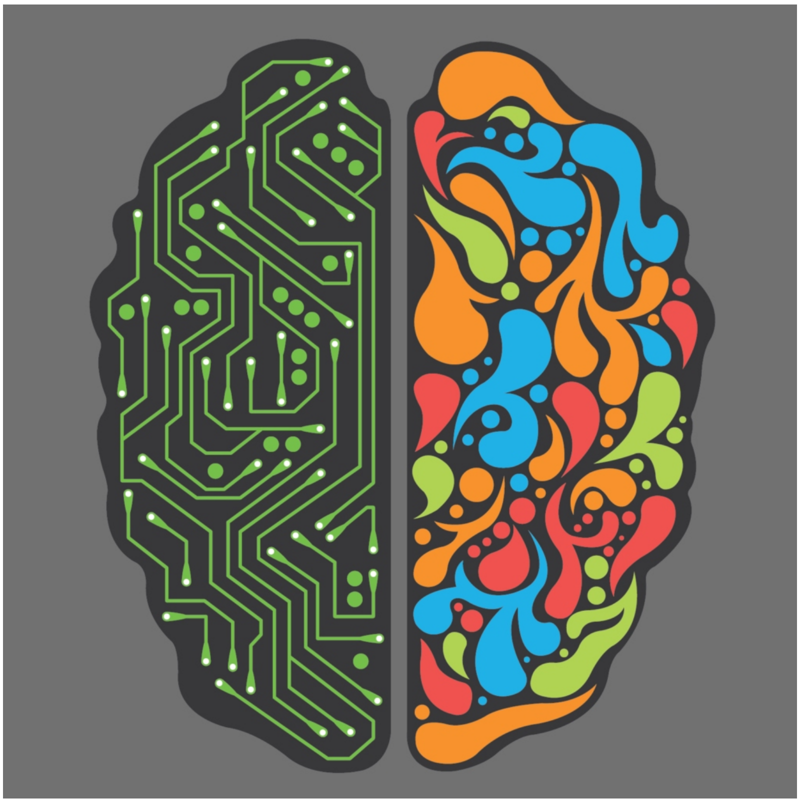
On the Emergence of Biologically-Plausible Representation in Recurrent Neural Networks for Navigation

Version of August 4, 2019



Daan Zeeuwe

On the Emergence of Biologically-Plausible Representation in Recurrent Neural Networks for Navigation

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Daan Zeeuwe
born in Amsterdam, the Netherlands

TU Delft

Interactive Intelligence Research Group
Faculty EWI, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Cover picture: While a significant portion of researchers focuses on the mathematical inspiration and optimization of neural networks, somehow we increasingly overlook biological inspiration and the cognitive science concepts which are part of the past, present and future of artificial intelligence.

On the Emergence of Biologically-Plausible Representation in Recurrent Neural Networks for Navigation

Author: Daan Zeeuwe
Student id: 4624750
Email: d.zeeuwe@student.tudelft.nl

Abstract

Biologically plausible representations have been found to emerge in particular recurrent neural networks when training on path-integration [1, 2]. This report explores factors influencing the occurrence of entorhinal-like representations in recurrent neural networks. Reproducing simplified models from existing studies and created a hybrid model to explore additional factors, including the input features, structural properties, and regularization techniques in recurrent neural networks. Additional experiments evaluate the difference in training performance when entorhinal-like representations are introduced to a recurrent neural network. This report also assesses existing and experimental visualization techniques in their ability to visualize the performance and representation of recurrent neurons. While some experiments show specialized representations, mostly due to regularization; none of the experiments showed typical entorhinal-like representation. These results show how sensitive the emergence of biologically-plausible representations is to network conditions and training procedure, casting some doubt on the generality of the conclusions proposed in earlier work.

Thesis Committee:

Chair:	Prof. Dr. C. M. Jonker, Faculty EWI, TU Delft
University supervisor:	Dr. J. D. Broekens, Faculty LIACS, Leiden University
Committee Member:	Dr. Ing. J. Kober, Faculty 3mE, TU Delft

Preface

In the summer of 2018, I was looking for inspiration on how to improve artificial intelligence and robotics. By coincidence, I stumbled upon textbooks on psychology and neuroscience and decided to look into the material. At first sight, it seemed fascinating, but I was not able to apply this knowledge directly to develop more sophisticated models. During the fall I came across the papers by DeepMind and especially the article by Banino et al. on "Vector-based Navigation using Grid-like Representations in Artificial Agents". Their work relied on psychological theory and neuroscientific models to perform artificial navigation. The work by Banino et al. and DeepMind showed me how cognitive science inspiration could apply to artificial intelligence. Their research inspired me to look beyond the current capabilities of artificial intelligence and use other cognitive science disciplines to provide a diverse balance for developing modern robotics.

Most popular models for robot perception, localization, and actions rely on mathematical models, which carry implicit constraints on performance and generality. However, developments surrounding recurrent neural networks and deep reinforcement learning could finally show advances in human-like robotics. This development would enable the advancement of cognitive robotics to look towards incorporating efforts of cognitive science instead of building exclusively on mathematical constructs.

This thesis explores the papers by Banino et al. and Cueva and Wei, who made similar claims on the emergence of entorhinal-like representation in recurrent neural networks. Currently, it is unknown why, how, and when this representation emerges in neural networks. Understanding the underlying dynamics of entorhinal-like representation, especially in recurrent neural networks, could be the first step to allow the artificial intelligence community and neuroscience researchers to collaborate. Creating a new research community called "Artificial Neuroscience" or "Neuroscience-inspired Artificial Intelligence".

The problem with current artificial intelligence and neuroscience research is that the gap between the research communities is quite large. Both artificial intelligence and neuroscience papers require a strong academic background in each research field. Also, the diffusion of artificial intelligence to and from neuroscience is minimal, which enlarges the gap even further. Hopefully, continued development and shared collaboration between artificial intelligence and neuroscience could set an example for reuniting cognitive science disciplines.

This past year I have been able to interact with several excellent professionals and academics, which have inspired and motivated me to continue my path towards developing human-like robotics. I want to thank my friends and family for their continued support, and they inspire me to work harder every day to realize my vision for the future. Moreover, I want to especially thank Joost Broekens for his persistence and guidance during the research, development, and writing process of this thesis.

Daan Zeeuwe
Delft, the Netherlands
August 4, 2019

Contents

List of Figures

Chapter 1

Introduction

Mobile robotics relies on artificial navigation to transition towards intelligent autonomous behavior [14, 15]. Modern robot navigation research uses deep learning approaches, such as deep reinforcement learning and recurrent neural networks to navigate autonomously [16, 17, 18, 19, 20, 21, 22]. While this research is promising, some researchers have expressed their concern on the generality of the proposed methods [23, 24, 25, 26]. An exciting novel approach tries to alleviate these problems by relying on biological inspiration to develop artificial navigation [1, 2, 27, 28].

Artificial navigation algorithms can use inspiration from the psychological construct of spatial cognition [29, 30]. Spatial cognition is dominated by the cognitive map theory, which originated in the study by E. C. Tolman and colleagues from 1948 [31]. The cognitive map theory proposes a shared mental representation underlying spatial knowledge, such as localization, navigation, and planning [3]. Tolman's findings inspired O'Keefe and other researchers to associate the cognitive map with the hippocampus [32, 33, 34]. Research into the neuroscientific origins led to the discovery of: grid [6], place [7], head-direction [10], speed [9] and border cells [8] (for more information on cell activation patterns see Figure 2.1). Those cell archetypes primarily occur in the entorhinal cortex and relay information to the hippocampus. Therefore, related cell representations are typically called entorhinal-like representation [1]. Each cell type contributes to the diverse geometric representation crucial for navigation [35]. For more information on the basic spatial representations of grid and place cells, see Figure 1.1.

Recent studies have drawn parallels between emergent representations in artificial neural networks and entorhinal-like representations [36, 37]. More specifically, studies by DeepMind [1] and Columbia University [2] have published results regarding the emergence of entorhinal-like representation in recurrent neural networks. Both studies [1, 2] report on comparable conceptual representations in their neural networks, despite model differences between the studies. The research groups used different model architectures and regularization techniques, which constrains deep learning networks to prevent the model from significantly overfitting on the input data. The training differences caused both studies to reflect adversely on the role of regularization and its link to causing emergent represen-

[1] https://www.nobelprize.org/prizes/medicine/2014/press-release/

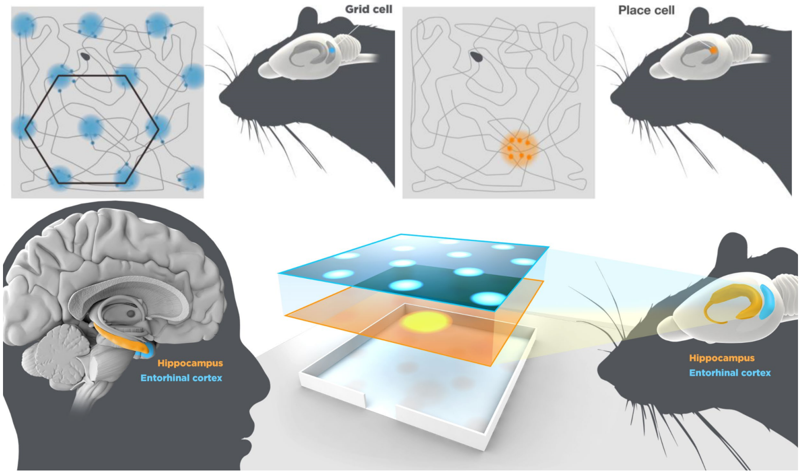


Figure 1.1: **Simplified representation of grid- and place cell activation**
Grid- and place cell activity occurs in the entorhinal cortex and hippocampus, respectively. A place cell (orange) is only active in one location. Whereas grid cell (blue) activation patterns resemble hexagonal grids spanning the environment. While most research is done on rats moving in small environments, the same representation has been found to relate to memory function and navigation in humans [3, 4, 5]. This image was taken from the Nobel Price website[1]without consent of the publisher.

tation. Thus, the results from the research groups lack a unified explanation regarding the factors underlying emergent representation. Additionally, the studies do not address the performance difference associated with introducing entorhinal-like representation in recurrent neural networks. A unified consensus on the emergence and performance of entorhinal-like representation is essential for determining the value of biologically-plausible representation in artificial neural networks.

1.1 Motivation

Not only is entorhinal-like representation involved in spatial navigation, but also non-spatial tasks, such as temporal context [38], social relations [39], and imagination [4]. Thus, research into entorhinal-like representation can inspire solutions for memory-, conceptual-, and planning problems [40]. Moreover, entorhinal-like representation is associated with general problem-solving [29, 41, 42, 43]. Identifying the potential of entorhinal-like representation in deep learning could help to develop new theories for solving artificial intelligence problems [43].

One of the most challenging deep learning problems relates to cognitive-behavioral tasks [44]. Taking inspiration from psychology and neuroscience research helps tackle complex problems [45]. Most popular strategies for deep learning (convolutional neural networks [46], recurrent neural networks [37], and deep reinforcement learning [47]) are based on inspiration from neuroscientific observations. Machine learning researchers can learn from other research disciplines to develop new and improved algorithms [48]. Taking inspiration from cognitive science research is not aimed at replicating biologically-plausible algorithms, but at diversifying existing approaches for solving artificial intelligence problems [49].

Neuroscience and artificial intelligence can both benefit from reciprocal research on biologically inspired algorithms [50, 51]. For example, patients with Alzheimer's disease experience problems remembering the past, and planning for the future [52, 53]. Developing theories on the interaction between grid cells and other entorhinal-like representations can lead to new insights into Alzheimer's disease [4]. These theories require extensive research on the interaction between the entorhinal cortex and hippocampal cells. Obtaining this type of information in an artificial setting could become more cost-effective compared to testing with animals in the coming ten years [54].

One of the applications that can benefit from biological inspiration is artificial navigation. A large portion of artificial navigation applications is dependent on Simultaneous Localization And Mapping [55]. These applications require constructing new models for unknown locations or revising the model in dynamic environments. Inspiration for autonomous navigation can come from models using entorhinal-like representations, which are theorized to provide context-independent navigation [56]. Additional features of entorhinal-like representation include uncertainty minimization [57] and path-integration [28], which integrates historical data about the direction and speed to represent the current position. Both error correction and localization are necessary for planning in dynamic environments. Thus, future models using entorhinal-like representation in neural networks could potentially replace Simultaneous Localization And Mapping to develop generalizable models for spatial navigation [58].

1.2 Research scope

The purpose of this study is to investigate factors responsible for the emergence of entorhinal-like representation in recurrent neural networks and determine its training performance.

This research effort aims to achieve three specific goals left by previous studies. These goals include: (1) determine the ability to reproduce reported results using models that are similar to the ones used in previous studies, (2) investigating which factors are responsible for the emergence of entorhinal-like representation, and (3) quantifying the performance difference of entorhinal-like representation in recurrent neural networks.

The executed experiments try to answer the following related research questions:

1. Is it possible to replicate emergent entorhinal-like representations using non-essential simplifications of earlier work models?

2. What factors can influence the emergence of entorhinal-like representation in recurrent neural networks?

    a) Are input factors, such as compound features, input noise, or ray-tracing, essential for generating entorhinal-like representation?

   b) Does optimizing structural properties, including the recurrent layer architecture, learning rate, or training optimizer, result in emergent entorhinal-like representation?
   c) Can regularization promote entorhinal-like specialization among neurons, for instance, in the case of recurrent regularization, regularization losses, or Dropout?

3. Does entorhinal-like representation in recurrent neural networks improve path-integration performance?

A limiting factor of this study is that the spontaneous emergence of entorhinal-like representation in recurrent neural networks has no theoretical background. Without this background, it is almost impossible to generalize the conclusions of this study to other network configurations. This study is part of an exploratory research effort since the amount of related research is limited, and the research topic is entangled with various under-researched and challenging problem domains, such as neural network analysis and biologically-plausible representation in neural networks. Exploring various factors requires extensive experimentation, which commands vast resources. This limitation causes experiments to use a slightly higher learning rate compared to earlier work to converge quicker. More experimental setups are explored due to the ability to train faster, but still reach converged performance with the different network models.

Chapter 2

Literature Review

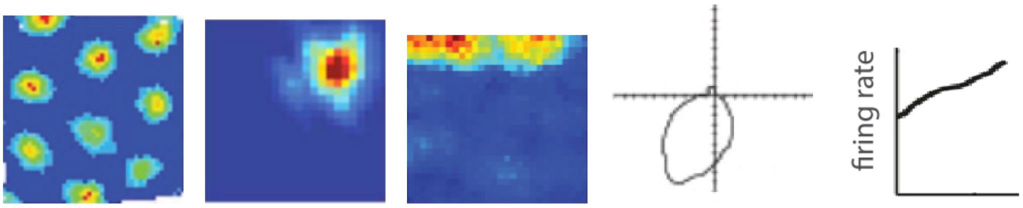
Related research and recent developments regarding the emergence of entorhinal-like representation in recurrent neural networks are explored and discussed in this literature review. First, introducing necessary background information on the function and diversity of entorhinal-like representation, and then follow up on the computational basics of recurrent neural networks. The related work section discusses developments surrounding neural network analysis, regularization techniques, and recent work integrating these techniques to observe artificial entorhinal-like representation.

2.1 Background

The concept of emerging entorhinal-like representation originates from cognitive map theory in psychology. The cognitive map theory motivated researchers to investigate the biological basis of spatial cognition, which led to discoveries in the entorhinal cortex and hippocampus regarding specialized spatial and non-spatial cells. Following these discoveries, neuroscience researchers proposed different theoretical models explaining the behavior of individual cells [56, 59, 60, 61, 62] and spatial cognition in general [58, 63, 64, 65]. Lately, researchers turned to deep learning models to replicate entorhinal-like representation using recurrent neural networks and regularization techniques. The following sections aim to provide a sufficient foundation towards understanding the recent interest in artificial entorhinal-like representations for spatial navigation.

2.1.1 Entorhinal Cells

The discovery of grid cells in 2005 [12] encouraged new research opportunities for investigating the behavior of association cortices (parietal-, temporal-, and occipital lobes) [66]. More specifically, the internal representation of navigation in the medial entorhinal cortex and hippocampus [66]. The neuronal representation of space in the medial entorhinal cortex is quite diverse, consisting of: grid cells, border cells, head-direction cells, and speed cells [67]. However, the hippocampus is mainly occupied by place cells [7], for more information on the representation of spatial and non-spatial cells, see Figure 2.1.



(a) Grid cell (b) Place cell (c) Border cell (d) Head-direction cell (e) Speed cell

Figure 2.1: **Entorhinal and Hippocampal cell diversity**
Grid-, place- and border cells are characterized by a particular spatial activation pattern. Additionally, the head-direction cell is activated by the facing direction of the animal, and the speed cell scales linearly to the running speed. The image was taken from [2] without consent of the publisher.

The underlying spatial representation in the brain relies on grid-, place-, and border cells; these geometric cell types allow self-motion cues to keep stability in the absence of visual cues [6]. Grid cells occur in the entorhinal cortex and are characterized by a hexagonal activity pattern, which can represent large environments and are not limited by the boundaries of the environment [68]. The hexagonal activation function of grid cells is proposed to provide the basis for goal-based navigation [35] and also facilitates an independent context source for path-integration [69]. Various models have used inhibitory competitive network interactions which are believed to underlie grid cell activations [66]. Place cells are hippocampal cells, and their activity is associated with a distinct location [70, 71]. The place cell is theorized to emerge from the summation of various grid cell scales [72]. The place representation could provide conjunctive information [29] since place cells are also necessary for the periodic firing of grid cells [6]. Additionally, place cells can remap their local activity patterns, allowing a place cell to represent distinct places in different contexts [10]. The last spatially active cell type is the border cell, which encodes the boundaries of the environment [8]. The activation of the border cell depends on the barriers, obstructions or other movement restrictions [70, 73, 74].

Additionally, speed cells and head-direction cells encode biological stimuli to support spatial representations [59, 66]. Head-direction cells are only active within a small angular range and follow a Gaussian-like activation pattern [10]. The head-direction cells are responsible for providing a directional signal during navigation [10]. Speed cells express a linear relationship between instantaneous velocity and cell activity [9]. This velocity-based activity allows speed cells to interact with head-direction cells and grid cells to perform path-integration [75, 76, 77]. The entorhinal-like cell representations serve to represent places, head-direction, speed, and grid-like structures. These specializations encode statistical regularities in spatial navigation [78, 79], and possibly encoding abstract representations of state spaces [80, 81]. The abstract representations could generalize to other aspects of cognition [38, 82] not necessarily linked to spatial domains [43, 83, 84, 85].

Appendix A contains additional information about entorhinal-like representation and the significance of other cognitive science disciplines for spatial navigation.

2.1.2 Recurrent Neural Networks

Recurrent neural networks (RNN) enable sequence processing and temporal prediction through learning the temporal context representation [86]. Natural language processing uses recurrent neural networks to process and predict words; but recurrent neural networks also improve performance for general temporal prediction problems [87], such as path-integration [22]. The recurrent architecture predicts the output based on the recurrent state and input data. However, recurrent neural networks can experience exploding or vanishing gradient problems during training. These gradient problems cause excessive weight oscillations preventing the network from learning long-term associations [88, 89].

Gated recurrent architectures [90] and regularization techniques [91, 92, 93] can mitigate most of the gradient problems. Gated recurrent networks, including the long short-term memory (LSTM) [94] and gated recurrent unit (GRU) [95], prevent the vanishing gradient problem through multiplicative gate units [94]. Additionally, the clipping regularization technique limits the gradients by scaling down the gradients if they are above a certain threshold [88], restricting exploding gradient problems.

Multiplicative gates learn to control the flow of information to and from the recurrent cell state. An LSTM has three trainable gates: forget gate, input gate, and output gate. GRU's have a similar setup compared to LSTM's, but only have a reset gate and update gate to revise the hidden state [96]. The forget or reset gate learns what proportion of the previous hidden state is preserved before integrating the current input and predicting the output. The input or update gate incorporates the input data with the recurrent state. The output gate (LSTM only) learns to convert the recurrent state and input features to produce an output vector. For more on the architectural differences between LSTM and GRU, see Figure 2.2.

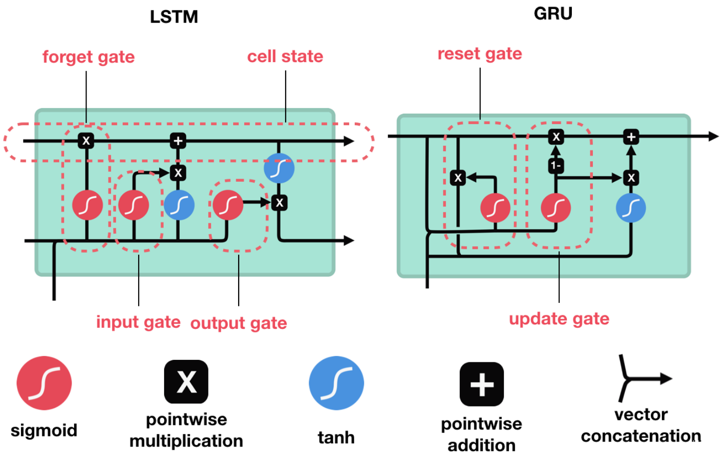


Figure 2.2: **Overview of LSTM and GRU cell**
LSTM's have three gates, two of which control the state, and the output gate controls the output state. The GRU uses two gates to adjust the recurrent state, which is also the output of the cell. The images were taken from a blog[1]without the consent of the publishers.

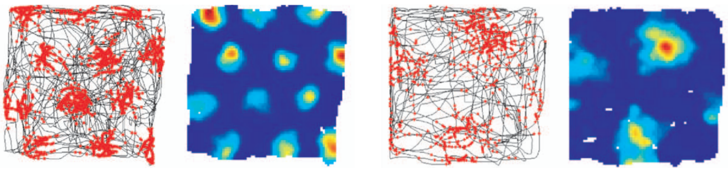


Figure 2.3: **Grid cell spatial activation measure**
Each entorhinal grid cells has a scale associated with the activation pattern [11]. Experiments record the neural activity spikes (representing red dots) during navigation. The activity spikes are processed using a Gaussian filter to create a spatial activity rate map, which is used for analysis. The image was taken from [7] and [12] without the consent of the publishers.

2.2 Related work

This section first introduces neural network analysis approaches and regularization techniques. This introduction enables discussing the regularized recurrent neural network models necessary for experimentation.

2.2.1 Recurrent Neural Network Analysis

Visualizing neural networks can help with the scientific understanding of grid cells by investigating the occurrence and frequency of different neuron representations [97, 98, 99]. Especially time-based visualization techniques are necessary for understanding the gradual emergence and progressive interaction between recurrent neurons.

The standard approach for visualizing training performance in neural networks is loss visualization. Plotting the evolution of loss throughout the training process can show over- and under-fitting performance using the difference between the testing and training error. These observations are necessary for adjusting the hyper-parameters or trying out different network approaches. However, sudden spikes or significant variance in the loss cannot be directly explained solely based on differences in the model setup [100]; therefore, researching other interpretable analysis and visualization techniques is necessary [101].

Neuroscientific studies regarding entorhinal and hippocampal cells use spatial or directional activation plots [6, 80] to visualize entorhinal-like representation. The activity plots average and filter the neural activity of animals to generate a continuous activation plot, see Figure 2.3. Artificial intelligence researchers use this approach to visualize the average spatial- and direction-dependent activity of recurrent- and linear layer neurons [1, 2].

Artificial intelligence researchers have significantly expanded the diversity of neural network visualization techniques over the past decade [102]. Some visualization approaches use input-dependent activation [103, 104] to characterize the activity of neural networks. These methods visualize the preferred activation patterns of trained kernels in convolutional neural networks. Other researchers apply projection-based techniques [96, 105] to

[1] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

cluster neural activity patterns. For example, dimensionality-reduction enables visualization of clustered neural activity. This technique could help classify the different emerging entorhinal cell types and determine when these cells initially emerged. This approach would be helpful when entorhinal-like representation emerges consistently in recurrent neural networks but is currently not directly applicable.

Currently missing from the analysis techniques is a straightforward approach to visualize the interaction between neurons. An explanation for this deficiency is that neural networks suffer from unexplainably and incomprehensibility [106]. This phenomenon makes interaction analysis complicated since it requires tracking, relating, and visualizing dynamic high-dimensional states in neural networks. The ability to visualize causal links between neurons would improve our understanding of neural interactions in the recurrent neural network, necessary for the analysis of emergent entorhinal-like representation.

Despite widespread neural network experimentation in the past thirty years, visualizing and understanding the behavior of deeper neural networks have only recently been systematically studied [107]. Despite this recent progress, researchers have not yet established the essential experimentation tools to explain the role of joint activity in neural networks necessary for understanding emergent entorhinal-like representation in recurrent networks.

2.2.2 Regularization Techniques

Regularization techniques reduce the ability of a learning model to overfit on the training data. Generally, regularization is any supplementary technique aimed at improving generalization performance during training [108].

The ability to generalize between the training and test data revolves around the ratio between the number of model weights and the number of training samples [109]. This concept of generalization relies on the bias-variance trade-off [110]. The model performance trade-off for reducing the loss variance through regularization will cause an increase in the loss bias and vice versa. Balancing the loss variance and bias of the model improves generalization between the training and test set. Managing the performance trade-offs is essential when there is not enough training data available to offset the number of free variables in the training model [109].

A neural network uses the total loss $L(w)$ of the model to optimize the network weights w, and this total loss consists of two components: prediction loss and regularization loss. In recurrent neural networks, the prediction loss usually calculates the difference between the target output and predicted values for each time-step t in the horizon T. A common approach for calculating the prediction loss is the least-squares metric. Neural network regularization is provided through the regularization loss. This loss is defined by a regularization metric $R(w)$ and is scaled by a regularization constant λ to control the balance between the prediction loss and regularization loss.

$$L(w) = \frac{1}{T}\sum_{t=0}^{T}(y_t^{pred} - y_t^{target})^2 + \lambda \cdot R(w)$$

The most common regularization approach is called weight decay, which penalizes the weights of a neural network layer. Large weights are often the result of over-fitting on the training data, thus preventing larger weights can improve generalization performance. Another common regularization approach is called Dropout [108]. Dropout temporally silences random units in a hidden layer [111, 112, 113]. The Dropout approach samples from a Bernoulli distribution to decide which features are silenced [115, 117]. Standard Dropout silences around half of the features each training iteration [114, 116]. This approach reduces the co-adaptation between neurons necessary for generalization [118].

2.2.3 Recurrent Neural Network Models

Banino et al. [1] reported on the emergence of grid-like representations using a recurrent neural network in 2016. Training their model follows three steps. First, the spatial trajectory is preprocessed to generate simulated place and head-direction cell activations. Second, the recurrent cell is initialized with the simulated place and head-direction cell activations from the initial position of the trajectory. Third, the model is then trained to predict the activation of simulated place and head-direction cells throughout the trajectory based on the velocity, direction, and processed visual features. Banino et al. developed a proprietary vision module to process the first-person perspective of the agent traveling through their DeepMind Lab simulator [119].

The successive layer between the recurrent cell and the output of the network, called the bottleneck layer, was regularized using Dropout and developed grid-like representations. Banino et al. concluded that Dropout was responsible for the development of emergent grid-like representation. Banino et al. argued that the introduction of noise through Dropout could lead to error-correction provided by emergent entorhinal-like representation. For more information on the Banino et al. model see Figure 2.4.

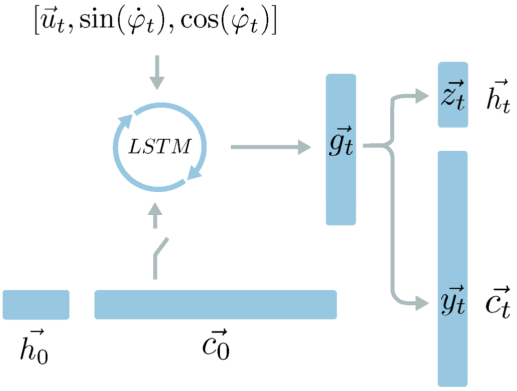


Figure 2.4: **Banino et al.'s architecture**
The LSTM is fed input data consisting of speed $\overrightarrow{u_t}$ and the split directional velocity input $\sin(\dot{\phi}_t)$ and $\cos(\dot{\phi}_t)$. The recurrent cell is initialized using the initial place cell $\overrightarrow{c_0}$ and head direction $\overrightarrow{h_0}$ distributions. The output of 128 LSTM cells is fed to a linear bottleneck layer $\overrightarrow{g_t}$ consisting of 512 neurons. The bottleneck layer output is split into two linear layers ($\overrightarrow{y_t}$ and $\overrightarrow{z_t}$) responsible for predicting the 256 place cell $\overrightarrow{c_t}$ and 12 head-direction cell $\overrightarrow{h_t}$ activations. The image were taken from [1] without consent of the publishers.

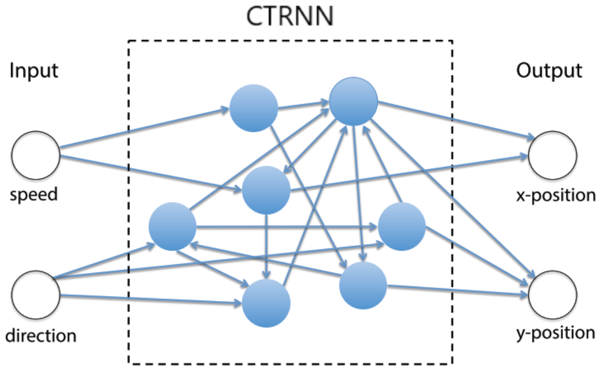


Figure 2.5: **Cueva and Wei's architecture**
Cueva and Wei's model trains their 100 recurrent neurons by using the speed and direction features to predict the x- and y-position of the agent's trajectory. The recurrent cell uses a Continuous-Time Recurrent Neural Network (CTRNN). The image was taken from [2] without the consent of the publishers.

Cueva and Wei followed up with their own recurrent neural network model in 2018, claiming biologically inspired regularization can support emerging entorhinal-like representation [2], see Figure 2.5. The biologically inspired regularization technique uses a metabolic incentive to penalize large recurrent weights. Their proposed method is called the Metabolic constraint and applies weight decay on the input weights, output weights, and the state of the recurrent neural network. The resulting representation shows grid-, border-, and band-like activity, and is consistent with formal models of entorhinal-like representation [120]. Cueva and Wei claim that the Metabolic constraint can create efficient coding in recurrent networks. The efficient coding hypothesis is generally associated with the compressed representation for sensory coding, which enables efficient and optimized signaling in the brain [22, 121, 122, 123, 124]. Recurrent weight decay in the Metabolic constraint simulates similar restrictions on neural communication, which could force neurons to communicate with other neurons more effectively [125].

Both Banino et al. [1] and Cueva and Wei [2] apply regularization to obtain entorhinal-like representation. However, both vary in argumentation why regularization is the main reason for the emergence of entorhinal-like representation. Both the Dropout approach by Banino et al. [1] and the Metabolic constraint by Cueva and Wei [2] are based on biologically-plausible mechanisms for emerging representation. Reducing the reliance on other neurons through Dropout leads to sparse representation [114]. Also, the Metabolic constraint on the recurrent weights and state could force the recurrent neurons to specialize and create sparse activity patterns. Both approaches seem equally plausible for causing emerging entorhinal-like representation. Thus it is surprising why both studies reflected adversely on the applied regularization techniques and its role in emergent entorhinal-like representation.

2.2.4 Summary

The construct of spatial cognition relies on psychological theory, which ultimately led to neurobiological observations and computational models in neuroscience. Modern artificial intelligence research can use cognitive science inspirations to investigate complex problems, such as modeling different cognitive abilities, including spatial navigation [34].

The results from Banino et al. [1] and Cueva and Wei [2] point to biologically plausible representation in recurrent neural networks during path-integration. The studies reported emergent entorhinal-like representation when regularizing their recurrent neural network. However, the studies have not systematically linked which aspect of regularization is responsible for the emergent entorhinal-like representation, nor explored the performance difference of entorhinal-like representation in recurrent neural networks. Thus, the exact role of regularization and its relation to biologically plausible representation is still unclear.

The following chapter provides a closer look at the experimental setup and analysis techniques used in this study.

Chapter 3

Methodology

The goal of this thesis is to investigate factors responsible for the emergence of entorhinal-like representation in recurrent neural networks and determine its influence on training performance. Preparing this study requires (1) data acquisition through simulation, (2) a configurable base model used for performing experimentation, (3) visualization techniques, and (4) experimental test variables.

3.1 Simulation Data

Banino et al. [1] used rat-like motion strategies to train their deep learning model. Their data sampling approach included a realistic rat-like motion strategy [126] to generate biologically-plausible velocity and directional data. Cueva and Wei used Brownian-like motion models to simulate mostly straight stochastic movement data. Both strategies [1, 2] focus on randomly moving within the environment and avoiding wall collisions. However, the proposed simulation approach in this thesis uses a 2D attractor-based approach. While there are many 3D navigational simulators made from game engines [119, 127, 128, 129, 130, 131], basic path-integration only requires generating two-dimensional trajectories; and this 2D setup allows for faster simulation of trajectories. The 2D attractor-based approach generates a position (attractor), which will be the navigational target for the agent. The simulator generates a new attractor whenever the agent reaches the attractor position.

The simulated agent creates a linear trajectory towards the attractor to simulate goal-based behavior. If the agent comes close to the border, then the speed is reduced, allowing the agent to turn away from environmental boundaries. The simulated trajectories provide velocity, direction, and positional data, which are used for training and testing the recurrent neural network models. An example subset of the trajectories is displayed in Figure 3.1.

