UiT The Arctic University of Norway

Faculty of Science and Technology

Autonomous Navigation

in (the Animal and) the Machine.

Per Roald Leikanger

A dissertation for the degree of Philosophiae Doctor

March 2022

Seek not to follow
in the exact footsteps of
the wise men of old;

seek what they sought.

Bashō Matsuo

Abstract

Understanding the principles underlying autonomous navigation might be the most enticing quest the computational neuroscientist can undertake. Autonomous operation, also known as voluntary behavior, is the result of higher cognitive mechanisms and what is known as executive function in psychology. A rudimentary knowledge of the brain can explain where and to a certain degree how parts of a computation are expressed. However, achieving a satisfactory understanding of the neural computation involved in voluntary behavior is beyond today's neuroscience. In contrast with the study of the brain, with a comprehensive body of theory for trying to understand system with unmatched complexity, the field of AI is to a larger extent guided by examples of achievements. Although the two sciences differ in methods, theoretical foundation, scientific vigour, and direct applicability, the intersection between the two may be a viable approach toward understanding autonomy. This project is an example of how both fields may benefit from such a venture. The findings presented in this thesis may be interesting for behavioral neuroscience, exploring how operant functions can be combined to form voluntary behavior. The presented theory can also be considered as documentation of a successful implementation of autonomous navigation in Euclidean space.

Findings are grouped into three parts, as expressed in this thesis. First, pertinent *background* theory is presented in Part I – collecting key findings from psychology and from AI relating to autonomous navigation. Part II presents a theoretical *contribution* to RL theory developed during the design and implementation of the emulator for navigational autonomy, before experimental findings from a selection of published *papers* are attached as Part III. Note how this thesis emphasizes the understanding of volition and autonomous navigation rather than accomplishments by the agent, reflecting the aim of this project – to *understand* the basic principles of autonomous navigation to a sufficient degree to be able to recreate its effect by first principles.

Contents

List of Figures

Acknowledgements

The desire to better understand intelligence has always been a part of me, a drive that was nourished by the input received at the Kavli Institute for Systems Neuroscience in Trondheim. Being a student at your magnificent research group, as endorsed by the Nobel comity the following year, deepened my interest in neuroscience and the biology of autonomous navigation. Thank you for introducing me to the intricacies of the navigating mind, May-Britt.

After formally starting this project, I've traveled the world; Svalbard, the French Alps, Sunnmøre, Toronto, Montreal, Alberta, and finally back to Tromsø. Many thanks to my first mate on this journey, Bjørn Batalden, for moral support and guidance. Your human compassion, and possibly your abilities as a submarine navigator, have been influential in this convoluted and obfuscated journey. My colleagues at UiT, the people I've interacted with in Grenoble, the inspiring people and community at the Kavli Institute in Trondheim, my friends in Canada, thank you for your inspiration and support. The Norwegian research school in neuroscience has been important in helping me out of an Arctic (scientific) isolation.

My visit to Montreal and Toronto during the summer of 2019 was defining for the second half of this project. Special thanks to Doina Precup, for giving me hours of her time as director at DeepMind Montreal to discuss the importance of my epsilon-free exploration; to Katya Kudashkina for inspiration and for inviting me to present this work on the Vector Institute of Artificial Intelligence in Toronto; to Peter Wittek for his interest in the project and inviting me to Toronto and becoming my friend before passing away in the Himalayas; thank you all for your interest and inspiration before even moving to Canada.

Rich, your invitation as a vising researcher at the RLAI lab for 2020 boosted my belief in the neoRL approach. Balancing on the edge between an introspective madness of academic solitude and acknowledging my own accomplishment, your interest in this project consolidated my faith in neoRL. Thank you for late discussions on the nature of consciousness, for your deep insight into research methodology, and your inquisitive mind; you are an inspiration, an inspiration I hope to bring on to other students of computational cognition.

From May-Britt Moser to Richard Sutton, this project has spanned a decade, two continents, and more solitude that I've appreciated; to all my friends and closest family that have supported me or endured my endless monologues of the nature of mind: Thank you.

Chapter 1.

Introduction

> One does not set out
> in search of new lands
> without consenting to
> go beyond sight of any shore.
>
> *André Gide*, Les Faux-monnayeurs

According to philosopher John Stuart Mill, an individual's *autonomy* reflects to what extent a person acts according to his/her own values, desires, and inclinations [50]. Human autonomy is an effect of higher cognitive processes referred to as *executive function*, indicating the difficulty of achieving autonomy in technology. Before using the expression autonomy in technology, the expression must be operationalized. First, autonomy should involve self-governed freedom of choice; autonomous behavior should originate from the decision agent's personal experience or desires. Excluding all aspects of external control, autonomous technology should not require set-points or algorithmic input. All forms of direct commands, including programmed policies or remote control, should disqualify technology from being autonomous. Second, autonomous technology implies a decision process that defines choices based on personal experience rather than inherited rules or programmed reflexes. Genuine autonomy comes from the autonomous agent's experience rather than being a product of rules or predefined algorithms.

Definition 1 *Autonomy requires freedom of choice and absence of external control; the control of an autonomous agent comes exclusively from the agent's experience, desires, and inclinations.*

Most problem-solving algorithms involve accurately defined recipes, effectively controlling the operation by predefined algorithms. Some operations are preferably executed by automatic algorithms; for example, robots on an assembly line – performing their work with quick precision and in perfect synchrony – would be difficult to improve beyond cybernetic control.

For predefined tasks, limited operational domain, or in predictable environments, algorithmic control can execute operations in what *appears* to involve autonomy. However, intricate tasks for comprehensive operations can be difficult to solve by algorithms.

Navigation, from *navis* "ship" and root of *agere* "to set in motion", deals with the planning and effectuation of trajectories in Euclidean spaces. Euclidean geometry has been an important tool for navigation for more than two millennia, initially for the planar geometry involved in sailing, later extended to general N-dimensional[1] spaces [19]. Similarly, *navigation* involves a more general concept than maritime displacement.

Definition 2 *Navigation is the planning and realization of transitions in Euclidean space.*

Skilled navigation requires knowledge on how best to achieve an objective. A navigational task could, for example, be concerned with how to move an end effector of a robot to a target configuration. In this project, navigation is further separated into two equally important aspects: *category* 1 *navigation* is concerned with the effectuation of a command or planned path in some Euclidean space. In the maritime context, this aspect of navigation has traditionally been the responsibility of the crew. *Category 2 navigation* is concerned with planning and executive decisions for navigation; reactive re-planning and generating set-points for a route, and executive control according to objectives, are responsibilities of the commanding officers on a ship. Whereas category 1 navigation is generally concerned with the execution of a plan, category 2 navigation is concerned with the development and revision of planned paths toward the objective. Both category 1 and category 2 navigation are required for autonomous navigation.

Autonomous navigation is a fundamental ability for any living entity. Spatial navigation is essential for locating food, shelter, a partner – or family, thus prolonging one's life or gene pool. Recent reports in theoretical neuroscience have further linked mechanisms of Euclidean representation with ideas and mental concepts [16], implying the importance of navigation in problem-solving. Three aspects appear to be crucial for the evolution of navigational capabilities. Navigation must be general; learned elements should be applicable to new situations; policies acquired in safe situations should be applicable to new environments or while under distress. Navigation must be dynamic, allowing the extension or adaptation of previous knowledge to a changing environment; knowledge acquired from one situation should be generalizable to related situations. Navigation must be efficient; problem-solving should handle unseen situations and quickly learn to solve new tasks. Accomplishing all three properties in adaptive technology appears difficult, making autonomous navigation by technology a challenging task.

[1] $N \in \mathbb{N}$, the set of natural numbers/ positive integers, i.e. $1, 2, 3, ...$

Definition 1 and 2 disqualify most technology from being autonomous. Certainly, several high-tech solutions can *appear* to be autonomous, performing a task exactly as specified by some human programmer. Half a century ago, such technology would have been referred to as an *automatic* solution. With the term automatic becoming old-fashioned, researchers in 2022 tend to refer to similar solutions as being autonomous. Technology companies claim to have autonomous solutions, a claim that is difficult to question due to the secrecy involved in proprietary technology.

> "Automated" connotes control or operation by a machine, while "autonomous" connotes acting alone or independently. Most of the vehicle concepts (that we are currently aware of) have a person in the driver's seat, utilize a communication connection to the cloud or other vehicles, and do not independently select either destinations or routes for reaching them. Thus, the term "automated" would more accurately describe these vehicle concepts. [91]

It is important not to confuse algorithmic complexity with autonomy; robot control and path planning can form complex sequences through algorithmic control, but always according to rigorous mathematical models or by pre-programmed rule sets. Although impressive performance and automatic effectuation of a task is great engineering, such solutions appear to be difficult to extend beyond lab conditions or simple challenges.

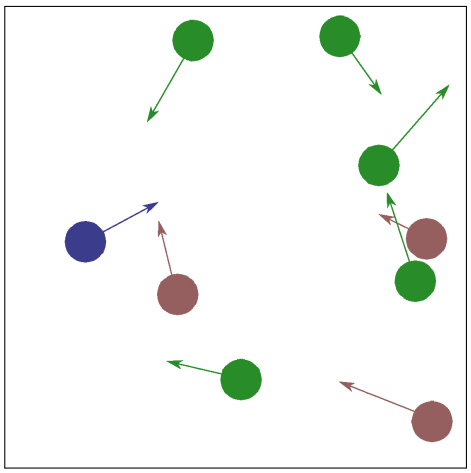


Figure 1.1.: **Maritime navigation can be quite complex**. Handling a number of desires, here represented in green, and aversive objects, represented as red dots in the Euclidean space, can be a challenging navigation challenge. Controlling the self, represented as a blue dot, among a multitude of external objects with different position and speed vectors, in a continuous environment with inertial dynamics, is the environment considered in this research; *the WaterWorld environment*[70]. Figure from [42]

1.1. Problem statement

Autonomy, according to definition 1, requires an ability to acquire knowledge from experience during the lifetime of an agent. Navigation, according to definition 2, requires that such knowledge is applicable to Euclidean spaces. *Autonomous navigation* requires efficient problem solving and learning in real-time while navigating. The aim of this project is to understand[2] the basic principles involved in autonomous navigation, to a sufficient degree to recreate in technology – a process that can further deepen our understanding of behavioral autonomy. To achieve this level of understanding, the aim can be divided into the following research objectives.

1. Identify basic principles underlying autonomous navigation from the psychology of learning and neuroscience of navigation.

2. Explore methods from the field of AI to find concepts applicable for online autonomy.

3. Design and test an agent based on results of research objective (1) using the findings from objective (2) to explore the basic principles involved in autonomous navigation.

In the context of this work, we consider the autonomous navigation of a general Euclidean space as the ultimate expression of autonomy. Hence, our definition of autonomy expresses that autonomous operation requires a behavior that is governed by an entity's own experience. Digital agents must be regarded as unique entities. Limiting the extent of an agent to the duration of a single run, definition 1 requires that autonomous navigation emerges from experience acquired during real-time execution. Further, the considered task for the agent should be the navigation of Euclidean space – not discrete or task-specific challenges. Experiments should highlight differences between implementations differing only by the examined mechanism – to deepen our understanding and theory required to implement autonomous technology, as well as achieving a better understanding of the principles underlying voluntary behavior.

[2] "What I cannot create, I do not understand" – Richard Feynman's blackboard at the time of his death[1]

1.2. Thesis overview

The thesis is divided into three parts, reflecting progression toward the main research aim. Part I considers background theory from psychology, in Chapter 2, and from adaptive algorithms by reinforcement learning in AI, in Chapter 3. These two chapters can be considered as a fulfillment of research objective 1 and 2, respectively. Part II presents how distributed latent learning is possible for digital decision agents, establishing operant desires by general value functions – as inspired by Skinner, and how the value function can be extracted from latently learned behavioral maps – as inspired by Tolman. Chapter 4 revisits the fundamentals of RL, before exploring how the persistence school – the alternative understanding to temporal difference learning – can be applied in RL. Chapter 5 concludes this thesis with a discussion of neoRL findings according to the aim of the projects. Part III presents three manuscripts, representing milestones in the developing understanding on the principles of autonomy. Paper A illustrates the importance of a decomposed value function, and how separation of concerns can be considered together with principles for neural representation to form autonomous navigation. Paper B explores the validity of autonomous navigation, by experiments for considering whether the presented framework is general, compositional, and efficient across modalities. Manuscript C considers deeper purposive graph structures, demonstrating how neoRL agents benefit from being guided by projections of desire from other neoRL nodes.

Part I.

Background

Chapter 2.

Psychology and navigation – true autonomy

it's very difficult
  to find some way of defining
rather precisely
  something we can do
that we can say that a computer
  will never be able to do

Richard Feynman, asked "Can machines think?"[22]

The autonomy and *free will* of an individual are central in most religions. Psychology, from ancient Greek, psyche (soul) + lògos (explanation), can be viewed as the study of (changes in) autonomous behavior. The study of the mind is difficult to approach through scientific methods, which resulted in several model explanations in the early twentieth century. Some of these models did not survive the test of time, e.g., explaining personal characteristics by the external shape of the skull – see figure 2.1. However, these early attemtps marked an important shift from considering the mind as a religious matter to being subject to scientific inquiry. First, the interest in measuring and observing the properties of the mind allowed for a deeper understanding of behavior. Since autonomy was no longer considered to come from an atomic soul, ultimately governed by God, researchers could explore the basic principles of learning without offending the church. Second, a new scientific movement opened for animal experiments while researching personal autonomy. Darwin's *On the Origin of Species by Means of Natural Selection* (1859) established how man and creature are part of a continuum [17], making research on animal behavior relevant to the understanding of human personality and autonomy. With the beginning of the twentieth century, functionalist and behaviorist psychologists started to study the change in animal behavior from the perspective of experience. Stimulus-response (S-R) experiments on simple animals allowed researchers to quantify changes in behavior. Considering man as part of a continuum rather than separate from nature allowed for deeper insights into human

autonomy based on animal experiments[62].

This chapter presents an overview of the history of psychology, with emphasis on the expression of personal autonomy and the neuroscience of navigation. First, Section 2.1 presents how functionalist biology evolved into behaviorism and eventually cognitive psychology. Section 2.2 introduces the work leading to the 2014 Nobel prize. The discovery of place cells and similar cell types representing one's navigational state has become an important part of modern neuroscience research. Neural representation of Euclidean space though, is involved in more than spatial navigation. The brain's navigation system has been implied in creative problem solving, inductive reasoning, and intelligence [11]. This chapter summarizes the key elements on the biological basis of learning and navigation, from the precognitive psychology of learning to the neuroscience of navigation. The theory presented in this chapter plays a crucial role in Part II Contribution.

2.1. Adaptive behavior in psychology

William James (1842-1910) was an early pioneer of functional psychology. As a professor at Harvard, James established the course *The Relation Between Physiology and Psychology* in 1875. "Psychology is the science of mental life, both in its phenomena and conditions" [31]. James' *Principles of Psychology* influenced functional psychologist John Dewey (1859-1952), who was among the first to consider behavior in light of evolution[1]. John Dewey was interested in inherited reflexes, automated policies where the animal reacts according to the whole situation it is in. According to Dewey, claiming that stimuli S would always cause the same physiological response, independent of the environment, would be a simplification. William McDougall (1871-1938) expanded Dewey's theories to *reflexive behavior* and developed stimulus-response (S-R) connections for behavior. Note that the Markov state, introduced in Section 3.1, can be viewed as an interpretation of the Dewey state. While McDougall considered adaptive policies across generations, the Jamesian school of functional psychology focused on *learning* as adaptive behavior according to an individual's experience. James' student, *Edward Lee Thorndike* (1874-1949) claimed that the learning process had strong similarities with the process of evolution. Referring to acquired S-R connections as conditioned reflexes, Thorndike explained learning by what he referred to as the *law of effect*, whereby a positive reinforcer increases the association between a situation and the taken action, whereas negative feedback diminishes this association. Both McDougall's S-R reinforcement principles and Thorndike's law of effect are effects of connections between stimuli and the outcomes of the stimuli. Where McDougall claimed that policy search is

[1] Darwin published *On the origin of species* in 1859

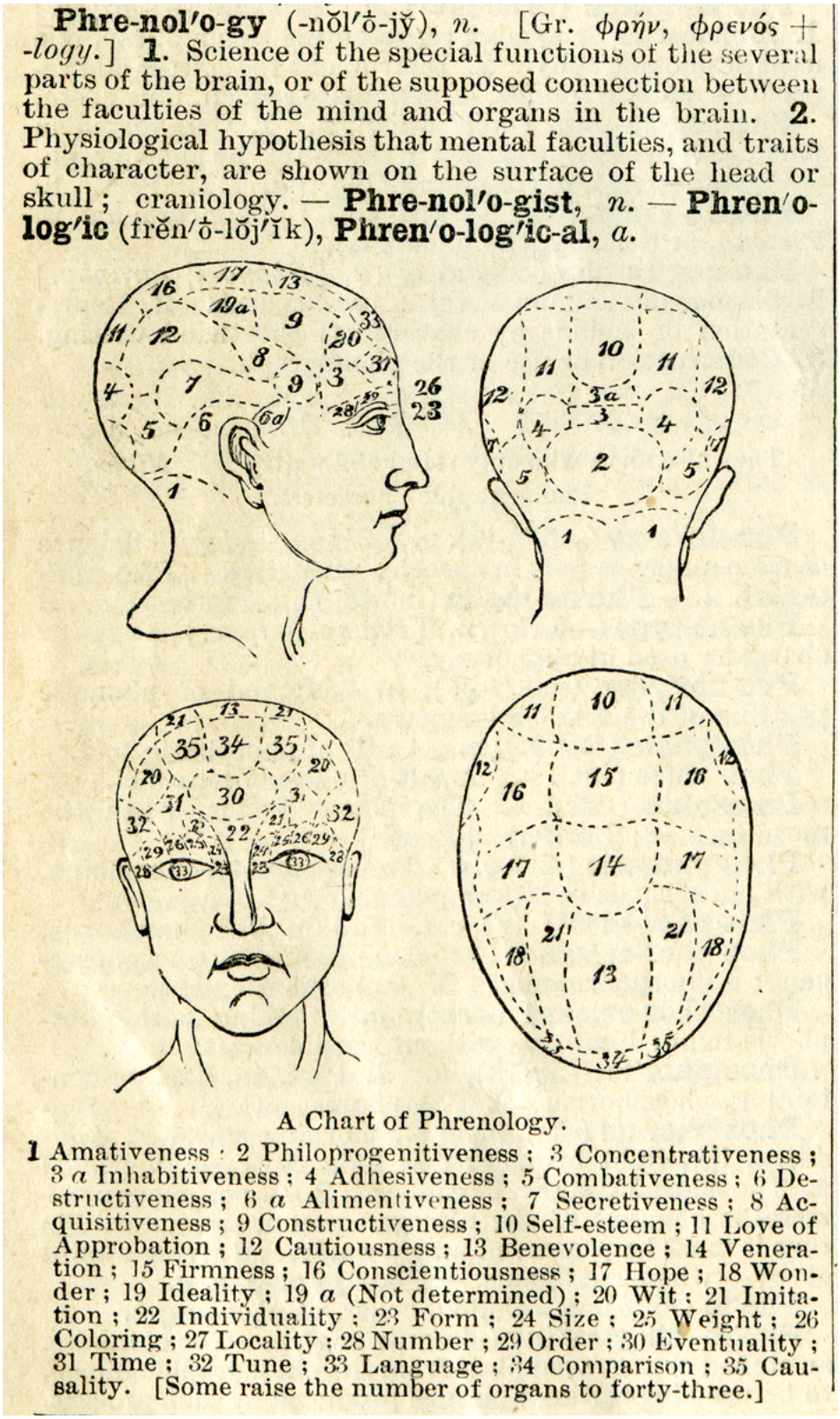


Figure 2.1.: **Phrenology, an early attempt on explaining behavior – ca. 1895.**
(figure from WikiMedia Commons)

an effect of evolution, Thorndike stated that reinforcement could happen internally in the individual such that reinforcement of conditioned stimuli is a learning process [62, 69, 72].

