating human intuition in challenging games such as Go [29]. This demonstrates the clear value of taking inspiration from natural intelligence when attempting to construct artificial intelligence in machines.

The converse is also true: given the right constraints and assumptions, results from artificial neural network models can help us understand the human brain. In order to decipher such a complicated system as the human brain, it is necessary to abstract whenever possible and to use simplified models to explore brain functionality across various levels of detail and realism [7]. Neural models can help us explore hypothetical situations in the system in a more detailed fashion than would be possible through mere thought experiments alone. Through seeking artificial intelligence we might therefore also better understand ourselves.

2.1.2 Biological neurons

Although biological neurons come highly varied, complicated and imperfectly understood, only a few main principles are needed in order to arrive at the canonical example of an artificial neuron. Biological neurons perform computation by receiving and transmitting signals with other neurons, and this communication occurs in a unidirectional fashion across a given neuron (Fig. 2.1a). That is, the neuron receives signals from upstream neurons, decides based on these inputs whether to transmit a signal of its own, and if so, conveys that signal to its downstream target neurons [25]. In this picture, the computational contribution from each individual neuron is comparatively modest, but, through communication, a large number of neurons can collectively perform sophisticated functions.

Neurons communicate through impulses called “action potentials”, which are waves of electrical release initiated at the cell body and propagated along the axon toward downstream neurons. The axon makes contact with downstream neurons at sites known as synapses, where the electrical impulse is converted into the release of chemicals known as neurotransmitters. When neurotransmitters diffuse across the gap between the two neurons at the synapse, the chemicals trigger mechanisms at the downstream neuron’s dendrite that cause the chemical to be converted back into an electrical signal. If the electrical potential in the downstream neuron attains a sufficient strength, it will in turn trigger a new action potential along its axon, and so on. The action potential is characterized by a brief spike in the electrical potential of the neuron, and is by itself not considered to convey any more information than that of its mere occurrence. By looking at the rate at which a neuron emits multiple spikes, however, we can speak of weakly or strongly activated neurons.

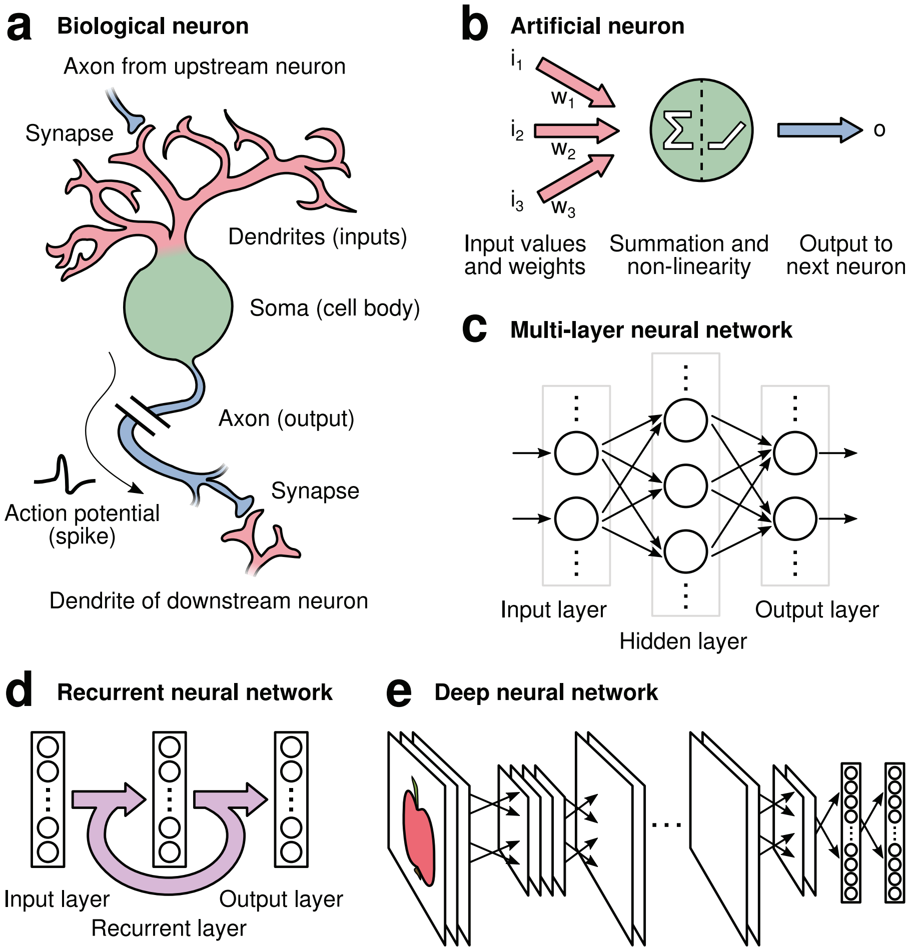


Figure 2.1: (A) Main components of biological neurons: Dendrites receive inputs from upstream neurons. Axon transmits action potentials (spikes) to downstream neurons, connecting at synapses. (B) Main features of artificial neurons: Upstream neurons are connected through weighted synapses. The input sum—passed through an activation function—determines the neuron's real-numbered activation value, which is transmitted to downstream neurons. (C) Artificial neurons become powerful when combined into networks, commonly arranged in fully connected layers. (D) Recurrent neural networks allow signals not only to flow between layers in a forward fashion, but also to loop back into earlier layers. (E) Central to the recently renewed interest in artificial neural networks is the newfound success in properly training deep neural networks using machine learning methods (deep learning).

2.1.3 Artificial neurons

While the brief introduction above leaves out many important details of biological neurons, not least of which how they learn, it introduces several key concepts we will recognize in their artificial counterparts [30]. The function performed by an individual artificial neuron is usually a simple summation of inputs combined with an *activation function* that performs a non-linear transformation of the input sum (Fig. 2.1b). The signals communicated between artificial neurons are usually real-numbered values interpreted as e.g. their firing rates, although more complicated variants where neurons communicate using individual spikes are also possible. Synapses are represented by connections between pairs of neurons and have real-numbered weights associated with them, representing the strength of the synapse.

Given the activation value i_n for upstream neuron n and synaptic weight w_n for the synapse corresponding to that input, the total input arriving at a given neuron can be expressed as the sum of $i_n \cdot w_n$ across all its upstream neurons. The spike-generating capacity of the cell body is represented by the activation function; a common such function is to threshold negative values to zero and otherwise let the values pass through unchanged, which can be represented as $f(x) = \max(0,\, x)$. The total calculation performed by a given artificial neuron can thus be expressed as simply as $o = \max\left(0,\, \sum_n i_n \cdot w_n\right)$.

2.1.4 Artificial neural networks

As in the brain, the true computational sophistication of an artificial neural network arises from the collective activity of many individual units and their interconnections. A typical configuration is to arrange neurons into layers, passing signals from one layer to the next in a fully connected fashion so that each neuron in a downstream layer receives connections from all neurons in the upstream layer (Fig. 2.1c). The first layer receives the initial input values for the computation, while the output values are read from the final layer of the network. In these sequential networks, because of the “feed-forward” nature in which each layer operates based only on the information provided from the previous layer, the network itself does not maintain any internal dynamics if the input is switched off: the network’s current output value is purely a function of its current inputs. However, by creating loops in the topology of the network, activity dynamics can continue inside the network even if the inputs are removed (Fig. 2.1d). These loop-back links, also known as *recurrent connections*, allow a form of short-term working memory due to how a signal can percolate through the network for many timesteps [31].

This accounts for how artificial neural networks (ANNs) can perform computation, but equally important is how these networks are obtained in the first place. A most striking capacity of the brain is its ability to acquire new skills and knowledge during the lifetime of an individual. This is also the main reason for ANNs' success, as these structures have proven amenable to powerful machine learning (ML) methods that automatically tune the synaptic weights [32]. In the past decade, these methods have successfully been applied to ever deeper networks, i.e. with many successive layers, due to advances in hardware, datasets and algorithms [33]. These *deep neural networks* (Fig. 2.1e) are now state-of-the-art in a wide range of application domains [33]; a common use-case is to recognize objects in images [26]. The sophistication of these object recognition networks is attributed to how the input is processed into ever more abstract representations as it propagates to deeper layers. While a neuron in early layers of the network might be responsible merely for representing an individual pixel or edge, neurons in deeper layers can detect more complicated, composite patterns in the input data.

On this issue of learning in neural networks, ANNs diverge from current understanding of biology: today's ML methods are based more on mathematical optimization of network output than they are on biological learning processes, themselves poorly understood. If ever we are to grasp how the brain implements true intelligence, understanding how the brain learns is essential. Any progress deciphering learning in the brain is thus also fruitful for AI.

There are yet more areas where we can study biological neural processing and apply our insights to artificial neural models. Such ANNs based on biological principles could even be useful to the field of biology in return. The potential reciprocal benefit between biology and AI becomes evident when we consider that complicated systems should be broken down into abstraction layers. The layers can then be studied separately, at levels of detail where the difference between biological and engineered systems might no longer be important. As discussed next, insights can then be shared between the two domains.

2.1.5 Neural representations as a biology–AI bridge

The notion of decomposing large, complicated problems into multiple levels of abstraction has many examples in both biology and engineering (Fig. 2.2a): Computer networking is best understood as a stack of abstraction layers [34], with lower layers responsible for, e.g., physical signal transmission, and higher layers only depending on generic capabilities from below. The network layer can thus handle traffic routing around the network without being concerned with maintaining proper voltage levels in copper cables, while further up the stack, an email client only depends on such high-level abstractions that it can easily work across a vast range of networking technologies. The hardware/software stack of today's computers similarly depends on abstractions [35], such that software developers can express their applications as code in a programming language, without regularly needing to worry about how the underlying hardware uses digital logic and transistors to make the computations come to life in a physical substrate. Conversely, as long as interfaces are adhered to, new hardware can be developed without changing any software.

The endeavor of understanding the brain also benefits from abstraction [36]. While the intelligent behavior produced by the human brain can ultimately be traced back to the effects of neurotransmitters acting on channel proteins in cell membranes, the macro-level behavior of a full neuron can be characterized independently of the lower-level molecular reality and thus be its own focus of study. Provided a good abstraction of the underlying neurons, the collective act of large numbers of neurons clustered into networks is then a yet higher-level concern that can be tackled separately, and so on.

Here a potential reciprocal benefit between biology and AI emerges. Computation in the brain depends on large groups of intercommunicating units, and this is also the main characteristic of ANNs. The two fields thus have a common interest in understanding how information is represented (at various stages of processing) in a fashion amenable to distribution among large numbers of comparatively simple units. Thus, *neural representations* as a layer of abstraction is relevant to both neuroscience and AI (Fig. 2.2b). Studying how neurons represent information in the brain informs us how to better build ANNs, and studying which representations emerge in trained neural networks can also guide neuroscientific research [7]. Indeed, neural representations

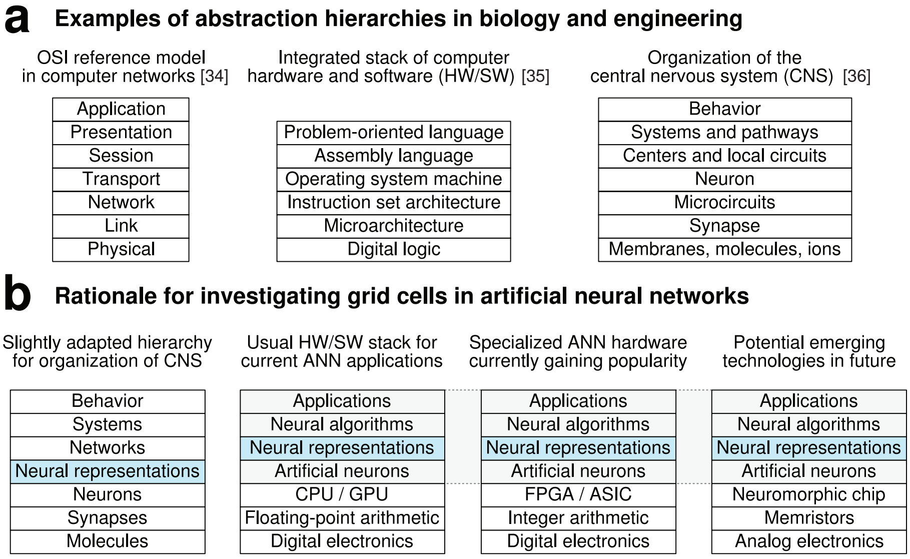

a
Examples of abstraction hierarchies in biology and engineering
OSI reference model in computer networks [34]
Application
Presentation
Session
Transport
Network
Link
Physical
Integrated stack of computer hardware and software (HW/SW) [35]
Problem-oriented language
Assembly language
Operating system machine
Instruction set architecture
Microarchitecture
Digital logic
Organization of the central nervous system (CNS) [36]
Behavior
Systems and pathways
Centers and local circuits
Neuron
Microcircuits
Synapse
Membranes, molecules, ions
b
Rationale for investigating grid cells in artificial neural networks
Slightly adapted hierarchy for organization of CNS
Behavior
Systems
Networks
Neural representations
Neurons
Synapses
Molecules
Usual HW/SW stack for current ANN applications
Applications
Neural algorithms
Neural representations
Artificial neurons
CPU / GPU
Floating-point arithmetic
Digital electronics
Specialized ANN hardware currently gaining popularity
Applications
Neural algorithms
Neural representations
Artificial neurons
FPGA / ASIC
Integer arithmetic
Digital electronics
Potential emerging technologies in future
Applications
Neural algorithms
Neural representations
Artificial neurons
Neuromorphic chip
Memristors
Analog electronics

Figure 2.2: (A) In both biology and engineering, the study of complicated systems benefits from abstracting across levels that can be treated independently from concerns above/below in the hierarchy: In the OSI model for computer networks [34], higher levels handle e.g. packet routing, unconcerned about lower-level details of signal transmission. A computer is physically based on digital logic in transistors, but chip architectures and instruction sets gradually abstract this into programming languages that hide the hardware details and allow the same software to run on different machines [35]. The study of the brain proceeds at multiple levels of organization [36]. While e.g. molecular processes in synapses are important, a higher-level treatment of how groups of neurons together perform computation needs only an abstract understanding of lower-level processes. (B) Abstraction levels underlie the rationale for studying grid cells in simplified models. Adapting the brain hierarchy above, *neural representations* can be highlighted as a bridge between individual neurons and the sophisticated behaviors of networks as the collective action of many simple units. Suitable neural representations are key to enabling distributed computation across neurons. This is also true in ANNs: while artificial neurons are much unlike biological ones, the principle of computation as the result of intercommunicating neurons remains. Studying how neural representations in the brain support sophisticated behaviors might yield new principles for ANNs. Computation in networks might become an ever more important paradigm in the future; neural algorithms can run wherever artificial neurons are implemented, whether this is a contemporary CPU/GPU architecture or specialized hardware using current/emerging electronics.

seen in early and late layers of deep neural networks trained for image recognition appear to have parallels to neurons found in the visual cortex of real brains [37, 38]. Hence, whether a neuron is implemented by biochemistry or transistors, the issue of neural representation is a shared concern.

Computer applications can thus benefit from "neural algorithms" that run on top of neural representations—e.g. obtained through ML methods. As a computational paradigm, though, ANNs are a complete departure from traditional software, and as such they also provide a promising roadmap toward new kinds of hardware. Contemporary computers are largely based on serial, deterministic execution of reliable operations on digital, bit-based representations of information. ANNs, however, are inherently parallel in their execution, usually operate on real-numbered (i.e. "analog") values, and, given their biological roots, are often robust to stochasticity and noise.

Artificial neural networks are therefore suitable for execution on computational devices quite different from typical CPUs (Central Processing Units), such as GPUs (Graphics Processing Units) [39], FPGAs (Field-Programmable Gate Arrays) [40] and ASICs (Application-Specific Integrated Circuits) [41]. In the future they could potentially also run on emerging electronics not based on digital logic in transistors, but e.g. using analog electronics such as memristors for components of the artificial neurons [42, 43]. These alternative hardware paradigms open an avenue to lower-power, more scalable computation than that accessible in traditional programming models. In order to fully utilize these hardware devices of the future, we might thus need inspiration from the brain on how to program them.

2.2 Hippocampus essential to understanding natural intelligence

2.2.1 The cerebral cortex—neocortex and hippocampus

There are many brain regions of interest, each with their separate responsibilities and specialities, when trying to understand how natural intelligence is implemented in the real brain. Of particular importance is the cerebral cortex, which contains two of the structures most closely linked to cognition and other

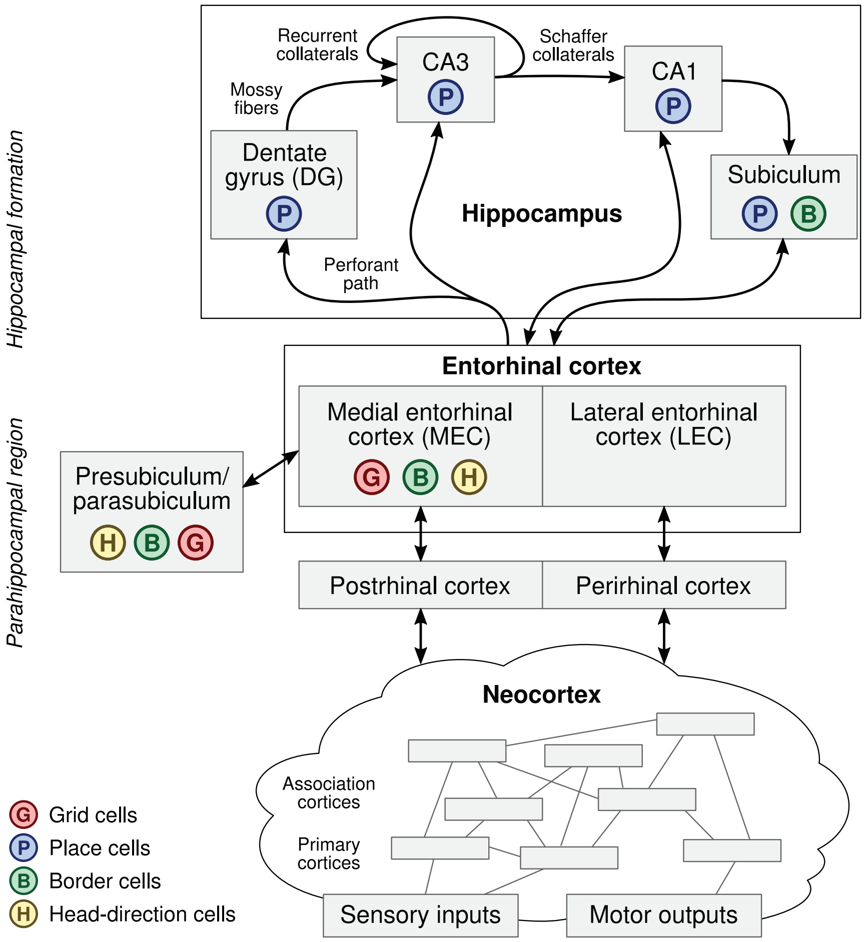


Figure 2.3: Simplified anatomical overview of some important brain areas in the hippocampal formation/neocortex, along with important connections between these areas. Neither the overview of areas nor connections should be considered exhaustive. Neocortex performs both low-level and high-level processing of sensory inputs of different modalities (vision/hearing/touch), as well as motor outputs. Processed information from the neocortex reaches the entorhinal cortex and eventually the hippocampus, within which information processing primarily proceeds along DG–CA3–CA1–subiculum and returns to the entorhinal cortex. Grid cells are found in the medial entorhinal cortex as well as pre-/parasubiculum [44], while place cells are found throughout the hippocampus. Schematic compiled from several sources [45, 46].

high-level information processing: the neocortex and the hippocampus. Neocortex, the prominent, wrinkled structure visible on the outside of the brain, divides into many subregions [25]. These neocortical areas are characterized as e.g. primary cortex or association cortex, depending on their particular input/output connections and types of information they process (Fig. 2.3). Initial processing of raw sensory information occurs in the primary cortices, e.g. primary visual cortex, primary auditory cortex and primary somatosensory cortex. Other neocortical areas, in turn, process information from these primary sensory areas into higher, more abstract representations, and eventually, information from multiple sensory modalities combine into association cortices. Neocortex also generates motor outputs through the motor cortex, and also on this motor-output side of the cortex there is a hierarchy of areas.

The neocortex is therefore a brain structure of major interest in trying to understand intelligence; this part of the brain contains circuitry responsible for both high-level and low-level processing of both sensory inputs and motor outputs, and distinct neocortical areas have also been identified that seem specialized for particular cognitive domains such as language processing, face recognition and planning [25]. Intriguingly, it has been suggested that these widely disparate responsibilities of the neocortex might be underpinned by common principles in their implementation [47], and that the neocortex might have a unified objective e.g. in trying to predict future world states [10].

The hippocampus, although not as prominent as the neocortex (being smaller in size and tucked away inside the temporal lobes), also plays an important role in several aspects of cognitive processing. The predominant research focus on human hippocampus has been its role in the formation of new memories. Patient H.M., after receiving bilateral hippocampal lesions as treatment for severe epilepsy, was famously afflicted with anterograde amnesia: the inability to form new memories [12, 25]. The prevailing view is that the hippocampus is responsible for quickly forming neural representations of episodic memories (i.e. episodes in an individual's life that might be committed to long-term memory), and that over time, these episodic memories are transferred into more permanent storage in the neocortex [48, 49].

This notion of a separate brain structure needed to quickly capture episodic memories, for later, gradual training of more long-term representations in the neocortex, even has appealing parallels in artificial neural networks. Training of these networks can suffer from catastrophic interference if the training examples are not presented in a sufficiently gradual and interleaved fashion [48, 49]. Understanding the hippocampus and its interactions with neocortex could therefore be key to understanding how the brain learns, and, in turn, essential to achieving true machine intelligence.

2.2.2 Role of hippocampus in navigation

Animal research on the hippocampus has focused more on spatial computation and navigation than on episodic memory. In the 1970s, electrophysiological recordings from rats revealed hippocampal neurons that primarily activated when the animal was situated in particular locations of the environment; these cells were thus named place cells [14]. Since that time, many other examples of spatial information have been found encoded by neurons in the hippocampus and adjacent areas, such as border cells [15], head-direction cells [16], speed cells [17] and grid cells [18] (Fig. 2.3; see next section).

Navigational capabilities, such as being able to explore physical space and then later return to an important location like the nest, should likely be considered a major reason for nervous systems—and eventually the brain—having evolved in the first place. That is, navigation could be an early function to appear in the brain, later to be supplemented by additional cognitive abilities. Indeed, while neocortex and hippocampus are brain regions common to mammals, there are homologues of the hippocampus in more distant species such as reptiles and birds [50]. These homologous brain areas appear to have a role in navigation in those species [51]. The hippocampus might thus have its origins as a navigational system, while the neocortex, as implied by the name, is a more recent development. The episodic memory capabilities of the hippocampus have indeed been suggested to be an evolutionary adaption of the earlier navigational system [13].

The reasons for wanting to study navigational processes in the hippocampal formation are therefore numerous. Navigation uniquely represents a high-level cognitive task where we nevertheless have access to neurons, in behaving animals, that encode relevant information in an interpretable fashion that is amenable to neuroscientific inquiry. Studying navigational processes in the hippocampal formation might thus give insights into memory, learning and cognition in general, and by association also help us understand the neocortex. Furthermore, changes in the entorhinal cortex (a “gateway” between the hippocampus and the neocortex) are linked to the onset of Alzheimer’s Disease [52]. The potential impact from understanding the cognitive processes in this part of the brain can therefore hardly be overstated.

2.3 Spatial neurons and computation in the hippocampal formation

2.3.1 Place cells, environmental context and remapping

Fig. 2.4 illustrates some of the spatial neurons in the hippocampal formation. First discovered were *place cells* (Fig. 2.4a), found throughout the hippocampus itself (Fig. 2.3), which activate whenever the animal is situated in a particular location within the larger environment [14]. Each plot in Fig. 2.4a depicts a top-down view of a hypothetical square enclosure, showing as a heatmap an idealized example of where an individual place cell might be active. The cells all have a particular spot in which they fire strongest, but the sizes of their “place fields” might vary. Some place cells might activate in more than one location within an individual environment (not shown here).

The specific firing location of each cell within a given enclosure will depend on the particular global environment currently inhabited by the animal. That is, a similarly-shaped box encountered in a novel setting, or *environmental context*, will trigger *global remapping*. The firing characteristics of place cells might then change completely: place field locations might move, some place cells might go silent, and previously silent cells might become active in the new context [53]. Thus, while a place cell can reliably recognize a previously visited location and reactivate whenever that place is revisited, these neurons do not provide a general way in which to relate distances and directions between given place fields. Place cells by themselves would therefore not be sufficient if, for example, you wanted to calculate the correct direction to get to a far-away goal (due to how these geometric relationships between pairs of place cells change during global remapping).

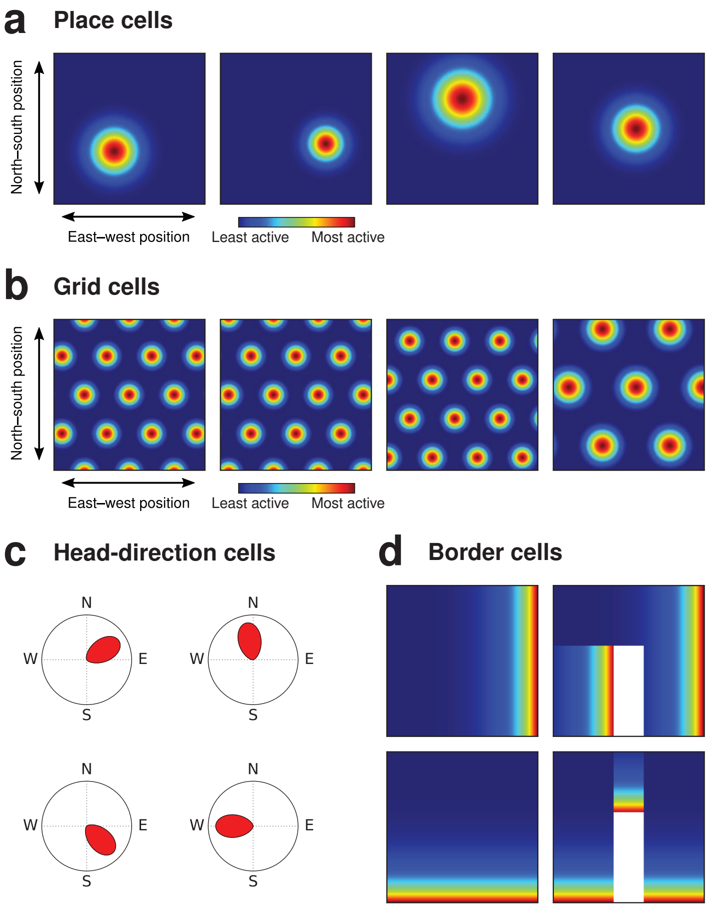


Figure 2.4: Idealized illustrations of spatial neurons in the hippocampal formation. (A) Four different place cells, each plot a top-down heatmap of where that cell might activate within the same 2 m side square box. (B) Whereas individual place cells only activate in a few spots per environment, grid cells activate in a multitude of locations. The first three grid cells have a grid scale (see Fig. 2.5a) of 0.6 m, while the fourth cell has a scale of 0.9 m. (C) Head-direction cells respond to the animal's head direction, irrespective of location, in an allocentric fashion (relative to a global reference frame). (D) Border cells respond to obstacles in a given allocentric direction. The first two heatmaps show a border cell tuned to eastern obstacles, first in an empty square box and then with an extra obstacle protruding from the southern wall. The next two plots similarly show a border cell tuned to southern obstacles.

2.3.2 Grid cells and path integration

Even if place cells did not exhibit global remapping, they would still be unsuitable for the goal vector calculations alluded to above, as the navigation mechanism would essentially have to learn how to navigate between all possible pairs of known locations—an exceedingly expensive process. However, place cells are not the only neurons to convey positional information, and indeed it is unclear whether the place cell system by itself would be able to accurately track the animal's current location. Place cells do receive sensory inputs through the lateral entorhinal cortex [54], so they might be able to determine the animal's location based on sensory configuration alone. However, place cells correctly reflect the animal's position even in sensory deprived conditions such as darkness [55]. The maintenance of these neurons therefore likely involves a path integration process, where the animal uses its speed and direction to update an internal representation of its total displacement from the last known point of reference. This presumed path integrator is unlikely to be implemented in place cells, due to global remapping as outlined above.

A different candidate for this hypothesized path integration system was found a little over a decade ago, with the discovery of grid cells in the medial entorhinal cortex [18]. These neurons activate not in single, individual spots, but in a multitude of firing locations—distributed across the environment in a hexagonal pattern that extends throughout all available space (Fig. 2.4b). By this fact alone, the grid cell system appears to represent spatial information in a more general fashion than the more environment-dependent place cell system. Like place cells, grid cells are able to maintain their firing patterns in the dark [18]. Grid cells are active in all environments, and grid patterns are expressed immediately in novel environments [18]. Furthermore, whereas place cells experience global remapping across environments, the mutual spatial relationships of grid cells' firing patterns are maintained across environments [56]. That is, while grid cells might shift and rotate their activity patterns between environments, they all do so in a coherent fashion (within modules, see next section) [56]. Thus, downstream networks that utilize geometric information from the grid cell system should retain proper function across all environments, without requiring any relearning of environment-specific relationships (as might be needed if solely relying on place cells).

2.3.3 A wealth of diverse spatial information

As the medial entorhinal cortex provides major inputs to the hippocampus, and for the reasons outlined above, grid cells are a viable candidate for the path integrator that supports the place cell system. If grid cells are indeed involved in path integration, we require inputs to grid cells that encode the direction in which the animal is headed and how fast it is moving. Reassuringly, the hippocampal formation does in fact harbor so-called head-direction cells: neurons that activate predominantly when the animal is facing a given allocentric direction (i.e. relative to a global frame of reference, as opposed to egocentric; Fig. 2.4c) [16]. Movement speed is also represented, e.g. in the recently-reported speed cells in the medial entorhinal cortex [17]. All information necessary for path integration is therefore available within the hippocampal formation. Neural representations for other kinds of spatial information have also been found, such as in border cells, which respond to borders in a particular allocentric direction in the environment (Fig. 2.4d) [15].

The discussion above only concerns patterns in neural *firing rates* within the hippocampal formation, but the region harbors yet more richness in how navigational information is encoded. Place cells notoriously exhibit a phenomenon known as *phase precession* [57], where place cells that are ahead of or behind the animal in its current path fire their spikes at different times relative to the background theta rhythm, thus encoding spatial information also in the precise spike timing. Phase precession is also exhibited by grid cells [58]. Moreover, when an animal is at rest or asleep, sudden bursts of activity known as *replay* might reactivate place cells to "play back" paths previously traveled in the environment [59, 60]. This phenomenon could be involved in processes of memory or planning [61], including a possible role in navigation [62].

In sum, the picture emerges of a hippocampal formation rich in both the different kinds of spatial information it encodes and how this information is represented. Even though the information involved is of a highly abstract and deeply processed nature, these neurons produce such strikingly interpretable neural representations. Spatial computation in the hippocampal formation is therefore a good platform for exploring the implementation of high-level cognitive algorithms in the brain.

2.4 Grid cells as a neural coordinate system

2.4.1 A metric for space and a GPS in the brain

Grid cells thus appear to implement a path integrator that functions across all environments, as outlined above. They are often said to provide a “metric for space” [63], due to how they provide for geometric computations on spatial position information, and in more popularized terms, grid cells have been said to embody a “GPS” in the brain [64]. This all alludes to grid cells implementing a spatial coordinate system, wherein grid cells (a) can represent arbitrary two-dimensional coordinates with unique activation patterns, (b) can update this representation through path integration, and (c) do all this with neural representations conducive for geometric computations—such as calculating vectors between locations—in downstream networks. To elucidate the basis for these claims that grid cells provide a neural coordinate system for space, we will next review some of the main properties of these neurons.

2.4.2 Grid patterns and grid modules

The pattern of an individual grid cell’s activity across space can be succinctly described by three properties (Fig. 2.5a). The characteristic hexagonal pattern has a *scale* (the distance from one location of peak activity to the next), an *orientation* (the alignment of the grid pattern’s axes relative to a frame of reference) and a *phase* (the two-dimensional offset of the pattern from a given point of reference). Although there are other factors that can affect the grid pattern, such as *grid distortions* (see Section 3.3), as a first approximation the behavior of any grid cell can be described by these three parameters.

What is the utility of having such a neuron? Hypothetically, if an animal only has one single grid cell, the animal will know whenever that neuron is active that it is located in one of the hotspots of activity in the grid cell’s hexagonal pattern: a highly ambiguous signal to begin with. Whenever the neuron is *not* active, the animal will know even less about its current location, as it will only know that it is *not* situated in one of the hotspots. It is clear that more information is needed for the grid cell signal to be truly useful.

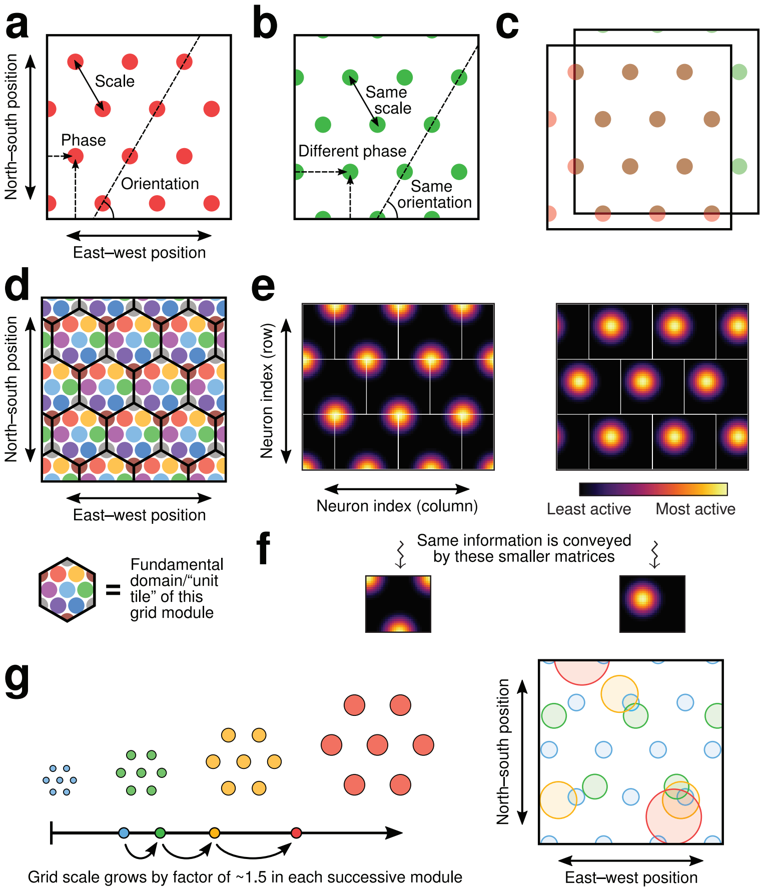


Figure 2.5: (A) Scale, orientation and phase of grid patterns. (B) Cells in the same grid module have the same scale and orientation. (C) Patterns in A and B overlap when one is shifted relative to the other; scale and orientation is thus the same. (D) With enough cells in a module, all phases are covered without any inactive areas (nine cells shown, one color each). Tiling the plane so each cell only has one field per tile, the module can tell the location within the tile—but this is ambiguous unless the correct tile is known. (E) A *neural sheet* organizes the cells of a module into a matrix arranged by grid phase. Plotting each cell's firing rate as a pixel reveals a snapshot of all concurrent activity in the module. (F) Phases will repeat in large neural sheets, i.e. some pixels are redundant. Non-redundant sheets contain exactly one activity packet. Its position in the matrix tracks animal position within the unit tile. (G) The brain has multiple modules of increasing scale, with a fixed ratio between modules [20]. Information from several modules can resolve ambiguity in a single module (in the rightmost panel, the activity from all modules overlaps in only one location).

However, individual grid cells do not operate in isolation, but participate in groups with other grid cells. Out of multiple grid cells observed in the same recording session, several of these co-recorded neurons might have activity patterns that neatly complement each other [20]. For example, Fig. 2.5b shows an illustration of a different grid cell that might coexist with the cell in Fig. 2.5a. This second grid cell has the same scale and orientation, differing only in its phase. That is, the only difference in behavior between these two grid cells is that their spatial activation patterns are offset from each other (Fig. 2.5c). Such grid cells, with the same scale and orientation but potentially different phases, are said to belong to a *grid module* [20].

The importance of grid modules is illustrated in Fig. 2.5d, which shows how a module can cover all possible phases such that there is always an active grid cell. The drawing depicts nine grid cells that might belong to the same module, with each individual cell assigned a unique color. With only this few number of grid cells, there is already sufficient "ground coverage" to ensure that all of the areas of inactivity in the spatial firing patterns of the example grid cells in Fig. 2.5a and Fig. 2.5b have been covered by other grid cells (with different phases) from the same module. This means that there is always an active subset of neurons from this grid module, no matter where the animal might be located. Combined with the evidence that grid cells maintain their activity patterns in darkness, are active in all environments and are immediately active even in novel environments without any advance familiarization, the signs thus point toward grid cells implementing a general system for representing and computing with spatial coordinate information in the brain.

2.4.3 Activity packets in twisted torus neural sheets

To visualize how grid cells in a particular grid module keep track of the agent's current location, we can plot the concurrent activity of all cells in the module as a "neural sheet" [65]. The neural sheet is a two-dimensional matrix where each cell is assigned a position (row/column) on the basis of the two-dimensional phase parameter of its grid pattern. For example, a grid cell whose spatial activity pattern is shifted east of another grid cell might be located to its right in the neural sheet, while a different grid cell offset south might be located down in the sheet. (Note that this does not correspond to any physical arrangement found in the brain. Neural sheets are merely a conceptual tool for understanding how grid modules operate in the abstract.)

By plotting the neural sheet so that each pixel shows the current firing rate of one individual neuron, we might thus get a result as in Fig. 2.5e: Neurons with similar phases, i.e. overlapping grid patterns, will be active at the same time, and these neurons are located closely together in the neural sheet. The activity in the network therefore clusters into *packets* of activity. When the animal moves around the environment—causing different grid cells to turn on and off—these packets will move across the neural sheet in a way that reflects the actual displacement of the animal (e.g. the animal might have moved slightly south-east between the left and the right neural sheets in Fig. 2.5e).

The neural sheets in Fig. 2.5e contain multiple activity packets, because some of these grid cells, while ostensibly having different grid phases, are in practice equivalent. For example, if the grid scale is 60 cm and two grid cells have the same parameters except for a difference in grid phase of 60 cm (along a grid axis), then these two cells will in effect have the same grid pattern. The matrices in Fig. 2.5e therefore contain redundant information, as multiple co-active cells are included. We can trim the neural sheet to a smaller matrix and still retain a representation for all possible grid phases (modulo the periodicity of the grid scale). The white lines superimposed on the neural sheets in Fig. 2.5e show how they can be subdivided into multiple identical copies of a smaller, more parsimonious matrix. This smaller, non-redundant neural sheet always contains exactly one packet of activity, and the location of this packet within the matrix reflects the agent's location within the environment.

Upon reaching an edge of the sheet, the packet re-emerges on another side, so that in total, the full activity packet is always preserved within the matrix. However, this "wrap-around" behavior does not adhere to the typical torus topology often seen in various applications (where the north/south and the east/west edges of the matrix are connected together). Rather, because grid cells activate in a hexagonal pattern, the neural sheets are connected along the edges according to a *twisted torus* topology [66]: the upper-left and upper-right corners wrap around to the lower center of the matrix, and vice versa (compare left and right matrices in Fig. 2.5f).

2.4.4 Multiple grid modules resolve the ambiguous grids

A major caveat is that grid cells' spatial activation patterns repeat—indeed, that is their defining characteristic. The information obtained from grid cells in a single grid module is thus *ambiguous*, as the behavior of each neuron in the module repeats for every "unit tile" across the environment (Fig. 2.5d). The position of the activity packet within the neural sheet corresponds one-to-one with a physical position within the grid module's unit tile, but if you do not know *which* unit tile is the correct one, there is a multitude of locations that the module might be interpreted to convey.

However, if we can assume that the correct unit tile is already known, then the information from the grid module can help the animal further localize within that tile. Evidence from the brain is that this information might in fact be available. Not only do grid cells cluster into modules, but each successive module has a larger grid scale than the previous one [20]. There appears to be a constant ratio of grid scales between successive modules, suggested to lie around the range of 1.4 to 1.7 by experimental data and theoretical considerations [20, 67–69]. The sequence of grid scales in an individual animal thus follows a *geometric progression*: as more grid modules are added to the system, the scale of the largest module increases exponentially. Taking information from all grid modules into account, the ambiguity in the smallest-scaled grid modules could thus be resolved. Observe e.g. in Fig. 2.5g, how the activity of grid cells from four successive modules overlaps in only one part of the environment. See Section 3.2.1 for more details on the different ways in which this resolution of ambiguity could conceivably happen.

2.5 Grid formation and the role of grid cells

After having introduced some of the main characteristics and properties of grid cells above, we now turn to the mechanisms behind them and what the purpose of these neurons could be (recall Fig. 1.1). How might these neurons form their characteristic grid patterns across space, and what role might grid cells play in the larger, overall function on the brain?

Many models have been proposed to explain how grid cells might arise, and two main classes of computational models emerged early on [70]. Both are based on the assumption that path integration is an essential aspect of the grid formation process: *Oscillatory interference models* [71] hypothesize that grid cells receive inputs from other path integrators further upstream, that accumulate the animal's displacement along three axes separated by 60°. The hexagonal pattern of grid cells is proposed to arise at the intersection of periodic activity along these three axes. *Continuous attractor networks* [65, 72, 73] propose that path integration occurs directly in the grid cell population. Grid cells are assumed to participate in grid modules, so that they make up neural sheets carrying activity packets (Fig. 2.5e). These packets are made to respond to velocity inputs by moving around the neural sheet, thus implementing path integration. As long as the movement of these activity packets in "neural space" exactly corresponds to the animal's movement in real space, the hexagonal pattern in the neural sheet will gradually be revealed through the activity of individual neurons across space, hence behaving as grid cells.

Continuous attractor networks can thus account for how the hexagonal grid pattern arises, assuming a similar pattern already exists in the neural sheet. This, in turn, is explained as follows: The grid cell network is connected recurrently, so that each grid cell in the network is inhibited by activity in neurons in a certain radius around it in the neural sheet. The synapses of each cell are assumed to be configured to reject activity from other grid cells with similar phase offsets, i.e. located within a certain range of distances in the neural sheet. The activity in the network will then spontaneously assemble into discrete packets of activity, and these packets will push away from each other until eventually they have distributed into a hexagonal pattern (e.g. Fig. 2.5e). According to continuous attractor models, hexagonal grid patterns are thus the result of a self-organizing process of activity packets repelling each other.

Later models have emphasized the fact that grid cells receive other classes of inputs besides self-motion velocity information, such as spatially modulated inputs from place cells. Simulations show that such inputs can be sufficient to generate grid patterns, e.g. through adaptation processes in the grid cell network [74] or learning processes related to Principal Component Analysis [75]. The issue of how grid cells are formed has thus not been settled, and represents an area of research in which theoretical work, neural network simulations and animal electrophysiological recordings go hand in hand.

However, regardless of how grid cells are formed, the separate question of how grid cells are *used* can be investigated independently.

Ever since the discovery of grid cells, a major line of interest concerning the role of these neurons has been the recognition that they produce unique population activity patterns for all visited locations across space. Grid cells can then e.g. provide coordinate inputs to place cells (through projections from entorhinal cortex to hippocampus; Fig. 2.3), enabling place cells to correctly activate based only on the path integrated movement history of the animal. Place cells would thus remain able to activate correctly in the absence of sensory inputs. However, even if sensory inputs were present, a coordinate-based input from grid cells could be important. Novel locations that appeared similar to previously visited locations would still be distinguishable by the coordinates conveyed by the grid cells, and conversely, if a previously visited location had changed since the last time the animal visited that place, it would still be able to activate the correct place cell based on the coordinate information [22].

This view thus emphasizes a possible role for the grid cell system as a path integrator. The required mechanism for reading out the grid cell signal for this purpose could be relatively straight-forward. By forming synapses with concurrently active grid cells across several modules of increasing scale, and forming inhibitory connections with other place cells, a given place cell would win the competition among place cells if and only if there were a good match with the memorized grid cell activity. The place cell would thus activate in the correct location, based on the grid cell inputs alone [21, 76].

However, might grid cells potentially be used for more than merely assigning unique population codes to spatial locations? Grid cells exhibit mathematically appealing—almost algorithmic—patterns that can be described by simple formulae and extrapolated across unknown space. An exciting question is whether this accessible structure in the grid cell signal can then be "turned in reverse": not only extrapolating how grid cells should update along a given velocity vector, but, given the grid cell activity for the current location and the goal, reversing that path integration process to extract the correct goal vector. This capability, of decoding the grid pattern to get the correct direction of travel in order to reach a specific set of "grid coordinates", could be used for *vector navigation* [77]. That is the topic for this thesis: investigating how grid cells might be decoded for vector navigation, and under which circumstances.

Chapter 3

Research results

3.1 Research Question 1 (RQ1): Decodability of grid cells

3.1.1 Reading out goal vectors directly from grid cells

The first research question asks whether it is at all possible to extract goal vector information from grid cells by decoding neural activity from the grid cell population. Earlier work [78] had proposed that one option for navigating with grid cells would be to use the grid cell system to simulate ahead from the agent's current location in various directions, using a mechanism that could be thought of as "virtual path integration". If one of the simulated paths fortuitously triggered the goal place cell, then the agent would know the correct goal direction. However, given the mathematically structured way in which grid patterns encode coordinate information, and the appealing interpretation that grid cells perform path integration by shifting an activity packet around in a neural sheet (Section 2.4.3), a prudent question was whether goal vector information could be extracted directly from grid cells without requiring time-consuming path simulations as in the aforementioned work. We are thus not merely interested in whether grid cells can support vector navigation, but whether grid cells can be read out directly—i.e. *decoded*—toward that end.

This is the question investigated in Paper A (later extended into Paper B). A positive answer to this research question subsequently lead to RQ2 and RQ3, so the remaining papers of the thesis (Papers C–E) also build upon these results. Papers C and D improve and extend the model to perform in a wider

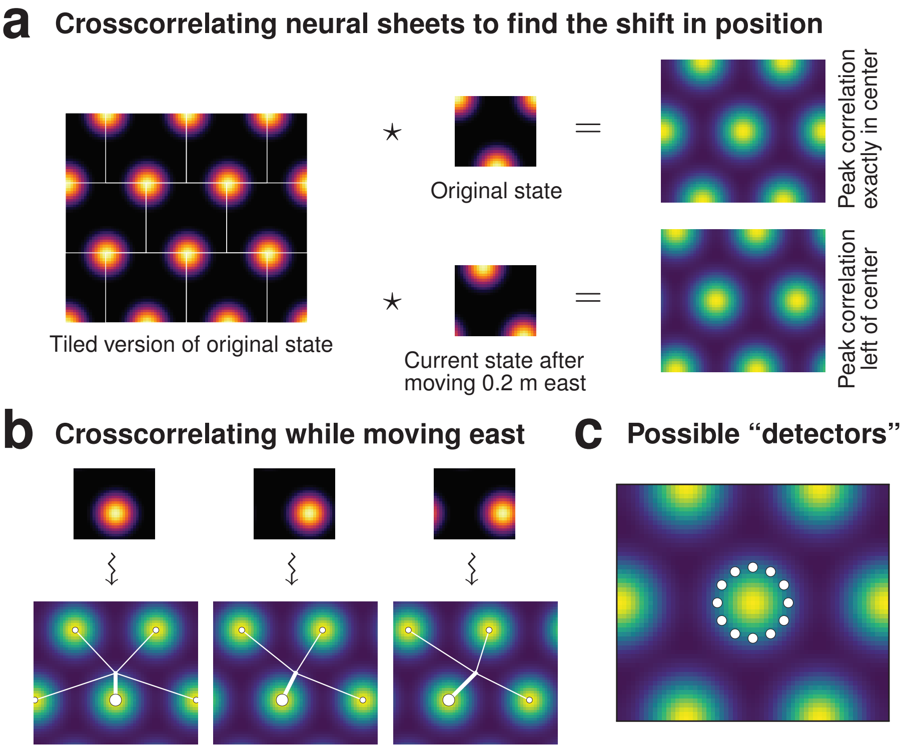

a
Crosscorrelating neural sheets to find the shift in position
⋆
=
Original state
Tiled version of original state
Peak correlation exactly in center
⋆
=
Current state after moving 0.2 m east
Peak correlation left of center
b
Crosscorrelating while moving east
c
Possible "detectors"

Figure 3.1: (A) Finding the shift in position. Top center matrix shows initial state of a neural sheet; when crosscorrelated with itself, the crosscorrelogram (top right) has a peak in the center, thus the best fit is attained with no offset. Crosscorrelation is performed against a tiled version, adhering to the twisted torus topology (left matrix, tiling shown by white lines). Bottom center matrix shows state after agent moved 0.2 m east; when crosscorrelated with the tiled matrix, the result (bottom right) shows the centermost peak off to the left, implying the agent should move west to reach the goal. (B) Three more examples of neural sheets crosscorrelated against the tiled matrix above, with the centermost correlation peak highlighted. First case is after agent moved 0.2 m north of initial location, second case then shows 0.1 m eastward, and third case shows yet 0.1 m further eastward. Centermost peak turns from south to south-west, as expected. (C) The decoding mechanism is based on finding the direction of the centermost crosscorrelogram peak. This can be done without calculating the full crosscorrelogram, but e.g. by calculating only a few of the correlation values in a circle around the center pixel, using them as "detector" pixels.

range of situations (RQ2; next section), and Paper E addresses the biological plausibility of the navigation model, specifically as it relates to the challenge of *grid distortions* known to occur biologically (RQ3; Section 3.3).

Although grid cell decoding was first addressed in Papers A and B, the model evolved over the course of the project. Paper E, in its investigation of grid distortions, used a simplified version of the model previously developed throughout Papers A–D. For example, rather than using continuous attractor networks to generate grid cells (as in the earlier papers), the model in Paper E explicitly generates grid activity directly from the agent's coordinates. It also uses a simpler version of the decoding mechanism itself. For ease of presentation, we will introduce the principles behind that final iteration of the decoding mechanism, noting that it shares many similarities with the model from the earlier papers. As the following account describes specific principles from the decoder in Paper E, we refer to that paper for more details, and to Papers A and B for the earlier version of the grid cell decoding model.

Finding the activity packet's displacement in the neural sheet

As seen in Section 2.4.3, when neurons in a grid module are arranged in a neural sheet, there is a packet of activity that moves around the neural sheet in proportion to the actual displacement of the animal out in the real environment. The challenge of decoding a goal vector from one grid module is therefore a question of finding the correct displacement of the activity packet between the current state of the neural sheet and the desired goal state. Fig. 3.1a shows such an example scenario in a given grid module, where the small, top center neural sheet shows the initial state of the module at the outset of the navigation trial. After the agent moved 0.2 m east, the neural sheet looks as in the bottom center. Assuming the agent now wanted to return to the initial location, the task would be to properly detect the direction of the displacement between the initial/goal state (top center) and the current state (bottom center).

In order to determine the correct shift between these two matrices, we can perform a *crosscorrelation* operation (indicated by star symbols in Fig. 3.1a), which evaluates a correlation value for each possible shift between the two operand matrices. However, in this process of crosscorrelating grid module neural sheets, we need to properly account for the twisted torus wrap-around behavior at neural sheet boundaries (see Section 2.4.3). Therefore, the crosscorrelation is not simply performed against the "small version" of the original neural sheet, but against a larger version where we have *tiled* the original neural sheet into a "brick wall" that reflects the twisted torus topology (large, leftmost matrix in Fig. 3.1a; cf. Fig. 2.5e). With the original neural sheet matrix thus correlated with itself, we get a result as in Fig. 3.1a, top right.