A training dataset of 100.000 trajectories is generated before experimentation to avoid data generation during the training process. The test set for each experiment is generated separately from the prerecorded trajectories dataset to avoid overlapping training and test sets. The training and testing process uses small batches of 100 trajectories to determine the total network loss.

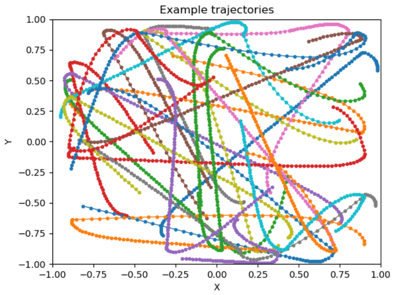


Figure 3.1: **Example batch of 25 trajectories**
The agent moves towards an attractor to simulate a 100-step trajectory.

3.2 Base Model

TensorFlow was used for this project to implement the neural network models. The same base model is used for all training experiments and adapts to each experimental setup. The base model was constructed using TensorFlow 1.12 with NVIDIA GPU support; this enabled significant recurrent neural network training speedup. Each experiment is characterized by a configuration file, which indicates the recurrent cell, intermediate neural layers, and training procedures used. For an elaborate discussion of the base model details, see appendix Appendix C.

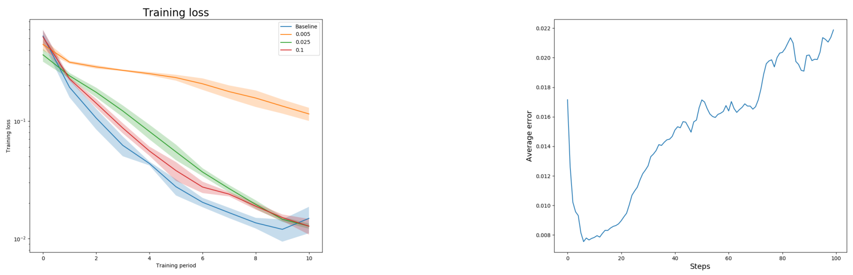
3.3 Visualization Techniques

The upcoming data analysis methods elaborate on traditional- and experimental visualization techniques. Each technique enables studying either the training performance or representation of artificial neurons in neural networks.

Analysis of spatial and non-spatial plots are indexed using a Cartesian coordinate system. Neurons will be referenced using a coordinate $[x,y]$, where x and y are the indexes of the neuron in the plot. All neurons have non-negative indexes, and the top-left neuron is referred to as neuron $[0,0]$.

3.3.1 Loss

Measuring the training performance of neural networks is usually done through least-squares loss (Figure 3.2a). This loss technique calculates the squared difference between the predicted output and target output [109]. While loss visualization compares the training performance of different training setups directly, it does not provide insight into neural activity and representation.



(a) **Loss plot**
The loss plot visualizes the training loss differences between different models. The x-axis depicts the number of training iterations, and the y-axis plots the training or testing loss.

(b) **Average performance plot**
The time-dependent loss is calculated using the average loss for each trajectory time-step. Both the initialization performance and linear error accumulation are important for performance analysis.

Figure 3.2: Basic performance plots

3.3.2 Average Performance

One way to expand on the loss plot is to visualize the average loss of the models through time. The average performance plot calculates the average difference between the predicted output and target output for each time-step in the trajectory (Figure 3.2b). Visualizing the time-dependent loss helps analyze the network capacity by observing both the initial performance (the first few time-steps) and linear error accumulation through time.

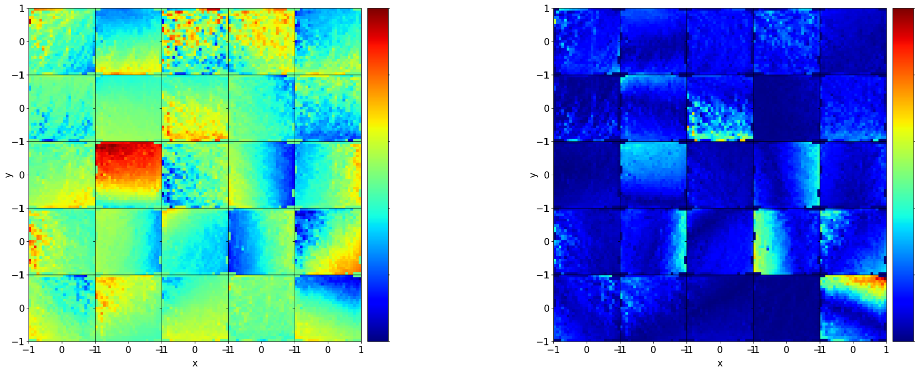
3.3.3 Spatial Activity

Various studies regarding entorhinal-like representation investigate neural representations using spatial activity plots [35]. Spatial activity plots (Figure 3.3a) visualize the average activity for each neuron throughout the environment. This visualization technique uses spatial bins to group the activity of each neuron within a spatial environment and calculate the average activity per spatial bin. Observations of stable spatial activation patterns can lead to subjective hypotheses on the neural activation strategy.

3.3.4 Spatial Influence

A new experimental impact analysis method proposed in this thesis quantifies the influence of neurons on other neural layers. The spatial influence plot (Figure 3.3b) calculates the scaled activation for each recurrent neuron using the weights of the subsequent linear output layer.

$$u(t) = \tan(x(t)) \qquad\qquad I(t) = \sum_{i=0}^{N_{linear}} |u(t) \cdot W_{i\cdot}{}^{linear}|$$



(a) **Spatial activity plot**
The 5x5 plot cells visualize 25 distinct neurons. Each plot cell visualizes the average spatial activity for an artificial neuron. The spatial activity is measured by binning the neural activity $u(t)$ over the entire environment $E \in (-1,1)^2$. Each plot cell pixel represents a spatially binned average of the neural activity.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. The spatial influence plot bins the neural influence $I(t)$ over the entire environment $E \in (-1,1)^2$. Each plot cell pixel represents a spatially binned average of the neural influence.

Figure 3.3: Spatial activity plots

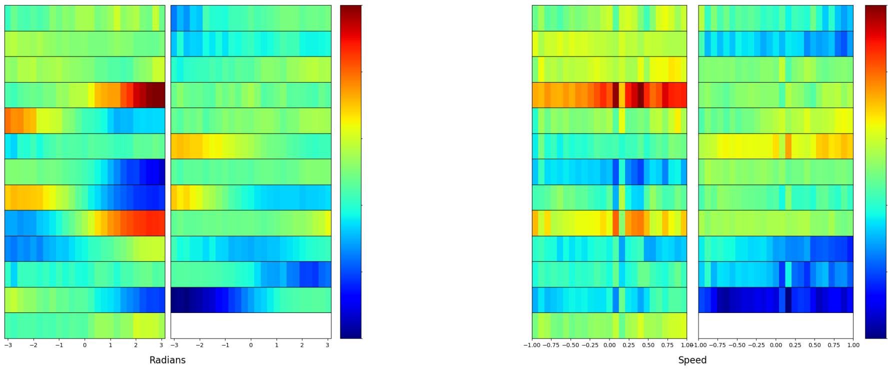
Multiplying the recurrent neural activity $u(t)$ with the individual weights from the successive output layer determines the absolute neural influence $I(t)$ for predicting the output. This method measures the contribution of representation by highlighting useful activation patterns in recurrent neurons.

3.3.5 Non-spatial Activity

Non-spatial activity plots were used by Cueva and Wei [2] to quantify the speed and directional tuning of neurons. This approach measures the dependence on input data through velocity- or direction-dependent activity per neuron (Figure 3.4a and Figure 3.4b). The non-spatial activity plot bins the neural activity $u(t)$ for each input variable to estimate the average activity per neuron. Visualizing velocity- and direction-dependent activity helps to analyze the underlying neural activation strategy from a non-spatial perspective.

3.3.6 Maximum Activity

Measuring the functional specialization of different neurons is difficult even for smaller networks. Analyzing the spatial activation patterns in a network of 25 neurons is already cumbersome. Therefore, this thesis introduces a maximum activity plot, which can highlight spatially distinct activity for the most active neuron per visualization bin (Figure 3.5a). This plot provides a quick insight into the underlying activation patterns of neurons, especially in the case of extensive neural networks.



(a) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. The activity plot bins the neural activity $u(t)$ over the entire directional range $d \in (-\pi, \pi)$. Each plot cell pixel represents a directionally binned average of the neural activity.

(b) **Velocity activity plot**
Each plot cell visualizes the average velocity-dependent neural activity. The activity plot bins the neural activity $u(t)$ over the entire velocity range $v \in (0, 1)$. Each plot cell pixel represents a velocity-based binned average of the neural activity.

Figure 3.4: Non-spatial activity plots

3.3.7 Trajectory Performance

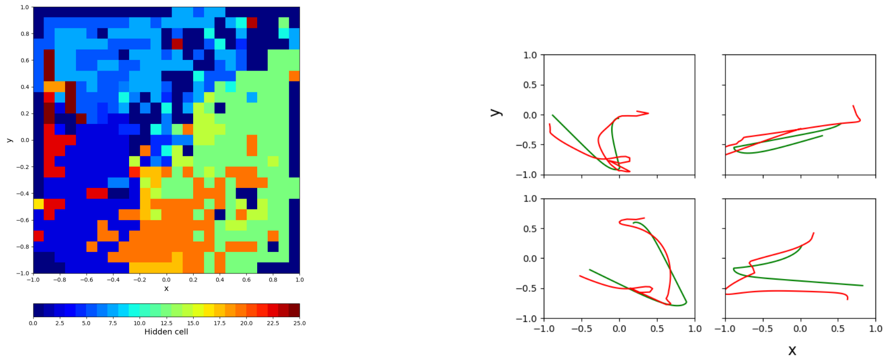
Another approach for analyzing the training performance is through the trajectory performance plot, which visualizes the differences between the predicted and target trajectories (Figure 3.5b). The overlaid trajectories provide additional analysis through subjective evaluation of the trajectory consistencies. Excessive trajectory perturbations would indicate either under-fitted performance or could point to exploding gradient problems [94]. Both observations can highlight learning difficulties, which can notify Deep Learning researchers to tune their network architecture.

3.4 Experimental Factors

The investigated factors split into three categories: (1) input features, (2) structural properties, and (3) regularization techniques. The input features discuss alternative strategies for processing input, which are used to train recurrent neural networks. The structural properties include recurrent layer configurations and training procedures. Lastly, the regularization techniques include traditional and experimental approaches for imposing constraints on recurrent networks.

3.4.1 Input Features

Input features are essential for deep learning network performance. Especially input features with multiple modalities aid the feature extraction process of neural networks [132, 133]. The improved feature extraction process enables networks to find better hierarchi-



(a) **Maximum spatial plot**
Visualizes the most active neuron per spatial bin. Each color represents a different neuron, see the legend on the bottom of the plot. The maximum spatial activity plot helps to quickly analyze the spatial specialization among neurons.

(b) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. Visualizing the trajectory differences helps to analyze performance problems. The shape and smoothness of the trajectories can signal training problems.

Figure 3.5: Additional performance plots

cal features for optimizing performance [134]. Proper multi-modal input features could be necessary for emerging entorhinal-like representation.

The default path-integration features consist of speed and directional input. The following approaches are designed to explore additional input features: (1) data reorganization, (2) adding noise, and (3) ray-tracing.

Starting with the data reorganization method called split direction proposed by Banino et al. [1]. This approach separates the angular velocity into two separate features. The x- and y-components are extracted from the agent's angular velocity $\dot{\phi}$ to reduce the complexity of path-integration.

$$x = cos(\dot{\phi}) \qquad y = sin(\dot{\phi})$$

The second approach adds noise to the input, which could provide the necessary conditions for generating entorhinal-like representation according to research by Banino et al. [1]. They claim that Dropout introduces noise to their network, which they noted was essential for emerging entorhinal-like representation. This technique is introduced to evaluate the noise introduction hypothesis and its relation to emergent entorhinal-like representation.

The latter approach, called ray-tracing, introduces vision-like features. Ray-tracing uses a color-coding system for the walls around the agent. The color-coding system creates a biologically-inspired encoding for each location in the environment. Each ray-trace is a separate feature that provides a readout of the wall color. The color-coding process is visualized in Figure 3.6.

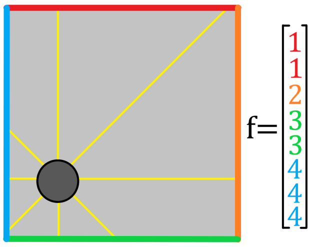


Figure 3.6: **Ray-tracing calculation**
The agent in gray observes a fixed number of sampled ray-traces from the environment. The ray-tracing method uses a wall color-coding system to reflect the approximate location of the agent. Each ray trace hits a particular wall segment and uses the wall index as a feature.

3.4.2 Structural Properties

The experimental factors include structural properties, such as network architecture and training procedures. The network architectures and training procedures are responsible for a large part of the performance [135]. Moreover, training for optimal solutions is associated with efficient representation [122], which could lead to sparse activation patterns, such as entorhinal-like representation [35].

The explored recurrent neural network properties are: the number of hidden recurrent neurons, recurrent cell type, and recurrent initialization. The number of hidden neurons influences the maximum capacity of neural networks, and the increase in network complexity allows the model to find more complex solutions to the problem [136, 137]. Furthermore, exploring the performance of different recurrent cells (LSTM, GRU, and RNN) helps visualize the functional variations between recurrent architectures. Additionally, recurrent initialization strategies have an essential role in finding the optimal solution in neural networks [138]. Therefore including alternative initialization strategies explores additional differences in training performance.

Another critical aspect of network optimization is the selection of the learning rate and training optimizer. The learning rate was determined to be the most critical hyper-parameter to tune [135]. Additionally, an incorrectly tuned training optimizer will prevent the network from learning optimally [138]. Both training procedures are essential for optimizing recurrent neural network performance.

3.4.3 Regularization Techniques

Regularization was essential for causing emergent entorhinal-like representation, according to Banino et al. [1] and Cueva and Wei [2]. The experimental factors include usual regularization techniques, such as Dropout and weight decay. These approaches are implemented to test both the training performance and the ability to create emergent representation. Additionally, some new regularization techniques are introduced to be able to address additional hypotheses.

Several Dropout techniques are tested to reduce the co-adaptation between neurons

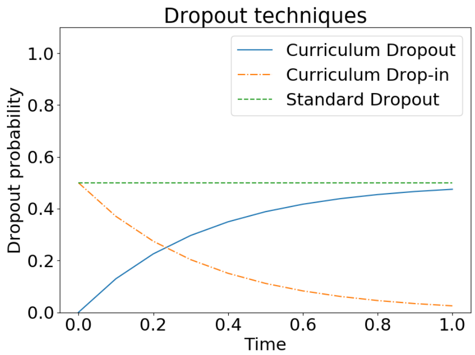


Figure 3.7: **Dropout progression**
All three Dropout techniques have a different Dropout probability progression over time. No neurons are dropped out with a probability of 0.0, and all neurons are silenced with a probability of 1. Standard Dropout has a fixed Dropout probability rate of 0.5. Curriculum Dropout exponentially increases the Dropout probability until it reaches the 0.5 threshold. Conversely, the Curriculum Drop-in algorithm exponentially decreases the Dropout probability over time until no neurons are silenced anymore.

[114]. Standard Dropout stochastically silences a fixed proportion of the neurons within a network layer. Curriculum Dropout is becoming a popular alternative approach, which progressively increases the Dropout probability until it reaches the standard Dropout probability, thus increasingly silencing more neurons [135]. A spin-off idea called Curriculum Drop-in is introduced in this thesis. This approach starts with the standard Dropout probability, and reduces the Dropout probability over time, thus silencing fewer neurons over time. For more information on the Dropout probability progression, see Figure 3.7.

Weight decay is an alternative regularization approach, and is applied in two circumstances: linear weight decay and recurrent weight decay. Linear weight decay penalizes excessively large neural weights, as used by Banino et al.'s [1] bottleneck layer $W_i^{bottleneck}$.

$$L(w) = \frac{1}{T}\sum_{t=0}^{T}(y_t^{pred} - y_t^{target})^2 + \lambda \cdot \frac{1}{N_{weights}} \sum_{i=0}^{N_{weights}} (W_i^{bottleneck})^2$$

Additionally, recurrent weight decay penalizes the recurrent weights. Recurrent weight decay is part of the Metabolic constraint in Cueva and Wei [2], and penalizes recurrent input weights W^{in}, recurrent output weights W^{out}, and recurrent state activation $u(t)$. Also, a fixed weighted relationship between the Metabolic constraint elements is introduced using c_{out}, c_{in}, and $c_{recurrent}$.

$$L(w) = \frac{1}{T}\sum_{t=0}^{T}(y_t^{pred} - y_t^{target})^2 + \lambda \cdot (\frac{c_{in}}{NN_{in}} \sum_{i,j=0}^{N,N_{in}} (W_{ij}^{in})^2 +$$

$$\frac{c_{out}}{NN_{out}} \sum_{i,j=0}^{N,N_{out}} (W_{ij}^{out})^2 + \frac{c_{recurrent}}{NT} \sum_{i,t=0}^{N,T} (u_i(t))^2$$

Regularization techniques could force neurons to specialize and minimize collaboration entirely. Winner-Takes-All (WTA) [139] can be used to investigate competition among neurons. This method silences all neurons, except for the highest activated neuron. Thus, WTA forces a small subset of neurons to specialize in their activation pattern [37].

$$a(t) = \max_{i}(u_i(t))$$

Alternative Winner-Takes-All strategies exist, such as the soft-competition approach using the softmax function. The softmax approach transforms the neural activity $u(t)$ using the fraction of the neural activation over the exponential sum of all neural activity. This soft-competition approach can highlight the collaboration between neurons but also silence the activation of redundant neurons.

$$a(t) = \frac{e^{u(t)}}{\sum_{j=0}^{N} e^{u_j(t)}}$$

Additionally, this study introduces a new regularization technique called Neural Fatigue, which is based on the signaling timeout from real neurons. The signaling timeout is the delay between action potentials, in which neurons cannot physically signal [140]. This phenomenon can be translated into artificial neurons by silencing the most active recurrent neuron in the next time-step. This approach should encourage precise timing and activation for each neuron to signal information efficiently [141].

3.5 Summary

The data simulator generates attractor-based agent trajectories to gather velocity, directional, and positional information for the training process. A base model is used to replicate earlier studies, enables investigation of various underlying factors, and measure the effect of entorhinal-like representation on training performance in recurrent neural networks.

Traditional models visualize entorhinal-like representation in recurrent networks using spatial and non-spatial (velocity and direction-based) activity plots. This study introduces new visualization techniques using a maximum activity plot, average performance plot, trajectory performance plot, and spatial influence plot. The analysis diversity is useful for developing a comprehensive understanding of the underlying activation pattern of neurons.

Additionally, input features, structural properties, and regularization techniques are introduced to investigate the factors underlying entorhinal-like representation.

The upcoming chapter outlines the results from the three main research questions testing the representation robustness, underlying factors, and effects on training performance linked to the emergence of entorhinal-like representation in recurrent neural networks.

Chapter 4

Results

The research questions translate into five experiments: (1) replicating earlier work, (2) experimenting with input features, (3) varying structural properties, (4) evaluation of regularization techniques, (5) and analysis of entorhinal-like representation training performance in recurrent neural networks.

4.1 Replication Studies

Earlier work is replicated using models from Banino et al. [1] and Cueva and Wei [2] to investigate emerging grid-like representation in recurrent neural networks. The experiments are designed to answer the question: “Is it possible to replicate emergent entorhinal-like representations using non-essential simplifications of earlier work models?”. A couple of non-essential factors were omitted from the replicated models since they are either proprietary or not related to the emergence of entorhinal-like representation.

The functional differences between the earlier work models are highlighted in Figure 2.4 and Figure 2.5. The main differences between the models are the recurrent neural cell and regularization techniques. Banino et al.'s model used an LSTM gated recurrent network and observed grid-like representation in a bottleneck layer regularized by Dropout. However, Cueva and Wei's model uses a Continuous-Time Recurrent Neural Network (CTRNN [142]) and concludes that the Metabolic constraint on the recurrent weights and recurrent state was responsible for emergent grid-like representation in recurrent cells.

4.1.1 Banino et al.

The executed experiments aim to reproduce the results from Banino et al. through a simplified model to observe if grid-like representation would emerge. The simplified experiments have to leave out the proprietary vision module by Banino et al., but this should not impair the results of the experiment since Cueva and Wei's model showed emerging representation without vision. All other experimental details and configurations from the original study were used to train the recurrent neural network. For more information on the experimental setup for the model, see Figure 4.1.

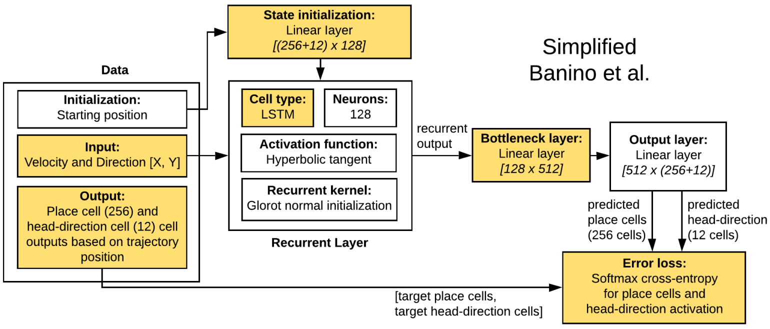
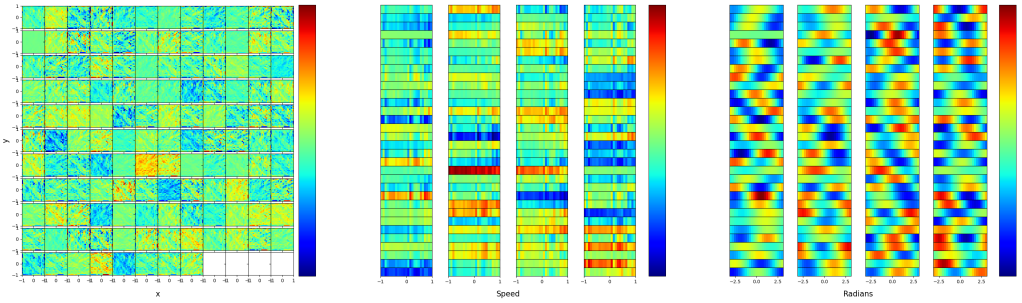

State initialization:
Linear layer
[(256+12) x 128]
Simplified
Banino et al.
Data
Initialization:
Starting position
Input:
Velocity and Direction [X, Y]
Output:
Place cell (256) and head-direction cell (12) cell outputs based on trajectory position
Cell type:
LSTM
Neurons:
128
Activation function:
Hyperbolic tangent
Recurrent kernel:
Glorot normal initialization
Recurrent Layer
recurrent output
Bottleneck layer:
Linear layer
[128 x 512]
Output layer:
Linear layer
[512 x (256+12)]
predicted place cells (256 cells)
predicted head-direction (12 cells)
Error loss:
Softmax cross-entropy for place cells and head-direction activation
[target place cells, target head-direction cells]

Figure 4.1: **Banino et al. simplified model**
The simplified model trains using (1) the split direction input approach, (2) the network predicts the activity of place- and head-direction cells associated with the trajectory, (3) LSTM uses 128 neurons, (4) bottleneck layer of 512 neurons, and (5) the model predicts 256 place cell and 12 head-direction cell activations.

Banino et al.'s model predicts artificial place- and head-direction cell activations related to the spatial position of the agent. The model is initialized with the initial activations of the place- and head-direction cells and predicts the place- and head-direction cells at the output layer. The softmax cross-entropy loss is used to train on the distribution difference between the predicted and target activity.

The replication results for Banino et al.'s model are visualized using activity plots in Figure 4.2, and performance plots in Figure 4.3.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. All neurons have stochastic spatial activation patterns.

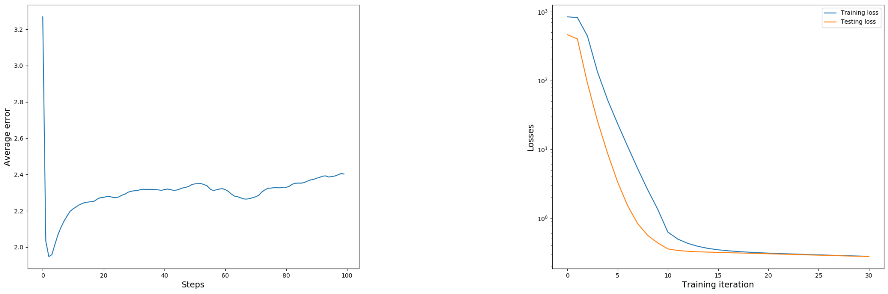
(b) **Velocity activity plot**
Each plot cell visualizes the average velocity-dependent neural activity. The representation for each neuron consists of noisy uniform activations.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Most neurons have directional activity Gaussian peaks, which shows angular velocity specialization.

Figure 4.2: **Representation from simplified Banino et al. model**
Replicating Banino et al's model through non-essential simplifications results in head-direction-like representation in most neurons.

The spatial activity plot in Figure 4.2a shows stochastic spatial activation in all neurons. None of the neurons produce geometric shapes in the spatial activity plots. The velocity-dependent activation plot (Figure 4.2b) also shows uniform and noisy neural activations. Similarly to the spatial activity plots, there is no resemblance of any specialization according to the velocity input. However, the directional plot in Figure 4.2c shows angular velocity specialization for almost all neurons. The neurons show preferred directional activity (neuron $[1,0]$ and $[0,4]$) resembling Gaussian-like activation patterns. These patterns are similar to head-direction cells, which have a preferred directional activation range. These results lead us to conclude that head-direction activated cells have emerged, but unfortunately, this does not align with the experimental aim since the intention was to reproduce Banino et al. 's emergent grid-like representations.

The average performance error in Figure 4.3a highlights an unusual spike in the first few trajectory steps. An initialization problem might be the culprit, potentially caused by improper initialization. The loss plot in Figure 4.3b shows that the performance has converged, but very peculiar is the gap between the training and testing error. The test error is usually higher compared to the training error.



(a) **Average performance plot**
The time-dependent loss is calculated using the average loss for each trajectory time-step. This plot shows initialization problems, the first few steps have excessively large error compared to the remainder of the trajectory.

(b) **Loss plot**
The loss plot visualizes the trained loss differences between the training (blue) and testing (orange) error. The loss continues to show training improvements until the performance converges around epoch 15.

Figure 4.3: **Performance results for the simplified model of Banino et al.**
Replicating Banino et al's model through non-essential simplifications results in initialization problems, leading to fast convergence to a local minima.

The results from Banino et al.'s simplified model do not replicate the original findings. Neurons show stochastic spatial activity and directional dependence similar to head-direction cells. The lack of grid-like representations is possibly due to the omission of the first-person vision module. These visual features could introduce useful non-linear features related to the spatial position, potentially necessary for entorhinal-like representation.

4.1.2 Cueva and Wei

The Metabolic regularization constraint was responsible for the emergence of irregular-, grid-, border-, and band-like representation in Cueva and Wei [2] recurrent neural network. The Metabolic constraint uses weight decay on recurrent input weights, recurrent output weights, and the recurrent state. The simplified model uses Cueva and Wei's recommended starting weighted relation to train the model. For more information on the simplified model see Figure 4.4.

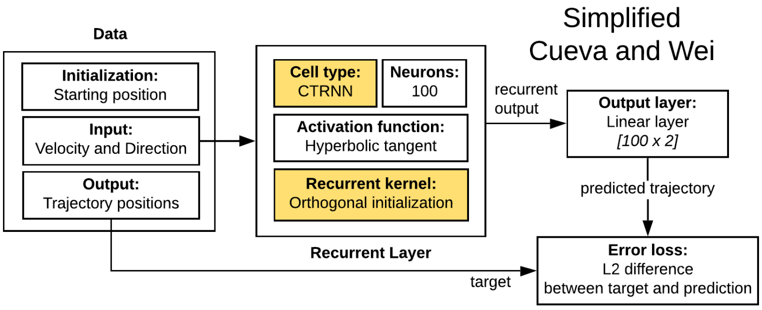
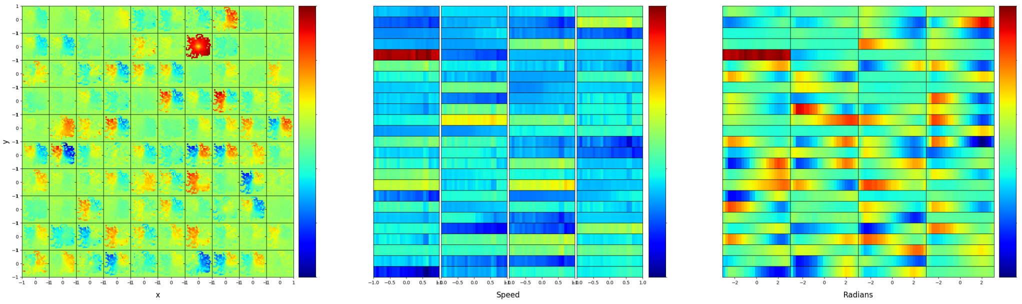


Figure 4.4: **Cueva and Wei simplified model**
Cueva and Wei's model contains four essential aspects, the model uses (1) orthogonal recurrent weight initialization, (2) 100 recurrent neurons, (3) a Continuous-Time Recurrent Neural Network (CTRNN), and (4) the recurrent layer is connected to the output layer.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Neurons are partially spatially active, but show no geometric activation patterns.

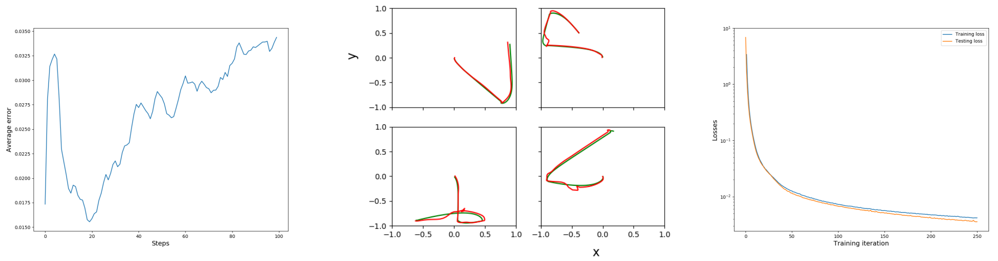
(b) **Velocity activity plot**
Each plot cell visualizes the average velocity-dependent neural activity. Cells show uniform activation patterns when averaging activity over velocity.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Prevalent linear dependence on direction explains spatial and velocity activity irregularities.

Figure 4.5: **Simplified Cueva and Wei model representations**
Replicating Cueva and Wei's model through non-essential simplifications results in partial linear dependence on head-direction.

Cueva and Wei's model is trained slightly differently compared to Banino et al.'s model. Each trajectory starts at the center of the environment $(0,0)$; thus, the model does not require state initialization. Furthermore, the network trains to predict the x- and y-coordinate of the agent during the trajectory.

The results are visualized in spatial and non-spatial activity plots in Figure 4.5, and performance plots in Figure 4.6. The spatial activity results in Figure 4.5a from the simplified Cueva and Wei [2] model is less noisy compared to the simplified approach from Banino et al. [1]. The velocity-dependent plot in Figure 4.5b shows a similar uniform activity pattern among recurrent neurons compared to the simplified Banino et al. model. Additionally, a dependence on the direction is noted in Figure 4.5c, see neuron $[3,1]$ or $[1,9]$. This directional-dependence would lead us to conclude that a significant part of the network is dependent on head-direction. The activation pattern is linear, and not Gaussian compared to the simplified Banino et al. model. Thus it is not possible to conclude that head-direction-like representation emerged in this model.



(a) **Average performance plot**
The time-dependent loss is calculated using the average loss for each trajectory time-step. The average performance plot shows initialization problems in the first few trajectory time-steps.

(b) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. The plot highlights that the model performs close to optimal.

(c) **Loss plot**
The loss plot visualizes the trained loss differences between the training (blue) and testing (orange) error. Both the training and testing loss show desirable generalization and speedy convergence.

Figure 4.6: **Simplified Cueva and Wei model performance results**
Replicating Cueva and Wei's model through non-essential simplifications results in good network generalization performance.

The average performance plot in Figure 4.6a shows initialization problems in the first few steps, after which the average performance increases almost linearly. The network potentially suffers from improper initialization, even though the network achieves impressive performance loss. This performance is best visualized in the trajectory performance plot in Figure 4.6b. This figure shows that the trajectory overlap is close to optimal. Additionally, the loss plot (Figure 4.6c) shows a good balance between training and testing performance.

Neither experimental results have shown resemblance to the emerging grid-like representation from Banino et al. [1] and Cueva and Wei [2]. Only head-direction cells could be distinguished due to the directional specialization (Gaussian activity peaks) in various recurrent cells in the simplified Banino et al. model.

4.2 Hybrid Baseline Model

Exploring the potential factors underlying emergent entorhinal-like representation requires a structured approach. The experiments test three main factors: (1) input features, (2) structural properties, and (3) regularization techniques. These experiments attempt to answer the following research question: "What factors can influence the emergence of entorhinal-like representation in recurrent neural networks?". However, to reflect on the representation requires first introducing a hybrid baseline model to relate the performance and representation of these three factors.