2.1.1. Behaviorism & reinforcement learning in psychology

"*Psychology as the behaviorist views it is a purely objective experimental branch of natural science. Its theoretical goal is the prediction and control of behavior*"[85]. In his 1913 article, John B. Watson (1878-1958) campaigned for strengthening psychology as a science by adopting methods from natural sciences. Only observable and measurable qualities would be reported in the new science, excluding introspection and subjective methods from earlier psychology research. Watson wrote: "Speaking overtly or to ourselves (thinking) is just as objective a type of behavior as baseball" [86]. Via (the change in) animal *responses* after stimuli, behaviorists could objectively measure the dynamic mind. Responses could either be explicit, expressed by directly observable qualities in the external behavior of the animal, or implicit, expressed as visceral movement, glandular secretions, or neural activity. Watson's animal research considered physiological responses and explicit actions, including both Thorndike's physiological law-of-effect and learning for voluntary actions [62]. Despite the stated goal of reducing all behavior to S-R connections, the ultimate objective of behaviorism was to understand the mechanisms involved in human behavior [62].

In a public debate between McDougall and Watson, McDougall criticized behaviorism for being too deterministic – not leaving room for free will or voluntary actions[87]. How could functionalistic views on reflexive behavior account for art, music, or unconditional love? Behaviorists attempted to explain all aspects of human behavior by reinforcement principles, concepts mainly used as models for the formation of instincts or reflexive behavior. The use of relatively simple S-R connections, acquired by randomly encountered connections during a single lifetime, can hardly explain higher human cognition. Chapter 3 covers how similar reasoning applies to RL in AI, effectively limiting machine intelligence by RL to simple problems and problem-specific agents. Classical behaviorism was critiqued for being too simplistic, that reinforcement of simple S-R connections could not account for the complexities of human behavior.

Burrhus Frederic Skinner (1904-1990) represented a renewal of Watson's behaviorism. Skinner's behaviorism was devoted to the study of responses; concerned with describing rather than explaining behavior, Skinner could investigate deeper S-R structures. With a focus on how an agent operates on an environment for achieving an effect, *operant conditioning* considers deeper $S^D - R - S^R$ sequences[2]. The *discriminative stimuli* S^D defines the relevance

[2]Note for the computer scientist: expressed with terms from RL in AI, Skinner used discriminator S^D to denote a pre-condition (one aspect of agent state s), the animal reflex R as the agent's action a, and S^R as reward R.

of the link, i.e., the Dewey-state that relates to the conditioned response. The conditioned reflex R denotes the response or action of the animal associated with the operant link, and S^R the conditioning signal responsible for changing behavior. The response can be positively reinforcing, increasing the drive for choosing action R under antecedent state S^D. The response can be negatively reinforcing, decreasing the drive for choosing action R under antecedent state S^D. Since the conditioning signal S^R comes after the conditioned reflex R, we say that conditioning works on a link that operates toward an objective. Operant conditioning allows for differentiating behavior, better representing Skinner's experimental findings [72].

2.1.2. Purposive Behaviorism & Cognitive Maps

Edward C. Tolman (1986-1959) found the reflex model in classical behaviorism to be too simplistic for explaining complex behavior. Tolman questioned the assumption that simple $S-R$ connections could account for complex behavior, instead proposing a richer $S-O-R$ model. The internal state of the organism, O, allowed Tolman's model to explain observed differences in an individual's behavior; a hungry animal would be more motivated for reaching food than right after feeding – effectively changing the animal's behavior. Driven by the *purpose* of feeding, the animal would activate latent knowledge on how to achieve satiety. Stateful mechanisms for behavior allowed for more accurate models of animal behavior. Tolman believed that the animal traversed the maze for a reason, to achieve something – reflecting purpose rather than conditioned reflexes as proposed by reinforcement learning and classical conditioning[72, 76].

Purposive behaviorism considers learning and motivation to be equally important for behavior. In Tolman's early experiments, Tolman found that problem solving was greatly affected by the motivation of the animal. In a maze experiment, Tolman and Honzik (1930) measured the time it took for rats to escape a maze for three different test groups. One group was rewarded with food outside the maze, resulting in better performance than a second group that did not receive any extra incentive other than escaping the maze. The third group did not receive any reward until day 11, and from then on started to follow the same reward schedule as the regularly rewarded group [77]. Tolman's results are presented in Figure 2.2. All groups learned how to escape the maze, possibly motivated by the auxiliary reward of escaping the discomfort of being in a maze. The regularly rewarded group performed better than the two other groups when unrewarded. When later motivated by food, however, the rats in the initially unrewarded group started to perform better than all other subjects. Classical behaviorism was unsuccessful at explaining these results. Tolman proposed a mechanism he called *latent learning*, a learning process that existed independently of external reward [76]. By considering behavioristic systems as information processing systems with an internal state,

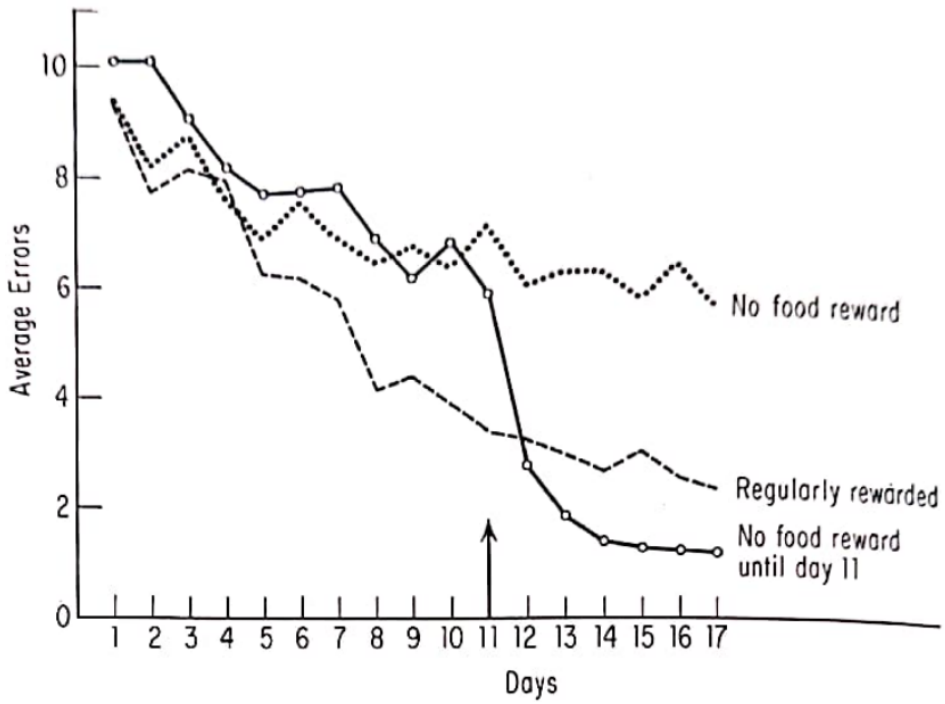


Figure 2.2.: **Reward is more important in forming behavior than for learning** Evidence for latent learning by Tolman and Honzik (1930). (After [78] according to [14]).

guided by an internal representation, Tolman advanced behaviorism beyond considering functionalistic S-R mechanisms into seeing stateful, cognitive entities [62]. When motivated by food, the animal could extract purposive behavior according to internal knowledge of the world – in the form of a *cognitive map* for solving a task [76].

2.2. The neuroscience of cognitive maps

The 1906 Nobel price in physiology and medicine was awarded Santiago Ramón Y Cajal for what resulted in the neuron doctrine; that all behavior originates from a large number of cells with signaling capabilities [57]. These specialized cells, *the neuron*, are surrounded by a lipid bilayer membrane that is impermeable to electrons. The electrochemical potential across the membrane, the neuron's *membrane potential*, is sustained by active ion pumps. See Figure 2.3 for a schematic representation of the parts of the neuron. Most neurons have four distinct functional parts; the *dendrite* is generally where input connections to the neuron are located; the *soma* of the neuron integrates excitatory and inhibitory input; the *axon* conduct processed information, potentially across large distances; the *synapse* propagates information onto the next neuron. The neuron is the processing unit of the brain, processing and conducting information in large neural networks by spatio-temporal integration in the soma according to synaptic input.

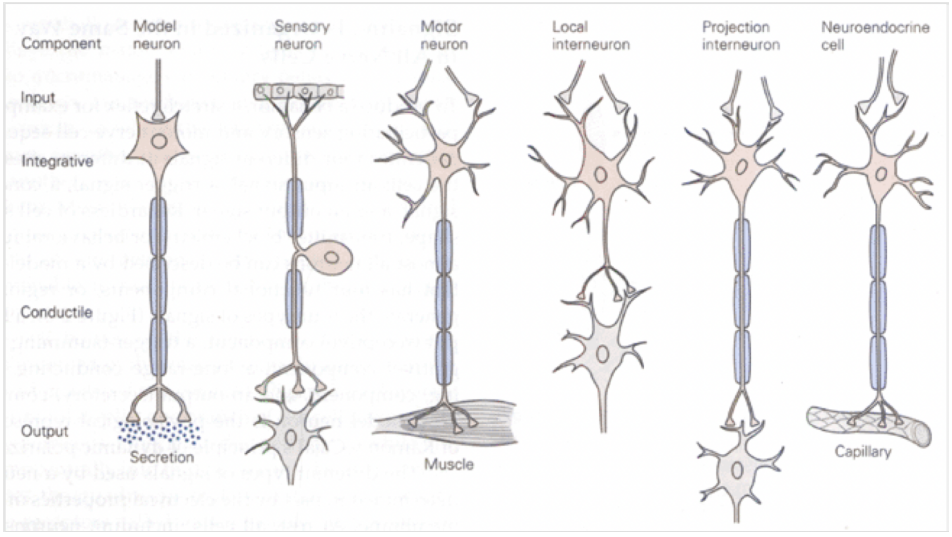


Figure 2.3.: **Most neurons have four functional elements in common: an input component, a trigger or integrative component, a conductile component, and an output component.** The four functional components are referred to as *Dendrite*, *Soma*, *Axon*, and *Synapse*. A large variety of neurons exist [33].

Specialized ionic gates in the neural membrane can be opened by neurotransmitters, depolarizing the neuron's membrane potential. When the potential at the axon wall is sufficiently depolarized, ionic gates temporarily open along the axon. The resulting sweeping surge of depolarization is known as the *action potential*. When the action potential reaches a synapse, *neurotransmittors* are released into the synaptic cleft between this neuron and the next. Some neurotransmitters activate voltage-gated channels, with a depolarizing (excitatory) or repolarizing (inhibitory) effect on the post-synaptic neuron; other neurotransmitters initiate longer-lasting changes to the post-synaptic neuron signaling properties. When the post-synaptic neuron is sufficiently depolarized, it fires an action potential, propagating the signal on through that neuron's output connections [33].

Eric Kandel later (1965) demonstrated how synaptic connections can change as a function of activity [34], resulting in the current view that learning and memory originate from synaptic plasticity. A persistent rewiring happens continuously as a function of neural activity, resulting in short-term or long-term change in a synapse's efficiency for eliciting response in the post-synaptic neuron [5]. Short-term changes could have significance in neural computation, whereas long-term synaptic remapping is the underlying mechanism of *learning*. Computation by networks are said to be *sparsely coded*, a computational scheme where individual nodes of the network have but a fraction of any computation [23]. The state of the network at any given time is both the method and the output of computation – there is no clear distinction between what is the result of neural computation and what belongs to its computational state. Although individual neurons have a minor effect on problem-solving, the collective pattern of activity and the changing connections between neurons is seen as the origin of behavior. Any persistent remapping could provoke beneficial patterns to emerge, e.g., the activation of specific nodes as a result of external phenomena. Specific neurons have been identified that directly reflect external properties of the environment, properties that are intimately involved in autonomous navigation [13].

2.2.1. Neural representation of Euclidean state

Central to all navigation is knowledge of one's current (navigational) state; knowledge of one's location, orientation, and heading, and for objects that can block or otherwise affect the path, is fundamental for proficient path-planning. Separate considerations of the navigational state can be mathematically formalized as vectors in a corresponding Euclidean space. Euclidean geometry was first used for representing locations in the physical world and later extended to involve general N-dimensional spaces [21]. With a set of axioms founded in "measuring the world", Euclidean geo-metry allows for measuring and planning displacement. Vectors represent the distance and direction from some origin to some point in that Euclidean

space. For example, the Cartesian vector $\vec{a} = [1.0, 3.0]$ can represent a point in a plane ($2D$), one unit size from the origin along the first dimension, and three units along a second dimension. A polar coordinate $\vec{a} = [r, \phi]$ defines a point with distance r along a line with angle ϕ relative to some reference. The difference between Cartesian representation and polar coordinate representation is illustrated in Figure 2.4. Euclidean geometry is a mathematical framework for representing navigational states, using vectors to define location, orientation, or translation in an environment.

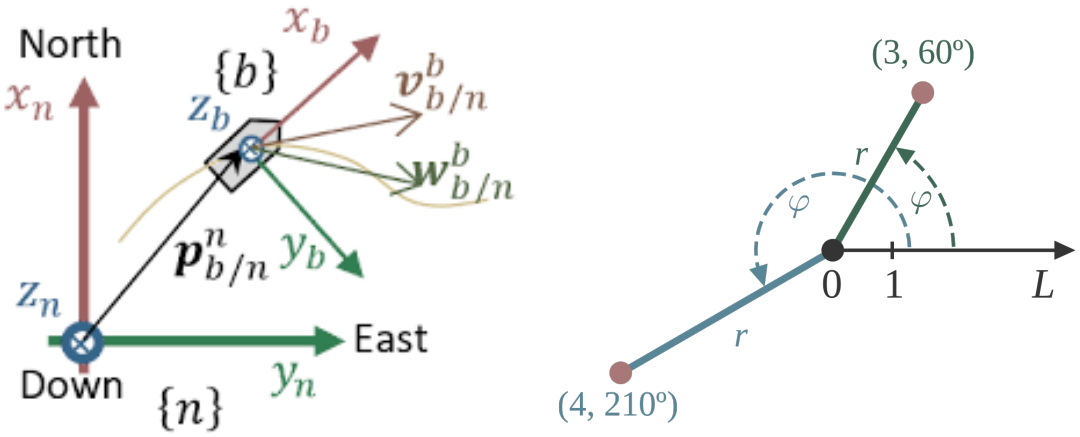


Figure 2.4.: **Euclidean information can be represented in polar or Cartesian coordinates.** [a] Cartesian coordinate systems according to different reference frames. Information represented in allocentric reference frame North/East (x_n, y_n) or egocentric Forward/Starboard (x_b, y_b). [b] Vectors expressed in polar coordinates, e.g., the point with polar coordinate (3, 60°) has distance 3 units along the axis with angle 60°.

The importance of Euclidean geometry for navigation – and the importance of navigation for basic survival – implies that the brain has an effective representation for Euclidean information. Thinking in terms of Euclidean geometry is beneficial for two reasons: First, representing spatial information as vectors facilitates discussion and allows a mathematical analysis of navigation. Vectors in the Euclidean representation must be according to a reference frame. An *allocentric* representation is a vector with at least one parameter represented according to an external reference frame. For example, "two steps north and one step west" represents relative displacement according to the Cardinal directions. An *egocentric* representation measures parameters relative to the current situation of the agent. For example, "two steps forward and one to your left" represents relative displacement measured according to one's current parameter configuration. Second, considering independent

navigational concerns across separate Euclidean spaces facilitates discussion and analysis. Rather than approaching navigational challenges as monolithic and inseparable tasks, as indicated in classical behaviorism, independent representation of orthogonal information in simpler Euclidean spaces decreases the complexity of the operation. A maritime operation could, for example, involve (1) the final objective of the navigation, and (2) local obstructions and dangerous traffic. Decomposing the problem into separate concerns can simplify this navigational task compared to a monolithic approach. A decomposed state representation would be economical compared to monolithic approaches; the brain appears to represent its navigational state across multiple simultaneous partial representations [13] Neural representation of Euclidean state (NRES) can hold Euclidean information for various navigational modalities.

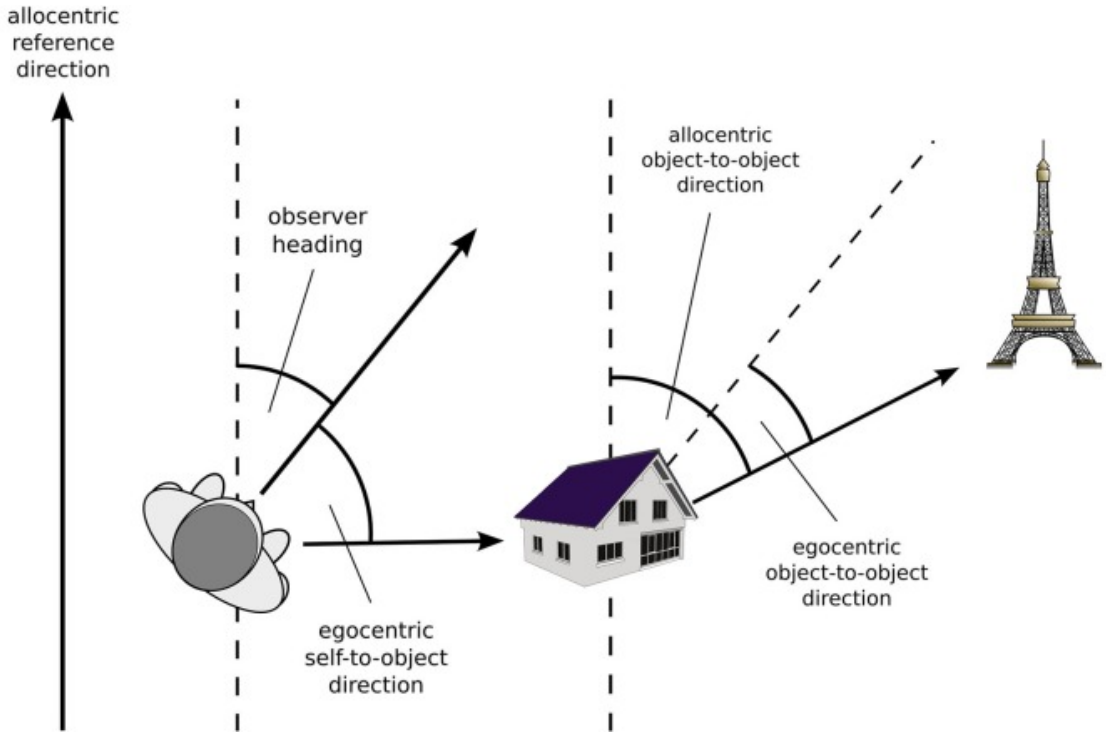


Figure 2.5.: **Illustration of allocentric and egocentric reference frame.** Direction and position can be represented in a egocentric or allocentric reference frame. The relative location between a house and the Eiffel tower can be given according to allocentric Cardinal directions (North-East) or egocentric reference ("over there"). (Figure from [89])

Navigational state – decomposed across multiple NRES

The first NRES cell to be explored was the *place cell*. O'Keefe and Dostrovsky (1971) observed that specific cells in the hippocampal formation responded to navigational conditionals on the animal's allocentric position in an environment. Place cell activation reflected navigational information; whenever the location of the animal corresponded to the receptive field of one place cell, this neuron exhibited a heightened neural activation [54]. The activation level of the place cell in terms of firing[3] frequency appears to be governed by a Euclidean conditional. Several other cell types have been reported by modern neuroscience, each representing a separate consideration of navigational state.