Each pixel in this matrix resulting from the crosscorrelation operation, shows the correlation value for a particular offset location of the small sheet placed atop the large tiled sheet (yellow and blue indicating high and low correlation, respectively). As expected, there is a peak in the correlation values in the center of the matrix, indicating that the small neural sheet has a maximum correlation when overlaid in the center of the large neural sheet. Indeed, this operation represents a comparison of the initial neural sheet matrix with itself, so there should be no shift indicated between these two matrices. When the current-situation neural sheet is crosscorrelated similarly (bottom right), the peak correlation is no longer in the center, but has moved slightly to the left of the center. Thus, to find the best fit for the current neural sheet, it should be overlaid slightly to the left of the center in the large neural sheet. This signals that the current neural sheet encodes a location to the right of the goal neural sheet, and that the correct goal direction is therefore to the left.

Fig. 3.1b shows three more examples of neural sheets compared to the large neural sheet from Fig. 3.1a. Superimposed on each crosscorrelation matrix is also a visualization of which peak is closest to the center and thus indicative of the likely goal direction. The first case shows the neural sheet with the agent located slightly north of the initial goal location, so the peak correlation should be found when the current-state matrix is shifted downward during crosscorrelation. The crosscorrelogram does indeed show a centermost peak that signals a southward goal direction. In the two successive matrices, the agent progressively moved east, causing the goal direction indicated by the crosscorrelogram peak to gradually turn westward as expected.

Detector neurons to determine the best matching direction

The introduction above describes the main idea behind our decoding approach, namely finding the best matching offset between the current situation in the

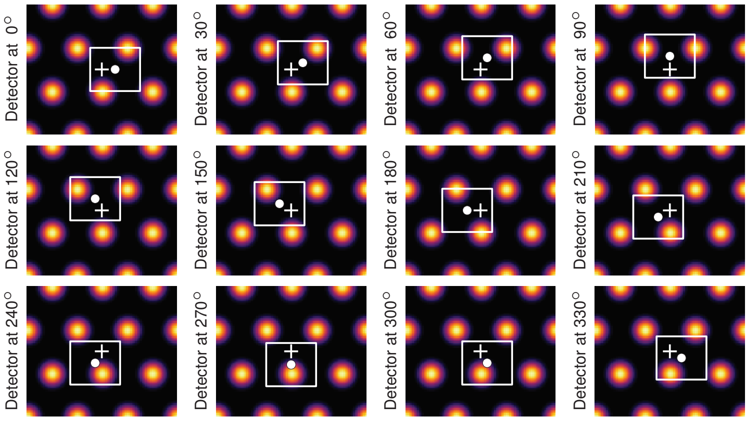


Figure 3.2: Each pixel in the crosscorrelograms in Fig. 3.1 is the result of an element-wise multiplication of the current matrix with a slice of the original, tiled matrix, where the offset of the slice corresponds to the pixel's position in the crosscorrelogram. To implement the 12 detectors illustrated in Fig. 3.1c, we therefore extract 12 slices from the tiled neural sheet, offset from the center by a fixed distance in equally spaced directions. Each plot illustrates the slice sub-matrix for a particular detector (pluses indicate the center of the tiled matrix, and circles the slice center).

neural sheet and the target state. The process can, however, be simplified: While there are several peaks in the crosscorrelogram (due to the repeating nature of the grid pattern), we are only interested in the one closest to the center. Moreover, we do not need the exact offset value for that peak correlation, but only its direction from the center. Finally, note that the correlation increases gradually as you get closer to the peak pixel in the matrix. Consequently, we do not need to perform the full crosscorrelation to extract the goal direction, but only to monitor the values of a few strategically located "detector" pixels arranged in a ring surrounding the center point of the crosscorrelogram (Fig. 3.1c). If we know the value at these pixels, we have sufficient information to determine the direction of the centermost crosscorrelogram peak.

The peak correlation in Fig. 3.1c is in the center, thus all detectors will observe the same correlation value—but, in cases with an off-center crosscorrelogram peak, such as in Fig. 3.1b, the detectors on one side of the ring will be more strongly activated than the other side. The detectors are thus able to sense the proximity of the crosscorrelation peak in their respective directions away from the center of the matrix, and this is the basis for our decoding mechanism.

What, then, is the computation underlying these pixels? Each pixel shows the correlation for a given offset between the current neural sheet matrix laid atop the target matrix, with the correlation itself calculated simply as the element-wise multiplication of the two matrices. Fig. 3.2 shows the 12 sub-matrices that would thus be used to calculate the correlation value at the 12 different detector locations in Fig. 3.1c. To calculate the ring of detector pixels, we can then take the current neural sheet and compare it by element-wise multiplication to each of these 12 matrices. Each such matrix in Fig. 3.2 can be thought of as a detector *neuron*, onto which converges a multiplicative synapse from each pair of corresponding cells from the current and target neural sheet matrices. Each detector neuron is thus responsible for comparing the current state of the neural sheet to a "template state" signaling a particular goal direction.

In Fig. 3.3a, the 12 detectors have been arranged in a ring according to their respective goal directions. The radiating bars each show the activation value of a detector neuron, given the situation in the current neural sheet from Fig. 3.1a (same matrix is also depicted in the middle of the figure). The detectors pointing west are evidently more strongly activated. To calculate the final goal direction, we take the vector sum of unit vectors pointing in the direction of each detector neuron, weighted by their respective activation values (i.e. a form of population vector average). The black semi-circular notch in the figure shows that the final goal direction, calculated this way, correctly indicates a goal direction west from the current location. Fig. 3.3b and Fig. 3.3c show two more decoding examples, corresponding respectively to the first and second situations in Fig. 3.1b. In the first example, the current location is north of the goal, and the goal is indeed correctly decoded to be south. In the second example the agent has moved east since the situation in the first example, and we see that the decoded goal direction now has turned west, as expected.

Note that the detectors are not tied to any particular goal location. Although they have been visualized in Fig. 3.2 and Fig. 3.3abc to consist of slices of neural sheet activity for a particular goal location, the mechanism would work the same if the underlying target neural sheet indicated another location. The template matrices assigned to each detector are not static snapshots of neural sheet activity, but represent a specific reconfiguration of the subset of active

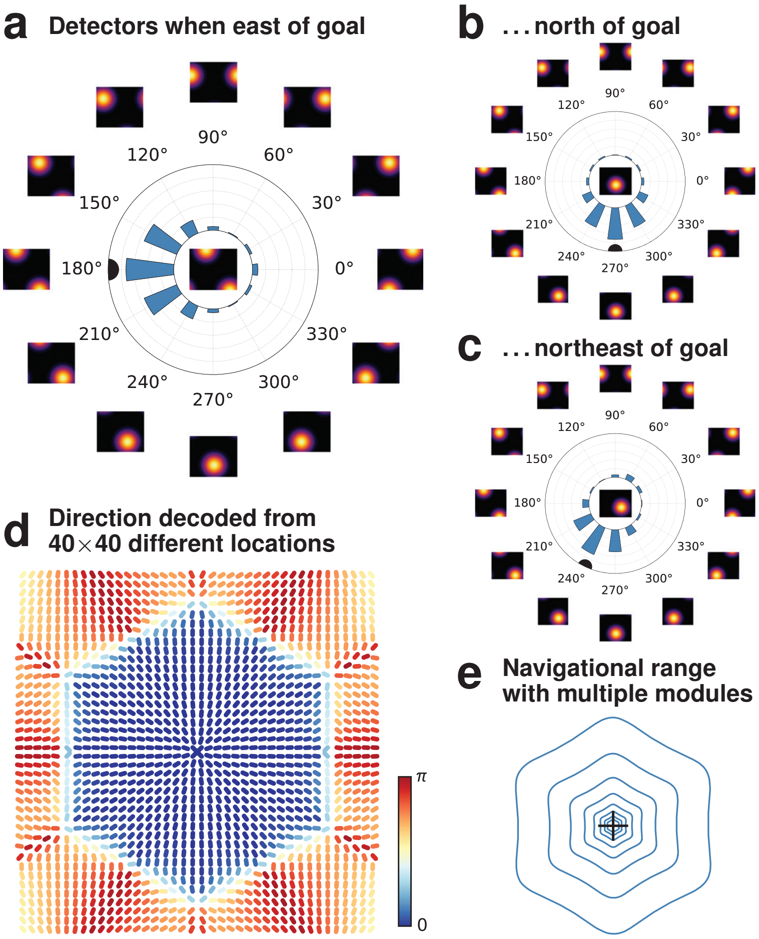


Figure 3.3: (A) Detectors in Fig. 3.2 (set to navigate toward goal state in Fig. 3.1a top center) tested against matrix in Fig. 3.1a bottom center. Western detectors are more strongly activated, so the correct direction was found. Black notch shows final result based on population vector average of all detectors. (B) Same detectors tested against matrix in Fig. 3.1b, 1st column, correctly showing goal is south. (C) Detectors tested against Fig. 3.1b, 2nd column. (D) Decoder tested at 40^2 locations, equally spaced from -0.4 m to $+0.4$ m on each axis. Each line points in decoded direction, colored by the deviation from true goal direction. Successful trials form a goal-centered hexagon; outside, vectors point to erroneous targets (cf. Fig. 1d in Paper E, p. 215). (E) Extra modules enable navigation beyond range of one module. Illustrated here is the range of eight modules if scale is assumed to grow by a factor of 1.5 at each step.

neurons within the sheet. Once a detector is appropriately configured to receive multiplicative synapses from the current and target state grid cell populations, it does not require any changes to its mechanism to change the goal location—this is simply a matter of updating the “goal grid cell” inputs.

Bounds for successful navigation with only one module

To assess how the outlined decoder works from many different starting locations, Fig. 3.3d plots the results from $40 \times 40 = 1600$ different trials arranged in a grid of locations spaced from -0.4 m to $+0.4$ m along each axis and centered on the goal location. Each trial is drawn as a short line segment pointing in the decoded goal direction, colored according to its deviation from the true goal vector (blue indicating no deviation and red indicating 180° deviation from the true goal vector). Among these trials, there is a clear domain within which the decoder is able to successfully calculate the correct goal vector—but, outside this region, the goal vectors lead the agent away from the correct target. This is because of the ambiguity of the repeating grid cell pattern: for excursions that exceed half of the grid scale, the activity packet may have moved sufficiently far in the neural sheet for the closest crosscorrelation peak to now indicate an erroneous goal direction.

At this point, when the grid pattern has started to repeat, navigation with only one grid module will no longer succeed. A plausible solution from the brain is then to integrate information from multiple grid modules of increasing scale. As discussed in Section 2.4.4, grid modules in the brain appear to form a geometric progression of grid scales. Such a progression of eight modules is illustrated in Fig. 3.3e. Even though the smallest-scaled grid module might have a short navigational range, the exponential growth in grid scale causes the valid navigational range to rise quickly in the successive grid modules.

The approach taken here is to assume that you have a sufficiently large number of modules, such that the largest module always covers the required navigational range of the agent. The largest-scaled module can then always provide a rough indication of the correct goal direction, while the smaller-scaled modules can gradually refine this signal as the goal gets closer. This is one of the possible interpretations of how the grid cell system might enable navigation over very long distances, which biologically remains an open question. In the next section we give a more detailed treatment to this topic of combining information from multiple modules, when we summarize how Paper C improves on the model from Papers A and B to function over long distances.

3.2 Research Question 2 (RQ2): Foundation for a navigational system

3.2.1 Navigating over arbitrarily long distances

With the positive finding under RQ1 that grid cell populations can be decoded for vector navigation, the next research question asks whether this principle can be used as a foundation for a larger navigational system. Can a more fully-featured navigational agent be built upon the concept of decoding grid cells? Although merely investigated in artificial agents simulated under highly simplified conditions, a positive outcome here could carry significance also for the biological system—through providing a proof-of-concept for what an overall role for grid cells might be in the real brain.

The investigation under RQ2 has mainly followed two tracks, corresponding respectively to Papers C and D. The first track concerns whether the grid cell-based navigation system can be applicable over long distances. For the coordinate system and the decoder to be the most useful, they should be able to work over distances greatly exceeding the range of the smaller-scaled modules. The second track concerns whether grid cell-based vector navigation can be integrated into a larger neural navigation system that can negotiate more complicated environments. A grid cell decoder will produce a goal vector pointing along the shortest, straight-line path to the goal, but that path might be blocked by obstacles. Can an agent overcome such obstacles, while still benefitting from using grid cells as part of its navigational algorithm?

Multiple approaches to grid cell decoding

When attempting to use grid cells for navigation over long distances, a single grid module is not enough. Due to the repeating nature of the grid pattern,

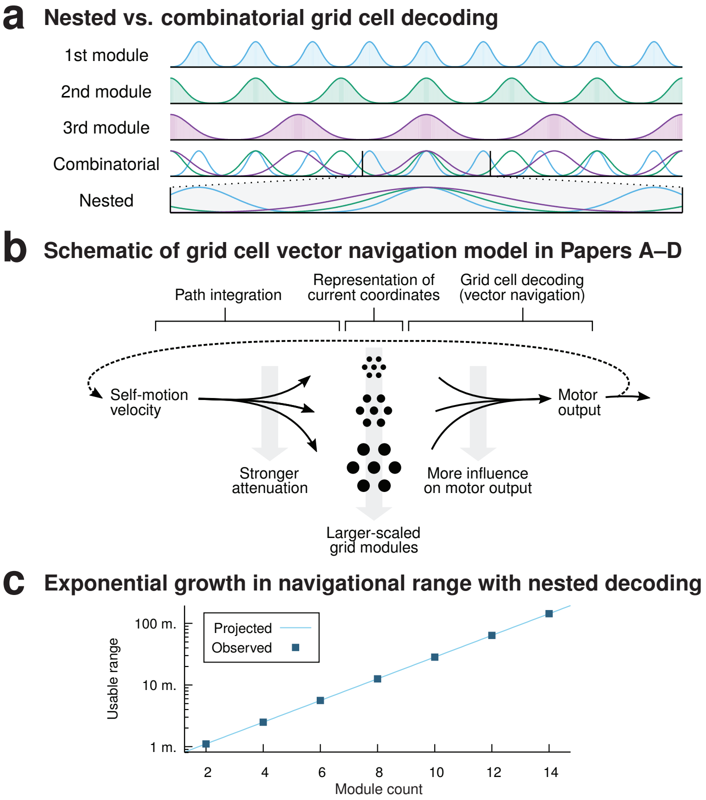

a
Nested vs. combinatorial grid cell decoding
1st module
2nd module
3rd module
Combinatorial
Nested
b
Schematic of grid cell vector navigation model in Papers A–D
Path integration
Representation of current coordinates
Grid cell decoding (vector navigation)
Self-motion velocity
Motor output
Stronger attenuation
More influence on motor output
Larger-scaled grid modules
c
Exponential growth in navigational range with nested decoding
Usable range
100 m.
10 m.
1 m.
Projected
Observed
2
4
6
8
10
12
14
Module count

Figure 3.4: (A) Adapted from Fig. 5 in Paper C, p. 140. Two contrasting views on how multiple grid modules together can support navigation. Three successive modules, scale ratio 1.5. Combinatorial: Exploits collective patterns across all modules, which remain unique far beyond range of largest module. Largest module has several peaks along spatial axis shown here, but all three modules are co-active only in the center. Nested: Assumes largest module is larger than the desired navigational range. Collective patterns across all modules not needed to navigate. Largest module offers rough indication of goal direction, while smaller modules gradually refine this estimate. (B) Adapted from Fig. 6 in Paper C, p. 142. Schematic of vector navigation model in Papers A–D. Agent's current coordinates maintained by a series of modules, which update their representation by path integration of self-motion inputs. Upon decoding modules for vector navigation, larger modules get more influence on the final direction (according to nested view). (C) Adapted from Fig. 13a in Paper C, p. 153. Final version of navigation model in Paper C tested for its usable navigational range with increasing numbers of grid modules. Thin line shows the usable range for two modules extrapolated to larger numbers of modules, assuming an exponential growth in range (note logarithmic y-axis). As expected, range increased exponentially.

the information from the single module is ambiguous once the agent ventures outside the unit tile corresponding to that module. This ambiguity can be resolved by incorporating information from multiple modules of different scales. However, how should this combination of information across modules specifically be done? Two main standpoints have emerged, the *combinatorial* view and the *nested* view (Fig. 3.4a), both briefly introduced next.

The combinatorial view emphasizes the fact that even a small number of grid modules will generate unique conjunctive activity patterns potentially far beyond the range of the largest-scaled module. The large theoretical capacity arises because periodic activity across modules does not repeat with the same frequency. When a given module has started repeating its activation patterns, this is not necessarily aligned with similar repetition of activity in other modules. Note for instance in Fig. 2.5g that all four modules overlap in only one location, even though the size of the enclosure exceeds the grid scale of the largest module. In the example in Fig. 3.4a (see also Fig. 5 in Paper B, p. 105), observe that the three grid modules are all co-active only in the center. If the animal travels to the next activity peak in the largest-scaled module, even though that module is now ambiguous, the different pattern in the smaller-scaled modules could still be used to discern that this is a different location.

Nested decoding of grid cells

While the combinatorial view offers long navigational ranges with only a few modules, it might be highly susceptible to noise and require complex decoding circuitry. The nested view, on the other hand, emphasizes the fact that grid modules are arranged in a geometric progression, such that there is an exponential growth in the range of successive modules. Given a sufficient number of modules, the largest-scaled grid module should therefore exceed the required navigational range of the agent. With that assumption, the grid cell decoding problem can be simplified greatly, as the agent no longer needs to consider conjunctions of activity across all grid modules in order to successfully navigate. According to the nested view, the largest-scaled grid module will always provide a rough indication of the goal direction. Because of the presumably large scale at which that grid module represents space, though, this estimate might be very noisy. However, as long as the agent gets guided into the valid domain of smaller-scaled modules, these modules can then take over control and gradually refine the goal vector.

In this project we thus take the nested view of the grid cell system. A schematic of the nested grid cell-based vector navigation model used in Papers A–D is shown in Fig. 3.4b. The current position of the agent is represented by a set of grid modules arranged in a geometric progression. This representation is updated by a path integration process in the continuous attractor networks used to model the grid cells. Separate continuous attractor networks are instantiated for each module, with the velocity input attenuated for the larger-scaled grid modules in order to achieve the larger grid scale in these modules (from otherwise identical grid cell networks). When the modules are decoded to produce the goal vector used to drive the agent, the larger-scaled modules are given more influence on the output vector—according to the nested view.

Solving path integration for large-scaled modules

The question addressed by Paper C is whether a nested grid cell-decoding agent can indeed navigate over long distances, which should be possible given a sufficient number of grid modules. Papers A and B, our initial investigation of grid cell decoding, ultimately only used four grid modules, with the smallest and largest grid scale separated by a factor of twelve. This setup was thus not suited for long-range navigation, so in Paper C we extended the model to investigate whether the same type of model can also navigate over distances on the order of hundreds of meters. To this end, the model was tested in a systematic way to determine the navigational range when using a given number of grid modules, and that number of modules was then varied to see whether the achieved navigational range indeed followed a geometric progression.

Initially, the navigational range turned out to cease increasing at around ten grid modules (see Fig. 8 in Paper C, p. 144). This shortcoming was, however, not due to any issues in the decoding network itself, but rather due to an inability of the continuous attractor networks to correctly perform path integration in the larger-scaled grid modules, as implemented. Because of the velocity-based attenuation used to produce larger-scaled grids (simply scaling down the strength of the velocity signal), the continuous attractor networks would after a certain point simply stop responding to the weakened input signals.

A number of possible fixes were proposed and evaluated (see Paper C for the full details). As the ultimate solution was chosen a method where grid attenuation is performed by using a probabilistic neural update rate rather than by attenuating the velocity input. Large-scaled (as well as small-scaled) grid modules were then able to correctly perform path integration, and the expected exponential growth in navigational range was accordingly restored (Fig. 3.4c). The valid navigational range with one module was determined to be around 1 meter with the setup in Paper C; this was successfully extended to 14 modules (with a scale ratio of 1.5 between successive modules), for a successful navigational range on the order of 100 meters. This could expectedly be extended to even longer distances by simply adding more modules.

Furthermore, the computational requirements of the model grew at most linearly in the number of modules (Fig. 13 in Paper C, p. 153). As the achieved navigational range is exponential in the number of modules, the computational demand is thus logarithmic in the desired range. This shows that even with the conceptually simpler approach of nested over combinatorial decoding, the system is computationally scalable to long-distance scenarios. Paper C thus demonstrates an important feature of the model were it to be applied within a larger navigational system (e.g. in a bio-inspired robotics setting for long-distance robot navigation using neural networks for path integration and vector navigation): a nested grid cell navigation model can indeed scale well to long distances in both the path integration and the grid cell decoding aspects of the model. The results also carry biological relevance, by demonstrating that long-distance vector navigation does not depend on a combinatorial view of the grid cell system. Nested grid cell decoding should also be considered among the viable hypotheses for vector navigation processes in the brain.

3.2.2 Integrating grid cells with border cells and place cells

A fully-fledged navigational system requires more than a coordinate system for long-range vector calculations as here provided by grid cells. Importantly, an agent will need to handle obstacles that might arise on its way to the goal. Will the goal vector information provided by grid cells then be of diminished value, or is there still a place for grid cell-based vector decoding in a navigational circuit built to handle obstacles? In Paper D we investigate this question, proposing a larger navigational model where grid cells successfully participate in a navigation process for obstacle-cluttered environments. The model includes other cell types known from the hippocampal formation, and provides a biologically relevant proposal for how grid cell-based navigation might interact with these other representations of space found in the hippocampal circuit (see Fig. 1 in Paper D, p. 160; cf. anatomical overview in Fig. 2.3).

Border cell-based deflection around slanted obstacles

Our contributions in Paper D separate into dealing with two distinct classes of obstacles, here termed *slanted* and *perpendicular* obstacles. Obstacles that form an angle of less than 90° with the goal vector—slanted obstacles—are easily negotiable by the agent, as the agent can continue its goal approach simply by deflecting its direction of motion away from the obstacle (Fig. 3.5a). As long as the obstacle keeps presenting an angle of less than 90° with the goal vector, the goal distance will keep decreasing as the agent follows the deflected trajectory. Thus, in some situations, this deflection mechanism can be sufficient to ultimately proceed all the way to the goal. In Paper D we describe a simple approach to implementing this border deflection strategy: Border cells, known to exist alongside grid cells in the medial entorhinal cortex, provide the information needed to calculate the deflected movement direction. By using information from borders cells to inhibit neural activity in a ring of “motor cells”, the agent’s final direction of motion will gently steer away from the obstacle as it is approached (see Fig. 2ab in Paper D, p. 164).

Note that this border deflection mechanism fundamentally depends on the ability of grid cells to constantly update the goal vector as the agent makes its way around obstacles. In the situation sketched in Fig. 3.5a, for instance, after the agent has cleared the first obstacle, the goal is now found in a different direction. Grid cells can provide this updated goal direction, so border deflection goes hand in hand with grid cell-based vector navigation.

Diverting to new subgoals to avoid perpendicular obstacles

Many obstacles are not this easily traversable. An obstacle might present a perpendicular boundary to the goal vector (Fig. 3.5b), or initially be slanted

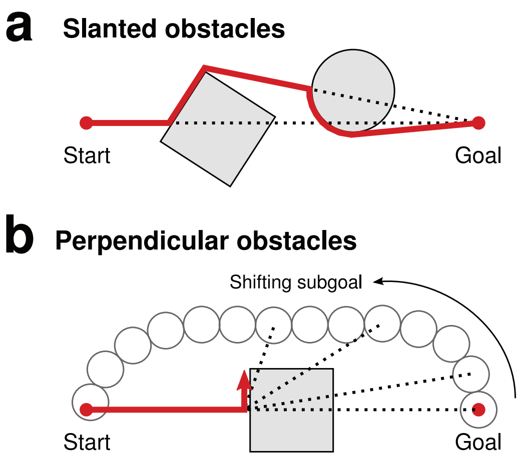


Figure 3.5: (A) In Paper D the agent architecture was extended to handling obstacles in the environment. Slanted obstacles—presenting an angle of less than 90° with the goal vector—can be traversed by deflecting the direction of motion away from the obstacle, e.g. as sketched in this illustration. At any point along the path from start to goal the goal distance decreases, even as the agent had to deflect its trajectory around an obstacle. Details in Paper D. (B) Obstacles perpendicular to the goal vector will block the agent from making further progress using only the deflection mechanism sketched out in A. The solution proposed in Paper D is to select a different subgoal for vector navigation. We propose that place cells (grey circles) can provide candidates for subgoals, and that a process similar to the biological phenomenon of hippocampal replay events, where activity in remote place cells spontaneously plays back along paths previously traveled, could be used to select the next subgoal (see Paper D).

but then gradually lead the agent into a basin of attraction where the deflected trajectory terminates at a perpendicular point. When boundary deflection is no longer sufficient to find a course for the agent, one possibility enabled by grid cell decoding is to change the goal for vector navigation: the agent can temporarily attempt to vector-navigate to a different destination as its subgoal. Different subgoals produce different goal vectors, such that perpendicular obstacles might become slanted and vice versa. If the agent can find a more suitable subgoal, it could thus become unstuck from the current obstacle.

In Paper D we propose that place cells might serve the role of representing different subgoals. We further suggest that the subgoal reselection process could be related to the phenomenon of *replay* known from the hippocampus, where hippocampal activity during rest or sleep has been observed to spontaneously play back neural trajectories along paths previously traveled. We model the place cell network as a topological graph of locations, and let the subgoal shift through this network of locations along the shortest path between the agent's current location and the goal location. Whenever the agent gets stuck, the subgoal moves gradually closer to the agent until a more suitable destination has been found, enabling vector navigation to resume—or, if none can be found, eventually guiding the agent purely based on the topological information from place cells. Our model in Paper D thus not only suggests a potential role for replay in navigation, but also shows how grid cell-based vector navigation can interact with other navigational strategies thought to be supported by the hippocampal formation, such as place cell-based topological navigation.

Together, the two Papers C and D show how a grid cell-based vector navigation model can serve as the foundation for a larger navigational system: Paper C by demonstrating that the coordinate system can be used over long distances with modest resource requirements, and Paper D by showing that external sensory inputs such as border cells can be used to augment vector navigation in more complicated environments. In Fig. 1 in Paper B, p. 99, we sketched out a proposal for how grid cells might fit into a larger navigational architecture. The model developed in Paper D largely adheres to this overall framework, while also introducing border cells for local obstacle avoidance. The model does not, however, use sensory inputs as suggested in Paper B to generate place cells or to correct errors in the path integration process—though it could later be extended also in that direction.

3.3 Research Question 3 (RQ3): Plausibility of the decoding approach

3.3.1 Nested decoding of distorted grid modules

The final research question asks about the resilience of the proposed decoding mechanism to various biological phenomena known from real grid cells. Given our current knowledge of grid cells, and particularly their increasingly evident variability, does it remain plausible that a nested grid cell decoder, as

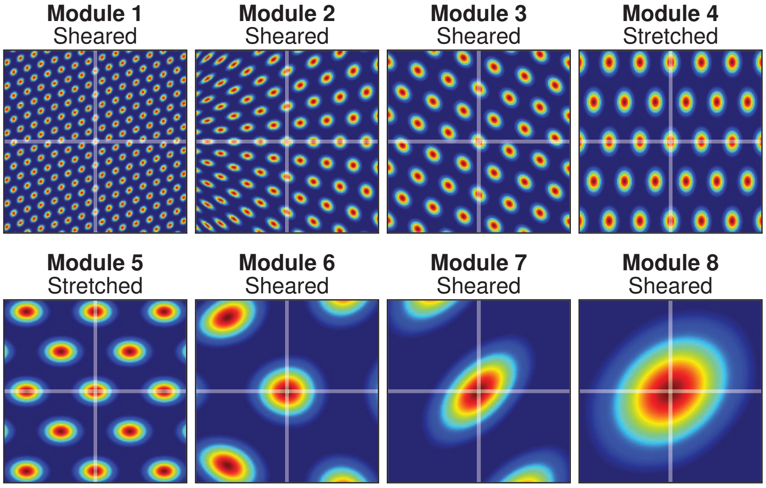


Figure 3.6: Adapted from Fig. 4a in Paper E, p. 224, where a nested grid cell decoder is tested with populations of distorted grid cells. Example grid cell shown for each of the eight modules part of an experiment in Paper E. Each module experiences a given distortion to its grid patterns, all cells in a given module experiencing the same distortion but with different distortions in different modules. Modules 4 and 5 experience stretch distortions along either the east–west or north–south axis, while the other modules experience various forms of shearing distortions. The nested decoding model was shown able to navigate despite these distortions (see Paper E for details).

investigated in the previous research questions, could work biologically? Real grid cells have been found not to strictly adhere to the rigid hexagonal lattice evoked by the three parameters of scale, orientation, and offset (Fig. 2.5a), but to deviate from this idealized pattern in various ways. For example, grid patterns might *shear* away from an axis of the environment with increased exposure to that environment (e.g. Fig. 3.6, module 3), or they might stretch or compress along an axis (e.g. Fig. 3.6, module 4). These distorted grids challenge assumptions underlying grid cell decoding models, and must be adequately addressed if these models are to be considered viable biologically.

Paper E addresses this challenge by testing the nested grid cell decoding approach with such distorted grid cells. As in the previous papers, the model uses multiple grid modules, of increasing scale, to navigate longer distances by decoding them in a nested fashion as described above. Paper E shows that the distortions considered here—stretching and shearing—do not constitute a problem when decoding individual grid modules. Grid cells from the same module are assumed to distort coherently, as this appears to be the case biologically. Our proposed decoding mechanism, based on finding the offset of activity packets in the neural sheet, will then still perform adequately when a given grid module is affected by distortions. Furthermore, because the nested grid cell decoder does not rely on precisely interlocked conjunctions of activity across all modules in order to navigate, it essentially only requires a good goal direction signal from a single grid module at a time. Nested decoding therefore remains functioning properly even when multiple grid modules are distorted. This holds true also when different grid modules are distorted in a non-coherent fashion (Fig. 3.6), which has been observed biologically.

The results in Paper E therefore defend the concept of nested grid cell decoding against the challenge of distorted grids, by demonstrating that a nested decoding strategy is fundamentally able to cope with these situations. This is further underscored through separate simulations showing that nested grid cell decoding works even when large amounts of noise are added to each grid module. The smaller-scaled grid modules are “perturbed” to such a degree that they no longer resemble grid cells when the agent is far away from the goal location, yet the nested principle of grid cell decoding ensures that the agent can still find its way back to the correct location. Paper E thus shows that the nested decoding mechanism is highly resilient to imperfect, noisy grids, even with no particular changes made to the decoder itself. This resilience and simplicity of the nested decoding approach should encourage further investigation into whether evidence of similar processes can be found in the brain.

Chapter 4

Discussion

4.1 Summary of research contributions

This section summarizes the main research contributions of this thesis, as they pertain to the three Research Questions set out in the introduction (Section 1.2) and as discussed in detail in the previous chapter. The summary below makes brief references to several key related publications, but see also the next section for a more detailed treatment of the related work.

Research Question 1 (Decodability of grid cells):

> Can grid cells be decoded by a downstream network, in order to make full use of the embodied coordinate system?

- In Paper A we demonstrated that goal vectors can be read out directly, i.e. decoded, from grid modules by tracking the displacement of the activity packet in the grid module's neural sheet (Section 3.1.1).
- Distances beyond the range of one grid module can be navigated by adding more grid modules to the system; biological grid modules are indeed known to be organized with a geometric progression of grid scales.
- Ours is the first demonstration of nested decoding by neural networks, although two other papers from the same year as Paper A support the notion that grid cells can be decoded for vector navigation: Bush et al. [79] demonstrated neural decoding of grid cells in a combinatorial fashion (as opposed to the nested decoding considered here), while Stemmler et al. [69] analyzed the theoretical principles behind nested decoding.

Research Question 2 (Foundation for a navigational system):

> Can a grid cell decoder be the foundation for a
> larger agent architecture for navigational tasks?

- For the grid cell decoder to properly fulfill the role of a coordinate system in a larger navigational architecture, it is important that arbitrarily long distances can be represented and computed with. We demonstrated that an agent controller integrating grid cell-based path integration with vector navigation, can indeed be made able to function over arbitrarily long distances (Section 3.2.1)—taking the nested view of the grid cell system and ensuring that there is a sufficiently large number of grid modules available (logarithmic in the desired navigational range).
- For grid cell-based vector navigation to be useful in realistic environments, obstacles in the environment must be dealt with. We showed that grid cell-based vector navigation can usefully be integrated into a larger architecture containing both border cells and place cells, that respectively augment the agent's vector navigation capabilities by deflecting away from nearby obstacles, and by diverting vector navigation to different subgoals when the former is not possible (Section 3.2.2).
- This extended vector navigation architecture is based on spatial neurons known from the hippocampal formation, and could therefore be viewed as a hypothesis for how these spatial representations work together to support navigation in complex environments. The model could also be the starting point for a neuromorphic, bio-inspired robot controller.
- Erdem and Hasselmo [78] used grid cells to "search" for a goal direction in complex environments, but our work shows that grid cells instead can support such tasks through decoding (possibly a more efficient process).

Research Question 3 (Plausibility of the decoding approach):

> Does grid cell decoding seem biologically plausible,
> given our current understanding of real grid cells?

- One challenge raised against the plausibility of grid cell-based vector navigation is that grid cells do not strictly adhere to perfectly hexagonal grids, but distort in various ways. We showed that the nested decoding approach is highly resilient to distortions such as shearing/stretching, even when different grid modules are distorted differently (Section 3.3.1).

- Nested grid cell decoding can even remain functioning despite severely perturbed grid cell activity outside the goal vicinity. These findings all stem from fundamental properties of the nested decoding approach.
- It remains unclear whether combinatorial decoding could similarly be made able to handle distorted and perturbed grids. Further theoretical and neurophysiological investigation might help distinguish between these different proposals for grid cell-based vector navigation.

The overarching Research Goal for this thesis was to understand the potential role of grid cells: *advancing our understanding of what grid cells might be useful for, and under which circumstances*. Through our findings in this thesis, as summarized above, we hope to have advanced the notion that grid cells could be used for vector navigation; that this can work over arbitrarily long distances, given a sufficient number of grid modules; that grid cell-based vector navigation can be integrated with other navigational strategies known from the hippocampal formation; that the proposed mechanism of nested grid cell decoding is fundamentally resilient to challenges posed by biological grid patterns; and finally, that the mechanism nevertheless remains resource efficient, with only logarithmic requirements in the desired navigational range.

Our investigation of these questions has used highly simplified models of biological neurons, in the form of artificial neural networks. Though far from a detailed biophysical account for how such processes might occur in the brain, the models presented in this thesis nevertheless provide proofs of concept that the *neural representation* of grid cells might hold the properties laid out above. We thus hope that, beyond potentially being useful in bio-inspired robotic scenarios, our results might inspire further neuroscientific investigation into the potential role of grid cells in vector navigation in the real brain.

4.2 Related work on the possible role of grid cells

As this work forms part of an active area of research, several other viewpoints and related models concerning the potential role of grid cells bear mentioning. This section expands on the discussion of related work above (pertaining to the specific Research Questions of the thesis), but also includes other perspectives on the broader topic of the role of grid cells in the brain.

Grid cells for vector navigation: On the possible role of grid cells within vector navigation, multiple models have been put forth in recent years. Erdem and Hasselmo [78] proposed driving the grid cell system in an active way along possible future trajectories, to test whether any of these virtual paths might trigger the goal place cell and thus indicate the correct direction forward. This is thus an approach where grid cells are used for vector navigation not through any decoding process (which is the approach considered in this thesis), but by using the grid cell system to actively *search* for the goal direction by virtual travel through space.

Bush et al. [79] proposed, around the time of Paper A's publication, several interrelated models for performing *combinatorial* decoding of grid cells, based on instantiating arrays of detectors that look for each possible unique conjunction of grid cell activity across all grid modules present in the system. Closer to the approach taken in this thesis, Stemmler et al. [69] proposed a mechanism for *nested* grid cell decoding based on recursive population vector readout, where each grid module is iteratively decoded to successively adjust the final readout vector. This ultimate, refined goal vector will then appropriately reflect the coordinate information provided from all grid modules. Our neural network-based model, however, suggests that you only need to read out a single module at a time: far away from the goal it is only necessary to decode the larger-scaled modules, and only upon approaching the goal do you need to consider the information from the smaller-scaled modules.

Banino et al. [80] recently described a system where a neural navigation model is trained using reinforcement learning based on self-motion and other sensory inputs. A recurrent neural network was first successfully trained to generate place cell and head-direction cell outputs based on path integration inputs, and units resembling grid cells were shown to emerge in late layers of the architecture. These units were then further used as inputs to a second recurrent network that eventually learned to take shortcuts in the environment, putatively by decoding the grid cell activity. It is encouraging to see such machine learning methods arrive at solutions similar to the brain and those proposed here, and it would be interesting to see whether this work in the next step could help address, e.g., the question of nested versus combinatorial decoding.

Grid cells as inputs to downstream areas: Beyond the potential role of grid cells in vector navigation, other possible roles for these neurons are also actively being investigated. Grid cells were early on suggested to potentially provide inputs to hippocampus sufficient for generating place cells [21, 76]. This view is complemented by recent findings that grid-like units might emerge from learning processes related to Principal Component Analysis (PCA), that seek to find an efficient encoding for representing place cell-activity [75]. Further work has suggested that certain grid distortions might be explained by grid cells providing a PCA representation of reinforcement learning-related signals in the hippocampus [81].

Both of these views, while providing a possible account for the *formation* of grid cells (Fig. 1.1), also implicitly suggest that the *role* of grid cells might primarily be to provide an efficient representation for supporting a desired neural activity in the downstream hippocampus. However, while this might be one possible account for the role of grid cells, that does not preclude the same neural signal also supporting vector navigation processes as investigated in this thesis. A neural representation evolved for one particular role might later on have been adopted for other purposes.

Grid cells for non-navigational information: The entorhinal cortex, where grid cells are primarily found, is a gateway between the hippocampus and the neocortex. As such, it is expectedly also involved in other information processing—beyond navigation—related to the hippocampus. Recent work suggests that grid cells might encode several kinds of non-navigational information, such as gaze position [82], task-related sound frequencies [83] and continuous variables in conceptual space [84]; this raises the exciting notion that grid cells might relate to more general principles for cognitive processing in the brain. Modeling work on neocortical information processing has even led to the suggestion that grid cell representations might be useful as a general computational principle throughout the neocortex [85]. Further modeling could help elucidate these possible general computational properties of grid cells, in biological or artificial neural networks.

4.3 Future directions and conclusion

A continued investigation into the potential role of grid cells in vector navigation could proceed in many directions. As an interdisciplinary project, the research questions opened by this work point into several fields. There are thus both biological, theoretical and engineering tracks of future work.

Biological evidence of grid cell-based vector navigation: It is not yet known whether biological grid cells in fact do participate in vector navigation. If they do, then what—and where—is the biological mechanism? Cells in the CA1 area of the bat hippocampus have been reported to activate as a function of goal direction [86]. Are these neurons part of a vector navigation mechanism? If so, how do they work, and can they be found in other species as well?

Nested versus combinatorial decoding: Does biological vector navigation follow the principles of combinatorial decoding, nested decoding, or some other approach [87]? Both neurophysiological and theoretical work could advance our understanding of this question. Importantly, the nested approach assumes a larger number of concurrent grid modules than have currently been reported. Are the currently reported numbers the actual biological limit, or have the remaining modules simply not yet been found? If so, the indications that larger-scaled modules consist of fewer neurons could help explain why they have not yet been reported [88]. Biological recordings of grid cells in much larger spaces than current laboratory environments could help illuminate this issue [89], e.g. by recording bats in their natural environment [90].

Utility of grid cells for non-spatial information: The possibility that grid cells also encode non-spatial information [82–84] suggests that there might be circuitry in the brain for vector calculations involving these non-spatial variables. As with navigation, this is a question that could be pursued both theoretically and physiologically. The intriguing results of grid-like neurons emerging in models optimized for navigational tasks [75, 80, 81], also hint at a more general role for grid cells in information processing. These interactions between grid cells and machine learning methods represent an exciting direction for further research, both biologically and for artificial intelligence.

Robotic and neuromorphic hardware implementations: Bio-inspired navigation methods have been successfully developed before, e.g. in “RatSLAM”: a hippocampus-inspired architecture that uses continuous attractor networks for path integration and place recognition, and which also in other ways is inspired by rat navigation [91]. This architecture has been successfully used in several applications, such as mapping out the street layout of a suburb based on camera inputs alone [92]. The navigation models developed in this thesis could potentially also have robotic applications, and might be developed further to run on emerging neuromorphic hardware [42].

—

In conclusion, in this work we started out with a biological phenomenon and ventured to build an artificial system using the same principles. As an interdisciplinary project, the models developed in this project operate on a high level of abstraction above the biophysical realities. However, when trying to understand such a complicated system as the brain, it is essential to be able to work across multiple levels of abstraction. Our successful results, implementing vector navigation using simulated grid cells, provide a proof of concept that suggests that the biological system might harbor some of the same features. The results also provide a platform for further development of artificial, bio-inspired agents. This demonstrates how a close interaction between neuroscience and artificial intelligence can be fruitful, both in trying to understand natural intelligence, and in endeavoring to develop machine intelligence.

Bibliography

[1] V. Edvardsen, “A passive mechanism for goal-directed navigation using grid cells,” in *Proceedings of the 2015 European Conference on Artificial Life (ECAL)*, 2015, pp. 191–198.

[2] ——, “Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network,” *Natural Computing*, 2016.

[3] ——, “Long-range navigation by path integration and decoding of grid cells in a neural network,” in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.

[4] V. Edvardsen, A. Bicanski, and N. Burgess, “Navigating with grid and place cells in cluttered environments,” *Hippocampus* (in press), 2019.

[5] V. Edvardsen, “Navigating with distorted grid cells,” in *Proceedings of the 2018 International Conference on Artificial Life (ALIFE)*, 2018.

[6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.

[7] P. S. Churchland and T. J. Sejnowski, *The Computational Brain.* MIT Press, 1992.

[8] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, “Neuroscience-inspired artificial intelligence,” *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[9] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[10] J. Hawkins and S. Blakeslee, *On Intelligence*. Times Books, 2004.

[11] E. I. Moser, M.-B. Moser, and B. L. McNaughton, “Spatial representation in the hippocampal formation: a history,” *Nature Neuroscience*, vol. 20, no. 11, p. 1448, 2017.

[12] W. B. Scoville and B. Milner, “Loss of recent memory after bilateral hippocampal lesions,” *Journal of Neurology, Neurosurgery, and Psychiatry*, vol. 20, no. 1, p. 11, 1957.

[13] G. Buzsáki and E. I. Moser, "Memory, navigation and theta rhythm in the hippocampal-entorhinal system," *Nature Neuroscience*, vol. 16, no. 2, p. 130, 2013.

[14] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat," *Brain Research*, vol. 34, no. 1, pp. 171–175, 1971.

[15] T. Solstad, C. N. Boccara, E. Kropff, M.-B. Moser, and E. I. Moser, "Representation of geometric borders in the entorhinal cortex," *Science*, vol. 322, no. 5909, pp. 1865–1868, 2008.

[16] J. S. Taube, R. U. Muller, and J. B. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis," *Journal of Neuroscience*, vol. 10, no. 2, pp. 420–435, 1990.

[17] E. Kropff, J. E. Carmichael, M.-B. Moser, and E. I. Moser, "Speed cells in the medial entorhinal cortex," *Nature*, vol. 523, no. 7561, p. 419, 2015.

[18] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–806, 2005.

[19] E. I. Moser, Y. Roudi, M. P. Witter, C. Kentros, T. Bonhoeffer, and M.-B. Moser, "Grid cells and cortical representation," *Nature Reviews Neuroscience*, vol. 15, no. 7, p. 466, 2014.

[20] H. Stensola, T. Stensola, T. Solstad, K. Frøland, M.-B. Moser, and E. I. Moser, "The entorhinal grid map is discretized," *Nature*, vol. 492, no. 7427, pp. 72–78, 2012.

[21] T. Solstad, E. I. Moser, and G. T. Einevoll, "From grid cells to place cells: A mathematical model," *Hippocampus*, vol. 16, no. 12, pp. 1026–1031, 2006.

[22] I. R. Fiete, Y. Burak, and T. Brookings, "What grid cells convey about rat location," *Journal of Neuroscience*, vol. 28, no. 27, pp. 6858–6871, 2008.

[23] M. Carandini, "From circuits to behavior: a bridge too far?" *Nature Neuroscience*, vol. 15, no. 4, p. 507, 2012.

[24] F. Carpenter and C. Barry, "Distorted grids as a spatial label and metric," *Trends in Cognitive Sciences*, vol. 20, no. 3, pp. 164–167, 2016.

[25] D. Purves, G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. LaMantia, and L. E. White, Eds., *Neuroscience*, 5th ed. Sinauer Associates, 2012.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[27] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[28] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[29] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[30] D. Floreano and C. Mattiussi, *Bio-inspired artificial intelligence: Theories, methods, and technologies*. MIT Press, 2008.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. Addison Wesley, 2009.

[35] A. S. Tanenbaum, *Structured Computer Organization*, 5th ed. Prentice Hall, 2005.

[36] G. M. Shepherd, *Neurobiology*, 2nd ed. Oxford University Press, 1988.

[37] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8619–8624, 2014.

[38] D. L. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature Neuroscience*, vol. 19, no. 3, pp. 356–365, 2016.

[39] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[40] E. Chung, J. Fowers, K. Ovtcharov, M. Papamichael, A. Caulfield, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman *et al.*, "Serving DNNs in real time at datacenter scale with Project Brainwave," *IEEE Micro*, vol. 38, no. 2, pp. 8–20, 2018.

[41] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2017, pp. 1–12.

[42] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[43] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, p. 61, 2015.

[44] C. N. Boccara, F. Sargolini, V. H. Thoresen, T. Solstad, M. P. Witter, E. I. Moser, and M.-B. Moser, "Grid cells in pre-and parasubiculum," *Nature Neuroscience*, vol. 13, no. 8, p. 987, 2010.

[45] M. P. Brandon, J. Koenig, and S. Leutgeb, "Parallel and convergent processing in grid cell, head-direction cell, boundary cell, and place cell networks," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 5, no. 2, pp. 207–219, 2014.

[46] N. M. Van Strien, N. Cappaert, and M. P. Witter, "The anatomy of memory: an interactive overview of the parahippocampal–hippocampal network," *Nature Reviews Neuroscience*, vol. 10, no. 4, p. 272, 2009.

[47] V. B. Mountcastle, "An organizing principle for cerebral function: The unit module and the distributed system," in *The Mindful Brain*, G. M. Edelman and V. B. Mountcastle, Eds. MIT press, 1978.

[48] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory." *Psychological Review*, vol. 102, no. 3, p. 419, 1995.

[49] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? Complementary learning systems theory updated," *Trends in Cognitive Sciences*, vol. 20, no. 7, pp. 512–534, 2016.

[50] M. A. Tosches, T. M. Yamawaki, R. K. Naumann, A. A. Jacobi, G. Tushev, and G. Laurent, “Evolution of pallium, hippocampus, and cortical cell types revealed by single-cell transcriptomics in reptiles,” *Science*, vol. 360, no. 6391, pp. 881–888, 2018.

[51] S. Reiter, H.-P. Liaw, T. M. Yamawaki, R. K. Naumann, and G. Laurent, “On the value of reptilian brains to map the evolution of the hippocampal formation,” *Brain, Behavior and Evolution*, vol. 90, no. 1, pp. 41–52, 2017.

[52] L. Pini, M. Pievani, M. Bocchetta, D. Altomare, P. Bosco, E. Cavedo, S. Galluzzi, M. Marizzoni, and G. B. Frisoni, “Brain atrophy in Alzheimer’s disease and aging,” *Ageing Research Reviews*, vol. 30, pp. 25–48, 2016.

[53] S. Leutgeb, J. K. Leutgeb, C. A. Barnes, E. I. Moser, B. L. McNaughton, and M.-B. Moser, “Independent codes for spatial and episodic memory in hippocampal neuronal ensembles,” *Science*, vol. 309, no. 5734, pp. 619–623, 2005.

[54] S. S. Deshmukh and J. J. Knierim, “Representation of non-spatial and spatial information in the lateral entorhinal cortex,” *Frontiers in Behavioral Neuroscience*, vol. 5, p. 69, 2011.

[55] G. J. Quirk, R. U. Muller, and J. L. Kubie, “The firing of hippocampal place cells in the dark depends on the rat’s recent experience,” *Journal of Neuroscience*, vol. 10, no. 6, pp. 2008–2017, 1990.

[56] M. Fyhn, T. Hafting, A. Treves, M.-B. Moser, and E. I. Moser, “Hippocampal remapping and grid realignment in entorhinal cortex,” *Nature*, vol. 446, no. 7132, p. 190, 2007.

[57] J. O’Keefe and M. L. Recce, “Phase relationship between hippocampal place units and the EEG theta rhythm,” *Hippocampus*, vol. 3, no. 3, pp. 317–330, 1993.

[58] T. Hafting, M. Fyhn, T. Bonnevie, M.-B. Moser, and E. I. Moser, “Hippocampus-independent phase precession in entorhinal grid cells,” *Nature*, vol. 453, no. 7199, p. 1248, 2008.

[59] D. J. Foster and M. A. Wilson, “Reverse replay of behavioural sequences in hippocampal place cells during the awake state,” *Nature*, vol. 440, no. 7084, p. 680, 2006.

[60] K. Diba and G. Buzsáki, “Forward and reverse hippocampal place-cell sequences during ripples,” *Nature Neuroscience*, vol. 10, no. 10, p. 1241, 2007.

[61] A. M. Wikenheiser and A. D. Redish, "Decoding the cognitive map: ensemble hippocampal sequences and decision making," *Current Opinion in Neurobiology*, vol. 32, pp. 8–15, 2015.

[62] B. E. Pfeiffer and D. J. Foster, "Hippocampal place-cell sequences depict future paths to remembered goals," *Nature*, vol. 497, no. 7447, p. 74, 2013.

[63] E. I. Moser and M.-B. Moser, "A metric for space," *Hippocampus*, vol. 18, no. 12, pp. 1142–1156, 2008.

[64] J. J. Knierim, "From the GPS to HM: Place cells, grid cells, and memory," *Hippocampus*, vol. 25, no. 6, pp. 719–725, 2015.

[65] Y. Burak and I. R. Fiete, "Accurate path integration in continuous attractor network models of grid cells," *PLoS Computational Biology*, vol. 5, no. 2, 2009.

[66] A. Guanella, D. Kiper, and P. Verschure, "A model of grid cells based on a twisted torus topology," *International Journal of Neural Systems*, vol. 17, no. 04, pp. 231–240, 2007.

[67] C. Barry, R. Hayman, N. Burgess, and K. J. Jeffery, "Experience-dependent rescaling of entorhinal grids," *Nature Neuroscience*, vol. 10, no. 6, p. 682, 2007.

[68] X.-X. Wei, J. Prentice, and V. Balasubramanian, "A principle of economy predicts the functional architecture of grid cells," *eLife*, vol. 4, p. e08362, 2015.

[69] M. Stemmler, A. Mathis, and A. V. Herz, "Connecting multiple spatial scales to decode the population activity of grid cells," *Science Advances*, vol. 1, no. 11, p. e1500816, 2015.

[70] L. M. Giocomo, M.-B. Moser, and E. I. Moser, "Computational models of grid cells," *Neuron*, vol. 71, no. 4, pp. 589–603, 2011.

[71] N. Burgess, C. Barry, and J. O'Keefe, "An oscillatory interference model of grid cell firing," *Hippocampus*, vol. 17, no. 9, pp. 801–812, 2007.

[72] M. C. Fuhs and D. S. Touretzky, "A spin glass model of path integration in rat medial entorhinal cortex," *Journal of Neuroscience*, vol. 26, no. 16, pp. 4266–4276, 2006.

[73] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, "Path integration and the neural basis of the 'cognitive map'," *Nature Reviews Neuroscience*, vol. 7, no. 8, p. 663, 2006.

[74] E. Kropff and A. Treves, "The emergence of grid cells: Intelligent design or just adaptation?" *Hippocampus*, vol. 18, no. 12, pp. 1256–1269, 2008.

[75] Y. Dordek, D. Soudry, R. Meir, and D. Derdikman, “Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis,” *eLife*, vol. 5, p. e10094, 2016.

[76] E. T. Rolls, S. M. Stringer, and T. Elliot, “Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning,” *Network: Computation in Neural Systems*, vol. 17, no. 4, pp. 447–465, 2006.