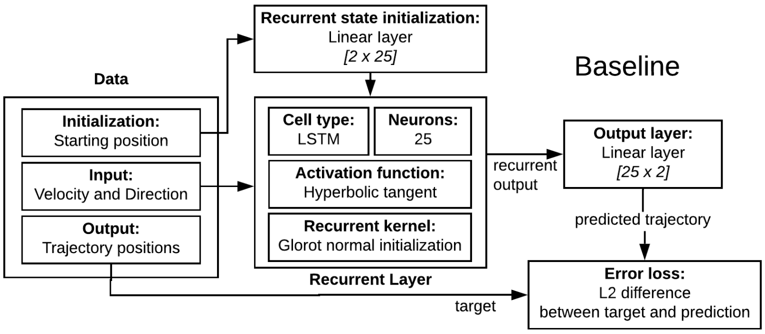
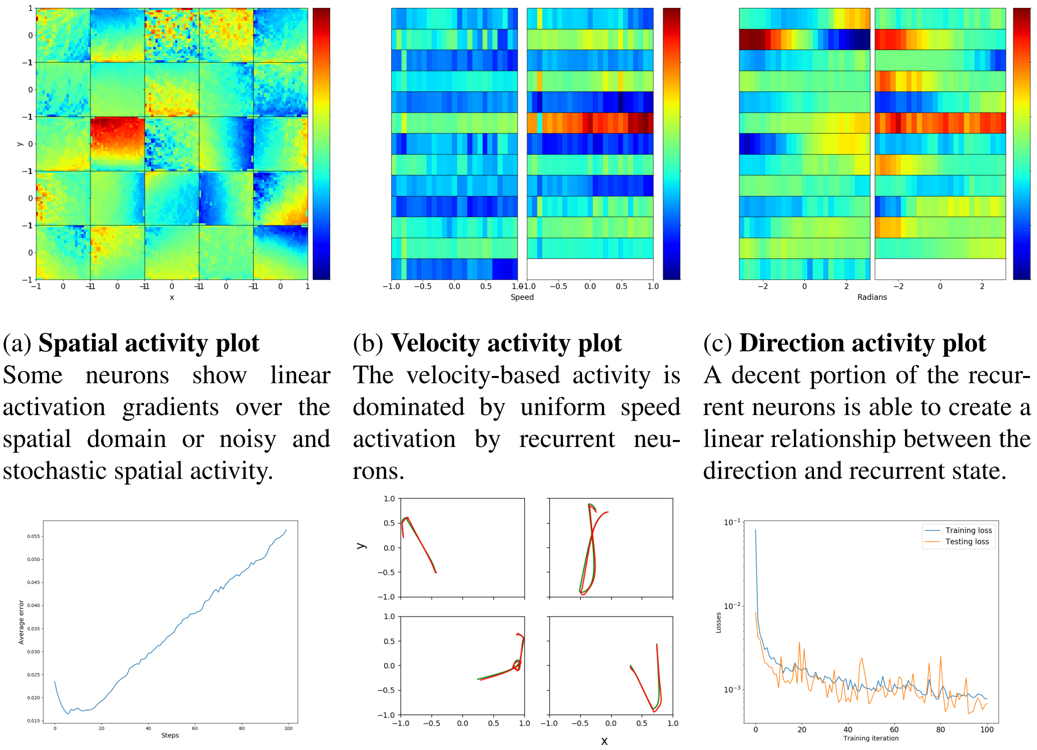


Figure 4.7: **Baseline model**
The standard experimental setup consists of 25 LSTM neurons and a linear output layer.

The hybrid baseline model is an amalgamation of the Banino et al. [1] and Cueva and Wei's model [2]. The hybrid baseline uses an LSTM gated recurrent cell, inspired by the work from Banino et al. [1], but uses fewer neurons compared to their original model. The model trains to integrate the position from the velocity and direction, similar to Cueva and Wei's approach [2]. Thus, training on the difference between the predicted and target trajectories to optimize the neural network weights. For more details on the hybrid baseline model, see Figure 4.7.

The plots in Figure 4.8 visualize the converged performance and representation for the trained model. The baseline configuration results in Figure 4.8a shows stochastic activations (neuron $[0,2]$), but mostly linear spatial activation gradients (neuron $[1,2]$ and $[4,3]$). The velocity-dependent plot in Figure 4.8b shows uniform but noisy activation when averaging and binning the recurrent activity based on speed alone.

Furthermore, the direction activity plot in Figure 4.8c shows partial dependence on direction. Mainly due to the linear directional-dependent activity gradients shown in a couple of neurons. For example, neurons $[0,0]$, $[0,1]$, and $[1,1]$ are prime examples of this type of linear activity gradient. The noisy and stochastic spatial activations could be explained by the reliance on other variables, such as direction or velocity. Thus, the network can specialize both in input features (mainly direction) and output dimensionality (spatial).



(a) **Spatial activity plot**
Some neurons show linear activation gradients over the spatial domain or noisy and stochastic spatial activity.

(b) **Velocity activity plot**
The velocity-based activity is dominated by uniform speed activation by recurrent neurons.

(c) **Direction activity plot**
A decent portion of the recurrent neurons is able to create a linear relationship between the direction and recurrent state.

(d) **Average performance plot**
The time-dependent loss is calculated using the average loss for each trajectory time-step. The initialization error is very small, and the error increases linearly over time.

(e) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. The trajectory shows a large overlap between the target and predicted trajectories.

(f) **Loss plot**
The loss plot visualizes the trained loss differences between training (blue) and testing (orange) error. The training and testing loss averages follow closely and converge to a suitable solution.

Figure 4.8: **Hybrid baseline representation and performance**
The hybrid baseline model results in spatial- and directional activity and provides good path-integration performance. Each plot cell visualizes the average (a) spatial, (b) velocity, (c) directional activity for an artificial neuron.

Looking at the performance plots in Figure 4.8d and Figure 4.8e shows that the network performs close to optimal. Since the average performance plot in Figure 4.8d shows no initialization problems in the first few steps, and the error increases linearly during the trajectory. The spatial trajectory performance in Figure 4.8e shows the network is predicted path is close to the target trajectory with sufficient accuracy. While emerging grid-like representation is lacking, the model seems to provide an excellent candidate for path-integration. Additionally, as seen in the loss plot (Figure 4.8f) the network performance converges quickly to balance performance between training and testing error.

4.3 Input Features

The input experiments aim to define the role between input features and generating biologically-plausible representation. The four input features experiments are: separating the input direction into x- and y- component (direction), adding input noise to the velocity and direction (noise), ray-tracing replacing the input data (ray-tracing), and the combination of both input data with ray-tracing (input with ray-tracing).

The results in Figure 4.9 show that the split direction performs better compared to the default baseline. However, the most stable and optimal solution combines the original input with ray-tracing, as seen in the loss comparison plot.

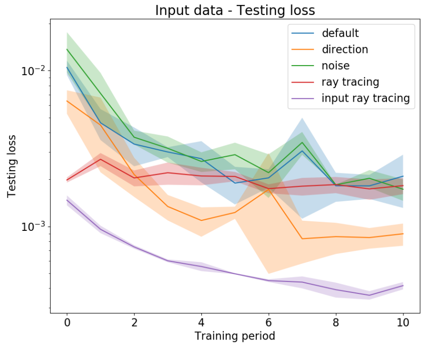


Figure 4.9: **Input feature approaches testing losses**
The loss plot visualizes the trained loss differences between input feature approaches. The direction, ray-tracing, and input ray-tracing approach can improve path-integration performance in recurrent neural networks.

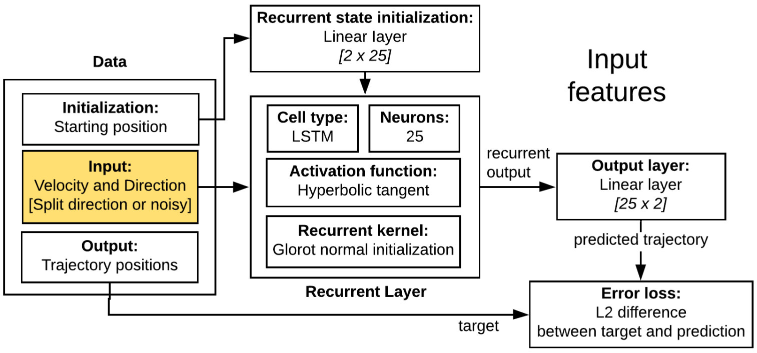


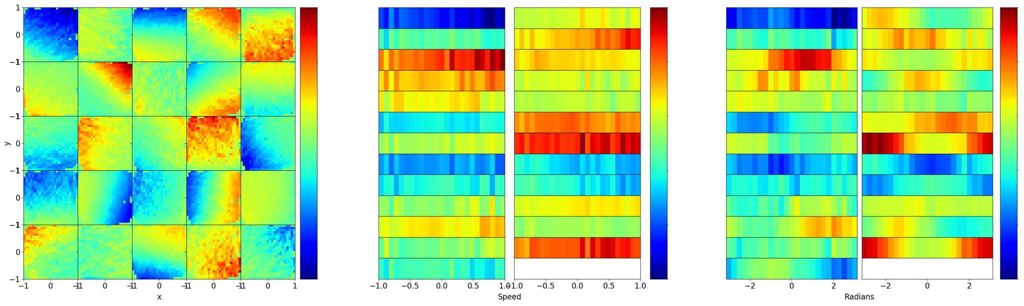
Figure 4.10: **Split direction and noisy input feature model**
The changes to the experiment include the use of split direction and noisy input. The approaches replace the input data and initialize with the starting position.

The first two experiments explore the split direction and noisy input features. The architectural differences are laid out in the architectural diagram of Figure 4.10.

4.3.1 Split Direction

This experiment tests if the split direction approach proposed by Banino et al. is associated with better performance and if it might be the key to entorhinal-like representation.

The representation from the split direction technique in Figure 4.11a highlights predominantly linear spatial activity gradients, for example neurons $[1,1]$ and $[3,1]$. The velocity-dependent activation in Figure 4.11b shows similar uniform and noisy activation compared to the hybrid baseline configuration. However, the cells in the directional activity plot (Figure 4.11c) show Gaussian-like activity patterns over the angular range (for example neurons $[0,2]$ and $[1,3]$), similar to head-direction cells. The performance improvements are likely due to the predominantly linear spatial activity gradients and directional tuning.



(a) **Spatial activity plots**
Each plot cell visualizes the average spatial activity for an artificial neuron. Most cells clearly show linear spatial activity gradients, however some cells show more noisy activity gradients.

(b) **Velocity activity plots**
Each plot cell visualizes the average velocity-dependent neural activity. Mostly uniform velocity dependent activity, speed is likely not a dependent factors for the model.

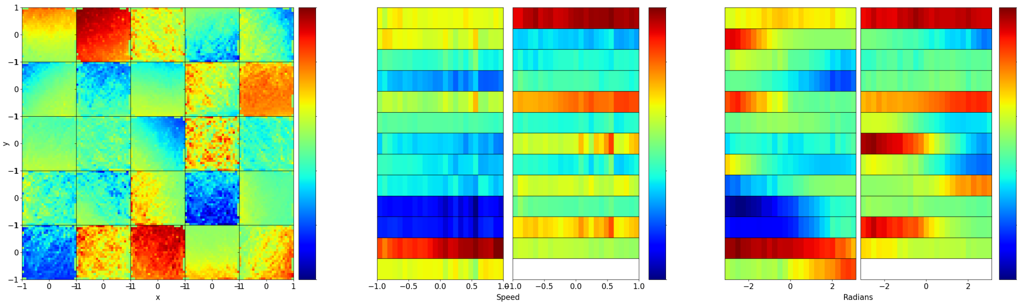
(c) **Direction activation plot**
Each plot cell visualizes the average direction-dependent neural activity. Some cells show Gaussian activity-like patterns which is similar to the results from Banino et al.'s simplified model.

Figure 4.11: **Split direction input representation and performance**
The split direction input method separates the original angular input into its x- and y-components. The spatial activity plot shows predominantly linear spatial activity gradients and directional dependence.

4.3.2 Noisy Input

Banino et al. motivate the use of Dropout through the hypothesis that it would add noise to the training model, due to the stochastic silencing of random neurons. Therefore adding a small amount of Gaussian noise to our velocity and directional input could replicate the same effect and test their hypothesis.

The resulting spatial representation (Figure 4.12a) shows similar stochastic spatial activity patterns compared to the hybrid baseline model. Some of the cells show linear spatial activity gradients, such as neurons $[0,0]$ and $[2,2]$. However, most neurons have spatially stochastic activity patterns, such as neurons $[1,4]$ and $[3,2]$, these cells could rely on other features. The velocity-dependent activation (Figure 4.12b) shows predominantly uniform but noisy activations. As seen before, speed might not be an informative feature for the simulated dataset. A sufficient number of recurrent cells have specialized in the directional input, as shown in Figure 4.12c. The recurrent cells show linear directional activity gradients, for example in neurons $[0,1]$ and $[1,6]$.



(a) **Spatial activity plots**
Each plot cell visualizes the average spatial activity for an artificial neuron. A moderate amount of neurons show linear spatial activity gradients, but most cell are stochastically activated.

(b) **Velocity activity plots**
Each plot cell visualizes the average velocity-dependent neural activity. Again speed is likely not an informative feature, only showing uniform and noisy velocity-based activations.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Most cells shows partial or globally consistent linear directional activity gradients.

Figure 4.12: **Noisy input representation and performance**
Noisy input introduces additional noise to the velocity and direction. The spatial activity plots show both linear spatial activity gradients and linear directional activity gradients.

4.3.3 Ray-tracing

Ray-tracing was designed to explore biologically-inspired input representations. Ray-tracing uses a sparse representation for the environment to encode the current location, through the wall color-coding systems (see subsection 3.4.1). The ray-tracing experiments used 50 ray-trace samples to measure the environment and create features reflecting the approximate location of the agent. The model uses ray-tracing both to initializes the recurrent cell and as input features, as seen in Figure 4.13.

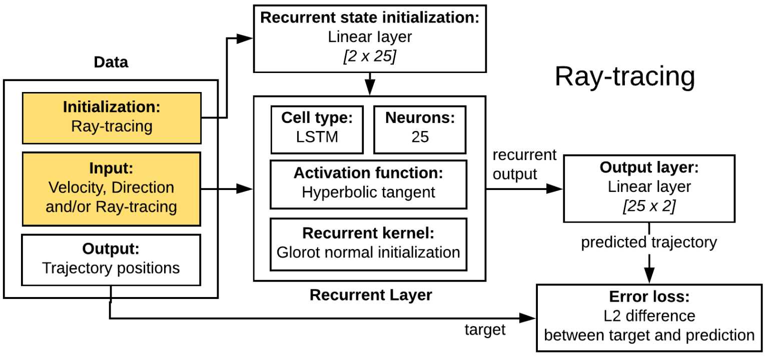


Figure 4.13: **Input ray-tracing model**
The changes to the experiment include the use of ray-tracing as initialization and input source. The first experiment replaces the input with ray-tracing, and the second experiment combines ray-tracing with the original input.

The ray-tracing input replacement experiment results visualized in Figure 4.15a show distinctly different spatial representations. The spatial representation shows signs of consistent curved and linear activity gradient patterns. For example, neurons $[0,3]$ and $[3,3]$ shows non-linear gradient activations. Due to the input replacement, it is not possible to plot velocity and directional dependence.

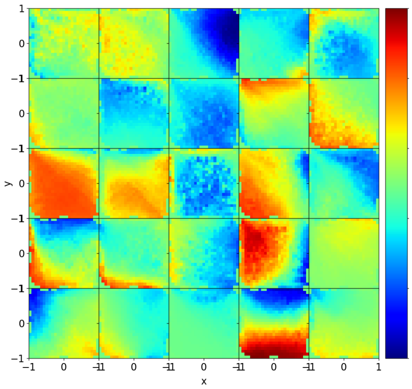
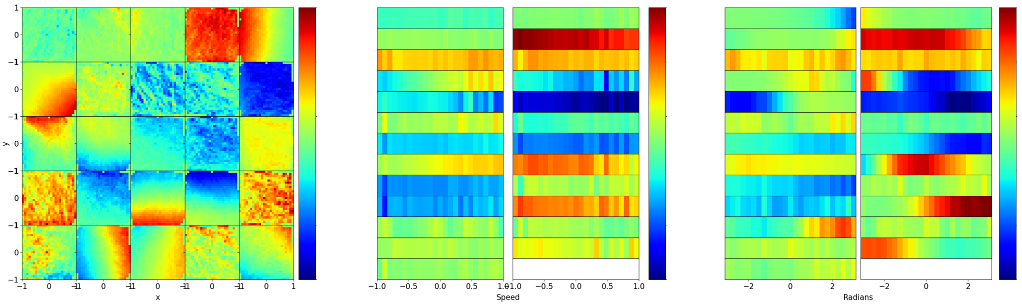


Figure 4.14: **Ray-tracing Spatial activity plot**
This ray-tracing experiment replaces the velocity and directional input by vision-like features from the environment. Each plot cell visualizes the average spatial activity for an artificial neuron. The first-person wall-color coding features create non-linear spatial activity gradients for a wide range of recurrent neurons. Although some cells stay linearly or stochastically activated.

Additionally, combining input with ray-tracing features yields linear spatial activity gradients (Figure 4.15a). Only some of those representations shows slanted activity gradients (neurons $[0,1]$ and $[0,2]$), but most curved activity gradients do not occur anymore. The spatial activity is a combination of noisy representation (see neuron $[0,3]$ and $[4,3]$) and linear activation gradients (see neuron $[4,0]$ and $[1,4]$). The velocity-dependent activity plot (Figure 4.15b) predominantly shows uniform and noisy activation patterns. The representation is very similar to all other experimental results.

The directional activity plot in Figure 4.15c shows specialized directional activity. A portion of the recurrent cells are stochastically active; however, a small fraction shows linear directional activity gradients, such as neuron $[0,10]$ and $[1,11]$. Even more special is the occurrence of Gaussian-like activation in neurons $[1,3]$ and $[1,7]$. While there is a possibility that head-direction-like representation emerged, it is not prevalent, and thus, it is not possible to directly jump to conclusions about the underlying activity behavior.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Neurons show linear spatial activity gradients or have noisy and stochastic activation patterns.

(b) **Velocity activity plots**
Each plot cell visualizes the average velocity-dependent neural activity. Still no sign of velocity-based specialization due to mostly uniform and noisy speed activated recurrent activity.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Some neurons are directionally activated and show linear activation gradients and even Gausian-like representation.

Figure 4.15: **Default and ray-tracing input spatial and non-spatial plots**
The input and ray-tracing approach adds the vision-like features to the velocity and directional input. The neurons show predominantly a combination of spatial or directional linear activity gradients, with some recurrent neurons showing head-direction-like activation patterns.

Varying the input features have not shown to be beneficial in the search towards grid-like representation. Only showing Gaussian-like directional activity for the split direction and input with ray-tracing approaches, similar to head-direction cells. However, ray-tracing can significantly boost the performance compared to the baseline model. The multi-modal input features results in partial non-linear representation and could potentially provide stepping stones towards entorhinal-like representation. The spatial and directional plots show both positional and direction specialization, thus the combination of both position and direction are critical features for performing path-integration.

The velocity-based activity plot has not shown significant relations between recurrent neural activity and velocity input. Thus future experiments will not refer to velocity activity plots since they do not add to the discussion of recurrent activity. Instead, a new approach is introduced using the spatial influence plot to investigate which spatially activated neurons have a significant influence when predicting the output.

While the input experiments have not directly manifested diverse representations similar to the entorhinal cortex, it might require to expand the search to other influential factors in recurrent neural networks. Therefore, the next section focuses on the structural properties of recurrent neural networks.

4.4 Structural Properties

Structural properties involve both network attributes and training procedures. Various papers point to network attribute and training procedures being highly influential for the performance of neural networks; this is especially the case for recurrent networks [135]. Also, the initialization strategy and optimizer choice have a significant influence on the network performance [138]. Therefore optimizing the structural properties of neural networks is an excellent addition to the input feature experiments.

The network attribute experiments use different recurrent cell types, adjust the number of hidden neurons, and change the recurrent weight initialization approach. Furthermore, the training procedures include alternative learning rates and various training optimizers. These model differences are highlighted in Figure 4.16.

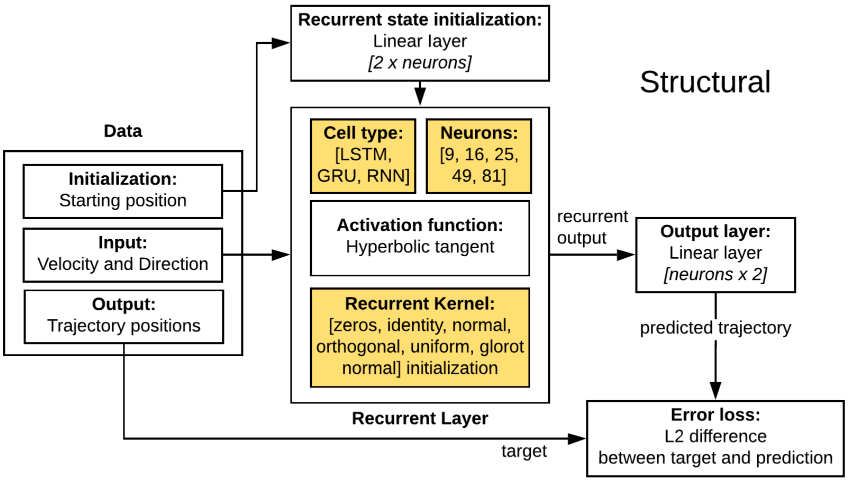
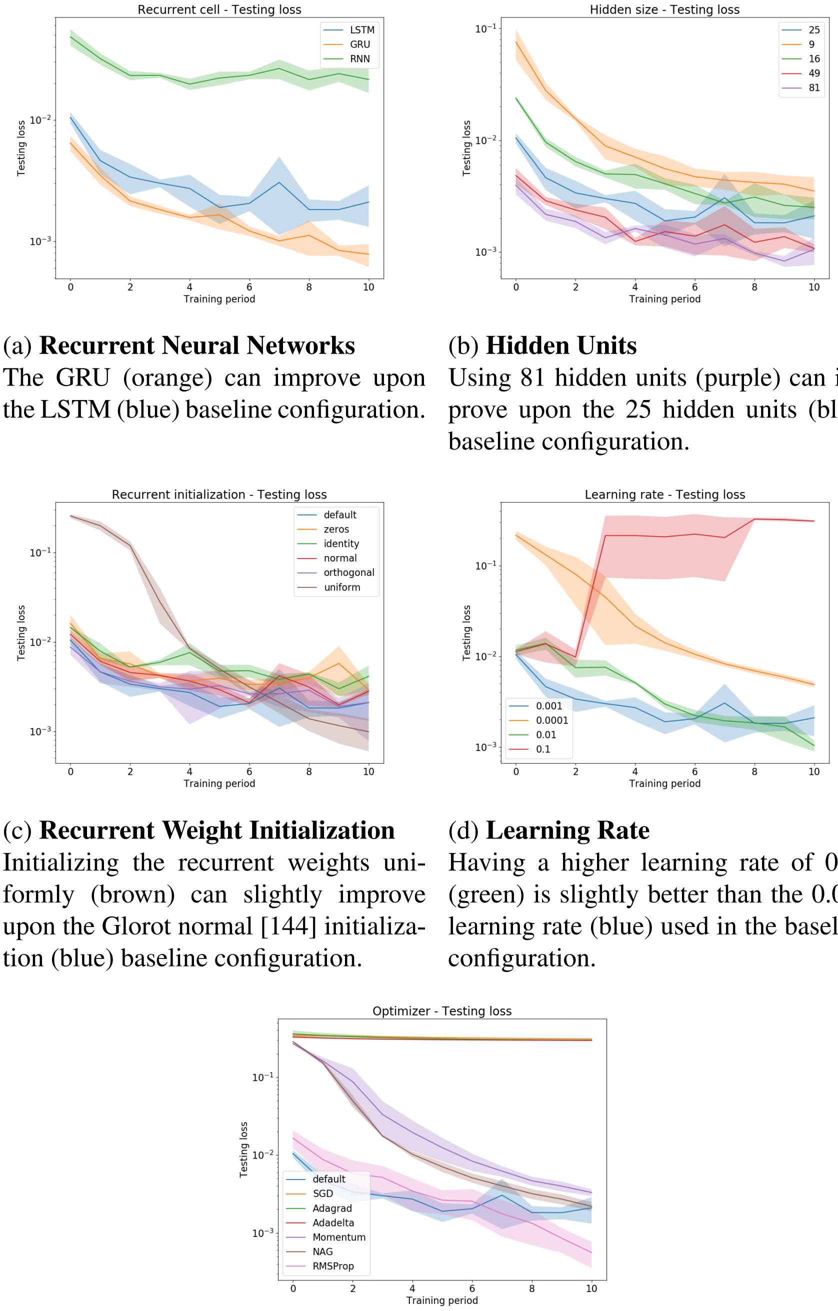


Figure 4.16: **Structural properties experimentation model**
Various aspects of the model are explored, such as network architecture and training procedures.

Several experiments show slightly improved performance compared to the hybrid baseline configuration (25 LSTM cells using Glorot normal [144] initialization), such as using a GRU cell (Figure 4.17a), increasing the number of hidden units to 81 (Figure 4.17b), and uniform recurrent weight initialization (Figure 4.17c). Additionally, the training performance can improve when using a higher learning rate of 0.01 as opposed to the default 0.001 (Figure 4.17d), and optimize the weights using the RMSProp optimizer (Figure 4.17e).



(a) **Recurrent Neural Networks**
The GRU (orange) can improve upon the LSTM (blue) baseline configuration.

(b) **Hidden Units**
Using 81 hidden units (purple) can improve upon the 25 hidden units (blue) baseline configuration.

(c) **Recurrent Weight Initialization**
Initializing the recurrent weights uniformly (brown) can slightly improve upon the Glorot normal [144] initialization (blue) baseline configuration.

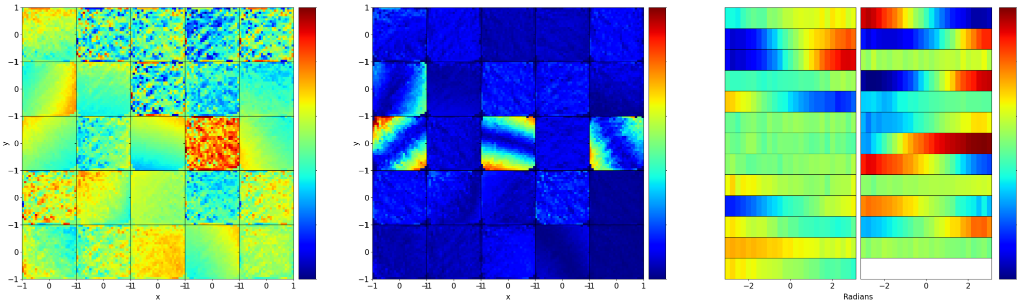
(d) **Learning Rate**
Having a higher learning rate of 0.01 (green) is slightly better than the 0.001 learning rate (blue) used in the baseline configuration.

(e) **Optimizers**
Optimizing the loss using RMSProp (pink) can achieve a lower loss compared to the Adam optimizer (blue).

Figure 4.17: **Network attribute and training procedure experiments performance**
The loss plots visualize the trained loss differences between learning rates. Several experiments have explored the performance differences for network architectures and training procedures.

4.4.1 Recurrent Cell

Varying the recurrent cell types (Figure 4.17a) resulted in poor performance from the RNN cell but showed improvements for the GRU cell compared to the LSTM cell. The spatial representation for the GRU cell in Figure 4.18a is more stochastically activated compared to the hybrid baseline model. The neuron spatial representation consists of stochastically activated cells (such as neuron $[1,0]$ and $[2,0]$) and shows only slight linear spatial activity gradients for some of the cells involved (see neuron $[0,1]$ and $[2,2]$).



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. The spatial activity plots show a combination of noisy- and linear spatial activity gradients.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. Only four cells contribute to the majority of the output representation.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. A large fraction of the recurrent neurons have a linear relation with directional input.

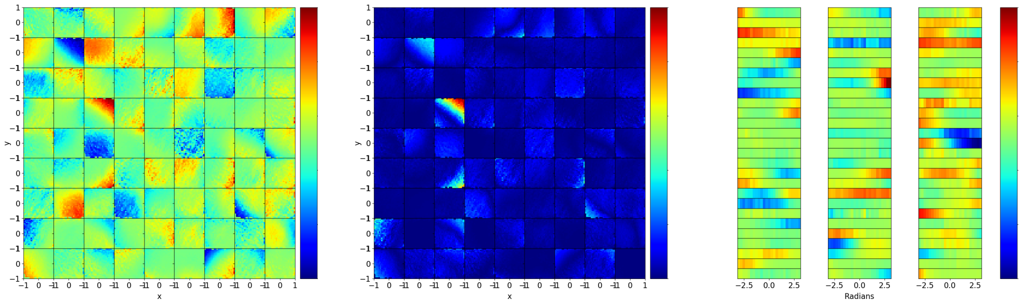
Figure 4.18: **GRU recurrent cell representation**
The GRU cell representation consists of both directionally tuned neurons and linear spatial activity gradients.

Despite the small portion of the cells showing linear activated gradients, according to the spatial influence plot (Figure 4.18b) these cells influence the output computation the most. Showing that the four directional bands can complement each other to perform path-integration. These activations are not similar to entorhinal-like representation but are reminiscent of the three band-like cells oriented at 0, 60, and 120 degrees presumably underlying grid-like representation [145]. Additionally, the direction activity plot in Figure 4.18c highlights linear directional activity gradients for a decent portion of the recurrent cells. For example neurons $[1,0]$ and $[0,2]$ are prime examples of linear directional activation. While most cells are directionally activated, none of the directionally activated neurons seem to influence the output. When looking closer at the spatial influence plot, only the spatially activated neurons show an influence on the output values. The directional specialization could be used for influencing the recurrent state; however, this is not quantified in these sets of experiments.

4.4.2 Hidden Units

The performance improvements for increasing the number of recurrent neurons decreases exponentially, as seen in Figure 4.17b. Despite the diminishing returns, using 81 neurons can still slightly improve upon all other hidden unit configurations.

Changing the number of hidden units is visualized in Figure 4.19a and shows that the ratio of linear and stochastically activated cells is similar to previous experiments. The emerging representation consists mostly of linear spatial activity gradients (see neuron $[1,1]$ and $[2,3]$). Additionally, increasing the number of hidden units does slightly improve the representational diversity, such as introducing non-linear spatial representation in neurons $[0,1]$ and $[1,6]$. Despite the non-linear spatial activity generation, the network does not seem to benefit from representational diversity. The influence plot Figure 4.19b only shows partial linear spatial activation. One neuron ($[2,3]$) is very influential, but it might also be the case that there is an imbalance in the influence metric and discounting other spatially consequential neurons. The directional plot shows significantly less directional specialization compared to previous experiments. Some cells show even linear- and Gaussian-like activation patterns ($[2,13]$ and $[2,16]$), but most neurons are not dependent on directional input.



(a) **Spatial activity plot** Each plot cell visualizes the average spatial activity for an artificial neuron. The spatial activity plot shows predominantly linear activity gradients and stochastic spatially active neurons.

(b) **Spatial influence plot** Each plot cell visualizes the average spatial influence of an artificial neuron. The influence separated into multiple neurons, being influential in the corners of the spatial environment.

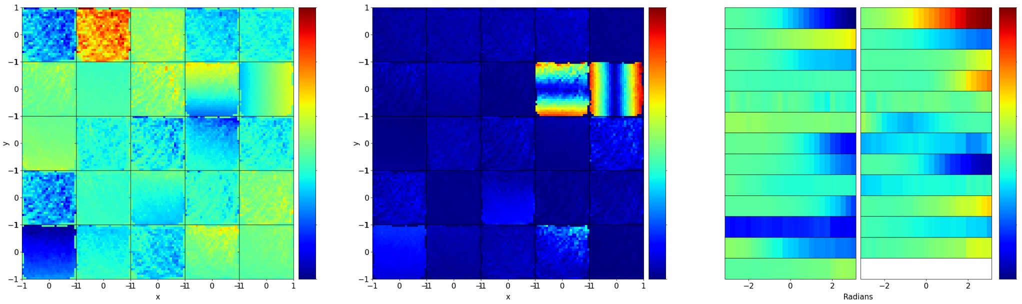
(c) **Direction activity plot** Each plot cell visualizes the average direction-dependent neural activity. Both linear and Gaussian-like directionally active neurons occur in the recurrent neural network.

Figure 4.19: **81 hidden units representation**
The representation for 81 neurons shows mainly linear spatial activity gradients. Increasing the number of neurons reduces the directional dependencies, increases the representational diversity, and distributes the spatial influence over more neurons.

4.4.3 Recurrent Initialization

The uniform initialization has a slight performance edge compared to default initialization, see Figure 4.17c. While all other initialization techniques also converge to a similar optimum, it is still interesting to investigate the uniform initialization technique and observe the underlying representation.

The representation in Figure 4.20a, shows widespread stochastic spatial activity (see neuron $[0,0]$ and $[0,1]$) and linear spatial activity (see neuron $[3,1]$ and $[4,1]$). The two linear spatially activated neurons show their influence on the output in Figure 4.20b. The x- and y-activity gradients are ideal for two-dimensional path integration. Potentially causing other neurons to have less spatial influence compared to those spatial basis neurons. The directional plot is quite uneventful, see Figure 4.20c. Some neurons $[0,0]$ and $[1,0]$ show directional linear gradients, but most neurons are uniformly active or have noisy activation behavior regarding the directional input features.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Neurons reflect stochastically activated spatial behavior, with two neurons showing linear activity gradients along the x- and y-axis.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. Only two cells are spatially relevant enough to cause a high influence for determining the output.