Definition 3 *Information represented as vectors can only hold information about the parameters involved in the constituting space; the* modality *of an NRES is the set of parameters involved in the mapped Euclidean space.*

The *NRES modality* of a place cell mapping is the *allocentric position of the mouse*. The activation pattern across the population involved in an NRES map forms the neural representation of this NRES modality. Other examples involve boundary vector cells [44] or border cells [65], responding to the location of borders or boundaries in the environment. The modality of head direction cells [71] corresponds to the current heading of the animal, and speed cells to the animal's current velocity [38]. Landmark vectors cells [18] and object vector cells [29] reflects the location of external objects. Note how the object vector cell is directly analogous to the radar in maritime navigation, representing objects in different allocentric directions and distances from the animal's position. A selection of allocentric NRES modalities from Bicanski and Burgess' comprehensive review paper on neural vector coding [13] is presented in Table 2.1.

Navigation can be concerned with absolute or relative states. First, navigating from one's current position A to objective B requires knowledge about one's allocentric position and heading. One's position is reflected by place cells [54]. The landmark vector cell encodes the location of landmarks around the animal, but as object + position conjunctive conditionals [18]. An animal's current heading is reflected in the head-direction cell, encoding the current allocentric orientation of the head [71]. Second, safe navigation requires knowledge of immediate dangers or objects that can block one's path. The object vector cell (OVC) represents the location of near objects [29]; the border cell represents insurmountable borders in a proximal position NRES modality [65]; the boundary vector cell represents any boundary in a distal position NRES modality similar to OVC encoding [44]. In contrast with the proximal

[3] Neuroscientists refer to the event of an action potential as a "firing" in the neuron; the firing frequency is a common measure for a neuron's activation level, reflecting the rate of transmission in its output synapses.

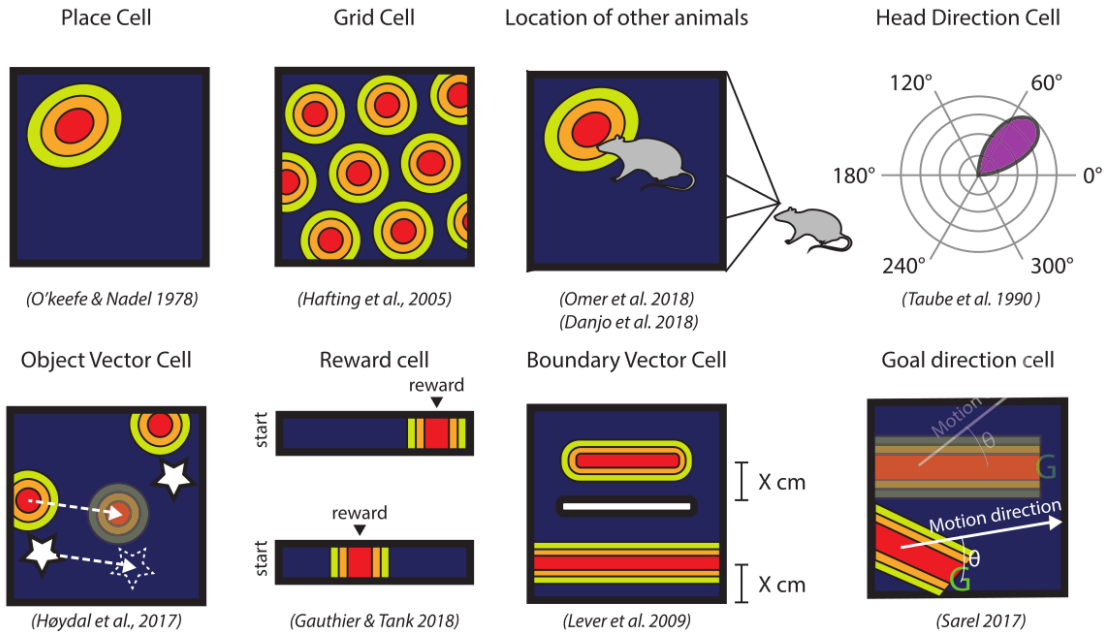


Figure 2.6.: **A selection of identified NRES modalities of importance for navigation**, with reference to the original publication. (Illustration adopted from [6])

requirement of border cells, OVC cells have receptive fields at a range[4] of distances. Additional NRES cell types are discovered every year; although it can be rewarding to recognize navigational information as NRES modalities from neuroscience, this project benefits more from identifying the basic principles behind neural representation of one's navigational state. Distributed state representation in the brain reflects orthogonal aspects of the navigational state as separate NRES modalities.

2.3. Discussion on the biology of autonomous navigation

In this chapter, we have reviewed how matters of the soul became a scientific discipline, a science devoted to understanding the foundation of biological autonomy along with theories from behavioral psychology and the psychology of learning, highlighting crucial aspects for agent autonomy. We have seen how Skinner's theories on operant conditioning, Tolman's concept of a cognitive map and latent learning, and recent findings from the neuroscience of navigation could better represent behavioral complexity than Thorndike's law-of-effect. The identified mechanism for neural representation of Euclidean space (NRES) has been essential in the development of autonomous agents in chapter 4.

[4]OVC respond to a range of distances; most OVC cells have a receptive field up to 30cm away in mice [29].

	Locality Repr.	Direction Repr.	Concern
Head-Direction Cell	-	allocentric	Head direction
Place Cell	allocentric	-	Agent position
Border Cell	allocentric	-	Location of borders
Boundary Vector Cell	[distance]	allocentric	Location of boundaries.
OVC	[distance]	allocentric	Location of objects
LVC	[distance]	allocentric	Location of landmarks

Table 2.1.: **Neural representation for different Euclidean spaces of importance for navigation**: Head-direction cell respond to the current allocentric angle of the head (1D). The place cell and border cell respond by the proximal allocentric location (2D). The remaining NRES respond to conditions that combine the current heading with distance to map 2D space, combined with auxiliary conditions (e.g., the existence of boundaries, objects, or landmarks at the location).

Psychology can well be considered as the study of human behavior and the formation and loss of personality – the expression of one's autonomy. Early researchers like McDougall, investigating reflexive behavior and instincts shaped by reinforcement, and Thorndike, considering similar mechanisms internally in the individual, were pioneers in the psychology of functional behavior. Although McDougall investigated the development of reflexive action and Thorndike considered learning within the individual, the primary difference between the two involves the model explanation for the mechanism of adaptation. McDougall researched inherited instincts, whereas Thorndike/Watson focused on adaptive behavior internally in one individual – what is referred to as learning in biology. The debate between early movements in behavioral psychology has relevance for today's behavioral AI, and can provide deeper insight into the methodological basis for RL in AI.

Two scientists of the *neobehaviorist* movement have received extra attention. First, experimentalist B. F. Skinner was mainly concerned with describing observed behavior – observing the rules of adaptive behavior rather than explaining their functional foundation. What Skinner referred to as operant conditioning, the reinforcement of operant behavior toward some objective, could inspire a mechanistic model of intrinsic rewards for RL agents. Achieving a goal can be rewarding in itself for the operant link. Second, the more theoretically inclined Edward C. Tolman introduced the concept of cognitive state for behavior. Tolman proposed that reward has more impact on motivation than on learning, resulting in a model Tolman referred to as purposive behaviorism. Claiming that reward affects motivation and behavior rather than learning, Tolman introduced concepts like *latent learning*, stateful execution, and *cognitive maps* to explain the full complexity of animal autonomy.

Electrophysical recordings have verified Tolman's hypotheses of cognitive maps for a range

of navigational modalities. One's absolute position, head direction, the relative or absolute location of others, or heading and distance to path-blocking obstructions are reported in recent neuroscience publications. Neural representation of navigational state appears to be distributed across separate concerns, each represented by NRES. Similar coding has been identified for other continuous parameters, like sound frequency in bird song [2] or conceptual space [6, 11, 16, 20] for human thinking. Neural representation of one's navigational state is expressed across several cognitive maps formed by NRES, demonstrating the importance of distributed state representation in the brain.

Chapter 3.

Adaptive algorithms and navigation

When a measure
becomes a target
it ceases to be a good measure

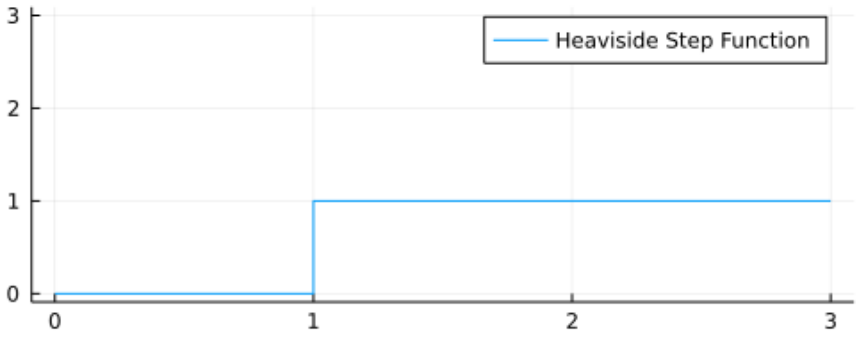
Goodhart's Law

Inspired by early neuroscience, Rosenblatt (1957) demonstrated how a one-layered computational graph can recognize simple patterns by what has become known as the *perceptron*.

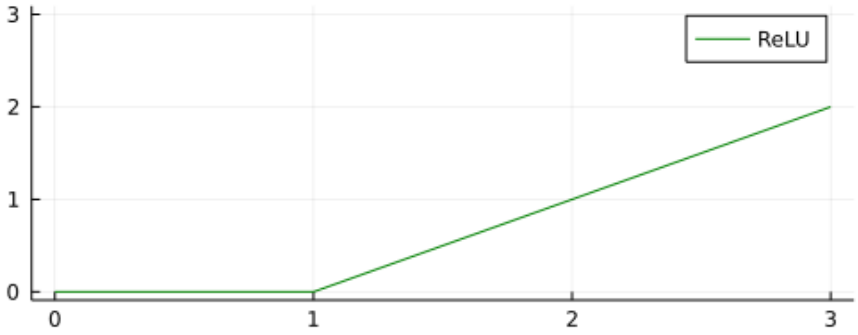
> "The proposed system depends on probabilistic rather than deterministic principles for its operation, and gains its reliability from the properties of statistical measurements obtained from large populations of elements."[59].

The original perceptron was inspired by the McCulloch-Pitts neuron model as an activation function, as shown in Figure 3.1a. Later iterations of the perceptron have used differentiable activation functions, allowing for deeper error signal propagation through the application of the chain rule [61]. *Backpropagation* allows for deeper networks[60], with multi-layered perceptrons (MLP) and what has later been referred to as artificial neural networks (ANN) or deep learning [61]. ANN could be considered a simple frequency-domain interpretation of biological neuronal networks [58]. These methods are referred to as *perceptron-class* adaptive filters in this text. A common activation function in deeper filters is the rectified linear unit (ReLU) function[24] – the integral of the activation signal in the original perceptron, as presented in Figure 3.1. Today's successor to the perceptron could be thousands of layers deep, requiring large amounts of labeled examples to categorize patterns or train estimators based on regression.

Behavioral autonomy requires more of the agent than what can be provided by perceptron-class adaptive filters. Although situation awareness is essential for understanding the world, perceptron-class AI would not be sufficient for behavioral autonomy; behavioral adaption



(a) The McCulloch-Pitts model.



(b) The ReLU activation function.

Figure 3.1.: **Two perceptron-class activation functions, the McCulloch-Pitts model and the ReLU activation function.** [a] The original perceptron activation function is an implementation of the McCulloch-Pitts neuron model. [b] The ReLU activation function can be seen as the integral of the original activation function (with a gradient defined by curve a).

requires an ability to behave according to earlier experience. The objective of reinforcement learning (RL) in AI is often summarized by the *reward hypothesis*[66].

> "That all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)."

According to the reward hypothesis, a stochastic decision process can emulate learning by means of an iterative search for optimal parameters in the stochastic decision process. Both RL and perceptron-class AI adapt according to external feedback, but while perceptron-class filters adapt according to supervised feedback and rote learning, the RL agent *reinforces* beneficial behavior as measured by the scalar reward signal R. Autonomy requires that the behavior of the agent adapt according to an agent's own experience, making RL a promising candidate for the journey toward autonomous navigation.

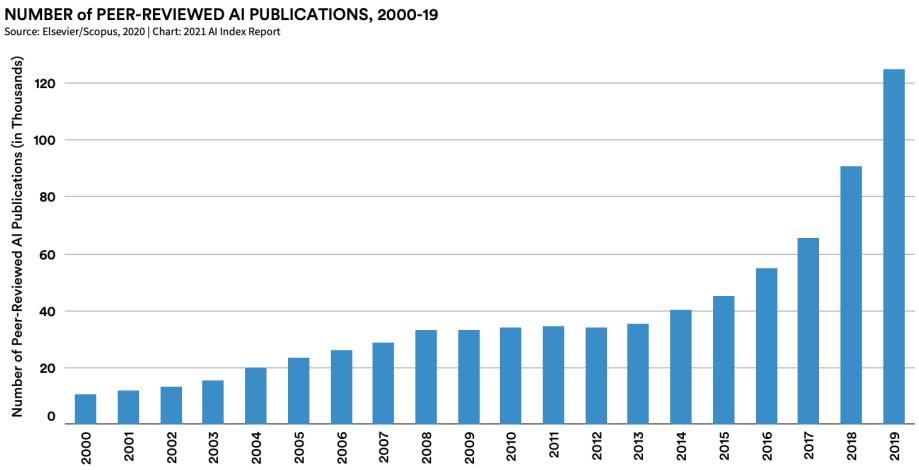


Figure 3.2.: The number of peer-reviewed AI publications, 2000-2019 [92]

Although AI publications represented 3.8% of all scientific publications worldwide last year[92], one could ask whether these reports represent great engineering rather than science. A substantial part of AI articles could be summarized as "our method *outperforms* the state-of-the-art in X" where X is some challenge claimed not to have any satisfactory solution. Constructing methods for data-driven filters can be impressive engineering. However, the otherwise data-driven research field can be criticized for being driven by examples of accomplishment rather than scientific vigor. This chapter presents a short overview of pertinent findings from computing science, with emphasis on distributed representation and computation that could allow for purposive mechanisms in AI.

3.1. Reinforcement learning in AI

A classical formulation of a sequential decision optimization problem is the *Markov decision process*(MDP)[8]. An MDP is a clear formulation of behavioristic theories as formalized by Thorndike; the desire for repeating a certain response is reinforced when followed by reward[75]. The learner and decision-maker of an MDP is called the *agent*. The agent interacts with a system, referred to as the *environment* in the classic RL literature. The MDP description requires a simple discrete-time formulation of the joint system comprised of the agent and the environment. At every time step t, the three aspects of an MDP are updated as shown in Figure 3.3. The agent's choice interacts with the environment as the *action* a_t, whereby the interaction at time t results in an updated environment state, s_{t+1}, and a possible reward signal R_{t+1} . Learning is expressed as a gradual improvement of the choices of the agent, selecting the best action from a revisited state. If a state-action pair eventually leads to reward, the desire for taking this action at the next visit to this state is reinforced. The MDP problem representation requires that each state be time-invariant and contains sufficient information to fully represent the next choice. The state and action representation must be such that a state-action pair uniquely defines the probability distribution of the next state. Sutton and Barto (2018) regards the Markov property to be a property of the state alone [66], thus assuming that the action-set is predefined by the environment and held constant for the duration of the algorithm[1]. Note that the MDP was originally defined for sequential processes with a finite number of states, explicitly excluding continuous stochastic processes[8]. A proper introduction to the MDP and the RL formalism for establishing adaptive procedures is presented by Sutton & Barto (2018) [66]. The MDP is a probabilistic formulation of a sequential process where behavior can be optimized according to an external scalar signal R. Methods in RL consider the search for optimal behavior according to R in the MDP framework.

The *value function* represents experience in the form of the expectancy of future reward. The value of a state $v_\pi(s)$ reflects expected accumulated reward by following policy π from state s and onwards. The reward hypothesis of RL redefines appropriate behavior as the maximization of *expected return* for accumulated R, denoted by the *return* G_t:

$$G_t \doteq R_{t+1} + R_{t+2} + ... + R_T \tag{3.1}$$

where T defines the length of the episode. In its simplest form, the return is the sum of all future reward – as defined by T in Equation 3.1. Assuming that the importance of a *cause* is inversely proportional with the time difference for some *effect* to happen, a simple solution

[1]Sutton & Barto (2018) refers to the Markov property as a trait of the state, *the Markov state*

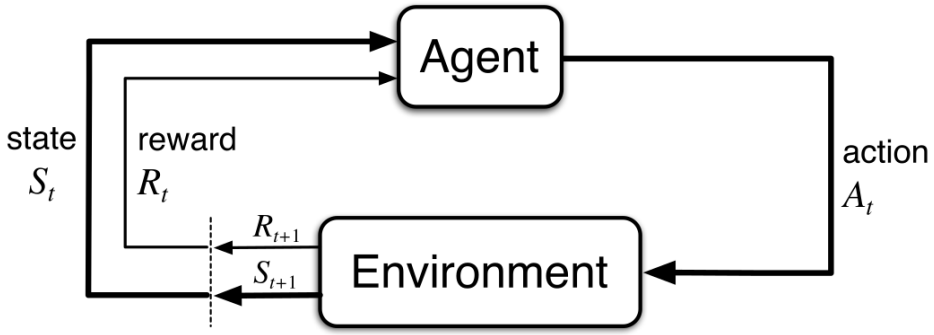


Figure 3.3.: **The MDP considers decision making to be a discrete-time stochastic process**; learning is expressed as a search for optimal stochastic parameters according to a scalar measure of success. The decision *agent* interacts with an *environment* through actions a_t, according to the environment state s_t. The purpose of RL is to optimize agent performance according to a scalar measure of success – the accumulation of reward R [66].

would be to introduce a geometric *discount* factor.