[77] C. Barry and D. Bush, “From A to Z: a potential role for grid cells in spatial navigation,” *Neural Systems & Circuits*, vol. 2, no. 1, p. 6, 2012.

[78] U. M. Erdem and M. Hasselmo, “A goal-directed spatial navigation model using forward trajectory planning based on grid cells,” *European Journal of Neuroscience*, vol. 35, no. 6, pp. 916–931, 2012.

[79] D. Bush, C. Barry, D. Manson, and N. Burgess, “Using grid cells for navigation,” *Neuron*, vol. 87, no. 3, pp. 507–520, 2015.

[80] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran, “Vector-based navigation using grid-like representations in artificial agents,” *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[81] K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman, “The hippocampus as a predictive map,” *Nature Neuroscience*, vol. 20, no. 11, p. 1643, 2017.

[82] N. J. Killian, M. J. Jutras, and E. A. Buffalo, “A map of visual space in the primate entorhinal cortex,” *Nature*, vol. 491, no. 7426, p. 761, 2012.

[83] D. Aronov, R. Nevers, and D. W. Tank, “Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit,” *Nature*, vol. 543, no. 7647, pp. 719–722, 2017.

[84] A. O. Constantinescu, J. X. O’Reilly, and T. E. Behrens, “Organizing conceptual knowledge in humans with a gridlike code,” *Science*, vol. 352, no. 6292, pp. 1464–1468, 2016.

[85] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, “A framework for intelligence and cortical function based on grid cells in the neocortex,” *Frontiers in Neural Circuits*, vol. 12, 2019.

[86] A. Sarel, A. Finkelstein, L. Las, and N. Ulanovsky, “Vectorial representation of spatial goals in the hippocampus of bats,” *Science*, vol. 355, no. 6321, pp. 176–180, 2017.

[87] L. Vágó and B. B. Ujfalussy, "Robust and efficient coding with grid cells," *PLoS Computational Biology*, vol. 14, no. 1, p. e1005922, 2018.

[88] N. Mosheiff, H. Agmon, A. Moriel, and Y. Burak, "An efficient coding theory for a dynamic trajectory predicts non-uniform allocation of entorhinal grid cells to modules," *PLoS Computational Biology*, vol. 13, no. 6, p. e1005597, 2017.

[89] K. B. Kjelstrup, T. Solstad, V. H. Brun, T. Hafting, S. Leutgeb, M. P. Witter, E. I. Moser, and M.-B. Moser, "Finite scale of spatial representation in the hippocampus," *Science*, vol. 321, no. 5885, pp. 140–143, 2008.

[90] M. Geva-Sagiv, L. Las, Y. Yovel, and N. Ulanovsky, "Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation," *Nature Reviews Neuroscience*, vol. 16, no. 2, pp. 94–108, 2015.

[91] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–1153, 2010.

[92] M. J. Milford and G. F. Wyeth, "Mapping a suburb with a single camera using a biologically inspired SLAM system," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, 2008.

Part II

Publications

Paper A

A passive mechanism for goal-directed navigation using grid cells (Edvardsen, 2015)

Author(s):
Vegard Edvardsen

Published at conference:
2015 European Conference on Artificial Life (ECAL)

Note:

This conference paper was later invited for submission in extended version to the journal *Natural Computing*, and that extended paper is included as Paper B in this thesis. As the extended paper contains most of the material presented in this version, the reader may choose to skip directly to Paper B for expediency.

Paper A

Edvardsen (2015)

A Passive Mechanism for Goal-Directed Navigation using Grid Cells

Vegard Edvardsen

Department of Computer and Information Science,
Norwegian University of Science and Technology, Trondheim, Norway

Abstract

As more is becoming understood about how the brain represents and computes with high-level spatial information, the prospect of constructing biologically-inspired robot controllers using these spatial representations has become apparent. Grid cells are particularly interesting in this regard, as they provide a general coordinate system of space. Artificial neural network models of grid cells show the ability to perform path integration, but important for a robot is also the ability to calculate the direction from the current location, as indicated by the path integrator, to a remembered goal. Present models for goal-directed navigation using grid cells have used a *simulating* approach, where the networks are required to actively test successive locations along linear trajectories emanating from the current location. This paper presents a *passive* model, where differences between multi-scale grid cell representations of the present location and the goal are used to calculate a goal-direction signal directly. The model successfully guides a simulated agent to its goal, showing promise for implementing the system on a real robot in the future. Some possible implications for neuroscientific studies on the goal-direction signal in the entorhinal/subicular region are briefly discussed.

1 Introduction

Results from neuroscience are gradually uncovering the neural basis for navigation, as cell types such as place cells and grid cells, first discovered in the hippocampal region of rats, have been shown to represent high-level features of the animal's spatial environment. These findings offer the prospect of beginning to understand how the brain computes and represents abstract cognitive features. Inspired by these advances, the basis for this project has been to devise and implement a neural model to enable a robot to find its way to a previously visited goal location using these neural representations of space known from the brain. Through crafting these models, we hope to gain insights into how these spatial representations might be utilized for navigational purposes by neural systems, artificial and real alike.

The first evidence of spatially responsive cells in the rat hippocampus came with the discovery of place cells, which were seen to respond at distinct locations in the environment (O'Keefe and Dostrovsky, 1971). However, place cells do not appear to encode any metric relations, such as distances and angles (Spiers and Barry, 2015). The place cell representation by itself is thus not sufficient to be able to navigate between arbitrary locations, because it does not offer any means to calculate the direction of travel from one place cell's firing location to that of another. Grid cells, discovered later in the neighboring *entorhinal cortex* (Hafting et al., 2005), offer a possible solution, as the grid cell system can be seen as a general spatial coordinate system. Given the grid cell representations of two locations it is possible to compute the distance and angle between them, thus providing the needed metric of space.

Grid cells are thought to update their firing activity based on self-motion information, in other words to perform path integration/dead reckoning (McNaughton et al., 2006). However, for a path integrator to be fully useful for navigational tasks, an agent should be able to use this information to find its way back to previously visited locations. In this paper we shall see how the activity of path integrating grid cell networks can be used to guide a simulated agent toward a remembered goal location.

2 Related Work

Path integration is the basis for several computational models of grid cells, the collection of which can roughly be divided into two major categories; oscillatory-interference models and attractor-network models (Giocomo et al., 2011). Several biologically-inspired models for navigation have used such models of grid cells (Milford and Schulz, 2014; Spiers and Barry, 2015). Often this has been for the purpose of position tracking, as in the bio-inspired robot navigation system “RatSLAM” (Milford and Wyeth, 2010), but in some cases grid cells have also been used for direction-finding.

Erdem and Hasselmo (2012) use a model with oscillatory interference-based grid cells to find directions to remembered goal-locations. The mechanism involves testing a number of “look-ahead probes” that trace out linear beams radially from the current location of the agent. Each of these probes orchestrate activity across the entire population of grid cells and place cells to make it appear as if the agent were actually situated at the tested coordinates. If any of the successive locations tested during a given look-ahead probe triggers a reward-associated place cell, the agent is impelled to travel in the specific direction of that probe. Kubie and Fenton (2012) show how a Hebbian learning mechanism between conjunctive grid cells can train the grid cell networks to be able to generate look-ahead trajectories similar to those suggested by Erdem and Hasselmo (2012). The authors propose that this is a “viable candidate for vector-based navigation”. Common to these two approaches is the requirement for the model to explicitly test a wide range of different directions emanating from the current position, in order to expectedly trigger the goal-reward in some specific direction. In this sense, we can term these models *active* mechanisms for goal-directed navigation.

The model proposed in this paper goes the other way about the problem, by presuming that the grid cell-representation of the goal location is known beforehand. The current grid cell state and the goal grid cell state propagate through a pre-wired network of neurons that calculate the offsets between the two representations in order to generate a direction signal. This process can be constantly ongoing in the background, without requiring exclusive use of or otherwise interfering with any grid cell or place cell population for the purpose of “simulating” forward locations. We thus consider our model a *passive* mechanism for goal-directed navigation.

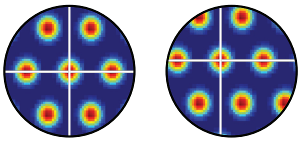


Figure 1: Idealized illustration of the activity of two different grid cells, as a function of the 2D location of the agent, in a cylindrical enclosure of e.g. two meters in diameter (top-down view). Red and blue colors indicate high and low firing rates, respectively. The two grid cells belong to the same grid module, as per the identical scale and rotation of their respective grid patterns, but are seen to have different phases, i.e. different offsets of these patterns, as indicated by the lines.

3 Background

3.1 Grid Cells are Organized in Modules

The name “grid cell” stems from the spatial activity patterns of these neurons; the cells are not active only within single spatial fields in the environment as the place cells are, but have a periodic pattern of activity that repeats at the vertices of a triangular tiling of the plane. The result is a hexagonal grid pattern, extending indefinitely throughout space, that can be characterized by the three properties of scale, rotation and phase—respectively the distance between two neighboring vertices of the grid pattern and the rotation and translation of the grid pattern compared against a frame of reference. A grid cell does not operate in isolation, but participates in a *module* of grid cells that share the same scale and rotation of their individual firing patterns (Stensola et al., 2012). The only distinguishing property between neurons within the same grid cell module is thus that of their phase (Figure 1).

Assuming that a sufficient number of grid cells belong to a given module, the module as a whole has the ability to encode a given set of 2D coordinates in a nearly continuous fashion. The limitation lies in the periodic nature of the grid cell pattern, in that the information carried by a grid cell module can only be interpreted relative to one specific hexagonal tile of the infinitely repeating pattern. A possible solution comes from the fact that the entorhinal cortex harbors grid cell modules of multiple different scales (Stensola et al., 2012). It is conceivable that the smaller-scaled grid cell modules represent space at a finer resolution than the larger-scaled ones, but with the sacrifice

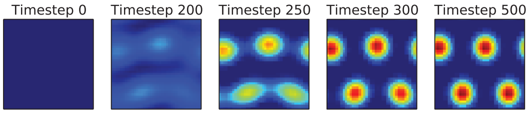


Figure 2: Spontaneous formation of a grid-like activity pattern in the neural sheet of an attractor-network grid cell module, due to random initial conditions and the recurrent connectivity.

of having shorter "ranges of validity" due to the more rapid periodicity of the grid pattern. The activity of all modules taken collectively, however, ought to retain both the low-precision/long-range information of the larger-scaled modules as well as the high-precision/short-range information of the smaller-scaled modules. The utilization of this multi-scale mix of information is a key idea behind the model presented in this paper.

3.2 Attractor-Network Models of Grid Cells

Attractor-network models of grid cells conceptualize their neurons as being laid out in a 2D neural sheet. Proximity between neurons in this sheet implies that the neurons should have similar phases of their grid patterns, not necessarily that the neurons would be co-located in the brain. The neurons are recurrently connected to each other, with a connectivity profile based on distances in the neural sheet, in such a way that grid-like patterns of activity will form in the network from random initial conditions (Figure 2).

These network patterns, which are the attractor states of the network, can then be made to shift around in the network in response to self-motion signals in order to perform path integration. Assuming these shifts consistently reflect the actual movements of the agent, the network pattern will over time become visible in the spatial activity patterns of *individual* neurons in the network. Attractor-models of grid cells thus have grid-like patterns both in their time-averaged spatial activity plots (Figure 1) and in their momentary network activity plots (Figure 2), and this is an important distinction to be aware of.

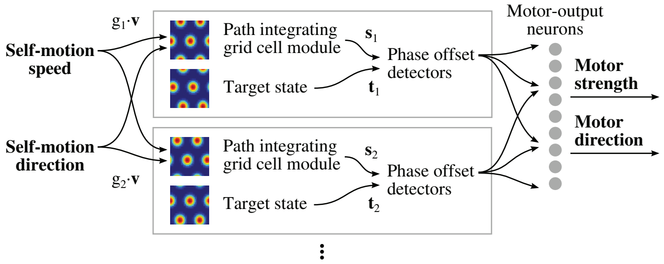


Figure 3: A schematic overview of the model.

4 Model

The main part of the model comprises a configurable number of *modules*, seen as the two rectangular blocks in the middle of Figure 3. Each module m consists of (a) a grid cell module, (b) a *target signal*, and (c) a network of *phase offset detectors*. The grid cell modules perform path integration on the incoming self-motion signal (composed of speed and direction), and output vectors of grid cell-activity $\mathbf{s}_m$ that are passed on to the corresponding networks of phase offset detectors. These phase offset detectors also receive a copy of the intended grid cell-activity vector $\mathbf{t}_m$ for the desired target location—the "target signal". The task of the phase offset detectors is to find the required direction of travel to make up for the offset in the grid patterns between the path integrator signal $\mathbf{s}_m$ and the target signal $\mathbf{t}_m$. The intended outputs of the model are a motor direction signal, giving the direction toward the target location, and a motor strength signal, indicating whether the agent has arrived at the target location or to keep going.

4.1 Multiple Modules with Different Spatial Scales

The model has these multiple parallel modules in order to utilize information from a variety of grid cell modules representing space at different scales; this will provide the direction-finding process with long-range/low-precision signals as well as short-range/high-precision signals. The different grid scales are achieved by modulating the velocity inputs to each grid cell module. The velocity signal to module m is multiplied by the gain factor g_m before reaching the grid cell network. Smaller gain factors will cause the path integrator to respond more slowly to the same velocity inputs, thus causing the grid to appear larger, and vice versa.

The path integrator model used in these simulations was found to respond acceptably to velocity inputs at least in the range from 0.1 m/s to 1.2 m/s. As the actual speed of the simulated agent was fixed to 0.2 m/s, the range of acceptable gain factors could then be determined to be $[g_{min}, g_{max}] = [0.5, 6.0]$. The model uses a geometric progression from g_{min} to g_{max} for the gain factors. Given a specific number of modules M to be used, the g_m values can then be calculated as

$$R = \sqrt[M-1]{g_{max}/g_{min}}, \qquad g_m = g_{min} \cdot R^{m-1}. \tag{1}$$

4.2 Path Integrating Grid Cell Modules

The path integrator modules are closely based on the attractor-network grid cell model by Burak and Fiete (2009), and the following formulas are based on their presentation of the model. Each grid cell module consists of a 2D sheet of neurons of size $n \times n$, where $n = 40$. The activation values of these $n^2 = 40^2 = 1600$ neurons are contained in a vector $\mathbf{s}$, fully representing the current state of the path integrator.

Each grid cell i receives recurrent inputs from all other neurons in $\mathbf{s}$. Let $\mathbf{x}_i$ be the neural sheet coordinates of neuron i. The weight from afferent neuron i' onto neuron i can then be calculated from the connectivity profile $rec(d)$ by letting d be the shortest distance between $\mathbf{x}_{i'}$ and $\mathbf{x}_i$ in the neural sheet, taking into consideration that connectivity may wrap around the N/S and W/E edges. The recurrent connectivity profile $rec(d)$ is a difference of Gaussians, seen as the inhibitory "doughnut" in Figure 4, top left. Specifically,

$$rec(d) = e^{-\gamma d^2} - e^{-\beta d^2}, \tag{2}$$

where $\gamma = 1.05 \cdot \beta$, $\beta = \frac{3}{\lambda^2}$ and $\lambda = 15$. λ approximately specifies the periodicity of the grid cell network, i.e. the number of neurons from one peak of activity to the next.

To express the update rule for grid cell i using vector notation, let $\mathbf{w}_{\mathbf{c}}^{rec}$ be the weight vector derived from the distance-to-weight-profile $rec(d)$ centered on the point **c** in the neural sheet. The update rule can then be described as

$$\tau \frac{ds_i}{dt} + s_i = f\left(\mathbf{s} \cdot \mathbf{w}^{rec}_{\mathbf{x}_i - \hat{\mathbf{e}}_{\theta_i}} + B_i\right), \tag{3}$$

solved for ds_i, where $dt = 10$ ms, $\tau = 100$ ms, $f(x) = max(0,x)$ and **s** is the vector of the activation values at the end of the previous timestep.

The center point **c** of the connectivity profile for efferent neuron i is here given as $\mathbf{x}_i - \hat{\mathbf{e}}_{\theta_i}$, i.e., there is an extra offset of $\hat{\mathbf{e}}_{\theta_i}$ in addition to $\mathbf{x}_i$ when positioning the connectivity profile for neuron i. The offset $\hat{\mathbf{e}}_{\theta_i}$ is the unit vector in the direction of θ_i, which in turn is the *directional preference* of neuron i. The directional preference is used to shift the activity pattern among the grid cells in response to asymmetrical velocity inputs. Preferences for each of the four cardinal directions are distributed among the neurons in each 2×2 block of neurons. Namely, the x, y coordinates of a neuron are used to calculate an index $(2 \cdot (y \bmod 2) + x \bmod 2)$ into the list $[W, N, S, E]$ to determine θ_i. In the absence of velocity inputs, the four distinct preference-offsets counterbalance each other to keep the activity pattern at rest in the network. During motion, however, the external input B_i to each neuron becomes velocity-tuned according to the directional preference of the neuron. This input is calculated as

$$B_i = 1 + g_m \alpha \hat{\mathbf{e}}_{\theta_i} \cdot \mathbf{v}, \tag{4}$$

where **v** is the movement velocity and $\alpha = 0.10315$ is a scaling constant specified by Burak and Fiete (2009).

4.3 Phase Offset Detectors

The vector of activation values **s** is passed on to a network of phase offset detectors. In addition to receiving the input vector **s** from the path integrating grid cell module, the phase offset detectors also receive a similarly-shaped vector **t** that represents the grid cell activity of the target location, i.e. a grid cell-encoding of the desired target coordinates.

Each phase offset detector j has an associated origin location $\mathbf{x}_j$ and a preference direction θ_j. The neuron is tuned to respond when an activity peak is near the origin location $\mathbf{x}_j$ in the neural sheet of the path integrator grid cell module (**s**) and there *simultaneously* is an activity peak near the location $\mathbf{z}_j = \mathbf{x}_j + \delta \hat{\mathbf{e}}_{\theta_j}$ in the grid cell-encoded target-location-input (**t**). Specifically, the activation value of phase offset detector j is calculated as

$$p_j = f\left(\mathbf{s} \cdot \mathbf{w}^{in}_{\mathbf{x}_j} + \mathbf{t} \cdot \mathbf{w}^{ex}_{\mathbf{z}_j}\right), \tag{5}$$

where **w** again refers to weight vectors derived from given connectivity profiles centered on given points in the neural sheet, but with new connectivity profiles *in* and *ex*. The path integrator inputs **s** are fully connected using the connectivity profile $in(d)$ centered at $\mathbf{x}_j$, while the target location inputs **t** are fully connected using the connectivity profile $ex(d)$ centered at $\mathbf{z}_j$. These connectivity profiles are defined as

$$in(d) = \eta \cdot \left(e^{-\beta d^2} - 1\right), \qquad ex(d) = e^{-\beta d^2}, \tag{6}$$

where $\eta = 0.25$. The offset length δ is set to be $\delta = 7$, in the neighborhood of half of λ.

The effect is to respond the most strongly when there is an offset of length δ in direction θ_j between the activity patterns in **s** and **t**, given that the path integrator currently has activity in the vicinity of $\mathbf{x}_j$. An example situation is shown in Figure 4, where a phase offset detector with $\mathbf{x}_j = (20, 20)$ and $\theta_j = 45°$ receives inputs of favorable characteristics from **s** and **t**.

In order for the network of phase offset detectors to work independently of the current location of network activity in the path integrator, there needs to be a sufficient number of phase offset detectors that sample different origin locations $\mathbf{x}_j$. Additionally, the network needs to sample a range of different preference directions θ_j. This is realized using two parameters S_θ and S_{xy} that respectively specify the number of directions sampled in the interval $[0, 2\pi)$ and the number of steps to use along each of the two dimensions of the neural sheet when sampling origin locations. The total number of phase offset detectors will then be $S_\theta \cdot S_{xy}^2$ per module.

4.4 Motor-Output Neurons

The activity from the phase offset detectors are aggregated in a set of *motor-output neurons*. Whereas the grid modules and phase offset detectors are instantiated separately for each module, the motor-output neurons comprise a

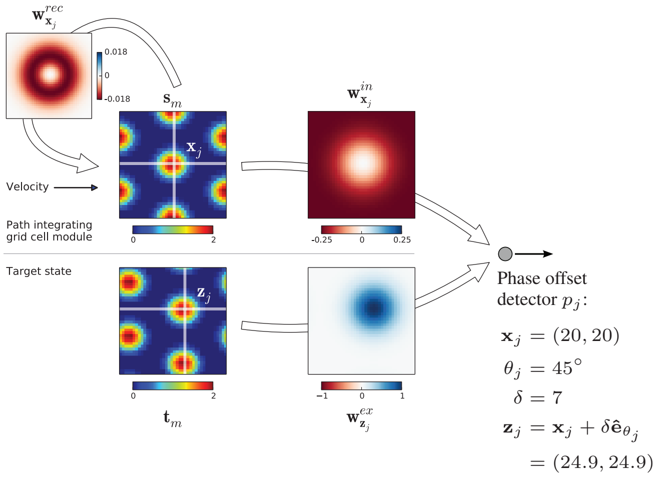

$\mathbf{w}^{rec}_{\mathbf{x}_j}$
0.018
0
-0.018
$\mathbf{s}_m$
$\mathbf{x}_j$
$\mathbf{w}^{in}_{\mathbf{x}_j}$
Velocity
Path integrating grid cell module
0
2
-0.25
0
0.25
Target state
$\mathbf{z}_j$
0
2
−1
0
1
$\mathbf{t}_m$
$\mathbf{w}^{ex}_{\mathbf{z}_j}$
Phase offset detector p_j:
$\mathbf{x}_j = (20, 20)$
$\theta_j = 45^\circ$
$\delta = 7$
$\mathbf{z}_j = \mathbf{x}_j + \delta \hat{\mathbf{e}}_{\theta_j}$
$= (24.9, 24.9)$

Figure 4: Example of a phase offset detector, p_j, showing the input networks $\mathbf{s}$ and $\mathbf{t}$ and the connectivity profiles with which these two networks are connected to p_j. Depicted matrices are 40×40.

common network receiving inputs from all of the modules. The number of motor-output neurons is the same as the number of sampled preference directions S_θ in the phase offset detector networks. The motor-output neurons sample the same directions as the phase offset detectors.

The activity in each motor-output neuron is essentially the sum of the activity in all of the phase offset detectors that share preference direction with the motor neuron. An important detail, however, is that these contributions are weighted by the inverse of the gain factors of their respective modules. In other words,

$$u_k = \sum_m \left[g_m^{-1} \sum_j^{\theta_j = \theta_k} p_{m,j} \right], \tag{7}$$

where u_k is the activity of motor-output neuron k and θ_k is the preference direction of k. This weighting will give priority to the direction signals from the modules with low gain factors g_m, i.e. the modules where the quality of the path integration information is long-range-applicable but with low precision. As the agent gets closer to the target location, the intention is for these signals to fade off to sufficiently weak strengths so that the shorter-range, higher-precision signals will pick up in motor influence. The purpose is to achieve the trade-off of a long-range *and* high-precision signal.

To calculate the final motor-output signal $\boldsymbol{\Theta}$, the values of u_k are considered as vector contributions in the direction of θ_k, i.e. the vectors $u_k \cdot \hat{\mathbf{e}}_{\theta_k}$ are summed together, and this sum is then scaled to compensate for the variable number of inputs and their weighting. The final calculation is thus

$$\boldsymbol{\Theta} = \rho \cdot \sum_k u_k \cdot \hat{\mathbf{e}}_{\theta_k}, \quad \rho = \frac{1}{S_\theta \cdot S_{xy}^2 \cdot \sum_m g_m^{-1}}, \tag{8}$$

whereafter the angle of $\boldsymbol{\Theta}$ makes up the motor direction signal and the vector length becomes the motor strength signal.

4.5 Experiment Setup

Each experiment trial consists of a succession of stages, specifically (a) pattern formation in grid cell modules, (b) capture of path integrator states into target states, (c) the agent performing a random walk for T seconds, and (d) the agent attempting to return "home" to the target location.

At the beginning of the simulation, in order for the grid cell networks to form grid-like activity patterns, all s_i values are initialized randomly in the range $[0, 10^{-4})$ before the networks are then allowed to settle for 1000 timesteps (Figure 2). When this pattern formation process is done, the grid-like activity patterns will have been initialized to essentially random starting-coordinates. The model now copies these activity patterns $\mathbf{s}_m$ into the target state vectors $\mathbf{t}_m$. The m different target state vectors $\mathbf{t}_m$ henceforth remain unchanged for the rest of the trial, as a memory of the coordinates of the starting location ("home").

The agent then performs a random walk for a configurable duration of time T seconds. The time duration for a single iteration of the model has been set to be 10 ms, so there are 100 timesteps/s. During both the random-walk and the return-home stages, the agent moves with a constant speed of 0.2 m/s with only the movement direction changing. The random walk starts with a uniformly distributed random value from $[0, 2\pi)$ as the movement direction. At every timestep it is updated by adding a radian value from a normal distribution with $\mu = 0$, $\sigma = 1$.

After T seconds have elapsed, the return-home stage begins. The motor-direction output from the network is used to set the movement direction of the agent, whereas the motor-strength output is used as a termination criterion for determining when to end the trial. Three different termination criteria are used; (a) the motor-strength signal is less than 10^{-6}, (b) the return-home stage has lasted for at least a second and the straight-line distance to the point traversed one second ago is less than 0.01 m, or (c) the return-home stage has lasted $2 \cdot T$ seconds. Whichever termination criteria ends the trial, the straight-line distance to the starting location from the final stopping location is deemed the *error* of the trial. The favorable outcome is a low overall error value.

4.6 Parameter Search

A parameter search was conducted to find the best values for M, S_θ and S_{xy} to use for the rest of the experiments. An exhaustive test was performed on all combinations of values in the intervals $M \in [2,6]$, $S_\theta \in [4,32]$, $S_{xy} \in [5,40]$. However, to penalize expensive solutions and to place an upper bound on the complexity of the solutions to be tested, a *synapse cost* C was calculated for each parameter combination. This value provides an estimate on the number of synapses in the model and consequently a rough estimate on the number of floating-point operations required to update the model (without optimization). C was calculated as $C = M \cdot \left(n^2 \left(n^2+1\right) + 2n^2 \cdot S_\theta \cdot S_{xy}^2 + S_\theta \cdot S_{xy}^2\right)$, with the three terms representing the synapse cost to operate respectively a grid cell module, the phase offset detectors and the axons to the motor-output neurons. Only the combinations with $C < 10^8$ were tested, leaving 2685 combinations to test. For each combination, 100 trials with a random-walk duration of $T = 30$ s were performed and the mean error was reported (Figure 5a). The parameter combination $M = 4$, $S_\theta = 28$, $S_{xy} = 9$ was selected for further use (highlighted).

To get a sense for how the individual parameters affect the outcome, new sets of runs were performed where each parameter in turn was changed within the defined intervals and evaluated over 100 trials, while the two other parameters were left unchanged (Figure 5b). M and S_{xy} seem to affect the results little

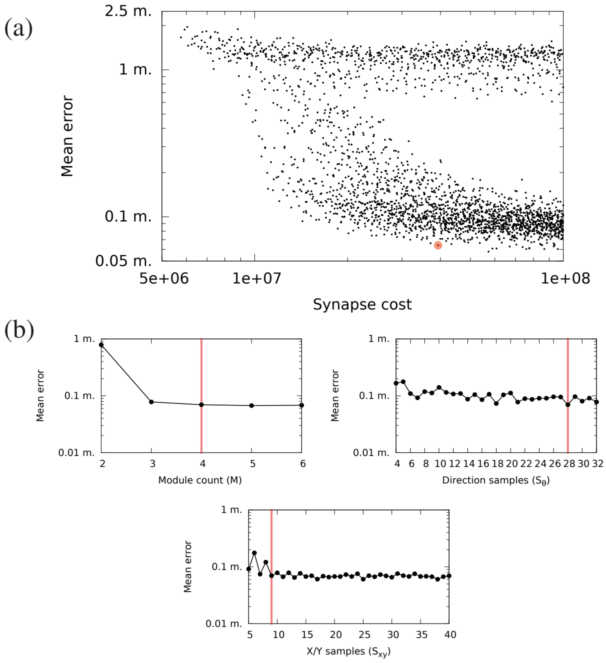


Figure 5: (a) Scatter plot of parameter combinations tested during the parameter search, with each dot showing the mean error over 100 runs for a given parameter combination, plotted against the respective synapse cost. (b) For each of the three parameters M, S_θ and S_{xy}, the effect of modifying that parameter from the chosen parameter combination ($M = 4$, $S_\theta = 28$, $S_{xy} = 9$; indicated by vertical lines) while leaving the other parameters unchanged. Mean error over 100 runs. The combination $M = 4$, $S_\theta = 28$, $S_{xy} = 9$ is represented by the same set of trials in all three figures.

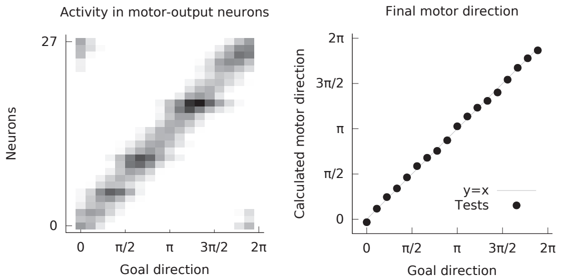


Figure 6: Example of network operation at locations a fixed radius $r = 0.5$ m away from the home location in given directions. Left: Momentary activity in the $S_\theta = 28$ motor-output neurons. Right: Final motor direction calculated from the motor-output neurons, plotted at the closest 2π period to the goal direction.

above thresholds of respectively $M = 2$ and $S_{xy} = 8$. S_θ, on the other hand, appears to be more sensitive to the particular value to which it is assigned.

4.7 Implementation Details

All random values used by the implementation in this paper were generated using the Mersenne Twister pseudo-random number generator included with the C++11 standard library. The model and the simulator used single-precision floating-point values throughout.

5 Results

5.1 Direction-Finding Ability

Two different examples of how the system operates in practice are presented in Figures 6 and 7. In the first example, the direction-finding ability of the model is tested at multiple points along a circle centered on the goal location. For each of the 18 uniformly spaced directions tested, the agent was driven a distance of 0.5 m in the opposite direction of the intended "goal direction" and allowed to settle for 250 timesteps before the motor outputs were examined. For each trial, the figure shows the recorded activity from all S_θ motor-output

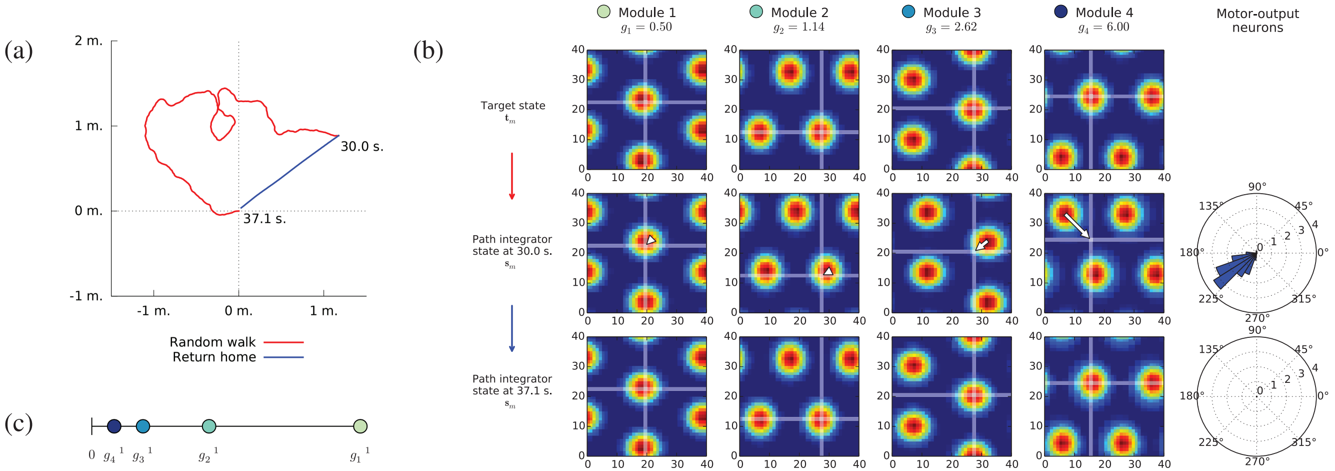


Figure 7: Example of network operation during various stages of a trial. (a) Trace of the trajectory followed during random-walk and return-home stages. (b) Top row: Target state. Rest: Momentary activity of the grid cell modules and motor-output neurons at various points in time. (The s_i and t_i values were arranged according to $\mathbf{x}_i$ and convolved with a uniform 2×2 filter to hide artifacts due to different preference-offsets. This procedure was used for all **s** and **t** visualizations in this paper.) (c) Plot of the inverses of the gain factors g_m used when the model is configured for four modules ($M = 4$), in order to illustrate the difference in the ranges of the modules.

neurons as well as the motor-direction signal from the model. As evidenced by the figure, the model is able to accurately calculate the goal direction at this specific distance of $r = 0.5$ m.

Figure 7 demonstrates a full trial with both random-walk and return-home stages as described above. After a $T = 30$ s random walk, the agent successfully attempts a return to the home location. The figure includes the momentary activity of all of the grid cell modules ($\mathbf{s}_m$) both at the beginning and the end of the return-home stage. In each of these cases, the motor-neuron activity is also shown. The plots of $\mathbf{s}_m$ show possible interpretations of how the activity patterns might have shifted from the target state $\mathbf{t}_m$, which was also the initial state of the grid modules at $T = 0$ s. From the leftmost to the rightmost columns, the grid modules progress from long-range/low-precision to short-range/high-precision. The first, second and third modules show a correct assessment of the goal direction at $T = 30$ s, whereas the fourth module is "out of range" and in this case has an ambiguous response.

At $T = 37.1$ s, we see that the grid modules have aligned closely with the corresponding target states. The trial thus terminated because of the weak motor-strength signal, bringing the agent to a halt at a distance of 4.74 cm from the goal location, from an initial goal distance of 1.47 m at the end of the random walk.

5.2 Effect of Multiple Grid Modules

The effect of using multiple grid modules is further demonstrated in Figure 8, which shows, as a function of the distance to the goal, the strengths and errors of each module's contribution to the motor-output network when seen in isolation. To illustrate their relative influences, the signal strength of each module is also shown in terms of its ratio of the sum. Lastly, the final motor-direction error is shown overlaid on the direction-error plots from the individual modules.

For each module, there is a distinctive bell-shape in the strength curve as the tested radius approaches and recedes from the "optimal detection distance" of the module's offset detectors. The vicinity of the peak of the bell curve is also where the module's direction-error is at a minimum. Past this region,

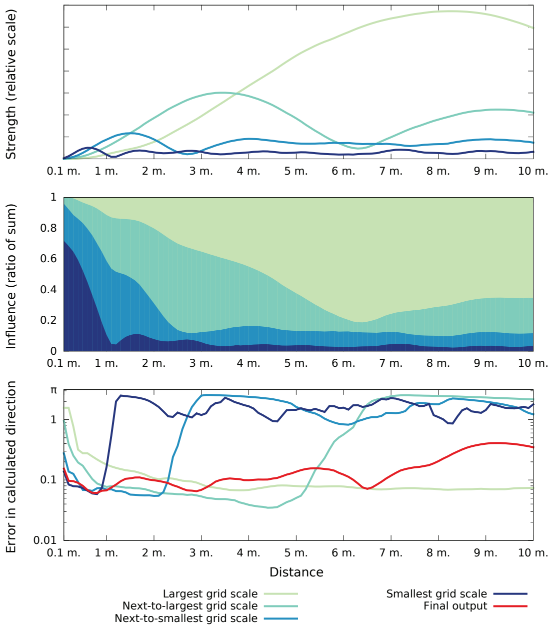


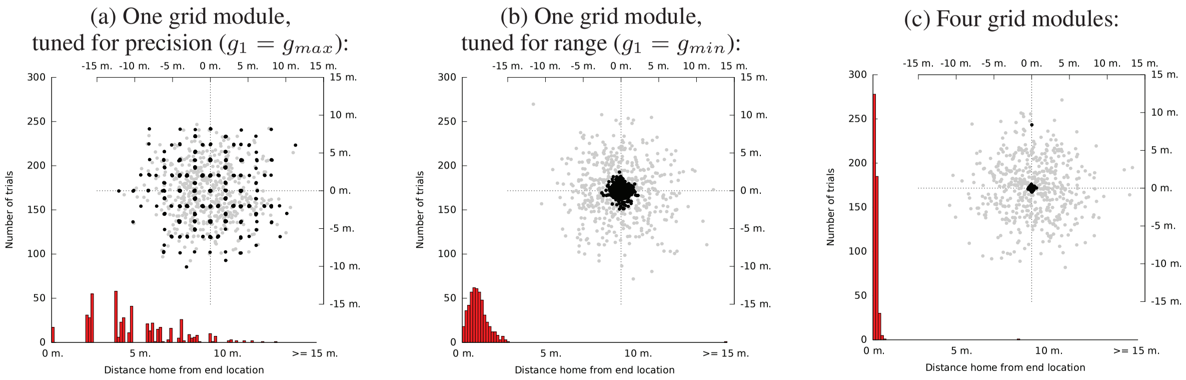
Figure 8: Behavior of the $M = 4$ different modules at increasing distances from the goal. The model was tested as in Figure 6, but r was varied at 0.1 m increments in the interval $[0.1\text{ m}, 10\text{ m}]$. For each tested radius the motor-output strengths and direction-signal errors are reported as the mean over 18 tested directions. In order to report values individually for each module, extra motor-output networks were instantiated such that each only received phase offset detector-inputs from one given module. For these plots, $\rho = 1$ in order not to cancel out the scaling differences.

the module abruptly becomes unreliable due to the periodicity of the grid cell signal. Because of the gain-based weighting of module contributions, however, one of the larger-scaled modules is able to overpower the contributions of the smaller-scaled modules and thus ensure that the final direction signal is still valid. As seen by the red line in the lower diagram, the final direction-output achieves a trade-off between range and precision not seen in any of the individual modules.

Figure 9 demonstrates the importance of this combination of precision and range information. The figure contains results from three different sets of 500 trials, each with $T = 180$ s. Whereas the rightmost diagram shows the results from trials with the default parameters ($M = 4$, $S_\theta = 28$, $S_{xy} = 9$), the two other diagrams only use one module ($M = 1$). The leftmost diagram has the gain factor set to $g_1 = g_{max}$, for rapid periodicity and short-range/high-precision signals, while the middle diagram has $g_1 = g_{min}$, i.e. tuned for long-range/low-precision signals.

The distributions of termination locations seen in the scatterplots confirm our expectations from the known qualities of the grid module signals. With one module tuned for precision (Figure 9a), the agent either precisely returns home to the target location or it ends up in an attractor location that is part of a repeating pattern of possible attractors. This shows the periodic nature of the grid cell encoding of space. With one module tuned for range (Figure 9b), all but one of the 500 trials terminate in a cluster centered on the target location. However, the improved range has carried a penalty of worse precision. This penalty is seen to be mostly alleviated by integrating information from multiple grid modules; in Figure 9c, where four grid modules are used, all but six of the 500 trials end up within 0.5 m of the target, with only one ending up more than 1 m away.

To get a sense for the trajectories the model follows during these attempts to reach the target, Figure 10 contains traces of the 50 runs from Figure 9c with the farthest goal distance at the end of the random walk. With some exceptions, the paths taken are all largely straight lines toward the target. All but one trajectory (seen near the top) end up within 0.5 m of the goal.



(a) One grid module, tuned for precision ($g_1 = g_{max}$):

(b) One grid module, tuned for range ($g_1 = g_{min}$):

(c) Four grid modules:

Figure 9: End locations after returning home, for various configurations of grid cell modules. 500 runs for each configuration. Gray dots show end of random walk and black dots show end of return home. Histograms below show the distribution of goal distances at trial termination, with a bin size of 15 cm. (The upper-left run in part b stopped at a distance of 29.7 m from the target.)

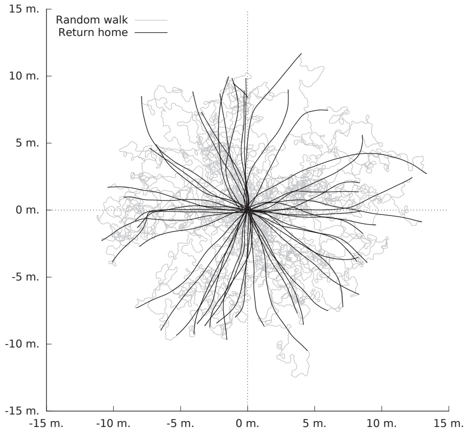


Figure 10: Traces of the 50 runs from Figure 9c with the farthest distance from the goal at the end of the random-walk stage.

6 Discussion

The basis for this project was to use neural representations of space for direction-finding in a robot. The paper has presented a model that integrates an existing model of path integrating grid cells with a novel mechanism that is able to use the grid cell representation to direct the agent to a remembered goal. The successful simulation results show promise for using the model in a physical robot in the future. The translation into the physical world will bring with it its own set of challenges, such as noisy self-motion inputs and imprecise motor control. The integration of sensory information into the model is thus one important area for further study, as has been done in other grid cell-based robot controllers (Milford and Wyeth, 2010).

We consider the model at its current abstraction level to be biologically plausible. The inputs and outputs of the model are geocentric direction and speed signals, which is supported by the existence of head-direction cells. Attractor-network models are considered viable candidates for understanding the operation of grid cells, and the phase offset detectors and motor-output neurons are simple input-summing neurons. The target state signals are assumed to be a grid cell-encoding of the target coordinates; this could be provided in the form of backprojections from the hippocampus.

Chadwick et al. (2015) used VR-supported fMRI to look for a goal-direction signal in the human brain. They found that there would be similar brain activity patterns in the entorhinal/subicular region when a given geocentric direction was used as either the current facing direction or the goal direction, and the activity patterns were found to be best accounted for as a mixture of the encodings of the facing direction and the goal direction. The authors see these results as evidence that some form of goal-directed simulation of spatial representations is involved in navigation, citing the model by Erdem and Hasselmo (2012) as an example of how this mechanism could work.

This is not the only possible conclusion of these results, especially since the model by Erdem and Hasselmo requires simulation of look-ahead trajectories in many different directions from the current location in order to discover the goal location. A mechanism similar to the one presented in this paper would allow the goal direction to be calculated directly from grid cell representations of the current location and the goal, avoiding the need for extensive simulations in multiple directions. The mixture representations reported by Chadwick et al. could still conceivably be accounted for by oscillations in the entorhinal/subicular region between encodings of the present and the future spatial states. Experiments at finer spatial and temporal resolutions would hopefully be able to distinguish the extents of these two types of contributions.

7 Acknowledgements

The author is grateful to Keith Downing and Trygve Solstad for helpful discussions, feedback and advice.

References

Burak, Y. and Fiete, I. R. (2009). Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, 5(2):e1000291.

Chadwick, M. J., Jolly, A. E., Amos, D. P., Hassabis, D., and Spiers, H. J. (2015). A goal direction signal in the human entorhinal/subicular region. *Current Biology*, 25(1):87–92.

Erdem, U. M. and Hasselmo, M. (2012). A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *European Journal of Neuroscience*, 35(6):916–931.

Giocomo, L. M., Moser, M.-B., and Moser, E. I. (2011). Computational models of grid cells. *Neuron*, 71(4):589–603.

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

Kubie, J. L. and Fenton, A. A. (2012). Linear look-ahead in conjunctive cells: an entorhinal mechanism for vector-based navigation. *Frontiers in neural circuits*, 6.

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

Milford, M. and Schulz, R. (2014). Principles of goal-directed spatial robot navigation in biomimetic models. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655):20130484.

Milford, M. and Wyeth, G. (2010). Persistent navigation and mapping using a biologically inspired SLAM system. *The International Journal of Robotics Research*, 29(9):1131–1153.

O’Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1):171–175.

Spiers, H. J. and Barry, C. (2015). Neural systems supporting navigation. *Current Opinion in Behavioral Sciences*, 1:47–55.

Stensola, H., Stensola, T., Solstad, T., Frøland, K., Moser, M.-B., and Moser, E. I. (2012). The entorhinal grid map is discretized. *Nature*, 492(7427):72–78.

Paper B

Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network (Edvardsen, 2016)

Author(s):

Vegard Edvardsen

Published in journal:

Natural Computing

Note:

This paper is an extended version of Paper A, and contains most of the material from that paper as well as extra background material.

The paper was accepted and published online in 2016, and was originally cited as such. However, the paper was recently assigned to a specific journal issue, published this year, and an updated reference would therefore be *Natural Computing*, vol. 18, no. 1, pp. 13–27, 2019. For consistency, the former version of the reference has been retained throughout the thesis.

Paper B

Edvardsen (2016)

Goal-Directed Navigation based on Path Integration and Decoding of Grid Cells in an Artificial Neural Network [*]

Vegard Edvardsen

Department of Computer and Information Science,
Norwegian University of Science and Technology, 7491 Trondheim, Norway

Abstract

As neuroscience gradually uncovers how the brain represents and computes with high-level spatial information, the endeavor of constructing biologically-inspired robot controllers using these spatial representations has become viable. Grid cells are particularly interesting in this regard, as they are thought to provide a general coordinate system of space. Artificial neural network models of grid cells show the ability to perform path integration, but important for a robot is also the ability to calculate the direction from the current location, as indicated by the path integrator, to a remembered goal. This paper presents a neural system that integrates networks of path integrating grid cells with a grid cell decoding mechanism. The decoding mechanism detects differences between multi-scale grid cell representations of the present location and the goal, in order to calculate a goal-direction signal for the robot. The model successfully guides a simulated agent to its goal, showing promise for implementing the system on a real robot in the future.

1 Introduction

This paper presents a brain-inspired neural network capable of performing goal-directed navigation in a simulated robot. The neural network receives

[*] This paper is an extended version of a previously published conference paper (Edvardsen, 2015). Sections 5 and 6 in this paper report on the same data as the earlier conference paper.

information about the robot's self-motion velocity, and based on these inputs performs path integration by updating an internal estimate of the robot's current coordinates. Later on, when given the coordinates of a goal location, the network is able to calculate the direction toward the goal based on its internal estimate of the current coordinates from the path integration process. This position estimate in the neural network is maintained using the spatial representation of *grid cells*, known from neurophysiological experiments in rats (Hafting et al, 2005). Grid cells are thought to constitute a spatial coordinate system in the brain and to participate in path integration processes (McNaughton et al, 2006), and for these reasons they are a compelling source of biological inspiration for a coordinate system in a neural robot controller.

Grid cells are found in the medial entorhinal cortex, close to the hippocampus. The hippocampal region is rich in neurons that represent high-level features of the animal's spatial context—in addition to grid cells, this area of the brain also contains cell types such as place cells (O'Keefe and Dostrovsky, 1971), head-direction cells (Taube et al, 1990) and border cells (Solstad et al, 2008). These findings gradually uncover the neural basis for navigation in the brain, and offer a window into how the brain computes with and represents abstract cognitive features. Inspired by these advances, the basis for this project has thus been to devise and implement a neural model to enable a robot to find its way to a previously visited goal location using these neural representations of space known from the brain. Through crafting these models, we hope to gain insights into how these spatial representations might be utilized for navigational purposes by neural systems, artificial and real alike.

Section 2 describes related work, Section 3 presents relevant background material, Section 4 details the implemented system and the proposed decoding mechanism, while Section 5 describes the experimental setup. Section 6 presents our results, which are then discussed in Section 7.

2 Related Work

The first evidence of spatially responsive cells in the rat hippocampus came with the discovery of place cells, which were seen to respond at distinct locations in the environment (O'Keefe and Dostrovsky, 1971). However, place cells do not appear to encode any metric information, such as distances and angles (Spiers and Barry, 2015). The place cell representation by itself is thus not sufficient to be able to navigate between arbitrary locations, because it does not offer any means to calculate the direction of travel from one place cell's firing location to that of another. Possible solutions to this problem in a robotic context include learning the required directions of movement for given place cell transitions (Giovannangeli and Gaussier, 2008).

Grid cells offer another possible solution, as the grid cell system can be seen as a general spatial coordinate system. Given the grid cell representations of two locations, it is possible to compute the distance and angle between them, thus providing the needed metric of space. Several biologically-inspired models for navigation have indeed used models of grid cells (Milford and Schulz, 2014; Spiers and Barry, 2015; Bush et al, 2015). Often this has been for the purpose of position tracking, as in the bio-inspired robot navigation system "RatSLAM" (Milford and Wyeth, 2010), but in some cases grid cells have also been used for direction-finding.

Erdem and Hasselmo (2012) consider a model with interconnected grid cells and place cells, and introduce a mechanism for finding directions to remembered goal-locations using these neurons. The mechanism involves testing a number of "look-ahead probes" that trace out linear beams radially from the current location of the agent. Each of these probes orchestrate activity across the entire population of grid cells and place cells to make it appear as if the agent were actually situated at the tested coordinates. If any of the successive locations tested during a given look-ahead probe triggers a reward-associated place cell, the agent is impelled to travel in the specific direction of that probe (Erdem and Hasselmo, 2012, 2014; Erdem et al, 2015). An important question is how this mechanism might be implemented in a neural network. Part of the answer might be provided by Kubie and Fenton (2012), who show that a Hebbian learning mechanism between conjunctive grid cells can train the grid cell networks to be able to generate look-ahead trajectories similar to those suggested by Erdem and Hasselmo (2012).

Common to these approaches is the requirement for the model to explicitly test a wide range of different directions emanating from the current position, in order to expectedly trigger the goal-reward in some specific direction. An alternative approach is to decode the grid cell representations of the current location and the goal location and to calculate the movement direction directly from this. A recent overview of neural network models for such navigation with grid cells is provided by Bush et al (2015); as enumerated by them, several neural models have been proposed for decoding grid cell signals: Fiete et al (2008) consider the grid cell system to implement a residue number system, and propose that grid cells could be read out using a neural architecture earlier proposed by Sun and Yao (1994) for decoding such systems. The decoding process involves settling a recurrent neural network into a stable state, whereafter the decoded position value would be represented by the firing rate of the network. Huhn et al (2009b,a) show that distance information can be decoded from grid cells through competition among neurons that each receive grid cell inputs of a particular grid scale (for definitions, see Section 3.2). However, this mechanism assumes that there is a large selection of grid orientations for each particular grid scale, which is now known not to be the case (Stensola et al, 2012). Masson and Girard (2011) demonstrate that position can be decoded from grid cells, interpreted as a residue number system, by applying the Chinese Remainder Theorem to the problem and wiring up a neural network to perform the same calculations.

These models are primarily concerned with decoding information from grid cells into explicit distance or position representations—not into goal directions—so these systems would require additional stages to be able to generate the desired movement direction signals. Bush et al (2015) introduce a set of neural network models that do perform the full computation of a movement direction signal from grid cell representations: The "distance cell" model assumes a set of neurons that are pre-wired to activate whenever the grid cells represent specific displacements from an origin. By producing outputs from these neurons in proportion to the recognized distance, the required direction of movement can be determined for arbitrary pairs of grid cell-encoded current and goal locations. The "rate-/phase-coded vector cell" models generate a goal-direction signal by detecting particular combinations—across grid modules—of phase differences between the grid cell representations of the current location and the goal location.