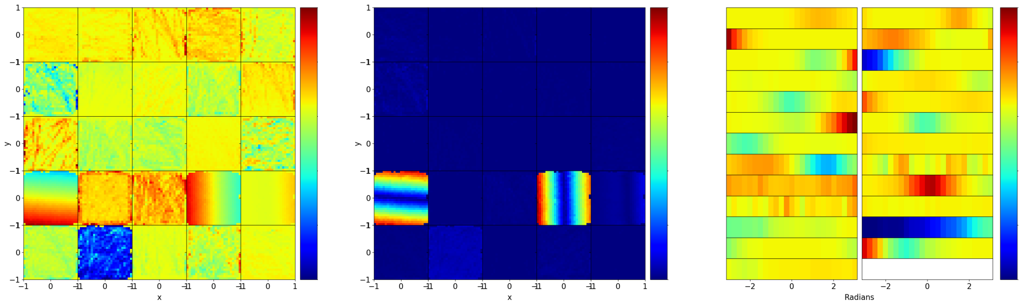
(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Most neurons are not directly directionally dependent, while some show linear activity gradients it does not dominate the majority of neuronal activations.

Figure 4.20: **Uniform recurrent initialization representation**
Using uniform recurrent initialization results in both noisy and linear spatial activity gradients. The improved performance is likely due to the simplified two-dimensional integration possible in two of the recurrent neurons.

4.4.4 Learning Rate

The representational differences, according to the change in the learning rate, are depicted in Figure 4.21. The differences are small between the baseline learning rate of 0.001 and a learning rate of 0.01, but we are still interested in the representational differences that could explain the performance gap.

The representation of a higher learning rate (0.01) shows a similar representation compared to the uniform recurrent initialization experiment (Figure 4.21a). Both the noisy spatial activity and linear spatial activity gradients are reoccurring representational phenomena. The linear spatial activity gradients (see neuron $[0,3]$ and $[3,3]$) are very similar to the previous results, creating a two-dimensional integration, with the y-axis activated neuron being slightly slanted. According to the spatial influence plot in Figure 4.21b, these two neurons $[0,3]$ and $[3,3]$ are very influential for predicting the path integrated position. The directional activity plot shows some Gaussian-like activation patterns $[0,0]$ and $[1,8]$, but most neurons are uniformly activated.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Most neurons express spatially stochastic activation patterns with two neurons acting as a two-dimensional integrator.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. Two neurons are spatially influential for predicting the output of the task, providing both the x- and y-dependent spatial activation necessary for path-integration.

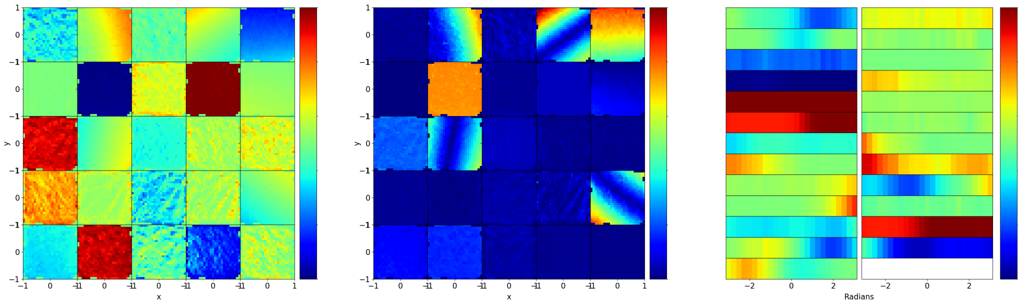
(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Neurons generally express uniform, noisy or Gaussian-like activation patterns.

Figure 4.21: **Learning rate 0.01 representation**
A learning rate of 0.01 facilitates both noisy and linear spatial activity gradients. The performance improvement is quite likely to be caused by combination of the spatially activated recurrent neurons, already providing the spatial basis functions for path-integration.

4.4.5 Optimizer

The loss optimizer experiments show comparable performance between the default baseline approach (Adam optimizer) and RMSProp. However, RMSProp shows slight performance benefits over the baseline approach.

The spatial representation for the RMSProp optimizer, in Figure 4.22a, shows common spatial and directional representations. Both showing stochastic (see neuron $[0,2]$ and $[2,1]$) and linear spatial activity gradients (see neuron $[1,2]$ and $[4,3]$) in the spatial dimensions. Looking closely at the spatial influence plot (Figure 4.22b), shows a combination of y-component based integration ($[4,0]$), multiple stripe-like cells ($[1,0]$, $[1,2]$, $[3,0]$ and $[4,3]$) and one bias cell ($[1,1]$). Additionally, the directional plot (Figure 4.22c) shows localized directional specialization in some cells but is not directly involved in predicting the output. More research is necessary for connecting the role between directional specialization and spatial activity; unfortunately, these basic plots are not able to uncover this direct relation.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Some recurrent neurons are spatially active, but most neurons show stochastically activated patterns.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. A small set of neurons show high spatial influence mainly due to their linear spatially activated neurons.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. The recurrent neurons are partially directionally activated, but most neurons are not directly related to directional activity.

Figure 4.22: **Optimizer experiments**
RMSProp representation consists of both noisy and linear spatial activity gradients. The slight improvements of the RMSProp optimizer is likely due to the spatial specialization due to linear spatial activity gradients of a small set of neurons.

All results regarding the structural properties of recurrent neural networks lack emerging entorhinal-like representation. This inability is not surprising since Banino et al. [1], and Cueva and Wei [2] clearly state their dependence on regularization. This agreement presumes that only regularization could provide the necessary drive within the network to shape and support entorhinal-like representation.

4.5 Regularization Techniques

The regularization technique experiments explore traditional and experimental regularization approaches and their relation to the emergence of entorhinal-like representation. As noted in earlier works, Dropout [1] and the Metabolic constraint [2] could be responsible for the emergence of entorhinal-like representation. It is also possible that other specialization or competitive strategies can produce biologically-plausible representation [6]. Experimental techniques explore various specialization techniques since the majority of neuroscientific studies point to competitive behavior or lateral inhibition required for entorhinal-like representation [58].

The following experiments use traditional (Dropout, recurrent regularization, and regularization loss) and experimental regularization (Winner-Takes-All, Neural Fatigue, and Specialization) approaches to investigate factors for emerging entorhinal-like representation. Figure 4.23 shows the model setup for the different regularization experiments.

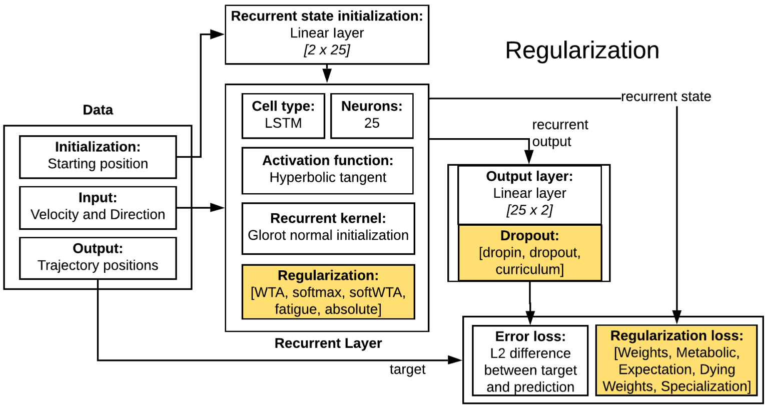
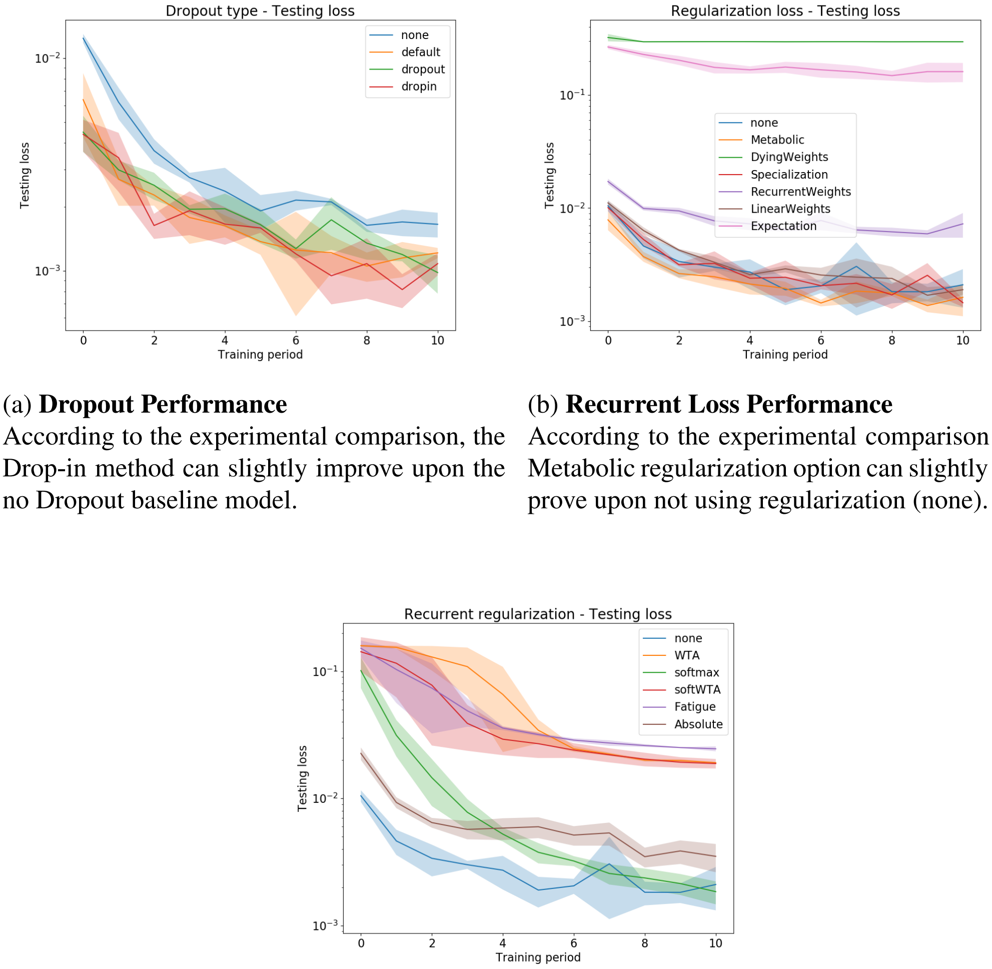

Regularization
Data
Initialization: Starting position
Input: Velocity and Direction
Output: Trajectory positions
Recurrent state initialization: Linear layer [2 x 25]
Cell type: LSTM
Neurons: 25
Activation function: Hyperbolic tangent
Recurrent kernel: Glorot normal initialization
Regularization: [WTA, softmax, softWTA, fatigue, absolute]
Recurrent Layer
recurrent state
recurrent output
Output layer: Linear layer [25 x 2]
Dropout: [dropin, dropout, curriculum]
target
Error loss: L2 difference between target and prediction
Regularization loss: [Weights, Metabolic, Expectation, Dying Weights, Specialization]

Figure 4.23: **Regularization model**
The three essential changes to the experimental model include the recurrent regularization methods, Dropout, and additional regularization losses. Recurrent Regularization applies limitations on the recurrent output. Dropout stochastically silences neurons in the output layer. Regularization loss penalizes the network weights and/or state of recurrent neurons.

The regularization experimental results are shown in Figure 4.24a for Dropout, Figure 4.24b for regularization loss, and Figure 4.24c for recurrent regularization techniques. Dropout experimental results in Figure 4.25 shows that the different dropout technique slightly exceeded performance beyond the baseline. The Metabolic constraint and linear weight regularization show comparable performance to the baseline model in Figure 4.24b. The representation of all three models is very similar; thus, it is possible to pick the Metabolic constraint and reflect on its representation. Figure 4.24c shows the loss for various recurrent regularization techniques. The softmax and absolute regularization approaches show similar performance compared to the baseline model. Both the Winner-Takes-All and Neural Fatigue approaches show suboptimal performance.



(a) **Dropout Performance**
According to the experimental comparison, the Drop-in method can slightly improve upon the no Dropout baseline model.

(b) **Recurrent Loss Performance**
According to the experimental comparison the Metabolic regularization option can slightly improve upon not using regularization (none).

(c) **Recurrent Regularization Performance**
According to the experimental comparison, the Winner-Takes-All (WTA) can perform worse compared to the hybrid baseline model. Other techniques perform similarly to the hybrid baseline model, such as absolute and softmax.

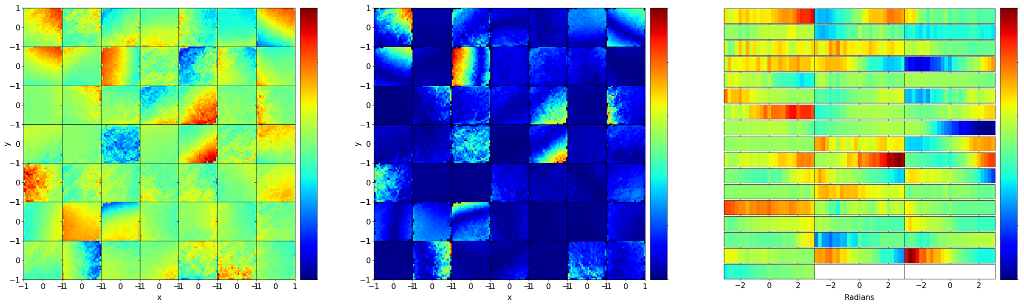
Figure 4.24: **Regularization experiments performance**
The loss plots visualizes the trained loss differences between regularization loss approaches. Dropout, regularization loss, and recurrent regularization experiments were executed to determine the influence of regularization techniques on performance.

4.5.1 Dropout

The Dropout regularization is applied to the linear layer weights and silencing about half of the connections from the recurrent state. The Dropout experiments use 49 recurrent neurons instead of 25 neurons to offset the number of dropped out neurons.

The standard Dropout (default), Curriculum Dropout (dropout), Curriculum Drop-in (dropin), and no Dropout (none) perform similar to each other. However, the Dropout approaches can improve slightly on not using Dropout at all, allowing us to investigate the representation of Curriculum Drop-in, an approach which incrementally reducing the Dropout probability.

Curriculum Drop-in representation is visualized in Figure 4.25. Both spatial activity (Figure 4.25a) and influence activity (Figure 4.25b) plot show noisy ($[0,4]$ and $[2,3]$) and linear ($[2,1]$ and $[6,0]$) spatial activity gradients. Similarly the directional activity plots in Figure 4.25c shows uniform ($[1,1]$ and $[2,1]$) and linear activity ($[1,0]$ and $[2,8]$) gradients. The performance increase for Drop-in, and likely all other Dropout approaches, is due to the increased number of neurons since the representational diversity does not increase compared to the hybrid baseline model without Dropout.



(a) **Spatial activity plot** Each plot cell visualizes the average spatial activity for an artificial neuron. Neurons are typically noisy or linearly activated over the spatial domain.

(b) **Spatial influence plot** Each plot cell visualizes the average spatial influence of an artificial neuron. The influence measurement highlights several linearly activated neurons and their contribution towards the output.

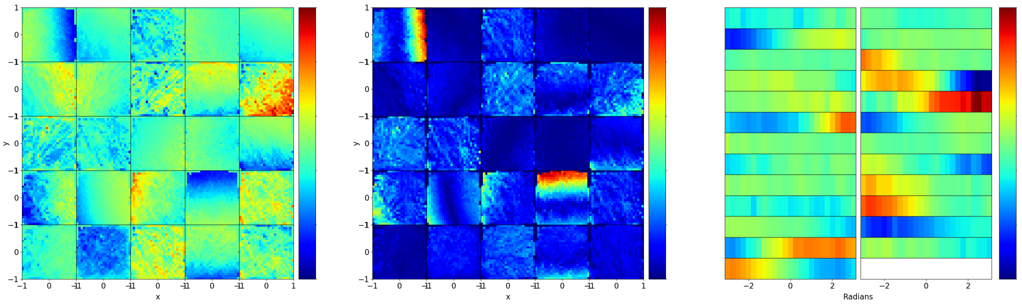
(c) **Direction activity plot** Each plot cell visualizes the average direction-dependent neural activity. Neurons are uniformly activated or show slight linear activity gradients.

Figure 4.25: **Dropin regularization experiments**
The spatial representation of Curriculum Dropin consists of typical activation behavior of recurrent neurons. Showing both directional and spatially activated recurrent neurons.

4.5.2 Regularization Loss

The regularization loss is added as an additional learning signal to promote auxiliary training objectives. The performance differences between regularization loss approaches are visualized in Figure 4.26. While multiple regularization approaches are performing similarly, it is possible to investigate the underlying representation of the Metabolic constraint to see if the Metabolic constraint itself develops any biologically-inspired representations.

The Metabolic constraint uses weight decay on the recurrent input weights, recurrent output weights, and recurrent state. The Metabolic constraint results highlight both noisy neural activation patterns and linear activity gradients (Figure 4.26a). The developed representation is similar to structural properties experiments. The spatial representation develops similar linear spatial activity gradient ($[1,3]$ and $[3,4]$), and stochastic spatial activations. The influence plot (Figure 4.26b) mainly highlights the contribution of two neurons ($[0,0]$ and $[3,3]$) providing x- and y-components for spatial integration. The directional plot in Figure 4.26c is also very similar to previous results, providing only partial linear activity gradients ($[0,5]$ and $[1,4]$) but nothing out of the ordinary. The results point to the possibility that the untrained weighted relation between the Metabolic constraint components is not competent enough to generate emerging entorhinal-like representation.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Neurons have both linear and stochastic spatial activity.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. Two neurons are most influential for predicting the output, providing both the x- and y-component for path integration.

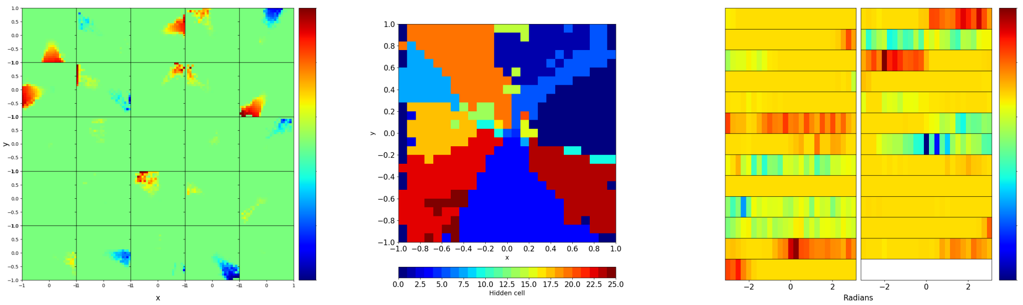
(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. A small amount of the neurons shows directional specialization through linear directional activity gradients.

Figure 4.26: **Regularization loss experiments**
Metabolic regularization achieves similar performance to having no regularization at all. According to the representational plots, the application of the unweighted Metabolic constraint did not lead to biologically-inspired representation.

4.5.3 Recurrent Regularization

Recurrent regularization adjusts the output of the recurrent neurons to force specialization between neurons, and tests if the competition is a suitable alternative to the conclusions from Banino et al. [1] and Cueva and Wei [2].

The best performing regularization approaches, such as softmax and Absolute, have similar performance and representation compared to previous experiments. However, Winner-Takes-All and Neural Fatigue showed distinct activity patterns. Winner-Takes-All results in spatial specialization, as seen in the spatial, influence, and directional plot (Figure 4.27). The neurons are mainly activated based on the spatial position of the agent. The maximum activity plot Figure 4.27b signifies the strong spatial specialization through Winner-Takes-All. Unfortunately, the performance is significantly worse compared to other regularization options, thus making it not relevant for future experimentation.

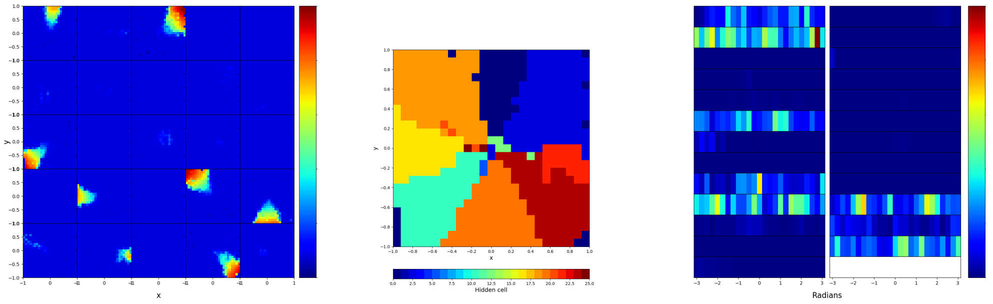


(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Neurons activity is mostly active near border boundaries and spreads towards the center of the environment.

(b) **Maximum spatial plot**
Visualizes the most active neuron per spatial bin. Some neurons are able to specialize in linearly shaped activity cones. Which combined share the entire spatial activation pattern.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Most neurons are insensitive to directional input, because they are uniformly- or stochastically activated regarding direction.

Figure 4.27: **Recurrent regularization experiments**
The spatial representation of Winner-Takes-All consists of local specialization among neurons.



(a) **Spatial activity for Neural Fatigue regularization**
Each plot cell visualizes the average spatial activity for an artificial neuron. The neurns show very local specialization activation patterns.

(b) **Maximum spatial plot**
Visualizes the most active neuron per spatial bin. The maximum spatial plot for the Neural Fatigue regularization technique shows shared specialization across the environment.

(c) **Direction activity plot**
Each plot cell visualizes the average direction-dependent neural activity. Most neurons are never activated, and some are only stochastically activated based on directional input.

Figure 4.28: **Neural Fatigue experiments representation and performance**
The competitive nature of Neural Fatigues shares the spatial activations between neurons, but this results in suboptimal performance.

The Neural Fatigue regularization technique approach (see Figure 4.28) shows similar activation patterns compared to Winner-Takes-All. Neural Fatigue silences the most active neuron in the next time-step, similar to the neurological phenomenon of the signaling timeout in neurons.

Additionally, the specialization posed by Neural Fatigue shows a similar dependence on spatial activity, as shown in the maximum spatial plot in Figure 4.28b. Less neurons are able to be spatially active (Figure 4.28a) compared to the Winner-Takes-All experiment, however this results in more defined place-like ($[0,2]$ and $[3,3]$) and border-like representation ($[4,3]$) in neurons. The directional activity plot Figure 4.28c shows that the cells are not directionally activated at all, only showing stochastically activated neurons.

4.6 Entorhinal Performance

In addition to studying the emergence of entorhinal-like representations themselves, this study also investigates the relation between entorhinal-like representation and performance. Thus addressing the research question: “Does entorhinal-like representation in recurrent neural networks improve path-integration performance?”.

Previous studies lack any conclusive evidence on the effect of entorhinal-like representation. Also, none of the experiments in this thesis show conclusive evidence towards spatial entorhinal-like representation. Additionally, we still do not know if the development of entorhinal-like representation is beneficial for recurrent neural networks. Therefore this section finally attempts to relate entorhinal-like representation with training performance.

The experiment requires to artificially generate entorhinal-like representation to observe the effect of entorhinal-like representation since natural emergence has not been observed in any of the previous experiments. Two approaches are used: replacing input data and replacing the recurrent cell with artificially generated entorhinal-like representation based on trajectory data from the agent. For more information on the model layout, see Figure 4.29.

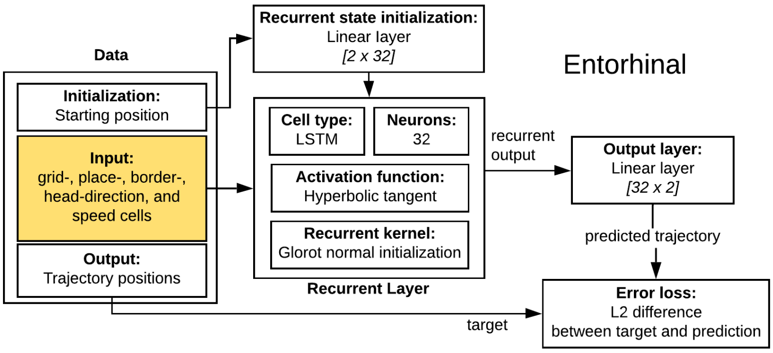
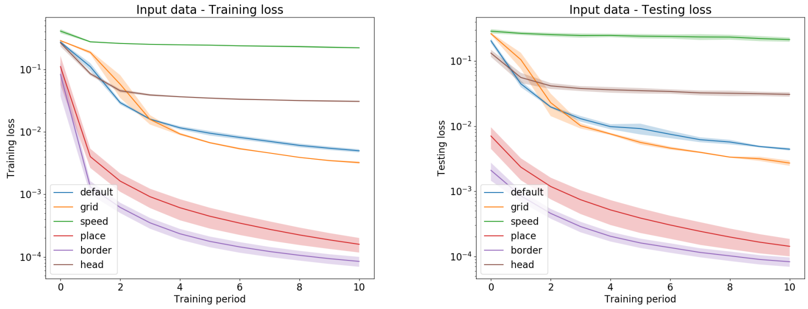

Data
Initialization: Starting position
Input: grid-, place-, border-, head-direction, and speed cells
Output: Trajectory positions
Recurrent state initialization: Linear layer [2 x 32]
Cell type: LSTM
Neurons: 32
Activation function: Hyperbolic tangent
Recurrent kernel: Glorot normal initialization
Recurrent Layer
Entorhinal
recurrent output
Output layer: Linear layer [32 x 2]
predicted trajectory
Error loss: L2 difference between target and prediction
target

Figure 4.29: **Entorhinal simulation model**
Basic setup for replacing the input data with entorhinal-like representation.

The artificially generated entorhinal-like representation simulates the activation pattern of individual cells based on their theoretical activation functions. The cells have a randomly assigned center position or head-direction depending on the cell type on which the activation calculation is performed. Each cell uses available data, such as the spatial position for grid- and place-like cells, the velocity for speed cells, or moving direction for simulating head-direction and border-like cells. For more information on the activation functions for the different artificially generated entorhinal cells, see Appendix D.

4.6.1 Input Features Replacement

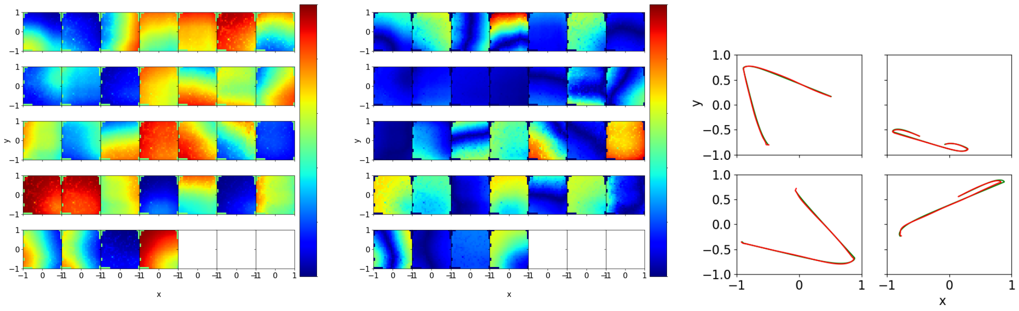
Replacing the input data requires generating entorhinal-like representations based on the positional, velocity, and directional data. The input data available during the training process is solely based on the entorhinal-like input features; the directional and velocity data are not available during training. Each experiment tests a different entorhinal-like representation replacement, consisting of grid, place, border, speed, and head-direction cells.



(a) Training loss

(b) Testing loss

Figure 4.30: **Entorhinal input replacement losses**
The loss plot visualizes the trained loss of between input replacement techniques inspired by entorhinal-like representation. Input techniques differences shows that the border- and place cells are more optimal compared to the baseline configuration (default).



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. The internal activation strategy of neurons is primarily geared towards spatial activation.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. More neurons are spatially active and influence the output prediction compared to previous experiments.

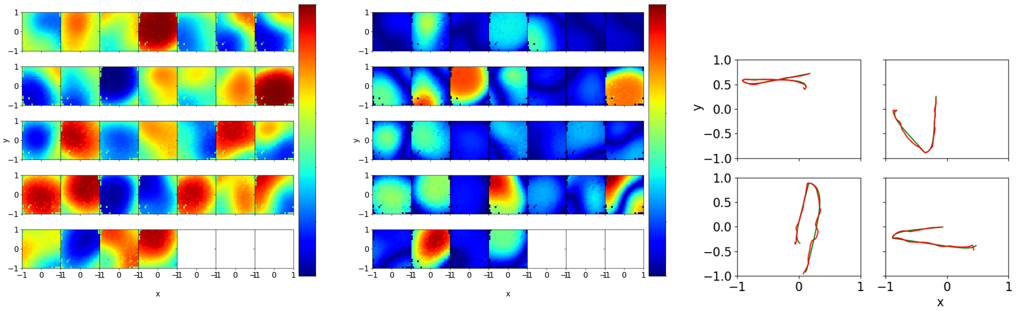
(c) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. Precision of trajectory prediction is very close to optimal.

Figure 4.31: **Entorhinal border cell input representation and performance**
Border cell input replacement performs best compared to other entorhinal-like representation. Possibly the dominant spatial activation and shared influence of multiple neurons is part of this improvement.

The performance differences are shown in the loss plot (Figure 4.30). From these results, we can conclude that border- and place-like representation can improve over the hybrid baseline model. Whereas, the speed, head-direction, and grid-like cells show bad performance or do not significantly improve upon the baseline performance.

Looking more closely at the representation associated with border cells (Figure 4.31) shows that the recurrent cell creates linear and non-linear spatial representation from border-like input. The input variety provided by multiple border-like cells could explain the diversity of spatially activated neurons. Also, the trajectory performance plot (Figure 4.31c) shows robust accuracy and overlap of the predicted and target trajectories. The only explanation available for the improved performance is either the non-linear spatially active neurons or the fact that the influence plot shows more neurons participating in the path-integration process as before (Figure 4.31b).

Place cells are also an excellent approach to replace the input data according to the loss comparisons (Figure 4.30). The spatial activity resulting from a network only trained by place-like input data Figure 4.32a seems to carry-over the blob-like activation pattern. The spatial representation does show consistent confined place-dependent peaks. Again, the spatial influence plot highlights the shared contribution of multiple neurons for prediction the output for path-integration (Figure 4.32b). The performance, according to Figure 4.32c, shows that the trajectory overlap is similar, but less accurate compared to the border-like input replacement approach.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. The spatial activity plots show reminiscent activation from the place cell input distribution.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. Each cells uses the place-like activation to represent a part of the spatial environment.

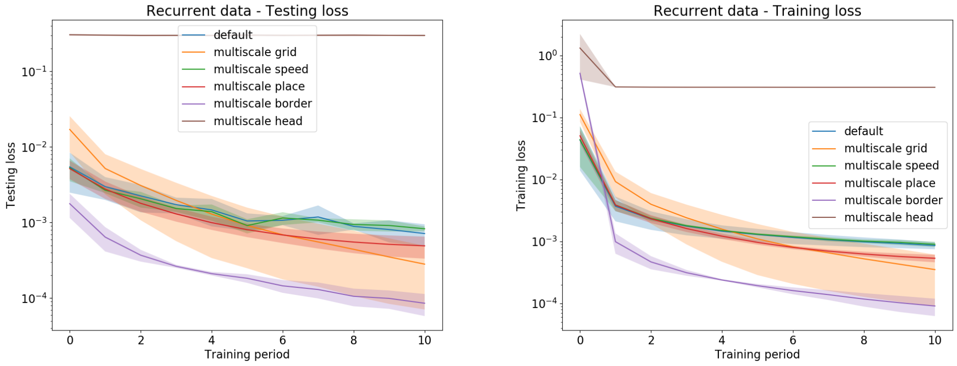
(c) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. Trajectory prediction has good performance but is more noisy in predicting straight-forward paths.

Figure 4.32: **Entorhinal place cell input representation and performance**
The recurrent cell maintains and combines place-like representations, which creates a small clusters of spatial activity patterns, providing good path-integration performance.

4.6.2 Recurrent Cell Replacement

Alternatively, the activity of the recurrent neurons can be replaced using the different entorhinal-like activation distributions. A multi-scale approach is introduced to ensure enough representational diversity. Especially grid cells appear to exist in multiple discrete scales [11]. This scaling could be necessary for achieving scalable solutions, especially when the recurrent cell is bypassed. The multi-scale option replaces the recurrent cell state with entorhinal-like representation based on the position, velocity, and direction of the trajectory. Each cell has a random scale associated with each cell representation.

The multi-scale recurrent replacement results are displayed in Figure 4.33. The border representation performs even better compared to the input replacement approach. Additionally, the multi-scale approach helps grid cells improve compared to the hybrid baseline model.

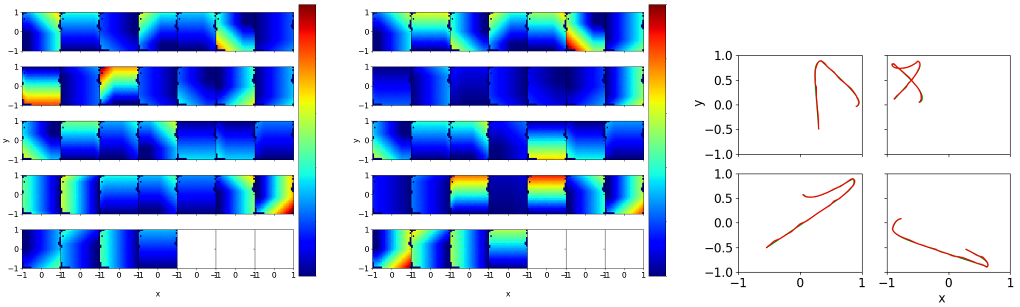


(a) Training loss

(b) Testing loss

Figure 4.33: **Entorhinal recurrent replacement losses**
The loss plot visualizes the trained loss differences between recurrent cell replacement techniques inspired by entorhinal-like representation. The testing loss shows only improved performance for border- and grid-like cells.