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{\,t+k+1} \,, \tag{3.2}$$

where $\gamma \in (0,1)$ is the scalar discounting factor. The influence of the summed term in Equation 3.2, decreases toward 0 with long time intervals between cause and effect; the effect of $\lim_{t\to\infty} \left[\gamma^k R_{\,t+k+1}\right] = 0$ is stability in learning, and it is no longer necessary to limit learning to episodes. The value function can be written as

$$\upsilon_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \,\middle|\, S_t = s \right] \tag{3.3}$$

$$= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+K+1} \middle| S_t \;=\; s \right] \;\forall\, s \in S \,, \tag{3.4}$$

where $\mathbb{E}_\pi(\cdot)$ denotes the expectation value of the argument $\cdot$ while following policy π. The value function $\upsilon_\pi(s)$ can be thought of as the agent's expectancy for future reward R from state s while following policy π.

A *policy* is a mapping from individual Markov states of an MDP to an agent's drive toward the different choices the agent can make. The RL literature refers to this drive as the *probability* of the agent taking the corresponding action. RL agents change behavior with experience, as expressed by agent policy π being defined by agent experience. Agent experience is represented by the value function $\upsilon_\pi(s)$, defined under the acting policy π. The mutual

dependency between $v_\pi(s)$ and behavior policy π makes the search for optimality challenging. The learning process becomes a slow asymptotic process toward optimality, whereby the agent alternates between improving the policy π and evaluating this policy in terms of updating $v_\pi(s)\ \forall\ s \in S$. Challenges resulting in to slow RL are further explored in Section 3.4. Training time increases exponentially with increasing state space[9], an effect Bellman referred to as *the curse of dimensionality*.

Watkins (1989) proposed an extension to dynamic programming, referred to as *primitive learning* in his thesis[84]. Rather than learning a model that estimates the transition probability and the value function for each state, Watkins proposed to gather experience about state-action pairs directly. Referred to as *Q-learning*, one can think of this model as directly estimating the *quality* of performing action $a \in A$ from state $s \in S$.

$$q_\pi(s,a) \doteq \mathbb{E}_\pi \left[G_t \,\middle|\, S_t = s, A_t a \right] \tag{3.5}$$

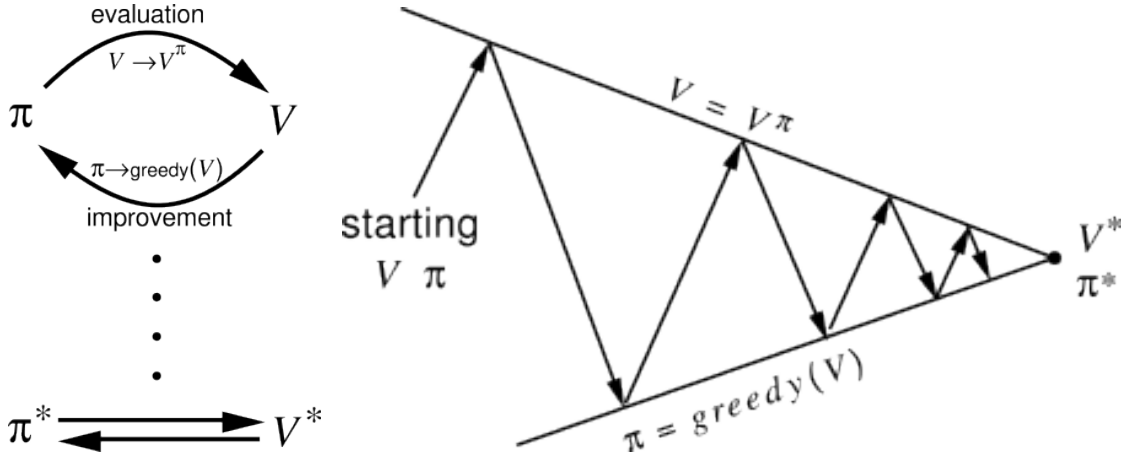
$$= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s,\ A_t = a \right] \tag{3.6}$$

Any practical application of dynamic programming is severely limited in problem size when the agent must learn the probability of transitions from any state to every other state $s \in \mathbb{S}$[28]. When considering the expected return of taking action a from state s and thereafter following policy π, learning proceeds similar to temporal difference learning but without having to consider value and policy separately[83]. Learning the full transition model increases the task according to $\left|S^2\right| \cdot \left|A\right|$, while Q-learning only requires the value for each state-action pairs, $\left|S\right| \cdot \left|A\right|$. Still, the Q-learning agent has to visit every state-action pair several times before a balanced opinion is formed, as illustrated in Figure 3.4b. Q-learning is subject to the curse of dimensionality, limiting any practical use of tabular RL to simple environments and research endeavors.

3.2. Value function by approximation; artificial experience

Despite the prediction known as *Moore's law*[2], proposing an exponential development of computational hardware, the sequential nature of MDP prohibits RL from scaling with computational resources. The latency associated with interacting with the environment limits the learning speed for MDP, not computational resources. Purely digital environments can, or course, speed up execution – thus decreasing the latency associated with each interaction.

[2]Moore predicted that the number of transistors per area would double every two years, a famous prediction which has been fairly accurate since 1965.



(a) policy and value (b) . . . gradually approach optimality as an asymptotic function.

Figure 3.4.: **Learning for MDP can be a slow asymptotic process** due to the mutual dependency between agent policy π^V and agent value function V^{π}. The search for optimality can be slow or unstable (figures from [66])

Physical environments, however, are constrained to real-time execution, and learning speed is unaffected by additional computational resources. When the problem size requires an excessive amount of training time, i.e., the number of interactions required by RL multiplied with the latency involved in an interaction, it is common to consider *artificial experience* – RL agents governed by an estimated value function. Distributed processing in perceptron-class adaptive filters[3], has benefited considerably from the increase in transistor count. Deep networks can be trained for regression, i.e., for estimating the value function for intermediate values between points of experience. RL supported by deep function approximation has been successful for a range of board games and computer games [73, 63, 64, 46, 51, 52]. Whereas distributed graph-based processing scales well with additional hardware, the sequential nature of MDP learning is a more important limitation than computational hardware is for RL algorithms.

Function approximation can be applied in value space and in policy space[12]. Value space function approximation imitates experience by synthesized value function entries for possibly unexplored parts of the state space. The reward signal can be used for supervised learning of perceptron-class networks, making MDP a rich source of automatically generated labeled samples. Tailored deep function approximation has been shown to provide effective support for RL in board games[74, 4], computer games[51, 52, 82], and for learning specific abilities in robotics (reviewed by [36]). Although perceptron-class approximation can be efficient for

[3] Multi-layered perceptrons, ANN, and deep learning are commonly referred to as perceptron-class filters in this text; see the introduction to this chapter for more on perceptron-class algorithms.

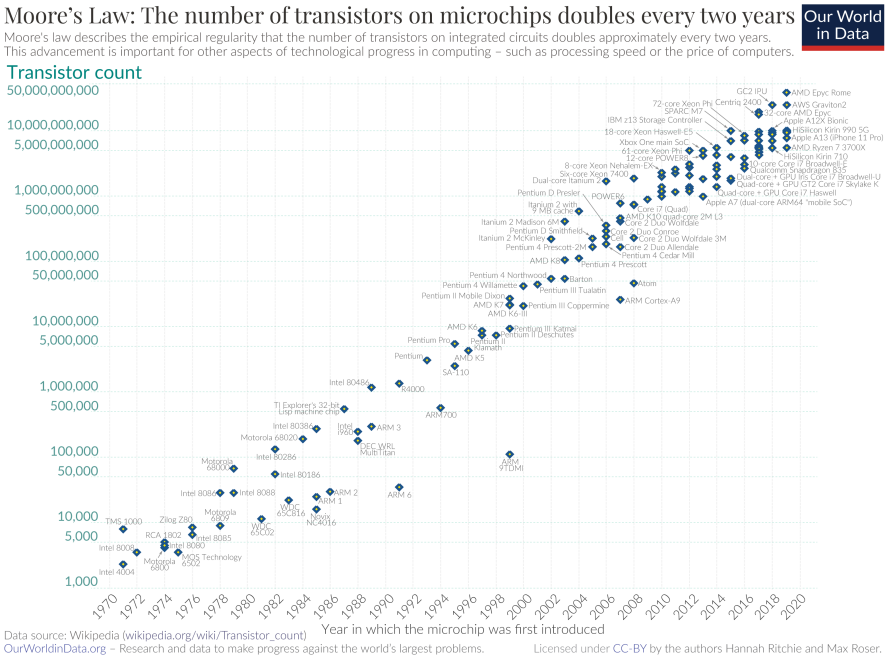


Figure 3.5.: **Moore's law of 1965 is still accurate in 2020.** Moore's law predicts an exponential increase in transistor count per area.

creating artificial values to guide the agent, the use of non-linear function approximation can diverge when used with temporal-difference learning[79].

In addressing the challenges presented here, RL powered by deep function approximation has become a substantial part of modern RL research. Digital environments could clone an MDP or speed up execution to increase training data, facilitating the use of deep function approximation by creating training data. One could even argue that data-driven methods for perceptron-style AI are but another modality of experience, indirectly fulfilling the reward hypothesis by association. However, generating the necessary amount of training data by RL bootstrapping can be prohibitively expensive for real-time interaction learning. Deep RL was responsible for one of the greatest accomplishments in modern computing science when the AlphaGo algorithm accomplished superhuman gameplay in the ancient game of Go[63]. Note, however, how the algorithm required weeks of training on Google's massive machine parks, generating experience equal to millions of above-average human life spans[4]. This amount of training is not plausible for autonomous navigation agents and online navigation.

[4]The policy network was trained on 30.000.000 samples, each collected from a separate game to avoid divergence[63]. Average game time in real-life = 2 hours → 60 million hours = 0.75 million life spans á 80 years. The value network was trained for even longer (50 M batches in 32 games = 1.600.000.000 games).

3.3. Value function by superposition; collaborative experience

Central to RL policy making is the value function. The *prediction problem* refers to the challenge of finding the value function $v_\pi(s)$. The value function reflects the estimated return when following policy π from state s and forwards. After every visit to state s, the value function $v(s)$ can be updated by the *Bellman equation*:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right] \tag{3.7}$$

Scaling the probabilistic discounted return after $a|s$ by the probability of performing a under π highlights how Equation 3.7 learns the value under π. *On-policy learning* updates the value function $v_\pi(s)$ while following policy π [66]. The value function can also be updated *off-policy* – updating the value function under target policy π_t while following another behavior policy $\pi_b \neq \pi_t$. A common example of off-policy learning is in exploration; an agent can initially follow a more exploratory policy to get to know the environment.

Combining Watkins' primitive-learning[84] with off-policy learning for arbitrary reward signals, Sutton et al. (2011) demonstrated how general value functions (GVF) could be formed. Q-learning could form value functions according to signals that could be orthogonal to R and thus unrelated to the behavioral policy. GVF can learn to estimate future values of an auxiliary signal by the same principles as $v(s)$, e.g., the value of some unelated temperature sensor [67]. To avoid confusion, the scalar reward signal for these partial agents is referred to as the learner's *intent* signal in the remainder of this text. The original GVF does not imply a direct involvement in behavior but learns auxiliary information about the environment that could have an indirect role in policy synthesis[67]. Since GVF learns by off-policy learning, there is no limit to how many GVF learners can observe the same stream of experience.

Wiering and van Hasselt (2008) explored the applicability of probabilistic superposition principles for the use in policy [88]; specifically, the effectiveness of Boltzmann addition and Boltzmann multiplication in value space was compared to a voting scheme between agents. The article found that rank voting could be better than Boltzmann addition and multiplication, but one of the authors revealed to me that this conclusion was based on overly specific tasks[5]. Later, Van Seijen et al. proposed an approach for decomposing monolithic tasks into simpler sub-tasks that could be learned by a set of single-minded agents [80]. The separation of concerns was crucial when AI first solved the Atari game Ms. Pac-Man [81]. A set of manually designed reward signals allowed separate learners to form value functions according to different considerations in the Ms. Pac-Man game. Each intent was relevant to

[5] In a discussion between the author and Hado van Hasselt at RLDM 2018, Montreal.

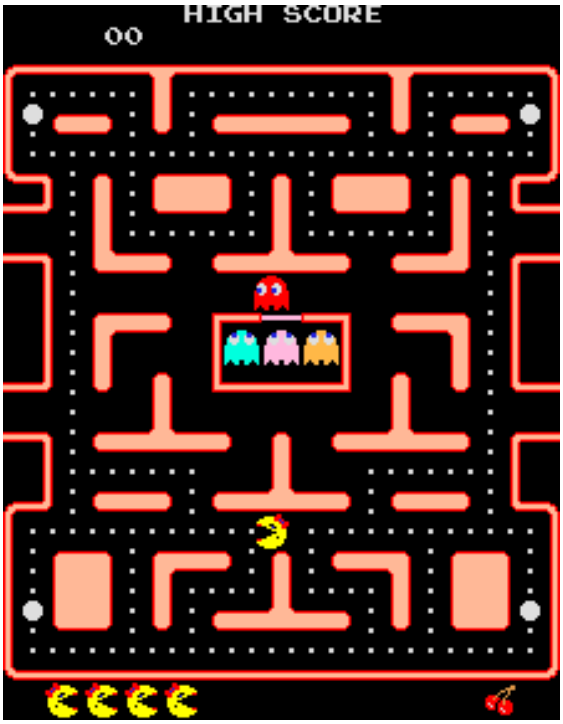


Figure 3.6.: **An autonomous agent for Ms. Pac-Man required *separation of concerns*.** The main objective in the Ms. Pac-Man task can be summarized as capturing desirable elements while avoiding aversive elements. Approaching the objective by separate learners, each learning the value function according to a separate concern, allowed Van Seijen et al. (2017) to reach the maximum score in Ms. Pac-Man [80].

main task, allowing the agent to base the next decision on the aggregated state-action value across the full set of learners. Since GVFs are trained off-policy, several learners can train based on the same sequential training data, making training more efficient. More importantly, Van Seijen et al. found that decomposing the task into a set of smaller MDPs resulted in an exponential break-down of task complexity as measured by the Markovian state-space[6] allowing Ms Pac-Man to be solved without the use of artificial experience.

3.4. Curses for physical interaction learning.

Interaction with the physical world introduces challenges of a different nature than those relevant for board games. First, the physical world is continuous both in state and temporal flow. The real world contains an infinite amount of information; whereas board games have a finite number of possible board positions. A substantial part of the physical world is continuous – requiring discretization to create a digital representation by which the agent can train. Discretizing the continuous parameters creates a digital representation of the system, a representation in which the agent can train. Coarser resolution for Euclidean parameter discretization or considering longer time-steps could decrease training time, measured in the number of interactions. However, the low resolution resulting required by this approach would ultimately be dictated by the curse of dimensionality, severely limiting the capabilities of the trained agent. Second, requirements for real-time execution disqualify many data-driven methods for artificial experience by perceptron-class AI. Even the most accomplished approaches and reports on deep RL for games would be infeasible when training in real-time. The monetary cost of equipment, time involved in repairs, and safety concerns further limit the available training samples when interacting with the psychical world. This section introduces some of the difficulties associated with learning in Euclidean environments, focusing on artificial experience by function approximation and temporal concerns for interaction learning.

3.4.1. Deep concerns with RL for Euclidean interaction learning

A promising application of Euclidean navigation is the planning and execution of movement by a robot manipulator[7]. Many parameters from the physical world are intrinsically continuous, a property that does not comply well with the MDP framework. The original MDP

[6] The raw state space corresponding to the binary channels is estimated to be on the order of 10^{77}, whereas the state space for each of the 1.800 GVFs is estimated to approximately 10^3 states[81].

[7] Euclidean navigation is as relevant for planning and effectuating the angles of a robot manipulator as it is for maritime navigation.

description requires a finite number of states [10]. Typical solutions include function approximation and simplification. First, function approximation can be applied to estimate the value function instead of comprehensive coverage for the value function. We have seen how artificial experience is possible by supervised learning by automatically generated samples from RL and how such approaches can be challenging to apply in real-world interaction learning. Deep RL agents are known to quickly become overly specific for solving the task used in training, lacking the generality to transfer to similar tasks [32]. Second, the system can be simplified by considering coarser discretization, simpler tasks, designed reward functions, or in environments with limited operational range. Although such research can be of academic interest, any application of the trained solution becomes severely limited [36]. As there is less interest in task-specific demonstrations than genuine autonomy, this section presents some of the challenges and shortcomings of deep RL.

As sparsely rewarded MDPs require longer training time to distribute the significant component of the value function across the state space, it can be tempting to help the agent adopt the appropriate behavior by *reward shaping* [39]. Reward shaping involves artificially created intermediate rewards, effectively guiding the agent in the right direction. Assuming a perfectly shaped reward that fully represents the original objective, intermediate reward signals help training by guiding the agent [39]. However, redefining the reward signal changes the MDP and the task for which the agent becomes proficient. Measuring agent proficiency by the same signal that is optimized[8], can be challenging. Measuring the proficiency of the agent by an unverified representation of the reward signal further convolutes the measure of success; it becomes unclear whether it is the proficiency of the agent that is being assessed or the accuracy of the shaped reward. Successful attempts are published, whereas unsuccessful attempts can be forgotten or tried elsewhere. As with the famous example of the Atari game CoastRunners, where a shaped reward resulted in crazed behavior [15], shaped rewards could easily result in dangerous situations when applied in the real-world. It can be tempting to "help" the agent by specifying an artificial objective via shaped rewards, but doing so leads to task-specific training and obfuscated solutions [30].

A second plausible solution would be to train on a simulation; training in a simulation is preferred when training human operators for difficult or dangerous tasks. Training on a simulation could have two appealing properties for digital agents. First, using simulations to generate training samples can be more effective than in the physical world; simulation time does not have to run in real-time – the number of training samples acquired per wall-clock time is limited by available hardware. Second, training in a simulated environment allows

[8] Remember Goodhart's law from the epigraph in the intro of this chapter; Goodhart's law originates[25] from economics, but can be equally relevant to probabilistic algorithms.

for safer training. Some Markov states may be damaging to equipment or personnel. While some states can be crucial to practice on due to a proximity to danger, doing so in the real world would pose a risk to equipment and personnel. Simulated states is not concerned with the same problem; terminal states can be visited in a simulation without extra cost or risk. Unfortunately, RL training on a simulation is known to pose challenges when transferring to the real world. As RL is notorious for exploiting shortcuts and loopholes in a model, an RL agent quickly becomes a specialist on the simulation rather than the real system[36]. Inherently unstable systems are difficult to model correctly for RL training [3]. Although less affected, agents trained for stable systems would still form sub-optimal policies when trained on models rather than the real system [36]. Analogous to our earlier discussion on reward shaping, training on a simulation could form agents that specialize in the digital representation rather than becoming proficient in the real system.