The model presented in this paper resembles Bush et al's latter approach. The current grid cell state and the goal grid cell state propagate through a pre-wired neural network that calculates the offsets between the two representations in

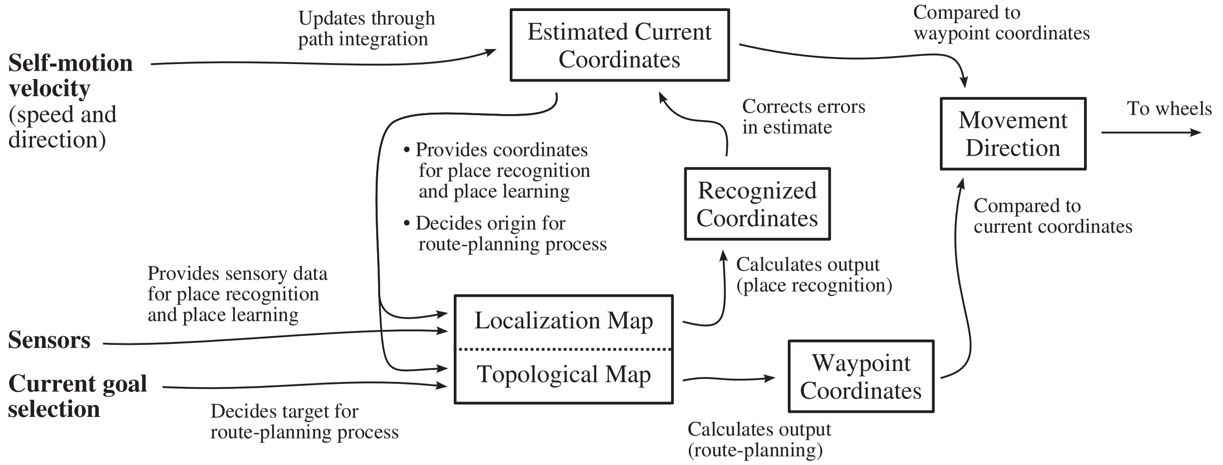

Self-motion velocity (speed and direction)
Updates through path integration
Estimated Current Coordinates
Compared to waypoint coordinates
Movement Direction
To wheels
Corrects errors in estimate
Recognized Coordinates
• Provides coordinates for place recognition and place learning
• Decides origin for route-planning process
Compared to current coordinates
Calculates output (place recognition)
Provides sensory data for place recognition and place learning
Sensors
Localization Map
Topological Map
Waypoint Coordinates
Current goal selection
Decides target for route-planning process
Calculates output (route-planning)

Figure 1: A possible decomposition of the mobile robot navigation problem into subcomponents, as currently envisioned as the future direction for the system presented in this paper. The figure shows how the suggested subcomponents might interact in order to implement path integration, Simultaneous Localization and Mapping and map-based navigation in the robot controller.

order to generate a direction signal. Whereas Bush et al (2015) investigate the problem of grid cell-based navigation in isolation, in this paper we integrate our decoding mechanism with a particular class of computational models for grid cells, namely continuous attractor networks, that are based on path integration. We show that the grid cell signals resulting from path integration processes in the continuous attractor networks, can be successfully used by our proposed decoding mechanism to perform navigation in a simulated robot. This provides a promising building block toward a future neural robot navigation system based on grid cells and place cells.

3 Background

This part of the paper provides background material relevant for the rest of the paper. Section 3.1 describes how the presented system fits into the larger picture of a research project to create a brain-inspired neural robot navigation system, based on grid cells and place cells. Section 3.2 describes how grid cells, as known from neurophysiological experiments, can be understood to implement a general coordinate system of space in the brain. Section 3.3 describes a family of computational models for grid cells known as continuous attractor networks, and how these networks use path integration to model grid cells.

3.1 Robot Navigation with Neural Networks

The eventual goal for the research project of which this paper is a part, is to devise and implement a full robot navigation system for indoor mobile robots, where the majority of the computation takes place in neural networks utilizing spatial representations known from the brain. The robot should be able to keep track of where it is, even with noisy sensors and imprecise motor control. It should be able to calculate routes toward goals that might potentially involve moving around walls and other obstacles in the environment. Finally, any information the robot requires about its environment to implement these features it should be able to learn by itself as it explores the environment. The

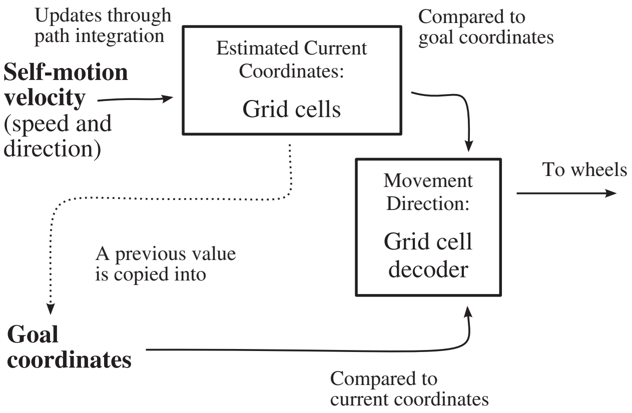

Updates through path integration
Self-motion velocity (speed and direction)
Estimated Current Coordinates:
Grid cells
Compared to goal coordinates
Movement Direction:
Grid cell decoder
To wheels
A previous value is copied into
Goal coordinates
Compared to current coordinates

Figure 2: The specific "sub-system" that will be considered in this paper, out of the broader, future system described in Figure 1. The neural network does not perform localization, mapping or route-planning, but navigates and performs path integration with the coordinate system provided by the simulated grid cells.

scope of this problem thus involves aspects of path integration, Simultaneous Localization and Mapping (SLAM) and map-based navigation.

To consider how to approach a neural implementation of such a navigational system, and also to see where the work presented in this paper fits into the larger picture, it is useful to break apart the full problem into a composition of smaller, interacting subcomponents. One possible decomposition of the mobile robot navigation problem as defined above, is shown in Figure 1. In the figure, path integration is shown as the process that updates the Estimated Current Coordinates component based on velocity inputs. To prevent errors due to noisy sensors and motors from accumulating in this path integration process, the position estimate is corrected whenever a previously visited location is recognized from the sensory inputs. This involves the sensory inputs triggering the recognition of a familiar location in the Localization Map component and thereafter a flow of this information back to the Estimated Current Coordinates component.

In order for the robot to be able to navigate to goal locations that are not necessarily reachable by moving in a straight line from the current location, the robot needs to employ some kind of route-planning process. This occurs in the Topological Map component, where information is available about how the discrete locations in the environment are connected to each other. From this information, the system might then calculate that in order to go from location Ⓐ to Ⓔ, the robot should follow as its route the sequence of known locations Ⓐ–Ⓑ–Ⓒ–Ⓓ–Ⓔ. Currently residing at location Ⓐ, the robot would then have as its immediate subgoal to first move to location Ⓑ, which we will call the robot's current *waypoint*. Whereas the final goal location might be blocked by a wall or some other obstacle, we assume that the waypoint location is one that can be reached in a straight line from the current location. Calculating the required direction of movement to reach the waypoint location is thus a matter of comparing the estimated coordinates for the current location with the coordinates associated with the waypoint location. This calculation happens in the Movement Direction component, the output of which is finally used to control the motors of the robot.

The maps used by the place recognition and the route-planning processes have to be learnt by the system through experience. This happens by associating sensory inputs and position estimates to discrete place representations in the localization map, and by learning associations *between* these place representations in the topological map. The primary data representations in this system are thus *coordinates* and *places*—the path integration process continually maintains an estimate of the current position as a set of coordinates, and these coordinates are combined with sensory inputs to form place representations in the localization and topological maps. By structuring the system in this way, it becomes clear that place cells can play the role of the discrete place representations in the map, while grid cells can be used to represent coordinates. This is consistent with the fact that grid cells and place cells interact also in the real brain.

The work presented in this paper has thus been concerned with the question of how to implement the coordinate system of the broader, future navigation system. Specifically, the "sub-system" considered in this paper is shown in Figure 2. A system of grid cells can collectively be understood to encode coordinates in 2D space (as described in Section 3.2), and by implementing our artificial grid cells with a specific model of grid cells known as continuous attractor networks, we will implicitly get support for path integration (as described in Section 3.3). This means that the Estimated Current Coordinates component of our navigation system can be fully realized by using these con-

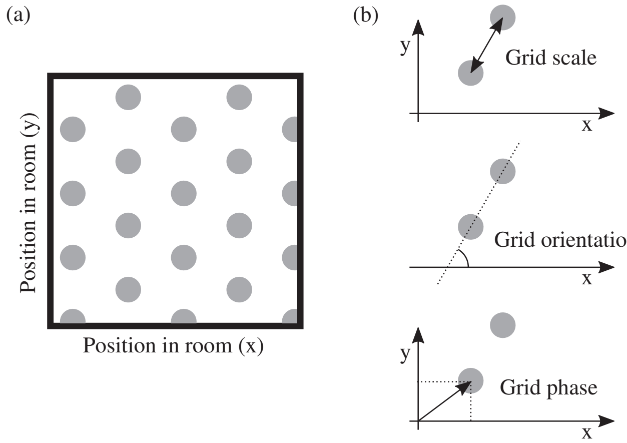


Figure 3: (a) Idealized illustration of the activity of an individual grid cell. The figure shows a top-down view of a square enclosure with a side length of, e.g., two meters. The shaded areas are the regions in this 2D space where the example grid cell will fire actively. (b) The spatial pattern of activity from a grid cell can be characterized by the three parameters of scale, orientation and phase.

tinuous attractor networks. The question that arises, however, is how two sets of grid cell-based coordinates then can be compared in order to calculate the required direction of movement between them. This is the computation to be performed by the Movement Direction—or grid cell decoder—component in Figure 2, and the main contribution of this paper is thus to demonstrate a mechanism to perform this operation when working with grid cell-based coordinates.

3.2 Grid Cells as a Coordinate System

In this section we will see how the activity patterns of grid cells, as observed in neurophysiological experiments, can be understood to implement a general coordinate system for two-dimensional space in the rodent brain.

The name "grid cell" comes from the activity patterns that these neurons make as the animal travels across space (Hafting et al, 2005). An idealized example is shown in Figure 3a. In contrast to place cells, grid cells are not active

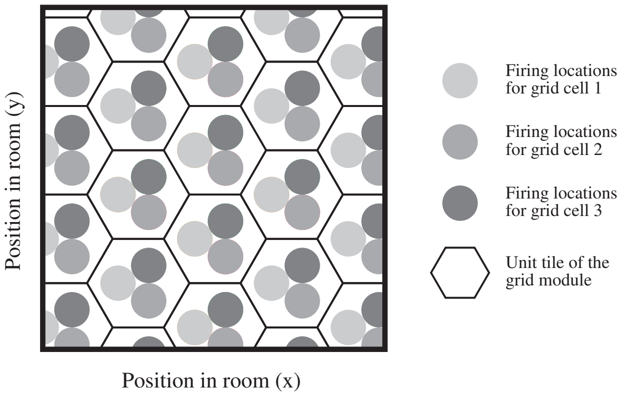

Position in room (y)
Position in room (x)
Firing locations for grid cell 1
Firing locations for grid cell 2
Firing locations for grid cell 3
Unit tile of the grid module

Figure 4: When multiple grid cells are recorded simultaneously, the neurons are found to cluster into *grid modules*. Within a module, the grid patterns have the same scale and orientation but different phases. The three shades of gray in the figure represent the grid patterns from three grid cells that belong to the same module. By assuming that the neurons in a grid module collectively cover all possible phases, it becomes possible to always determine the agent's position relative to the grid module's unit tile (shown as hexagonal tiles in this example). However, the activity within the module does not contain any information about *which* unit tile is the correct one.

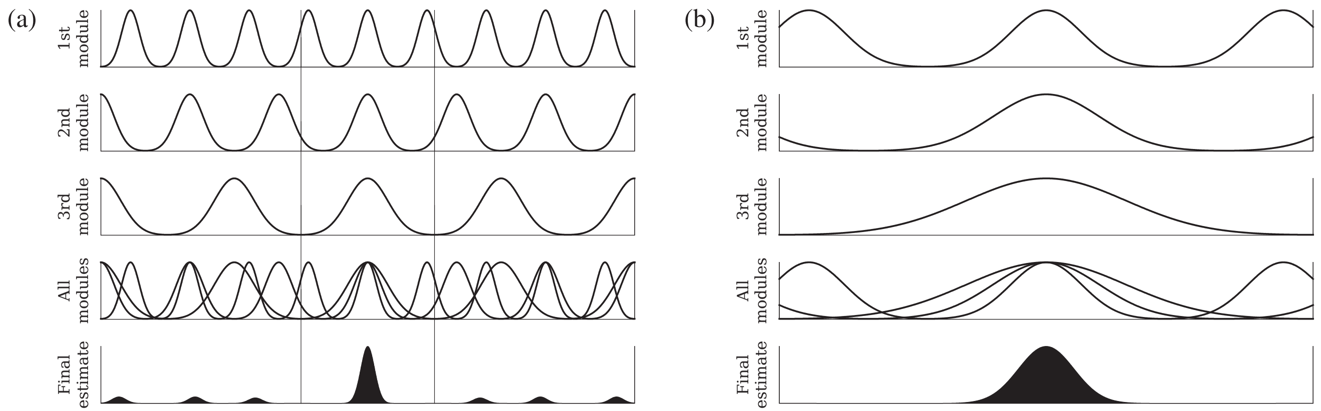


Figure 5: Illustrations of two possible views on how the grid cell system can represent unique locations despite repeating patterns of activity. In each subfigure, the first three rows show probability distributions of location given the neural activity in each of three grid modules. The distributions are sums of Gaussians separated by the period of the corresponding grid module. The bottom row is the final distribution over position given the activity from all three grid modules, calculated as the product of the three per-module distributions. These examples follow a similar approach as Wei et al (2015). (a) Residue number system-like decoding. The vertical bars demarcate one period of the largest grid module. Position is decoded over distances exceeding this range. (b) Hierarchical decoding. Position is decoded within the range of the largest grid module, i.e. only within the range marked with vertical bars in the previous subfigure.

only within single spatial fields in the environment, but have a periodic pattern of activity that repeats at the vertices of a triangular tiling of the plane. The result is a hexagonal grid pattern, extending indefinitely throughout space (Figure 3a). These spatial activity patterns can be characterized by the three properties of scale, orientation and phase—respectively the distance between two neighboring vertices of the grid pattern, the rotation of the grid pattern when compared against an axis of reference, and the translation of the grid pattern when compared against a point of reference (Figure 3b).

When multiple grid cells are recorded from at the same time, it becomes apparent that the individual grid cells do not operate in isolation. Rather, the neurons are found to cluster into *grid modules*—groups of grid cells that share the same scale and orientation of their individual firing patterns (Stensola et al, 2012). The only distinguishing property between neurons within the same grid module is thus that of their phase, i.e. the relative translations in space of their otherwise identical activity patterns (Figure 4).

Assuming that a sufficient number of grid cells participate in a given module, the module as a whole has the ability to encode a given set of 2D coordinates in a nearly continuous fashion—with one caveat. The limitation lies in the periodic nature of the grid cell pattern, in that the information carried by a grid cell module can only be interpreted relative to one specific hexagonal unit tile of the infinitely repeating pattern. Unless there is some additional information to indicate which particular unit tile is the correct one, the decoded position will thus be ambiguous.

A possible solution comes from the fact that the entorhinal cortex harbors multiple grid cell modules of different scales (Stensola et al, 2012). There appears to be a constant ratio between the grid scales of successively larger modules—in other words, the scale values of the grid modules form a geometric progression. The implication is that, by integrating information from modules of different scales, the ambiguity in single grid modules can be resolved.

However, there are multiple ways in which this resolution of ambiguity can be interpreted to occur. Two possible interpretations are depicted in Figure 5 (inspired by the approach in Wei et al, 2015). The two subfigures show a scenario where the position is read out from three grid modules, with a scale ratio between successive modules of 1.5. For simplicity, we consider a one-dimensional situation, so the horizontal axes in the figure represents e.g. position along a linear track. The actual location of the animal is in the center of the axes.

The first three rows show the probability distributions of position given the observed information from each of the modules. We assume that the position can be decoded to a Gaussian distribution within the unit tile of each module. However, because grid cells have a repeating pattern of activity, the final probability distribution used for each module is a sum of multiple instances of that Gaussian, separated by the period/scale of the grid module. We see that module 1 has the smallest grid scale, and thus the most number of peaks in its probability distribution, while module 3 has the largest grid scale and thus the fewest number of peaks. Assuming that the responses from the grid modules are independent from each other, the final probability distribution for position given the responses from all modules will be the product of the per-module distributions, as shown in the bottom rows of the figure.

In Figure 5a, because the periods of these three modules (1.5^0, 1.5^1 and 1.5^2) collectively repeat only after making an excursion of 9 units, we see that the correct position can be uniquely determined in the range -4.5 units to 4.5 units. In this interpretation of the grid cell system, the position can thus be decoded for excursions exceeding the scale of the largest grid module, due to the combinatorially long distance over which the activity across all of the modules remains unique (Fiete et al, 2008). This is related to how residue number systems represent numbers as their unique combinations of residues after performing different modulo operations on the original number (Fiete et al, 2008).

While this gives a large theoretical capacity of the grid cell system, it also requires precise readouts from each module (Wei et al, 2015). If there is too much uncertainty in the per-module readout, then the chance will increase of erroneously decoding the position to a location far away from the correct one. An alternative interpretation is shown in Figure 5b, where we have "zoomed in" the horizontal axis to match one period of the largest module. This illustrates an assumption that the largest module has a large enough scale to unambiguously encode the position over the behavioral range of the animal.

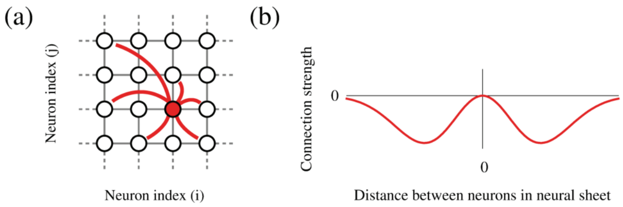


Figure 6: Connectivity in continuous attractor networks. (a) Neurons are assigned a row and a column in a two-dimensional "neural sheet". All neurons are recurrently connected to each other. (b) The strengths of the recurrent connections are determined by the connectivity profile, which is a function that maps distance of separation between two neurons in the neural sheet to a connection strength. This specific connectivity profile is based on Burak and Fiete (2009).

With this assumption, the grid cell system can be seen as gradually refining the position estimate from the largest grid module by adding information from the smaller-scaled modules (Wei et al, 2015; Stemmler et al, 2015). The smaller-scaled grid cell modules, while highly ambiguous, might represent space at a finer resolution than the larger-scaled ones. The activity of all modules taken collectively would then contain both the low-precision/long-range information of the larger-scaled modules as well as the high-precision/short-range information of the smaller-scaled modules. This interpretation of the grid cell system is the one used for the grid cell decoder in this paper.

3.3 Path Integration in Computational Models of Grid Cells

Continuous attractor network-based models are one of the major families of computational models that have been proposed for grid cells (Giocomo et al, 2011). The network dynamics in these models implicitly supports path integration, which makes them a compelling choice for the coordinate system in this project. In this section we will see the main principles behind these models.

In continuous attractor network-based models of grid cells, all of the neurons that belong to a grid module are conceptualized as being organized in a two-dimensional "neural sheet", such that each neuron is assigned a row and a column in this "matrix" of grid cells (Burak and Fiete (2009); Figure 6a).

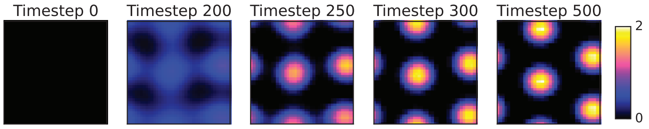


Figure 7: Spontaneous formation of a grid-like activity pattern in the neural sheet of an attractor-network grid cell module, due to random initial conditions and the recurrent connectivity.

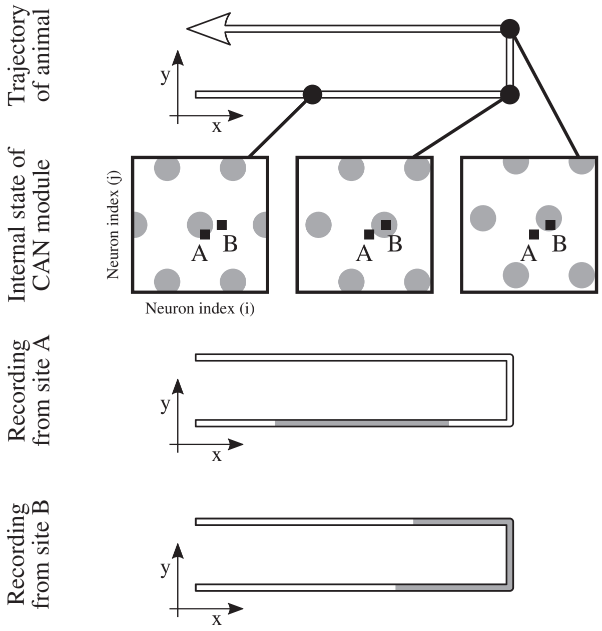


Figure 8: How grid cells arise from continuous attractor networks (CAN). Path integration, by shifting around the patterns of activity in the neural sheet, will cause the grid-like pattern in the neural sheet to also become visible through the spatial responses of individual neurons in the sheet. In this simplified example, an animal moves east, north and then back west again. As the animal makes these movements, the activity pattern in the neural sheet is shifted correspondingly, as illustrated for three different points along the animal's trajectory. If the activity level of an individual neuron is recorded and plotted as a function of the animal's position, as shown for two example neurons A and B in the bottom of the figure, then the hexagonal pattern in the neural sheet will eventually become visible across space. These neurons thus behave as grid cells.

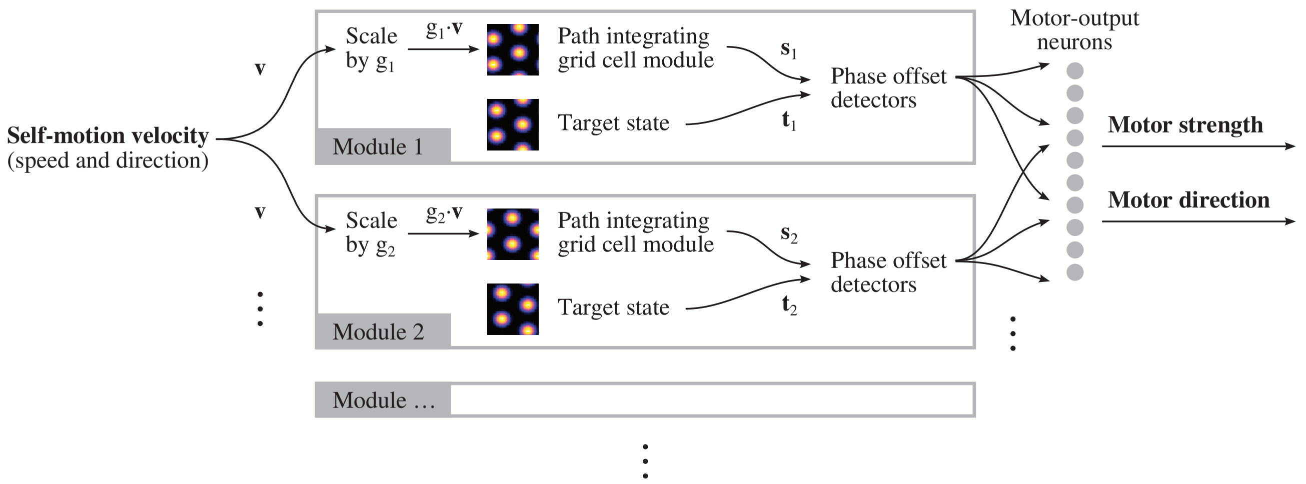


Figure 9: A schematic overview of the system. The system consists of a number of modules M, all of which receive as input the self-motion velocity of the agent ($\mathbf{v}$). Each module m scales this velocity input by a module-specific gain factor g_m, before performing path integration on this signal in a continuous attractor network ("path integrating grid cell module"). The activation vector $\mathbf{s}_m$, containing the outputs of the grid cells in module m, is compared to the target state $\mathbf{t}_m$ in a network of phase offset detectors. A common network of motor-output neurons receives inputs from phase offset detectors from all modules, in order to calculate the final motor signal.

This row/column combination will later correspond to the grid phase of the neuron, so proximity between neurons in this sheet implies that the neurons will have similar phases in their respective grid patterns.

The neurons are recurrently connected to each other, in an “all-to-all” fashion. The strengths of these connections are decided by the connectivity profile of the network, which is a function that relates the distance of separation between a given pair of neurons in the neural sheet, to the strength of the connection between them (Figure 6b). According to the connectivity profile in Figure 6b, neurons will be inhibited by activity in other neurons that are located in a certain range of radiuses away from the current neuron in the neural sheet. This particular wiring of the network will cause grid-like patterns of activity to spontaneously form in the network from random initial conditions (Figure 7).

These network patterns, which are stable attractor states of the network, can be made to shift around in the network in response to self-motion signals, which in effect is to perform path integration. Assuming that these shifts consistently reflect the actual movements of the agent, the hexagonal pattern of activity in the network will, over time, become visible in the spatial activity patterns of *individual* neurons in the network, as illustrated in Figure 8. This process is responsible for generating the grid cell-like behavior of each neuron in the attractor network.

Notice that in attractor-network models of grid cells, we thus have grid-like patterns both in the time-averaged spatial activity plots of individual neurons (e.g. Figure 3) and in the momentary network activity plots (e.g. Figure 7). This is an important distinction to be aware of.

4 Methods

This part of the paper describes the implemented system, including our mechanism for decoding the grid cell-signals into movement directions.

The main part of the system comprises a configurable number of *modules*, seen as the two rectangular blocks in the middle of Figure 9. Each module *m* consists of (a) a grid cell module, (b) a *target signal*, and (c) a network of *phase offset detectors*. The grid cell modules perform path integration on the incoming self-motion signal (composed of speed and direction), and output vectors of grid cell-activity $\mathbf{s}_m$ that are passed on to the corresponding networks of phase offset detectors. These phase offset detectors also receive a copy of the intended grid cell-activity vector $\mathbf{t}_m$ for the desired target location—the "target signal". The task of the phase offset detectors is to find the required direction of travel to make up for the offset in the grid patterns between the path integrator signal $\mathbf{s}_m$ and the target signal $\mathbf{t}_m$. The intended outputs of the model are a motor direction signal, giving the direction toward the target location, and a motor strength signal, indicating whether the agent has arrived at the target location or to keep going.

4.1 Multiple Modules with Different Spatial Scales

The model has multiple parallel modules in order to utilize information from a variety of grid cell modules representing space at different scales; this will provide the direction-finding process with long-range/low-precision signals as well as short-range/high-precision signals. The different grid scales are achieved by modulating the velocity inputs to each grid cell module. The velocity signal to module m is multiplied by the gain factor g_m before reaching the grid cell network. Smaller gain factors will cause the path integrator to respond more slowly to the same velocity inputs, thus causing the grid to appear larger, and vice versa.

The path integrator model used in these simulations was found to respond acceptably to velocity inputs at least in the range from 0.1 m/s to 1.2 m/s. As the actual speed of the simulated agent was fixed to 0.2 m/s, the range of acceptable gain factors could then be determined to be $[g_{min}, g_{max}] = [0.5, 6.0]$. The model uses a geometric progression from g_{min} to g_{max} for the gain factors. Given a specific number of modules M to be used, the g_m values can then be calculated as

$$R = \sqrt[M-1]{g_{max}/g_{min}}, \qquad g_m = g_{min} \cdot R^{m-1}. \tag{1}$$

4.2 Path Integrating Grid Cell Modules

The path integrator modules are closely based on the attractor-network grid cell model by Burak and Fiete (2009), and the following formulas are based on their presentation of the model. Each grid cell module consists of a 2D sheet of neurons of size $n \times n$, where $n = 40$. The activation values of these $n^2 = 40^2 = 1600$ neurons are contained in a vector $\mathbf{s}$, fully representing the current state of the path integrator.

Each grid cell i receives recurrent inputs from all other neurons in $\mathbf{s}$. Let $\mathbf{x}_i$ be the neural sheet coordinates of neuron i. The weight from afferent neuron i' onto neuron i can then be calculated from the connectivity profile $rec(d)$ by letting d be the shortest distance between $\mathbf{x}_{i'}$ and $\mathbf{x}_i$ in the neural sheet, taking into consideration that connectivity may wrap around the N/S and W/E edges. The recurrent connectivity profile $rec(d)$ is a difference of Gaussians, seen in Figure 6b or as the inhibitory “doughnut” in Figure 11, top left. Specifically,

$$rec(d) = e^{-\gamma d^2} - e^{-\beta d^2}, \tag{2}$$

where $\gamma = 1.05 \cdot \beta$, $\beta = \frac{3}{\lambda^2}$ and $\lambda = 15$. λ approximately specifies the periodicity of activity bumps in the neural sheet, i.e. the number of neurons from one bump of activity to the next. To express the update rule for grid cell i using vector notation, let $\mathbf{w}^{rec}_{\mathbf{c}}$ be the weight vector derived from the distance-to-weight-profile $rec(d)$ centered on the point $\mathbf{c}$ in the neural sheet. The update rule can then be described as

$$\tau \frac{ds_i}{dt} + s_i = f\left(\mathbf{s} \cdot \mathbf{w}^{rec}_{\mathbf{x}_i - \hat{\mathbf{e}}_{\theta_i}} + B_i\right), \tag{3}$$

solved for ds_i, where $dt = 10$ ms, $\tau = 100$ ms, $\mathbf{s}$ is the vector of the activation values at the end of the previous timestep, B_i is a velocity-dependent external input to the neuron and the activation function is $f(x) = max(0,x)$.

The center point $\mathbf{c}$ of the connectivity profile for efferent neuron i is here given as $\mathbf{x}_i - \hat{\mathbf{e}}_{\theta_i}$, i.e., there is an extra offset of $\hat{\mathbf{e}}_{\theta_i}$ in addition to $\mathbf{x}_i$ when positioning the connectivity profile for neuron i. The offset $\hat{\mathbf{e}}_{\theta_i}$ is the unit vector in the direction of θ_i, which in turn is the *directional preference* of neuron i. The directional preference is used to shift the activity pattern among the grid cells in response to asymmetric velocity inputs. Preferences for each of the four cardinal directions are distributed among the neurons in each 2×2 block of neurons. Namely, the x, y coordinates of a neuron are used to calculate an index $(2 \cdot (y \bmod 2) + x \bmod 2)$ into the list $[W, N, S, E]$ to determine θ_i. In the absence of velocity inputs, the four distinct preference-offsets counterbalance each other to keep the activity pattern at rest in the network. During motion, however, the external input B_i to each neuron becomes velocity-tuned according to the directional preference of the neuron. This input is calculated as

$$B_i = 1 + g_m \alpha \hat{\mathbf{e}}_{\theta_i} \cdot \mathbf{v}, \tag{4}$$

where $\mathbf{v}$ is the movement velocity and $\alpha = 0.10315$ is a scaling constant specified by Burak and Fiete (2009).

4.3 Phase Offset Detectors

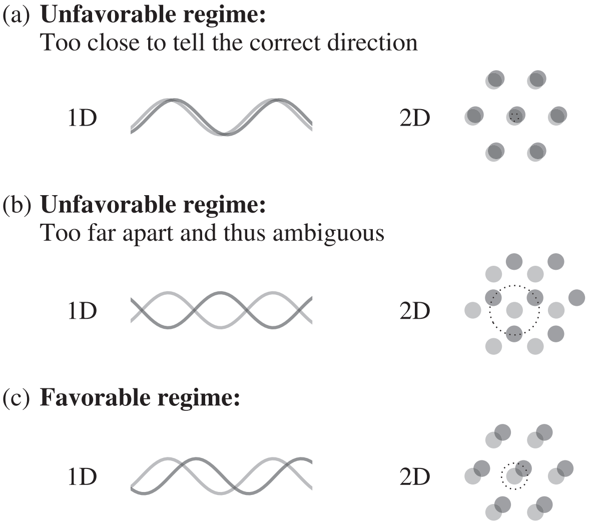
The vector of activation values $\mathbf{s}$ is passed on to a network of phase offset detectors. In addition to receiving the input vector $\mathbf{s}$ from the path integrating grid cell module, the phase offset detectors also receive a similarly-shaped vector $\mathbf{t}$ that represents the grid cell activity of the target location, i.e. a grid cell-encoding of the desired target coordinates.

Each phase offset detector j has an associated origin location $\mathbf{x}_j$ and a preference direction θ_j. The neuron is tuned to respond when an activity bump is near the origin location $\mathbf{x}_j$ in the neural sheet of the path integrator grid cell module ($\mathbf{s}$) and there *simultaneously* is an activity bump near the location $\mathbf{z}_j = \mathbf{x}_j + \delta \hat{\mathbf{e}}_{\theta_j}$ in the grid cell-encoded target-location-input ($\mathbf{t}$), δ being a fixed offset length. Specifically, the activation of phase offset detector j is calculated as

$$p_j = f\left(\mathbf{s} \cdot \mathbf{w}^{in}_{\mathbf{x}_j} + \mathbf{t} \cdot \mathbf{w}^{ex}_{\mathbf{z}_j}\right), \tag{5}$$

where $\mathbf{w}$ again refers to weight vectors derived from given connectivity profiles centered on given points in the neural sheet, but with new connectivity profiles *in* and *ex*. The path integrator inputs $\mathbf{s}$ are fully connected using the connectivity profile $in(d)$ centered at $\mathbf{x}_j$, while the target location inputs $\mathbf{t}$ are fully connected using the connectivity profile $ex(d)$ centered at $\mathbf{z}_j$. These connectivity profiles are defined as

$$in(d) = \eta \cdot \left(e^{-\beta d^2} - 1\right), \qquad ex(d) = e^{-\beta d^2}, \tag{6}$$



(a) **Unfavorable regime:**
Too close to tell the correct direction

(b) **Unfavorable regime:**
Too far apart and thus ambiguous

(c) **Favorable regime:**

Figure 10: Concept behind the phase offset detectors. (a) If the current pattern in the neural sheet, **s**, is similar/close to the target pattern, **t**, it is difficult to tell the correct direction of movement. (b) If the current pattern **s** is shifted too far apart from the target pattern **t**, then the ambiguity of the repeating pattern makes it impossible to tell the correct direction. (c) If the shift between **s** and **t** is somewhere between the regimes of (a) and (b), then it might be possible to extract the correct direction.

where $\eta = 0.25$. The offset length δ is set to be $\delta = 7$, in the neighborhood of half of λ.

The effect is to respond strongest when there is an offset of length δ in direction θ_j between the activity patterns in **s** and **t**, given that the path integrator currently has activity in the vicinity of $\mathbf{x}_j$. The concept behind this mechanism is illustrated in Figure 10—given that the phase offset detectors are configured to respond to offsets in a "favorable regime", the neurons may be able to extract a direction of movement from the grid module. An example situation is shown in Figure 11, where a phase offset detector with $\mathbf{x}_j = (20,20)$ and $\theta_j = 45°$ receives inputs of favorable characteristics from **s** and **t**.

In order for the network of phase offset detectors to work independently of the current location of network activity in the path integrator, there needs to be a sufficient number of phase offset detectors that sample different origin

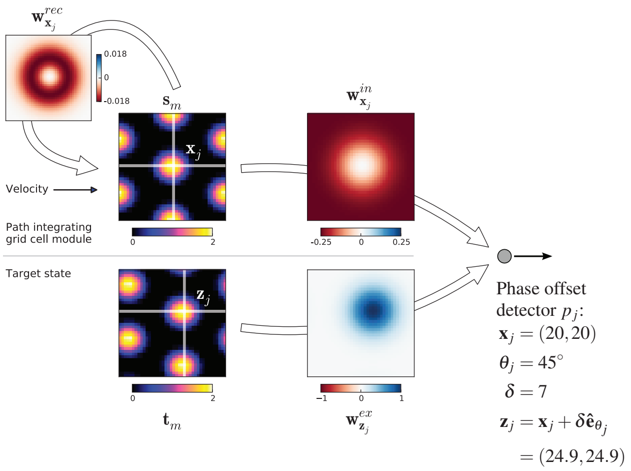


Figure 11: Example of a phase offset detector, p_j, showing the input networks $\mathbf{s}$ and $\mathbf{t}$ and the connectivity profiles with which these two networks are connected to p_j. Depicted matrices are 40×40.

locations $\mathbf{x}_j$. Additionally, the network needs to sample a range of different preference directions θ_j. This is realized using two parameters S_θ and S_{xy} that respectively specify the number of directions sampled in the interval $[0, 2\pi)$ and the number of steps to use along each of the two dimensions of the neural sheet when sampling origin locations. The total number of phase offset detectors will then be $S_\theta \cdot S_{xy}^2$ per module.

4.4 Motor-Output Neurons

The activity from the phase offset detectors are aggregated in a set of *motor-output neurons*. Whereas the grid modules and phase offset detectors are instantiated separately for each module, the motor-output neurons comprise a common network receiving inputs from all of the modules. The number of motor-output neurons is the same as the number of sampled preference directions S_θ in the phase offset detector networks. The motor-output neurons sample the same directions as the phase offset detectors.

The activity in each motor-output neuron is essentially the sum of the activity in all of the phase offset detectors that share preference direction with the motor neuron. An important detail, however, is that these contributions are weighted by the inverse of the gain factors of their respective modules. In other words,

$$u_k = \sum_m \left[g_m^{-1} \sum_j^{\theta_j = \theta_k} p_{m,j} \right], \tag{7}$$

where u_k is the activity of motor-output neuron k and θ_k is the preference direction of k. This weighting will give priority to the direction signals from the modules with low gain factors g_m, i.e. the modules where the quality of the path integration information is long-range-applicable but with low precision. As the agent gets closer to the target location, the intention is for these signals to fade off to sufficiently weak strengths so that the shorter-range, higher-precision signals will pick up in motor influence. The purpose is to achieve the combination of a long-range *and* high-precision signal.

To calculate the final motor-output signal $\mathbf{\Theta}$, the values of u_k are considered as vector contributions in the direction of θ_k, i.e. the vectors $u_k \cdot \hat{\mathbf{e}}_{\theta_k}$ are summed together, and this sum is then scaled to compensate for the variable number of inputs and their weighting. The final calculation is thus

$$\mathbf{\Theta} = \rho \cdot \sum_k u_k \cdot \hat{\mathbf{e}}_{\theta_k}, \quad \rho = \frac{1}{S_\theta \cdot S_{xy}^2 \cdot \sum_m g_m^{-1}}, \tag{8}$$

whereafter the angle of $\mathbf{\Theta}$ makes up the motor direction signal and the vector length becomes the motor strength signal.

5 Experimental Setup

5.1 Trials

Each experiment trial consists of a succession of stages, specifically (a) pattern formation in grid cell modules, (b) capture of path integrator states into target states, (c) the agent performing a random walk for T seconds, and (d) the agent attempting to return "home" to the target location.

At the beginning of the simulation, in order for the grid cell networks to form grid-like activity patterns, all s_i values are initialized randomly in the range $[0, 10^{-4})$ before the networks are then allowed to settle for 1000 timesteps (Figure 7). When this pattern formation process is done, the grid-like activity patterns will have been initialized to essentially random starting-coordinates. The system now copies these activity patterns $\mathbf{s}_m$ into the target state vectors $\mathbf{t}_m$. The m different target state vectors $\mathbf{t}_m$ henceforth remain unchanged for the rest of the trial, as a memory of the coordinates of the starting location ("home").

The agent then performs a random walk for a configurable duration of time T seconds. The time duration for a single iteration of the model has been set to be 10 ms, so there are 100 timesteps/s. During both the random-walk and the return-home stages, the agent moves with a constant speed of 0.2 m/s with only the movement direction changing. The random walk starts with a uniformly distributed random value from $[0, 2\pi)$ as the movement direction. At every timestep it is updated by adding a radian value from a normal distribution with $\mu = 0$, $\sigma = 1$.

After T seconds have elapsed, the return-home stage begins. The motor-direction output from the network is used to set the movement direction of the agent, whereas the motor-strength output is used as a termination criterion for determining when to end the trial. Three different termination criteria are used; (a) the motor-strength signal is less than 10^{-6}, (b) the return-home stage has lasted for at least a second and the straight-line distance to the point traversed one second ago is less than 0.01 m, or (c) the return-home stage has lasted $2 \cdot T$ seconds. Whichever termination criteria ends the trial, the straight-line distance to the starting location from the final stopping location is deemed the *error* of the trial. The favorable outcome is a low overall error value.

5.2 Parameter Search

A parameter search was conducted to find good values for M, S_θ and S_{xy} to use for the rest of the experiments. An exhaustive test was performed on all combinations of values in the intervals $M \in [2, 6]$, $S_\theta \in [4, 32]$, $S_{xy} \in [5, 40]$. However, to penalize expensive solutions and to place an upper bound on the complexity of the solutions to be tested, a *synapse cost* C was calculated for each parameter combination. This value provides an estimate of the number of synapses in the model and consequently a rough estimate of the number of floating-point operations required to update the model (without optimization). C was calculated as $C = M \cdot \left(n^2\left(n^2+1\right) + 2n^2 \cdot S_\theta \cdot S_{xy}^2 + S_\theta \cdot S_{xy}^2\right)$, with the three terms representing the synapse cost to operate respectively a grid cell module, the phase offset detectors, and the axons to the motor-output neurons. Only the combinations with $C < 10^8$ were tested, leaving 2685 combinations to test. For each combination, 100 trials with a random-walk duration of $T = 30$ s were performed and the mean error was reported (Figure 12a). The parameter combination $M = 4$, $S_\theta = 28$, $S_{xy} = 9$ was selected for further use (highlighted).

To get a sense for how the individual parameters affect the outcome, new sets of runs were performed where each parameter in turn was changed within the defined intervals and evaluated over 100 trials, while the two other parameters were left unchanged (Figure 12b). M and S_{xy} seem to affect the results little above thresholds of respectively $M = 2$ and $S_{xy} = 8$. S_θ, on the other hand, appears to be more sensitive to the particular value to which it is assigned.

5.3 Implementation Details

All random values used by the implementation in this paper were generated using the Mersenne Twister pseudo-random number generator included with the C++11 standard library. The model and the simulator used single-precision floating-point values throughout.

6 Results

6.1 Direction-Finding Ability

Two different examples of how the system operates in practice are presented in Figures 13 and 14. In the first example, the direction-finding ability of the model is tested at multiple points along a circle centered on the goal location. For each of the 18 uniformly spaced directions tested, the agent was driven a distance of 0.5 m in the opposite direction of the intended “goal direction” and

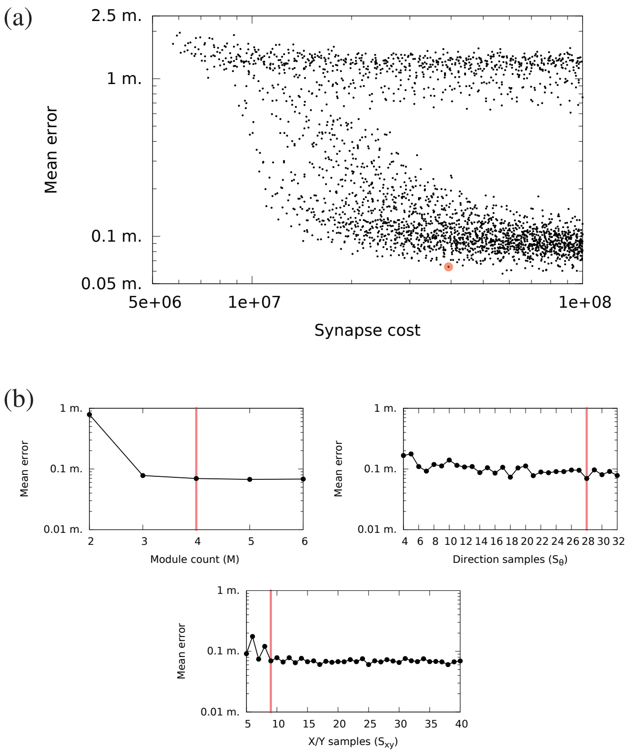


Figure 12: (a) Scatter plot of parameter combinations tested during the parameter search, with each dot showing the mean error over 100 runs for a given parameter combination, plotted against the respective synapse cost. (b) For each of the three parameters M, S_θ and S_{xy}, the effect of modifying that parameter from the chosen parameter combination ($M = 4$, $S_\theta = 28$, $S_{xy} = 9$; indicated by vertical lines) while leaving the other parameters unchanged. Mean error over 100 runs. The combination $M = 4$, $S_\theta = 28$, $S_{xy} = 9$ is represented by the same set of trials in all three figures.

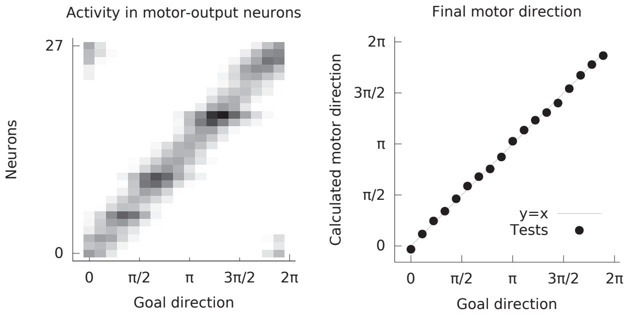


Figure 13: Example of network operation at locations a fixed radius $r = 0.5$ m away from the home location in given directions. Left: Momentary activity in the $S_\theta = 28$ motor-output neurons. Right: Final motor direction calculated from the motor-output neurons, plotted at the closest 2π period to the goal direction.

allowed to settle for 250 timesteps before the motor outputs were examined. For each trial, the figure shows the recorded activity from all S_θ motor-output neurons as well as the motor-direction signal from the model. As evidenced by the figure, the model is able to accurately calculate the goal direction at this specific distance of $r = 0.5$ m.

Figure 14 demonstrates a full trial with both random-walk and return-home stages as described above. After a $T = 30$ s random walk, the agent successfully attempts a return to the home location. The figure includes the momentary activity of all of the grid cell modules ($\mathbf{s}_m$) both at the beginning and the end of the return-home stage. In each of these cases, the motor-neuron activity is also shown. The plots of $\mathbf{s}_m$ show possible interpretations of how the activity patterns might have shifted from the target state $\mathbf{t}_m$, which was also the initial state of the grid modules at $T = 0$ s. From the leftmost to the rightmost columns, the grid modules progress from long-range/low-precision to short-range/high-precision. The first, second and third modules show a correct assessment of the goal direction at $T = 30$ s, whereas the fourth module is “out of range” and in this case has an ambiguous response.

At $T = 37.1$ s, we see that the grid modules have aligned closely with the corresponding target states. The trial thus terminated because of the weak motor-strength signal, bringing the agent to a halt at a distance of 4.74 cm from the goal location, from an initial goal distance of 1.47 m at the end of

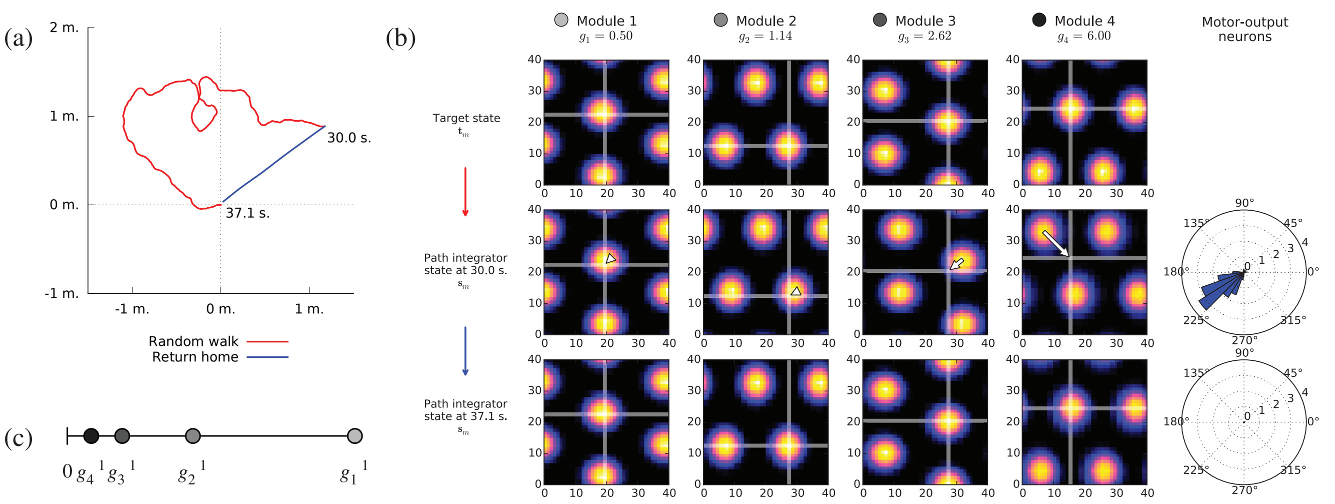


Figure 14: Example of network operation during various stages of a trial. (a) Trace of the trajectory followed during random-walk and return-home stages. (b) Top row: Target state. Rest: Momentary activity of the grid cell modules and motor-output neurons at various points in time. (The s_i and t_i values were arranged according to $\mathbf{x}_i$ and convolved with a uniform 2×2 filter to hide artifacts due to different preference-offsets. This procedure was used for all $\mathbf{s}$ and $\mathbf{t}$ visualizations in this paper.) (c) Plot of the inverses of the gain factors g_m used when the model is configured for four modules ($M = 4$), in order to illustrate the difference in the ranges of the modules.

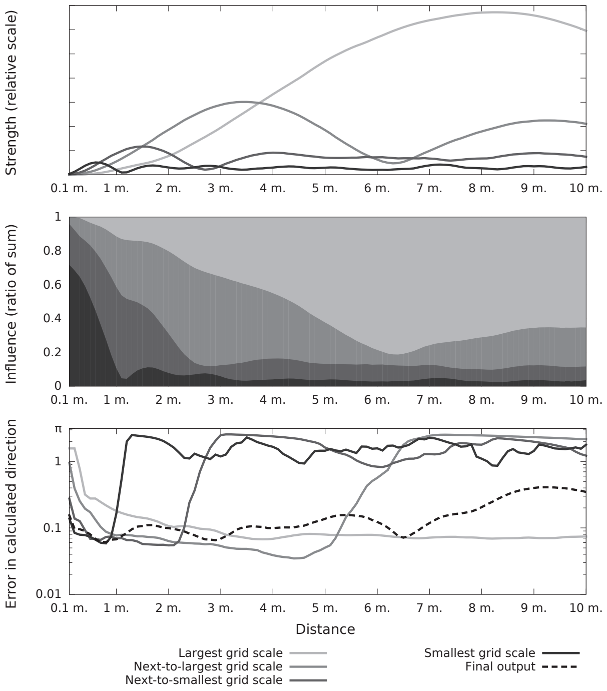
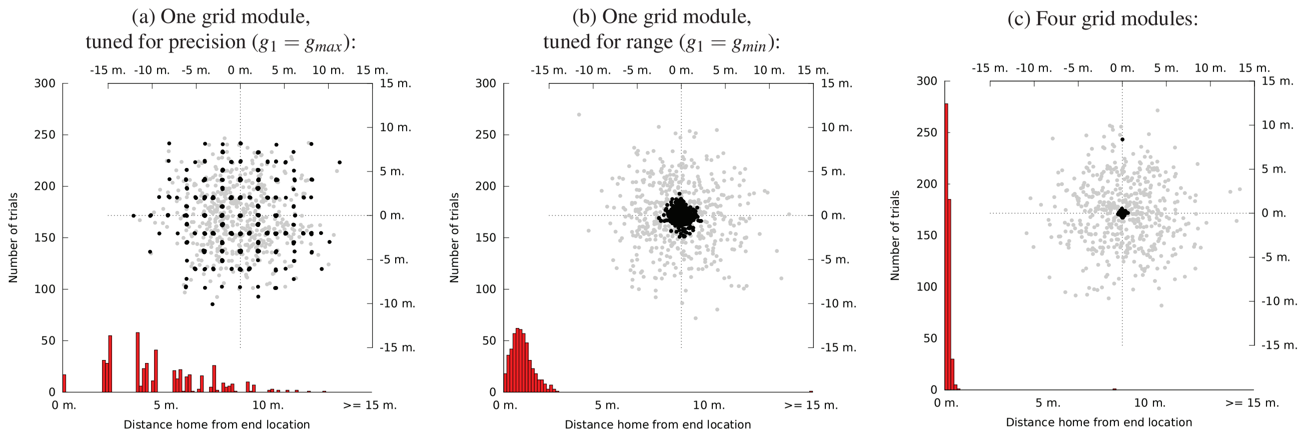


Figure 15: Behavior of the $M = 4$ different modules at increasing distances from the goal. The model was tested as in Figure 13, but r was varied at 0.1 m increments in the interval $[0.1\text{ m}, 10\text{ m}]$. For each tested radius the motor-output strengths and direction-signal errors are reported as the mean over 18 tested directions. In order to report values individually for each module, extra motor-output networks were instantiated such that each only received phase offset detector-inputs from one given module. For these plots, $\rho = 1$ in order not to cancel out the scaling differences.