The best performing recurrent cell replacement is the border cell. Looking closely at the representation for artificially generated border cells highlights a regular maximum plot pattern (Figure 4.34). The performance of multi-scale border cells is an order of magnitude better compared to the entorhinal input replacement experiments. This performance leap could be due to the linear activation of cells leading to more consistent tracking of positions within the 2D environment.

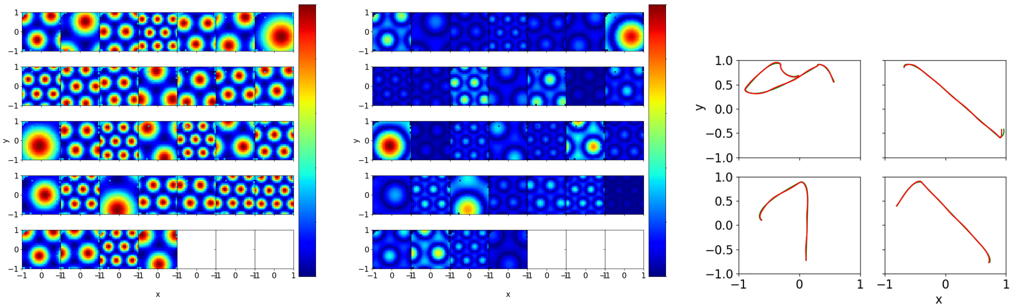


(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Each cells is assigned a particular head-direction angle, which results in direction dependent border-like activaty patterns.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. The spatial influence is shared by many border-like neurons collaborating to provide path-integration capabilities.

(c) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. Predicted trajectories are close to optimal, showing good path-integration performance.

Figure 4.34: **Entorhinal multi-scale recurrent replacement border cell**
The recurrent cell is replaced by border-like representation, which creates shared spatial activity patterns, providing good path-integration performance.



(a) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron. Grid cells have a varying scale allowing the network to train in different granularities.

(b) **Spatial influence plot**
Each plot cell visualizes the average spatial influence of an artificial neuron. Both place-like and grid-like representations contribute to the output prediction of the agent's trajectory.

(c) **Trajectory performance plot**
The overlapping trajectories are visualized with the target path in green and the predicted path in red. The differences between the trajectories are minimal.

Figure 4.35: **Recurrent entorhinal multi-scale grid cell**
The recurrent cell is replaced by grid-like representation, which creates shared activity patterns, providing good path-integration performance.

The grid-like representation shows slight improvements over the hybrid baseline model.

Let us take a closer look at the representation and performance of the grid-like recurrent replacement approach. The spatial activity plot and influence plot in Figure 4.35 indicate that both the large scales and small scales contribute towards path integration. Additionally, the trajectory differences in Figure 4.35c are minimal. Leading us to conclude that both border- and grid-like representation are both suitable improvements for performing path-integration.

These experiments show that individual cells can improve upon the standard performance of path-integration. An important note, however, is the fact that entorhinal-like representation currently relies on output data using the spatial position and environmental shape, which simplifies the path-integration process. It could be possible that the diversity of many entorhinal-like cells can remove this dependency on the trajectory position by integrating spatial information internally. The interaction between entorhinal-like representation could provide an essential role in reducing the performance error and maintaining the stability of grid-like representations.

4.7 Summary

The results discussed simplified models of earlier work, a hybrid baseline model, different input features, additional structural properties, various regularization techniques, and performance evaluation of entorhinal-like representation in recurrent neural networks. All in all, the analysis has not shown consistent results regarding the emergence of entorhinal-like representation in recurrent neural networks. Almost all experiments show similar representations, consisting of linear spatial and directional activation patterns. The regularization technique experiments showed glimpses of specialization; however, the representation is not representative of entorhinal-like representation. The regularization strategies either create extremely local specialization or regular linear spatial activity gradients. Furthermore, experiments with ray-tracing input showed distinct activity representation with several neurons having more arched activation patterns compared to the baseline model. These non-linear activation patterns could eventually lead to entorhinal-like representation; however, at this point in the experimentation, it is not clear whether this pattern will evolve to biologically-plausible representation. Additionally, replacing the input features and recurrent states with entorhinal-like representation can improve performance with border-, place-, and grid-like representation. However, in its current form, it is not directly possible to improve the performance of neural networks due to relying on positional oracle data.

The next chapter finishes up this report by discussing the study and its findings regarding the emergence of entorhinal-like representation in recurrent neural networks.

Chapter 5

Discussion and Conclusion

This thesis analyzed possible factors influencing emerging entorhinal-like representation and performance in recurrent neural networks. This chapter will reflect on the extent that this study was able to uncover this phenomenon using a summary of the study, its findings, implications, conclusions, and recommendations for future research.

5.1 Study Reflection

Although Banino et al. [1] and Cueva and Wei [2] showed impressive results regarding the emergence of entorhinal-like representation in recurrent neural networks. Both studies lack a shared consensus for emergent entorhinal-like representation and the training performance associated with this type of representation.

Neuroscience points to the competitive nature between neurons as a possible explanation for the emergence of entorhinal-like representation [60]. While the Metabolic constraint [2] and Dropout [1] regularization techniques could follow a similar interpretation, the authors claim that regularization is related to the efficient coding hypothesis [122] and the introduction of noise to the network, respectively.

No theoretical models exist regarding the emergence of entorhinal-like representation in recurrent neural networks, and this leads to exhaustive experimentation on the potential factors causing this phenomenon. The experiments are grouped into three clusters: replicating simplified models from previous studies, analyzing potential factors influencing the emergence, and determining the performance difference of entorhinal-like representation in recurrent neural networks.

Replicating previous studies enables investigating the robustness of emergent entorhinal-like representation in simplified network models. The experimental investigation of underlying factors can lead to different hypotheses on the influence of input features, structural properties, and the role of regularization techniques. Additionally, studying the training performance of entorhinal-like representation helps to unveil the potential of the reported emerging representation.

The exhaustive exploration of possible factors did not develop emergent entorhinal-like representation. However, performance experiments highlighted improved path-integration performance using some of the entorhinal-like representations. For a visual representation of the research design and findings see Appendix E.

5.1.1 Study Findings

While the exhaustive experimentation helped shed light on the research questions, an additional goal of this research was to evaluate different analysis techniques for visualizing the emergence of entorhinal-like representation.

5.1.2 Emergent Representation

The replicated experiments showed that the simplified model by Banino et al. [1] and Cueva and Wei [2] was mainly dependent on direction. Since the neurons were mainly active for a particular angle. Both experiments did not show grid-like representations as the original studies claimed. It is possible to conclude that the reported representations are tied to the original model configurations and training process used by both studies [1, 2] which are not directly replicable. This model dependence highlights the fact that the reported models are not robust against model variations and trivial differences in the regularization approach.

The experiments regarding the input features highlighted suboptimal performance when using uni-modal input using velocity and direction input. For example, the multi-modal approach, called ray-tracing, generated non-linear spatial representations and improved performance significantly. The remaining input approaches, splitting the directional variable into two features and introducing noise, were not useful for improving the performance in a recurrent neural network. One interesting thing to note is that the split direction approach allowed the model to learn a Gaussian-like directional dependence, similar to head-direction cells. This observation could be the first sign of emergent entorhinal-like representation, but more research is necessary to either confirm or refute this hypothesis.

Both the network architecture and training procedure are explored in the structural properties experiments. These factors range from the number of hidden cells, recurrent cell type, initialization strategy, learning rate, and optimizer choice. Changing the recurrent cell to a GRU, having more neurons, and using the RMSProp optimizer can optimize the performance. However, these configurations did not develop entorhinal-like representation. Thus concluding that structural properties are essential for optimizing performance, but, they are not directly associated with entorhinal-like representation.

Results regarding regularization showed signs of specialization but were not sufficient to serve as entorhinal-like representation. Both Winner-Takes-All and an experimental regularization approach called Neural Fatigue showed spatially clustered recurrent representation. However, this type of regularization can not replicate the geometric complexity of place or grid-like cells from the entorhinal cortex. Additionally, these techniques performed worse compared to the baseline model. These observations show that regularization can assist the neural activity to specialize spatially; however, our experiments did not show that regularization led to entorhinal-like representation. Previous work and the results from this thesis suggest that regularization, when optimized correctly, could balance the local specialization and global coordination necessary to support entorhinal-like representation.

The training performance experiments assessed the relation between entorhinal-like representation and recurrent neural network performance. Border- and place-like input representations can improve upon the baseline model. Also, replacing the recurrent output with border-like and grid-like representation has the potential to improve the performance significantly. From these results, it is possible to conclude that the artificial place-, grid-, and border-like representation can improve path-integration performance in recurrent neural networks. However, more work is necessary to remove the oracle, which is central to generating artificial entorhinal-like representation.

5.1.3 Visualization Techniques

Spatial activity plots are widely adopted by researchers to visualize entorhinal-like representation, such as grid-, place-, and border-like cells. The non-spatial activity plot can help formulate alternative hypotheses on the activation strategy of neurons [2, 146]. One aspect missing in these plots is the representational variance. The plots only show the mean spatial activity. Having a low variance for the entire spatial or non-spatial plotting range is necessary for concluding any link between the stated representation and biologically-plausible representation. Despite lacking variance calculation, activity plots will remain one of the most insightful analysis techniques available for detecting the emergence of entorhinal-like representation.

Two new representational plots introduced in this study include the maximum spatial plot and spatial influence plot. The maximum spatial plot helped to aggregate the activity of many neurons in one plot, which presents an overview of the collaboration between neurons. Also, the spatial influence plot provided additional information alongside the spatial activity plot on the value of neural representation.

Lastly, introducing two performance analysis plots (trajectory performance plot and average performance plot) helped to reflect on the training progress during the learning process. These plots helped to visualize adverse training results and robustness of prediction accuracy during experimentation. While these plots do not directly relate to the underlying representation of entorhinal-like representation, the performance analysis is still an essential tool for monitoring the training progress in neural networks.

5.2 Conclusions

While entorhinal-like representation can improve the performance of recurrent neural networks, it is still unclear which factors are responsible for the emergence of biologically-plausible representation. One possible explanation for this phenomenon is regularization, which can ensure competition and specialization among cells necessary for entorhinal-like representation to emerge. However, this thesis was unable to replicate and explore additional factors regarding the emergence of entorhinal-like representation.

Replicating conditions necessary for the emergence of entorhinal-like representation has turned into chasing a white whale. Other studies have shown their occurrence in recurrent neural networks, but this study was not able to replicate similar representations. Nevertheless, the results highlighted essential observations for researching and developing with recurrent neural networks, including verifying better performance through multi-modal input features, performing a structured analysis of recurrent network properties, and introducing alternative regularization techniques for competition and specialization.

5.3 Future research

The two hardest questions left by this study remain the purpose of regularization in emergent entorhinal-like representation and the relationship between representation and performance. Unfortunately, this research was unable to replicate spatial entorhinal-like representation; therefore, the results were not able to verify the specialization hypothesis or other potential roles of representation in recurrent neural networks.

The results should stimulate future research to study the emergence of biologically-plausible representations in recurrent neural networks [147]. Especially the relation between entorhinal-like representation and many other cognitive abilities should motivate investigation into the potential of biologically-plausible representation and the application of entorhinal-like representation in deep learning and cognitive robotics [50, 148].

Perhaps replicating the exact network configurations from Banino et al. [1] and Cueva and Wei [2] is necessary for observing entorhinal-like representation. From there, it is possible to modify the network or introduce new factors in efforts to reduce the differences between reported models and create novel hypotheses on the emergence of entorhinal-like representation. Additionally, the use of realistic behavioral simulation approaches by earlier work could relate to the emergence of biologically-plausible representation, and should not be excluded from strict replication studies.

Border-, grid-, and place-like representation has shown to be more effective at performing path-integration, possibly due to having access to oracle information about the position of the agent. More investigation on these types of representation could lead to removing the oracle and creating stable spatial and non-spatial representation simultaneously. One approach would involve theoretical models to explore this concept more formally before applying it to recurrent neural networks.

In addition to exploring the underlying entorhinal-like representations separately, an alternative 'diverse' approach could use a combination of different entorhinal-like representations. This approach is motivated by the fact that spatial navigation requires the entire spectrum of entorhinal-like representations. Especially finding the correct balance between different cell types will be critical for creating stable entorhinal-like representation and assessing the ability to optimizing performance.

Additionally, when the mentioned approaches focused on developing emergent entorhinal-like representation in recurrent neural networks are not productive, then research might require reproducing hippocampal circuits in search of emerging entorhinal-like representation. This research approach would combine the neuroscience paradigm on competitive neural communication [60], possibly through regularization, and the emergence of biologically-plausible representation in deep learning [149, 150].

Bibliography

[1] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-based Navigation using Grid-like Representations in Artificial Agents. 2018.

[2] Christopher J Cueva and Xue-xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. pages 1–19, mar 2018.

[3] Russell A Epstein, Eva Zita Patai, Joshua B Julian, and Hugo J Spiers. The cognitive map in humans: spatial navigation and beyond. (October), 2017.

[4] Aidan J. Horner, James A. Bisby, Ewa Zotow, Daniel Bush, and Neil Burgess. Grid-like processing of imagined navigation. *Current Biology*, 26(6):842–847, 2016.

[5] Christian F Doeller, Caswell Barry, and Neil Burgess. Evidence for grid cells in a human memory network. *Nature*, 463(7281):657–661, 2010.

[6] David C. Rowland, Yasser Roudi, May-Britt Moser, and Edvard I. Moser. Ten Years of Grid Cells. *Annual Review of Neuroscience*, 39(1):19–40, 2016.

[7] Stefan Leutgeb, Jill K. Leutgeb, May Britt Moser, and Edvard I. Moser. Place cells, spatial maps and the population code for memory. *Current Opinion in Neurobiology*, 15(6):738–746, 2005.

[8] Trygve Solstad, Charlotte N Boccara, Emilio Kropff, May-britt Moser, and Edvard I Moser. Representation of Geometric Borders in the Entorhinal Cortex. *Methods*, 1865(December):1865–1869, 2008.

[9] Emilio Kropff, James E. Carmichael, May Britt Moser, and Edvard I. Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.

[10] William N. Butler, Kyle S. Smith, Matthijs A.A. van der Meer, and Jeffrey S. Taube. The Head-Direction Signal Plays a Functional Role as a Neural Compass during Navigation. *Current Biology*, 27(9):1259–1267, 2017.

[11] Vegard Edvardsen. Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network. *Natural Computing*, (1):1–15, 2016.

[12] Edvard I. Moser, Torkel Hafting, May-Britt Moser, Sturla Molden, and Marianne Fyhn. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

[13] I.T. Mark, A.P. Klein, F.D. Raslau, V. Mathews, J.L. Ulmer, and L.P. Mark. Memory Part 2: The Role of the Medial Temporal Lobe. *American Journal of Neuroradiology*, 36(5):846–849, 2014.

[14] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomas Krajnik. Artificial Intelligence for Long-Term Robot Autonomy: A Survey. *IEEE Robotics and Automation Letters*, 3(4):4023–4030, 2018.

[15] Li Li, Yi Lun Lin, Nan Ning Zheng, Fei Yue Wang, Yuehu Liu, Dongpu Cao, Kunfeng Wang, and Wu Ling Huang. Artificial intelligence test: a case study of intelligent vehicles. *Artificial Intelligence Review*, 50(3):441–465, 2018.

[16] Jingwei Zhang, Jost Tobias Springenberg, Joschka Boedecker, and Wolfram Burgard. Deep reinforcement learning with successor features for navigation across similar environments. *IEEE International Conference on Intelligent Robots and Systems*, 2017-Septe:2371–2378, 2017.

[17] Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, and Sergey Levine. Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation. sep 2017.

[18] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to Navigate in Complex Environments. 2017.

[19] Piotr Mirowski, Matthew Koichi Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, and Raia Hadsell. Learning to Navigate in Cities Without a Map. (NeurIPS):1–12, 2018.

[20] Peter Karkus, David Hsu, and Wee Sun Lee. Integrating Algorithmic Planning and Deep Learning for Partially Observable Navigation.

[21] Wei Gao, David Hsu, Wee Sun Lee, Shengmei Shen, and Karthikk Subramanian. Intention-Net: Integrating Planning and Deep Learning for Goal-Directed Autonomous Navigation. (Figure 1):1–10, 2017.

[22] Ardi Tampuu, Tambet Matiisen, H. Freyja Olafsdottir, Caswell Barry, and Raul Vicente. Efficient neural decoding of self-location with a deep recurrent network. *bioRxiv*, pages 1–20, 2018.

[23] Vikas Dhiman, Shurjo Banerjee, Brent Griffin, Jeffrey M Siskind, and Jason J Corso. A Critical Investigation of Deep Reinforcement Learning for Navigation. 2018.

[24] Shurjo Banerjee, Vikas Dhiman, and Jason J. Corso. Do deep reinforcement learning algorithms really learn to navigate? pages 1–11, 2018.

[25] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun. Benchmarking Classic and Learned Navigation in Complex 3D Environments. 2019.

[26] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A Study on Overfitting in Deep Reinforcement Learning. pages 1–25, 2018.

[27] Vegard Edvardsen. Long-Range Navigation by Path Integration and Decoding of Grid Cells in a Neural Network. 2017.

[28] Ruiqi Gao, Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning Grid-like Units with Vector Representation of Self-Position and Matrix Representation of Self-Motion. 1(2005):1–14, 2018.

[29] James C. R. Whittington, Timothy H. Muller, Caswell Barry, and Timothy E. J. Behrens. Generalisation of structural knowledge in the Hippocampal-Entorhinal system. (Nips), 2018.

[30] Dane Corneil. Attractor Network Dynamics Enable Preplay and Rapid Path Planning in Maze – like Environments. pages 1–9.

[31] Edward C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.

[32] Lucia F. Jacobs. The Evolution of the Cognitive Map. *Brain, Behavior and Evolution*, 62(2):128–139, 2003.

[33] Adam Johnson and David A. Crowe. Revisiting Tolman, his Theories and Cognitive Maps. pages 111–127, 2012.

[34] K L Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. Design Principles of the Hippocampal Cognitive Map. *Advances in Neural Information Processing Systems 27*, pages 1–9, 2014.

[35] Hugo J. Spiers and Caswell Barry. Neural systems supporting navigation. *Current Opinion in Behavioral Sciences*, 1:47–55, 2015.

[36] François Rivest, Yoshua Bengio, and John Kalaska. Brain Inspired Reinforcement Learning. *Advances in neural information processing systems*, pages 1129–1136, 2015.

[37] Rafael Yuste. From the neuron doctrine to neural networks. *Nature Reviews Neuroscience*, 16(8):487–497, 2015.

[38] D. Schiller, H. Eichenbaum, E. A. Buffalo, L. Davachi, D. J. Foster, S. Leutgeb, and C. Ranganath. Memory and Space: Towards an Understanding of the Cognitive Map. *Journal of Neuroscience*, 35(41):13904–13911, 2015.

[39] Howard Eichenbaum. The Hippocampus as a Cognitive Map ... of Social Space. *Neuron*, 87(1):9–11, 2015.

[40] Oliver M. Vikbladh, Michael R. Meager, John King, Karen Blackmon, Orrin Devinsky, Daphna Shohamy, Neil Burgess, and Nathaniel D. Daw. Hippocampal Contributions to Model-Based Planning and Spatial Memory. *Neuron*, 102(3):683–693.e4, 2019.

[41] Jacob L.S. Bellmund, Peter Gärdenfors, Edvard I. Moser, and Christian F. Doeller. Navigating cognition: Spatial codes for human thinking. *Science (New York, N.Y.)*, 362(6415), 2018.

[42] George Konidaris. On the necessity of abstraction. *Current Opinion in Behavioral Sciences*, 29:1–7, 2019.

[43] Timothy E.J. Behrens, Timothy H. Muller, James C.R. Whittington, Shirley Mark, Alon B. Baram, Kimberly L. Stachenfeld, and Zeb Kurth-Nelson. What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior. *Neuron*, 100(2):490–509, 2018.

[44] Marcel van Gerven. Computational Foundations of Natural Intelligence. *Frontiers in Computational Neuroscience*, 11(December):1–24, 2017.

[45] Nikolaus Kriegeskorte and Pamela K. Douglas. Cognitive computational neuroscience. *Nature Neuroscience*, 21(9):1148–1160, 2018.

[46] David Daniel Cox and Thomas Dean. Neural networks and neuroscience-inspired computer vision. *Current Biology*, 24(18):R921–R929, 2014.

[47] Volodymyr Mnih, Koray Kavukvuoglu, and David Silver. Human-level control through deep reinforcement Learning. *IJCAI International Joint Conference on Artificial Intelligence*, 2016-Janua:2315–2321, 2016.

[48] Junjun Li, Zhijun Li, Fei Chen, Antonio Bicchi, Yu Sun, and Toshio Fukuda. Combined Sensing, Cognition, Learning and Control to Developing Future Neuro-Robotics Systems: A Survey. *IEEE Transactions on Cognitive and Developmental Systems*, PP(c):1–1, 2019.

[49] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-Inspired Artificial Intelligence. *Neuron*, 95(2):245–258, 2017.

[50] Katherine R. Storrs and Nikolaus Kriegeskorte. Deep Learning for Cognitive Neuroscience. pages 1–26, 2019.

[51] Steven Poulter, Tom Hartley, and Colin Lever. The Neurobiology of Mammalian Navigation. *Current Biology*, 28(17):R1023–R1042, 2018.

[52] Brandy Schmidt, Andrew M. Wikenheiser, and A. David Redish. *Goal-Directed Sequences in the Hippocampus*. Elsevier Inc., 2018.

[53] Lukas Kunz, Tobias Navarro Schröder, Hweeling Lee, Christian Montag, Bernd Lachmann, Rayna Sariyska, Martin Reuter, Rüdiger Stirnberg, Tony Stöcker, Paul Christian Messing-Floeter, Juergen Fell, Christian F. Doeller, and Nikolai Axmacher. Reduced grid-cell-like representations in adults at genetic risk for Alzheimer's disease. *Science*, 350(6259):430–433, 2015.

[54] Henry Markram. The Blue Brain Project. *Engineering in Medicine and Biology Society*, 7(February):153–160, 2006.

[55] Erwin Prassler, Mario E. Munich, Paolo Pirjanian, and Kazuhiro Kosuge. Domestic Robotics. In *Springer Handbook of Robotics*, pages 1253–1281. 2008.

[56] Lisa M. Giocomo, May Britt Moser, and Edvard I. Moser. Computational models of grid cells. *Neuron*, 71(4):589–603, 2011.

[57] Tobias Navarro Schroeder, Benjamin W Towse, Matthias Nau, Neil Burgess, Caswell Barry, and Christian F Doeller. Entorhinal cortex minimises uncertainty for optimal behaviour. *bioRxiv*, page 166306, 2018.

[58] Peter E. Welinder, Yoram Burak, and Ila R. Fiete. Grid cells: The position code, neural network models of activity, and the problem of learning. *Hippocampus*, 18(12):1283–1300, dec 2008.

[59] Benjamin J. Clark and Jeffrey S. Taube. Vestibular and attractor network basis of the head direction cell signal in subcortical circuits. *Frontiers in Neural Circuits*, 6(March):1–12, 2012.

[60] Louis Kang and Vijay Balasubramanian. A geometric attractor mechanism for self-organization of entorhinal grid modules. pages 1–11, 2018.

[61] Ulises Rodríguez-Domínguez and Jeremy B. Caplan. A hexagonal Fourier model of grid cells. *Hippocampus*, 29(1):37–45, 2019.

[62] Emilio Kropff and Alessandro Treves. The emergence of grid cells: Intelligent design or just adaptation? *Hippocampus*, 18(12):1256–1269, dec 2008.

[63] Garrison W Cottrell. Attractor Networks. *Encyclopedia of Cognitive Science*, page 5, 2006.

[64] H. Sebastian Seung. Learning Continuous Attractors in Recurrent Networks. *Advances in Neural Information Processing Systems*, 10:654–660, 1997.

[65] Xuelong Sun, Michael Mangan, and Shigang Yue. An Analysis of a Ring Attractor Model for Cue Integration. *Biomimetic and Biohybrid Systems: Living Machines 2018*, 2018.

[66] Edvard I. Moser, May Britt Moser, and Yasser Roudi. Network mechanisms of grid cells. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1635), 2014.

[67] Fabio Anselmi, Benedetta Franceschiello, Micah M. Murray, and Lorenzo Rosasco. A computational model for grid maps in neural populations. 2019.

[68] Juan I Sanguinett-Scheck and Michael Brecht. Home, head direction stability and grid cell distortion. *bioRxiv*, page 602771, 2019.

[69] Nicolai Waniek. Transition scale-spaces : A computational theory for the discretized entorhinal cortex. 2019.

[70] Daniel Bush, Caswell Barry, and Neil Burgess. What do grid cells contribute to place cell firing? *Trends in Neurosciences*, 37(3):136–145, 2014.

[71] Roddy M Grieves, Selim Jedidi-ayoub, Karyna Mishchanchuk, Anyi Liu, Sophie Renaudineau, and Kate J Jeffery. The place-cell representation of volumetric space in rats. pages 1–31, 2019.

[72] Trygve Solstad, Edvard I. Moser, and Gaute T. Einevoll. From Grid Cells to Place Cells: A Mathematical Model. *International Journal of Pharmaceutical Sciences Review and Research*, 37(1):36–41, 2016.

[73] J. O'Keefe, S. Burton, A. Jeewajee, C. Lever, and N. Burgess. Boundary Vector Cells in the Subiculum of the Hippocampal Formation. *Journal of Neuroscience*, 29(31):9771–9777, 2009.

[74] Iva K. Brunec, Morris Moscovitch, and Morgan D. Barense. Boundaries Shape Cognitive Representations of Spaces and Events. *Trends in Cognitive Sciences*, 22(7):637–650, 2018.

[75] Bruce L McNaughton, Francesco P Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the 'cognitive map'. *Nature reviews. Neuroscience*, 7(8):663–78, 2006.

[76] Rosamund F. Langston, James A. Ainge, Jonathan J. Couey, Cathrin B. Canto, Tale L. Bjerknes, Menno P. Witter, Edvard I. Moser, and May-Britt Moser. Development of the Spatial Representation System in the Rat. *Science*, 321(September):1673–1675, 2008.

[77] Benjamin Kuipers, Dan G. Tecuci, and Brian J. Stankiewicz. The Skeleton In The Cognitive Map. *Environment and Behavior*, 35(1):81–106, jan 2003.

[78] Matthew F. Nolan. Neural Mechanisms for Spatial computation. *The Journal of Physiology*, 594(22):6487–6488, 2016.

[79] Mengmi Zhang, Keng Teck Ma, Joo Hwee Lim, Shih-Cheng Yen, Qi Zhao, and Jiashi Feng. Egocentric Spatial Memory Network. pages 1–16, 2018.

[80] Kiah Hardcastle, Surya Ganguli, and Lisa M. Giocomo. Cell types for our sense of location: Where we are and where we are going. *Nature Neuroscience*, 20(11):1474–1482, 2017.

[81] Alon B. Baram, Timothy H. Muller, James C. R. Whittington, and Timothy E.J. Behrens. Intuitive planning: global navigation through cognitive maps based on grid-like codes. *bioRxiv*, page 8, 2018.

[82] John Lisman, György Buzsáki, Howard Eichenbaum, Lynn Nadel, Charan Ranganath, and A. David Redish. Viewpoints: how the hippocampus contributes to memory, navigation and cognition. *Nature Neuroscience*, 20(11):1, 2017.

[83] Teruko Danjo, Taro Toyoizumi, and Shigeyoshi Fujisawa. Spatial representations of self and other in the hippocampus. *Science*, 359(6372):213–218, jan 2018.

[84] David B Omer, Shir R Maimon, Liora Las, and Nachum Ulanovsky. Supplementary Material - Social place-cells in the bat hippocampus. *Science*, 359(January):218–224, 2018.

[85] Howard Eichenbaum. Time (and space) in the hippocampus. *Current Opinion in Behavioral Sciences*, 17:65–70, 2017.

[86] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and Understanding Recurrent Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, jun 2015.

[87] Hasim Sak, Andrew Senior, and Francoise Beaufays. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. (September):338–342, 2014.

[88] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks. *Proceedings of the 30th International Conference on Machine Learning*, 28(2):9, 2013.

[89] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jurgen Schmidhuber. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. Technical report, 2001.

[90] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *IEEE International Conference on Rehabilitation Robotics*, 2015-Septe:119–124, dec 2014.

[91] Wojciech Zaremba and Oriol Vinyals Ilya Sutskever. Recurrent Neural Network Regularization. pages 1–8, 2014.

[92] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. Overcoming the vanishing gradient problem in plain recurrent networks. (Section 2):1–20, 2018.

[93] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN. (1), 2018.

[94] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[95] Rafa Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An Empirical Exploration of Recurrent Network Architectures. *International Conference on Machine Learning*, pages 2342–2350, sep 2015.

[96] Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. Understanding Hidden Memories of Recurrent Neural Networks. oct 2017.

[97] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. (Ml):1–13, feb 2017.

[98] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. iNNvestigate neural networks! aug 2018.

[99] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining Recurrent Neural Network Predictions in Sentiment Analysis. 2017.

[100] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the Loss Landscape of Neural Nets. *NIPS*, dec 2017.

[101] Zachary C. Lipton. The Mythos of Model Interpretability. (Whi), jun 2016.

[102] Quan-shi Zhang and Song-chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[103] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks Through Deep Visualization. *The New phytologist*, 197(3):909–18, jun 2015.

[104] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush. LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks. jun 2016.

[105] Paulo E. Rauber, Samuel G. Fadel, Alexandre X. Falcão, and Alexandru C. Telea. Visualizing the Hidden Activity of Artificial Neural Networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, 2017.

[106] Roman V Yampolskiy. Unexplainability and Incomprehensibility of Artificial Intelligence. 2019.

[107] David GT Barrett, Ari S. Morcos, and Jakob H. Macke. Analyzing biological and artificial neural networks: challenges with opportunities for synergy? *Current Opinion in Neurobiology*, 55:55–64, 2019.

[108] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for Deep Learning: A Taxonomy. pages 1–23, 2017.

[109] Pirmin Lemberger. On Generalization and Regularization in Deep Learning. pages 1–11, 2017.

[110] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2016.

[111] Yarin Gal and Zoubin Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *30th Conference on Neural Information Processing Systems (NIPS 2016)*, 2016.

[112] Yarin Gal. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. 48, 2016.

[113] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. (Nips 2017), 2017.

[114] Ilya Sutskever, Geoffrey Hinton, Alex Krizhevsky, and Ruslan R Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[115] Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, and Vittorio Murino. Curriculum Dropout. pages 3544–3552, 2017.

[116] Lei Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. pages 1–9, 2013.

[117] Zhe Li, Boqing Gong, and Tianbao Yang. Improved Dropout for Shallow and Deep Learning. (Nips):1–9, 2016.

[118] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. pages 1–5, 2012.

[119] Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. DeepMind Lab. pages 1–11, 2016.

[120] Jochen Kerdels and Gabriele Peters. A Survey of Entorhinal Grid Cell Properties. pages 1–34, 2018.

[121] Matthew Botvinick, Ari Weinstein, Alec Solway, and Andrew Barto. Reinforcement learning, efficient coding, and the statistics of natural tasks. *Current Opinion in Behavioral Sciences*, 5:71–77, 2015.

[122] Matthew Chalk, Olivier Marre, and Gašper Tkačik. Toward a unified theory of efficient, predictive, and sparse coding. *Proceedings of the National Academy of Sciences*, 115(1):186–191, 2017.

[123] William Bialek, Rob R. De Ruyter Van Steveninck, and Naftali Tishby. Efficient representation as a design principle for neural coding and computation. *IEEE International Symposium on Information Theory - Proceedings*, pages 659–663, 2006.

[124] Karol Gregor, Arthur Szlam, and Yann LeCunn. Structured sparse coding via lateral inhibition. *Advances in Neural Information Processing Systems*, pages 1116–1124, 2011.

[125] Yihong Wang, Xuying Xu, and Rubin Wang. The place cell activity is information-efficient constrained by energy. *Neural Networks*, 116:110–118, 2019.

[126] Florian Raudies and Michael E. Hasselmo. Modeling boundary vector cell firing given optic flow as a cue. *PLoS Computational Biology*, 8(6), 2012.

[127] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. pages 1–4, 2016.

[128] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. *IJCAI International Joint Conference on Artificial Intelligence*, 2016-Janua:4246–4247, 2016.

[129] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning. *Annals of Oncology*, 25(6):1204–1208, may 2016.

[130] Joel Z Leibo, Cyprien De Masson, Daniel Zoran, David Amos, Charles Beattie, Keith Anderson, Antonio García Castañeda, Manuel Sanchez, Simon Green, Audrunas Gruslys, Shane Legg, Demis Hassabis, and Matthew M Botvinick. Psychlab: A Psychology Laboratory for Deep Reinforcement Learning Agents. pages 1–28, 2018.

[131] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A General Platform for Intelligent Agents. pages 1–18, 2018.

[132] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234(December 2016):11–26, 2017.

[133] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal Deep Learning. *Proceedings of the 28th Annual International Conference on Machine Learning (ICML'11)*, pages 689–696, 2011.

[134] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[135] Klaus Greff, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, and Jurgen Schmidhuber. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, oct 2017.

[136] Pierre Baldi and Roman Vershynin. Neuronal Capacity. 92697(1):1–10.

[137] Pierre Baldi and Roman Vershynin. The capacity of feedforward neural networks. *Neural Networks*, 116:288–311, 2019.

[138] James Martens and Geo Hinton. On the importance of initialization and momentum in deep learning. (2010), 2012.