A final option would be to lower expectations of the agent, for example by assigning simpler chores to the robot. Demonstrating simple tasks in lab conditions, without external noise or time-varying dynamics, would be interesting to report assuming that this is the limit of the technology. Examples of tasks for robot RL agents featured on high-impact venues includes opening doors [26], grasping irregular objects [45], or shifting a Rubik's cube by a hand robot [45]. The accomplishments in the above citations are performed by state-of-the-art deep RL, trained for hours or days on specific tasks. Despite being highly successful in games or environments with known and straightforward rules, current RL approaches fail to adapt to new tasks or new terrains [53]. Although deep RL can be demonstrated for a large set of physical environments, each task must be trained individually and from scratch, requiring a significant amount of training[47].

3.4.2. Temporal concerns with RL interacting with the physical world.

Finally, we focus on the role of temporality when applying the MDP framework to physical interaction learning. All physical mechanisms are governed by continuous space and time. The MDP formulation requires that the system state be expressed via discrete, unique entities $s \in \mathbb{S}$ [9]. As a result of the Markov property, the number of Markovian states in state space $\mathbb{S}$ vary with both spatial representation and also temporal resolution. Applying the MDP framework to physical interaction learning requires discrete update times, i.e., discrete time. This section introduces the considerations involved in representing physical systems as discrete-time MDPs, identifying the effect non-trivial temporal systems have on Markovian state spaces as *the curse of temporality*.

Contrary to the temporal flow for board games – where the mechanim driving causality is represented in its entirety by alternating which player makes a move – physical systems

operate in the continuum of time. Future states depend on prior states as well as on the *relative timing* of actuation of actions and external input to the system. Capturing temporal mechanisms in the digital computer requires sampling of the considered variable with a sufficiently high *sampling frequency*. The representation is only accurate down to the time interval between time steps. Higher temporal discretization naturally increases the number of Markov states, quickly growing the MDP problem size. As the sampling frequency must be high enough to represent the finest temporal mechanism of importance, non-trivial temporal systems are a limiting factor for physical RL. In addition, while being capable of representing the finest temporal mechanism, the agent must also be capable of learning slower mechanisms. Representing slow temporal mechanisms in a system with high sample rate, requires sequences of redundant sampling. To satisfy the requirements associated with the Markov property, each state $s \in \mathbb{S}$ must include all necessary information to define the probability distribution of the next state. Multi-step temporal effects would increase the amount of information to be included in each Markov state, increasing the number of states correspondingly. The resulting explosion of states would depend on the relative time horizon for the considered temporal mechanism and the sampling frequency of the temporal representation.

Observation 1 *Let a temporal mechanism require knowledge about system states at L separate time steps to satisfy the Markov property, and $\mathbb{C}$ be the set of possible permutations in the considered system representation. The full set of Markov states required to represent the temporal dynamics becomes $O(\Psi^L)$, where Ψ is the number of permutations in $\mathbb{C}$.*

Take, for example, a temporal mechanism that requires knowledge of the system state from the previous three time-steps to define the probability distribution of the next system state. The Markov state set for this system at time t_n would then be defined by the previous state at t_{n-1}, in turn defined by possible states at t_{n-2} and t_{n-3}, and t_{n-2}. Requiring all this information to be represented in each instance of the Markovian state set would increase the dimensionality of the state set with one axis per term.

For each temporal sub-mechanism, observation 1 states that the number of Markov states increases exponentially with the number of time-steps required to represent the mechanism. The logic applies to all considered mechanisms, implying that each temporal sub-mechanism creates an additional explosion of $\mathbb{S}$. Observation 1 could be viewed as an extension of Bellman's curse of dimensionality for temporal systems. Problems resulting from the temporal effects of the environment may be subsumed under the heading *"the curse of temporality"*.

The importance of temporal considerations is examined by exploring the size of the Markovian state space for different popular RL environments. The state space of non-temporal

MDPs for chess is estimated to be around 10^{50}, for no-limit *Texas Holdem Poker* around 10^{80}, and for *Go* around 10^{170} Markov states. The Markovian state space of the real-time strategy game *StarCraft* is estimated to be several orders of magnitude larger than any of these [55]. StarCraft is but a simulation with a limited number of temporal mechanisms compared to the physical world. Observation 1 proposes that the number of Markov states necessary to learn temporal inferences is exponential with the number of time steps determined to ensure that the slowest relevant mechanism is represented. Compared to the trivial temporal flow in most board games, where time iterates exactly one step after every action, or early computer games, where the introduction of a no-op "action" – choosing not to act – allows for a similar situation, the intricacy of real-world temporal dynamics does not easily fit into the MDP framework. Macado et al. (2020) demonstrated the importance of deterministic temporal propagation in MDP by creating an RL environment with a non-deterministic temporal flow. Extending the popular Arcade Learning Environment [7], Macado et al. introduced what they refer to as "sticky actions" – actions that could last for more than one time step [48]. Non-deterministic temporal propagation after action-selection imposed significant challenges on RL agent performance [48]. The ability to account for non-deterministic temporal flow becomes crucial for RL agents interacting with the physical world.

3.5. Discussion on adaptive algorithms and navigation

This chapter contains an overview of selected approaches for emulating adaptive behavior in the digital computer. The deterministic computer must simulate stochasticity according to a probability distribution with defined probabilistic parameters. The Markov decision process (MDP) emulates learning as a numerical search for optimal parameters in a generative decision process. RL in AI is one framework for this kind of learning model, governed by the following three aspects: the *state* of the system before interaction, the *action* through which the agent interacts with the system, and a *reward signal* that reflects the success of the interaction. The *value function* expresses the expectancy of reward from state s or of taking action a from s while following a specific behavioral policy. For large problems, the *curse-of-dimensionality* prohibits an extensive search as required by pure RL or other methods rooted in dynamic programming. RL supported by deep function approximation is referred to as deep RL, and is known to extend RL capabilities to handle colossal state spaces. However, deep function approximation requires extensive training, limiting its practical use for interaction learning with the physical world or when learning by real-time interaction.

Learning by RL can be dangerous when interacting with the real world. RL agents must explore to learn the task and the environment, a process that can be dangerous for equipment

or personnel. It may be tempting to train an RL agent on a simulation or with the help of reward-shaping to decrease training time. However, the RL framework is notorious for exploiting any shortcuts or loopholes in a simulation. Any discrepancy between the simulation and the real environment can lead to sub-optimal behavior or worse when interacting with the real environment. The sequential nature of MDPs, alternating between policy evaluation and policy improvement, further makes parallel learning through interaction with multiple environments difficult.

Esteemed roboticist and RL-researcher Leslie Kaelbling elegantly summed up state-of-the-art robot learning, implying that current approaches to deep RL are unlikely to succeed for real-world interaction learning. Reporting how today's approaches to intelligent solutions are incapable of general intelligent behavior, especially when having to train in real-time, Kaelbling (2020) points out how RL supported by perceptron-class function approximation is insufficient for learning through interactions in the real world. The paper lists four properties essential for real-world interaction learning. First, the solution must be *sample efficient*; deep perceptron-class function approximation is not sufficiently efficient for real-time learning required in online behavioral autonomy. Second, learning must be *generalizable* beyond the situation in which the agent is trained; the knowledge acquired by deep RL for one task in one environment does not transfer well to new tasks or for dynamic environments. Last but not least, knowledge should be *compositional* and *incremental* – represented in a form that can be combined and appended to existing knowledge. Current approaches for RL do not have these properties, and it is necessary to discover fundamentally new approaches to achieve artificial general intelligence or autonomous navigation[32].

Part II.

Contribution

Chapter 4.

Purposive behaviorism for navigation

> We want AI agents that can
> discover like we can,
> not which contain
> what we have discovered.
>
> *Richard Sutton* – the bitter lesson

The brain is the only mechanism considered capable of voluntary behavior. With unmatched capabilities for autonomous navigation, neural systems have been the main inspiration in this work. Chapter 2 provides an introduction on the relevant aspects of NRES and the psychology of learning and autonomy. Autonomous navigation requires more from adaptive behavior than what is accomplishable by rote learning or the slow process of forming reflexes. Full autonomy requires an agent capable of handling unexpected situations and finding creative solutions, which is referred to as online adaptation in this text, with the following four requirements. (1) All execution and learning must happen in real-time, without prior experience or requirements for pre-training. Experience should originate from a single run; indeed, restarting the whole environment a million times is obviously of little interest for online navigation. (2) The observations and variables considered should be expressed in continuous Euclidean space. The particular meaning of the Euclidean parameters is irrelevant for this research; that is, the agent should be capable of navigating the stock market price as well as the location of a ship. Euclidean spaces are general, and autonomous navigation agents should be capable of handling this generality. (3) Elements of attention, learning the mechanisms of the environment should be separated from the task of acquiring elements associated with reward. Removing or changing an element of interest should not affect the agent's knowledge of the environment. (4) Autonomous navigation is independent of reward specification while training. Since full autonomy requires an agent that constantly evolves,

online navigation requires a separation between the learning agent and the performing agent. Reward metrics can change, and agent priorities should be updated constantly in complex environments, highlighting the importance of learning and behavior separately.

The study of cognitive processes and voluntary actions in psychology is the best source of inspiration for the development of autonomous technology. Chapter 2 covers early movements in psychology, with an emphasis on voluntary behavior. Interestingly, Thorndike's Law of Effect, which has later served as an inspiration for RL in AI [66], was considered as being too simple for expressing the nature of autonomy [87]. The newfound *behaviorist* movement measured physiological and behavioral responses on animal subjects rather than considering introspection as a valid research methodology. Skinner's *operant conditioning* theory considered behavioral responses to be operant toward an objective, capable of better explaining animal behavior as a distributed set of policies. Combined with Tolman's cognitive maps, and in particular reports on NRES from modern neuroscience, operant conditioning and neobehaviorism could supply the necessary mechanistic understanding for reimplementing autonomous navigation in technology.

This chapter summarizes the main findings in this research, with a particular focus on accomplishing cognitive navigation for digital agents. Section 4.1 considers how theory from Chapters 2 and 3 can establish NRES-oriented RL (neoRL) agents for navigation. Latent learning can be expressed as operant behavior toward NRES cells, allowing for voluntary behavior by the activation of operant GVFs. After discussing the complexity of the value function and how function-approximation can be expressed via learned orthogonal value components, the concept behind NRES-oriented RL is described in detail. Section 4.1.3 introduces the two schools of inference learning from early behaviorism and how following a school other than temporal-difference learning is required for neoRL. In developing *inference inversion* for neoRL, a temporally robust approach for learning inferences, the off-policy NRES learning is explained in greater detail. Research on autonomous navigation of Euclidean space requires an appropriate RL *environment*. Section 4.2 discusses important attributes of an RL environment for purposive navigation. Autonomous navigation is demonstrated in the PLE WaterWorld environment by an agent implementing the presented theory. Section 4.3 provides a summary of the experimental results. The chapter is concluded by a brief discussion on possible implications and future directions for the presented theory on neoRL networks in behavioral AI.

4.1. Latent learning and purposive behaviorism by neoRL agents

"Almost all reinforcement learning algorithms involve estimating value functions – functions of states (or of state-action pairs) that estimate how good it is for the agent to be in a given state (or how good it is to perform a given action in a given state)." [66]. The agent can emulate behavioral autonomy in complex environments by function approximation of the value function, referred to as artificial experience in Section 3.2. Applying the same methods for real-world interaction learning – learning to interact with a physical system – is difficult due to expensive training data and intricate temporal mechanics[1]. Inspired by neural capabilities in autonomous navigation and the distributed nature of NRES, in this work, distributed approaches are explored. Rather than learning what generally happens, i.e., system representation by stochastic principles, this section considers whether the superposition principle is applicable for representing a deterministic value function.

4.1.1. The Arnold-Kolmogorov representation theorem

For complex systems dynamics, where modeling can be difficult, the system must sometimes be represented by what *generally* happens. Stochastic representations can predict future values based on measures like average, variance, or other moments of the distribution. Some challenges respond well to the stochastic simplification of the decision process presented in Section 3.1. Others, like the Euclidean navigation task appears to be less appropriate for a pure stochastic model. Wold's representation theorem states that any covariance-stationary time series x_t can be written as the superposition of a probabilistic component and one deterministic component

$$x_t = z_t + u_t \,, \tag{4.1}$$

where z_t is a linearly filtered white-noise process y_t, $z_t = y_t + b_1 y_{t-1} + b_2 y_{t-2} + \dots$, and u_t is a deterministic component [90]. Allowing for partial inclusion of deterministic components in the otherwise stochastic MDP, Wold's representation theorem allows for experimentation with deterministic behavioral components.

Consider first a pure white-noise navigation agent that chooses actions at random. The resulting Wiener process – a process that can be expressed as the integral of a white-noise process – is represented by the position of the agent after random movement. At any time $t > t_0$, the Wiener process W_t can increment with a random step in either direction along each axis, resulting in Brownian motion. When the decision process has an equal probability of choosing all directions, the location of the agent can be modeled as a Wiener process. The

[1] See Section 3.4.1 for more on the challenges of real-world interaction learning.

white-noise signal for the Wiener process can be affected by a linear filter b, establishing z_t from equation 4.1. Making the linear filter a function of the state, $b(s)$, the random process W_t can be shaped into expressing directed behavior. Learning can be regarded as inserting bias into this stochastic process, a bias that favors actions associated with reward R. RL can be seen as an approach for introducing bias into a white-noise behavioral process, in effect adapting behavior according to experience.

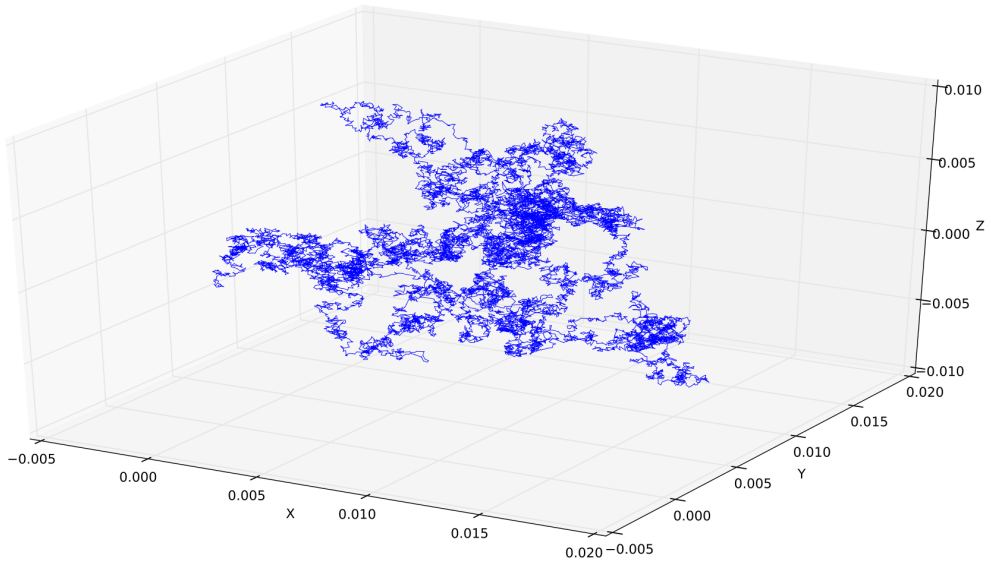


Figure 4.1.: **The Wiener process in 3D Euclidean space.** With an equal probability of stepping in every direction, the position of an ε-greedy navigation policy with $\varepsilon = 1.0$ can be modelled as a Wiener process. (Figure by Shiyu Ji // Creative Commons)

Consider the deterministic component of a value function represented by equation 4.1 as the *desire* of the agent. Representing the full complexity of desire as a deterministic function would involve intricate and multi-variate relations. In the year 1900, renowned mathematician David Hilbert compiled a list of 23 unsolved problems where the thirteenth problem entailed whether higher degree equations could be represented by means of functions of only two arguments [27]. The Kolmogorov-Arnold representation theorem states that every multivariate continuous function can be represented as the superposition of single-argument functions.

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \chi_q \left(\sum_{p=1}^{n} \psi^{q,p}(x_p) \right), \tag{4.2}$$

where $\chi_q(y)$ are continuous real functions [37]. Also referred to as the superposition theorem,

the Kolmogorov-Arnold representation theorem proves that complex multi-variate functions can be decomposed into a set of continuous functions of one variable.

Assuming that behavior can be represented by a sufficiently intricate desire function, Wold's theorem allows for partial simplification of desire by stochastic representation. Likewise, Wold's theorem opens for a partial introduction of a deterministic component for stochastic MDPs. Representing the full value function as a deterministic function would assuredly require intricate multi-variate mechanics of high complexity. The Kolmogorov-Arnold representation theorem allows for the separation of concerns, which can be beneficial for two reasons. First, the simpler, possibly mono-variate, general value functions facilitate learning by considering less intricate MDPs. Bellman's curse-of-dimensionality dictates that MDPs with larger state spaces require exponentially longer training time, or vice versa, how decreasing the considered state space facilitates learning. Second, considering the decomposed value function as agent desire could allow for transparency and partial control over the autonomous solution. *Operant desires* represent behavioral atoms analogous to Skinner's operant conditioning[2]. Introducing operant desires for RL could be a first step toward latent learning and purposive behavioral AI.

4.1.2. Latent learning and NRES-oriented navigation

Consider first a simple grid world with a single rewarded state $s^* \in \mathbb{S}$ and actions corresponding to positive and negative movement along each axis $\mathbb{A} = \{N, S, E, W\}$. Reaching s^* results in a unitary reward $R = +1.0$. No other rewards exist in this environment, and the agent should learn how to reach s^* from experience. For any Euclidean coordinate $\vec{c}$ in Figure 4.2, there exists exactly one corresponding grid world state $s \in \mathbb{S}$. Thus, the mapping from $\mathbb{E} \in R^2$ to $\mathbb{S}$ is unique and comprehensive; there exists one and only one state $s \in \mathbb{S}$ for every coordinate $\vec{c} \in \mathbb{E}$.

The NRES transform from Section 2.2.1 can establish a discrete representation of Euclidean information. An NRES population with mutually exclusive receptive fields could be considered as a biological expression of tile coding across distributed one-hot neurons. Let the receptive fields of the NRES population correspond to the grid world of Figure 4.2. When the Euclidean parameters change sufficiently to trigger a new NRES $s^+ \in \mathbb{S}$, events preceding this NRES cell change should qualify for the inference $s^- + e \rightarrow s^+$. Events that repeatedly take part in the transition from s^- to s^+ are likely to be important for this state change. Note how NRES state for complex systems forces us to consider state change as being an effect of the environment rather than controlled by agent action. Since NRES transitions and causalities

[2]Section 2.1.1 covers reinforcement learning in psychology and operant conditioning from the neobehaviorism perspective.

of the real-world can happen without agent interference, the learning agent should look for the cause of the transition rather than delving into one's own actions. The NRES-oriented RL agent can use the activation signal for individual NRES cells as reward signals for training operant desires by off-policy GVF in $\mathbb{S}_N$.