(a) One grid module, tuned for precision ($g_1 = g_{max}$):

(b) One grid module, tuned for range ($g_1 = g_{min}$):

(c) Four grid modules:

Figure 16: End locations after returning home, for various configurations of grid cell modules. 500 runs for each configuration. Gray dots show end of random walk and black dots show end of return home. Histograms below show the distribution of goal distances at trial termination, with a bin size of 15 cm. (The upper-left run in part b stopped at a distance of 29.7 m from the target.)

the random walk.

6.2 Effect of Multiple Grid Modules

The effect of using multiple grid modules is further demonstrated in Figure 15, which shows, as a function of the distance to the goal, the strengths and errors of each module's contribution to the motor-output network when seen in isolation. To illustrate their relative influences, the signal strength of each module is also shown in terms of its ratio of the sum. Lastly, the final motor-direction error is shown overlaid on the direction-error plots from the individual modules.

For each module, there is a distinctive bell-shape in the strength curve as the tested radius approaches and recedes from the "optimal detection distance" of the module's offset detectors. The vicinity of the peak of the bell curve is also where the module's direction-error is at a minimum. Past this region, the module abruptly becomes unreliable due to the periodicity of the grid cell signal. Because of the gain-based weighting of module contributions, however, one of the larger-scaled modules is able to overpower the contributions of the smaller-scaled modules and thus ensure that the final direction signal is still valid. As seen by the dashed line in the lower diagram, the final direction-output achieves a combination of range and precision not seen in any of the individual modules.

Figure 16 demonstrates the importance of this combination of precision and range information. The figure contains results from three different sets of 500 trials, each with $T = 180$ s. Whereas the rightmost diagram shows the results from trials with the default parameters ($M = 4$, $S_\theta = 28$, $S_{xy} = 9$), the two other diagrams only use one module ($M = 1$). The leftmost diagram has the gain factor set to $g_1 = g_{max}$, for rapid periodicity and short-range/high-precision signals, while the middle diagram has $g_1 = g_{min}$, i.e. tuned for long-range/low-precision signals.

The distributions of termination locations seen in the scatterplots confirm our expectations from the known qualities of the grid module signals. With one module tuned for precision (Figure 16a), the agent either precisely returns home to the target location or it ends up in an attractor location that is part of

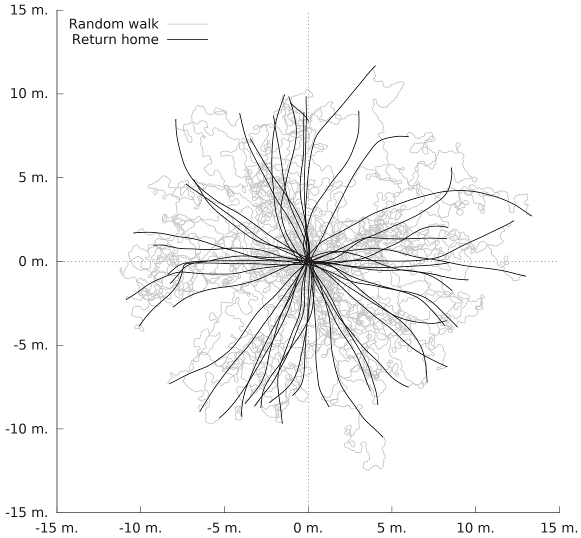


Figure 17: Traces of the 50 runs from Figure 16c with the farthest distance from the goal at the end of the random-walk stage.

a repeating pattern of possible attractors. This shows the periodic nature of the grid cell encoding of space. With one module tuned for range (Figure 16b), all but one of the 500 trials terminate in a cluster centered on the target location. However, the improved range has carried a penalty of worse precision. This penalty is seen to be mostly alleviated by integrating information from multiple grid modules; in Figure 16c, where four grid modules are used, all but six of the 500 trials end up within 0.5 m of the target, with only one ending up more than 1 m away.

To get a sense for the trajectories the model follows during these attempts to reach the target, Figure 17 contains traces of the 50 runs from Figure 16c with the farthest goal distance at the end of the random walk. With some exceptions, the paths taken are all largely straight lines toward the target. All but one trajectory (seen near the top) end up within 0.5 m of the goal.

7 Discussion

The basis for this project was to use neural representations of space for direction-finding in a robot. The paper has presented a model that integrates an existing model of path integrating grid cells with a novel mechanism that is able to use the grid cell representation to direct the agent to a remembered goal.

We consider the model at its current abstraction level to be biologically plausible. The inputs and outputs of the model are geocentric direction and speed signals, which is supported by the existence of head-direction cells. Attractor-network models are considered viable candidates for understanding the operation of grid cells, and the phase offset detectors and motor-output neurons are simple input-summing neurons. The target state signals are assumed to be a grid cell-encoding of the target coordinates; this could be provided in the form of backprojections from the hippocampus.

Chadwick et al (2015) used VR-supported fMRI to look for a goal-direction signal in the human brain. They found that there would be similar brain activity patterns in the entorhinal/subicular region when a given geocentric direction was used as either the current facing direction or the goal direction, and the activity patterns were found to be best accounted for as a mixture of the encodings of the facing direction and the goal direction. The authors see these results as evidence that some form of goal-directed simulation of spatial representations is involved in navigation, citing the model by Erdem and Hasselmo (2012) as an example of how this mechanism could work.

This is not the only possible conclusion of these results, especially since the model by Erdem and Hasselmo requires simulation of look-ahead trajectories in many different directions from the current location in order to discover the goal location. A mechanism similar to the one presented in this paper would allow the goal direction to be calculated directly from grid cell representations of the current location and the goal, avoiding the need for extensive simulations in multiple directions. The mixture representations reported by Chadwick et al could still conceivably be accounted for by oscillations in the entorhinal/subicular region between encodings of the present and the future spatial states. Experiments at finer spatial and temporal resolutions would hopefully be able to distinguish the extents of these two types of contributions.

The model presented in this paper resembles the "rate-coded vector cell" model by Bush et al (2015), in that both models detect salient differences in grid phase between the grid cell representations of the current location and the goal location. However, whereas Bush et al support displacements that can exceed the scale of the largest grid module, and therefore need to simultaneously consider information from all grid modules when decoding phase differences, in this paper we only support displacements within the scale of the largest grid module, and the phase decoding can therefore take place independently within each module. Our system operates based on the results of a path integration process incorporated "in the loop" together with the decoding mechanism, while Bush et al (2015) consider the decoding problem in isolation. Differences such as these make a direct comparison of model performance difficult. For a comprehensive comparison of decoding mechanisms, one would need to take into consideration the origin of the grid cell signal, any path integration-related requirements such as error correction (Sreenivasan and Fiete, 2011), the complexity of the decoder network in terms of neurons, synapses and temporal demands, the supported range of distances, etc. This remains as future work.

The successful simulation results show promise for implementing the system in a physical robot in the future. The translation into the physical world will bring with it its own set of challenges, such as noisy self-motion inputs and imprecise motor control. The integration of sensory information into the model is thus one important area for further study, as has been done in other grid cell-based robot controllers (Milford and Wyeth, 2010). As described for the proposed architecture of the future system in Section 3.1, the system could accommodate sensory inputs through a network of place cells that interacts with the grid cell system. A place cell-based mechanism could also play a role in enabling the robot to navigate in more complex environments, such as with walls and other obstacles.

Acknowledgements

The author is grateful to Keith Downing and Trygve Solstad for helpful discussions, feedback and advice.

References

Burak Y, Fiete IR (2009) Accurate path integration in continuous attractor network models of grid cells. PLoS computational biology 5(2):e1000291

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

Chadwick MJ, Jolly AE, Amos DP, Hassabis D, Spiers HJ (2015) A goal direction signal in the human entorhinal/subicular region. Current Biology 25(1):87–92

Edvardsen V (2015) A passive mechanism for goal-directed navigation using grid cells. In: Proceedings of the European Conference on Artificial Life 2015, pp 191–198

Erdem UM, Hasselmo M (2012) A goal-directed spatial navigation model using forward trajectory planning based on grid cells. European Journal of Neuroscience 35(6):916–931

Erdem UM, Hasselmo ME (2014) A biologically inspired hierarchical goal directed navigation model. Journal of Physiology-Paris 108(1):28–37

Erdem UM, Milford MJ, Hasselmo ME (2015) A hierarchical model of goal directed navigation selects trajectories in a visual environment. Neurobiology of learning and memory 117:109–121

Fiete IR, Burak Y, Brookings T (2008) What grid cells convey about rat location. The Journal of Neuroscience 28(27):6858–6871

Giocomo LM, Moser MB, Moser EI (2011) Computational models of grid cells. Neuron 71(4):589–603

Giovannangeli C, Gaussier P (2008) Autonomous vision-based navigation: Goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 676–683

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

Huhn Z, Somogyvári Z, Kiss T, Érdi P (2009a) Distance coding strategies based on the entorhinal grid cell system. Neural Networks 22(5):536–543

Huhn Z, Somogyvári Z, Kiss T, Érdi P (2009b) Extraction of distance information from the activity of entorhinal grid cells: a model study. In: 2009 International Joint Conference on Neural Networks, pp 1298–1303

Kubie JL, Fenton AA (2012) Linear look-ahead in conjunctive cells: an entorhinal mechanism for vector-based navigation. Frontiers in neural circuits 6(20)

Masson C, Girard B (2011) Decoding the grid cells for metric navigation using the residue numeral system. In: Advances in Cognitive Neurodynamics (II), Springer, pp 459–464

McNaughton BL, Battaglia FP, Jensen O, Moser EI, Moser MB (2006) Path integration and the neural basis of the 'cognitive map'. Nature Reviews Neuroscience 7(8):663–678

Milford M, Schulz R (2014) Principles of goal-directed spatial robot navigation in biomimetic models. Philosophical Transactions of the Royal Society B: Biological Sciences 369(1655):20130484

Milford M, Wyeth G (2010) Persistent navigation and mapping using a biologically inspired SLAM system. The International Journal of Robotics Research 29(9):1131–1153

O'Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain research 34(1):171–175

Solstad T, Boccara CN, Kropff E, Moser MB, Moser EI (2008) Representation of geometric borders in the entorhinal cortex. Science 322(5909):1865–1868

Spiers HJ, Barry C (2015) Neural systems supporting navigation. Current Opinion in Behavioral Sciences 1:47–55

Sreenivasan S, Fiete I (2011) Grid cells generate an analog error-correcting code for singularly precise neural computation. Nature neuroscience 14(10):1330–1337

Stemmler M, Mathis A, Herz AV (2015) Decoding the population activity of grid cells for spatial localization and goal-directed navigation. bioRxiv 021204

Stensola H, Stensola T, Solstad T, Frøland K, Moser MB, Moser EI (2012) The entorhinal grid map is discretized. Nature 492(7427):72–78

Sun H, Yao Tr (1994) A neural-like network approach to residue-to-decimal conversion. In: Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on, IEEE, vol 6, pp 3883–3887

Taube JS, Muller RU, Ranck JB (1990) Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. The Journal of Neuroscience 10(2):420–435

Wei XX, Prentice J, Balasubramanian V (2015) A principle of economy predicts the functional architecture of grid cells. eLife 4:e08362

Paper C

Long-range navigation by path integration and decoding of grid cells in a neural network

(Edvardsen, 2017)

Author(s):

Vegard Edvardsen

Published at conference:

2017 International Joint Conference on Neural Networks (IJCNN)

Paper C

Edvardsen (2017)

Long-Range Navigation by Path Integration and Decoding of Grid Cells in a Neural Network

Vegard Edvardsen

Department of Computer Science,
NTNU – Norwegian University of Science and Technology, Trondheim, Norway

Abstract

Neural modelers in the domain of robot navigation, e.g. within the fields of neurorobotics and neuromorphic engineering, can benefit from a wealth of inspiration from neuroscientific research in the hippocampal formation—cell types such as place cells and grid cells provide a window into the inner workings of high-level cognitive processing, and have spawned many interesting computational models. Grid cells are thought to participate in path integration and to implement a general coordinate system, both of which are useful features in a neural navigation model. Continuous attractor networks are a computational model that can embody both aspects of grid cells, and in previous work we showed that a neural network can successfully decode the outputs of such networks in order to implement vector navigation. That work assumes that the grid cell system represents long distances by employing a geometric progression in its spatial scaling of successive submodules, in such a way that “nested” grid cell decoding can be performed. For long-range navigation this requires that the continuous attractor networks can implement sufficiently long geometric progressions of grid scales, but this turns out to trigger the issue of “pinning”. In this paper we demonstrate conditions under which pinning occurs as well as its consequences for the grid cell-based navigation model. We propose and assess several candidate solutions to the problem, in particular based on differential adjustment of neurons’ update rates in the model. We finally demonstrate that the system is able to perform long-range navigation using our chosen solution.

1 Introduction

Neural networks have in recent years had a renaissance as a tool for building artificially intelligent computer systems [1]. Improved hardware, datasets and techniques for construction and training of deep neural networks have yielded systems that have advanced the state-of-the-art in areas of such wide variety as image recognition [2] and beating human players in the board game Go [3]. Neural networks for such tasks are often feed-forward in architecture, especially in sensory processing applications [4]. However, the need for a short-term memory capacity has also been emphasized, particularly for problems demanding higher-level cognitive processing [5].

One domain of cognitive tasks that usually requires a short-term memory is that of navigation—keeping track of where you come from and where you are going; planning how to get to a goal location and thereafter back home. In earlier work [6] we showed how an artificial neural network inspired by the brain's grid cell system can keep track of an agent's current 2D coordinates and use this information to guide the agent to a goal location. In this paper we will build upon that work to improve our grid cell-based neural navigation system to work over longer distances.

Section 2 presents pertinent background material on grid cells and a computational model for them. In Section 3 we demonstrate the range of our navigation system when using a single module of grid cells. Section 4 discusses ways in which the range of the grid cell system can be extended by introducing multiple modules. The navigation system is then tested with increasing numbers of modules and is shown to experience a shortfall in range improvement after a certain number of modules have been added. Section 5 demonstrates the issue of "pinning" that is responsible for this shortfall. In Section 6 we propose and assess several candidate solutions to the pinning problem, with our chosen solution evaluated in Section 7. Section 8 concludes the paper.

2 Background

2.1 Spatial neurons in the brain

Neuroscientific studies of navigation are often concentrated on the mammalian brain region around the hippocampus, primarily in rats and other rodents. This brain region is believed to implement a *cognitive map*—a neural implementation of high-level cognitive information about the spatial environment. Cell types in this region, such as place cells [7], head-direction cells [8], border cells [9], and grid cells [10], provide a wealth of inspiration for modelers of artificial neural navigation systems. In the context of systems that benefit from a short-term memory capacity, grid cells are particularly interesting. Like place cells and border cells, grid cells are neurons that activate depending on the animal's location in space. However, whereas place cells usually activate at only a particular place in an environment and border cells activate only along particular borders, a given grid cell is active whenever the animal is located at the vertices of an imaginary hexagonal grid extending throughout the 2D plane (Fig. 1). This relationship between the cell's activity and the animal's location in 2D space persists even in complete darkness, indicating that grid cells reflect an internally maintained neural activity and that this activity can be generically updated by self-motion inputs. This suggests that grid cells participate in a path integration process in the brain, continuously adding the animal's current velocity vector to an internal variable representing a vector of the total displacement from a point of reference (Fig. 3).

2.2 Continuous Attractor Networks (CANs)

"Continuous Attractor Networks" (CANs) are recurrent neural networks that are wired in a particular way so that the energy landscape of the network contains a continuum of stable network states of a particular dimensionality [11]. One-dimensional CANs are for example used as a model for head-direction cells—the stable states of the network each represent a particular head-direction, i.e. the stable states fall along a 1D line, and inputs signaling head-turns cause the network to shift to new network states that reflect the updated head-direction. These CANs thus perform path integration on the 1D

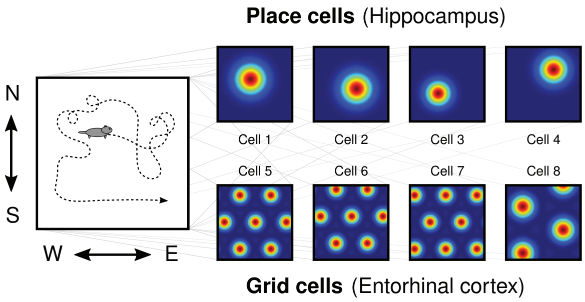


Figure 1: Illustration of two types of spatial neurons found in the hippocampal formation of rodents such as rats—place cells and grid cells. Left: Neurons are recorded as the animal explores an enclosure. The animal's position in the horizontal plane is recorded simultaneously. Right: By plotting heatmaps for each neuron—showing the neuron's average activation in each visited position bin—the characteristic spatial responses of place cells and grid cells are revealed. Place cells respond typically in one or a few areas of the enclosure, whereas grid cells respond at the vertices of an infinite hexagonal grid pattern.

head-direction variable—by extending a CAN to two dimensions, it will be able to use inputs representing velocity in the 2D plane to update a network state representing the current 2D total displacement (Fig. 2). It is possible to create 2D CANs whose individual neurons produce the repeating, hexagonal grid patterns across space that are characteristic of grid cells, and CANs have indeed emerged as one of the major classes of computational models for grid cells [12].

2.3 Vector navigation with grid cells

The robot navigation system RatSLAM [13] is an example of the potency of looking to neuroscientific findings in the hippocampus for neural principles to use in an artificial navigation system. The core of this algorithm is a CAN that performs path integration and participates in place recognition. A CAN such as used by RatSLAM will, within a certain range, generate unique neural activity patterns for each distinct location, and reactivate these patterns whenever a "loop is closed" during revisits to old locations. The outputs of the CAN can therefore be used to generate novel "labels" for new locations and

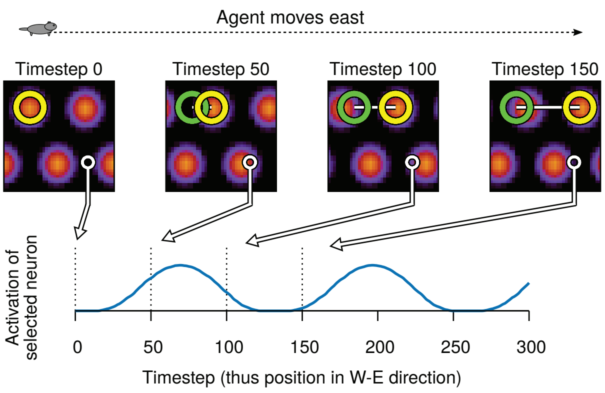


Figure 2: Illustration of key concepts behind continuous attractor networks, which constitute a major class of computational models for grid cells. Each neuron is assigned a row and a column in a "neural sheet". Neurons are fully connected recurrently, with connection strengths pre-wired in such a way that bumps of activity will spontaneously emerge in the neural sheet and distribute in a hexagonal pattern. This hexagonal pattern becomes visible in the readouts of individual neurons across space because the pattern in the neural sheet is made to shift in proportion to the agent's movements in 2D space. The square heatmaps show instantaneous snapshots of the activation levels of the neurons in the neural sheet of a CAN-based grid module, each pixel indicating the activation level of one neuron. These snapshots are shown at four different timesteps, all while the agent is moving eastward. In response to the velocity input, the pattern shifts rightward in the neural sheet (indicated by the yellow circle, which tracks the motion of one of the bumps in the pattern). By reading out the activation level of a single neuron and plotting it over time, the grid pattern is revealed and the neuron therefore acts as a grid cell.

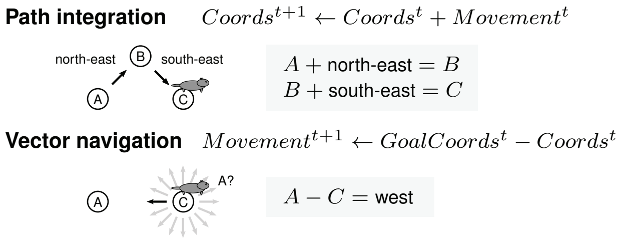


Figure 3: Path integration maintains an internal estimate of the agent's current coordinates based on its movement history. Vector navigation calculates the direction to a goal location based on the current coordinates and the goal coordinates.

then reactivate those labels when an old place is visited anew [14], therefore helping the system downstream of the CAN to perform place learning and place recognition.

However, using the neural representation of grid cells, another function also presents itself. Not only is there a systematic way in which grid cell activity can be generated by a path integration process, but the reverse can also be said: Grid cell activity from location A and location B can be compared in order to extract the direction of movement between them. The pieces are therefore in place for grid cells to be used as a 2D coordinate system in artificial neural navigation systems; coordinates can be updated by adding new velocity vectors to them, and movement directions can be recovered by comparing/"subtracting" coordinates (Fig. 3). Bush et al. [15] provide several possible neural networks for decoding grid cells to movement vectors, and in earlier work [6] we showed a neural system where both path integration and grid cell decoding is integrated in the same agent controller. There are yet, however, many possibilities and questions left to explore in this area, one of which—the possibility of using grid cells for navigation over long distances—is the topic of this paper.

3 Navigating with a single grid module

One of the defining characteristics of a grid cell is that the neuron's activity pattern repeats across space, so it is not possible to unambiguously determine the animal's location by making a read-out of one grid cell's current activation level. The simultaneous activation levels of other grid cells must therefore also be considered. Biological grid cells are known to cluster into *grid modules*, where all neurons in a grid module share the same hexagonal grid pattern across space (identical scaling and orientation of the pattern), except for a shift/offset in the pattern between neurons. Cells 5–7 in Fig. 1 might thus belong to the same grid module. The read-out of an entire grid module will have the same ambiguity due to repeating grid patterns as an individual grid cell—after traveling a distance sufficient to make one grid cell start to repeat itself, then all the other grid cells in the module will also have started repeating. Grid cells organized in a module make it possible to determine the animal's location within the boundaries of the module's "unit tile", but it does

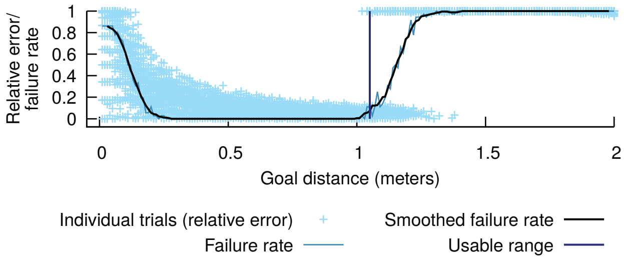


Figure 4: Navigating with a single grid module. The agent was tested for "goal distances" in 1 cm increments from 1 cm to 200 cm, conducting 36 trials in 10 degree increments for each distance. A trial consisted of the agent driving in the specified direction, halting once the distance to the origin was within 10^{-6} m of the "goal distance", and then being allowed twice the number of outbound timesteps to try to navigate back to the origin. The closest approach to the origin was divided by the initial halting distance to obtain a "relative error" for the trial. Trials with relative errors of 0.5 or more were considered failures, and the failure rate was calculated for each distance bin. This rate was then smoothed by averaging it in a sliding window of 5 bins. The "usable range" was determined as the last distance bin before the smoothed failure rate reached a threshold of 0.1 (skipping the initial region of failed trials).

not reveal any information about where this unit tile might be located in global space.

We sought to demonstrate the limits of navigating with only one grid module in our current grid cell-based navigation system. The system performs path integration using a configurable number of CAN-based grid modules as the simulated agent drives away from the origin location, and then later tries to find its way back to the origin by decoding the information residing in the recurrent short-term memory of the grid modules. The version of our navigation model used in this paper is in most respects similar to how it was described in [6].

Fig. 4 shows the results from an experiment where the agent was tested at various goal distances between 1 cm and 200 cm. Using a criterion described in detail in the figure caption, we calculated the "usable range" to be 1.05 meters.

A navigational range on the order of one meter will clearly not be sufficient for a number of applications, and neither is this range behaviorally sufficient

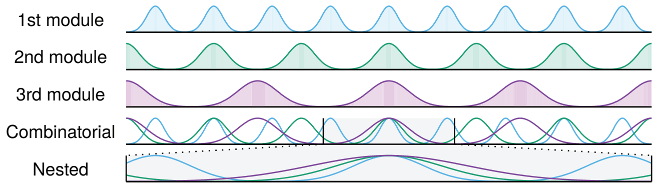


Figure 5: Conceptual comparison of two different views on how the ambiguity from a single grid module can be resolved—and the usable range thus be increased—by utilizing multiple grid modules of increasing scale. Depending on the particular grid scales used, the grid modules can continue to generate unique activity combinations over distances exceeding the range of the largest-scaled module in a "combinatorial" fashion. A different view is to assume that there is a hierarchy of grid modules such that the largest module is sufficiently large to cover the entire behavioral range of the animal and to guide the animal into the usable range of the smaller-scaled modules. The smaller-scaled modules are thus "nested" within the larger-scaled ones, their purpose being to increase precision beyond what the larger-scaled modules might provide. Grid scales observed in neuroscientific experiments are known to follow a geometric progression, so both of these views remain viable options.

for animals such as rats and bats that forage across distances up to kilometers away from the nest [16]. However, the fact that there are multiple grid modules in the brain might provide a solution.

4 Navigating with multiple grid modules

There are multiple grid modules in the brain, and there appears to be a constant scaling factor between the spatial grid scales of successive modules, so that the grid scales of a sequence of grid modules form a geometric progression [17]. This enables two distinct views of how the grid cell system might unambiguously represent 2D coordinates over longer distances (Fig. 5), which we will refer to as respectively as the "combinatorial" and the "nested" approaches. The "combinatorial" approach emphasizes that the collective activity of a set of grid modules can remain unique over a total range far exceeding that of the largest grid module. This range can theoretically be as much as the least common multiple of all of the modules' grid scales [18], which increases quickly when adding just a few grid modules together—however, it has been argued that utilization of the full range requires precise readouts from each module [19], and the possibility has been raised that an error correction mechanism might be required during ongoing path integration operation [20]. The "nested" approach embraces the fact that the grid scales of successive modules appear to follow a geometric progression. As the number of grid modules increases, the grid scale of the largest module grows exponentially. In this view, one therefore assumes that there is a sufficient number of grid modules so that the largest grid scale is larger than the behavioral range of the animal, i.e., that there is a grid module large enough to by itself unambiguously indicate the animal's location in 2D space. While the resolution of a grid module might grow coarser as the grid scale increases—so that a very large-scaled grid module might only give a rough estimate of the animal's location—smaller-scaled grid modules could provide refinements to the initial estimate from the largest module [21].

It is the latter view we will consider in this paper; we showed in [6] that there exists a simple neural grid cell decoding mechanism for such setups. The decoder considers the grid cell signal in each grid module individually and then at the output end of the system gives priority to the largest-scale grid module outputs, reminiscent of the nested refinement described by [21]. Following this principle, in order to increase the range of the navigation system from $\sim$1 meter to e.g. 150 meters, we should repeatedly add extra grid modules to the system in a geometric progression until the projected range of the largest module is sufficiently large. Using a scale ratio of 1.5, as suggested by theoretical studies and within the range of scale ratios reported from neuroscientific experiments, we would need 14 grid modules in total ($1.05\text{ m} \cdot 1.5^{13-1} \approx 136$ m, $1.05\text{ m} \cdot 1.5^{14-1} \approx 204$ m).

Fig. 6 illustrates an overall schematic of the full grid cell-based navigation system with all of these pieces in place. A number of CAN-based grid modules, following a geometric progression of grid scales, receive self-motion velocity in order to perform path integration. The output from these modules is decoded in a nested fashion to perform vector navigation. The illustration also alludes to the specific mechanism by which the different grid scales are attained in the otherwise identical CAN modules—by increasingly *attenuating* the velocity input to each module. We will return to this topic in later sections.

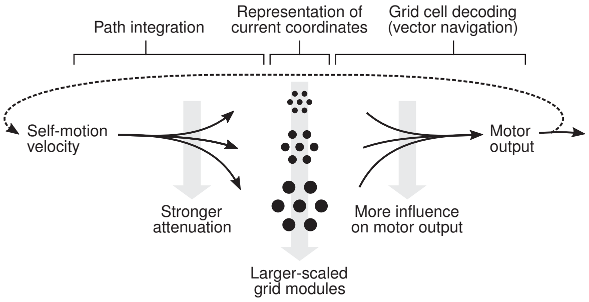

Path integration
Representation of current coordinates
Grid cell decoding (vector navigation)
Self-motion velocity
Motor output
Stronger attenuation
More influence on motor output
Larger-scaled grid modules

Figure 6: Conceptual overview of the full navigation model considered in this paper. Self-motion velocity is used to update multiple CAN-based grid modules of increasing grid scales that follow a geometric progression. Larger grid scales are achieved by attenuating the input to the grid module, so that the module's produced grid pattern appears stretched across space. The collective activity of all grid modules represents a set of "coordinates" in the navigation model. Larger-scaled grid modules are given priority in the decoding process, in accordance with the view that smaller-scaled grid modules are "nested" within the larger ones [21].

Fig. 7 shows the results of an experiment that follows the paradigm in Fig. 4, but where much longer distances are tested and where separate sets of trials were performed with the model configured to use 2, 4, 6, 8, 10, 12 and 14 grid modules. The set of trial distances was selected to be equally spaced on a logarithmic scale, because we are interested in observing behaviors of the system that are expected to follow an exponential development. Usable ranges were determined as in Fig. 4, and aggregated results are shown in Fig. 8.

The system behaves as expected for low numbers of modules—as extra modules are added, the usable range increases exponentially. However, at 12 and 14 modules there are clear deviations from this trend. The usable range appears to level off around the level attained with 10 modules. The extra grid modules added from 10 to 12 and 14 do not seem to have contributed appreciably to the usable range of the system. In the remainder of this paper we will look into the cause of this problem and discuss possible solutions.

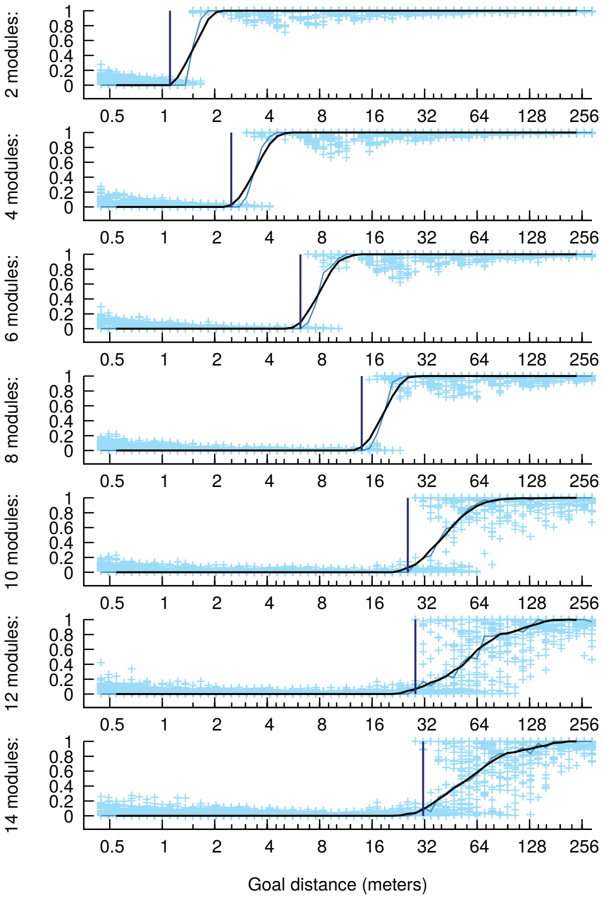


Figure 7: Navigating with multiple grid modules. The trials and analyses were conducted as in Section 3, but with either 2, 4, 6, 8, 10, 12 or 14 modules, and with 65 tested distances selected to be equally spaced on a logarithmic scale from $1.5^{-2} \approx 0.444$ m to $1.5^{14} \approx 292$ m. Legend as in Fig. 4.

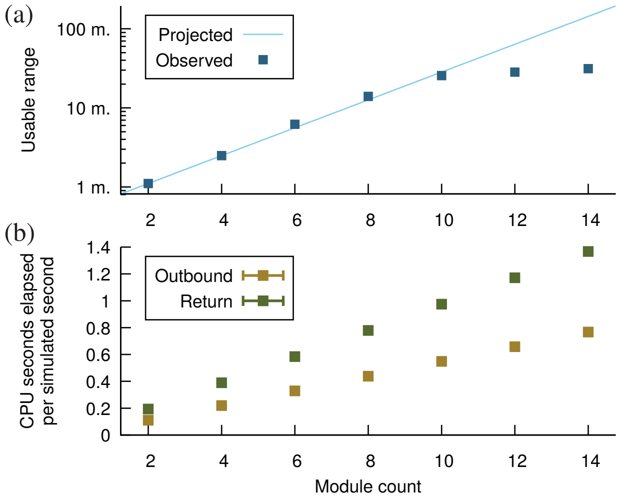


Figure 8: Various results from the trials shown in Fig. 7, aggregated by the number of modules. (a) Usable range for each number of modules, as determined in Fig. 7. Also shows the expected range as a function of number of modules, projected from the range value found for 2 modules and using a factor of 1.5 per module. (b) Mean ± std.dev. across trials, of CPU time spent per simulated time. 1 simulated second corresponds to travelling 0.2 meters. “Outbound” is the initial excursion away from the starting location, and “return” is the following navigation phase where a return back to the starting location is attempted.

5 “Pinned” pattern flows in CANs

The source of the shortcoming is demonstrated in Fig. 9. In this experiment, the flow of the activity pattern in a CAN is measured for different strengths of net input velocity. The relevance to the grid scaling problem is that this is currently how the CANs’ grid scales are increased from the baseline grid scale—by increasingly attenuating the velocity inputs to the modules that should have larger grid scales.

The average flow reported for the unattenuated case in Fig. 9 was $\sim$981 neurons per 100 meters. In order to produce a grid module with e.g. twice the grid scale, the velocity input to that module should be half of the original signal—the module would then use twice the amount of time/distance to produce the same output as the unattenuated module, in effect producing a grid pattern stretched to twice the spatial scale. In general, to scale the grid pattern by a factor s, we attenuate the velocity input by $a = s$ by dividing it by a.

For this strategy to be successful, the network response should always remain proportional to the net velocity input. Thus, with an attenuation of e.g. 100—in order to produce a grid scale of 100 times the normal scale—we want the network pattern to flow $981/100 = 9.81$ neurons per 100 meters, etc. Fig. 9 explores whether this is the case, by measuring the network flow at various attenuation levels between 1.5^{-5} and 1.5^{13}, equally spaced on a logarithmic axis.

The specific attenuation levels used for the 14 different modules in Section 4 are indicated in the figure. At the attenuation levels observed by modules 1–8, the network behavior is indeed inversely proportional to the attenuation, so that the reported flow values are around 1 as a proportion of the “target flow”. However, for stronger attenuations the behavior starts to break down. Between the attenuation levels corresponding to the 8th and the 13th modules, there is an increased spread in the observed flow values, and consequently the grid module is less reliable as a path integrator. At yet higher attenuation levels, the CAN barely seems affected by the input signal and is unable to perform any path integration at all. This phenomenon can account for the ineffectiveness seen in Figs. 7 and 8 of adding any extra modules beyond 10. The phenomenon corresponds to what Burak & Fiete refer to as “pinning” of the network pattern at low velocities [11].

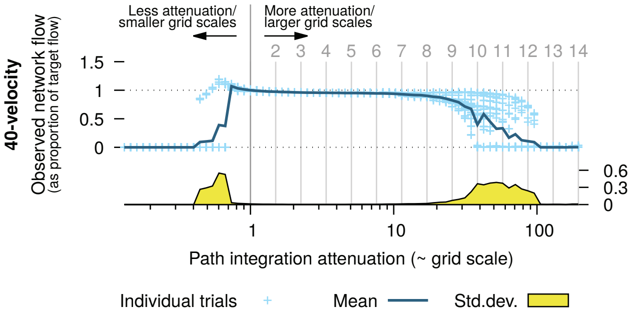


Figure 9: Tracking the flow of the network pattern in a CAN-based grid module, across a range of different attenuation levels for the velocity inputs to the CAN. The attenuation is 1 for the smallest-scaled, i.e. the first, grid module. To create larger grid scales, the velocity inputs are attenuated increasingly—the attenuation levels corresponding to the 14 grid modules used in Fig. 7 are marked by vertical lines. For each of 73 different attenuation levels, equally spaced on a logarithmic scale from 1.5^{-5} to 1.5^{13}, 36 trials were conducted in 10 degree increments. Each trial consisted of the agent making an outbound excursion for 600 meters in the specified direction, with the velocity inputs to the single grid module attenuated at the specified level, followed by a rest phase. The flow of the pattern in the neural sheet of the CAN was tracked by following the motion of one of the activity bumps in the sheet, and reported as the number of neurons the pattern shifted in the x and the y directions. The Euclidean distance of this pattern shift was considered the "observed network flow" of the trial. The "target flow", i.e. the amount of flow we would expect for a given attenuation level if the module behaved completely linearly, was calculated as the mean observed flow at $x = 1$ divided by the attenuation. The outcome of each trial was plotted as the observed flow divided by the target flow, and the goal is for these points to fall along $y = 1$. The mean across the 36 trials for each particular attenuation level is shown as a dark line, and the corresponding std.dev. is shown on a separate y axis for clarity.

6 Avoiding pinning

With this limited range of attenuation levels that the CAN will accept, we will not be able to extend the navigational range of the system no matter how many extra modules we include. All of the large-scaled modules will experience pinning and therefore not provide any useful navigational signal for their respective scales. In order to extend our grid cell-based navigation model to longer ranges, we thus need to solve the problem of implementing CAN modules for large grid scales while avoiding pinning. The issue at hand is that the input signals to the network simply get too weak for the network dynamics to be able to respond, i.e. for the network pattern to reliably shift to adjacent states. As an example, the attenuated velocity signal that reaches the 14th module is only $1.5^{-13} \approx 0.005$, i.e. half a percent, in strength compared to what the first module receives. However, we have implemented all of the CAN modules using identical network dynamics. Implicitly we therefore require our CAN modules to respond proportionally for inputs across several orders of magnitude, which is quite a tall order. In this section we will consider a few options for solving this problem.

One class of candidate solutions is to expand the CAN to make it better equipped to respond to input levels from a wider range. In this paper we will consider the effects of increasing the size of the CAN's neural sheet, i.e. increasing the number of neurons that participate in the grid module. The idea is that the CAN might get a better response range the more neurons it contains—the weak input signal reaches a larger number of neurons, thus the chances might be higher that the collective response of the network will be able to overcome the pinning phenomenon. We will test the effects of increasing the neural sheet size from 40^2 neurons to 48^2, 56^2 and 64^2 neurons.

The other class of solutions we will consider is to adjust the update rates of the grid cells as a function of the attenuation level—essentially to "subsample" the sequence of velocity values provided to the grid cells. Take our previous example of a module that shifts its network pattern 981 neurons per 100 meters, whose grid pattern we now want to scale up by a factor of 10; our current baseline approach is to achieve this grid scaling by dividing the input speed by 10. However, assuming that the network response correctly reflects the time integral of the input signal, an equivalent way of obtaining the same effect

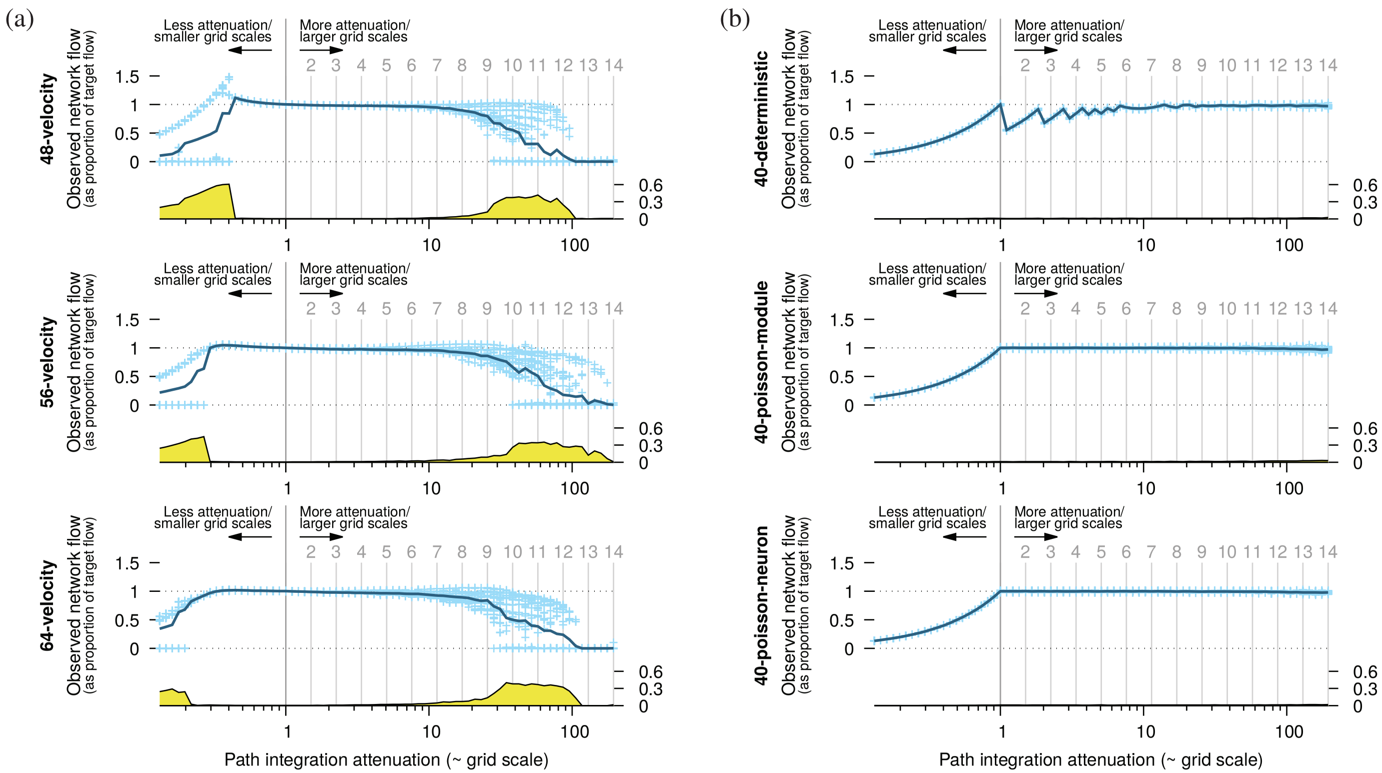


Figure 10: Tracking the network flow under our various proposed solution schemes. Experiments were conducted as in Fig. 9, but with the model configured as indicated in bold to the left of each figure. "Target flow" is based on values in Fig. 9. (a) Trials with the number of neurons in the CAN module increased from 40^2 to 48^2, 56^2 or 64^2. (b) Trials with attenuation implemented through other means than by scaling the input velocity. Legend as in Fig. 9.

should be to use the original strength for the input signal, but to *only update the network a tenth as often*. From the perspective of the CAN module, the input history will reflect moving at full speed for 10 meters, rather than moving at 1/10th speed for 100 meters. However, because the agent will in fact have moved 100 meters spatially, the module's output should appear 10 times stretched across space, as intended.

There are several ways to go about implementing this basic idea. One is to update the entire module on fixed, repeating timestep intervals, and to skip updating the module at all other timesteps. We refer to this as the "deterministic" update mode. Two aspects of the deterministic update mode motivate a further development. First, this update mode is not able to generate all possible grid scales—there is e.g. no way to achieve an attenuation of 1.5 with this scheme (the skip interval necessarily has to be an integer, so the first possible skip amount beyond 0 is 1, skipping every other network update and resulting in an attenuation of 2). Second, with a deterministic update mode the system will be susceptible to failure in cases e.g. where the agent moves in periodic patterns that match the module's update rate. For example, if the agent follows a movement pattern that cycles every 10 timesteps, a module that updates precisely every 10th timestep will not be able to adequately sample the full trajectory of the agent.

To alleviate this, we introduce stochasticity into our update rule. Instead of calculating a discrete, deterministic update interval from the desired attenuation level, we calculate an update *probability* instead. At every timestep of the model, the entire module is updated by chance according to that update probability. Because this is a memoryless criterion—not relying on the model's timestep counter, as the deterministic mode does—we call this the "poisson-module" update mode. This scheme should solve our two reservations about the deterministic update mode. An attenuation of 1.5 would be achieved by updating the module with a probability of $1/1.5 \approx 0.67$ on every timestep, etc. Due to the probabilistic nature of the update rule, it should not be vulnerable to periodic fluctuations in the animal's velocity.

In both of these update modes, for strong attenuations, the module might go a large number of timesteps between each update and thus miss out on a substantial amount of velocity information. As our final proposed update scheme, we suggest to apply the probabilistic update rule *individually to each neuron*

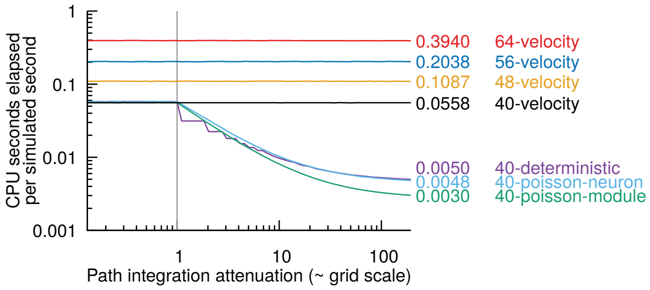


Figure 11: CPU demand for the various schemes presented in Figs. 9 and 10. Mean CPU seconds per simulated seconds across trials for each attenuation level and model configuration. The labels to the right show the mean values at the rightmost data points.

on every timestep, rather than to the module as a whole. Each neuron should over time experience the same update rate as with the module-wide probabilistic update rule, but because it is applied individually to each neuron every timestep, there might potentially be an active subset of neurons in any given timestep. We term this the "poisson-neuron" update mode.

To evaluate the effect of these proposed solution schemes, Fig. 10 shows the results of performing the same experiment as in Section 5 for each of the following six configurations:

48-velocity, 56-velocity, 64-velocity:

Velocity-based attenuation, i.e. the original approach used in earlier sections of this paper, but with the number of neurons in the CAN increased from 40^2 to respectively 48^2, 56^2 and 64^2.

40-deterministic, 40-poisson-module, 40-poisson-neuron:

Subsampling-based attenuation as described above, while leaving the CAN size unchanged at 40^2.

From Fig. 10a we can see that increasing the number of neurons in the CAN module does help to increase the range of viable attenuation levels, and thus the range of grid scales attainable. However, the improvement occurs at the wrong end of the scale, for attenuation levels less than 1. These attenuation levels would be used to produce smaller-scaled grid modules, but there is not any major improvement for the high attenuation levels, which is what we need in order to increase the overall usable range of the system. Fig. 10b shows the results from the update modes based on subsampling. The deterministic mode, as expected, shows artifacts from only being able to correctly represent integral attenuation levels. Both the per-module and per-neuron probabilistic update modes show promising results; for all attenuation levels from 1 and up, the observed network flows are close to the desired target flows needed to produce the intended grid scales. Neither of the subsampling-based update modes are built to adhere to attenuation levels less than 1, which would require updating the modules/neurons more often than once per timestep.

Fig. 11 shows timing results obtained during the trials in Figs. 9 and 10. For each trial the amount of CPU seconds elapsed per simulated outbound second was calculated, and these values were then averaged across trials for each attenuation value, for each tested model configuration. As expected, the CPU time for the velocity-based configurations is constant as the attenuation level changes, because the same amount of calculation takes place regardless of the attenuation. Note, however, that there is a substantial increase in CPU time going from 40^2 neurons to larger CAN modules. For all of the subsampling-based configurations, the CPU time decreases as the attenuation grows.

7 Successful long-range navigation

The results in the previous section lead us to favor the “poisson-module” and “poisson-neuron” update modes—as shown in Fig. 10b they are both able to represent large grid scales, and as shown in Fig. 11 they both require gradually less computational time as the grid scale increases. Future work should characterize whether there is any substantial difference between the two update modes e.g. in their ability to handle more fluctuating trajectories than the straight lines tested here, but for now we will proceed with the “poisson-neuron” update mode as our choice to evaluate in the full navigational system.

We re-ran the experiments from Section 4 using the “poisson-neuron” update mode, with the results corresponding to Figs. 7 and 8 shown respectively in Figs. 12 and 13. Particularly comparing Figs. 8a and 13a, we can see that the

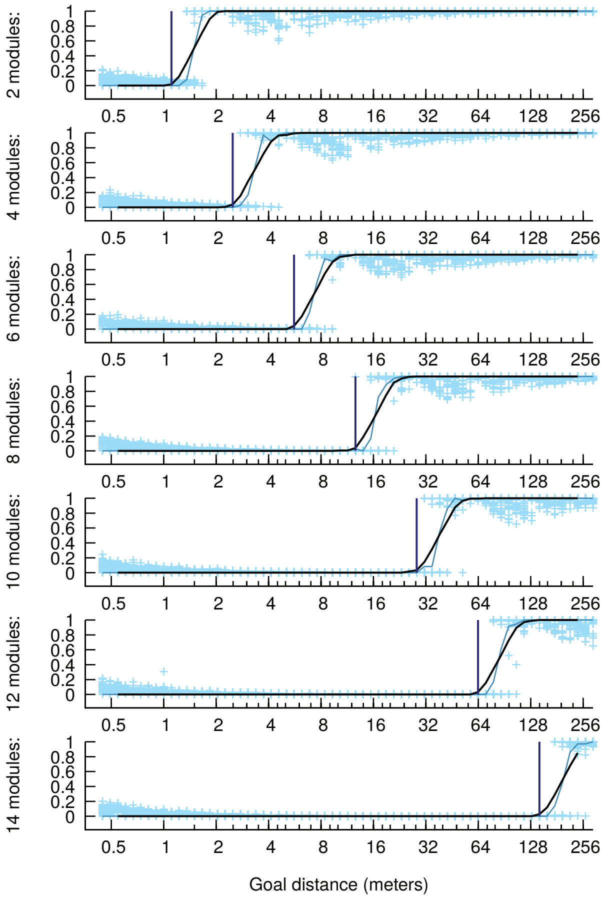


Figure 12: Same experimental setup as in Fig. 7, but now using the "poisson-neuron" update mode. Legend as in Fig. 4.

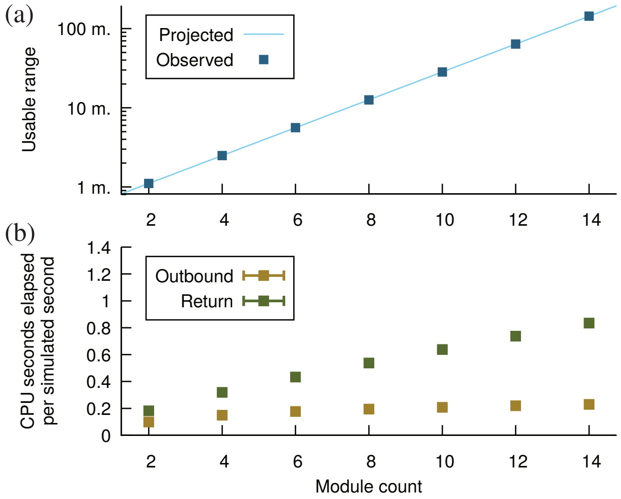


Figure 13: Aggregate results as in Fig. 8, but now based on the experiment shown in Fig. 12, where the "poisson-neuron" update mode is used.

shortfall previously experienced at 12 and 14 modules has been resolved—the viable range keeps growing exponentially as more modules are added beyond 10. Comparing Figs. 8b and 13b we can see that the updated model requires less CPU time than before, and that the growth is at most linear in the number of grid modules added to the system.

8 Discussion

The basis for this project was a grid cell-based neural navigation system capable of vector navigation based on path integration processes in a geometric progression of grid modules [6]. We sought to determine whether such a system, based on CANs and nested decoding of grid cells, can support vector navigation over long distances. We found that the geometric progression of grid scales is interrupted due to "pinning" of network patterns in the CAN modules at the low input velocities used to implement large grid scales. Though this work specifically used a CAN grid cell model, the key issue of accommodating strong attenuation may also be relevant for the dynamics in other grid cell models. We assessed several candidate solutions and found that a probabilistic update rule in each grid module/grid cell can successfully implement large grid scales. Using this new approach, the system restored its exponential growth in range as the number of grid modules increases, enabling navigation beyond $\sim$100 meters as seen in these experiments and expectedly to much farther distances.