[139] J Braun, D.K. Lee, L Itti, and C Koch. Attention activates winner-take-all competition among visual filters. *Nature Neuroscience*, 2(4):375–381, 1999.

[140] Sen Song, Kenneth D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, 2000.

[141] Steven E Petersen and Olaf Sporns. Brain Networks and Cognitive Architectures. *Neuron*, 88(1):207–219, 2015.

[142] Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806, 1993.

[143] James Martens and Ilya Sutskever. Learning recurrent neural networks with Hessian-free optimization. *Icml*, pages 1033–1040, 2011.

[144] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9:249–256, 2010.

[145] Davide Spalla, Alexis Dubreuil, Sophie Rosay, Remi Monasson, and Alessandro Treves. Can grid cell ensembles represent multiple spaces? *bioRxiv*, pages 1–10, 2019.

[146] Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, May-Britt Moser, and Edvard I. Moser. Conjunctive Representation of Position, Direction, and Velocity in Entorhinal Cortex. *Science*, 312(May):758, 2006.

[147] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. Towards Biologically Plausible Deep Learning. *Assessment and Evaluation in Higher Education*, 32(4):457–474, 2007.

[148] Asim Roy, Leonid Perlovsky, Tarek R. Besold, Juyang Weng, and Jonathan C. W. Edwards. Editorial: Representation in the Brain. *Frontiers in Psychology*, 9(August):8–10, 2018.

[149] Ingmar Kanitscheider and Ila Fiete. Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. (Nips), 2016.

[150] Sang Wan Lee and Ben Seymour. Decision-making in brains and robots — the case for an interdisciplinary approach. *Current Opinion in Behavioral Sciences*, 26:137–145, 2019.

[151] Alexandra G Rosati. Foraging Cognition : Reviving the Ecological Intelligence Hypothesis. *Trends in Cognitive Sciences*, xx:1–12, 2017.

[152] Ralph Adolphs. Social Cognition and the human brain. *Trends in Cognitive Sciences: Human social cognition*, 3(12):469–479, 1999.

[153] Andrew T. D. Bennett. Do animals have cognitive maps? *The Journal of Experimental Biology*, 199(199):219–224, 1996.

[154] Philippe Gaussier, Jean Paul Banquet, Nicolas Cuperlier, Mathias Quoy, Lise Aubin, Pierre-Yves Jacob, Francesca Sargolini, Etienne Save, Jeffrey L. Krichmar, and Bruno Poucet. Merging information in the entorhinal cortex: what can we learn from robotics experiments and modeling? *The Journal of Experimental Biology*, 222(Suppl 1):jeb186932, 2019.

[155] N. J. Mackintosh. Do not ask whether they have a cognitive map, but how they find their way about. *Psicológica*, 23:165–185, 2002.

[156] Kathryn J. Jeffery. Neural Odometry: The Discrete Charm of the Entorhinal Cortex. *Current Biology*, 23(5):R204–R206, 2013.

[157] Upinder S. Bhalla. Dendrites, deep learning, and sequences in the hippocampus. *Hippocampus*, 29(3):239–251, 2019.

[158] Barbara Tversky. Cognitive maps, cognitive collages, and spatial mental models. In Andrew U. Frank and Irene Campari, editors, *Progress and New Trends in 3D Geoinformation Sciences Lecture Notes in Geoinformation and Cartography*, volume 716 of *Lecture Notes in Computer Science*, pages 14–24. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[159] Stea D Downs RM. Cognitive Maps and Spatial Behavior: Process and Products. In *The Map Reader: Theories of Mapping Practice and Cartographic Representation*, chapter 4.3, pages 312–317. 2011.

[160] Edvard I Moser and May-britt Moser. Grid Cells and Neural Coding in High-End Cortices. *Neuron*, 80(3):765–774, 2013.

[161] John O'Keefe. An allocentric spatial model for the hippocampal cognitive map. *Hippocampus*, 1(3):230–235, jul 1991.

[162] Tamas Madl, Ke Chen, Daniela Montaldi, and Robert Trappl. Computational cognitive models of spatial memory in navigation space: A review. *Neural Networks*, 65:18–43, 2015.

[163] Fabian Chersi and Neil Burgess. The Cognitive Architecture of Spatial Navigation: Hippocampal and Striatal Contributions. *Neuron*, 88(1):64–77, 2015.

[164] Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated. *Trends in Cognitive Sciences*, 20(7):512–534, 2016.

[165] Kiyohito Iigaya and John P. O'Doherty. Hippocampus Is What Happens while You're Busy Making Other Plans. *Neuron*, 102(3):517–519, 2019.

[166] Dean Mobbs, Pete C Trimmer, Daniel T Blumstein, and Peter Dayan. Foraging for foundations in decision neuroscience: insights from ethology. *Nature Reviews Neuroscience*, 2018.

[167] P Read. Montague, Peter Dayan, and Terrence J. Sejnowski. Foraging in an Uncertain Environment Using Predictive Hebbian Learning. pages 598–605, 1993.

[168] Thomas Wolbers and Mary Hegarty. What determines our navigational abilities? *Trends in Cognitive Sciences*, 14(3):138–146, 2010.

[169] Zé Henrique T.D. Góis and Adriano B.L. Tort. Characterizing Speed Cells in the Rat Hippocampus. *Cell Reports*, 25(7):1872–1884.e4, 2018.

[170] Edvard I. Moser, Emilio Kropff, and May-Britt Moser. Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience*, 31(1):69–89, 2008.

[171] Marius M. Stanciu. The Explanatory Gap: 30 Years after. *Procedia - Social and Behavioral Sciences*, 127:292–296, 2014.

[172] Shachar Maidenbaum, Jonathan Miller, Joel M. Stein, and Joshua Jacobs. Grid-like hexadirectional modulation of human entorhinal theta oscillations. *Proceedings of the National Academy of Sciences*, 115(42):10798–10803, 2018.

[173] Arne D. Ekstrom and Charan Ranganath. Space, Time and Episodic Memory: the Hippocampus is all over the Cognitive Map. *Hippocampus*, 28(9):680–687, 2018.

[174] Cian O'Donnell and Terrence J. Sejnowski. Street View of the Cognitive Map. *Cell*, 164(1-2):13–15, 2016.

[175] Julija Krupic, Marius Bauza, Stephen Burton, and John O'Keefe. Local transformations of the hippocampal cognitive map. *Science*, 359(6380):1143–1146, 2018.

[176] Matthew G. Buckley, Alastair D. Smith, and Mark Haselgrove. Thinking outside of the box II: Disrupting the cognitive map. *Cognitive Psychology*, 108(December 2018):22–41, 2019.

[177] Ian C. Ballard, Anthony D. Wagner, and Samuel M. McClure. Hippocampal pattern separation supports reinforcement learning. *Nature Communications*, 10(1), 2019.

[178] Silvy HP Collin, Branka Milivojevic, and Christian F. Doeller. Hippocampal hierarchical networks for space, time, and memory. *Current Opinion in Behavioral Sciences*, 17:71–76, 2017.

[179] Nathaniel J Killian and Elizabeth A Buffalo. Grid cells map the visual world. 21(February):161–162, 2018.

[180] Honi Sanders, Cesar Renno-Costa, Marco Idiart, and John Lisman. Grid Cells and Place Cells: An Integrated View of their Navigational and Memory Function. 38(12):763–775, 2016.

[181] Salman E Qasim, Jonathan Miller, Cory S Inman, Robert E Gross, Jon T Willie, Bradley Lega, Jui-Jui Lin, Ashwini Sharan, Chengyuan Wu, Michael R Sperling, Sameer Sheth, Guy M McKhann, Elliot H Smith, Catherine Schevon, Joel Stein, and Joshua Jacobs. Single neurons in the human entorhinal cortex remap to distinguish individual spatial memories. *bioRxiv*, 2018.

[182] Marcus K Benna and Stefano Fusi. Are place cells just memory cells? Memory compression leads to spatial tuning and history dependence. *bioRxiv*, page 624239, 2019.

[183] Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, 2017.

[184] Rodrigo Quian Quiroga. Concept cells: the building blocks of declarative memory functions. *Nature Reviews Neuroscience*, 13(8):587–597, aug 2012.

[185] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):1–23, 1958.

[186] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[187] Kunihiko Fukushima. Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. *Neural Networks*, 1:119–130, 1988.

[188] Michael Freitag, Shahin Amiriparian, Sergey Pugachevskiy, Nicholas Cummins, and Björn Schuller. auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks. *The Journal of Machine Learning Research*, 18(1):6340–6344, 2017.

[189] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cuDNN: Efficient Primitives for Deep Learning. pages 1–9, oct 2014.

[190] Daniel L.K. Yamins and James J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016.

[191] Jon H. Kaas. The evolution of brains from early mammals to humans. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(1):33–45, 2013.

[192] Gerhard Roth and Ursula Dicke. Evolution of the brain and intelligence in primates. *Progress in Brain Research*, 195(5):413–430, 2012.

[193] David D. Franks. Chapter 2: Evolution of the Human Brain. *Neurosociology: The Nexus Between Neuroscience and Social Psychology*, pages 1–216, 2010.

[194] Robert K. Naumann, Janie M. Ondracek, Samuel Reiter, Mark Shein-Idelson, Maria Antonietta Tosches, Tracy M. Yamawaki, and Gilles Laurent. The reptilian brain. *Current Biology*, 25(8):R317–R321, 2015.

[195] Kenji Doya and Tadahiro Taniguchi. Toward evolutionary and developmental intelligence. *Current Opinion in Behavioral Sciences*, 29(Box 1):91–96, 2019.

[196] Rodney Brooks, Demis Hassabis, Dennis Bray, and Amnon Shashua. Is the brain a good model for machine intelligence. *Nature*, 462(482.7386):462–463, 2012.

[197] Stephen Grossberg. *A Half Century of Progress Toward a Unified Neural Theory of Mind and Brain With Applications to Autonomous Adaptive Agents and Mental Disorders*. Elsevier Inc., 2019.

[198] Uri Hasson and Howard C. Nusbaum. Emerging Opportunities for Advancing Cognitive Neuroscience. *Trends in Cognitive Sciences*, 23(5):363–365, 2019.

[199] Joshua I. Glaser, Ari S. Benjamin, Roozbeh Farhoodi, and Konrad P. Kording. The roles of supervised machine learning in systems neuroscience. *Progress in Neurobiology*, 175:126–137, 2019.

[200] Radoslaw M. Cichy and Daniel Kaiser. Deep Neural Networks as Scientific Models. *Trends in Cognitive Sciences*, 23(4):305–317, 2019.

[201] Morgan R Frank, Dashun Wang, Manuel Cebrian, and Iyad Rahwan. The Evolution of Citation Graphs in Artificial Intelligence Research. *Nature Machine Intelligence*, 1(February):79–85, 2019.

[202] M. Yuan, B. Tian, V.A. Shim, H. Tang, and H. Li. An entorhinal-hippocampal model for simultaneous cognitive map building. *Proceedings of the National Conference on Artificial Intelligence*, 1:586–592, 2015.

[203] Callie Federer and Joel Zylberberg. A Self-organizing memory network. pages 1–9, 2018.

[204] Joseph D. Monaco, Grace M. Hwang, Kevin M. Schultz, and Kechen Zhang. Cognitive swarming: an approach from the theoretical neuroscience of hippocampal function. *Micro- and Nanotechnology Sensors, Systems, and Applications XI*, (May):84, 2019.

[205] William B. Kristan. Early evolution of neurons. *Current Biology*, 26(20):R949–R954, 2016.

[206] O Shoval, H Sheftel, G Shinar, Y Hart, O Ramote, A Mayo, E Dekel, K Kavanagh, and U Alon. Evolutionary Trade-Offs, Pareto Optimality, and the Geometry of Phenotype Space. *Science*, 336(6085):1157–1160, 2012.

[207] T V Chernigovskaya. Evolutionary Physiology: History , Principles. 118(1):63–79, 1997.

[208] Jaime C. Confer, Judith A. Easton, Diana S. Fleischman, Cari D. Goetz, David M.G. Lewis, Carin Perilloux, and David M. Buss. Evolutionary Psychology: Controversies, Questions, Prospects, and Limitations. *American Psychologist*, 65(2):110–126, 2010.

[209] R.D. Fernald. Evolution of Vertebrate Eyes. *The Senses: A Comprehensive Reference*, (April):9–23, 2008.

[210] Barret Zoph and Quoc V Le. Neural Architecture Search with Reinforcement Learning. pages 1–16, 2017.

[211] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical Representations for Efficient Architecture Search. pages 1–13, 2018.

[212] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. 2017.

[213] Frederic Kaplan and Pierre-yves Oudeyer. In search of the neural circuits of intrinsic motivation. 1(1):225–236, 2007.

[214] Adrien F. Baranes, Pierre Yves Oudeyer, and Jacqueline Gottlieb. The effects of task difficulty, novelty and the size of the search space on intrinsically motivated exploration. *Frontiers in Neuroscience*, 8:1–9, 2014.

[215] Peter Dayan and Bernard W Balleine. Reward, Motivation, and Reinforcement Learning. 36:285–298, 2002.

[216] A. H. Maslow. A theory of Human Motivation. (13):370–396, 1943.

[217] D. E. Berlyne. Curiosity and Exploration. 251:25–34, 1966.

[218] Pierre-Yves Oudeyer. Computational Theories of Curiosity-Driven Learning. 2018.

[219] Pierre-Yves Oudeyer and Linda Smith. How Evolution May Work Through Curiosity-Driven Developmental Process. *Topics in Cognitive Science*, 8(2):492–502, 2016.

[220] Adrien Laversanne-finot, Alexandre Péré, and Pierre-yves Oudeyer. Curiosity Driven Exploration of Learned Disentangled Goal Spaces. (CoRL), 2018.

[221] Sebastian Bader and Pascal Hitzler. Dimensions of Neural-symbolic Integration - A Structured Survey. 2005.

[222] Jacqueline Gottlieb, Manuel Lopes, and Pierre-Yves Oudeyer. Motivated Cognition: Neural and Computational Mechanisms of Curiosity, Attention, and Intrinsic Motivation. *Recent Developments in Neuroscience Research on Human Motivation*, pages 149–172, 2016.

[223] Jacqueline Gottlieb, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes. Information seeking, curiosity and attention: computational and neural mechanisms. 1(3):233–245, 2012.

[224] John E Laird. Extending the Soar Cognitive Architecture. 2008.

[225] Etienne Koechlin, Crystele Ody, and Frederique Kouneiher. The architecture of cognitive control in the human prefrontal cortex. *Science*, 302(November):1181–1186, 2003.

[226] Eveline A. Crone and Nikolaus Steinbeis. Neural Perspectives on Cognitive Control Development during Childhood and Adolescence. *Trends in Cognitive Sciences*, 21(3):205–215, 2017.

[227] James A Reggia, Di-wei Huang, and Garrett Katz. Exploring the Computational Explanatory Gap. *Philosophies*, 2(1):1–20, 2017.

[228] Gail A. Carpenter. Adaptive Resonance Theory. 2009.

[229] Wesley Clawson, Ana F. Vicente, Maëva Ferraris, Christophe Bernard, Demian Battaglia, and Pascale P. Quilichini. Computing hubs in the hippocampus and cortex. *Science Advances*, 5(6):eaax4843, 2019.

[230] Denis Buehler. The central executive system. *Synthese*, 195(5):1969–1991, 2018.

[231] Lars Nyberg. Cognitive control in the prefrontal cortex: A central or distributed executive? *Scandinavian Journal of Psychology*, 59(1):62–65, 2018.

[232] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines. pages 1–26, 2014.

[233] Benjamin Sinclair, Narelle K. Hansell, Gabriëlla A.M. Blokland, Nicholas G. Martin, Paul M. Thompson, Michael Breakspear, Greig I. de Zubicaray, Margaret J. Wright, and Katie L. McMahon. Heritability of the network architecture of intrinsic brain functional connectivity. *NeuroImage*, 121:243–252, 2015.

[234] Sophia Mueller, Danhong Wang, Michael D. Fox, B. T.Thomas Yeo, Jorge Sepulcre, Mert R. Sabuncu, Rebecca Shafee, Jie Lu, and Hesheng Liu. Individual Variability in Functional Connectivity Architecture of the Human Brain. *Neuron*, 77(3):586–595, 2013.

[235] Vinod Menon. Developmental pathways to functional brain networks: Emerging principles. *Trends in Cognitive Sciences*, 17(12):627–640, 2013.

[236] Olaf Sporns, Dante R Chialvo, Marcus Kaiser, and Claus C Hilgetag. Organization, development and function of complex brain networks. 8(9), 2004.

[237] M. P. van den Heuvel and O Sporns. Network hubs in the human brain. *Trends in cognitive sciences*, 17(12):683–96, 2013.

[238] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: A structural description of the human brain. *PLoS Computational Biology*, 1(4):0245–0251, 2005.

[239] Helen Barbas. General Cortical and Special Prefrontal Connections: Principles from Structure to Function. *Annual Review of Neuroscience*, 38(1):269–289, 2015.

[240] Wolpert D.M., Miall R.C., and Kawato M. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998.

[241] K Doya. What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? 12:961–974, 1999.

[242] Oliver Baumann, Ronald J. Borra, James M. Bower, Kathleen E. Cullen, Christophe Habas, Richard B. Ivry, Maria Leggio, Jason B. Mattingley, Marco Molinari, Eric A. Moulton, Michael G. Paulin, Marina A. Pavlova, Jeremy D. Schmahmann, and Arseny A. Sokolov. Consensus Paper: The Role of the Cerebellum in Perceptual Processes. *Cerebellum*, 14(2):197–220, 2015.

[243] Kenji Doya. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, pages 732–739, 2000.

[244] Jean Piaget. Cognitive Development in Children: Piaget Development and Learning. *Journal of research in science teaching*, 2(3):176–186, 1964.

[245] Anna Bullock Drummey and Judith G Wiley. The Development of Spatial Location Coding: Place Learning and Dead Reckoning in the second and third years. 200(1998):185–200, 2014.

[246] Iroise Dumontheil. Development of the social brain during adolescence. *Psicologia Educativa*, 21(2):117–124, 2015.

[247] Michael Tomasello, Brian Hare, Hagen Lehmann, and Josep Call. Reliance on head versus eyes in the gaze following of great apes and human infants: the cooperative eye hypothesis. 52:314–320, 2007.

[248] Mary Helen Immordino-yang and Antonio Damasio. We Feel, Therefore We Learn: The Relevance of Affective and Social Neuroscience to Education. 1(1):3–10, 2007.

[249] Josep Call and Michael Tomasello. Does the chimpanzee have a theory of mind ? 30 years later. pages 187–192, 2008.

[250] Kevin N. Laland. Social learning strategies. *Animal Learning & Behavior*, 32(1):4–14, 2004.

[251] Kristen A. Lindquist and Lisa Feldman Barrett. A functional architecture of the human brain: Emerging insights from the science of emotion. *Trends in Cognitive Sciences*, 16(11):533–540, 2012.

[252] Maiken Nedergaard, Bruce Ransom, and Steven A. Goldman. New roles for astrocytes: Redefining the functional architecture of the brain. *Trends in Neurosciences*, 26(10):523–530, 2003.

[253] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059–1069, 2010.

[254] Richard E. Passingham, Klaas E. Stephan, and Rolf Kötter. The anatomical basis of functional localization in the cortex. *Nature Reviews Neuroscience*, 3(8):606–616, 2002.

[255] Vida Demarin, Sandra Morović, and Raphael Béné. Neuroplasticity. 116(2):209–211, 2014.

[256] Robin Holliday. Epigenetics: A historical overview. *Epigenetics*, 1(2):76–80, 2006.

[257] Javier Defelipe and Lidia Alonso-nanclares. The Synapse: Differences Between Men and Women. (May 2016):43–57, 2013.

[258] Alan Baddeley. Working memory. 20(4):136–140, 1992.

[259] Bernard J. Baars and Stan Franklin. How consciousness experience and working memory interact. *Trends in Cognitive Sciences*, 7(4):166–172, 2003.

[260] Edward Awh and John Jonides. Overlapping mechanisms of attention and spatial working memory. *Trends in Cognitive Sciences*, 5(3):119–126, 2001.

[261] Laura Lee Colgin, Edvard I. Moser, and May Britt Moser. Understanding memory through hippocampal remapping. *Trends in Neurosciences*, 31(9):469–477, 2008.

[262] Steven Kapturowski, Georg Ostrovski, Will Dabney, John Quan, and Remi Munos. Recurrent Experience Replay in Distributed Reinforcement Learning. *International Conference on Learning Representation*, pages 1–15, 2019.

[263] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic Curiosity through Reachability. pages 1–21, 2018.

[264] Ida Momennejad, A. Ross Otto, Nathaniel D. Daw, and Kenneth A. Norman. Offline replay supports planning in human reinforcement learning. *eLife*, 7:1–25, 2018.

[265] A Saez, M Rigotti, S Ostojic, S Fusi, and C D Salzman. Abstract Context Representations in Primate Amygdala and Prefrontal Cortex. *Neuron*, 87(4):869–881, 2015.

[266] W. R. Stauffer. The biological and behavioral computations that influence dopamine responses. *Current Opinion in Neurobiology*, 49:123–131, 2018.

[267] Matthew P.H. Gardner, Geoffrey Schoenbaum, and Samuel J. Gershman. Rethinking dopamine as generalized prediction error. *Proceedings of the Royal Society B: Biological Sciences*, 285(1891), 2018.

[268] J Baars Bernard. Global workspace theory of consciousness: toward a cognitive neuroscience of human experience? 2005.

[269] Istituto Cibernetica and Arco Felice. Neuronal Bases and Psychological Aspects of Consciousness. 8(October):13–18, 1997.

[270] John A Bargh and Ezequiel Morsella. The Unconscious Mind. 3(1):73–79, 2008.

[271] Reginald G Golledge. Spatial Cognition, Cognitive Mapping, and Cognitive Maps. *Spatial behavior : a geographic perspective*, pages 224–266, 1997.

[272] Steven L Bressler. Understanding Cognition through large-scale cortical networks. *Current directions in psychological science*, 2002.

[273] Steven L Bressler and Vinod Menon. Large-scale brain networks in cognition: emerging methods and principles. *Trends in Cognitive Sciences*, 14(6):277–290, 2010.

[274] Howard Eichenbaum. Barlow versus Hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? *Neuroscience Letters*, 680:88–93, 2018.

[275] Rui Ponte Costa, Yannis M. Assael, Brendan Shillingford, Nando de Freitas, and Tim P. Vogels. Cortical microcircuits as gated-recurrent neural networks. (Nips 2017), 2017.

[276] Margaret L. Schlichting and Alison R. Preston. Memory integration: Neural mechanisms and implications for behavior. *Current Opinion in Behavioral Sciences*, 1:1–8, 2015.

[277] Anil Seth. The strength of weak artificial consciousness. *International Journal of Machine Consciousness*, 1(1):71–82, 2009.

[278] Robert H Wortham and Joanna J Bryson. A role for action selection in consciousness. *CEUR Workshop Proceedings*, 1855:25–30, 2016.

[279] Giorgio Buttazzo. Artificial consciousness: Utopia or Real Possibility? *Computer*, 34(7):24–30, 2001.

[280] James A Reggia. Conscious Machines: The AI Perspective. *AAAI Fall Symposium Series*, pages 34–37, 2014.

[281] H. Francis Song, Guangyu R. Yang, and Xiao Jing Wang. Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks: A Simple and Flexible Framework. *PLoS Computational Biology*, 12(2):1–30, 2016.

[282] Stefano Fusi, Earl K. Miller, and Mattia Rigotti. Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74, 2016.

[283] Ron Chrisley and Aaron Sloman. Architectural Requirements for Consciousness. *CEUR Workshop Proceedings*, 1855:31–36, 2016.

[284] Takuma Okawa and Junichi Takeno. Development of Self-cognition through Imitation Behavior. *Procedia Computer Science*, 88:46–51, 2016.

[285] J. Broekens and D. DeGroot. Emergent representations and reasoning in adaptive agents. (May 2014):207–214, 2005.

[286] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015.

[287] Cao Chunshui, Huang Yongzhen, Wang Zilei, Wang Liang, Xu Ninglong, and Tan Tieniu. Lateral Inhibition-Inspired Convolutional Neural Network for Visual Attention and Saliency Detection. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 6690–6697, 2018.

[288] Alexander JE Kell and Josh H. McDermott. Deep neural network models of sensory systems: windows onto the role of task constraints. *Current Opinion in Neurobiology*, 55:121–132, 2019.

[289] Ana F. Almeida, Rui Figueiredo, Alexandre Bernardino, and Jose Santos-Victor. Deep Networks for Human Visual Attention: A hybrid model using Foveal Vision. *Iberian Robotics conference*, pages 117–128, 2017.

[290] M. N. Hebart and G. Hesselmann. What Visual Information Is Processed in the Human Dorsal Stream? *Journal of Neuroscience*, 32(24):8107–8109, 2012.

[291] D H Hubel and T N Wiesel. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. pages 106–154, 1962.

[292] Zhaoping Li. A saliency map in primary visual cortex. *Trends in Cognitive Sciences*, 6(1):9–16, 2002.

[293] Qi Li and Haiyan Geng. Progress in cognitive neuroscientific studies of visual awareness. *Progress in Natural Science*, 19(2):145–152, 2009.

[294] Qianli Liao and Tomaso Poggio. Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex. (047):1–16, 2016.

[295] Frank Tong. Primary visual cortex and visual awareness. *Nature Reviews Neuroscience*, 4(3):219–229, 2003.

[296] Ralph Linsker. From basic network principles to neural architecture: Emergence of orientation columns. 83(3):8779–8783, 1986.

[297] Yesmina Jaafra, Jean Luc Laurent, Aline Deruyver, and Mohamed Saber Naceur. A Review of Meta-Reinforcement Learning for Deep Neural Networks Architecture Search. pages 1–29, 2018.

[298] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018, 2018.

[299] Daniel S. Levine. *Theory of the Brain and Mind.* Elsevier Inc., 2019.

[300] Bernard Widrow, Youngsik Kim, Dookun Park, and Jose Krause Perin. *Nature's Learning Rule: The Hebbian-LMS Algorithm.* Elsevier Inc., 2019.

[301] Kent C. Berridge and Terry E. Robinson. Parsing reward. *Trends in Neurosciences*, 26(9):507–513, 2003.

[302] D. Gowanlock R. Tervo, Joshua B. Tenenbaum, and Samuel J. Gershman. Toward the neural implementation of structure learning. *Current Opinion in Neurobiology*, 37:99–105, 2016.

[303] Sergey Bartunov, Adam Santoro, Blake A. Richards, Geoffrey E. Hinton, and Timothy P. Lillicrap. Assessing the Scalability of Biologically-Motivated Deep Learning Algorithms and Architectures. *Polskie Archiwum Medycyny Wewnetrznej*, 126(1-2):100–101, 2016.

[304] Jeffrey L Krichmar, Florian Röhrbein, Informatics Vi, and Technische Universität München. Value and reward based learning in neurorobots. 7(September):1–2, 2013.

[305] Rodney Brooks. New Approaches to Robotics. *Science*, 253(5025):1227–1232, 1991.

[306] Levent Bayindir. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016.

[307] Sandra Devin, Michelangelo Fiore, Aurerile Clodic, and Rachid Alami. Some essential skills and their combination in an architecture for a cognitive and interactive robot. 2016.

[308] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[309] Guglielmo Tamburrini. Robot Ethics: A View from the Philosophy of Science. *Ethics and Robotics*, (May), 2009.

[310] Niv Yael. Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154, 2009.

[311] John P. O'Doherty, Sang Wan Lee, and Daniel McNamee. The structure of reinforcement-learning mechanisms in the human brain. *Current Opinion in Behavioral Sciences*, 1:94–100, 2015.

[312] Nathaniel D. Daw and Kenji Doya. The computational neurobiology of learning and reward. *Current Opinion in Neurobiology*, 16(2):199–204, 2006.

[313] Sonia J. Bishop and Christopher Gagne. Anxiety, Depression, and Decision Making: A Computational Perspective. *Annual Review of Neuroscience*, 41(1):371–388, 2018.

[314] Bradley B. Doll, Dylan A. Simon, and Nathaniel D. Daw. The ubiquity of model-based reinforcement learning. 22(6):1075–1081, 2013.

[315] Nils Kolling and Thomas Akam. (Reinforcement ?) Learning to forage optimally. *Current Opinion in Neurobiology*, 46:162–169, 2017.

[316] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Z Joel, Demis Hassabis, Matthew Botvinick, Kings Cross, Joel Z Leibo, Demis Hassabis, Matthew Botvinick, Matthew Botvinick Deepmind, and Pancras Square. Prefrontal Cortex As a Meta-Reinforcement Learning System. 2018.

[317] Rahul Ramesh, Manan Tomar, and Balaraman Ravindran. Successor Options: An Option Discovery Framework for Reinforcement Learning. 2019.

[318] Tejas D. Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J. Gershman. Deep Successor Reinforcement Learning. 2016.

[319] Samuel J. Gershman. The Successor Representation: Its Computational Logic and Neural Substrates. *The Journal of Neuroscience*, 38(33):7193–7200, 2018.

[320] Nathaniel D. Daw. Are we of two minds? *Nature Neuroscience*, 21(11):1497–1499, 2018.

[321] Eatures For. Universal Successor Features For Transfer Reinforcement Learning. 2019.

[322] Ida Momennejad and Marc W. Howard. Predicting the Future with Multi-scale Successor Representations. *bioRxiv*, page 449470, 2018.

[323] Tamas J. Madarasz. Inferred successor maps for better transfer learning. pages 1–19, 2019.

[324] Ricky Loynd, Matthew Hausknecht, Lihong Li, and Li Deng. Now I Remember! Episodic Memory for Reinforcement Learning. (1998):1–12, 2018.

[325] Charles Blundell, Alexander Pritzel, Jack Rae, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-Free Episodic Control. pages 1–12, 2016.

[326] Emilio Parisotto and Ruslan Salakhutdinov. Neural Map: Structured Memory for Deep Reinforcement Learning. pages 1–13, 2017.

[327] Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z. Leibo, Adam Santoro, Mevlana Gemici, Malcolm Reynolds, Tim Harley, Josh Abramson, Shakir Mohamed, Danilo Rezende, David Saxton, Adam Cain, Chloe Hillier, David Silver, Koray Kavukcuoglu, Matt Botvinick, Demis Hassabis, and Timothy Lillicrap. Unsupervised Predictive Memory in a Goal-Directed Agent. 2018.

[328] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature Publishing Group*, 538(7626):471–476, 2016.

[329] Joshua Achiam and Shankar Sastry. Surprise-based Intrinsic Motivation for Deep Reinforcement Learning. pages 1–13, 2017.

[330] Nick Haber, Damian Mrowca, Li Fei-Fei, and Daniel L. K. Yamins. Emergence of Structured Behaviors from Curiosity-Based Intrinsic Motivation. 2018.

[331] Marc G Bellemare, Tom Schaul, David Saxton, and Georg Ostrovski. Unifying Count-Based Exploration and Intrinsic Motivation. (Nips), 2016.

[332] Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in Model-based Reinforcement Learning by Empirically Estimating Learning Progress. *Advances in neural information processing systems.*, pages 206–214, 2012.

[333] Oleksii Zhelo, Jingwei Zhang, Lei Tai, Ming Liu, and Wolfram Burgard. Curiosity-driven Exploration for Mapless Navigation with Deep Reinforcement Learning. (Icm):1–5, apr 2018.

[334] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-Scale Study of Curiosity-Driven Learning. 2018.

[335] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven Exploration by Self-supervised Prediction. 2017.

[336] Yuhang Song, Jianyi Wang, Thomas Lukasiewicz, Zhenghua Xu, Shangtong Zhang, and Mai Xu. Mega-Reward: Achieving Human-Level Play without Extrinsic Rewards. may 2019.

[337] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is All You need Learning Skills without a Reward Function. pages 1–22, 2019.

[338] Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. *Emotion in reinforcement learning agents and robots: a survey*, volume 107. Springer US, 2018.

[339] Michael Alvarino. A Review of Hierarchical Reinforcement Learning. 21(1):103–105, 2017.

[340] Tejas D. Kulkarni, Karthik R. Narasimhan, Ardavan Saeedi, and Joshua B. Tenenbaum. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. (Nips), 2016.

[341] Matthew Michael Botvinick. Hierarchical reinforcement learning and decision making. *Current Opinion in Neurobiology*, 22(6):956–962, 2012.

[342] Wenhao Ding, Shuaijun Li, Huihuan Qian, and Yongquan Chen. Hierarchical Reinforcement Learning Framework Towards Multi-Agent Navigation. *2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018*, pages 237–242, 2019.

[343] Nachum Ofir, Shixiang Gu, Honglak Lee, and Sergey Levine. Near-Optimal Representation Learning for Hierarchical Reinforcement Learning. *ICRL 2019*, 2019.

[344] Stefan Elfwing. Biologically Inspired Embodied Evolution of Survival. pages 1–7.

[345] Tim Salimans and Xi Chen. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. pages 1–13, 2016.

[346] Hado Van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

[347] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling Network Architectures for Deep Reinforcement Learning. (9), 2015.

[348] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A Brief Survey of Deep Reinforcement Learning. *IEEE SIGNAL PROCESSING MAGAZINE*, pages 1–16, 2017.

[349] Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Rezende, Adria P. Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter Battaglia, Demis Hassabis, David Silver, and Daan Wierstra. Imagination-Augmented Agents for Deep Reinforcement Learning. *Imagination-augmented agents for deep reinforcement learning*, 2018.

[350] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards Deep Symbolic Reinforcement Learning. pages 1–13, 2016.

[351] Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement Learning, Fast and Slow. *Trends in Cognitive Sciences*, 23(5):408–422, 2019.