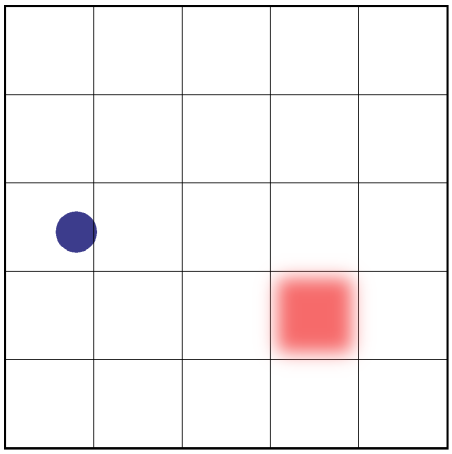


Figure 4.2.: **A simple NRES representation of Euclidean space.** The simplest comprehensive and mutually exclusive set of receptive fields results in the grid-world representation. The $5x5$ representation, referred to as $N5$ in this work, can define 25 separate OVF – each with a different intent. The OVF with intent toward $[4, 4]$ will learn the value function according to a positive reward when activating the corresponding receptive field. [41]

Tolman's purposive behaviorism involves a clear separation between learning and behavior. An expression of latent learning could be achieved by distributed off-policy learning by operant desires, trained operant GVFs toward the activation of NRES cells. Purposive behavior can further be expressed as a generative process similar to the series-expansion of orthogonal desire components. Orthogonality in the value space can be defined as

Definition 4 *Orthogonal value functions of state space $\mathbb{S}$ are value functions that are trained on mutually exclusive reward signals.*

Operant desires in the form of GVF toward NRES activation can form a set of OVF, assuming that NRES fields are independent and mutually exclusive. Using the activation signal of NRES cells with mutually exclusive receptive fields as reward signal for a set of GVF effectively forms singular reward signals in $\mathbb{S}_N$. Operant desires trained with mutually exclusive NRES activation signals for reward, result in one set of OVF. The set of OVF establishes a basis in the value space domain, motivating the use of the Kolmogorov-Arnold representation theorem for desire.

Motivation and purpose can be expressed as elements-of-interest – projections of the agent's expectancy of reward in the considered NRES modality.

Definition 5 Elements-of-interest *of an NRES modality represent parameter configurations associated with reward. Elements-of-interest are associated with valence, a measure of an agent's motivation for reaching the it; elements-of-interest are associated with location, the parameter configuration of the element in the considered NRES modality.*

Let the location of elements-of-interest activate and extract latent knowledge in the form of OVF. The agent's motivation for reaching an element-of-interest is proportional to its *valence*, the expectation of reward associated with the element. Scaling the corresponding OVF by reward expectancy associated with elements-of-interest contained in the associated NRES cell, allows for purposive behavior by desire through series expansion:

$$Q_N(s,a) = \sum_{i \in \mathbb{S}_R} w_i Q_{\mathcal{L}i}(s,a) \ , \tag{4.3}$$

where w_i is proportional to the valence of element i, and $Q_{\mathcal{L}i}$ signifies the OVF value function activated by the NRES containing element-or-interest i. Since reward only comes from elements-of-interest ξ_i in the domain $\mathbb{E}$, reward expectancy is limited to the set,

$$\mathbb{S}_R = \{s \in \mathbb{S} \ \big| \ \exists\, \xi_i \in \mathbb{E},\ \xi_i \in s\} \tag{4.4}$$

For simple challenges with direct reward mechanisms, valence can be programmed individually. For autonomous navigation and life-long learning, valence could be learned or adapted over time – a simple task for other branches of machine intelligence. Note how positive valence implies an attractive element-of-interest in $\mathbb{e}$, whereas negative valence results in an aversive effect for the element-of-interest. The full NRES value function Q_E can be formed as the weighted sum of OVF activated by elements-of-interest.

4.1.3. Inference inversion in neoRL

Finally, the alternative to temporal difference for early behaviorism should be discussed. Tolman (1948) writes about two schools in behaviorism, where the temporal difference school is analogous to TD learning in RL. The equally acknowledged *persistence school* of behaviorism focuses on how essential events are presented simultaneously with the desired response more often than the sporadic occurrences of less relevant events. Stimulus-response pairs that occur more frequently tend to "get strengthened at the expense of the incorrect connections"[76]. The discrepancy between the two explanations may be subtle, and the

operant inference trinity from Section 4.1.2 is repeated to facilitate discussion:

$$s^- + e \rightarrow s^+ \quad , \tag{4.5}$$

where s^- has the role of Skinner's discriminatory stimuli S^R – the pre-condition for the inference; event e represents the action –the conditioned reflex involved in the causality; s^+ represents the conditioning signal S^R – the outcome of the inference. Whereas the temporal difference school focused on the time difference between cause $\{s^-, e\}$ and effect $\{s^+\}$, the persistence school of behaviorism would instead look for a persistent reason $\{e\}$ behind the transition $s^- \rightarrow s^+$. The two directions can be summarized as: (a) the temporal difference movement focusing on the *cause* as the learning driver, while (b) the persistence school emphasizing the *effect* of the inference as what drives learning. This section considers the development and conceptual basis for *inference inversion* in RL.

Definition 6 *The temporal driver of a process is the signal responsible for the flow of time.*

The temporal driver of the learning process and the temporal driver for the underlying inferences should be considered separately. The temporal driver of a discrete-time process is the event that makes time progress by one step such that $t \leftarrow t + 1$. Most MDPs have the event of a new action $a \in \mathbb{A}$ as the temporal driver. Examples include chess, where time progresses after the player makes a choice, or Atari, where introduction of a no-operation action $a_{noop} \in \mathbb{A}$ after a time-out allows for similar temporal dynamics[3]. In real-world systems, it may be necessary to differentiate between the learning driver and the temporal driver responsible for the inference being studied.

In traditional RL environments, the occurrence of an *action* establishes the temporal driver of both the learning process and for inferences in the environment. The Bellman equation updates the state-action value after every action:

$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha)\, Q(s_t, a_t) + \alpha \left(R_t + \gamma \max_a Q(s_{t+1}, a) \right) \quad , \tag{4.6}$$

where α is a 'learning factor' [84]. Using the occurrence of $a \in \mathbb{A}$ as a temporal driver allows the Markov property to be expressed as an attribute of the state alone – referred to as the Markov state by Sutton and Barto (2018). Whereas some challenges respond well to action-driven learning, others require temporal abstractions that challenge traditional RL. Temporal abstractions in RL have been explored for half a century with limited success – with the *options* framework being the most accomplished. Sequences of basic actions can

[3] A modified version of the Atari environment has been developed [48] with non-deterministic temporal propagation, a modification and a new environment that has received far less attention than deserved

be expressed through temporally extended options, allowing the semi-MDP framework to study temporal abstractions by action-driven MDP [68]. For the temporal complexity of the real world, however, even RL supported by options and deep function approximation is challenged when the task becomes interesting [36, 32]. In some cases the two temporal systems may be independent, such as the temporal abstraction for inferences is separate from the scale in which the agent performs actions; hence, purely action-driven RL may express a fundamental flaw for behavioral AI.

The persistence school of behaviorism could be emulated by assigning credit to events after each transition $s^- \rightarrow s^+$ as a leaky integrator. All events that happened while at s^- should be assigned equal weight when transferring to s^+, for example by changing the learning-constant α to include a measure of persistence.

$$\alpha_i = \psi_i(N)\alpha \quad , \tag{4.7}$$

where the ψ_i is a function of the number of times event i happened during pre-state s^-. Multiple occurrences of the same event should affect learning positively through $\psi_i(N)$, whereas zero occurrences would affect learning negatively according the result of $\psi(0)$. Accepting the occurrence of an event as the basis for learning individual inferences, it is still possible to utilize a transition-driven update by inference inversion,

$$Q^{new}(s_t, a_i) \leftarrow (1 - \alpha_i)\, Q(s_t, a_i) + \alpha_i \left(R_t + \gamma \max_a Q(s_{t+1}, a) \right) \quad , \tag{4.8}$$

after $s^- \rightarrow s^+$ for all inferences $s^- + a_i \rightarrow s^+ \,\big|\, a_i \in \mathbb{A}_I$. $\mathbb{A}_I$ is the set of all operand events considered during the update. Equation 4.8 affects all inferences considered between s^- and s^+, increasing the inference value for events that happened and decreasing the value otherwise. Note how equation 4.8 learns the inference value from the pre-state s^- to post-state s^+, rather than learning state-action values toward some global reward. The neoRL agent is governed by the reward hypothesis via elements-of-interest, the agent's projections of expectancy or reward in the considered Euclidean space.

4.2. Research environment for autonomous navigation

Behavioral AI research involves two components: the agent, representing the autonomous entity, and the environment, the system where the agent expresses behavioral autonomy. The environment defines the set of choices available for the agent, thus deciding what can be learned by the agent. Most available environments for RL research highlight popular challenges for traditional RL research; hence, the low number of environments considering

Euclidean navigation or challenges where the flow of time is separate from the occurrence of an action[4] indicates the difficulty for achieving these aspects by RL alone. Although new environment can easily be implemented, tailoring environments while trying to demonstrate an effect might raise concerns with the generality of the results. This section describes the main priorities for an environment for research on online Euclidean autonomy, and our search for an existing environment that satisfies these criteria.

An implicit assumption in most RL research is related to the scalability of results. Learning to play games or solve toy problems is interesting under the assumption that (a) methods scale to real-world complexity and (b) results are general. The hypothesized curse-of-temporality[5] states that even the simplest temporal mechanisms can drastically increase the number of Markov states, possibly beyond what monolithic RL agents can handle. Temporality for the selected environment should follow non-trivial mechanics, a requirement that excludes most mainstream RL environments. Many task-specific environments exist for toy problems or algorithmic challenges, whereas finding an appropriate environment for general Euclidean navigation with temporal concerns can be challenging. The four requirements for online navigation listed in the introduction of this chapter have guided the search for an appropriate RL environment for navigational autonomy: (1) Interaction with the environment should happen in real-time, limiting the number of samples and making statistical function approximation difficult. (2) Navigational state and observations should be reported as continuous (Euclidean) coordinates. (3) Latent learning and the separation between behavior and learning, together with (4) changing reward structure, requires that the environment reports one's position and elements of interest separately. Traditional RL environments can not account for these requirements, forcing an unconventional choice of research environment for autonomous navigation.

The PyGame Learning Environment (PLE) implementation of Karpathy's WaterWorld environment [70] considers a navigation challenge for an agent, governed by inertia mechanics. Basic actions accelerate the agent's body (blue dot) in the allocentric directions up, down, right, left – (N, S, E, W). The representation of the self (blue), in addition to a fixed number of elements of interest with varying valence (green, red), exist in the environment. Green objects are associated with a positive reward of +1.0; red objects are associated with a negative reward of -1.0. Encountering an element removes this entity and rewards the agent according to the element's valence, after which a new object is initiated with random valence, location, and speed vector. The capture of the last green entity causes a full reset, reinitializing all elements according to the described mechanics. Rewards come exclusively from

[4] Remember the direct link between action selection and the flow of time in traditional RL; see Section 3.4.2.
[5] See Section 3.4.2 for more on temporal concerns in RL.

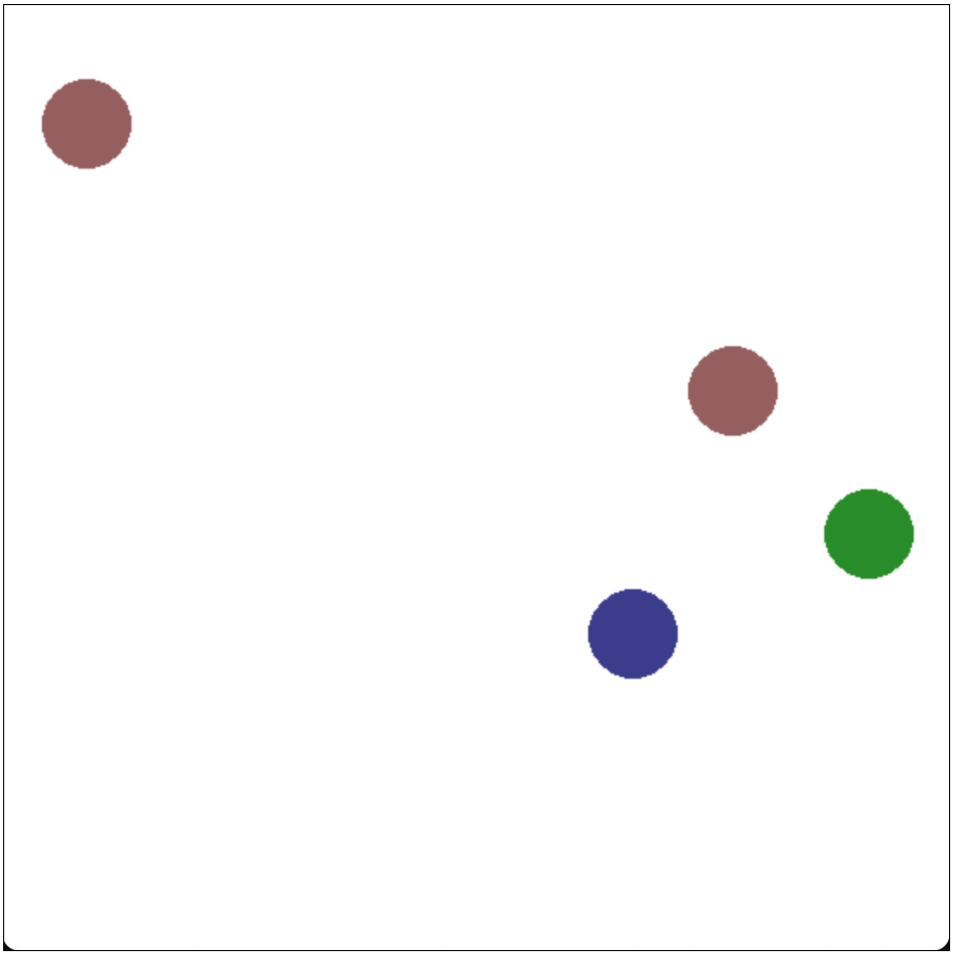


Figure 4.3.: **The WaterWorld environment for autonomous navigation research.** The PLE WaterWorld challenge [70] represents the agent in a 2D Euclidean space. Green and red elements signify locations with positive and negative reward expectancy, respectively. The green and red elements of WaterWorld change location with a constant speed and are reflected upon encountering the wall. The agent (blue) moves according to inertia mechanics, with an action set that accelerates the body in the four cardinal directions N, S, E, W.

encountering elements-of-interest, making accumulated reward an appropriate measure for the agent's navigational capabilities. Accumulated reward directly represents how well the agent has succeeded in capturing green elements and avoiding red elements. Note how the PLE version of WaterWorld differs from the JavaScript version, with REINFORCEjs being implemented with a simpler egocentric representation – where directional "eye sensors" measure the existence of and direction to objects of interest [35].

The allocentric PLE WaterWorld implementation can test all four requirements for autonomous navigation listed in the introduction of this chapter. WaterWorld considers Euclidean reactive navigation, with real-time execution, according to several free-roaming objects of interest. The fourth requirement is expressed by a possible change in valence after the capture of an element, establishing a chaotic and reactive navigation scenario. All experiments study autonomous navigation in the WaterWorld environment with learning happening in real-time while navigating. Online navigation (learning) capabilities are directly observable as the (change in) immediate proficiency of the agent.

4.3. Results for neoRL autonomy; contribution and publications

An neoRL navigational agent have been designed as presented in Section 4.1.1 and 4.1.2 to navigate the WaterWorld challenge. All learning is a result of real-time execution and autonomous navigation in real-time during a single run. Each experiment starts with no priors other than what has been presented in this chapter. A selection of the findings and the development process have been published and presented at conferences and seminars, in venues ranging from computational neuroscience to artificial general intelligence intelligence.

The main contribution of this project can be divided into three milestones, as presented in paper A-C. First, paper A proposes how a distributed state representation is possible by considering the value function as a potential. A selection of findings and theory extracted from the development of the neoRL framework has been presented in venues on computational neuroscience[6], and biological neuroscience[7], before being summarized in the first included paper [41]. After introducing the basic principles of purposive neoRL navigation, the article presents experimental results demonstrating an agent capable of forming behavior across multiple state spaces to facilitate autonomous navigation. Paper B explores this further by considering multi-NRES navigation originating from OVF from separate NRES modalities. Learning inferences in different NRES modalities, i.e., independent navigational information as represented by separate Euclidean spaces, emphasizes the temporal robustness of inference

[6]The conference of cognitive and computational neuroscience [40]
[7]Invited talk on National Neuroscience Symposium, organized by the Kavli Institute of Systems Neuroscience.

inversion. Papers A and B consider combining separate value function outputs by linear principles. With conceptual similarities to the one-layered perceptron, the multi-resolution and multi-modal neoRL can be considered a one-layered behavioral graph. A future paper included as a manuscript C explores this analogy – whether deep neoRL graphs are possible by generating elements-of-interest as output from neoRL nodes. Note how autonomously formed elements-of-interest induce full category II navigation, whereby objective location and valence in one NRES modality are formed based on learned inferences. This chapter presents an overview of the main findings in milestone A-C.

4.3.1. Decomposing the prediction problem; Autonomous Navigation by neoRL

This first milestone establishes the basic principles for neoRL navigation, as presented in Section 4.1. Considering the value function as a potential field, paper A demonstrates how different forces can contribute independently to an aggregate value function. Orthogonal value functions can be learned by off-policy GVF learning, rewarded by NRES activation. Hence, off-policy training of OVF becomes analogous to latent learning of cognitive maps for digital agents. The resulting map formed by the full OVF set can guide behavior according to purpose, as represented by elements-of-interest. Experiments conducted in the WaterWorld environment verify theoretical findings, as well as guiding further development of the neoRL framework.

First, a comparison of NRES with different resolutions uncovers the importance of NRES resolution for agents' navigational capabilities; both learning speed and final proficiency vary with NRES design. Inspired by the NRES representations in the brain which steadily increase in resolution along the dorsal-to-ventral axis of the hippocampus, a multi-resolution neoRL was implemented. The multi-resolution neoRL agent across multiple NRES sets performed better than single-resolution neoRL agents on all accounts such that the agent learned quicker and attained higher proficiency, when the neoRL value function originated from multi-resolution NRES compared to single-NRES neoRL performance. A 3.5-fold increase in final performance was accomplished during the same execution time, implying quicker learning despite involving almost six times the number of states for the multi-res neoRL agent.

Directed exploration is proposed as a plausible explanation for improved learning when navigating according to additional information. Whereas an ε-greedy policy ensure sufficient exploration by imposing complexity by partially adhering to a white-noise process[8], directed exploration based on earlier experience could instead be viewed as an analogy to intelligence as psychology defines it[9]. An improved learning performance by assessing more auxiliary information could be a first indication of emulated intelligence – problem-solving by purposive exploration. Directed exploration, or intelligent extraction of auxiliary experience when attempting new parts of the state space, is essential for autonomous navigation.

[8] See Section 4.1.1 for how a Wiener process can be expressed as a pure white-noise process, i.e, with $\varepsilon = 1.0$.

[9] Although the definition of *intelligence* is a disputed subject in psychology, it is reasonable to view intelligence and problem solving as the ability to achieve a goal in a novel situation [49].