Bush et al. [15] demonstrated successful vector navigation with grid cells over distances of hundreds of meters, albeit in a setup where the grid cell signal was externally generated and thus would not suffer any issues arising from a neural path integration process. In any case, their largest grid scale is only $\sim$5 meters, the large final navigational range achieved by using a decoder capable of utilizing the "combinatorial" view of the grid cell code. In our present work we show that navigation at distances of $\sim$100 meters can also be accomplished when taking a "nested" view of the grid cell code and producing the grid cell signals in path integrating neural networks.

We saw an exponential growth in viable range as extra grid modules were added, with at most a linear growth in CPU time. We have thus demonstrated a grid cell-based neural navigation system where the CPU requirements are logarithmic in the desired navigational range. This is encouraging for the prospects for implementing this system to run in real time on a robot. For such an application it would also be natural to integrate additional spatial neurons, such as place cells.

Acknowledgement

Thanks to Keith Downing for helpful feedback on this work.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[4] D. L. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nat. Neurosci.*, vol. 19, no. 3, pp. 356–365, 2016.

[5] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[6] V. Edvardsen, "Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network," *Natural Computing*, 2016.

[7] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat," *Brain Res.*, vol. 34, no. 1, pp. 171–175, 1971.

[8] J. S. Taube, R. U. Muller, and J. B. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis," *J. Neurosci.*, vol. 10, no. 2, pp. 420–435, 1990.

[9] T. Solstad, C. N. Boccara, E. Kropff, M.-B. Moser, and E. I. Moser, "Representation of geometric borders in the entorhinal cortex," *Science*, vol. 322, no. 5909, pp. 1865–1868, 2008.

[10] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–806, 2005.

[11] Y. Burak and I. R. Fiete, "Accurate path integration in continuous attractor network models of grid cells," *PLoS Comput. Biol.*, vol. 5, no. 2, 2009.

[12] L. M. Giocomo, M.-B. Moser, and E. I. Moser, "Computational models of grid cells," *Neuron*, vol. 71, no. 4, pp. 589–603, 2011.

[13] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *Int. J. Robot. Res.*, vol. 29, no. 9, pp. 1131–1153, 2010.

[14] F. Carpenter and C. Barry, "Distorted grids as a spatial label and metric," *Trends Cogn. Sci.*, vol. 20, no. 3, pp. 164–167, 2016.

[15] D. Bush, C. Barry, D. Manson, and N. Burgess, "Using grid cells for navigation," *Neuron*, vol. 87, no. 3, pp. 507–520, 2015.

[16] M. Geva-Sagiv, L. Las, Y. Yovel, and N. Ulanovsky, "Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation," *Nat. Rev. Neurosci.*, vol. 16, no. 2, pp. 94–108, 2015.

[17] H. Stensola, T. Stensola, T. Solstad, K. Frøland, M.-B. Moser, and E. I. Moser, "The entorhinal grid map is discretized," *Nature*, vol. 492, no. 7427, pp. 72–78, 2012.

[18] I. R. Fiete, Y. Burak, and T. Brookings, "What grid cells convey about rat location," *J. Neurosci.*, vol. 28, no. 27, pp. 6858–6871, 2008.

[19] X.-X. Wei, J. Prentice, and V. Balasubramanian, "A principle of economy predicts the functional architecture of grid cells," *eLife*, vol. 4, 2015.

[20] S. Sreenivasan and I. Fiete, "Grid cells generate an analog error-correcting code for singularly precise neural computation," *Nat. Neurosci.*, vol. 14, no. 10, pp. 1330–1337, 2011.

[21] M. Stemmler, A. Mathis, and A. V. Herz, "Connecting multiple spatial scales to decode the population activity of grid cells," *Sci. Adv.*, vol. 1, no. 11, p. e1500816, 2015.

Paper D

Navigating with grid and place cells in cluttered environments
(Edvardsen et al., 2019)

Author(s):

Vegard Edvardsen, Andrej Bicanski and Neil Burgess

To appear in journal:

Hippocampus *(in press; accepted on July 19th, 2019)*

Note:

This paper is accompanied by a Supplementary Materials document and four Supplementary Videos. These are included here directly after the paper itself; the videos in excerpt form, represented by a selection of video frames.

The paper was accepted by *Hippocampus* in July 2019, and is currently awaiting publication.

Paper D

Edvardsen et al. (2019)

Navigating with grid and place cells in cluttered environments

Vegard Edvardsen[1], Andrej Bicanski[2], and Neil Burgess[2]

[1]Department of Computer Science, NTNU – Norwegian University of Science and Technology, 7491 Trondheim, Norway
[2]Institute of Cognitive Neuroscience, University College London, Alexandra House, 17 Queen Square, WC1N 3AZ London, United Kingdom

Abstract

The hippocampal formation contains several classes of neurons thought to be involved in navigational processes, in particular place cells and grid cells. Place cells have been associated with a topological strategy for navigation, while grid cells have been suggested to support metric vector navigation. Grid cell-based vector navigation can support novel shortcuts across unexplored territory by providing the direction towards the goal. However, this strategy is insufficient in natural environments cluttered with obstacles. Here, we show how navigation in complex environments can be supported by integrating a grid cell-based vector navigation mechanism with local obstacle avoidance mediated by border cells and place cells whose inter-connections form an experience-dependent topological graph of the environment. When vector navigation and object avoidance fails (i.e. the agent gets stuck), place cell replay events set closer subgoals for vector navigation. We demonstrate that this combined navigation model can successfully traverse environments cluttered by obstacles and is particularly useful where the environment is under-explored. Finally, we show that the model enables the simulated agent to successfully navigate experimental maze environments from the animal literature on cognitive mapping. The proposed model is sufficiently flexible to support navigation in different environments, and may inform the design of experiments to relate different navigational abilities to place, grid and border cell firing.

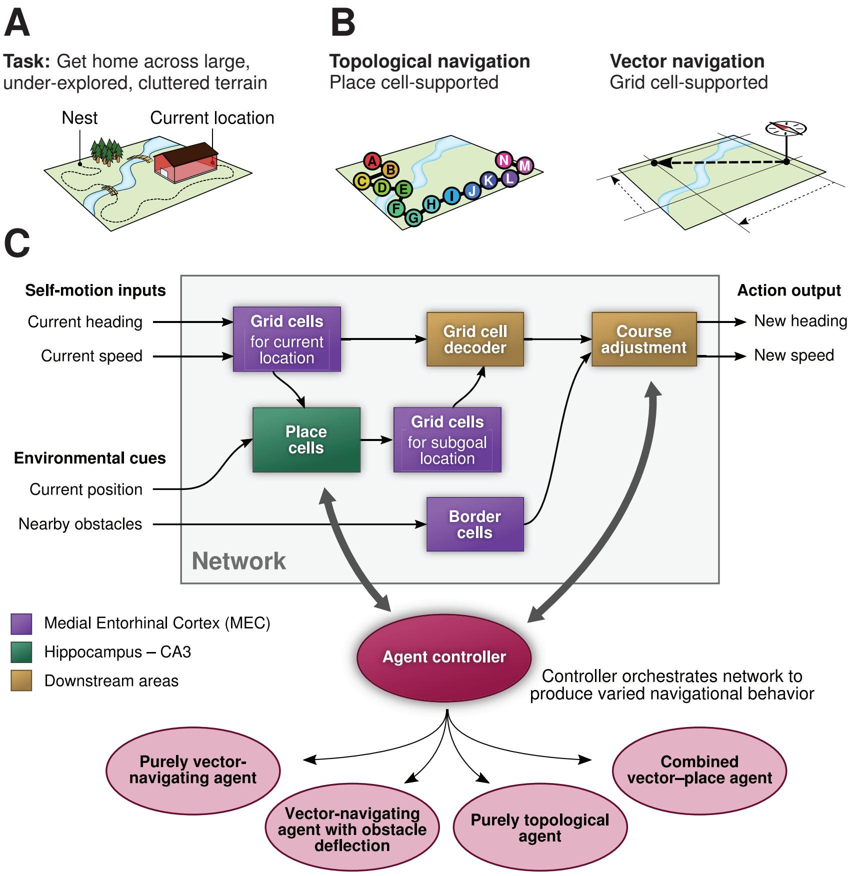


Figure 1: (A) Stereotypical navigation task. An agent has traveled across unknown terrain to a remote location and wishes to return to its nest, with limited knowledge of the environment. (B) Two major navigation paradigms supported by neurophysiological evidence. Place cells likely support topological navigation, where knowledge about locations' interconnectivity is used to reach the goal. Grid cells likely enable the calculation of distances and angles for straight-line trajectories between arbitrary pairs of previously visited locations (vector navigation). (C) Model overview. Network portion (gray box) remains fixed across all trials. An external agent controller orchestrates components in the network in order to produce a variety of navigational strategies; either primarily vector-based navigation, primarily topological navigation, or combined strategies utilizing a mixture of information from grid cells, place cells and border cells. A grid cell decoder performs vector navigation toward a subgoal provided by the place cells. Border cells provide local obstacle information to a course adjustment mechanism. Box colors indicate related areas in the hippocampal formation.

1 Introduction

Successfully navigating the environment is a problem common to most animals. There is a wide range of approaches to navigation, mirroring the wide range of behavioral requirements across different species (Trullier et al., 1997). In mammals, navigation is thought to be supported in part by a "cognitive map" (O'Keefe and Nadel, 1978), an internal neural representation of space. Such a map would endow an animal with navigational planning capabilities that should enable it to robustly find its way to previously visited locations (Fig. 1A). The theoretical notion of the cognitive map is supported by compelling neurophysiological evidence. Hippocampal place cells represent unique locations in the environment (O'Keefe and Dostrovsky, 1971), and the more recently-discovered grid cells in the medial entorhinal cortex (Hafting et al., 2005) appear to provide a spatial metric by encoding the animal's coordinates in the two-dimensional plane. The discoveries of head-direction cells (Taube et al., 1990a,b) and border cells/boundary vector cells (Barry et al., 2006; Solstad et al., 2008; Lever et al., 2009) further strengthen the hypothesis of an internal neural map of space.

Place cells, particularly in hippocampal area CA3, are thought to form interconnections through recurrent synapses such that neighborhood relationships between locations in the explored environment might be retrievable from the synaptic strengths between place cells. Such a system could implement a "topological navigation" strategy (Fig. 1B), whereby the agent navigates to its goal location by calculating the shortest path in this internal representation of the environment and then following the resultant sequence of place cells' firing fields as its itinerary. Many models of place cell-based navigation have emphasized this topological view, considering recurrent synapses among place cells to encode connectivity, distance or transition probability between locations (Mataric, 1991; Muller et al., 1996; Blum and Abbott, 1996; Redish and Touretzky, 1998; Gillner and Mallot, 1998; Stachenfeld et al., 2017).

Grid cells have been suggested to support goal vector representations (Erdem and Hasselmo, 2012; Kubie and Fenton, 2012; Edvardsen, 2015; Bush et al., 2015). Given grid cell activity for the present location and a trace of the grid cell activity for the goal location, the appropriate straight-line vector across two-dimensional space can be determined. The grid cell network can then support a “vector navigation” strategy (Fig. 1B; Mittelstaedt and Mittelstaedt, 1980; Etienne et al., 1996), assuming the allocentric goal vector can be transformed to the agent’s egocentric frame of reference. Head-direction cells and parietal gain field neurons (Snyder et al., 1998; Pouget and Sejnowski, 1997) have been suggested to provide this function (Burgess et al., 2001; Byrne et al., 2007; Bicanski and Burgess, 2018). The agent can then find the correct bearing toward the goal location even across large stretches of unexplored space, due to the metric properties of grid cells (Fiete et al., 2008; Carpenter et al., 2015).

According to these separate classes of computational model, place cells and grid cells seem to support complementary navigational strategies. Either strategy alone has its strengths and weaknesses: Place cell-based topological navigation excels at finding the shortest possible paths needed to reach goals in cluttered and complicated environments, possibly including detours around known obstacles, but only if the agent has explored the environment extensively in advance, forming the necessary topological map. Conversely, grid cell-based vector navigation can rely on goal vectors, even across long stretches of potentially unknown terrain, but obstacles along the straight-line path to the goal might cause the agent to get stuck. A navigational strategy based on grid cells alone would not be sufficient outside of obstacle-free open-field environments, raising the question of whether or how grid cells participate in the navigation process under real-world conditions.

Here we show how a grid cell-based vector navigation model can be augmented to cope with environments cluttered by obstacles, based on known aspects of hippocampal function. Medial entorhinal cortex (mEC) layer II, where grid cells are most prevalent, is a major input to the hippocampus, and the hippocampus in turn projects back to deeper layers of mEC. While the suggested interplay of grid cells and place cells has been modeled extensively at the circuit level, such work has usually focused on maintaining the firing properties of one population based on inputs from the other (Rolls et al., 2006; Solstad et al., 2006; Kropff and Treves, 2008; Hardcastle et al., 2015; Mulas et al., 2016; Dordek et al., 2016; Stachenfeld et al., 2017). Here we investigate how the distinct characteristics of these two representations of space can interact to guide behavior. We suggest a role for hippocampal replay events during navigation, using place cells to dynamically adjust the target for the vector navigation process, based on the intriguing possibility that place cells and grid cells can fire coherently during replay (Ólafsdóttir et al., 2016). Additionally, the existence of border cells (Solstad et al., 2008) suggests that a grid cell-supported vector navigation mechanism might have access to information about nearby obstacles, and hence the ability to make course adjustments based on their presence. Boundary vector cells (Barry et al., 2006; Lever et al., 2009) could serve a similar function, signaling boundary presence at greater distance, forgoing the need for actual boundary contact before deflecting a trajectory. We combine these aspects of topological navigation and local obstacle avoidance with a grid cell-based vector navigation model, and demonstrate that such an augmented vector navigation mechanism can efficiently navigate cluttered environments. The combined navigational strategy enables the agent both to negotiate complicated obstacles and to efficiently traverse long distances of unexplored space, potentially exploiting shortcuts.

2 Materials and Methods

Here we present the architecture and main features of the proposed hippocampal navigation model (Fig. 1C), consisting of grid cells decodable to goal vectors, border cells for local obstacle avoidance, and a topological map implemented by place cells. Different types of obstacles present different challenges during navigation, and we describe how the model utilizes its components in concert to overcome these challenges. Grid cells, the grid cell decoder and obstacle avoidance mechanism are represented by rate-based neural networks in our implementation. The networks do not need any advance training, as the weights have been explicitly pre-configured for their intended roles in the model (see McNaughton et al. (2006), Kubie and Fenton (2012) for how such grid cell networks might be obtained through learning). For simplicity, the place cell system is represented directly by a graph data structure, and the agent's high-level control logic is represented by explicit rules. A more detailed description of the implementation is given in the Supplementary Materials.

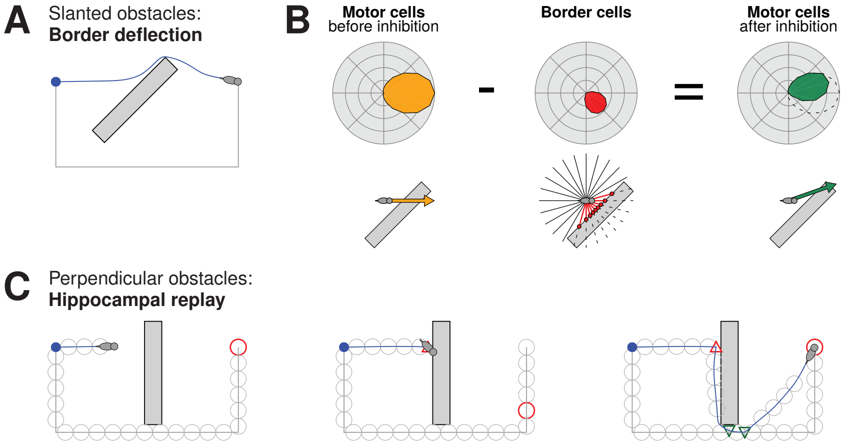


Figure 2: (A) Obstacles that form a slanted angle (less than 90°) with the goal vector can be negotiated by deflecting the direction of motion away from the obstacle. Goal distance keeps decreasing along the diverted trajectory. (B) Obstacle deflection mechanism: A bump of activity is induced in a ring of motor cells, pointing in the direction of the goal vector decoded from the grid cells. Each border cell responds to nearby obstacles in a particular allocentric direction and inhibits the corresponding motor cell, causing the population vector readout of the motor cells to steer away from the obstacle. (C) Perpendicular obstacles have no direction in which to successfully deflect the agent's motion—upon reaching a perpendicular obstacle (first panel), a new vector navigation subgoal must thus be selected. A hippocampal replay event is initiated at the goal location and propagated toward the current location, while concurrently updating the goal vector (second panel; red circle shows location of replay event). When a new viable destination is found, the agent diverts there and performs topological navigation for a while, before eventually resuming vector navigation (third panel; see Fig. 3 for legend).

2.1 Grid Cell Decoding for Vector Navigation

At the core of the model—alongside place and border cells—is a set of grid cells together with a grid cell decoding mechanism. The main output of the network is the allocentric direction in which the agent should move next; this output is primarily driven by the grid cell decoder. The decoding mechanism confers vector navigation capabilities onto the model, by processing inputs from two separate grid cell populations and calculating the vector between the two respective locations represented by those populations. One of the grid cell populations encodes the agent's current location in the two-dimensional plane, whereas the other population encodes the agent's destination. This arrangement is similar to previous work on vector navigation by grid cell decoding (Edvardsen, 2015; Bush et al., 2015), with one crucial extension: Here, the vector navigation destination does not necessarily have to be the same as the agent's ultimate goal location, but can change between different "subgoals" throughout the navigation process (see below).

Note that the model of combined vector and topological navigation proposed below is indifferent to the particular workings of the grid cell decoding mechanism, or indeed to the origin of the grid cell signal itself—we assume only that vector navigation can be performed through the readout of grid cells. Our specific implementation used here builds on the implementation from Edvardsen (2017), where grid cell decoding is performed according to a "nested" view of the grid cell system (Stemmler et al., 2015). Feed-forward decoder neurons receive inputs from the two grid cell populations, and are pre-configured to detect specific patterns of directional offset between the two inputs. A goal vector can be inferred in as few as 1-2 synapses (Edvardsen, 2018), thus even a fast sweep through multiple locations (e.g. within the time frame of a replay event) can continuously update the goal vector. Assuming reasonably fast synaptic integration and neuronal time constants, a post-synaptic neuron could fire within 5-10 ms, which would allow for approximately 10 place fields to be sampled within a replay episode. In addition, replay events (chained together; Davidson et al., 2009) can last longer than 120 ms, potentially covering longer trajectories, suggesting the order of magnitude estimate for the timescales in the model are consistent with published data.

The grid cells are implemented as a set of recurrent neural networks (specifically, continuous attractor networks; Burak and Fiete, 2009), which perform path integration on a self-motion velocity input in order to maintain an updated grid cell representation. Although no noise is explicitly added to this process, some drift may nevertheless occur over time due to imperfect path integration by the grid cell networks. Within the context of the presented simulations the drift is negligible, though future versions of the model could be extended to utilize sensory inputs for error correction in the grid cell system.

2.2 Place Cells Learn a Topological Map

Place cells, particularly in hippocampal area CA3, have been suggested to represent space as a graph structure by virtue of their synaptic interconnections (Muller et al., 1996; Redish and Touretzky, 1998). For simplicity we implemented the place cells directly as nodes in a place graph. New place nodes are created whenever the agent is sufficiently far away from the place field of any previously created place node, i.e. the agent instantaneously memorizes novel locations. Each node takes a snapshot of the grid cell ensemble's current activity, effectively establishing a link between a given place cell and the grid cells for later coordinated replay.

Next, bidirectional links are formed between pairs of place cell nodes whenever the agent moves from the place field of one node to another, corresponding to one-shot Hebbian learning between traversed place cells. The resulting place cell graph reflects the topology of the explored environment, and contains sufficient information to calculate the shortest paths between arbitrary pairs of start and goal place cells across those explored parts; the graph can e.g. determine which of the current immediately adjacent place fields lies on the shortest path to the destination. By always moving toward the neighboring place cell located on that shortest path, the agent would implement a topological navigation behavior. We implemented a graph search algorithm directly using the place graph structure, but we assume that the hippocampal place cell system can support a similar search mechanism e.g. via the resistive network of recurrent synapses (Muller et al., 1996).

2.3 Combining Topological and Vector Navigation

Topological navigation does not require grid cells, as navigation is always directed toward a local place cell. However, combining a topological map with grid cell decoding yields additional navigational capabilities. We assume that an active place cell can trigger the reinstatement of the corresponding grid cell activity for the corresponding location, possibly mediated by projections from CA1/subiculum to medial entorhinal cortex (van Strien et al., 2009; Bush et al., 2014). In the model, each place node is associated with a snapshot of the grid cell activity at the time of the place cell's creation. By associating each visited place field with its unique grid cell activity pattern, any location can become the start or end point for vector navigation. This enables powerful combinations of vector navigation and topological navigation, and can help an agent overcome obstacles by selecting a more suitable place field as its subgoal. We propose that hippocampal replay events can be used to sample possible subgoals among place cell firing fields to allow the agent to change its current destination (see below).

Hence, the model accommodates purely topological, purely vectorial, and combined navigational strategies within the same network architecture. Throughout the navigation process, the relative impact of each strategy is influenced by external factors (bold arrows in Fig. 1C) that may affect the strength of obstacle deflection (see below), trigger new hippocampal replay events or induce periods of random exploration (see Supplementary Materials).

2.4 Negotiating Obstacles via Border Cells

Grid cell-derived goal vectors do not account for any obstacles that might lie in the direct path. To steer the agent clear of such obstacles we employ border cell signals as inputs to a course adjustment mechanism. This obstacle avoidance mechanism can be sufficient to overcome some of the obstacles encountered by the agent, but—due to the local nature of the information conveyed by the border cells—there will inevitably be certain obstacles that are insurmountable to it.

Obstacles that form a slanted angle with the goal vector (less than 90°) can be avoided by deflecting the agent's direction of motion away from the obstacle, all the while remaining on a course that brings it closer to the goal, as the deflected vector still points less than 90° away from the true goal direction (Fig. 2A). Since the goal vector is continuously updated (see description of decoder in Section 2.1), the vector will continue to point to the target as the agent is deflected by obstacles. The agent can follow the deflected vector until either of two events occurs: The obstacle has been cleared, in which case the agent can resume navigating along the true goal vector, or the agent has followed the slope of the obstacle to a point where the border now forms a perpendicular obstacle to the goal vector and the deflection mechanism fails. Note that some obstacles might technically be perpendicular to the goal vector yet be trivially negotiable by small amounts of random exploration, such as a cylindrical shape encountered head-on. We treat these among the slanted obstacles and use the term "perpendicular" throughout to refer to obstacles that remain a problem even after limited random exploration (e.g. a partially concave shape). In these cases the agent selects a new subgoal via replay (see next section).

The deflected movement vector (distinct from the vector derived from grid cell decoder) is calculated by ring networks of "motor cells" (resembling head-direction cells), that combine the true goal direction decoded from grid cells with obstacle information from border cells (Fig. 2B; see also Supplementary Materials). The border cells each respond to obstacles in a particular allocentric direction, with stronger activity as distance decreases, and inhibit corresponding motor cells with the same allocentric tuning direction (see Burgess et al. (1994) for a similar approach). The population vector readout of the motor cell population will then tend to steer away from the obstacle (Fig. 2B, rightmost panel). When the agent is initially far away from an obstacle, the inhibition, and thus the deflection, is barely noticeable. However, because of rapid growth in border cell inhibition, the deflection increases in strength as the obstacle is approached—resulting in a trajectory that gently curves away from the obstacle (Fig. 2A; Supplementary Video 1).

The goal vector is constantly updated to reflect the detour undertaken by the agent during the deflected trajectory. For example, whereas the goal vector originally pointed due East in the example in Fig. 2A, after deflecting to the Northern corner of the obstacle the goal is now located in a Southeasterly direction. Because the grid cell decoder can always recalculate the correct goal direction from any potentially novel location along the deflected trajectory, the agent remains able to find the goal.

2.5 Selecting New Subgoals through Hippocampal Replay Events

Faced with "perpendicular obstacles" (locally perpendicular boundaries where random exploration is unable to trigger further progress), the local obstacle deflection mechanism will fail to find a viable path forward. Motor cells will be equally inhibited on either side of the goal vector, so there is no remaining direction of “least resistance” toward which to deflect the agent, which is now stuck. The agent must then select a new location as the currently active subgoal for the vector navigation process. We propose that this takes place through *hippocampal replay events*, which have been reported to occur when navigating animals stop at choice points or otherwise come to rest during maze sessions. These events are characterized by quick bursts of hippocampal neural activity that appear to play back, or “replay”, traces of earlier place cell activity along paths previously traveled—possibly remote from the animal’s current location (Foster and Wilson, 2006; Ji and Wilson, 2007; Ólafsdóttir et al., 2018). Intriguingly, simultaneously recorded grid cells have been reported to activate in coherence with the replaying place cells (Ólafsdóttir et al., 2016), suggesting that the grid cell population might mirror the replay trajectory by recalling the corresponding spatial locations of the reactivating place cells. As the grid cell decoder can infer a goal vector in 1-2 synapses (Edvardsen, 2018), this suggests that the (sub-)goal vector can follow along on the timescale of the replay events. These would then be replay events where grid cells follow a replay in the place cell population through hippocampal–entorhinal projections (Ólafsdóttir et al., 2016). However, note that temporal coding phenomena may also arise in the grid cell population independently of the hippocampus (Hafting et al., 2008; O’Neill et al., 2017).

Whenever the agent gets stuck, a replay event originating at the goal location and propagating toward the current location could thus be used to find candidates for the new subgoal. The agent initially tries to reach the ultimate goal location, but if the goal vector is blocked by an insurmountable obstacle, the subgoal shifts step by step along a replay trajectory—in our model the shortest path according to the place cell graph (that is, among all previously visited locations)—toward the current location. As soon as a place cell is encountered for which the grid cell-decoded goal vector points sufficiently clear of any obstacles (that is, allowing the motor cells to activate despite inhibition from border cells), the agent resumes moving in that direction. The agent will thus initially prefer to navigate toward locations close to the goal location, attempting to quickly find shortcuts across open space using the grid cells. However, if all of these locations are blocked, the agent will eventually resort to finding its way back via a previously visited place cell close to the current location. Fig. 2C and Supplementary Video 1 show examples of the process during navigation; when the agent reaches the wall, a replay event propagates along the chain of place cells until a feasible subgoal past the obstacle has been found, and the agent then diverts there.

Since the replay events occur across place cells contained in the place graph, the choice of subgoal is restricted to previously visited locations—however, shortcuts can be taken due to the grid cell system. Once a subgoal has been reached, the agent follows a topological navigation strategy for a while, in order to ensure that it escapes the catchment area of the obstacle. Otherwise—if the agent immediately reverted to vector navigation—it might risk running back into the same obstacle. The agent eventually resumes vector navigation, to enable more potential shortcut discovery later in the trial. The duration of this topological navigation phase is governed by a configurable "resetting" probability (Supplementary Materials). Also note that, should a replay event propagate all the way to the current place cell—indicating that none of the place cells activated earlier during the replay were accepted as the new subgoal—we consider the path forward to be blocked. The agent will then unlearn the connection to the most recently activated place cell (the one across the blocked gap), so that the place graph again correctly reflects the topology of the environment.

3 Results

Here we present simulation results from our combined vector/topological navigation model, first demonstrating successful navigation in cluttered environments, next addressing key characteristics of environments where our combined navigation model is particularly well-suited, and finally demonstrating how the model is sufficiently flexible to solve certain experimental mazes from the literature.

3.1 Navigating Cluttered Environments

Fig. 3A shows an example of the kinds of environments employed to test the simulated agent's navigational abilities. Large open spaces are interspersed

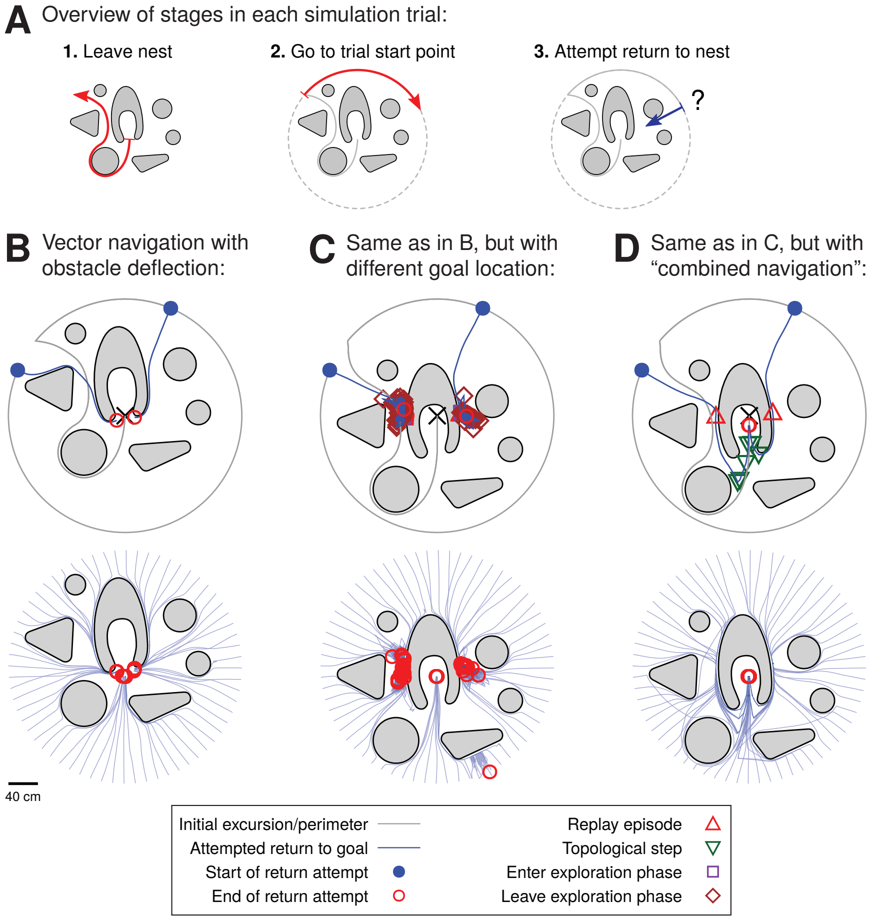

A Overview of stages in each simulation trial:
1. Leave nest
2. Go to trial start point
3. Attempt return to nest
?
B Vector navigation with obstacle deflection:
C Same as in B, but with different goal location:
D Same as in C, but with "combined navigation":
40 cm
Initial excursion/perimeter
Attempted return to goal
Start of return attempt ●
End of return attempt ○
Replay episode △
Topological step ▽
Enter exploration phase □
Leave exploration phase ◇

Figure 3: (A) Overview of trial stages. The agent leaves the nest (black cross in B/C/D) along a pre-defined trajectory for the initial exploration of the environment, performs a partial traversal of a perimeter in order to reach its pre-specified starting position (all trials equally spaced across the full perimeter), and then attempts to return to the nest. (B) Results from 64 trials in an environment with a diverse set of obstacles, using a strategy of vector navigation with border cell-based obstacle deflection. Upper panel shows two example trials from the set of simulations, while lower panel shows all trajectories from the full set of 64 trials, superimposed in the same plot. (C) Results from 64 trials with the same agent strategy as in B, but with the environment modified so that the nest is now located further into the cave-like structure. This creates perpendicular obstacles where the agent gets stuck. (D) Results from 64 trials in the same environment as in C, but now using the combined vector–place strategy that can utilize the topological map to select a new subgoal whenever stuck.

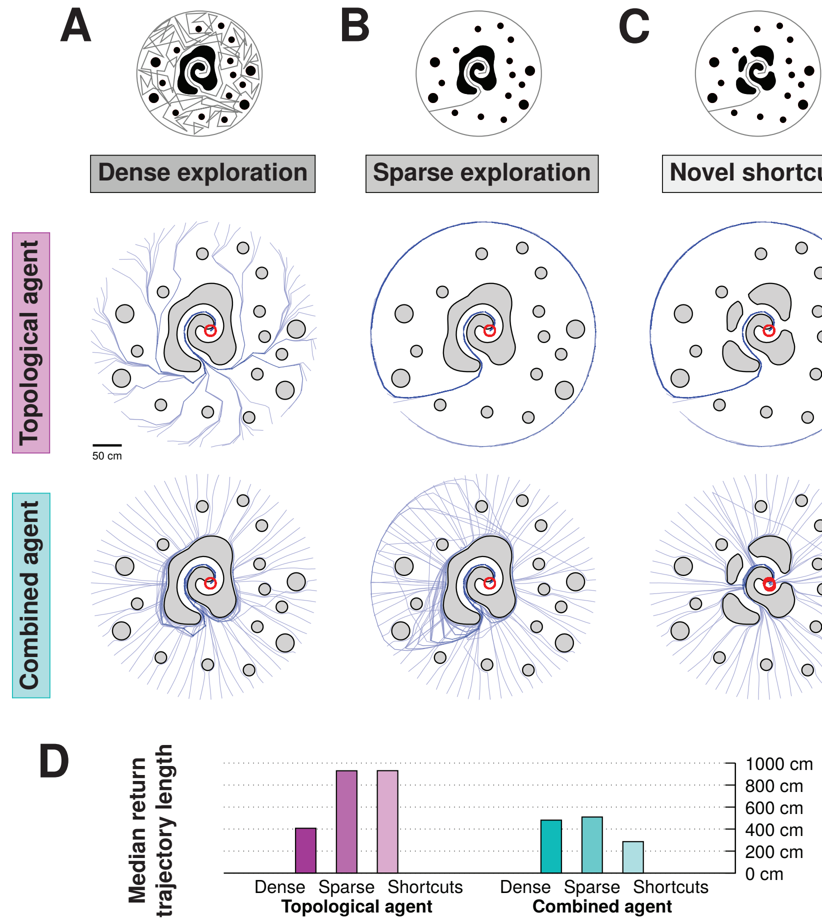


Figure 4: Comparing purely topological and combined vector–place navigation strategies. (A) Results from a densely explored environment. Upper panel shows 64 trials using a purely topological strategy, while lower panel shows 64 trials using the combined vector–place strategy. (B) As in A, but now with only sparse exploration of the environment. (C) As in B, but now with novel shortcuts introduced between the training and test phases. (D) Median return lengths for each set of trials shown in A, B and C.

with obstacles of various shapes centered around a “nest” location in the middle of the arena. Each trial consists of a training phase and a test phase, with the agent initially located at the nest location without any pre-existing knowledge of the environment. The agent first follows a given outbound path from the nest, while updating grid cell and place cell population activity as usual during this initial excursion (performing path integration in the grid cells, generating new place cells in the place cell graph, associating them with the contemporary grid cell population snapshot, and forming links among place cells when moving from one field to the next). It then moves along a perimeter around the environment to a given starting location, before navigating back to the nest. These starting locations are spread out along the perimeter in order to assess the robustness of the agent’s navigational ability under different conditions.

Simulation results from augmenting pure vector navigation with an obstacle deflection mechanism are presented in Fig. 3B. Two example trials show the agent successfully returning to the nest location from two different starting points along the circular perimeter (Fig. 3B, upper panel), and the results from all 64 trials superimposed in one plot shows that the agent is indeed successful in reaching the nest location from all tested starting points (Fig. 3B, lower panel). The obstacle deflection mechanism allows the agent to locally deflect away from obstacles that lie ahead in its vector toward the goal, in this case enabling it to navigate all the way back to the goal. Whether boundary deflection alone is sufficient or not for successful navigation is determined by environmental characteristics; if all encountered obstacles present slanted surfaces (i.e. appear convex to an agent heading toward the goal; Fig. S2A in Supplementary Materials), then this form of navigation will succeed.

An environment with only slanted obstacles will be the exception rather than the norm. Even in the favorable situation discussed above, the situation looks quite different if we move the nest just a short distance into the surrounding “cave” structure. Fig. 3C shows results from a new set of trials with the nest in this changed location. A majority of the trials are no longer successful in reaching the goal, and most of the trials end with the agent stopping at two seemingly unremarkable locations along the outer cave wall. The obstacle forms a perpendicular border against the goal vector in these locations (i.e. appears concave to an agent heading toward the goal; Fig. S2B). When the agent finds itself in these locations, practically all motor cell activity gets canceled by inhibition from border cells in the direction of the goal. In this situation, the agent initiates a brief period of random exploration before resuming attempted navigation towards the goal. However, as the goal distance increases on both sides of the perpendicular point (Fig. S2B), the agent always ends up back at the same obstacle. These trials eventually expired after a timeout of 100 simulated seconds.

To avoid these failures, the agent diverts toward a different subgoal when halted by a perpendicular obstacle. Fig. 3D presents results where the agent employs our proposed “combined” approach of augmenting vector navigation with replay-based selection of a subgoal when stuck. A new subgoal is selected from the place graph by gradually shifting the subgoal closer to the current location until the decoded goal vector is no longer blocked by the obstacle. In the two example trajectories singled out in more detail, red triangular markers indicate the locations in which the agent gets stuck on a perpendicular obstacle and has to initiate a replay episode to find a new subgoal—each trajectory can be seen to continue onwards from the replay location on a diverted course toward a new subgoal. The “combined vector–place navigation” strategy is successful in guiding the agent away from perpendicular obstacles and ultimately to the final goal location, from all tested starting locations. See Supplementary Video 2 for animated examples of trials performed as in Fig. 3.

3.2 Advantages in Sparsely Explored Environments

To compare the combined navigational strategy with a purely topological strategy, we used environments that had been densely explored (Fig. 4A), sparsely explored (Fig. 4B), or had novel shortcuts introduced after the training phase (Fig. 4C). A quantitative comparison of the agent’s navigational behavior across these configurations is depicted in Fig. 4D, showing the median length of the paths needed to return to the goal location across the 64 trials for each unique configuration.

Dense versus sparse exploration refers to how many different parts of the environment the agent had visited before the navigation trial. For dense exploration, the agent’s pre-programmed exploration trajectory was drawn to cover close to the full environment with place fields (Fig. 4A, top row), in order to facilitate a place graph with more nodes and connections. To quantify the difference in strategies we considered close to full exploration of the environment versus only one outbound trajectory. However, there is no fixed amount of "necessary" exploration for the model to work. The proposed mechanism for selecting subgoals can be engaged with any amount of exploration, though subgoals must necessarily coincide with a previously visited location.

When the environment is densely explored, the purely topological navigation strategy is able to immediately follow the shortest paths around obstacles, without first having to run into them (Fig. 4A). However, because the agent only navigates according to its learnt place cell graph, sparse knowledge of the environment results in suboptimal paths (Fig. 4B), and it does not utilize any of the novel shortcuts introduced to the environment after the training phase (Fig. 4C). The combined vector–place agent performs less well than the topological agent in the densely explored environment (Fig. 4A), but performs better in the sparsely explored scenario (Fig. 4B). The grid cell-provided vector navigation capability enables the agent to shortcut across the open space not initially explored, and also to discover the novel shortcuts introduced after the training phase (Fig. 4C). Because the combined agent's behavior is mostly driven by vector navigation until the later stages of the navigation trials, there is not much difference in performance between the densely and sparsely explored situations in this condition. See Supplementary Video 3 for animated examples of trials performed as in Fig. 4.

3.3 Flexibility to Solve a Variety of Experimental Mazes

We tested the navigation model in experimental environments from the animal navigation literature. Although relatively simple, we found that the flexible architecture underlying the navigation model made it possible to solve certain experimental tasks with minimal peripheral changes to the agent.

An early inspiration for development of the cognitive map theory was the *sunburst maze* (Tolman et al., 1946), in which rats deduced the correct corridor toward a goal location despite major modifications to the environment between the initial training sessions and the test session. Specifically, the circuitous outbound corridor available during training (the gray line showing the agent's initial excursion in Fig. 5A) was removed before the test session and replaced with a set of novel radial arms. Tolman et al. (1946) reported that 19 of 56 rats eventually chose the arm pointing directly toward the food location (arm 6), and hence demonstrated an ability to calculate the correct shortcut toward the goal. Though the methodology of this specific experiment has been challenged and their reported results have been difficult to reproduce (Gentry et al., 1947; Bennett, 1996; Grieves and Dudchenko, 2013), we nevertheless wanted to see if the navigation abilities claimed of the rats in this study were realizable within the framework of our navigation model.

To let our agent avoid the incorrect arms in the sunburst maze, we tune the motor cells in the boundary deflection mechanism to have a narrower width of their tuning curves. This causes the agent to abort its vector navigation-driven goal approach whenever the current goal vector is misaligned with the angle of the current corridor, i.e. whenever following the vector would make the agent run into corridor walls. The agent will then turn around to perform a period of random exploration, followed by another attempt to vector navigate toward the goal. Eventually, the agent might find itself in a favorable starting location where it has a clear path toward the goal location—the vector navigation process can then succeed in guiding the agent down the correct corridor without getting interrupted by the obstacle avoidance mechanism. This process of alternating between random exploration and vector navigation is visible in an example of a successful trial in Fig. 5A (note that the rats in Tolman et al. (1946) on average used around three and a half minutes to select an arm). Superimposed results from the full set of 64 trials is shown in Fig. 5B. Trials were terminated after a timeout of 100 simulated seconds or when the agents ventured a certain distance down a corridor (approximately corresponding to the length of the short arms; Tolman et al. (1946) also allowed rats to explore the initial segments of the arms). The majority of trials ended with the agent choosing the correct corridor (Fig. 5C).

While the sunburst maze lends itself to a vector navigation-based solution, an environment that might instead favor topological navigation is the detour maze (Tolman and Honzik, 1930; Alvernhe et al., 2011). This maze consists of a direct corridor between the start location and the goal location, as well as two detour corridors that branch off near the starting location—a short detour and a long detour. The direct corridor can be blocked in one of two locations, so that either both detours can reach the goal, or only the long detour can reach the goal. A cognitive map should enable the animal to choose the shortest possible detour, depending on the location at which the novel obstacle is encountered. Simulations are shown in Fig. 5D–G. When the agent encounters the novel obstacle, the place cell connection across the obstacle is severed and the chain of place cells linking the current location and the goal location down the main corridor is thus interrupted. The new shortest path in the place graph, along which the next replay event will propagate, then guides the agent toward the correct corridor in both the short detour scenario (Fig. 5E) and the long detour scenario (Fig. 5G). See Supplementary Video 4 for animated examples of trials performed as in Fig. 5.

4 Discussion

We have presented a hippocampal navigation model that is able to navigate in cluttered environments by utilizing a combination of grid cell-driven vector navigation, place cell-driven topological navigation and border cell-driven local obstacle avoidance. The proposed architecture, which maps well onto known anatomy and electrophysiology of the hippocampal formation, can support a diverse range of navigational strategies by allowing external modulation of network components to produce different navigational behaviors. The agent initially performs vector navigation, primarily driven by grid cells and aided by border cells for obstacle deflection. If progress is blocked by obstacles the agent initiates hippocampal replay that introduces aspects of topological navigation into the agent's overall behavior, allowing the agent to switch between different subgoals in order to successfully navigate complex environments. Our results demonstrate that grid cell decoders (Edvardsen, 2015; Bush et al., 2015; Stemmler et al., 2015) can be the primary driver of navigational processes even beyond the open-field environment, because such vector navigation mechanisms can indeed work in cluttered environments when aided by place cells and border cells to negotiate obstacles.

Such a combined navigational strategy can be particularly useful in large, under-explored environments (which applies to most natural, open environments), where the agent would otherwise have to resort to a topological strategy of navigating by retracing its original steps back to the origin. The grid cell-based vector navigation process can instead guide the agent across novel

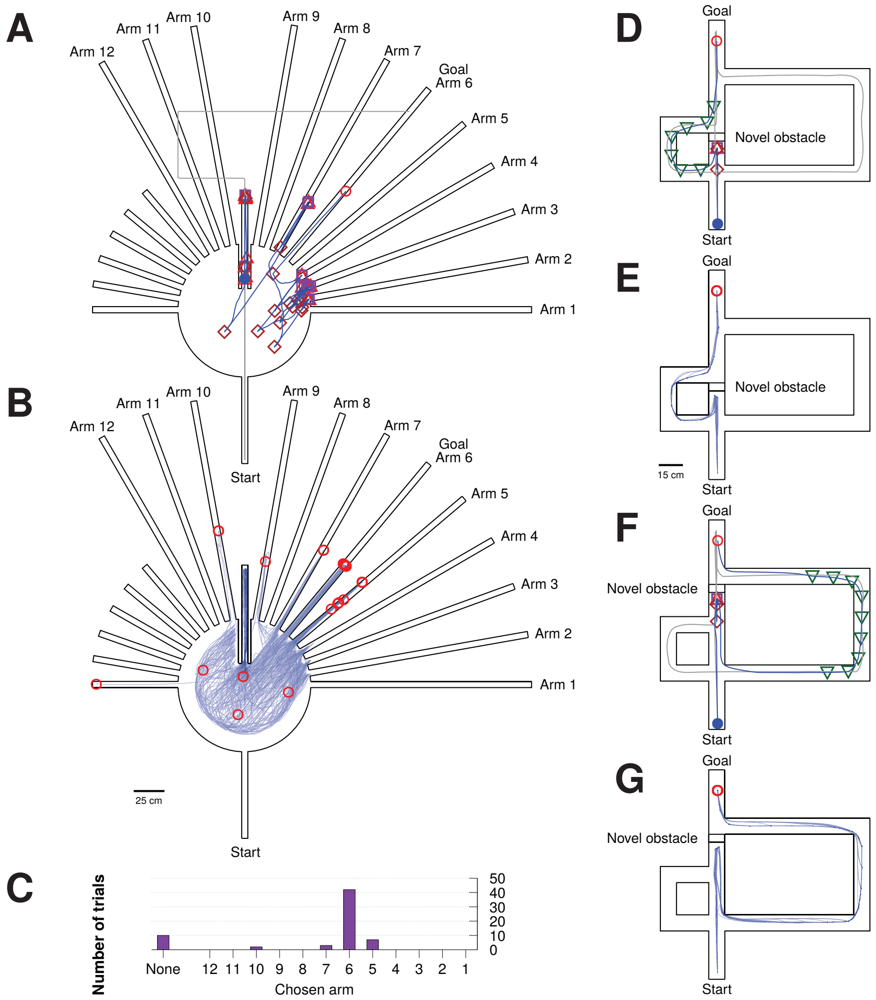


Figure 5: (A) A single example trial from a set of 64 trials in the sunburst maze, showing alternation between vector navigation attempts and random exploration that ultimately succeeds in finding the correct corridor. (B) Superimposed trajectories from the full set of trials in the sunburst maze. (C) Sunburst maze trials accumulated by outcome. (D) A single example trial from a set of 32 trials in the detour maze with the short corridor as the correct choice. (E) Superimposed trajectories from the full set of trials in the short version of the detour maze. (F) A single example trial from a set of 32 trials in the detour maze with the long corridor as the correct choice. (G) Superimposed trajectories from the full set of trials in the long version of the detour maze.

territory, eventually resorting to a place cell-driven topological navigation process should vector navigation turn out to be impossible. Besides potentially enabling more efficient navigation in under-explored scenarios, the combined strategy is useful for discovering pre-existing or novel shortcuts in the environment (Fig. 4). The agent is a “pragmatist”, trying out the fastest route first (a straight line). The hybrid aspect of the model (interacting with place cells for subgoal selection via replay events) is only engaged when it gets stuck. In experimental environments animals often bias their exploration trajectories differently (Kubie and Fenton, 2009), e.g. spending more time near walls, which could affect navigational performance, e.g. restricting the availability of subgoals. However, to the best of our knowledge the tendency of experimental animals to spend more time near the walls is likely due to perceived safety and not navigational considerations. In larger, open and under-explored environments the pragmatic approach proposed here may be a simple, non-demanding, yet effective strategy for shortcut discovery and quick return to the nest. Future work should systematically investigate simplicity (i.e. pragmatism) versus optimality trade-offs, which, however, will depend heavily on the structure of the environments used to assess optimality.

The flexibility of our proposed architecture is demonstrated by the model’s performance in two examples of experimental maze environments from the animal navigation literature, namely the sunburst maze and the detour maze. Interestingly, we found that—while both types of maze might be cited as examples of animals expressing cognitive map-based navigation capabilities—these two environments primarily exercised complementary parts of our navigation model. That is, the nature of these mazes is such that mostly only the vector navigation capacity or the topological navigation capacity of the model is utilized. Specifically, for the sunburst maze, only vector navigation with obstacle avoidance and random exploration was needed—the topological map was not used. On the other hand, in the detour maze, vector navigation would not strictly be necessary—the maze layout is fully known by the animal in advance, so navigating only according to the topological map should be sufficient (Martinet et al., 2011). If the cognitive map is considered to consist of both the topological aspects of place cells and the metric aspects of grid cells, then experimental environments should ideally be designed to engage the aspects of the navigational circuit intended to be probed by the experimenter.

Besides this more general conclusion about navigational strategies and experimental environments, the model also makes certain predictions about the nature of hippocampal replay events. Replay has been suggested to be involved in both planning and consolidation (Wilson and McNaughton, 1994; Foster and Wilson, 2006; Diba and Buzsáki, 2007; Karlsson and Frank, 2009; Girardeau et al., 2009; Carr et al., 2011). Here we considered a potential role for replay in the planning of a trajectory (Pfeiffer and Foster, 2013). These events are triggered in the model whenever the agent gets stuck during the vector navigation process—we hence predict a higher propensity for replays to occur when the agent encounters obstacles. Replays should be coherent between place cells and grid cells (Ólafsdóttir et al., 2016). Whenever the agent diverts its course, the new destination should be in the vicinity of a recent replay, and there should be a goal vector representation (Sarel et al., 2017) for this subgoal. The agent might follow different bearings across the same open field, depending on whether it encountered any obstacles earlier in the trial that caused it to change subgoals. In general, the model suggests that the analysis of behavioral and neurophysiological data might benefit from taking into account the location of obstacles and likely subgoals as relevant variables, not just the animal's own location and its ultimate destination. However, there seems to be a diversity of different forward/reverse replay/preplay phenomena (for review, see Ólafsdóttir et al., 2018), and the model proposed here only considers one type of replay.

The integration of grid cells and place cells in the same architecture for navigational purposes has been proposed before. Erdem and Hasselmo (2012, 2014) present a model for grid cell-based vector navigation that depends on place cells as a critical component of the system. The model relies on the grid cells "simulating" hypothetical forward trajectories in different directions in order to trigger the activation of the target place cell and thus to have detected the correct goal direction. That is, it exploits projections from grid cells onto place cells to determine the goal direction, whereas the grid cell decoder in our model produces its goal vector through direct readout of the grid cell population (Edvardsen, 2015; Bush et al., 2015; Stemmler et al., 2015). Whereas Erdem and Hasselmo (2012, 2014) perform subgoal selection by diffusing a reward signal throughout the topological graph of place cells and navigating in the direction of the place cell most strongly activated by a simulated forward trajectory, we propose that hippocampal replay events might interact with a grid cell decoder for the same purpose.

In reinforcement learning parlance, both aspects of the current model fall into the category of model-based approaches to navigation. A full model of rodent navigation should, in addition, contain interactions with (e.g. striatal) reinforcement learning mechanisms (Chavarriaga et al., 2005; Dollé et al., 2010; Chersi and Burgess, 2015), and/or possibly a mechanism akin to the successor representation (Dayan, 1993; Stachenfeld et al., 2017; Gershman, 2018). A successor representation is comparable to the topological place cell representation used in the current model, and replay events could similarly propagate through it, selecting new subgoals. The recent theoretical framework where grid cells form a low-dimensional state representation (obtained via an eigenvector decomposition of place cells; Dordek et al., 2016; Stachenfeld et al., 2017), can in principle identify bottleneck states in environments (e.g. doorways), though it is currently unclear how this computation would be carried at the level of neurons. Such states could also constitute interesting subgoals, but the eigenvector decomposition requires a thorough exploration of the environment, contrary to the present model. With regards to spatial navigation, other strategies such as taxis and landmark-based navigation (Trullier et al., 1997) are also known to guide an animal's behavior, and should be incorporated for a more complete navigation model. Finally, look-ahead and mental navigation could also interact with the combined vector–place strategy proposed here (Erdem and Hasselmo, 2012, 2014; Bicanski and Burgess, 2018). In mental navigation, simulated motion (potentially driven by mock motor efference and conveyed by grid cells; Bellmund et al., 2016; Horner et al., 2016) can be thought of as accompanied by a reinstatement of sensory representations bound to locations (via place cells) along the imagined trajectory (Bicanski and Burgess, 2018), and would hence be particularly useful if planning involves particular sensory aspects along the route.