[352] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical Learning Periods in Deep Learning. pages 1–14, 2019.

[353] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *Advances in neural information processing systems.*, 2016.

[354] Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-Shot Imitation Learning. (Nips), 2017.

[355] Sumaiya Farzana G and Angelina Geetha. Evolution and Prospects of Deep Learning in Current Research Scenarios. *International Journal of Innovations & Advancement in Computer Science*, 6(8):404–411, 2017.

[356] Wolfgang Maass. Networks of Spiking Neurons the third generation of Neural Network Models. page 7, 1997.

[357] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, Lawrence Carin, Nokia Bell Labs, and Murray Hill. Variational Autoencoder for Deep Learning of Images, Labels and Captions. (Nips), 2016.

[358] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. (Nips), 2017.

[359] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representational Learning with Deep Convolutional Generative Adversarial Networks. pages 1–15, 2016.

[360] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pages 2672–2680, 2014.

[361] Timothy J. Draelos, Nadine E. Miner, Christopher C. Lamb, Jonathan A. Cox, Craig M. Vineyard, Kristofor D. Carlson, William M. Severa, Conrad D. James, and James B. Aimone. Neurogenesis Deep Learning. 2017.

[362] Neel Kant. Recent Advances in Neural Program Synthesis. 2018.

[363] Ronan Collobert and Jason Weston. A unified architecture for natural language processing. *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 160–167, 2008.

[364] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very Deep Convolutional Networks for Text Classification. 2016.

[365] Richard Socher, Cliff Chiung-yu Lin, Andrew Y Ng, and Christopher D Manning. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. *Neural Networks*, 10(January 2011):203–226, 2011.

[366] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. pages 1–9, 2015.

[367] Chrisantha Fernando, Dylan Banarse, Malcolm Reynolds, Frederic Besse, David Pfau, Max Jaderberg, Marc Lanctot, and Daan Wierstra. Convolution by Evolution: Differentiable Pattern Producing Networks. 2016.

[368] Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. Early Visual Concept Learning with Unsupervised Deep Learning. 2016.

[369] Honglak Lee and Andrew Y Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. pages 609–616, 2009.

[370] P J Angeline, G B Saunders, and J B Pollack. An Evolutionary Algorithm that Evolves Recurrent Neural Networks. *IEEE Transactions on Neural Networks*, 5(2):54–65, 1993.

[371] Emmanuel Dufourq and Bruce A. Bassett. EDEN: Evolutionary deep networks for efficient machine learning. *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, PRASA-RobMech 2017*, 2018-Janua:110–115, 2018.

[372] Mohammad Javad Shafiee, Akshaya Mishra, and Alexander Wong. Deep Learning with Darwin: Evolutionary Synthesis of Deep Neural Networks. *Neural Processing Letters*, 48(1):603–613, 2018.

[373] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, and Babak Hodjat. *Chapter 15 - Evolving Deep Neural Networks*. Elsevier Inc., 2019.

[374] Rasmus Boll Greve, Emil Juul Jacobsen, and Sebastian Risi. Evolving Neural Turing Machines. pages 1–4, 2015.

[375] Sergio Peignier. Leaky Echo State Network. pages 1–3, 2017.

[376] Claudio Gallicchio and Alessio Micheli. Deep Echo State Network (DeepESN): A Brief Survey. pages 1–12, 2018.

[377] Claudio Gallicchio and Luca Pedrelli. Deep Reservoir Computing: A Critical Experimental Analysis. *Neurocomputing*, 268:87–99, 2017.

[378] David Snyder, Alireza Goudarzi, and Christof Teuscher. Computational Capabilities of Random Automata Networks for Reservoir Computing. 2:1–9, 2013.

[379] Grzegorz M. Wojcik and Wieslaw A. Kaminski. Liquid state machine built of Hodgkin-Huxley neurons and pattern recognition. *Neurocomputing*, 58-60:245–251, 2004.

[380] Tadashi Yamazaki and Shigeru Tanaka. The cerebellum as a liquid state machine. *Neural Networks*, 20(3):290–297, 2007.

[381] Ruslan Salakhutdinov, Joshua B Tenenbaum, and Antonio Torralba. Learning with Hierarchical-Deep Models. 35(8):1958–1971, 2013.

[382] Y Du, W Wang, and L Wang. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015.

[383] Constantine Dovrolis. A neuro-inspired architecture for unsupervised continual learning based on online clustering and hierarchical predictive coding. 2018.

[384] Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. pages 1–12, oct 2016.

[385] Fabio Anselmi, Joel Z. Leibo, Lorenzo Rosasco, Jim Mutch, Andrea Tacchetti, and Tomaso Poggio. Unsupervised learning of invariant representations with low sample complexity. *Theoretical Computer Science*, 633:112–121, 2016.

[386] Baolin Peng, Zhengdong Lu, Hang Li, and Kam-Fai Wong. Towards Neural Network-based Reasoning. pages 1–12, 2015.

[387] Adam Santoro, David Raposo, David G T Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, pages 4967–4976, 2017.

[388] David G T Barrett, Felix Hill, Adam Santoro, Ari S Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. 2001.

[389] Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew Mccallum. Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks. 2013.

[390] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal Reasoning from Meta-reinforcement Learning. 2019.

[391] Herbert Jaeger. Deep Neural Reasoning. *Nature*, 538:467–468, 2016.

[392] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. Relational Deep Reinforcement Learning. (2):1–15, 2018.

[393] Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. Relational recurrent neural networks. 2018.

[394] Stanley Kok and Pedro Domingos. Learning the structure of Markov logic networks. *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 441–448, 2005.

[395] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. Relational Deep Reinforcement Learning. (2):1–15, 2018.

[396] Andrew Trask, Felix Hill, Scott Reed, Jack Rae, Chris Dyer, and Phil Blunsom. Neural Arithmetic Logic Units. 2018.

[397] Wojciech Zaremba, Tomas Mikolov, Armand Joulin, and Rob Fergus. Learning Simple Algorithms from Examples. pages 1–12, 2016.

[398] Kevin W Mickey and James L McClelland. A neural network model of learning mathematical equivalence. *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, (1012-1017):1012–1017, 2014.

[399] Pedro Domingos, Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla. Unifying Logical and Statistical AI. *AAAI*, pages 2–7, 2006.

[400] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences from Natural Supervision. *ICLR*, pages 1–28, 2019.

[401] Marta Garnelo and Murray Shanahan. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, 29:17–23, 2019.

[402] Stuart Russel, Daniel Dewey, and Max Tegmark. Research Priorities for Robust and Beneficial Artificial Intelligence. *A Brief History of Computing*, pages 229–252, 2012.

[403] Yoshua Bengio. The Consciousness Prior. (1):1–4, 2017.

[404] Sinno J Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2010.

[405] Wenyuan Dai, O. Jin, G.R. Xue, Q. Yang, and Y. Yu. Eigentransfer: a unified framework for transfer learning. *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 193–200, 2009.

[406] Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from Simulation to Real World through Learning Deep Inverse Dynamics Model. pages 1–8, oct 2016.

[407] Sebastian Flennerhag, Pablo G. Moreno, Neil D. Lawrence, and Andreas Damianou. Transferring Knowledge across Learning Processes. pages 363–370, 2014.

[408] Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J Mankowitz, and Shie Mannor. A Deep Hierarchical Approach to Lifelong Learning in Minecraft. pages 1553–1561, 2017.

[409] German I. Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of human actions with deep neural network self-organization. *Neural Networks*, 96:137–149, 2017.

[410] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. 2017.

[411] Chrisantha Fernando, Jakub Sygnowski, Simon Osindero, Jane Wang, Tom Schaul, Denis Teplyashin, Pablo Sprechmann, Alexander Pritzel, and Andrei A Rusu. Meta-Learning by the Baldwin Effect. 2018.

[412] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. pages 1–17, 2016.

[413] Benjamin James Lansdell and Konrad Paul Kording. Towards learning-to-learn. pages 1–8, nov 2018.

[414] Abhishek Gupta, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Unsupervised Meta-Learning for Reinforcement Learning. 2018.

[415] Emilio Parisotto, Soham Ghosh, Sai Bhargav Yalamanchi, Varsha Chinnaobireddy, Yuhuai Wu, and Ruslan Salakhutdinov. Concurrent Meta Reinforcement Learning. 2019.

[416] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, (May):210–215, 2018.

[417] Antonio Lieto, Mehul Bhatt, Alessandro Oltramari, and David Vernon. The role of cognitive architectures in general artificial intelligence. *Cognitive Systems Research*, 48(June):1–3, 2018.

[418] Jelmer P. Borst and John R. Anderson. Using the ACT-R Cognitive Architecture in combination with fMRI Data. *An introduction to model-based cognitive neuroscience*, pages 339–352, 2015.

[419] Martin J. Pickering and Andy Clark. Getting ahead: Forward models and their place in cognitive architecture. *Trends in Cognitive Sciences*, 18(9):451–456, 2014.

[420] Paul E Silvey. Leveling Up: Strategies to Achieve Integrated Cognitive Architectures. *AAAI 2017 Fall Symposium Series-A Standard Model of Mind*, 17(5):460–465, 2017.

[421] Jordi Vallcerdu, Max Talanov, Salvatore Distefano, Manuel Mazzara, Alexander Tchitchigin, and Ildar Nurgaliev. A cognitive architecture for the implementation of emotions in computing systems. *Biologically Inspired Cognitive Architectures*, 15:34–40, 2016.

[422] John E. Laird, Christian Lebiere, and Paul S. Rosenbloom. A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine*, 38(4):13–26, 2017.

[423] Mostafa Fahmy. Artificial Life and the Philosophy of Science. 1992.

[424] Wendy Aguilar, Guillermo Santamaría-bonfil, Tom Froese, and Carlos Gershenson. The past, present, and future of artificial life. 1(October):1–15, 2014.

[425] Mark A Bedau, Norman H Packard, Chris Adami, David G Green, and Thomas S Ray. Open Problems in Artificial Life. 376(2000):363–376, 2001.

[426] Mark A Bedau. Artificial life: organization, adaptation and complexity from the bottom up. 7(11):505–512, 2003.

[427] Antonio Lieto. Representational limits in cognitive architectures. *CEUR Workshop Proceedings*, 1855:16–20, 2017.

[428] William Lotter, Gabriel Kreiman, and David Cox. Deep Predictive Coding Networks for Video Predictions and Unsupervised Learning. pages 1–18, 2017.

[429] Leonid Perlovsky. Learning in brain and machine — complexity , Gödel , Aristotle. 7(November):1–3, 2013.

[430] Jessica B. Hamrick. Analogues of mental simulation and imagination in deep learning. *Current Opinion in Behavioral Sciences*, 29:8–16, 2019.

[431] Francesco Savelli and James J. Knierim. AI mimics brain codes for navigation. *Nature*, 557:313–314, 2018.

[432] H. Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The Role of Hippocampal Replay in Memory and Planning. *Current Biology*, 28(1):R37–R50, 2018.

[433] Guan-Horng Liu, Avinash Siravuru, Sai Prabhakar, Manuela Veloso, and George Kantor. Learning End-to-end Multimodal Sensor Policies for Autonomous Navigation. (CoRL):1–13, may 2017.

[434] Charlotte N. Boccara, Michele Nardin, Federico Stella, Joseph O'Neill, and Jozsef Csicsvari. The Entorhinal Cognitive Map is Attracted To Goals. *Science*, 1447(March):1443–1447, 2019.

[435] Michael Garcia Ortiz and Softbank Robotics Europe. Unsupervised Emergence of Spatial Structure from Sensorimotor Prediction. pages 1–16, 2019.

[436] Michael Garcia Ortiz and Alban Laflaquière. Learning Representations of Spatial Displacement through Sensorimotor Prediction. may 2018.

[437] Stan Franklin. A Foundational Architecture for Artificial General Intelligence. (June 2007), 2014.

[438] Naoya Arakawa, The Whole, Brain Architecture, and Hiroshi Yamakawa. Whole brain architecture approach is a feasible way toward an Artificial General Intelligence. (October), 2016.

[439] Hiroaki Kitano. Artificial Intelligence to Win the Nobel Prize and Beyond: Scientific Discovery. *AI magazine*, 37(1):39–49, 2016.

[440] Roman V Yampolskiy. AI-Complete , AI-Hard , or AI-Easy – Classification of Problems in AI. (Searle 1980), 2011.

[441] Roman Ramamoorthy, Anand Yampolskiy. Beyond MAD: the race for artificial general intelligence. *ICT Discoveries*, Special Is(1):1–8, 2018.

[442] Ajay Agrawal, Joshua Gans, and Avi Goldfarb. The Impact of Artificial Intelligence on Innovation. *The Economics of Artificial Intelligence*, (September):115–148, 2019.

[443] Nick Bostrom. Ethical Issues in Advanced Artificial Intelligence. *Int. Institute of Advanced Studies in Systems Research and Cybernetics*, 2:12–17, 2003.

[444] Michaël Trazzi and Roman V. Yampolskiy. Building Safer AGI by introducing Artificial Stupidity. 2018.

[445] Vinge Vernor. The Coming Technological Singularity. *Whole Earth Review*, 81:88–95, 1993.

[446] Anthony Miller. The intrinsically linked future for human and Artificial Intelligence interaction. *Journal of Big Data*, 6(1), 2019.

[447] Matjaz Gams, Irene Yu Hua Gu, Aki Härmä, Andrés Muñoz, and Vincent Tam. Artificial intelligence and ambient intelligence. *Journal of Ambient Intelligence and Smart Environments*, 11(1):71–86, 2019.

[448] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[449] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444, 2007.

[450] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep Reinforcement Learning that Matters. pages 3207–3214, 2017.

Appendix A

Cognitive Science Background

Cognitive science inspiration for spatial navigation can include various scientific disciplines; however, this background mainly focuses on psychology, neuroscience, and artificial intelligence. We start with theories on the emergence of cognition, and transition to the conceptualization and development of spatial cognition. Furthermore, following up with discussions on the role of the hippocampus and highlighting the rise of deep learning.

A.1 Cognition through Foraging

The cognitive improvements of humans over millions of years eventually led to the development of spatial cognition; however, several theories point to reasons why this happened [151].

One hypothesis for the evolution of cognition comes from dietary changes, as opposed to the established social intelligence hypothesis [152]. Comparisons between different primate dietary strategies show complexity-based food acquisition. For example, bark or leaves from trees are widely available; therefore, simple cognitive strategies suffice. However, relying on patchy fruits or seasoned harvest requires sophisticated memory and spatial navigation for survival. Primates exploiting complex foods are more risk-seeking, perform better on spatial tasks, and have more complex decision-making skills compared to primates with simple diets [151]. These advanced skills are necessary for deciding when to leave a food patch and memorize the location of other food sources.

A.2 Cognitive Mapping

Edward C. Tolman's work was groundbreaking and questioned: "whether cognition can exist in animals other than humans and, if so, what is it, and how might it be manifested through behavior?" [33].

Tolman's idea for a cognitive map substantiated due to five maze experiments with rodents, which were inspired by the cognitive learning paradigm he adopted instead of the traditional stimulus-response learning approach. He concluded that animals use a wide range of cues during the initial exposure to the experiment. However, overtraining animals leads to an increasingly smaller and more specific set of cues relevant to the task [31].

The major problem with the theory of the cognitive map comes from animal experimentation results, which can be explained using more straightforward strategies [153]. Path integration could be the underlying mechanism, not the map itself [35, 51, 154]. A prominent aspect of the cognitive mapping theory relates to the use of novel shortcuts. However, testing the novel shortcutting hypothesis is challenging, since animal experiments have to validate that actions are truly novel [155].

Additionally, it is difficult to pinpoint the exact role of the hippocampus and entorhinal cortex in the cognitive map [155, 156, 157], and this also led to various new definitions and interpretations emerging from the cognitive map theory [158, 159]. New definitions and interpretations range from literal map interpretations to the psychological ordering of elements in a spatial setting [32].

A.3 Role of the Hippocampus

Brain functioning relies on signals from the sensory and association cortices. The integration of signals from different cortices and additional signals such as motivation, goals, and rewards have been proposed to be integrated using mixed selectivity encoding [6]. The integration of many signals due to the mixed selectivity theory makes it almost impossible to retrace observed behavior back to the original stimulus [80]. The mammalian hippocampal circuit could potentially be the first non-sensory cognitive function to be understood entirely [160].

The hippocampus receives information from the entorhinal cortex, parahippocampal cortex, and the frontal lobe, which could create and maintain a cognitive map [3]. The parahippocampal cortex is associated with the integration of spatial landmarks [162], and the frontal lobe is responsible for planning routes during active navigation [163]. Both signals are integrated into the hippocampus, enabling goal-based navigation and representation of detours based on landmarks [3]. The representation of the hippocampus is more context-dependent compared to the entorhinal and parahippocampal cortices [164], this leads to more flexible representations of space in the hippocampus [145, 165]. For more information on the interaction between the underlying areas in the medial temporal cortex, see Figure A.1.

The experiments from Tolman inspired numerous researchers, including O'Keefe and Nadel, who associated the neurobiological basis of the hippocampus with the cognitive map theory [38, 161]. Additionally, Redish utilized the cognitive map foundation through research on non-spatial domains [33].

The hippocampal interaction model explains why animals can track vast distances while searching for food [151, 166, 167], which is mainly through the integration of external cues, speed, and direction [58, 168, 169]. The entorhinal-like representation is theorized to aid complex navigation within the medial temporal cortex, such as path-integration and planning [170]. Also, many cells in the hippocampus provide insights into the navigational and memory-related responsibilities of the brain [171, 172], including spatial context, land-

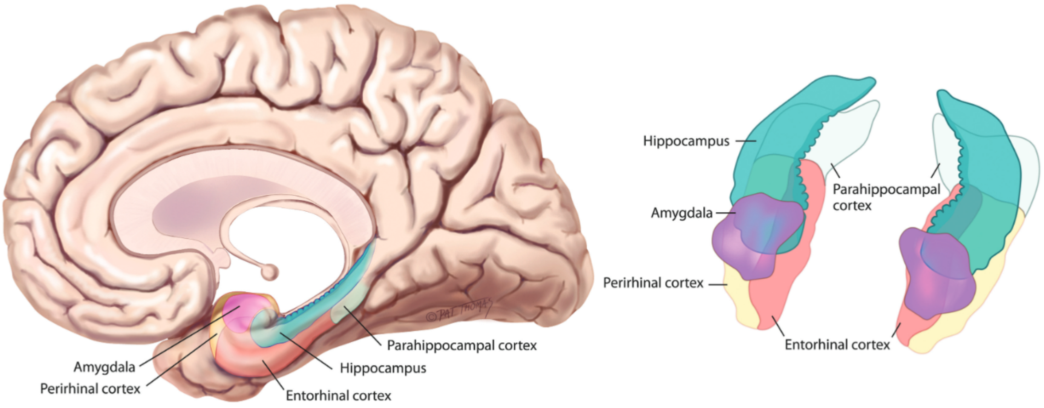


Figure A.1: **Sketch of the essential cortices in the medial temporal lobe interacting with the hippocampus.**
The processing stream flows from the parahippocampal and perirhinal cortex to the entorhinal cortex. From there, the information is passed on to the hippocampus. The image was copied without permission from original publisher Purves D, Brannon E, Cabeza R, et al. Principles of Cognitive Neuroscience. Sunderland, MA: Sinauer Associates; 2008 taken from [13]

marks, and goals [3]. Lesion studies in brain regions surrounding the hippocampus have more detrimental effects on spatial navigation compared to hippocampal lesions [6, 163]. Some studies even proposed that the role of spatial navigation relies on the entorhinal cortex as opposed to the traditional hippocampal model [173].

Space might be just one of many variables underlying the behavior of cells in the entorhinal cortex and hippocampus. Perhaps integrating these different signals creates a cognitive map [38, 173, 174, 175, 176]. Research has already shown that the hippocampus uses both the spatial and time-dependent signals for context- and task-dependent representation [177], which can support episodic memory [178, 179, 180, 181, 182], predict the future [4, 123, 183], and process abstract knowledge [38, 41, 184].

A.4 Progress in Deep Learning

Simple models for artificial neural networks started with the Perceptron model in 1958 [185], followed by Hopfield networks in 1982 [186], and later the Neocognitron model in 1988 [187]. The idea of stacking layers with simplistic neurons has survived two artificial intelligence winters between 1973 and 1993 [46]. This idea is still the conceptual foundation for large neural networks to this day. Each deep learning layer typically contains many neurons; each neuron weights incoming connections and transforms the sum of the input using a non-linear function to output a single value. Connected neurons receive the output and similarly transform the input signals [188]. Each layer of neurons is trained using the back-propagation algorithm.

The back-propagation algorithm is essential for deep learning architectures and enables discovering intricate structures in high-dimensional datasets [134]. Back-propagation uses the model's loss and calculates partial derivatives for weights in the network. The gradients trickle down the network from output to input, and the gradient direction vector resulting from this process is applied to the network weights to optimize the performance. A popular back-propagation approach for training deep learning architectures is called Stochastic Gradient Descent and calculates the network weight gradients based on a few training examples [134]. Stochastic Gradient Descent was believed to get trapped easily in local minima; however, theoretical evidence shows that the learning algorithm instead gets stuck in numerous suboptimal saddle points, which have no gradient [100].

The widespread adaptation of deep learning was made possible through the developments of efficient data storage, dedicated hardware, and open-source software tools [189]. Modern deep learning experts rely on graphical processing units to perform model optimization concurrently [190], which speeds up the training process.

Appendix B

Cognitive Research Impact on Artificial Intelligence

Understanding how species evolved and developed cognitive abilities over millions of years can inspire artificial intelligence research, for example, the development of cognitive regions in the brain [191, 192, 193, 194] and could relate to the order in which researching general artificial intelligence would be most productive [195].

While the brain is slower than state-of-the-art computer processors, the cortical hierarchy and parallel communication provides an efficient substrate for biological computations [196, 197]. Neuroscience and artificial intelligence have been in a reciprocal relationship, exchanging ideas and algorithms for decades [49, 50, 113, 150, 198, 199, 200]. A worrying trend, however, is that artificial intelligence researchers increasingly reference mostly only other artificial intelligence papers, limiting the diffusion of research to other cognitive science disciplines [201].

Various observations in neuroscience have inspired new algorithms and architectures in artificial intelligence research. Inspiration can come from sparse coding [122, 124, 184], Hebbian learning [67, 140, 167, 202, 203, 204], biological evolution [95, 192, 205, 206, 207, 208, 209, 210, 211, 212], intrinsic motivation [213, 214, 215, 216, 217], curiosity and novelty [218, 219, 220, 221, 222, 223, 224], cognitive control [225, 226, 227, 228, 229], central executive [230, 231, 232], functional connectivity in the brain [233, 234, 235, 236, 237, 238], interactions between cognitive regions [164, 13, 239, 240, 241, 242, 243], social cognitive development [244, 245, 246, 247, 248, 249, 250], brain analysis [251, 252, 253, 254, 255, 256, 257], memory functioning [258, 259, 260, 261], experience replay during sleep [262, 263, 264], abstract context representation [265, 266, 267], cognition [151, 152, 268, 269, 270, 271] and consciousness [272, 273, 274].

These biological inspirations influenced memory systems [275, 276], machine cognition [227, 277, 278, 279, 280, 281, 282], artificial consciousness [222, 283, 284, 285], convolutional neural networks [46, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298], deep neural networks in general [299, 300, 301, 302, 303, 304], robotics [48, 55, 305, 306, 307, 308, 309], and reinforcement learning (RL) [36, 150, 310, 311, 312, 313, 314, 315, 316], through successor representation [16, 30, 317, 318, 319, 320, 321, 322, 323], memory-based RL [324, 325, 326, 327, 328], intrinsic motivation and curiosity in RL [329, 330, 331, 332, 333, 334, 335, 336, 337, 338], hierarchical RL [339, 340, 341, 342, 343], evolutionary RL [212, 344, 345], and deep (neural) RL [47, 262, 346, 347, 348, 349, 350, 351].

Artificial intelligence currently lacks reliable working memory [258], efficient learning strategies [352], and the ability to perform cognitive tasks [45]. Features such as one-shot learning [353, 354] and episodic control [325] are essential for addressing the slow learning problem in deep neural networks. Additionally, developments in modeling working memory through differentiable neural computers could replace prevailing basic memory modules such as the LSTM, to process more complex tasks [328].

Deep learning has significantly expanded the number of trainable architectures over the past fifteen years [355, 356, 357, 358, 359, 360, 361, 362]. Especially architectures involved in applications such as Natural Language [363, 364, 365] and Image Processing [359, 366, 367, 368, 369] have experienced significant improvements. Some architectures take a biological approach through evolutionary- [344, 345, 370, 371, 372, 373, 374], reservoir- [375, 376, 377, 378, 379, 380], or hierarchical computing [211, 369, 381, 382, 383, 384, 385]. Additional approaches explicitly encode statistical regularities through abstract reasoning [386, 387, 388, 389, 390, 391], relational reasoning [387, 392, 393, 394, 395], logic [396, 397, 398, 399], and symbolic-based neural networks [221, 400, 401, 350].

Deep learning models are frequently required to learn abstract representations, and multiple modalities [133, 402, 403], due to the increasing application of deep learning in more general problems [404]. While humans naturally generalize experience during abstract tasks, deep learning struggles to exploit similar mechanisms to transfer knowledge [404, 405, 406, 407], learn multiple tasks simultaneously [408, 409, 410], and perform meta-learning [390, 411, 412, 413, 414, 415]. Bridging this gap between machine and natural intelligence requires decomposing problems into commonsense understanding through interactive experiments [49, 416].

Several modeling approaches have attempted to grasp the concept of natural intelligence through developing cognitive architectures [44, 141, 224, 417, 418, 419, 420, 421, 422], including Artificial Life [423, 424, 425, 426], Biophysical Modeling, Connectionism [148], Cognitivism, and Probabilistic Modeling. The approaches for modeling artificial intelligence split into two paradigms: bottom-up emergence and top-down abstraction [147, 273, 427, 428, 429]. None of the mentioned cognitive architectures encapsulate both aspects of learning. The inability to express complex reasoning based on individual neurons is called the computational explanatory gap [171, 280]. Closing this computational explanatory gap enables developing general solutions for artificial intelligence problems [227].

Entorhinal-like representation is associated with various cognitive concepts, such as imagination [349, 430], goal-directed navigation and planning [1, 11, 20, 21, 162, 165, 431, 432, 433, 434], and episodic future thinking [4, 52]. The current state of artificial intelligence cannot solve these problems in general, but entorhinal-like representations can help inspire new algorithms for solving these problems [49, 435, 436]. Also, the ability to solve general problems is of great interest to the artificial general intelligence community [42, 44, 280, 417, 437, 438, 439, 440, 441]. Realizing consistent and prevalent occurrence of entorhinal-like representation in recurrent neural network research could potentially streamline the three major factions in artificial intelligence, which are: robotics, artificial intelligence, and symbolic systems [442].

Even though artificial intelligence is booming [442], researchers are cautious when it comes to developing cognitive or general artificial intelligence [402, 443, 444]. Researchers warn about the sudden arrival, or singularity [445], of widespread artificial cognitive abilities, potentially destroying the inventiveness of humans. This sudden arrival could align with other technological progress [446], such as ambient intelligence [447] and quantum machine learning [125, 448]. One way to keep track of machine intelligence is through developing intelligence tests [15], for example, to determine artificial universal intelligence [449].

Appendix C

Experimental Base Model

All experiments use a unique recurrent neural network base model. Every aspect of the learning process is manipulatable, such as the layer settings, recurrent network properties, and hyper-parameters.

The TensorFlow base model divides the responsibilities of the program into the training model and the trainer. The model defines the architecture of recurrent cells, the individual layers between the input and output data, and regularization approaches. The base model pipeline consists of (1) preparing the recurrent cell, (2) set up the output layer, and (3) optimize the loss. The recurrent cell setup relies on the experiment configuration, which dictates the use of weight initialization, activation function, and cell type. The recurrent cell preparation also includes enabling GPU acceleration to minimize training time. The linear output layer projects the recurrent output to a two-dimensional trajectory. Finally, the optimizer uses the model loss, between the target and predicted trajectories, and the regularization loss to optimize the network using clipped gradients. The trainer has access to the data simulator and can request a new batch for each training epoch. The data preprocessor prepares the batch to fit the input dimensionality and the target distribution. The trainer also executes the training model and gathers data for visualization purposes.

The underlying program operating on the model and trainer manages the configuration and visualization of neural networks during training. The main program starts by reading the batch specifications, baseline configuration, and experiment configuration files. The program then repeats the baseline experiment and other experiments according to the configurations. The results from the experiments are used to compare the loss, performance, and resulting representation from recurrent neurons. The program visualizes the performance and representation of the training model regularly. The visualization approaches also save the plots automatically, allowing users to inspect the progression of the model during and after the training process. The individual plots are combined and animated to create time-based plots, which can be used to reflect on the training progress and representation.

Appendix D

Theoretical Model Entorhinal-like Representation

The following sections describe basic theoretical models for entorhinal cell types used in this thesis. The mathematical formula is similar, but not equal to the implemented function in software. The formula is designed to give a general indication of the activation function for each cell type.

D.1 Place cells

Place cells are modeled using a Gaussian activation function [1] (equation D.1), each cell is assigned its own place center $\mu_i^{(c)} \in \mathbb{R}^2$. The activation resembles the Euclidean distance between the agent's position $x \in \mathbb{R}^2$ and the place center. The scale of the place cell determines the radius of the place cell. The distribution of place-like activation is measured using the softmax function. The activity for each place cells N is weighted by the sum of all place cell activations (equation D.2).

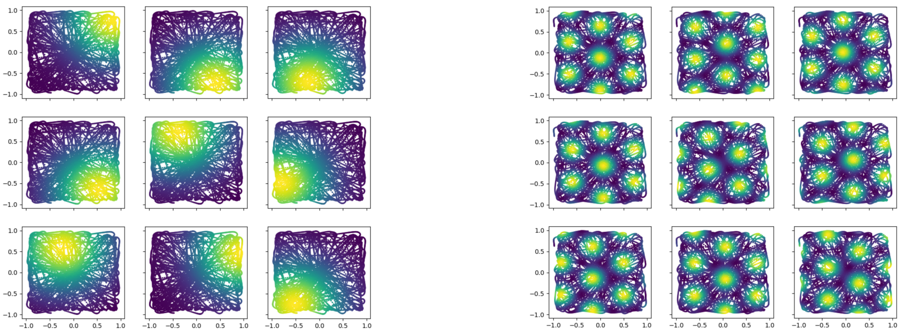
$$c_i = e^{-\frac{\|x-\mu_i^{(c)}\|_2^2}{2(\sigma^{(c)})^2}} \tag{D.1}$$

$$sc_i = \frac{c_i}{\sum_j^{|N|} c_j} \tag{D.2}$$

D.2 Grid cells

Mathematical models for grid cells use a random position $\mu_i^{(g)} \in \mathbb{R}^2$ to center the hexagonal activation pattern (equation D.3). The hexagonal activation pattern is generated using the vector product between the agent's positional vector and the three basis vectors. The agent's position vector $x \in \mathbb{R}^2$ is represented by the x- and y-coordinates and the three base vectors κ_j are offset by 0, 60, and 120 degrees [72].

$$g_i = \frac{1}{3}\sum_{j=1}^{3} \cos\left(\kappa_j (x - \mu_i^{(g)})\right) \tag{D.3}$$



(a) **Place activation**
Nine artificially generated place-like cells showing Gaussian activation patterns. The place cell activation concentrates on a randomly generated place center.

(b) **Grid activation**
Hexagonal pattern of nine randomly generated grid-cells. The grid cells have a different grid center leading to slight differences between the activation patterns.

Figure D.1: Place and grid-like cell activation patterns.

D.3 Head-direction cells

The head-direction metric calculates the angular activation [1] (equation D.4), for which each cell is assigned a random angle $\mu_i^{(h)} \in [-\pi, \pi]$. The cell activation is calculated using the scalar $\kappa^{(h)}$ and the cosine of the angular difference between the agent's direction ϕ and the assigned angle. The weighted activation of each cell is defined by the softmax function (equation D.5).

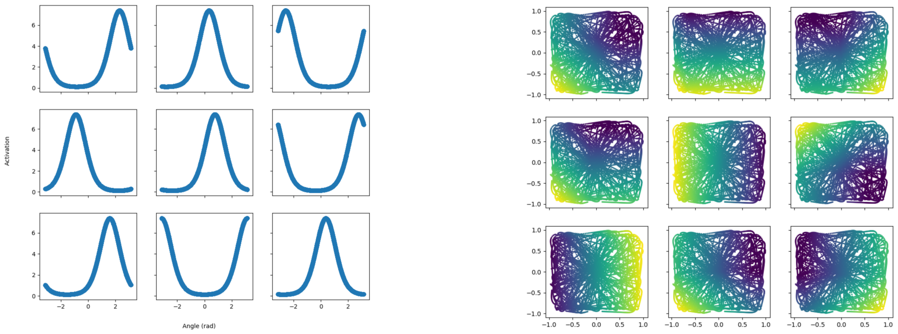
$$h_i = \kappa^{(h)} \cos\left(\phi - \mu_i^{(h)}\right) \tag{D.4}$$

$$d_i = \frac{e^{h_i}}{\sum_j^{|N|} e^{h_j}} \tag{D.5}$$

D.4 Border cells

The artificial border cells are modeled using the boundary vector model [73, 126], where each border cell has a preferential activation direction $\mu_i^{(b)}$. The angular difference between the boundary wall vectors w and preferential direction together with the wall proximity dictates the activation magnitude. If the preferential activation direction is parallel to the wall vector, then a border-wide activation pattern is generated. This product is scaled by a scalar κ to vary the activation of each border cell.

$$b_i = 1 - \max_j \frac{(w_j \cdot \mu_i^{(b)})}{\kappa} \tag{D.6}$$



(a) **Head-direction activation**
Nine artificial head-direction cells show directional specialization for a random angle.