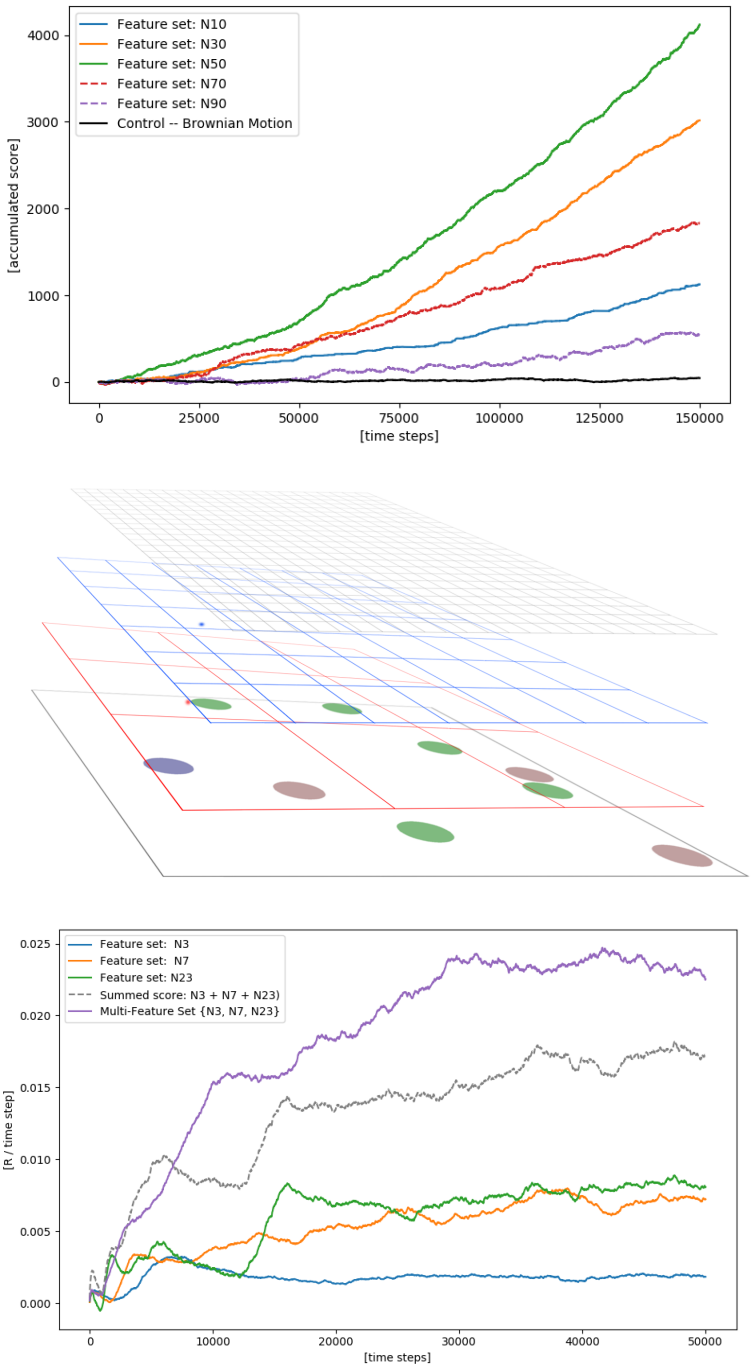


Figure 4.4.: **Autonomous navigation by purposive neoRL policy extraction.** [Top] The proficiency of single-NRES neoRL agents vary with NRES resolution; the correlation between NRES resolution and neoRL performance is considered as an important verification of the existence of a coherent mechanism. [Mid] Latent learning and purposive neoRL can collaborate between distinct NRES maps in forming behavior. The illustrated NRES layout represents the agent in experiment 2. [Bottom] Experiment 2: the multi-NRES oriented agent shows real promise for autonomous navigation. (Figure from [41])

4.3.2. Navigating conceptual space; a new take on Artificial General Intelligence.

The second part of this project explores the generality and modularity of the neoRL approach; paper B addresses Leslie Kaelbling's (2020) concerns regarding robot learning by deep RL. Kaelbling lists four challenges for the current direction in robot learning in a letter in Science (2020) stating that navigation must be (a) sample efficient – require little training to achieve the task, (b) generalizable – apply to other situations than it was trained on, (c) compositional and (d) incremental – possible to combine with earlier knowledge or extendable by new experience. By considering robot planning as a special case of Euclidean navigation, Kaelbling's concerns are used as a guide to test the capabilities of neoRL navigation in this paper.

Two experiments address Kaelbling's concerns, slightly adapted to address Euclidean NRES navigation. First, the agent is challenged by navigating to the position of elements-of-interest by another NRES modality. The recently discovered object-vector cell [29] (OVC) in neurophysiology is emulated for WaterWorld observations. The OVC neoRL agent is capable of comparable results as the native place-cell (PC) neoRL agent, demonstrating a generality in inference inversion across NRES modalities. Second, capabilities for incremental or compositional learning are put to the test by letting a single agent experience and learn in both PC and OVC NRES modality, and behave according to both. The neoRL agent is capable of learning (a) efficiently when challenged by (b) auxiliary information or when (c) supplied with more information in the form of additional NRES state sets. See Figure 4.5.

Contrary to traditional RL in AI, known to require longer training when assessing more information, the neoRL agent learns to a higher proficiency in a shorter time when considering more information. Increasing learning efficiency by considering more information is unheard of for monolithic RL; the curse-of-dimensionality has limited practical uses of RL for more than 70 years. Note that neoRL does not affect the curse-of-dimensionality directly; rather, the decomposed and distributed learning process is believed to alleviate the curse by considering smaller NRES sets and with simpler reward functions. A *decrease* in training time might be an effect of directed exploration, as proposed in Section 4.3.1.

Finally, a short word on the title: Theoretical neuroscience considers reasoning and general intelligence to be an effect of navigating a conceptual space, a cognitive representation of ideas as vectors expressed by NRES structures [11]. Conceptual spaces can encode ideas and connections as points or vectors represented by NRES structures [6]. MRI measurements on human subjects support theoretical results on NRES coding for ideas and reasoning [16]. Purposive navigation of such a space requires agents capable of finding inferences and of category II navigation in multi-dimensional cognitive space – without being limited to a predefined action set or action-driven learning. Article B proposes the neoRL framework as a potential first step toward conceptual navigation by digital agents.

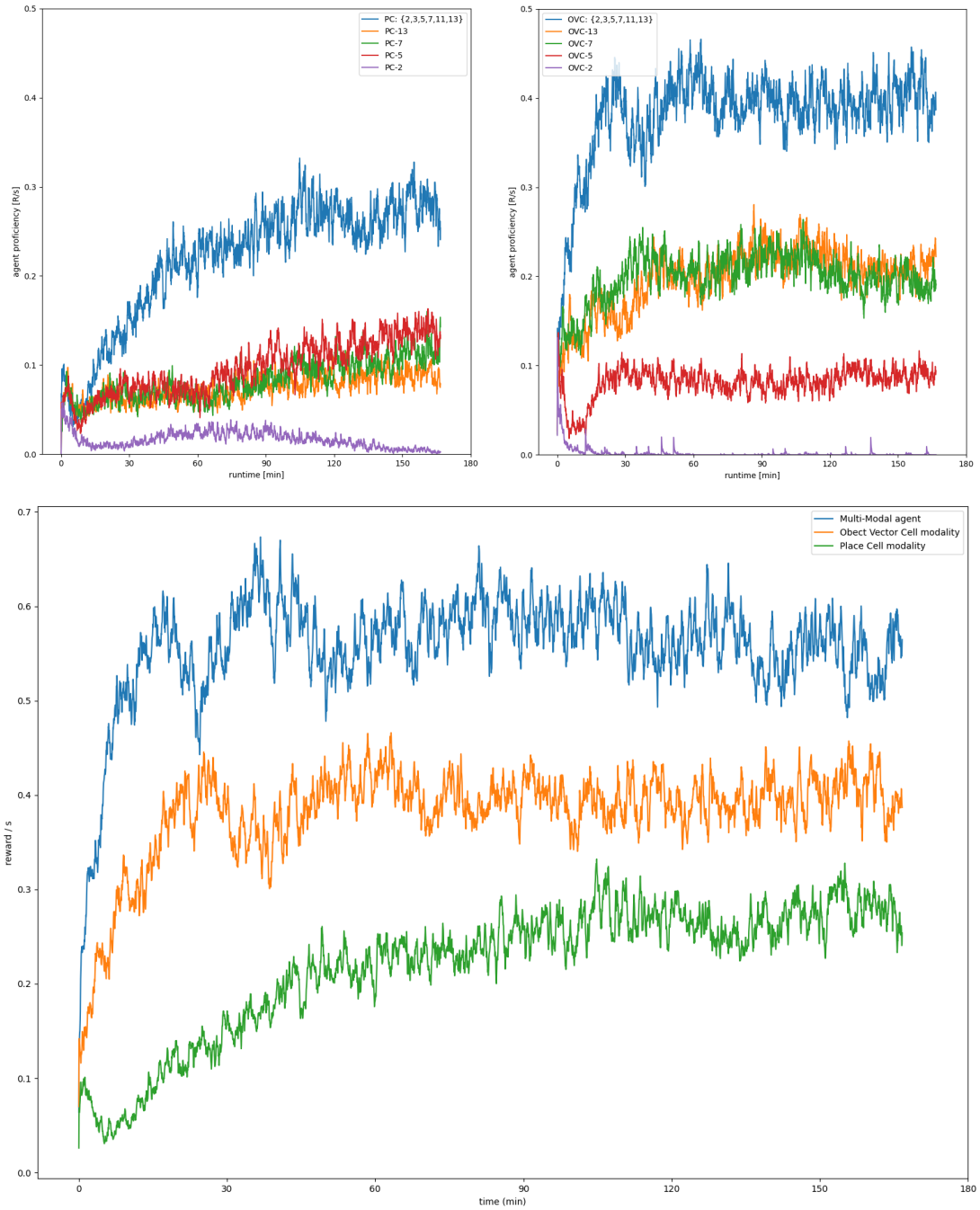
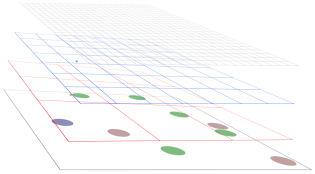


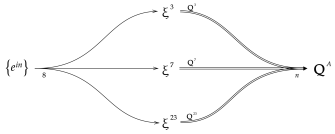
Figure 4.5.: [Top] The neoRL architecture is general across NRES modalities: (A) the original place cell (PC) NRES modality is implemented by applying NRES code directly on an allocentric location of the agent or elements of interest. (B) An emulated object vector cell (OVC) NRES modality is implemented by vector subtraction. OVC is centered on the self with an allocentric representation of other objects. [Bottom] The neoRL navigation agent is compositional across NRES modalities. An agent governed by both PC and OVC learned maps performs better than mono-modal agents. Note the y-axis of the plot; the top curve from figure A and B are repeated here for comparison. (figure from [42])

4.3.3. Towards neoRL graphs; the emergence of purposive networks.

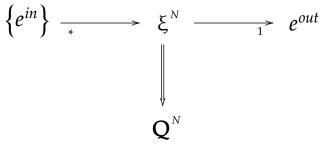
Finally, we address the challenges implied by definition 1 and 2. A major discrepancy between AI and systems behind real intelligence considers capabilities for reuse and recursion in psychology, as illustrated by e.g. learning how to learn or holding a desire for more purpose. Real-world complexity necessitates learning capabilities beyond what is possible by linear increments of agent capabilities; online autonomy in physical environments is not plausible with today's RL – presumably limiting neoRL performance as well. Reuse of operant reflexes, synergy effects between subsystems, and recursive connections could be crucial in neural cognition – motivating a study of network phenomena for neoRL.



(a) The multi-resolution NRES state space from paper A.



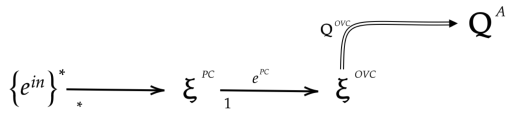
(b) An illustration of purposive value from figure a.



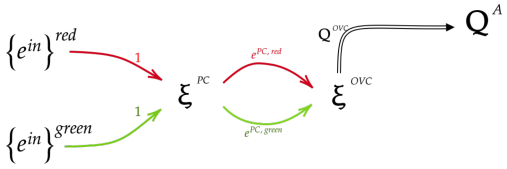
(c) The purposive neoRL node.

Manuscript C explores neoRL agents as purposive networks based on the Arnold-Kolmogorov representation theorem and learned projections of desire. Considering actions with a Euclidean significance, expressing displacement in the considered space, the vector sum of the resulting state-action value function could be interpreted as a Euclidean projection of agent desire. The manifestation of a learned state-action value map, harvested according to projections of desire, expresses a purposive desire vector e^{out}. Illustration b shows how the Q-values from three NRES maps can be combines into one neoRL Q-vector. Figure c illustrates the Euclidean neoRL node allows for two outputs Q^{out} and e^{out} from one set of elements-of-interest. As a projection of agent desire according to experience, one can consider purpose fragment e^{out} as being autonomous according to definition 1. The purposive desire-vector e^{out} can further establish an element-of-interest in compatible neoRL nodes.

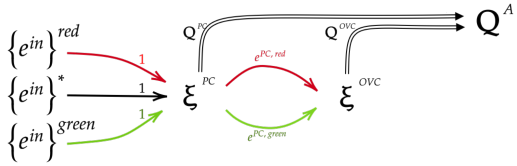
Autonomous desires imply a category 2 autonomy by neoRL agents. Experiments demonstrate four principles of purposive networks: (a) autonomous desires are possible by neoRL agents, (b) different desires can be extracted from the same learned map according to purpose, (c) the value function from different depths of the agent could contribute equally to agent value function, and (d) recursive desires can improve navigational performance. Results are repeated in Figure 4.7. Each neoRL node extracts desire-vectors according to purposive projections in a learned cognitive map, in effect implementing Tolman's purposive behaviorism for AI.



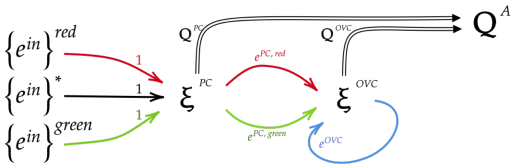
(a) Aspect 1: a single desire vector e^{PC} as input to ξ^{OVC}.



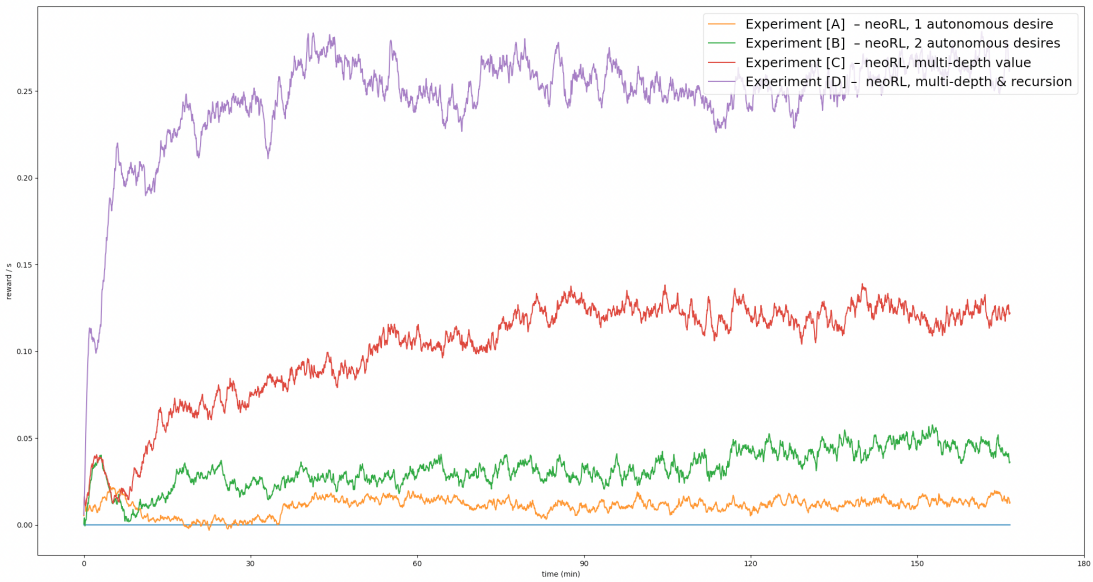
(b) Aspect 2: separate desire extraction; $e^{PC_{red}}$ and $e^{PC_{green}}$.



(c) Aspect 3: joint value function from sequential neoRL nodes.



(d) Aspect 4: recursive desires for neoRL autonomy.



(e) Transient proficiency of neoRL agent A-D.

Figure 4.7.: [**[Top]**] Illustrations of the neoRL architecture tested in experiment A-D. [**a**] The neoRL node ξ^{PC} forms a single desire e^{PC} for value-generating neoRL node ξ^{OVC}. [**b**] Experiment B formed separate desire-vectors $e^{PC_{red}}$ and $e^{PC_{green}}$ from ξ^{OVC} – grouping according to valence. [**c**] The value function output from neoRL node ξ^{PC} and node ξ^{OVC} contribute equally to agent value function. [**d**] Recursive desires are possible for neoRL nodes: the ξ^{OVC} is governed by three elements-of-interest, $e^{PC_{red}}$, $e^{PC_{green}}$, and recurrent desire e^{OVC}. [**[Down]**] Results from the four experiments: [**e**] Purposive neoRL networks allows for purposive autonomy by deep and/or recurrent desires. (Figure from [43]).

4.4. Discussion; autonomous navigation by neoRL agents

In this chapter, the neoRL framework has been developed based on theory presented in Chapters 2 and 3. First, after considering the mathematical formulation of MDP and how Wold's representation theorem allows for partial deterministic representation of the decision process, we proposed referring to the deterministic component of the value function as *desire*. The intricate and presumably multi-variate desire function can be decomposed by the Kolmogorov-Arnold representation theorem. The superposition theorem states that any multi-variate function can be represented as the weighted sum of mono-variate relations. Hypotheses from Skinner's operant behavior from Chapter 2 could inspire how simpler relations can be learned. Based on NRES maps from Section 2.2.1, we have seen how individual GVF can learn how to achieve the activation of singular NRES cells – in effect learning operant value functions for that objective. A set of OVF can establish the basis for value function composition by the Kolmogorov-Arnold representation theorem. Free-roaming elements-of-interest in the considered NRES modality can further activate OVF according to the element's valence. Purposive behaviorism is possible by neoRL agents, allowing for autonomous navigation in Euclidean space.

The development process of the neoRL framework involved reviewing discussions on the credit assignment problem as seen from behaviorists. Tolman(1948) reports two schools with different explanation of the credit assignment for learning, the temporal difference school and the persistence school of behaviorism. The temporal difference school is analogous to TD learning from RL, whereas the persistence school regards persistent co-activation between stimuli as the main mechanism for learning. Section 4.1.3 develops *inference inversion* for RL, inventing an update rule based on the persistence school and memory traces. With every state update $s^- \rightarrow s^+$, inferences are updated according to how active the corresponding event was before state transition. Learning by inference inversion in RL gives credit to events that persistently takes part in some operant inference. Since learning by inference inversion, in effect, operates independently of action selection, this learning paradigm allows for inference learning outside the constraints of having to consider a predefined action set. All events that persistently take part in operant inferences should be considered; credit assignment by inference inversion can learn inferences outside a predefined action set.