In conclusion, we have shown that grid cells can potentially be used to drive navigation and shortcut discovery even in cluttered environments, if aided by place cells and border cells. Realistic navigational strategies in cluttered, large and under-explored environments will likely utilize combinations of both vector navigation and topological navigation. Environments commonly used in animal navigation research may not exercise both of these systems at the same time. Designing experiments with under-explored parts of the environment could shed more light on the interplay between vector and topological strategies in animal behavior and lead to new insights in the role of grid cells and place cells in navigation. A flexible navigation system with a plausible neural implementation might also be of interest to the field of biologically inspired robotics, to enable robots to navigate according to biologically inspired principles in cluttered, under-explored environments.

Acknowledgements

AB and NB acknowledge funding by the ERC Advanced grant NEUROMEM, the Wellcome Trust and the European Union's Horizon 2020 research and innovation programme under grant agreement No. 785907 Human Brain Project SGA2. The simulations were performed on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway. We thank Daniel Bush and Caswell Barry for useful discussions, and Daniel Bush also for feedback on the manuscript. The authors declare no competing interests.

References

Alvernhe, A., Save, E., and Poucet, B. (2011). Local remapping of place cell firing in the tolman detour task. *European Journal of Neuroscience*, 33(9):1696–1705.

Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O'Keefe, J., Jeffery, K., and Burgess, N. (2006). The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, 17(1-2):71–98.

Bellmund, J. L., Deuker, L., Schröder, T. N., and Doeller, C. F. (2016). Grid-cell representations in mental simulation. *eLife*, 5:e17089.

Bennett, A. T. (1996). Do animals have cognitive maps? *Journal of Experimental Biology*, 199(1):219–224.

Bicanski, A. and Burgess, N. (2018). A neural-level model of spatial memory and imagery. *eLife*, 7:e33752.

Blum, K. I. and Abbott, L. F. (1996). A model of spatial map formation in the hippocampus of the rat. *Neural computation*, 8(1):85–93.

Burak, Y. and Fiete, I. R. (2009). Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, 5(2):e1000291.

Burgess, N., Becker, S., King, J. A., and O'Keefe, J. (2001). Memory for events and their spatial context: models and experiments. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 356(1413):1493–1503.

Burgess, N., Recce, M., and O'Keefe, J. (1994). A model of hippocampal function. *Neural networks*, 7(6-7):1065–1081.

Bush, D., Barry, C., and Burgess, N. (2014). What do grid cells contribute to place cell firing? *Trends in neurosciences*, 37(3):136–145.

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

Byrne, P., Becker, S., and Burgess, N. (2007). Remembering the past and imagining the future: a neural model of spatial memory and imagery. *Psychological review*, 114(2):340.

Carpenter, F., Manson, D., Jeffery, K., Burgess, N., and Barry, C. (2015). Grid cells form a global representation of connected environments. *Current Biology*, 25(9):1176–1182.

Carr, M. F., Jadhav, S. P., and Frank, L. M. (2011). Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature neuroscience*, 14(2):147.

Chavarriaga, R., Strösslin, T., Sheynikhovich, D., and Gerstner, W. (2005). A computational model of parallel navigation systems in rodents. *Neuroinformatics*, 3(3):223–241.

Chersi, F. and Burgess, N. (2015). The cognitive architecture of spatial navigation: hippocampal and striatal contributions. *Neuron*, 88(1):64–77.

Davidson, T. J., Kloosterman, F., and Wilson, M. A. (2009). Hippocampal replay of extended experience. *Neuron*, 63(4):497–507.

Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624.

Diba, K. and Buzsáki, G. (2007). Forward and reverse hippocampal place-cell sequences during ripples. *Nature neuroscience*, 10(10):1241.

Dollé, L., Sheynikhovich, D., Girard, B., Chavarriaga, R., and Guillot, A. (2010). Path planning versus cue responding: a bio-inspired model of switching between navigation strategies. *Biological cybernetics*, 103(4):299–317.

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

Edvardsen, V. (2015). A passive mechanism for goal-directed navigation using grid cells. In *Proceedings of the 2015 European Conference on Artificial Life (ECAL)*, pages 191–198.

Edvardsen, V. (2017). Long-range navigation by path integration and decoding of grid cells in a neural network. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4348–4355. IEEE.

Edvardsen, V. (2018). Navigating with distorted grid cells. In *Artificial Life Conference Proceedings*, pages 260–267. MIT Press.

Erdem, U. M. and Hasselmo, M. (2012). A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *European Journal of Neuroscience*, 35(6):916–931.

Erdem, U. M. and Hasselmo, M. E. (2014). A biologically inspired hierarchical goal directed navigation model. *Journal of Physiology-Paris*, 108(1):28–37.

Etienne, A. S., Maurer, R., and Séguinot, V. (1996). Path integration in mammals and its interaction with visual landmarks. *Journal of Experimental Biology*, 199(1):201–209.

Fiete, I. R., Burak, Y., and Brookings, T. (2008). What grid cells convey about rat location. *Journal of Neuroscience*, 28(27):6858–6871.

Foster, D. J. and Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680.

Gentry, G., Brown, W. L., and Kaplan, S. J. (1947). An experimental analysis of the spatial location hypothesis in learning. *Journal of Comparative and Physiological Psychology*, 40(5):309.

Gershman, S. J. (2018). The successor representation: its computational logic and neural substrates. *Journal of Neuroscience*, 38(33):7193–7200.

Gillner, S. and Mallot, H. A. (1998). Navigation and acquisition of spatial knowledge in a virtual maze. *Journal of Cognitive Neuroscience*, 10(4):445–463.

Girardeau, G., Benchenane, K., Wiener, S. I., Buzsáki, G., and Zugaro, M. B. (2009). Selective suppression of hippocampal ripples impairs spatial memory. *Nature neuroscience*, 12(10):1222.

Grieves, R. M. and Dudchenko, P. A. (2013). Cognitive maps and spatial inference in animals: Rats fail to take a novel shortcut, but can take a previously experienced one. *Learning and Motivation*, 44(2):81–92.

Hafting, T., Fyhn, M., Bonnevie, T., Moser, M. B., and Moser, E. I. (2008). Hippocampus-independent phase precession in entorhinal grid cells. *Nature*, 453(7199):1248.

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

Hardcastle, K., Ganguli, S., and Giocomo, L. M. (2015). Environmental boundaries as an error correction mechanism for grid cells. *Neuron*, 86(3):827–839.

Horner, A. J., Bisby, J. A., Zotow, E., Bush, D., and Burgess, N. (2016). Grid-like processing of imagined navigation. *Current Biology*, 26(6):842–847.

Ji, D. and Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100.

Karlsson, M. P. and Frank, L. M. (2009). Awake replay of remote experiences in the hippocampus. *Nature neuroscience*, 12(7):913.

Kropff, E. and Treves, A. (2008). The emergence of grid cells: Intelligent design or just adaptation? *Hippocampus*, 18(12):1256–1269.

Kubie, J. L. and Fenton, A. A. (2009). Heading-vector navigation based on head-direction cells and path integration. *Hippocampus*, 19(5):456–479.

Kubie, J. L. and Fenton, A. A. (2012). Linear look-ahead in conjunctive cells: an entorhinal mechanism for vector-based navigation. *Frontiers in neural circuits*, 6:20.

Lever, C., Burton, S., Jeewajee, A., O'Keefe, J., and Burgess, N. (2009). Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31):9771–9777.

Martinet, L. E., Sheynikhovich, D., Benchenane, K., and Arleo, A. (2011). Spatial learning and action planning in a prefrontal cortical network model. *PLoS computational biology*, 7(5):e1002045.

Mataric, M. J. (1991). Navigating with a rat brain: a neurobiologically inspired model. In *From Animals to Animats; Proceedings of the First International Conference on Simulation of Adaptive Behavior*. MIT Press, Cambridge, Mass.

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

Mittelstaedt, M. L. and Mittelstaedt, H. (1980). Homing by path integration in a mammal. *Naturwissenschaften*, 67(11):566–567.

Mulas, M., Waniek, N., and Conradt, J. (2016). Hebbian plasticity realigns grid cell activity with external sensory cues in continuous attractor models. *Frontiers in computational neuroscience*, 10:13.

Muller, R. U., Stead, M., and Pach, J. (1996). The hippocampus as a cognitive graph. *The Journal of general physiology*, 107(6):663–694.

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

O'Keefe, J. and Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford: Clarendon Press.

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

Ólafsdóttir, H. F., Carpenter, F., and Barry, C. (2016). Coordinated grid and place cell replay during rest. *Nature neuroscience*, 19(6):792–794.

O'Neill, J., Boccara, C. N., Stella, F., Schoenenberger, P., and Csicsvari, J. (2017). Superficial layers of the medial entorhinal cortex replay independently of the hippocampus. *Science*, 355(6321):184–188.

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

Pouget, A. and Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of cognitive neuroscience*, 9(2):222–237.

Redish, A. D. and Touretzky, D. S. (1998). The role of the hippocampus in solving the morris water maze. *Neural computation*, 10(1):73–111.

Rolls, E. T., Stringer, S. M., and Elliot, T. (2006). Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. *Network: Computation in Neural Systems*, 17(4):447–465.

Sarel, A., Finkelstein, A., Las, L., and Ulanovsky, N. (2017). Vectorial representation of spatial goals in the hippocampus of bats. *Science*, 355(6321):176–180.

Snyder, L. H., Grieve, K. L., Brotchie, P., and Andersen, R. A. (1998). Separate body- and world-referenced representations of visual space in parietal cortex. *Nature*, 394(6696):887.

Solstad, T., Boccara, C. N., Kropff, E., Moser, M. B., and Moser, E. I. (2008). Representation of geometric borders in the entorhinal cortex. *Science*, 322(5909):1865–1868.

Solstad, T., Moser, E. I., and Einevoll, G. T. (2006). From grid cells to place cells: a mathematical model. *Hippocampus*, 16(12):1026–1031.

Stachenfeld, K. L., Botvinick, M. M., and Gershman, S. J. (2017). The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643.

Stemmler, M., Mathis, A., and Herz, A. V. (2015). Connecting multiple spatial scales to decode the population activity of grid cells. *Science Advances*, 1(11):e1500816.

Taube, J. S., Muller, R. U., and Ranck, J. B. (1990a). Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435.

Taube, J. S., Muller, R. U., and Ranck, J. B. (1990b). Head-direction cells recorded from the postsubiculum in freely moving rats. ii. effects of environmental manipulations. *Journal of Neuroscience*, 10(2):436–447.

Tolman, E. C. and Honzik, C. H. (1930). "insight" in rats. *University of California Publications in Psychology*, 4(14):215–232.

Tolman, E. C., Ritchie, B. F., and Kalish, D. (1946). Studies in spatial learning. i. orientation and the short-cut. *Journal of Experimental Psychology*, 36(1):13–24.

Trullier, O., Wiener, S. I., Berthoz, A., and Meyer, J. A. (1997). Biologically based artificial navigation systems: Review and prospects. *Progress in neurobiology*, 51(5):483–544.

van Strien, N. M., Cappaert, N. L. M., and Witter, M. P. (2009). The anatomy of memory: an interactive overview of the parahippocampal–hippocampal network. *Nature Reviews Neuroscience*, 10(4):272.

Wilson, M. A. and McNaughton, B. L. (1994). Reactivation of hippocampal ensemble memories during sleep. *Science*, 265(5172):676–679.

Supplementary Materials – Edvardsen et al. (2019)

Navigating with grid and place cells in cluttered environments

1 Detailed model description

Fig. S1 shows a more detailed version of the navigation model's schematic from the main paper, now including the specific modulatory signals used by the agent controller to orchestrate network behavior. The model executes 1000 timesteps per simulated second. The agent moves with a fixed simulated speed of 20 cm/s (unless halted, see Sec. 1.5). On each timestep, the model components update in the following order: (1) Model inputs, including border cells, (2) modulatory signals, (3) grid modules, (4) place graph, and (5) motor output, labeled "course adjustment" in Fig. S1. Each step is described in more detail next.

1.1 Model inputs

In this step, besides updating velocity and x, y position inputs to the model, information about nearby obstacles is conveyed as follows: There are 72 border cells, each responsible for a particular allocentric direction equally spaced across 360°. For each border cell, a beam emanating from the agent's current location and extending 0.25 m in the cell's specified direction is tested for intersections with obstacles in the environment. If there were no intersections, the border cell gains an activation value of 0, otherwise the closest intersection

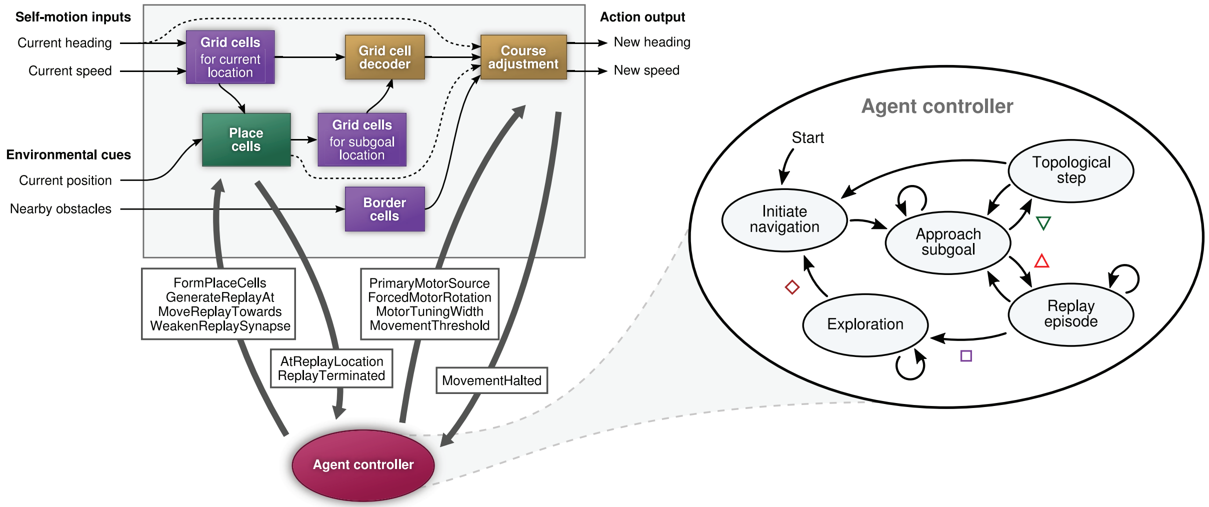


Figure S1: Detailed model schematic, including the modulatory signals used by the agent controller to orchestrate network behavior. The internal logic of the agent controller is organized as a "state machine", with distinct states and transitions between them. Symbols next to the state transition arrows correspond to the symbols used in the figures in the main paper. Dashed inputs to the course adjustment component indicate alternative sources for motor direction, see Sec. 1.5 for details.

Table S1: Pseudocode specification of the internal logic of the agent controller.

Default values reset before each timestep:	**FormPlaceCells** ← ρ **GenerateReplayAt** ← $\emptyset$ **MoveReplayTowards** ← $\emptyset$ **WeakenReplaySynapse** ← False **PrimaryMotorSource** ← GridCellDecoder **ForcedMotorRotation** ← 0 **MotorTuningWidth** ← $\delta_{approach}$ **MovementThreshold** ← $\eta_{approach}$
Agent logic for state **InitiateNavigation**:	If ϕ is False, then **GenerateReplayAt** ← GoalNode, else $\begin{cases} \textbf{GenerateReplayAt} \leftarrow \text{AgentNode} \\ \textbf{MoveReplayTowards} \leftarrow \text{GoalNode.} \end{cases}$ **NextState** ← ApproachSubgoal

Agent logic for state **ApproachSubgoal**:	If **AtReplayLocation**, then **NextState** ← TopologicalStep,
	else if **MovementHalted**, then $\begin{cases} \textbf{MovementThreshold} \leftarrow \eta_{replay} \\ \textbf{NextState} \leftarrow \text{ReplayEpisode,} \end{cases}$
	else **NextState** ← ApproachSubgoal.

Agent logic for state **TopologicalStep**:	**GenerateReplayAt** ← AgentNode **MoveReplayTowards** ← GoalNode If $\mathrm{U}(0,1) < \varepsilon_{reset}$, then **NextState** ← InitiateNavigation, else **NextState** ← ApproachSubgoal.

Agent logic for state **ReplayEpisode**:	If not **MovementHalted**, then **NextState** ← ApproachSubgoal,
	else if **ReplayTerminated**, then $\begin{cases} \textbf{WeakenReplaySynapse} \leftarrow \text{True} \\ \textbf{PrimaryMotorSource} \leftarrow \text{LastHeading} \\ \textbf{ForcedMotorRotation} \leftarrow \pi \\ \textbf{NextState} \leftarrow \text{Exploration,} \end{cases}$
	else $\begin{cases} \textbf{MotorTuningWidth} \leftarrow \delta_{replay} \\ \textbf{MovementThreshold} \leftarrow \eta_{replay} \\ \textbf{MoveReplayTowards} \leftarrow \text{AgentNode} \\ \textbf{NextState} \leftarrow \text{ReplayEpisode.} \end{cases}$

Agent logic for state **Exploration**:	**PrimaryMotorSource** ← LastHeading **MotorTuningWidth** ← $\delta_{exploration}$ **ForcedMotorRotation** ← $\mathrm{N}(\mu = 0,\ \sigma = 0.02)$ If $\mathrm{U}(0,1) < \varepsilon_{exploration}$, then **NextState** ← InitiateNavigation, else **NextState** ← Exploration.

Table S2: Description of parameters and their default values.

Parameter	Default	Description
ρ	True	Whether to form new place cells
ϕ	False	Whether to navigate topologically
$\varepsilon_{exploration}$	0.003	Probability of exploration ending
ε_{reset}	0.05	Probability of subgoal resetting
$\delta_{approach}$	0.75	Motor tuning for vector navigation[1]
δ_{replay}	0.1	Motor tuning for replay episodes[1]
$\delta_{exploration}$	0.1	Motor tuning for exploration periods[1]
$\eta_{approach}$	0.05	Halt threshold for vector navigation[2]
η_{replay}	0.2	Halt threshold for replay episodes[2]

[1] Tuning width in motor cells; larger values give more lenient obstacle deflection, small values require clearer line of sight.

[2] Threshold for confidence value c, below which agent will halt.

Table S3: Different versions of the agent used.

Vector-navigating agent with deflection *(Fig. 3BC)*
ρ = False, rest default

Purely topological agent *(Fig. 4ABC upper row)*
ϕ = True, rest default

Combined agent *(Fig. 3D, 4ABC lower row, 5DEFG)*
All parameters set to defaults

Combined agent, sunburst version *(Fig. 5AB)*
$\varepsilon_{exploration} = 0.0005$, $\delta_{approach} = 0.1$, rest default

Combined agent, exaggerated traits *(Fig. 2AC)*
$\varepsilon_{reset} = 0.25$, $\eta_{replay} = 0.9$, rest default

distance d (in meters) is mapped through $2 \cdot e^{-5 \cdot d/0.25}$ to get the final activation value for the border cell. In our implementation we used the Boost.Geometry library (`https://www.boost.org/`) to detect intersections.

1.2 Modulatory signals

The agent controller—in order to orchestrate the behavior of components in the hippocampal formation network—decides appropriate values for the modulatory inputs that flow from the controller into the network. The schematic in Fig. S1 shows that the internal logic of the agent controller is represented by a “state machine”, consisting of different states/phases of the navigational process and possible transitions between them. The internal logic of these states needs to monitor the agent’s progress in order to make decisions—this information is mediated by the modulatory outputs that flow from the network to the controller.

On each timestep, all modulatory inputs are first reset to their default values according to Table S1 upper row. The agent’s current state—determined by the value of NEXTSTATE, which starts out as “InitiateNavigation”—is then used to calculate the present values for the modulatory inputs according to the corresponding state definition in Table S1. $\mathrm{U}(0,1)$ samples a uniformly random value between 0 and 1, and $\mathrm{N}(\mu,\sigma)$ samples from a Gaussian distribution. Modulatory outputs used in these calculations retain their values from the end of the previous timestep, and the specific meaning of all the different modulatory input/output signals are described in more detail in their relevant sections below.

Parametrization of the state logic was used to achieve the different agent versions used throughout the paper—see Table S2 for a specification of these parameters and their default values, and Table S3 for adaptations relevant to the specific agent versions.

1.3 Grid modules

The grid cell system, as well as the grid cell decoder, builds on our implementation from earlier work (Edvardsen, 2017). The system consists of 12 grid modules, each implemented as a continuous attractor network based on the model by Burak and Fiete (2009). The smallest grid module exhibits a grid scale of around 0.2 m, and each succeeding grid module scales up by a fixed factor of 1.5 from the previous one. This geometric progression of grid modules is then decoded according to a nested view of the grid cell system (Edvardsen, 2015; Stemmler et al., 2015). However, our model does not depend on any particular aspects of the grid cell system/decoder besides the following:

- There is a mechanism that can generate/maintain grid cell activity to represent the current location/spatial coordinates
- Whenever a new place cell is generated, that cell may take a "snapshot" of the current state of the grid cell population
- The grid cell decoder is able to calculate the approximate direction from the current location to a previously visited location, whenever it is presented with one of these earlier snapshots of the grid cell population (provided by the place cell system)

The reader is refered to Edvardsen (2015, 2017) for a full description.

1.4 Place graph

The place graph updates according to the following steps:

1. Retrieve the place cell closest to the agent's current x, y coordinates as the *visited place cell*
2. If the center of the visited place cell is located farther away than twice the place field radius, and FORMPLACECELLS is true, then create a new place cell at the current coordinates and use that as the visited place cell instead. Take a snapshot of the current grid cell population and store it alongside the newly created place cell
3. If the visited place cell is different from last timestep's visited place cell, and these two cells are not already connected, then form a connection between the two cells with a strength of 2

4. If WEAKENREPLAYSYNAPSE is true, then decrease the connection strength between the *last replaying place cell* and the *second-to-last replaying place cell*, and delete the connection altogether if the connection strength reached 0
5. If GENERATEREPLAYAT is different from ∅, then update the *replay place cell* to be either the visited place cell or the goal place cell, depending on whether the value of GENERATEREPLAYAT is respectively "AgentNode" or "GoalNode"
6. If MOVEREPLAYTOWARDS is different from ∅:
    - Update the replay place cell to be the place cell one step closer in the graph to either the visited place cell or the goal place cell, depending on whether the value of MOVEREPLAYTOWARDS is respectively "AgentNode" or "GoalNode", according to Breadth-First Search
    - If the search failed to find a path in the graph, or if the replay place cell was already at the indicated target node for the graph search, then set the output signal REPLAYTERMINATED to true on the next timestep
7. If the agent's distance to the center of the replay place cell is no greater than the place field radius, then set the output signal ATREPLAYLOCATION to true on the next timestep
8. Send the grid cell snapshot associated with the replay place cell to the grid cell decoder as the "target" for vector navigation

The place field radius was set to 13 cm in Figs. 3 and 4, and 6.5 cm in Figs. 2 and 5.

1.5 Motor output

There are two consecutive *motor networks*. Each motor network consists of two stages:

1. **Generating motor activity:** Receive a directional input α and then generate a Gaussian bump of activity in a ring of 72 motor neurons, pointing in the given direction (with tuning width/standard deviation δ for the Gaussian function, and afterwards rescaling all activation values so the peak activation value in the network is 1)
2. **Inhibiting motor cells from border cells:** Target each motor neuron with inhibition from the corresponding border cell, and then threshold negative activation values to 0. The final directional output of the motor network is calculated from the population vector average of the 72 motor neurons

The tuning width δ_1 for the first motor network is set to MOTORTUNINGWIDTH. The directional input α_1 to the first motor network is determined as follows (possibly with an extra offset if FORCEDMOTORROTATION is different from 0):

- If PRIMARYMOTORSOURCE is "GridCellDecoder", update the grid cell decoder and set α_1 as follows:
  - If the agent's distance to the center of the replay place cell is within three times the place field radius, consider the subgoal to be within visible range and explicitly calculate α_1 as the true bearing to the center of the replay place cell
  - Otherwise, set α_1 to the direction indicated by the grid cell decoder
- If PRIMARYMOTORSOURCE is "LastHeading", set α_1 to the agent's current bearing

The directional input α_2 to the second motor network is the output from the first motor network, and $\delta_2 = 0.1$. The combination of two motor networks is used to alleviate potential errors, e.g. traveling head-on toward sharp corners, when the border cells might symmetrically inhibit the motor population on both sides of the bump; even after receiving border cell inhibition, the agent would still be headed directly toward the obstacle. By repeating the entire process in a duplicated circuit (with a narrower/stricter tuning width), the agent can be halted if the deflected goal vector is still on a collision course with the obstacle.

To determine whether to halt the agent or to allow it to move, we calculate a "confidence" value c as follows: Let s_n denote the strength of the motor signal in motor network n after border cell inhibition as a ratio of the corresponding strength before inhibition. "Strength" here refers to the length of the population vector. The confidence value is then calculated as $c = \sqrt{s_1 \cdot s_2}$. The agent is halted if c is less than MOVEMENTTHRESHOLD, otherwise it is allowed to move in the direction indicated by the second motor network. The output signal MOVEMENTHALTED is set accordingly.

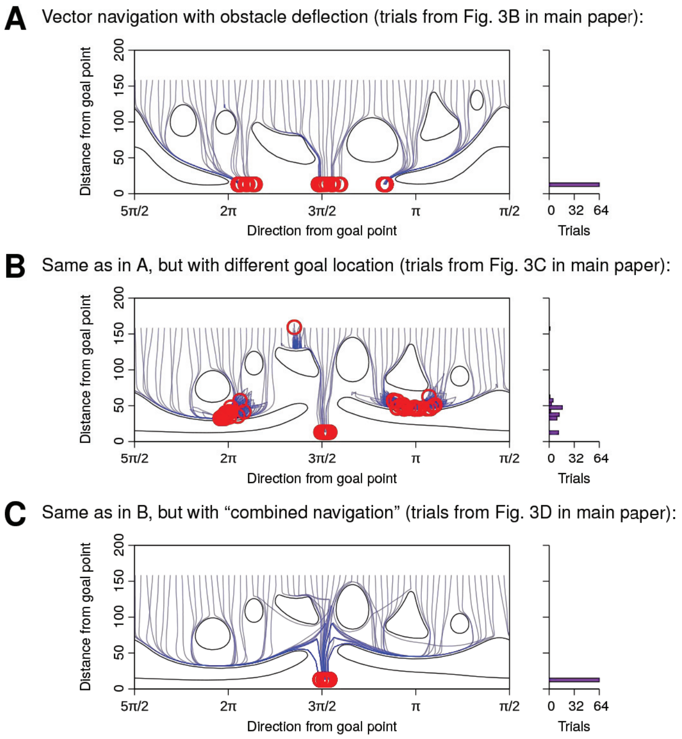
Special-case handling is added throughout to avoid divisions by zero, should e.g. all motor activity be cancelled out by border cell inhibition or the grid cell decoder return a zero-length goal vector. The agent will halt in these situations.

2 Energy landscapes for vector navigation with obstacle deflection

In the main paper we make the distinction between slanted and perpendicular obstacles, the former being negotiable by vector navigation augmented by an obstacle deflection mechanism. This property of slanted obstacles becomes apparent if we consider the navigation problem from the perspective of an "energy landscape"—considering each position in the environment to have as its "energy" the distance to the goal location—with the objective of the agent being to minimize this energy. The energy landscape can be visually inspected by drawing the geometry of the environment with a "reversed-polar" transformation; rather than drawing a given straight line according to its Euclidean coordinates, points along the line are transformed into goal direction θ and goal distance r—on respectively the x-axis and the y-axis—using the following expressions (varying t between 0 and 1):

$$\theta(t) = \operatorname{atan2}\left(y_{start} + t \cdot (y_{end} - y_{start}),\ x_{start} + t \cdot (x_{end} - x_{start})\right),$$

$$r(t) = \sqrt{[x_{start} + t \cdot (x_{end} - x_{start})]^2 + [y_{start} + t \cdot (y_{end} - y_{start})]^2}.$$



A Vector navigation with obstacle deflection (trials from Fig. 3B in main paper):

B Same as in A, but with different goal location (trials from Fig. 3C in main paper):

C Same as in B, but with “combined navigation” (trials from Fig. 3D in main paper):

Figure S2: (A–C) Trials from Fig. 3B–D in the main paper, but with obstacles and trajectories plotted according to a “reversed-polar” transformation—making it possible to visually inspect where local minima will be in the environment given the particular goal location. Histograms of goal distances at trial termination show how many trials were successful in reaching the goal.

In this transformed view of the environment, locations farther away from the goal—those with greater energy—are drawn higher on the y-axis, while the goal location—with zero energy—coincides with the zero level on the y-axis. Fig. S2A shows the results from Fig. 3B in the main paper transformed in this fashion. All trajectories start out along a horizontal line high in the energy landscape, because all instances of the agent were equally distant from the goal when they set out from their respective starting locations along the circular perimeter. The trajectories then proceed down along the y-axis toward the goal; vertical lines show trajectories that were able to progress directly toward the goal location unimpeded by obstacles, while encounters with slanted obstacles cause deflections to the left or right. Importantly, there are no "local minima" in this energy landscape—whenever the agent reaches an obstacle, there is always a deflected direction in which the goal distance keeps decreasing, and so the agent can remain on its downward trajectory through the energy landscape toward the goal location at the bottom.

Choosing a different goal location will alter the energy landscape; sometimes these changes will "warp" the landscape in such a way that the agent might become stuck where it previously would not, and vice versa. Fig. S2B shows the energy landscape after the goal location was moved in Fig. 3C in the main paper—local minima have now appeared, and the agent got stuck in these locations. However, this warping of the energy landscape—caused by moving the goal location—can also be used to the agent's advantage: During hippocampal replay episodes in our model, the subgoal shifts gradually closer to the current location until the decoded goal vector is no longer blocked by the obstacle. At that point, the energy landscape has warped in such a way that the current agent location is no longer a local minimum, and the agent can therefore make further progress toward the goal. In Fig. S2C (from Fig. 3D in main paper), using the combined strategy, the agent was able thus to escape the local minima.

References

Burak, Y. and Fiete, I. R. (2009). Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput. Biol.*, 5(2).

Edvardsen, V. (2015). A passive mechanism for goal-directed navigation using grid cells. In *Proceedings of the European Conference on Artificial Life (ECAL) 2015*, pages 191–198.

Edvardsen, V. (2017). Long-range navigation by path integration and decoding of grid cells in a neural network. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2017*.

Stemmler, M., Mathis, A., and Herz, A. V. (2015). Connecting multiple spatial scales to decode the population activity of grid cells. *Sci. Adv.*, 1(11):e1500816.

Supplementary Videos – Edvardsen et al. (2019)

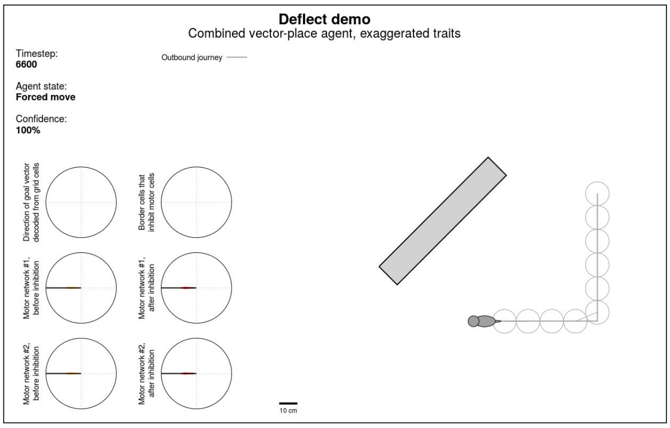
Navigating with grid and place cells in cluttered environments

[This paper is accompanied by four Supplementary Videos to support the figures and descriptions in the main paper. To give a sense for what these videos convey, here are presented a few frames from each video.]

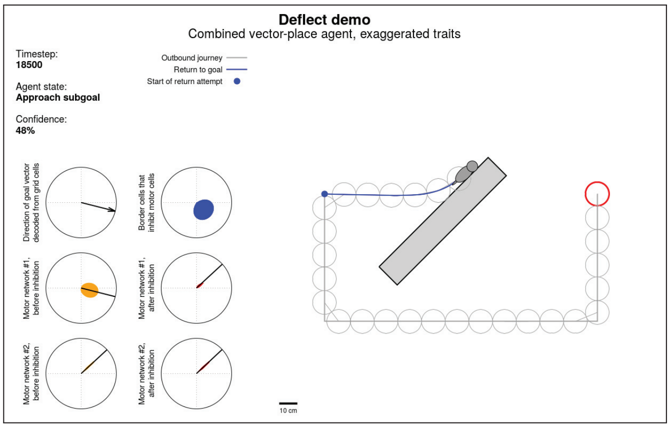
Supplementary Video 1 – Edvardsen et al. (2019)

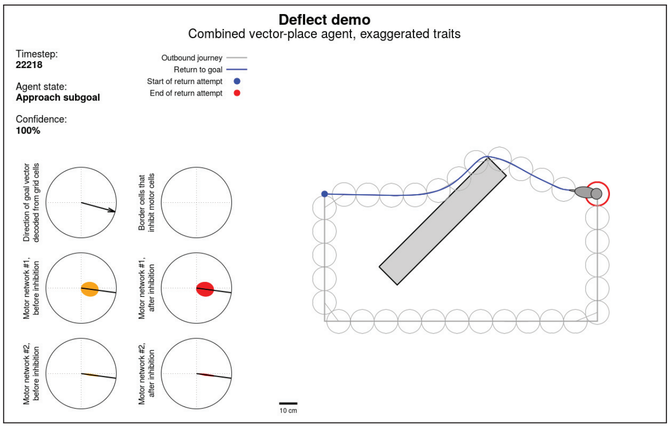
Proposed solutions to obstacles: Slanted obstacles – border deflection

Frame at ∼7.5 seconds


Deflect demo
Combined vector-place agent, exaggerated traits
Timestep:
6600
Agent state:
Forced move
Confidence:
100%
Outbound journey
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
10 cm

Frame at ∼13.5 seconds


Deflect demo
Combined vector-place agent, exaggerated traits
Timestep:
18500
Agent state:
Approach subgoal
Confidence:
48%
Outbound journey
Return to goal
Start of return attempt
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
10 cm



Deflect demo
Combined vector-place agent, exaggerated traits

Timestep:
22218

Agent state:
Approach subgoal

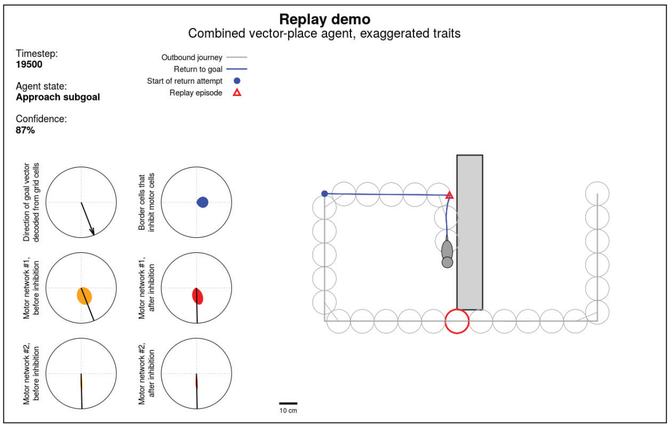
Confidence:
100%

Outbound journey
Return to goal
Start of return attempt
End of return attempt
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
10 cm

Frame at ~15.5 seconds

Supplementary Video 1 – Edvardsen et al. (2019)

Proposed solutions to obstacles: Perpendicular obstacles – hippocampal replay



Replay demo
Combined vector-place agent, exaggerated traits

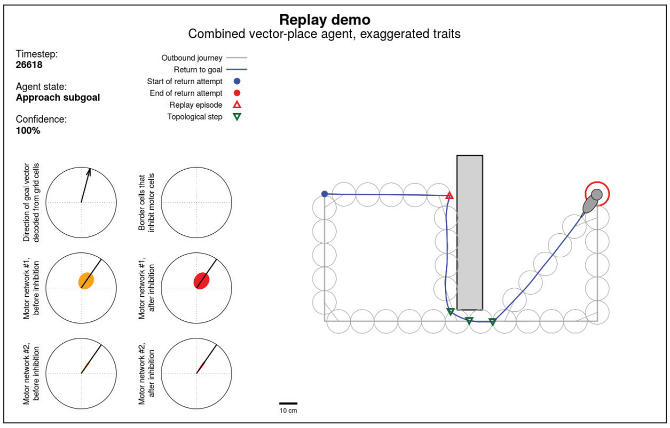
Timestep:
19500

Agent state:
Approach subgoal

Confidence:
87%

Outbound journey
Return to goal
Start of return attempt
Replay episode
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
10 cm

Frame at ~31 seconds



Replay demo
Combined vector-place agent, exaggerated traits

Timestep:
26618

Agent state:
Approach subgoal

Confidence:
100%

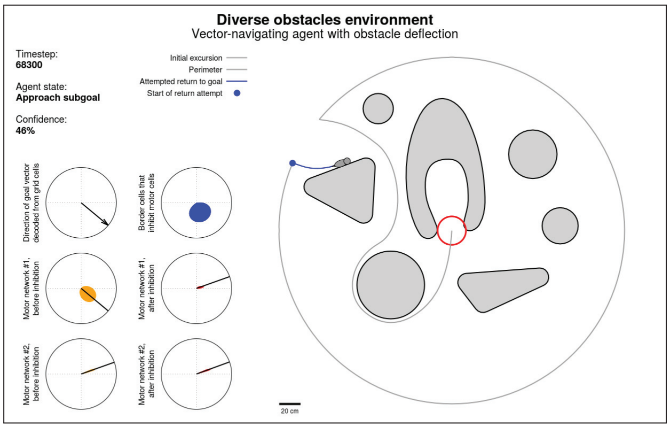
Outbound journey
Return to goal
Start of return attempt
End of return attempt
Replay episode
Topological step
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
10 cm

Frame at ~35 seconds

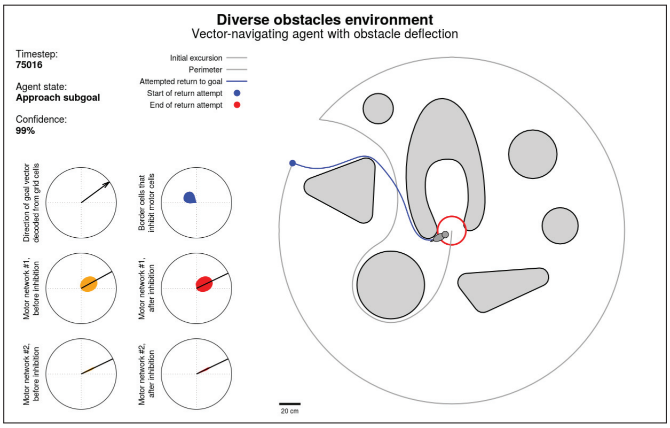
Supplementary Video 2 – Edvardsen et al. (2019)

Vector navigation with obstacle deflection (Fig. 3B)

Frame at ~11 seconds


Diverse obstacles environment
Vector-navigating agent with obstacle deflection
Timestep:
68300
Agent state:
Approach subgoal
Confidence:
46%
Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

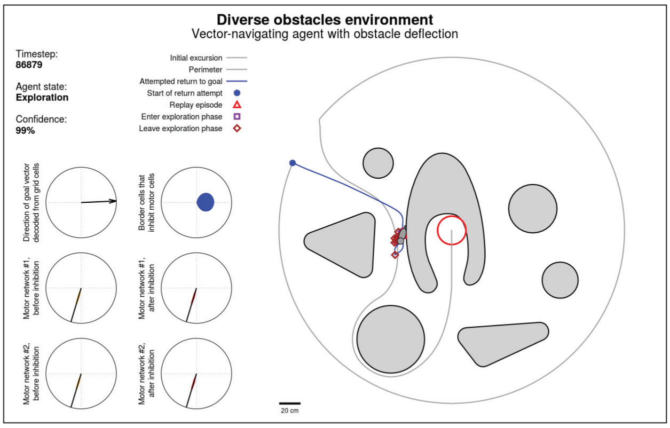
Frame at ~14.5 seconds


Diverse obstacles environment
Vector-navigating agent with obstacle deflection
Timestep:
75016
Agent state:
Approach subgoal
Confidence:
99%
Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
End of return attempt
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Supplementary Video 2 – Edvardsen et al. (2019)

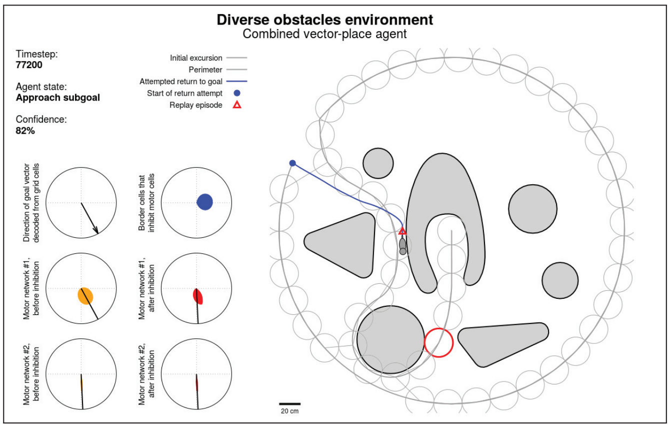
Vector navigation with local minima in energy landspace (Fig. 3C)

Frame at ~38 seconds


Diverse obstacles environment
Vector-navigating agent with obstacle deflection
Timestep:
86879
Agent state:
Exploration
Confidence:
99%
Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Replay episode
Enter exploration phase
Leave exploration phase
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Supplementary Video 2 – Edvardsen et al. (2019)

Combined vector-place navigation to overcome local minima (Fig. 3D)



Diverse obstacles environment

Combined vector-place agent

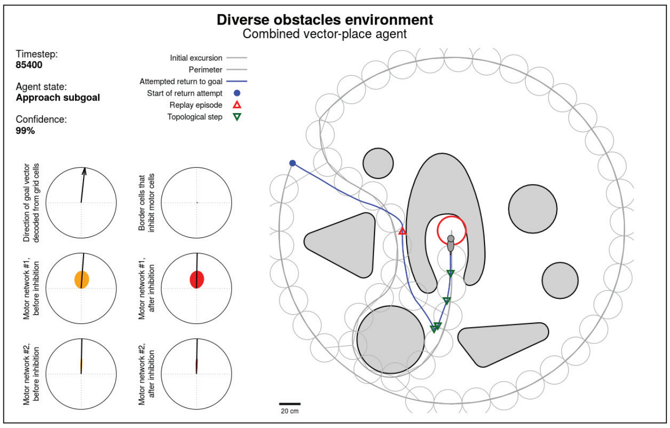
Timestep: **77200**

Agent state: **Approach subgoal**

Confidence: **82%**

Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Replay episode
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Frame at ~55 seconds



Diverse obstacles environment

Combined vector-place agent

Timestep: **85400**

Agent state: **Approach subgoal**

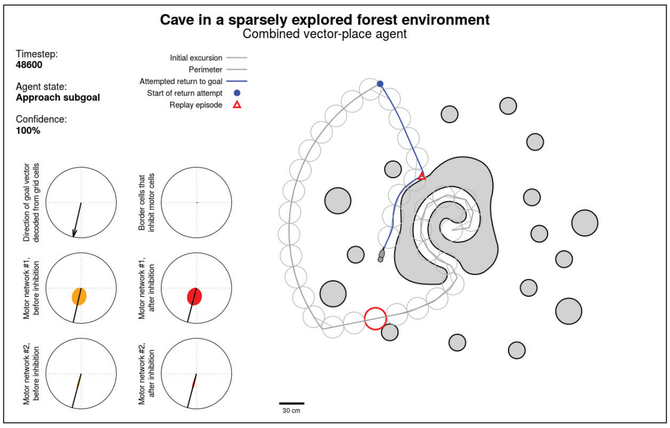
Confidence: **99%**

Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Replay episode
Topological step
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Frame at ~59.5 seconds

Supplementary Video 3 – Edvardsen et al. (2019)

Combined vector-place agent, sparse exploration, without shortcuts



Cave in a sparsely explored forest environment

Combined vector-place agent

Timestep: **48600**

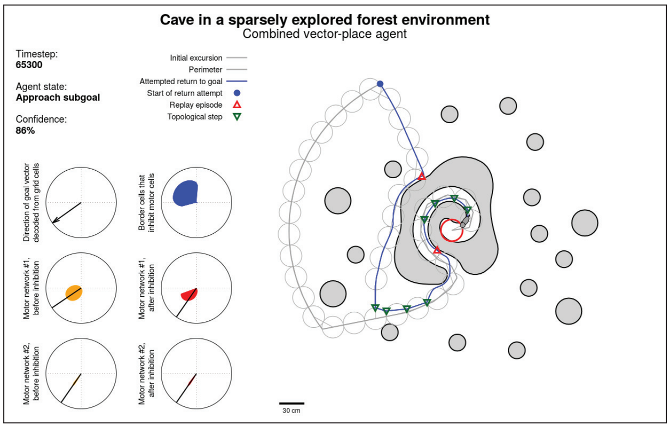
Agent state: **Approach subgoal**

Confidence: **100%**

Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Replay episode
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
30 cm

Frame at ~14 seconds

Frame at ∼23.5 seconds



Cave in a sparsely explored forest environment
Combined vector-place agent

Timestep:
65300

Agent state:
Approach subgoal

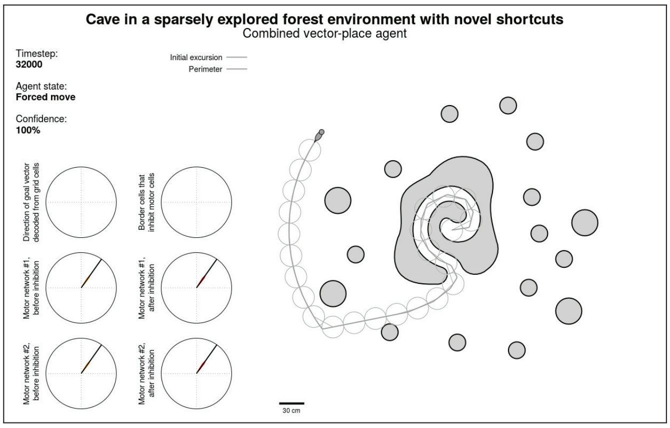
Confidence:
86%

Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Replay episode
Topological step
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
30 cm

Supplementary Video 3 – Edvardsen et al. (2019)

Combined vector-place agent, sparse exploration, with novel shortcuts

Frame at ∼31.5 seconds



Cave in a sparsely explored forest environment with novel shortcuts
Combined vector-place agent

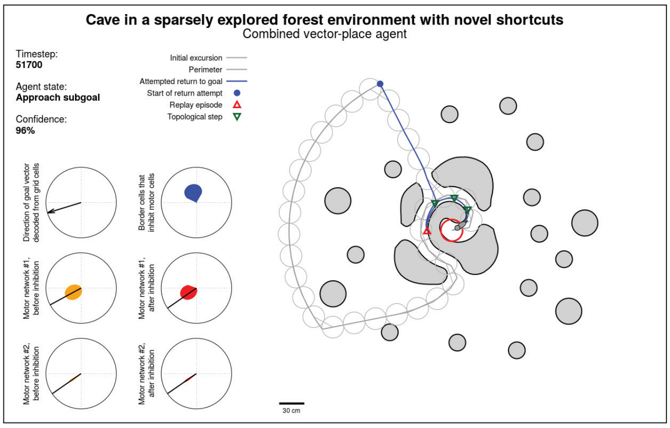
Timestep:
32000

Agent state:
Forced move

Confidence:
100%

Initial excursion
Perimeter
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
30 cm

Frame at ∼40 seconds



Cave in a sparsely explored forest environment with novel shortcuts
Combined vector-place agent

Timestep:
51700

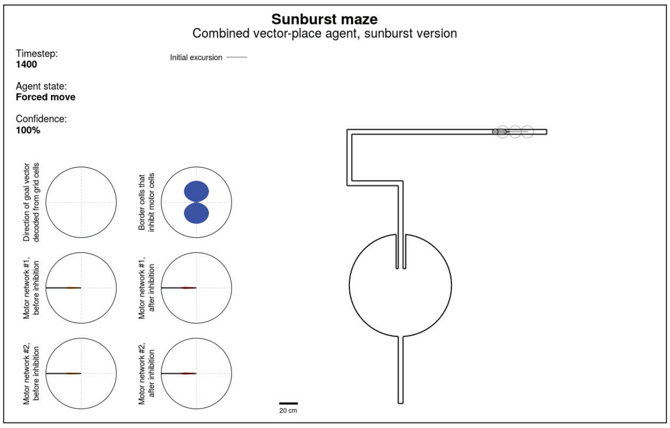
Agent state:
Approach subgoal

Confidence:
96%

Initial excursion
Perimeter
Attempted return to goal
Start of return attempt
Replay episode
Topological step
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
30 cm

Supplementary Video 4 – Edvardsen et al. (2019)

Experimental environments: Sunburst maze



Sunburst maze
Combined vector-place agent, sunburst version

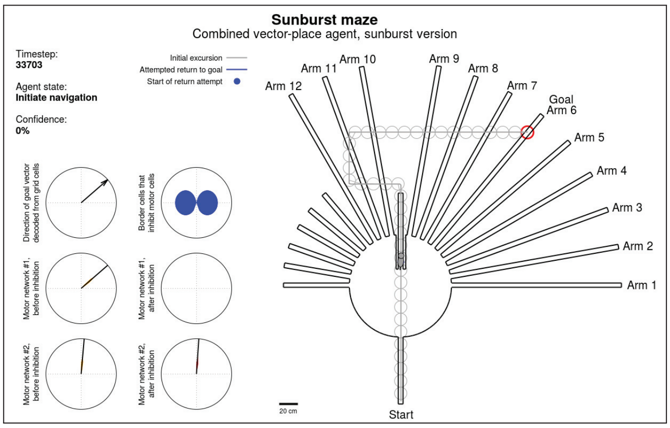
Timestep:
1400

Agent state:
Forced move

Confidence:
100%

Initial excursion
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Frame at ∼4.5 seconds



Sunburst maze
Combined vector-place agent, sunburst version

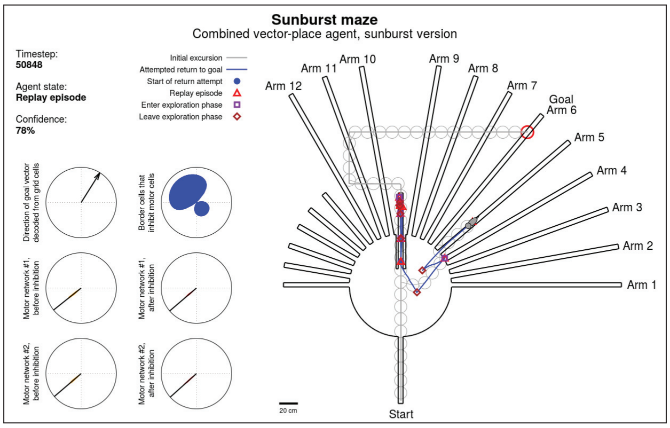
Timestep:
33703

Agent state:
Initiate navigation

Confidence:
0%

Initial excursion
Attempted return to goal
Start of return attempt
Arm 12
Arm 11
Arm 10
Arm 9
Arm 8
Arm 7
Goal
Arm 6
Arm 5
Arm 4
Arm 3
Arm 2
Arm 1
Start
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Frame at ∼9.9 seconds