(b) **Border activation**
Border cells reflects an activation pattern related to the directional preference and the boundary angle.

Figure D.2: Head-direction and border-like cell activation patterns.

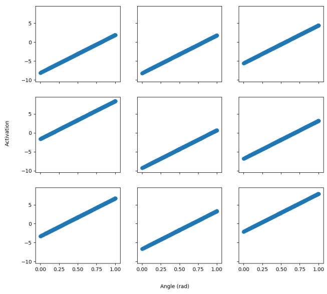


Figure D.3: **Speed activation**
Nine artificial speed cells show velocity-based specialization using a random bias.

D.5 Speed cells

Theoretical speed cells are defined using a linear relationship between traveling velocity and spiking frequency [9]. Each speed cell is assigned a base activation $\mu_i^{(s)}$ and the current velocity v determines the velocity-based activation. Additionally, the activation is adjusted with the velocity scale κ.

$$s_i = \frac{(v - \mu_i^{(s)})}{\kappa} \tag{D.7}$$

Appendix E

Research Design

The figures on the following pages show the research design and progress. The highlighted boxes and arrows shown in the following figures visualize the path taken by this study, and the other options show alternatives research routes. The three different stages of research are visualized in Figure E.1, Figure E.2, and Figure E.3.

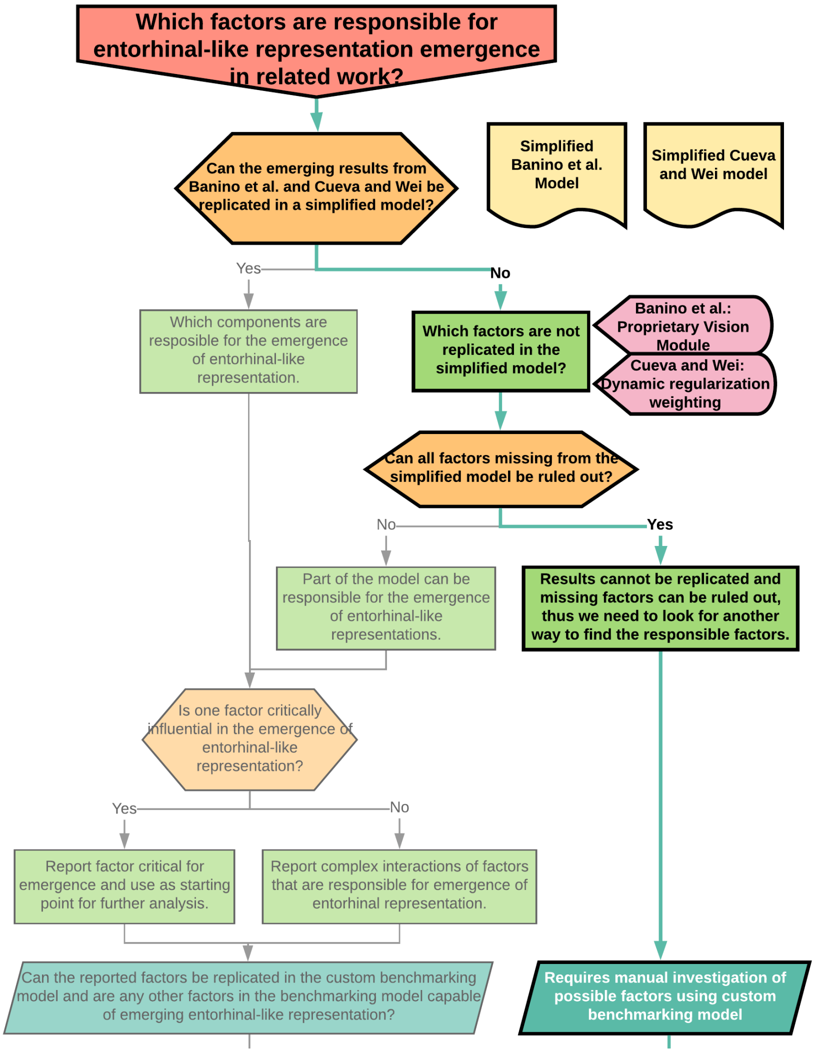
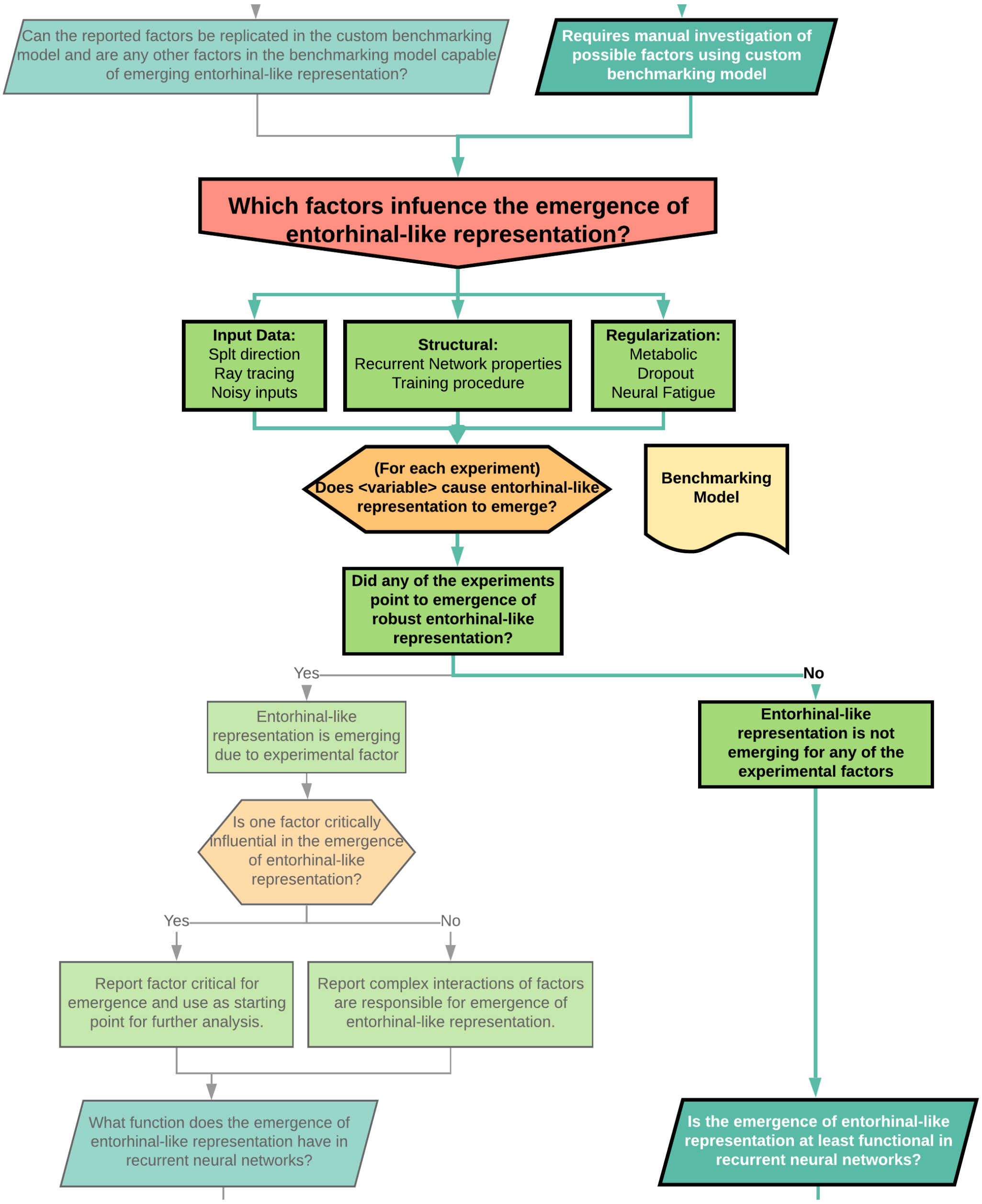

Which factors are responsible for entorhinal-like representation emergence in related work?
Can the emerging results from Banino et al. and Cueva and Wei be replicated in a simplified model?
Simplified Banino et al. Model
Simplified Cueva and Wei model
Yes
No
Which components are resposible for the emergence of entorhinal-like representation.
Which factors are not replicated in the simplified model?
Banino et al.: Proprietary Vision Module
Cueva and Wei: Dynamic regularization weighting
Can all factors missing from the simplified model be ruled out?
No
Yes
Part of the model can be responsible for the emergence of entorhinal-like representations.
Results cannot be replicated and missing factors can be ruled out, thus we need to look for another way to find the responsible factors.
Is one factor critically influential in the emergence of entorhinal-like representation?
Yes
No
Report factor critical for emergence and use as starting point for further analysis.
Report complex interactions of factors that are responsible for emergence of entorhinal representation.
Can the reported factors be replicated in the custom benchmarking model and are any other factors in the benchmarking model capable of emerging entorhinal-like representation?
Requires manual investigation of possible factors using custom benchmarking model

Figure E.1: **Related Work Research Design**
Earlier work experiments rely on the simplified Banino et al. and Cueva and Wei model. The proprietary vision module and dynamic regularization weighting from these models can be factored out. Banino et al.'s proprietary vision module is optional since Cueva and Wei did not rely on a vision source. Also, the dynamic regularization weighting by Cueva and Wei only improved the clarity of emergent representation but was not essential for the emergence of entorhinal-like representation, according to their appendix.



Can the reported factors be replicated in the custom benchmarking model and are any other factors in the benchmarking model capable of emerging entorhinal-like representation?
Requires manual investigation of possible factors using custom benchmarking model
Which factors infuence the emergence of entorhinal-like representation?
Input Data: Splt direction Ray tracing Noisy inputs
Structural: Recurrent Network properties Training procedure
Regularization: Metabolic Dropout Neural Fatigue
(For each experiment) Does <variable> cause entorhinal-like representation to emerge?
Benchmarking Model
Did any of the experiments point to emergence of robust entorhinal-like representation?
Yes
No
Entorhinal-like representation is emerging due to experimental factor
Entorhinal-like representation is not emerging for any of the experimental factors
Is one factor critically influential in the emergence of entorhinal-like representation?
Yes
No
Report factor critical for emergence and use as starting point for further analysis.
Report complex interactions of factors are responsible for emergence of entorhinal-like representation.
What function does the emergence of entorhinal-like representation have in recurrent neural networks?
Is the emergence of entorhinal-like representation at least functional in recurrent neural networks?

Figure E.2: **Factors Research Design**

Factors include analysis of input features, structural properties, and regularization techniques. The experiments showed no emergence of entorhinal-like representation; thus, we must reflect on the use of entorhinal-like representation in recurrent neural networks. Additional analysis, if such entorhinal-like representation were present, would be able to single out factors or a mixture of factors responsible for the emergence of entorhinal-like representation.

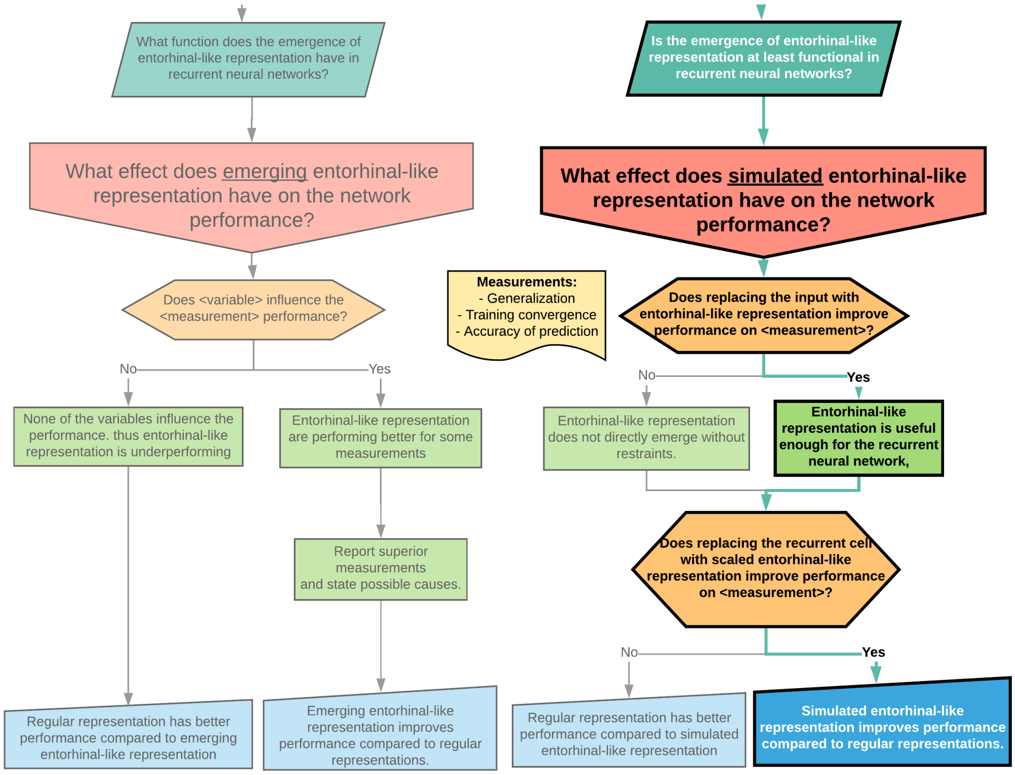

What function does the emergence of entorhinal-like representation have in recurrent neural networks?
What effect does emerging entorhinal-like representation have on the network performance?
Does <variable> influence the <measurement> performance?
No
Yes
None of the variables influence the performance. thus entorhinal-like representation is underperforming
Entorhinal-like representation are performing better for some measurements
Report superior measurements and state possible causes.
Regular representation has better performance compared to emerging entorhinal-like representation
Emerging entorhinal-like representation improves performance compared to regular representations.
Is the emergence of entorhinal-like representation at least functional in recurrent neural networks?
What effect does simulated entorhinal-like representation have on the network performance?
Measurements:
- Generalization
- Training convergence
- Accuracy of prediction
Does replacing the input with entorhinal-like representation improve performance on <measurement>?
No
Yes
Entorhinal-like representation does not directly emerge without restraints.
Entorhinal-like representation is useful enough for the recurrent neural network,
Does replacing the recurrent cell with scaled entorhinal-like representation improve performance on <measurement>?
No
Yes
Regular representation has better performance compared to simulated entorhinal-like representation
Simulated entorhinal-like representation improves performance compared to regular representations.

Figure E.3: **Performance Impact Research Design**
The previous experiments did not show emergent representation; therefore, this experiment has to rely on simulated entorhinal-like representation. Results indicate that replacing the input with entorhinal-like representation improves training performance for border- and place-like cells. Additionally, the training performance of replaced recurrent representation showed improvements over regular training for border- and grid-like cells. Thus, entorhinal-like representation performs better compared to regular, unstructured representations.

Appendix F

Additional Analysis

Additional steps were taken to confirm the results of this study. First, relating results between other open-source models and the model developed in this thesis. Second, the results from repeated experiments are displayed to investigate the consistency and diversity of spatially activated cells. Third, an additional experiment was executed using the best-practices from this study and trained longer to explore optimal training performance.

F.1 Open-Source Models

GitHub provides open-source models for various projects. Artificial intelligence developers and researchers have replicated the models from Banino et al. [1] and Cueva and Wei [2], and open-sourced their models and results.

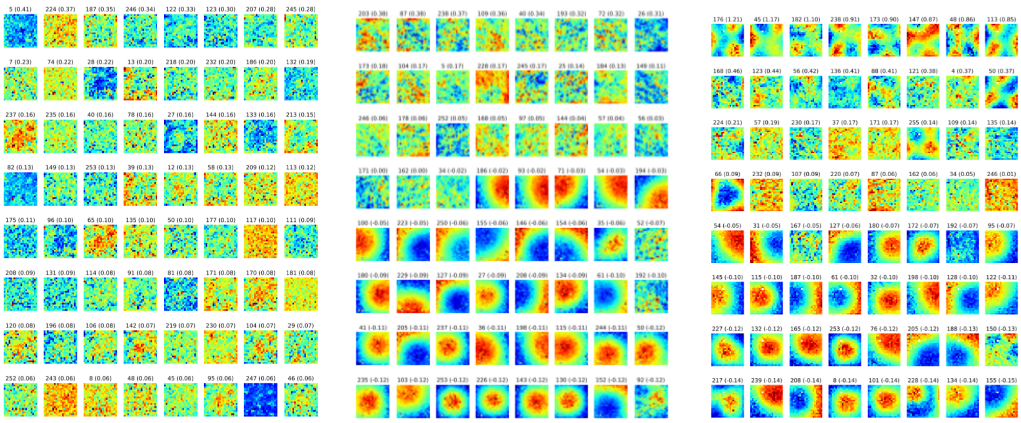
F.1.1 Banino et al.

DeepMind published their original model from Banino et al. [1] on GitHub[1] together with the original simulation data to generate the grid-like patterns. A partition of the neurons became spatially activated, similar to some of the non-linearly activated neurons from this thesis. Additionally, the results show a subset of neurons representation converged to grid-like activation patterns. Possibly when Rosa's network is trained longer, then this could result in gradual development of entorhinal-like representation.

Stefano Rosà created his version of the DeepMind model and published it on GitHub[2] as well. The results do not show grid-like patterns, but non-linear spatially activated neurons do occur. Possibly training the network longer can develop entorhinal-like representation just like DeepMind's original work.

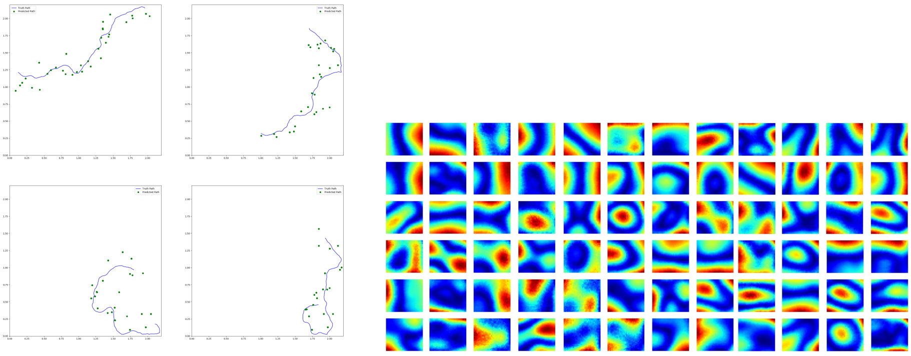
[1] `https://github.com/deepmind/grid-cells`

[2] `https://github.com/R-Stefano/Grid-Cells`



(a) Epoch 0 (b) Epoch 50 (c) Epoch 100

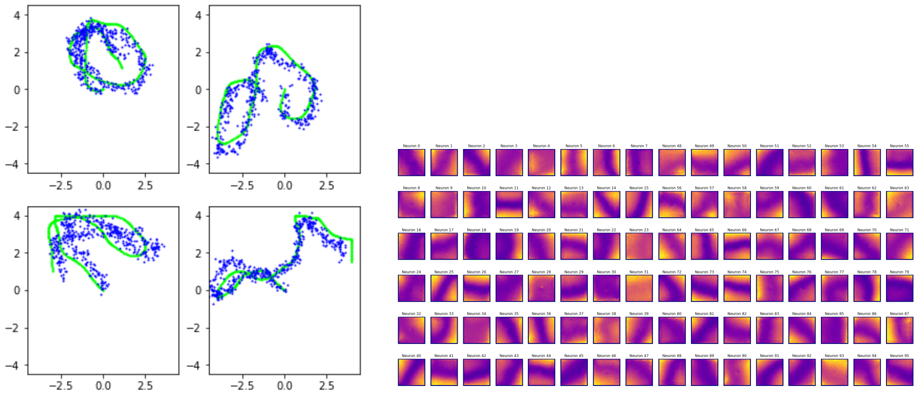
Figure F.1: **Deep Mind's original model spatial activity results**
Each plot cell visualizes the average spatial activity for an artificial neuron. The cells initially have no spatially activated representation. During the training procedure, the cells generate non-linear and even grid-like representations — the grid-like cells developed around epoch 100.



(a) **Trajectory plot**
Prediction is somewhat sloppy compared to the target trajectory.

(b) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron.

Figure F.2: **DeepMind replicated model results by Stefano Rosà**
The training performance shows average path integration performance, and the representation highlights sophisticated non-linear activation patterns.



(a) **Trajectory plot**
Trajectory generation shows similar shapes, but varying accuracy compared to the target trajectories.

(b) **Spatial activity plot**
Each plot cell visualizes the average spatial activity for an artificial neuron.

Figure F.3: **Cueva and Wei replicated model results by Unity Technologies**
The predicted path of the agent shows consistent path integration performance, and the cell representation show linear spatially activated neurons.

F.1.2 Cueva and Wei

Unity Technologies replicated Cueva and Wei's [2] model using their 3D modeling simulator [131], and published the data and model on GitHub[3]. The results show no grid-like representation, but the results are reminiscent of linear spatially activated cells from this thesis.

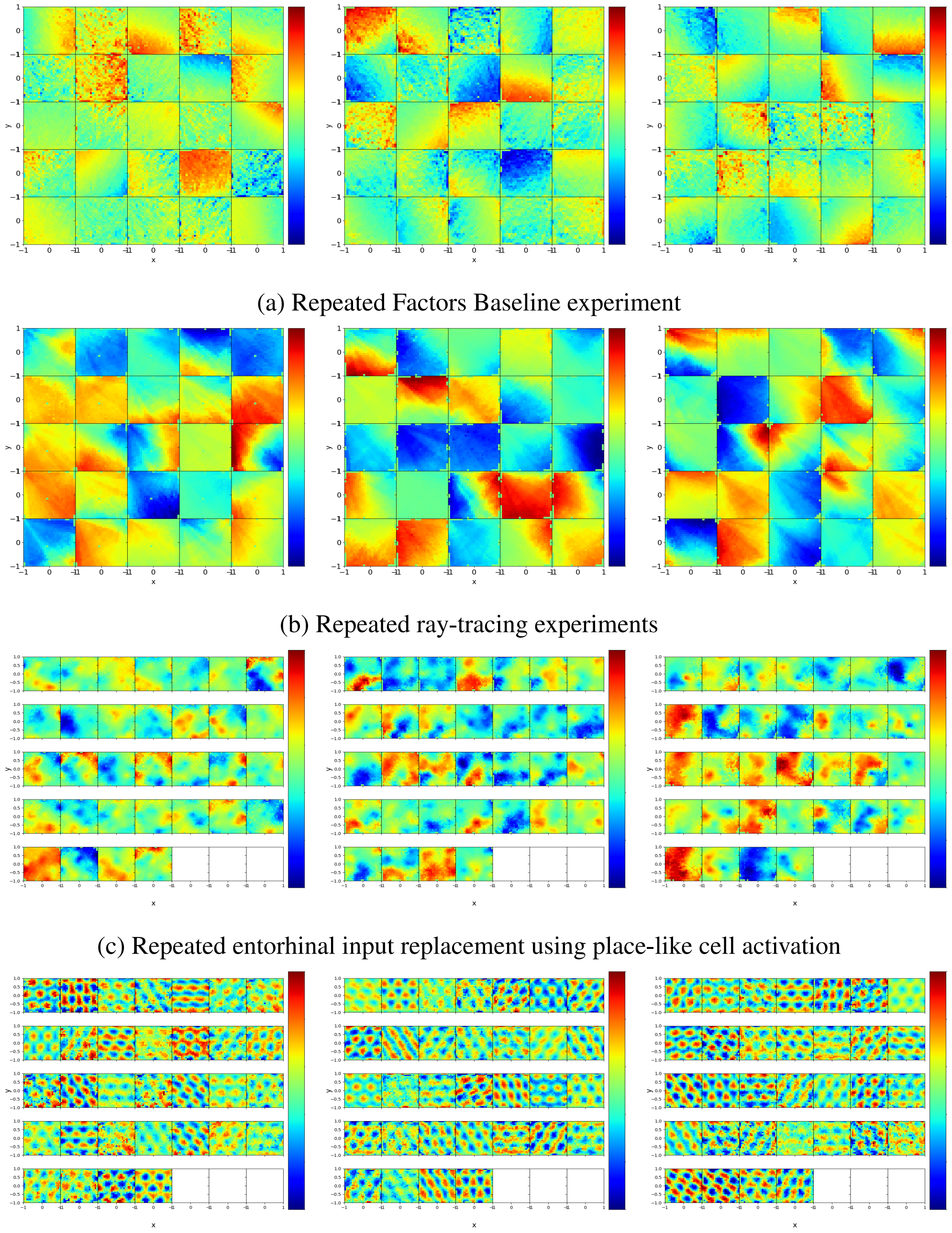
F.2 Robust Representation

The following figures show the repeated training activation representation for various experiments from this thesis. Each experiment shows consistent activation and comparable spatially activated representations between experiments. We can conclude that the representations are inherent to the network models and training approach.

F.3 Combined Experiment

This experiment integrates the best-performing configurations into one model, see Figure F.5. The model uses the GRU recurrent cell, ray-tracing input combined with regular velocity and direction, has 81 recurrent neurons, initialized uniformly, and uses recurrent regularization. This model created non-linear input representation, but training longer did not result in a more complex representation.

[3] https://github.com/Unity-Technologies/rat-rnn



(a) Repeated Factors Baseline experiment

(b) Repeated ray-tracing experiments

(c) Repeated entorhinal input replacement using place-like cell activation

(d) Repeated entorhinal input replacement using grid-like cell activation

Figure F.4: **Representation of repeated experiments**
Both the distribution and frequency of spatially activated neurons are consistent between experiments.

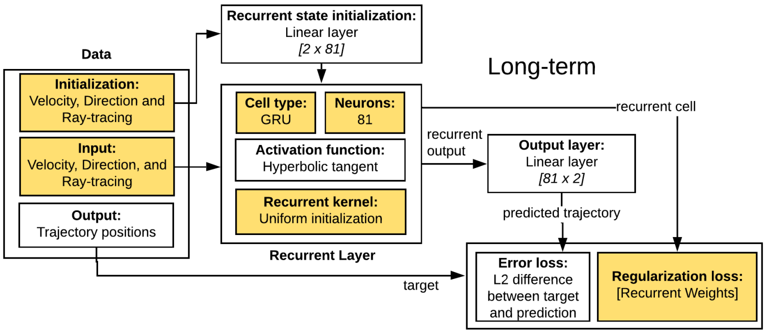
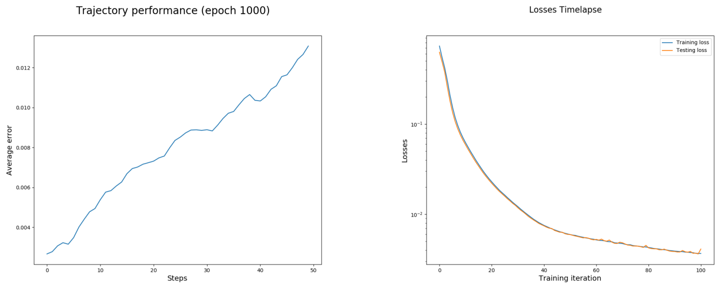


Figure F.5: **Model architecture**
The model consists of the GRU recurrent cell, uniform recurrent initialization, ray-tracing and default input, 81 recurrent neurons, and recurrent regularization.



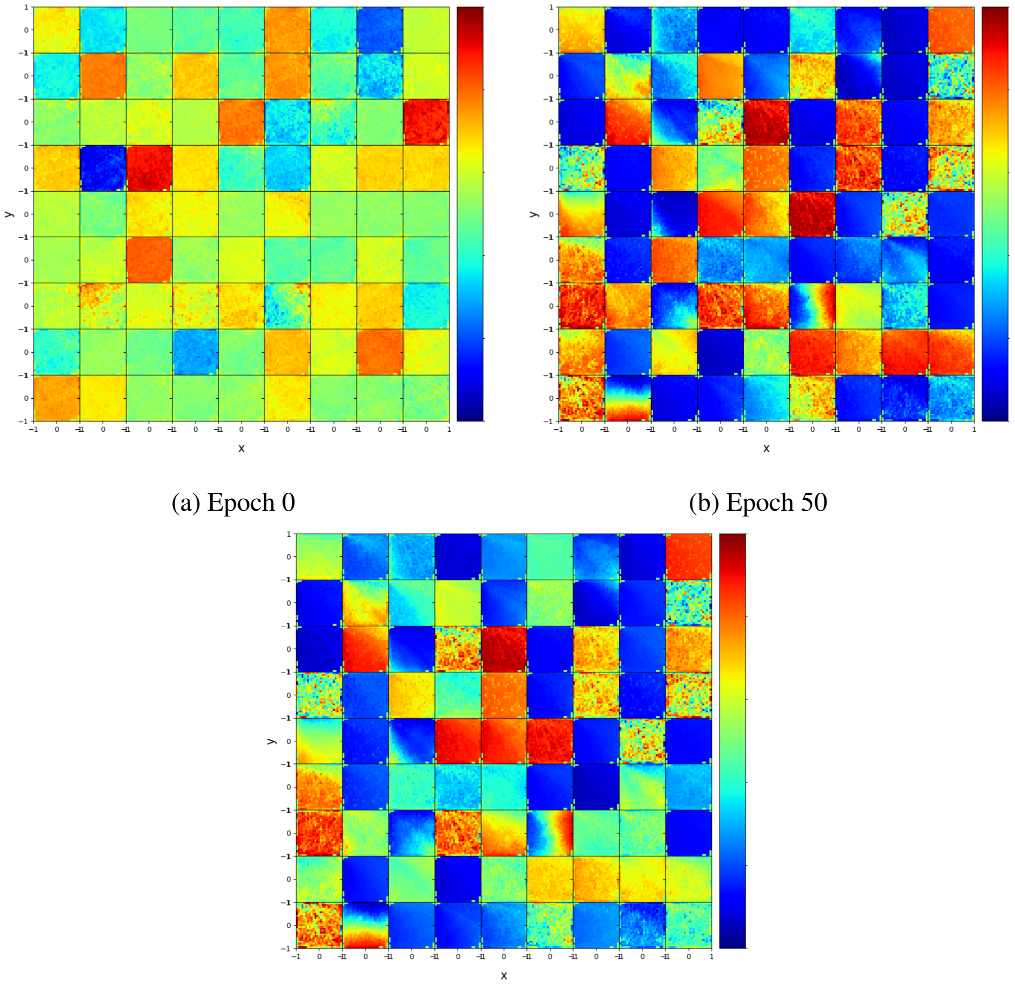
(a) **Average performance plot**
The time-dependent loss is calculated using the average loss for each trajectory time-step. Showing no initialization problems and linearly increasing prediction error over time.

(b) **Loss plot**
The loss plot visualizes the trained loss differences between training (blue) and testing (orange) error. The loss optimization stagnates around epoch 50.

Figure F.6: **Repeated combined experimental results**
The representation evolves until the loss optimalization stagnates. The representation saturates around epoch 500, and only minor changes to the representation develop at epoch 1000.

The loss plot in Figure F.6 shows close to optimal performance. The average performance plot (Figure F.6a) shows proper initialization and linear error accumulation through time. Both aspects are optimal, especially for a path-integration context. Additionally, the loss plot (Figure F.6b) converges quickly to an appropriate solution. The performance improves only slightly over the last 75 epochs. Both the training and testing error follow closely; thus, the underlying representation can generalize between the training and testing set.

The spatial representation evolves quickly in the first few epochs, but representation



(a) Epoch 0

(b) Epoch 50

(c) Epoch 100

Figure F.7: **Repeated combined experimental results**
Each plot cell visualizes the average spatial activity for an artificial neuron. The representation evolves until the performance stagnates. The representation saturates around epoch 50, and only minor changes to the representation develop at epoch 100.

stagnates at the end of the training process (Figure F.7). After initializing the network and training for one epoch, the neurons already show some spatial and directional specialization. Once again, the representation consists of noisy, stochastic, and linearly activated neurons. Each archetype presumably serves to represent the directional and spatial domain during path integration necessary for optimizing the task.

Appendix G

Experiment Baseline Configurations

The configurations in Table G.1 highlight the baseline configuration for earlier work, the factor experimentation process, and entorhinal performance experiments. Sharing the full details for experiments are essential for improving reproducibility [450] of future research.

exp_name	**DeepMind**	**CuevaWei**	**Factors**	**Entorhinal**
num_epochs	30	150	10	10
num_iter_per_epoch	10000	1000	1000	100
episode_length	100	100	100	100
learning_rate	0.00001	0.00001	0.001	0.001
batch_size	100	100	100	100
num_trajectories	1000000	100000	100000	100000
ray_tracing_resolution	50	50	50	50
dimensionality	2	2	2	2
hidden_size	128	100	25	32
behavior_dataset	free will location	free will	free will location	free will location
model_data	DeepMind	default	default	default
recurrent_cell	LSTM	CTRNN	LSTM	GRU
input_initialization	default	normal	default	default
recurrent_initialization	default	orthogonal	default	default
recurrent_activation	default	default	default	default
output_layer	intermediate	linear	linear	linear
error_loss	softmax	default	default	default
regularization_loss	LinearWeights	Metabolic	none	none
regularization_constant	0.00001	0.0001	0.0001	0.001
dropout_type	default	none	none	none
optimizer	DeepMindRMS	default	default	default
gradient_clipping	0.00001	0.0001	0.001	0.01

Table G.1: **Hyper-parameters and network setup for experiments**
The four base configurations above describe the configuration details for the replicated related work, experiments with different factors, and tests determining the effect of entorhinal-like representation during path-integration.