Sunburst maze
Combined vector-place agent, sunburst version

Timestep:
50848

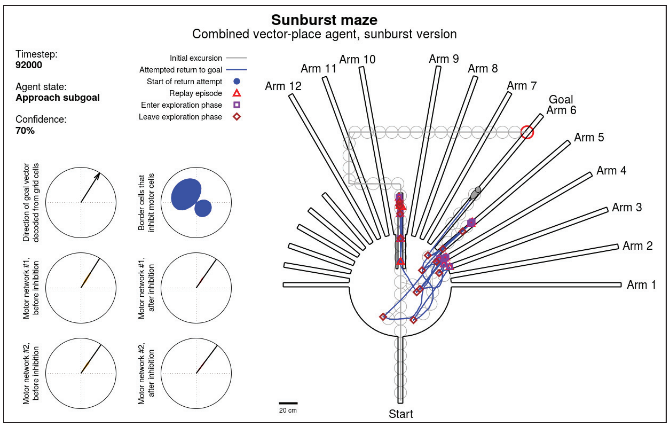
Agent state:
Replay episode

Confidence:
78%

Initial excursion
Attempted return to goal
Start of return attempt
Replay episode
Enter exploration phase
Leave exploration phase
Arm 12
Arm 11
Arm 10
Arm 9
Arm 8
Arm 7
Goal
Arm 6
Arm 5
Arm 4
Arm 3
Arm 2
Arm 1
Start
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
20 cm

Frame at ∼23 seconds

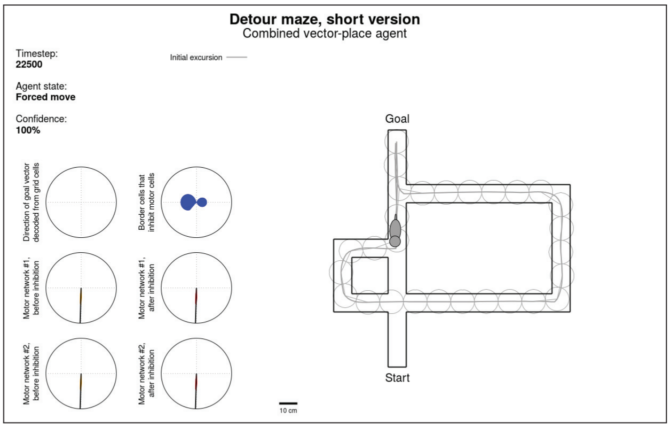
Frame at ~45.5 seconds



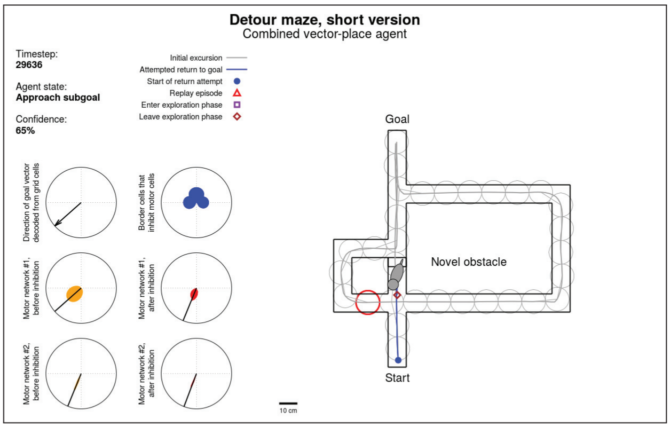
Supplementary Video 4 – Edvardsen et al. (2019)

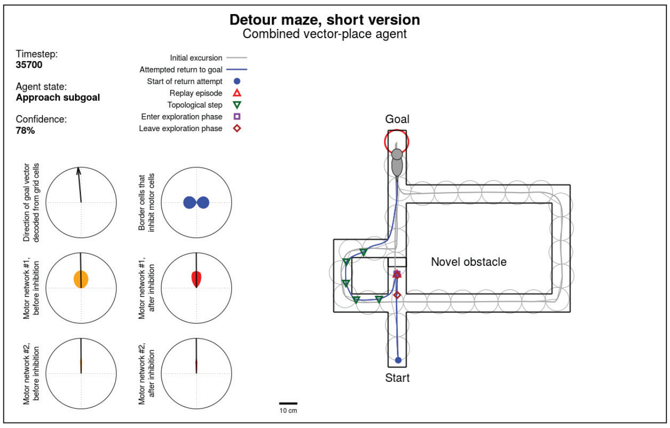
Experimental environments: Detour maze, short version

Frame at ~54.5 seconds


Detour maze, short version
Combined vector-place agent
Timestep:
22500
Agent state:
Forced move
Confidence:
100%
Initial excursion
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
Goal
Start
10 cm

Frame at ~58.4 seconds


Detour maze, short version
Combined vector-place agent
Timestep:
29636
Agent state:
Approach subgoal
Confidence:
65%
Initial excursion
Attempted return to goal
Start of return attempt
Replay episode
Enter exploration phase
Leave exploration phase
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
Goal
Novel obstacle
Start
10 cm



Detour maze, short version
Combined vector-place agent

Timestep:
35700

Agent state:
Approach subgoal

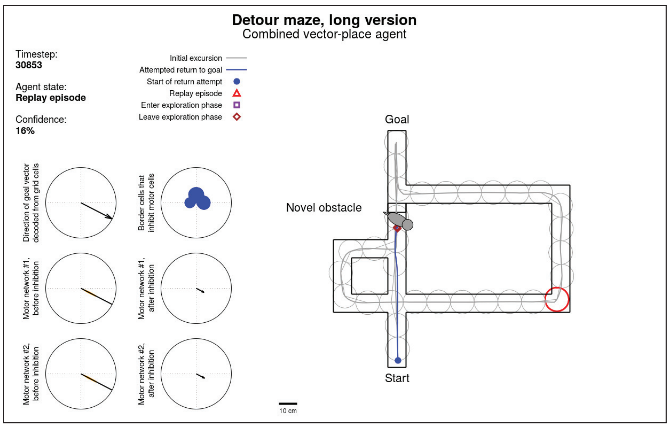
Confidence:
78%

Initial excursion
Attempted return to goal
Start of return attempt
Replay episode
Topological step
Enter exploration phase
Leave exploration phase
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
Goal
Novel obstacle
Start
10 cm

Frame at ∼62 seconds

Supplementary Video 4 – Edvardsen et al. (2019)

Experimental environments: Detour maze, long version



Detour maze, long version
Combined vector-place agent

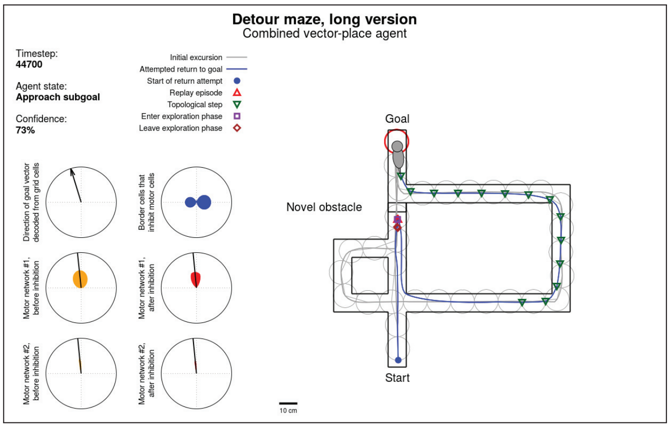
Timestep:
30853

Agent state:
Replay episode

Confidence:
16%

Initial excursion
Attempted return to goal
Start of return attempt
Replay episode
Enter exploration phase
Leave exploration phase
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
Goal
Novel obstacle
Start
10 cm

Frame at ∼75.5 seconds



Detour maze, long version
Combined vector-place agent

Timestep:
44700

Agent state:
Approach subgoal

Confidence:
73%

Initial excursion
Attempted return to goal
Start of return attempt
Replay episode
Topological step
Enter exploration phase
Leave exploration phase
Direction of goal vector decoded from grid cells
Border cells that inhibit motor cells
Motor network #1, before inhibition
Motor network #1, after inhibition
Motor network #2, before inhibition
Motor network #2, after inhibition
Goal
Novel obstacle
Start
10 cm

Frame at ∼84 seconds

Paper E

Navigating with distorted grid cells (Edvardsen, 2018)

Author(s):
Vegard Edvardsen

Published at conference:
2018 Conference on Artificial Life (ALIFE)

Copyright:

Paper E
Edvardsen (2018)

Navigating with distorted grid cells

Vegard Edvardsen

Department of Computer Science,
NTNU – Norwegian University of Science and Technology, Trondheim, Norway

Abstract

Grid cells in the hippocampal formation are a valuable system to study both for neuroscientists and for neural network researchers, as these neurons present both a window into higher-level cognitive processes such as navigation, as well as inspiration for how to build artificial neural navigation systems. Grid cells are believed to represent an animal's coordinates in two-dimensional space in a general fashion, useable for geometric computations by downstream neural networks, and earlier neural models have indeed shown how grid cells can be decoded for navigational purposes. However, accumulating evidence shows that grid cells are not as stable as assumed by models, but that they exhibit various geometric distortions depending on time and place. This presents a challenge to grid cell decoding models, which mainly separate into "nested" and "combinatorial" ones. Here we present a new and simplified version of a nested grid cell decoder, demonstrate that this decoder can cope with distortions, and show how this relates to a fundamental property of nested grid cell decoding. By providing positive proof that a nested decoder can navigate with distorted grid cells, we hope to inspire further neuroscientific investigation into the biological plausibility of different models for grid cell-based navigation.

1 Introduction

Navigation is an interdisciplinary problem area relevant to both artificial life and neuroscience; it is an important capability for situated and embodied agents, and neuroscientific research on navigation has provided a valuable window into the inner workings of higher-level cognition. Particularly in the hippocampal formation of mammals, numerous spatially selective neurons have so far been identified (Moser et al., 2017), including place cells, border cells, head-direction cells and grid cells. Grid cells are particularly intriguing because of the highly fascinating patterns these neurons generate across space (Hafting et al., 2005), that appear almost algorithmic in nature. Specifically, they respond to the animal's location in the two-dimensional plane in a hexagonal pattern; the neurons activate whenever the agent is located on the vertices of an equilateral triangular lattice imagined to span the entire space available to the animal (Fig. 1A).

The mathematically appealing patterns of grid cells have suggested they might implement a spatial coordinate system: The collective activity of all grid cells might uniquely describe the animal's current coordinates and be useful for inferring geometric relationships (Fiete et al., 2008), and grid cells have now been shown to be decodable into goal vectors by hypothetical downstream neural networks, enabling an agent to navigate toward arbitrary goal locations by reading out the current activity of the grid cell population and comparing this to a stored version of the grid cell activity corresponding to the goal (Edvardsen, 2015; Bush et al., 2015). While these models are interesting architectures for artificial agents, they also provide proofs of concept that real grid cells might contribute to "vector navigation"—thus guiding future neuroscientific research into the function of grid cells.

However, the universal decodability of grid cells has been questioned, particularly by recent discoveries that grid patterns are not as stable and predictable as models have assumed (Krupic et al., 2015; Carpenter and Barry, 2016). Grid patterns have been shown to distort in various ways, for example by shearing and rescaling as a function of the amount of time spent in a particular environment. As such distortions challenge the assumptions of different proposed grid cell decoders, they might help discern any differences in their biological viability. In particular, two major approaches to grid cell decoding are the "nested" (Stemmler et al., 2015) and the "combinatorial" (Fiete et al., 2008) decoders. In this paper we show that a nested decoder can cope with distorted grid cells, without addressing the question of whether a combinatorial decoder could similarly be made able to handle the distortions. Thus, while our results allay the criticism that grid distortions might present a problem for grid cell decoding in general, they leave open the question of whether combinatorial decoding remains viable.

Next in this paper we first describe grid cells, grid modules, how to decode one module and how to combine multiple modules in a nested fashion to navigate long distances. This presents a new, simplified version of our earlier model for nested decoding (Edvardsen, 2015) that makes it easier to reason about the effects of distortions. We then add distortions to the model, show that navigation with one module still succeeds, and demonstrate that navigation with multiple, differently distorted modules also succeeds. Before concluding, we show how our results relate to a fundamental property of nested decoders, by demonstrating that the model can cope even with "perturbed" grid patterns.

2 Grid cells organized into grid modules

Fig. 1A shows examples of four different idealized grid cells, each box a top-down heatmap of a 4 m wide square arena, showing where that particular grid cell might be active. All grid cells in this paper were generated by evaluating

$$\begin{aligned}\mathrm{GC}(x,y) = \max\Big[0,\ -0.2 + \prod_{d=0}^{2}\Big(1+\cos\big(&\\ (x-O_x)\cdot {}^{2\pi}\!/\!_{S}\cdot\cos\left(R+d\cdot {}^{\pi}\!/\!_{3}\right)\cdot {}^{2}\!/\!_{\sqrt{3}} +&\\ (y-O_y)\cdot {}^{2\pi}\!/\!_{S}\cdot\sin\left(R+d\cdot {}^{\pi}\!/\!_{3}\right)\cdot {}^{2}\!/\!_{\sqrt{3}}\big)\Big)\Big]&\end{aligned} \tag{1}$$

at any given (x,y) location, making a hexagonal pattern by intersecting three waves 60° apart (Solstad et al., 2006). A grid cell is here characterized by the parameters S, R and O; S is the scale of its pattern, given as the distance between two peaks, R is its orientation, fixed at $\pi/2$ in this paper, and O is the two-dimensional offset of the pattern from some arbitrary point of reference. Cells with the same scale and orientation are said to belong to a *grid module*, which is our fundamental unit for grid cell-based navigation; assuming there is sufficient coverage of cells in a given module, there will always be an active subset of neurons to help localize the animal no matter where it might be located. As the first three grid cells in Fig. 1A have the same scale (0.9 m) and orientation, they belong to the same module, while the fourth cell, of different scale (0.6 m), belongs to a different one.

The collective activity of all grid cells in a given module can be visualized by arranging them in a matrix according to their individual offset values O_x and O_y, so that cells with similar offsets—thus highly overlapping grid patterns—are located nearby in the matrix, while cells with less overlapping patterns are located farther apart. Fig. 1B shows examples of such “neural sheet” matrices at various snapshots in time, each pixel showing the instantaneous firing rate of a particular grid cell, with all pixels in a given matrix recorded simultaneously. The first matrix shows a grid module that includes the first three cells from Fig. 1A, recorded while the agent is located at the goal in the box center. The grid cells from Fig. 1A are in this matrix represented respectively by the 1st, 6th and 11th upper row pixels from the left.

Each matrix reveals a single “bump” of activity distributed among the grid cells, potentially wrapping around horizontally from the upper left corner to the upper right corner and vertically from both upper corners to the lower middle of the matrix. This is because the neural sheet reflects the “twisted torus” topology inherent to the hexagonal grid cell pattern (Guanella et al., 2007). The second matrix shows the same module, but with the agent located 20 cm east of the previous location—the activity bump has thus shifted to the right in this updated snapshot. The third matrix shows the activity a further 20 cm east, while the fourth matrix shows the activity after the agent has then moved 20 cm south.

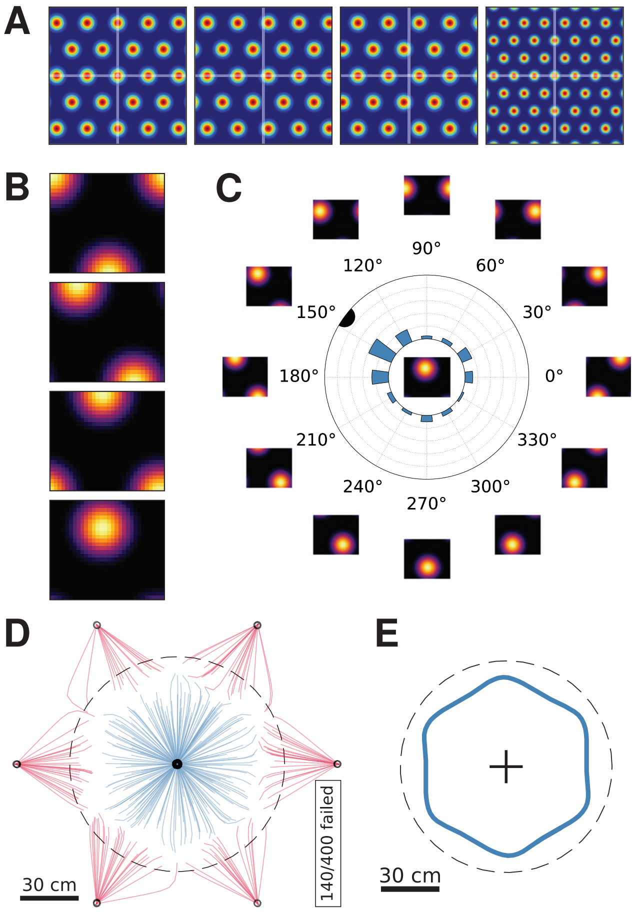
3 Navigating with a single grid module

Comparing the first and the fourth matrices, we see that the collective activity of the grid module has updated to reflect the agent’s new location southeast of the goal, by moving the bump to the right and down from its initial location in the matrix. The principle behind the grid cell decoder is to recognize the correct shift of the bump needed to bring it back to its correct place in the matrix. Fig. 1C shows how we propose to do this: A set of detectors is instantiated for 12 different directions, each detector consisting of a “template” of the expected state of the grid module should the goal be located in a particular direction away from the agent.

For a given current state of the grid cell module, the task of the decoder is then to compare that current state to each of these 12 template states. We have implemented this as the dot product between the template matrix and the current matrix (after flattening both to vectors), essentially corresponding to a neuron with multiplicative synapses between input signals from corresponding current state and target state grid cells. Fig. 1C shows how the different detectors activate in a situation corresponding to the fourth matrix in Fig. 1B—the detectors corresponding to northwesterly goal directions are the most strongly activated, thus correctly compelling the agent to move northwest in order to reach the goal.

In Fig. 1D we use this mechanism to navigate, starting the agent at random locations sampled from the area of a 0.6 m radius disc. In 1 cm timesteps, the agent updates the grid cell population to reflect its current location, compares this to the grid cell population for the goal using the mechanism outlined above, and calculates its next movement direction from the population vector average of a set of unit vectors pointing in the directions indicated by the 12 detector units, weighted by their corresponding activation values. This process iterates until the agent's current location is less than 2.5 cm from five timesteps ago, signalling that the agent has reached what it believes to be the goal and started oscillating around this point. Fig. 1D shows the outcome of 400 such trials, blue lines indicating successful trials that ended within 4.5 cm of the goal and red lines showing failed trials.

These trials clearly separate into two classes; the successful ones that precisely hit the intended goal location, and the unsuccessful ones that instead aim directly toward one of a discrete set of erroneous target locations, distributed in a hexagonal pattern. Because grid cells have a repeating pattern, such ambiguity will inevitably arise as you go too far away from the goal. Fig. 1E clarifies this by tracing out the boundary at which the decoded goal vector flips from pointing inward, i.e. less than 90 degrees away from the true goal direction, to instead pointing away from the goal (blue line). Considering the radius of starting locations (dashed line), some trials in Fig. 1D may evidently have started outside of the valid navigational range for this single grid module.


A
B
C
90°
60°
30°
0°
330°
300°
270°
240°
210°
180°
150°
120°
D
30 cm
140/400 failed
E
30 cm

[Figure on the facing page.]

Figure 1: Navigating with a single grid module. (A) Top-down view of 4 m wide square box; heatmaps of where four grid cells are active (blue/red colors for low/high activity). White cross indicates goal location in box center. First three grid cells have scale 0.9 m, y offset (O_y) of 0 cm and x offsets (O_x) of 0 cm, 15 cm and 30 cm. Fourth cell has scale 0.6 m. (B) Matrices with snapshots of concurrent activity within a full grid module at different points in time. Each plot shows collective activity of all 30×26 cells in the module, each pixel indicating a given neuron's instantaneous firing rate and arranged according to its offset values O_x and O_y. First matrix shows case with agent in box center, second matrix after moving 20 cm east, third matrix another 20 cm east and fourth matrix after then moving 20 cm south. (C) Mechanism for decoding a single grid module, consisting of 12 units each responsible for detecting the expected module state should the goal be located $\sim$8 neurons in a particular direction from the current location, here configured to navigate back to the location indicated in top matrix in B. Center matrix shows current module state. Radial bars show activation of each detector. Black semi-circular notch shows final goal direction from the population vector average of all detectors' contributions, correctly indicating the goal to be to the north-west. (D) Trajectories from navigating with decoder outlined above. 400 trials were started from locations randomly sampled across the area of a 0.6 m radius disc. Trials terminating within 4.5 cm of true goal location were deemed successful and shown in blue. Failed trials shown in red. Trajectories were plotted with 5× subsampling throughout paper. (E) Trace of boundary where decoded vector flips from pointing inward to pointing outward (blue line). Dashed line shows maximum possible starting distance for trials in D, showing that some trials might start outside this boundary and thus fail.

While the range of this system is quite short, the system is able to pinpoint the correct goal location with high precision; one immediate proposal for how to increase the navigational range of this system would thus be to trade some of this precision for an increase in range. We could e.g. zoom up the scale of the grid module by a factor of a thousand, and assuming the agent remained able to hit the goal location, we would then have gained a thousandfold increase in range "for free". However, we cannot expect this to be viable biologically, as there will inevitably be noise in the system e.g. due to error from path integration processes (proposed to participate in maintaining grid patterns in the real brain).

Hence, for a more realistic take on the grid cell decoding problem, we explicitly introduce some jitter into the coordinates represented by each of the grid modules; specifically, on each new trial, the true goal coordinates are offset by a noise vector with random direction and with magnitude sampled from a Gaussian distribution with standard deviation 5 % of the grid scale. Once such jitter is taken into account, simply scaling up one grid module might no longer be a viable way of increasing navigational range while maintaining sufficient precision. A possible solution, for which there is biological backing, is then to combine information from multiple modules of different scales—described next.

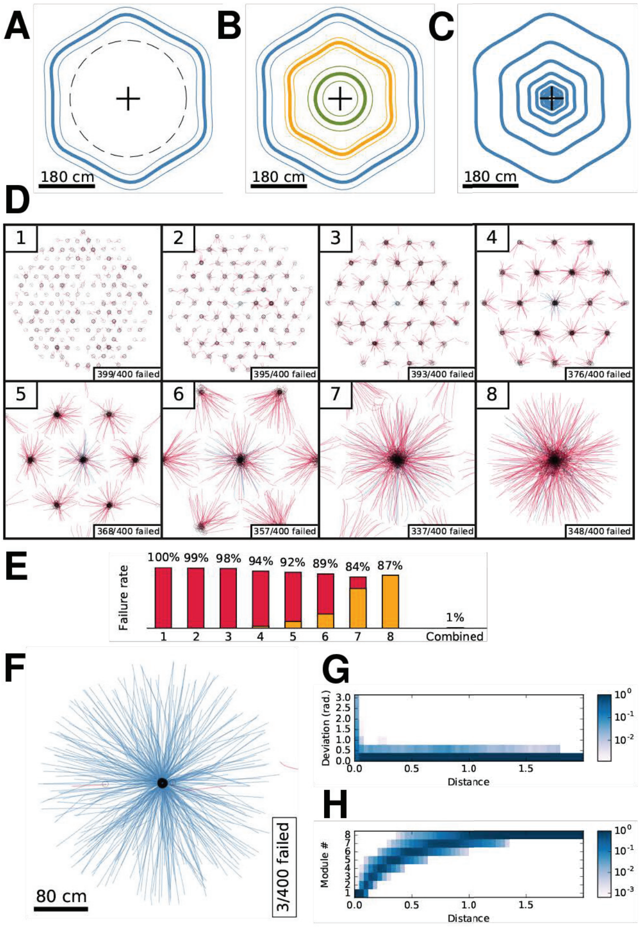
4 Combining multiple grid modules for nested grid cell decoding across longer distances

In a nested view of the grid system (Stemmler et al., 2015), we assume there exists a grid module with a sufficiently long navigational range to exceed the agent's behavioral requirements (Fig. 2A). While this module might be too jittery to successfully hit the goal location, we then assume that additional smaller-scaled modules can fine-tune the navigational resolution once the agent gets closer to the goal. Fig. 2B shows a smaller grid module (yellow) nested within a larger one (blue), as well as the criterion we propose for handing off control from the larger to the smaller module: An extra detector unit, configured to detect the target location itself, is added to the larger-scaled module—the green line here showing the area within which this unit will fire above a certain threshold. We propose that decoding is an iterative process, where the largest-scaled module has control of the decoded goal direction until its target detector exceeds its threshold. It will then yield control of the navigation process to the next, smaller module—with its own set of detectors—which assumes control until its respective target detector is sufficiently strongly activated, and so on. This nested process, executed on every timestep, can be maintained for as long as there are smaller-scaled grid modules available.

A high amount of jitter in large-scaled modules is thus acceptable as long as the agent can be delivered into the valid navigational range of smaller-scaled modules with less debilitating jitter (Fig. 2C). Grid cells in the real brain are known to cluster into multiple coexisting grid modules of increasing scale, and intriguingly, these modules seem to have a fixed ratio between their successive grid scales, so that the sequence of grid scales within an animal constitutes a geometric progression (Stensola et al., 2012). There is thus an exponential growth in range as more modules are added.

Say we wanted to navigate from distances of up to 2 m while coping with jitter—we could e.g. configure our agent to utilize eight distinct grid modules, with grid scales progressively increasing from a lower value of 0.3 m with a fixed ratio of 1.5 between modules, for a largest scale of $\sim$5 m. Fig. 2D first demonstrates how each of these modules would perform individually, each box showing 400 trials with only one of the eight modules present. Using only a small-scaled module, trials terminate early in a multitude of erroneous goal locations. As scale increases, the "catchment area" for the correct goal location grows too, thus improving the number of trials that head toward the correct target—however, the amount of jitter increases as well, so the precision is no longer sufficient for trials to succeed. The largest grid module's catchment area encompasses all trials, yet 87 % of trials still failed because of the jitter (Fig. 2E).

If we instead combine several modules as described above, the agent should enjoy both the long range of the larger modules and the precision of the smaller modules. In Fig. 2F, with the agent now combining all eight modules in a nested fashion, almost all trials successfully reached the goal within specified tolerances. Drilling further into the agent's behavior, histograms in Fig. 2G–H show respectively the deviation in decoded goal direction and the


A
B
C
180 cm
180 cm
180 cm
D
1
399/400 failed
2
395/400 failed
3
393/400 failed
4
376/400 failed
5
368/400 failed
6
357/400 failed
7
337/400 failed
8
348/400 failed
E
Failure rate
100% 99% 98% 94% 92% 89% 84% 87%
1%
1 2 3 4 5 6 7 8 Combined
F
80 cm
3/400 failed
G
Deviation (rad.)
3.0 2.5 2.0 1.5 1.0 0.5 0.0
0.0 0.5 1.0 1.5
Distance
10^0 10^{-1} 10^{-2}
H
Module #
8 7 6 5 4 3 2 1
0.0 0.5 1.0 1.5
Distance
10^0 10^{-1} 10^{-2} 10^{-3}

[Figure on the facing page.]

Figure 2: Navigating with multiple grid modules. (A) Maximum trial distance was increased to 2 m (dashed line). Catchment area of largest-scaled grid module (#8) is sufficiently large to contain all trials (blue line). Thin solid lines indicate jitter in the corresponding module. (B) Nested within module #8 is the next-largest-scaled module (#7, yellow line). Drop-off area from module #8 (green line) is inside catchment area of #7, thus nested navigation should succeed. (C) In total eight modules are nested within each other, grid scales organized in a geometric progression from a smallest value of 0.3 m with a fixed ratio of 1.5 between successive modules. (D) Trial outcomes if the agent were to navigate using only one of the eight possible modules. Upper left shows 400 trials with smallest module. Module scale increases in standard reading order. Smaller-scaled modules are more ambiguous and have a denser distribution of erroneous goal locations, but the larger-scaled modules experience more jitter and hence are also insufficient by themselves. Each box shows a 4.5 m wide square top-down view of environment. (E) Failure rates in each of the situations shown in D, red indicating trials that failed outside of the expected catchment area for a module of the given scale (calculated according to $0.15\text{ m}\cdot 1.5^{i-1}$ for module i), yellow indicating trials that failed within the catchment area but due to jitter were not able to hit within the success criterion of 4.5 cm. (F) Trials combining all eight modules in a nested fashion. Almost all trials were successful. (G) 2D histogram showing deviation in the decoded goal direction as a function of distance (normalized per column). Histogram includes samples from every 1 cm timestep in all successful trials in F. Decoding error is kept low until immediately adjacent to the goal. (H) Histogram as in G, but here showing which module was active at the given timestep.

identity of the currently driving grid module, as a function of distance, for all timesteps across all successful trials in Fig. 2F. At far distances, the larger modules are in control of the agent—gradually yielding control to the smaller modules as the goal is approached (Fig. 2H). This maintains the deviation at a low level until the agent is immediately adjacent to the goal (Fig. 2G), thus ensuring successful navigation in these trials.

5 Navigating with a single distorted grid module

In the preceding sections we assumed perfectly hexagonal grid patterns, reliably adhering to the equilateral lattice everywhere. However, this is not an accurate description of the real situation, as grid patterns in real animals have been shown to distort from the perfect hexagonal grid in various ways. For example, Stensola et al. (2015) showed that with increased exposure to a particular enclosure, grid patterns may gradually shear away from the walls of the box, while Barry et al. (2007) showed that when a familiar enclosure is compressed, grid patterns may also rescale to match the new dimensions of the box. Interestingly, other co-recorded grid modules might stay fixed at their previous scales (Stensola et al., 2012), highlighting the notion that different grid modules might distort incoherently. Several more examples of grid cells deviating from the idealized hexagonal pattern have been identified (Carpenter and Barry, 2016).

While the purpose and underlying mechanisms of these distortions are still unknown, their mere presence poses a challenge for grid cell navigation models: Can these decoders still function when grids are not perfectly hexagonal (Krupic et al., 2015; Carpenter and Barry, 2016)? To investigate how our decoder copes with distortions, we have thus incorporated one-axis stretching (Barry et al., 2007; Stensola et al., 2012), one-/two-axis shearing and symmetric shearing (Stensola et al., 2015) into our model as follows. Each distortion is represented in Fig. 3A by a function $D : (x,y) \mapsto (\widetilde{x},\widetilde{y})$ that maps the agent's true (x,y) coordinates to the distorted coordinates $(\widetilde{x},\widetilde{y})$. These distorted $\widetilde{x}$ and $\widetilde{y}$ values are then used when determining grid cell activation using Eq. 1. The specific expressions used for stretch, one-/two-axis shear and symmetric shear distortions are given respectively by $D_{stretch}$, D_{shear} and $D_{symmetric}$ (the latter two based on formulations in Stensola et al., 2015):

$$D_{stretch}(x,y) = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

$$D_{shear}(x,y) = \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

$$D_{symmetric}(x,y) = \left(\frac{x}{1+a}, \frac{y}{1+a \cdot {}^{x}\!/_{1+a}} \right) \tag{4}$$

See Fig. 3B for examples of resulting distorted grid cells.

The first question to address is whether navigation with a single grid module can still succeed when the constituent grid cells are distorted. If we assume that all grid cells within a given module distort in the same way, which seems to be the case biologically (Stensola et al., 2012, 2015), we expect the neural sheet to look the same as without distortions: Because distortion is a function of location, all cells will experience the same x and y distortion simultaneously—the activity bump in the neural sheet should thus retain its shape.

We can therefore attempt to navigate with a distorted grid module by using the same decoder as earlier, unmodified. Fig. 3C shows catchment areas traced as before, but with distortions in effect. Inside these boundaries, the decoded goal vectors point inward toward the goal, a promising sign that navigation with distorted modules might succeed. Fig. 3D presents a set of navigation trials performed with a sheared grid module as shown in Fig. 3A second column, demonstrating that a large proportion of trials remained able to reach the goal. Observe that the shape of the catchment area has been distorted as in Fig. 3C, and that the distribution of erroneous goal locations has also similarly been distorted.

Careful observation of the trajectories reveals that they might now curve; because the decoder is unaware of any distortions in effect, it would not know e.g. that a decoded goal vector pointing east might in fact correspond to a northeastern bearing under the distorted regime. Fig. 3F shows a scatter plot of this deviation from the true goal direction during successful trials in Fig. 3D (yellow dots), revealing a clear pattern of deviation. For grid distortions that produce constant $\widetilde{x}$ and $\widetilde{y}$ gradients throughout space, which includes $D_{stretch}$

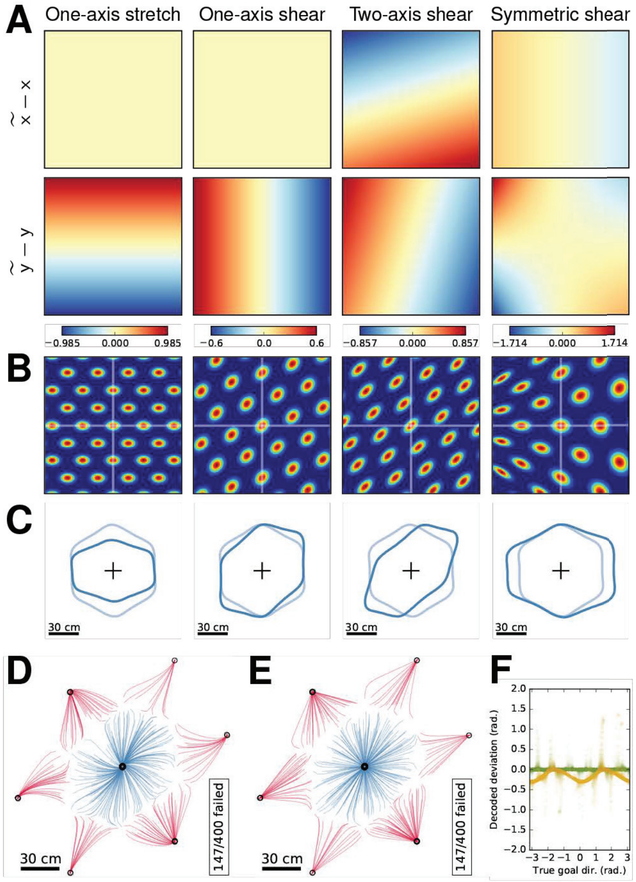


Figure 3: Distorted grid cells in a single module. (A) Examples of four distortions, shown here as the difference due to distortion, hence $\widetilde{x}-x$ in upper and $\widetilde{y}-y$ in lower row. Columns show respectively $D^{a=1,\ b=0.67}_{stretch}$, $D^{a=0,\ b=0.3}_{shear}$, $D^{a=0.3,\ b=0.3}_{shear}$ and $D^{a=0.3}_{symmetric}$. (B) Grid cells distorted as in A. (C) Catchment area of modules distorted as above (blue), with undistorted version from Fig. 1E shown for comparison (light blue). (D) Traces from 400 navigation trials with a single grid module experiencing one-axis shear as in second column above. Many trials are successful, but trajectories toward the goal often curve because the decoder is unaware of the distortion. (E) New set of 400 trials as in D, but now with compensation as in Eq. 5 applied to the decoder's output. (F) Scatter plot showing deviation from true goal direction as decoded at all timesteps from successful trials in D (yellow) and E (green). Without applying compensation to the decoder output, the decoded goal direction deviates depending on the particular distortion experienced.

and D_{shear} but not $D_{symmetric}$, there is a simple remedy in adjusting the decoded goal vector $(\widetilde{v}_x, \widetilde{v}_y)$ using

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \partial\widetilde{x}/\partial x & \partial\widetilde{x}/\partial y \\ \partial\widetilde{y}/\partial x & \partial\widetilde{y}/\partial y \end{bmatrix}^{-1} \begin{bmatrix} \widetilde{v}_x \\ \widetilde{v}_y \end{bmatrix} \tag{5}$$

as the final step. Fig. 3E shows a new set of trials employing this compensation mechanism. The trajectories no longer curve as in Fig. 3D, and Fig. 3F confirms that the deviation has been corrected (green dots). Note however that, because the curving mostly affects only the path lengths and has little influence on the ultimate success of the trials, we do not actually apply this compensation in the rest of this paper.

6 Navigating with multiple distorted modules

Having verified that navigation with a single grid module succeeds despite distortions, we now ask whether navigation over longer distances, combining multiple nested modules, can also succeed with distorted modules. We expect this to work as long as larger-scaled modules remain able to deliver the agent into the catchment area of the smaller-scaled ones. To investigate this we next ran a set of trials as in Fig. 2F, but with the eight modules distorted using a mix of one-axis stretching, one-/two-axis shearing and symmetric shearing (Fig. 4A, details in caption). In other words, we have configured the agent not only to utilize distorted grid modules, but to have *differently* distorted modules—should this succeed, it would be a stronger demonstration of capability than having one single distortion apply equally to all modules.

Example grid cells from each of the eight modules are shown in Fig. 4A, while Fig. 4B shows outlines of each module's catchment area together with the "drop-off areas" from their respective larger-scaled modules. The condition for successful navigation is that the agent should always be dropped off fully inside the catchment area for the lower-scaled module; Fig. 4B confirms this to apply to the present configuration, but note that in some areas, the extra trial-by-trial jitter in each module's represented coordinates might cause some of these boundaries to intersect and potentially cause problems. Fig. 4C shows 400 trials with these distorted grid modules, but without any additional jitter;

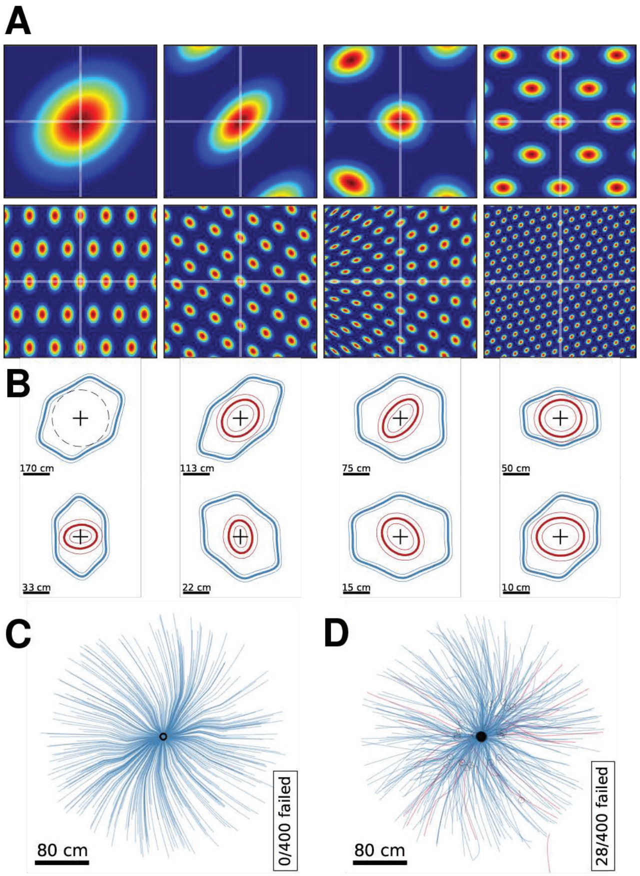


Figure 4: Navigating with multiple distorted modules. (A) Example grid cells from each of the eight modules in the navigation model, largest module (#8) upper left and decreasing in standard reading order. The specific distortions are the following, in the same order: $D^{a=0.3,\ b=0.0}_{shear}$, $D^{a=0.3,\ b=0.3}_{shear}$, $D^{a=0.15}_{symmetric}$, $D^{a=1.0,\ b=0.67}_{stretch}$, $D^{a=0.67,\ b=1.0}_{stretch}$, $D^{a=0.0,\ b=-0.3}_{shear}$, $D^{a=0.3}_{symmetric}$ and $D^{a=0.0,\ b=0.3}_{shear}$. (B) Catchment areas for each module (blue), and drop-off areas from the next larger-scaled module (red)—these must stay within the catchment areas of their smaller-scaled peers for navigation to succeed. The largest module has no relevant drop-off area from above; instead is shown the maximum possible starting distance during trials (dashed line). Note the different scale bars, reflecting the geometric progression of grid scale. (C) 400 trials with eight modules distorted as above, without any additional jitter applied to the modules. All trials were successful, despite the distortions and despite modules being incoherently distorted. (D) New 400 trials as in C, but now with additional trial-by-trial jitter of 5 % of the grid scale (applied to each module's coordinates prior to applying distortions and evaluating Eq. 1). Despite jittery and incoherently distorted modules, navigation succeeded in a vast majority of cases.

all trials were successful. Navigation is still largely successful when the trials are run with additional trial-by-trial jitter (Fig. 4D), with a few exceptions due to failures from the extra jitter. Except for this caveat, though, we see that grid distortions pose no problem for the navigation process with nested grid cell decoding, and that this applies even when different modules are distorted incoherently.

7 Navigating with perturbed grid cells

The previous results highlight an appealing feature of nested grid cell decoders, in that they don't require precisely interlocked conjunctions of grid activity across all modules in order to work. This is in contrast to combinatorial decoders—the main alternative to nested decoders—which exploit the combinatorial growth in the number of unique conjunctions of activity across all modules as new modules are added, to theoretically enable navigation far beyond the range of the largest module (Fiete et al., 2008). A nested decoder, on the other hand, can only navigate within the range of the largest module—although with a geometric progression of grid scales, this range does rise exponentially. In exchange, though, nested decoding faces a far simpler requirement for successful navigation, namely that larger-scaled modules need only deliver the agent into the realm of a smaller-scaled module. The smaller modules do not even need to participate in the process until the larger modules finish their jobs.

This should mean that a nested decoder can function in yet more adverse conditions than those considered so far. Specifically, we will consider what happens to the navigational system when modules are *perturbed* at locations far away from the goal. By "perturbed" we mean that the cells no longer keep perfect track of the agent's exact position relative to the goal, but that e.g. due to noisy path integration, the grid cells' firing locations might jump around and no longer resemble grid patterns when far away from the goal.

Fig. 5A shows how we have implemented these perturbations, by using the same formalism of a $D : (x,y) \mapsto (\widetilde{x},\widetilde{y})$ function. However, this function no longer changes gently across space as in Fig. 3, but instead presents noisy maps that might violently jumble the grid pattern around. Specifically, the

x and y offset maps were generated by selecting 300 random points within a 4 m radius disc, for each point sampling an offset value from a Gaussian distribution with standard deviation 15 cm, and then interpolating across these points (using Clough-Tocher interpolation from the SciPy software package). As mentioned above, though, we only want these perturbations to be in effect when the agent is far away from the goal—we assume that an agent would realistically be able to gradually correct these perturbations from sensory inputs as it approaches a familiar goal location. We therefore fade away the effect of perturbation toward the goal, using a Gaussian fade with standard deviation 0.6 m (Fig. 5B).

In Fig. 5C–D, to see the effects of the perturbation in more detail, we analyze a grid pattern before and after by performing two-dimensional autocorrelation on each 3×3 subdivision of the spatial response heatmap. Autocorrelograms are frequently used in the neuroscience literature to quantify the degree of "gridness" of a particular neuron, as the hexagonal symmetry of a grid cell becomes apparent when correlating the spatial response map in this fashion. Fig. 5C shows that, as expected, there is hexagonal symmetry in the autocorrelograms for the non-perturbed grid cell. In Fig. 5D, however, we see that the neuron only has hexagonal autocorrelation structure in the central subdivision, i.e. close to the goal. The neuron therefore behaves as a grid cell when close to the goal, but at farther locations the perturbation might mean that this neuron would not have been classified as a grid cell.

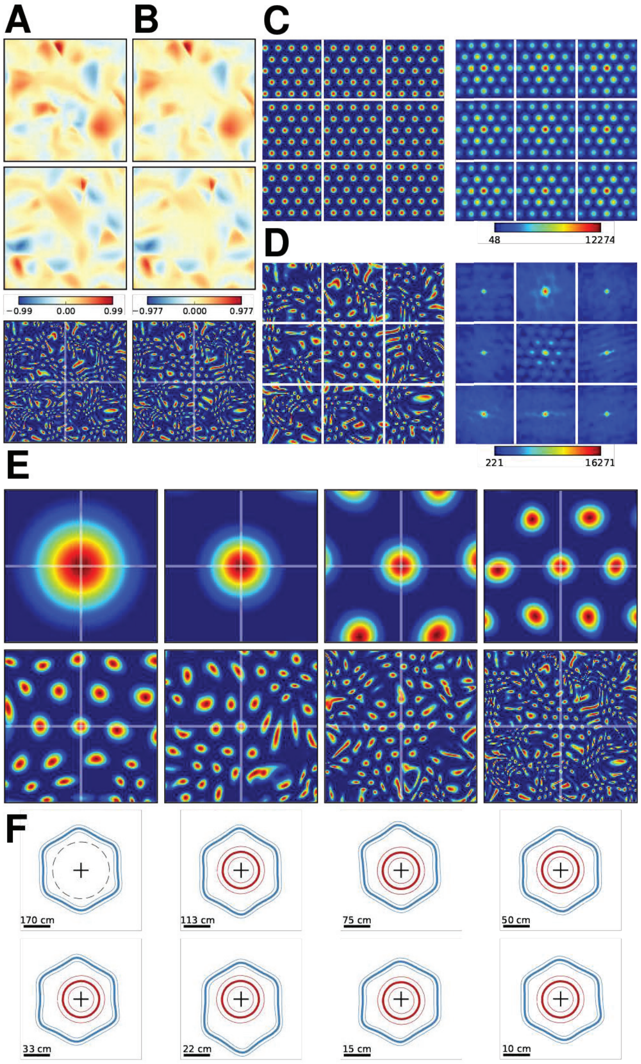
Thus, in some respects, these perturbed cells do not behave as grid cells when sufficiently far away from the goal location. Can these neurons still contribute to a successful navigational process? Fig. 5E–H present results similarly as in Fig. 4A–D, but with a new configuration of the model where the grid modules experience a perturbation as just described. Similar to how the trial-by-trial jitter was specified as a proportion of grid scale, we find it reasonable to assume that larger-scaled modules get perturbed on a proportionally larger scale—noise points are therefore more spread-out, but have a higher amplitude, for larger-scaled grids, and the radius for the fade-away near the goal is also larger for the larger-scaled modules. Fig. 5F shows that the catchment areas for each individual module has largely retained its shape from the undistorted situation—this is as expected, since we have assumed that the perturbation fades off toward the goal location. The conditions are therefore in place for nested decoding to succeed; when trials are run either without or with jitter, shown respectively in Fig. 5G and Fig. 5H, we see that navigation with perturbed grids indeed was successful.

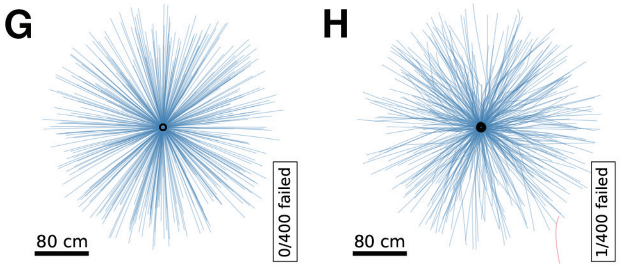
8 Discussion

In this paper we have presented a new and simplified version of our earlier nested grid cell decoder (Edvardsen, 2015), able to perform vector navigation over arbitrarily long distances using multiple modules. Our earlier model performs nested decoding as part of a larger neural navigation system that also includes path integration components to generate the grid cells; here we instead look at the decoding problem in isolation, thus we calculate the grid patterns directly for a simpler overall model. The decoding mechanism itself has also been simplified, now only requiring a set of 13 detector units per module. Whereas the earlier model decoded all modules in parallel and used different weighting of output from different modules, the new version uses a simpler approach to combining multiple modules, by making explicit the underlying principle that a nested decoder only strictly needs to follow the signal from one module at a time.

We then showed that this nested grid cell decoder can cope with distorted grid cells, and that navigation can succeed even when different grid modules experience different distortions. In Stemmler et al. (2015), which presents a nested decoder based on recursive population vector readout of the full grid cell population, they point out that it is possible to accommodate grid modules with distortions such as shearing in their decoder. However, while they compensate for these distortions directly in their readout mechanism, here we show that the decoder might not even need to be aware of the distortions for navigation to succeed.

We showed how this relates to a fundamental property of the nested view of the grid system, namely that navigation can proceed as long as each grid module can guide the agent from inside the drop-off area from a larger module and into the realm of a smaller module. We underscored this by showing that nested decoding succeeds even when grid patterns are perturbed—to the point of no longer resembling grid cells—in locations away from the goal. Our nested navigation model was able to cope with these challenges with no


A
B
C
D
E
F
−0.99
0.00
0.99
−0.977
0.000
0.977
48
12274
221
16271
170 cm
113 cm
75 cm
50 cm
33 cm
22 cm
15 cm
10 cm



[Figure split across the facing page and above.]

Figure 5: Navigating with perturbed grid modules. (A) Type of distortion maps used to generate perturbed grid modules, shown as in Fig. 3 by its offset maps and a resulting example cell. (B) As in A, but now with the distortion fading away toward the goal location. (C) Undistorted grid cell of scale 0.3 m. The rate map is split into 3×3 subdivisions (left), each of which is then autocorrelated to see whether there is any repeating grid structure in the cell's behavior (right). (D) Disrupted cell from B, analyzed as in C. The hexagonal structure in this neuron's behavior has been severely perturbed in the subdivisions away from the goal location, but some grid structure remains near the goal location. (E–H) Results as presented in Fig. 4A–D, but here with a navigation model consisting of eight perturbed grid modules. Navigation was successful despite grid cells being perturbed at locations away from the goal location.

change to its decoding mechanism, whereas e.g. a combinatorial system sensitive to unique conjunctions of activity across all modules, would likely have to be retrained or account for the distortions through some other means.

We have thus demonstrated a nested grid cell decoder robust to disturbances in the form of jitter, distortions and perturbations of grid patterns. Noisy path integration can hence support successful grid decoding, so this might be a viable architecture for neuromorphic robot navigation if extended with circuitry for motor control, obstacle avoidance, etc.

There is also the question of how these results relate back to the biological system. It remains an open question whether real grid cells participate in vector navigation, and if so, whether this occurs according to nested decoding, combinatorial decoding or some other mechanism. Our results here show that a nested decoder can be robust to the challenges of noise and distortions that are relevant in a biological setting, and also that nested decoding can have modest resource requirements in terms of numbers of neurons and synapses. To address these questions on the role of real grid cells in navigation, more data is needed e.g. about how these neurons behave when animals navigate across long distances in their natural environments (Geva-Sagiv et al., 2015).

References

Barry, C., Hayman, R., Burgess, N., and Jeffery, K. J. (2007). Experience-dependent rescaling of entorhinal grids. *Nature neuroscience*, 10(6):682.

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

Carpenter, F. and Barry, C. (2016). Distorted grids as a spatial label and metric. *Trends Cogn. Sci.*, 20(3):164–167.

Edvardsen, V. (2015). A passive mechanism for goal-directed navigation using grid cells. In *Proceedings of the European Conference on Artificial Life (ECAL) 2015*, pages 191–198.

Fiete, I. R., Burak, Y., and Brookings, T. (2008). What grid cells convey about rat location. *J. Neurosci.*, 28(27):6858–6871.

Geva-Sagiv, M., Las, L., Yovel, Y., and Ulanovsky, N. (2015). Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation. *Nat. Rev. Neurosci.*, 16(2):94–108.

Guanella, A., Kiper, D., and Verschure, P. (2007). A model of grid cells based on a twisted torus topology. *International journal of neural systems*, 17(04):231–240.

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

Krupic, J., Bauza, M., Burton, S., Barry, C., and O'Keefe, J. (2015). Grid cell symmetry is shaped by environmental geometry. *Nature*, 518(7538):232.

Moser, E. I., Moser, M.-B., and McNaughton, B. L. (2017). Spatial representation in the hippocampal formation: a history. *Nature neuroscience*, 20(11):1448.

Solstad, T., Moser, E. I., and Einevoll, G. T. (2006). From grid cells to place cells: a mathematical model. *Hippocampus*, 16(12):1026–1031.

Stemmler, M., Mathis, A., and Herz, A. V. (2015). Connecting multiple spatial scales to decode the population activity of grid cells. *Sci. Adv.*, 1(11):e1500816.

Stensola, H., Stensola, T., Solstad, T., Frøland, K., Moser, M.-B., and Moser, E. I. (2012). The entorhinal grid map is discretized. *Nature*, 492(7427):72–78.

Stensola, T., Stensola, H., Moser, M.-B., and Moser, E. I. (2015). Shearing-induced asymmetry in entorhinal grid cells. *Nature*, 518(7538):207.