The separation between what drives learning from what drives the inference being learned allows for temporal abstraction in the environment. Inferences learned for one NRES representation of a problem can be combined with other inferences, as seen from multimodal neoRL experiments in manuscript B. "Representing knowledge flexibly at multiple levels of temporal abstraction has the potential to greatly speed planning and learning on large

problems" [68]. With the flexibility allowed by the superposition theorem and the observed ability of the neoRL framework to combine value functions across NRES representations, inference inversion allows for temporal robustness across multiple NRES sets. Although it can be challenging to grasp the significance of considering multiple NRES in the same agent, the direct relation between larger receptive fields and a slower time scale is prominent. The neoRL framework allows for latent learning across different levels of temporal abstraction, as represented by the different Euclidean resolutions seen in manuscript A.

By considering mono-NRES behavioral modules as nodes in a network, the agent design in papers A and B can be regarded as one-layered networks by which agent value function is governed. In the Euclidean interpretation, where state-action values are interpreted to projections of desire, neoRL networks are conceivable where deeper nodes are motivated by earlier projections of desire. Where inference inversion and a separation between learning and effectuation in RL could be a first step toward autonomous navigation, further development and verification of deep or recursive neoRL graphs for purposive networks is a crucial second step toward behavioristic AI. Navigation by neoRL networks is promising for category II autonomy by behavioral neoRL networks.

Online navigation autonomy is possible by combining OVF scaled by elements-of-interest valence, allowing for marginally improved navigation capabilities over Brownian motion in the WaterWorld environment. Proficiency of the neoRL agent scales well with additional NRES layers. Assessing additional NRES maps in the same agent improves navigational capabilities significantly, does not degrade learning performance, and only results in a linear increase in computation. A multi-resolution and multimodal neoRL agent performs well for autonomous navigation in the WaterWorld environment, implying that neoRL would scale beyond the WaterWorld environment and into the real world. The neoRL framework is capable of producing autonomous Euclidean navigation in real-time, verifying earlier findings on the importance of distributed processing and learning for autonomous learning.

Chapter 5.

Computational cognitivism by navigation

> Science is everything
> we understand well enough
> to explain to a computer,
>
> art is everything else.
>
> *Donald E. Knuth*

This project has explored the basic principles of autonomous navigation, aiming to understand the principles of autonomy well enough to explain them to the computer. No current technology was found to be satisfactory for autonomous navigation by Definition 1 and 2; only living entities are capable of the required degree of autonomy, making neuroscience and the psychology of learning key to understanding autonomous navigation. A salient difference between neural control mechanisms and digital technology lies in the distributed nature of the former; whereas digital learning technology can be modelled as monolithic systems with defined input and output, neural systems are distributed across countless nodes – each with distinct inputs and outputs. Consequently, biological computation happens in spatiotemporal networks without any clear distinction between the computational state of the networks and the transient result of the computation. First, we explored the distributed nature in neural representation of Euclidean space (NRES), to find out how NRES is possible by a distributed pattern of activation based on geometric conditionals in a Euclidean space. When information expressed as a coordinate of the NRES modality lies within the receptive field of one NRES cell, this cell produces a positive definite output signal from this node. Second, off-policy RL methods can be applied to learn general value functions (GVF) according to scalar intent signals. Separate GVF learners with NRES activation signals as intent (reward) can learn orthogonal operant value functions (OVF) in one NRES map. The set of OVF from one NRES instance can form a digital analogy to a cognitive map, allowing the agent to

extract purposive policies by elements-of-interest associated with reward. The resulting agent of NRES-oriented RL, referred to as neoRL, implements Tolman's cognitive architecture by latent learning and purposive policy extraction. The change from considering Thorndike's conditioned reflexes, to instead let Tolman's purposive neobehaviorism inspire behaviorial AI, might have a similar impact on behavioral AI as neobehaviorism have had for cognitive psychology.

First of all, a comprehensive search for the neuroscience of navigation and the psychology of learning and autonomy was conducted, summarized in Chapter 2. Section 2.1 starts by introducing the Jamesian school of functionalism and Edward Lee Thorndike's *law-of-effect* from the very beginning of psychology as a science. The law-of-effect and S-R learning by reinforcement are of particular interest for their role in inspiring RL in AI. Although the theory of conditioned reflexes was accepted as a model explanation for reflexive actions, the S-R mechanism could not explain the complexity of human behavior. The behaviorist movement attempted to explain human behavior by expanding S-R reinforcement theory, but was only partially successful at the theoretical level – as introduced in Section 2.1.1. Edward C. Tolman modernized behaviorism by introducing stateful computation, latent learning, and cognitive maps in his *purposive behaviorism*. B. F. Skinner further introduced the concept of *operant behavior*, stating that an agent could activate operant reflexes *to achieve something*. Mechanisms similar to cognitive maps have been verified by modern neuroscience, resulting in the 2014 Nobel prize in physiology or medicine "for their discovery of cells that constitute the positioning system in the brain" [56]. Neural representation of Euclidean space (NRES) has been summarized for several navigational modalities in Section 2.2, leading to a discussion on how navigational state is distributed across multiple NRES, each representing one aspect of navigational state by distributed patterns of activation. Skinner's operant desires and Tolman's latent learning, together with concepts from NRES and cognitive maps, have been necessary for developing the neoRL framework for research objective three.

The second research objective motivated a study of how adaptive algorithms are implemented, and how learning could be emulated in the otherwise deterministic computer. An overview of computing sciences that allows for adaptive behavioral AI, including perceptron-class function approximation and algorithms inspired by Thorndike's law-of-effect, is presented in Chapter 3. Adaptive algorithms by RL in AI learn one behavioral schema by interaction learning based on a scalar reward signal, analogous to early functionalists' view on reinforcement of conditioned instincts. Off-policy derivates of RL can be used for learning auxiliary value functions, indicating the possibility of distributed value functions for RL agents. First, Wiering and Van Hasselt (2008) explored how ensemble methods for RL could be effective in toy problems; the proficiency of agents governed by voting schemes

was compared to others governed by Boltzmann addition and multiplication. Van Seijen et al. (2017) further demonstrated how the agent value function could be combined across multiple learners for separate concerns; the *hybrid reward architecture* allowed for autonomy in solving the complex and previously unsolved Ms. Pac-Man by a linear combination of value functions from multiple learners. A selection of important findings from RL and AI are reported in Chapter 3 – elements considered important in our attempt to implement purposive behaviorism for AI by NRES-oriented RL.

In accordance with the third research objective, as a method for better understanding the basic principles of autonomous navigation, an agent capable of autonomous navigation in Euclidean space was implemented. After the fundamentals of neoRL had been developed and implemented, Richard Sutton invited me to continue the project under his supervision at the RLAI lab. Section 4.1.1 proposes how an MDP can be considered as a biased white-noise process, expressing learning as a parameter search for the optimal bias for the stochastic decision process. Wold's representation theorem further allows the decomposition of a stochastic decision process into one stochastic and one deterministic component; accordingly, one could model parts of the presumably multivariate and complicated basis of autonomy as being stochastic while still being able to capture deterministic aspects. The Arnold-Kolmogorov representation theorem further allows for decomposition of a multi-variate deterministic component into a set of simple mono-variate functions. The deterministic part of the agent value function can be decomposed into simpler, learned, concerns – as introduced in Section 4.1.2. The theory-intensive section is rounded off by introducing *inference inversion* for RL, an updated model based on the alternative school of behaviorism to credit assignment by temporal difference. Although a proper introduction to time in RL is planned for future publications, Section 4.1.3 proposes a separation between what drives learning and what drives the inferences in RL. Inference inversion is believed to increase temporal abstraction capabilities and robustness for RL algorithms.

The neoRL agent is put to test in a reactive navigation challenge, introduced in Section 4.2, prior to summarizing the results in 4.3. An agent built by the proposed principles in Chapter 4 demonstrated both how autonomous navigation is plausible, while also uncovering additional knowledge for neoRL autonomy. First, the concept of OVF and desire aggregation by the Kolmogorov-Arnold theorem allowed for purposive behavior by a single-resolution NRES-agent. The results are displayed in Figure 4.4, revealing a strong correlation between NRES resolution and agent performance. Second, the performance of a multi-resolution agent was compared to the single-resolution agent, resulting in a significant improvement in navigational proficiency. These findings are reported in paper A. The neoRL agent appears to be capable of combining knowledge from latent learning across multiple state spaces;

paper B explores the capabilities of neoRL for inter-modal navigation. The neoRL framework is general across NRES modalities, compositional and incremental by additional neoRL modules, and effective across orthogonal NRES-modalities. Third, the neoRL framework for Euclidean navigation can form purposive graphs – graph structures where experience-based desires can establish projections of desire for neoRL nodes. Manuscript C discusses the possibility of deeper behavioral neoRL networks. Experimental results demonstrate how elements of desire in one neoRL learned space can propagate to compatible neoRL nodes by forming autonomous projections of desire. Figure 4.7 exemplifies how deep or recurrent desires can significantly improve neoRL navigational autonomy, making online autonomy plausible for the real-world.

5.1. Conclusion

Autonomous navigation is plausible for agents emulating identified principles from psychology and the neuroscience of navigation. Theoretical and experimental results emphasize two principles. First, the superposition principle appears to be have an equally important role for behavioral autonomy as with engineering disciplines and in natural sciences. Whereas a single-NRES node demonstrates a slight proficiency for autonomous navigation, this proficiency scales well with additional NRES behavioral maps. Multiple behavioral components can be combined across from different NRES of different resolution, representing a behavioral analogy to function representation by series expansion. Then desire from a neoRL node is expressed by latent learning across multiple NRES state representations with different resolution, navigational performance improves significantly.

Second, the neoRL framework demonstrates the importance of Tolman's purposive behaviorism – verifying the implications of separating learning from behavior on behavioral autonomy. Whereas knowledge is represented by latently learned OVF, purpose is extracted according to the agent's projections of desire. Different sets of desire, defined as the input to the neoRL node, can induce different behavioral components expressed by the neoRL node. Projections of desire can be established as a Euclidean interpretation of agent state-action values, allowing for self-driven purpose and deeper structures of desire. Deep and recursive structures for desire are demonstrated by experiments, validating the potency of multi-layered behavioral graphs on autonomous navigation.

In humble respect and admiration of the wise (wo)men of old, this contribution only seeks to attain a higher understanding of autonomy. Impressive insights by Skinner and Tolman, combined with great works by Sutton et al. (2011) and Van Seijen et al. (2017), have established a foundation upon which these results are built. Whereas earlier approaches

rooted in dynamic programming have struggled with an exponential decrease in learning efficiency with more information, the neoRL framework demonstrates an *increase* in learning efficiency when more information is available for the same task. *Improvisation* could be a valid interpretation of problem solving based on auxiliary information, an explanation that could have implications for the understanding of intelligence – possibly suggesting that neoRL expresses an emulated rather than an artificial intelligence.

Part III.

Papers

Appendix A.

Decomposing the prediction problem;
autonomous navigation by neoRL agents.

Per R. Leikanger.
ALIFE 2021: The 2021 Conference on Artificial Life, MIT press, 2021.

Decomposing the Prediction Problem; Autonomous Navigation by neoRL Agents

Per Roald Leikanger

UiT – Norges Artigste Universtet
Per.Leikanger@uit.no

Abstract

Navigating the world is a fundamental ability for any living entity. Accomplishing the same degree of freedom in technology has proven to be difficult. The brain is the only known mechanism capable of voluntary navigation, making neuroscience our best source of inspiration toward autonomy. Assuming that state representation is key, we explore the difference in how the brain and the machine represent the navigational state. Where Reinforcement Learning (RL) requires a monolithic state representation in accordance with the Markov property, Neural Representation of Euclidean Space (NRES) reflects navigational state via distributed activation patterns. We show how NRES-Oriented RL (neoRL) agents are possible before verifying our theoretical findings by experiments. Ultimately, neoRL agents are capable of behavior synthesis across state spaces – allowing for decomposition of the problem into smaller spaces, alleviating the curse of dimensionality.

Introduction

Autonomy or any form of self-governed activity implies an ability to adapt with experience; hard-coded algorithms, agents governed by external control, or deterministic model-based path planning can hardly be said to be autonomous. "Navigation can be defined as the ability to plan and execute a goal-directed path" (Solstad, 2009). Robot motion planning can be defined in similar terms (Latombe, 2012); however, cybernetics and robot motion control involves model with limited validity intervals or algorithms for deterministic control. The reward hypothesis from Reinforcement Learning (RL) is relevant in this context: *"That all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)." (Sutton and Barto, 2018).* With a proven track record for learning to solve digital challenges or for intelligent games, RL agents have demonstrated a capability of autonomy for specific challenges. Via methods from function approximation by Deep Learning, methods from RL can form agents with superhuman abilities for certain board games (Tesauro, 1994; Silver et al., 2016, 2017) and games of hazard (Heinrich and Silver, 2016). However, RL supported by deep function approximation is known to require a tremendous amount of training: Robot autonomy by RL remains an unsolved challenge, partially due to requirements for real-time execution and model-uncertainty – limiting the number of accurate samples for training (Kober et al., 2013). RL agents supported by deep function approximation can learn impressive abilities, but statistical machine learning approaches require much experience, do not generalize well, and are monolithic during training and execution (Kaelbling, 2020).

Autonomous navigation is an ability unique to the central nervous systems in the animal and insects. Determining one's parameter configuration relative to an external reference, one's *allocentric* coordinate, is critical for navigation learning (Whitlock et al., 2008). Several mechanisms have been identified in the brain that represent Euclidean coordinates at the single-neuron level (Bicanski and Burgess, 2020). Notable examples for navigation are Object Vector Cells (Høydal, 2020), representing the allocentric location of objects around the animal, Head-Direction Cells (Taube et al., 1990), representing the heading of the animal, and border cells (Solstad, 2009), representing the proximity of borders for navigation. Possibly the most well-known cell for Neural Representation of Euclidean Space (NRES) is the *Place Cell*. This first identified NRES modality represents the allocentric location of the animal (O'Keefe and Dostrovsky, 1971): When an animal's location is within the *receptive field* of one place cell, the neuron is active in terms of having a heightened firing frequiency. The activation pattern in an appropriate population of NRES neurons can thus map any position in a finite Euclidean space (Fyhn et al., 2004). Other NRES modalities have later been identified, with a similar mechanism for representing coordinates in other Euclidean spaces (Bicanski and Burgess, 2020). With our sense of orientation originating from multiple NRES modalities, distributed representation of state appears to be of critical importance for navigational autonomy.

This article starts out by presenting important considerations from RL and directions that could allow for a distributed representation of state. Off-policy learning allows

agents to learn general value functions for independent aspects of a task (Sutton et al., 2011). When a hoard of learners base their value function on a mutually exclusive reward signal, inspired by NRES cells, we propose a method for learning an orthogonal basis for behavior. Experiments with NRES-Oriented RL (neoRL) agents by the Place Cell NRES modality demonstrate how the proposed framework allows for reactive navigation in real-time.

Interaction learning by RL in AI

Reinforcement learning is the direction in machine learning concerning learning behavior through interaction with an environment. We say that the decision *agent* learns to achieve a task according to a scalar reward signal $\mathbb{R}$ by interaction with an *environment*. The accumulated experience takes the form of agent *value function*, reflecting the benefit of visiting different states or state-actions pairs according to the *reward signal* during training. When the algorithm learns the value of state-action pairs, i.e., learning the value of selecting specific actions from different states, this is referred to as Q-learning. An important aspect of RL environments is the *Markov property*: When a state-action pair uniquely defines the probability distribution of the next state, the decision process is referred to as a Markov Decision Process (MDP). When a problem can be represented as an MDP, an RL-agent can, in theory, learn an optimal solution to tasks expressed by a reward function from interaction alone (Sutton and Barto, 2018).

The *prediction problem* in reinforcement learning concerns estimating the value of visiting different states s while following policy π. The agent state is a compact representation of the history and necessary information for the agent to make a decision at time t. The value function can be updated according to the Bellman equation:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right] \quad (1)$$

Updating the value function under policy π from experience gathered while following the policy π, is referred to as on-policy learning (Sutton and Barto, 2018). Off-policy learning allows an agent to form the value function while following another behavior policy. Through off-policy learning, an agent can learn the value function under a target policy π_t while following a different behavior policy $\pi_b \neq \pi_t$. The agent can, for example, initially follow a more exploratory policy or learn while observing human control (Abbeel et al., 2007). Learning the value function is possible through pure observation.

General Value Function (GVF) is one identified use of off-policy learning, where the agent learns value functions potentially unrelated to the control problem (Sutton et al., 2011). These partial agents, only concerned with accumulating experience, can be seen as independent *learners* of an auxiliary value function used to answer questions about the environment. Examples of questions, as listed in the original paper, could be time-to-obstacle or time-to-stop for the Critterbot demonstration (Sutton et al., 2011). Auxiliary value functions can also be directly involved in policy, as demonstrated for the Atari game Ms. PacMan. A set of General Value Functions were trained for manually designed subchallenges in the Ms. Packman computer game, resulting in an exponential breakdown of problem size compared to "single-headed" RL agents (Van Seijen et al., 2017). Wiering and Van Hasselt (2008) gave a methodological overview over ensemble methods for integrating experience from multiple algorithms when forming policies. Notably, Boltzmann addition and Boltzmann multiplication could integrate policies from multiple sources before action selection (Wiener, 1948). Both Wiering and Van Hasselt (2008) and Van Seijen et al. (2017) propose ways multiple off-policy learners could be involved in forming policy. From these demonstrations on how multi-learner agents are possible, we shall dive further into the mechanism of behavior synthesis. But first, some neuroscience.

Neural Representation of Euclidean Space

The 1906 Nobel price in physiology and medicine was awarded Santiago Ramón Y Cajal for work initiating the neuron doctrine (Ramón y Cajal, 1911), claiming that behavior originates from a network of cells with signaling capabilities rather than a monolithic soul. The neuron doctrine supplied a mechanistic understanding of biological computation as a distributed network of weak computational units. Only by network phenomena and a delicately connected net of neurons can decisions, policies, and ultimately behavior emerge. Eric Kandel later reported how synaptic connections change with use and how learning and memory are consequences of synaptic plasticity (Kandel and Tauc, 1965). Before the neuron doctrine, the consensus was that behavior and decision-making originate from a monolithic entity that followed us in this life and beyond – *the soul*.

Neural Representation of Euclidean Space (NRES) have been reported for different Euclidean spaces on a per-neuron cellular activation: when the Euclidean coordinate falls within the receptive field of an NRES neuron, the neuron fires with a heightened firing frequency. A growing number of NRES modalities have been identified, with notable examples for navigation being place cells (O'Keefe and Dostrovsky, 1971), head-direction cells (Taube et al., 1990), and object-vector cells (Høydal, 2020). While some NRES neurons have simple receptive fields centered around a coordinate, others have complicated repeating shapes like the hexagonal pattern of *grid cells* (Moser et al., 2008). For a comprehensive review of NRES modalities identified in neuroscience, see (Bicanski and Burgess, 2020).

Neural state is very different from the monolithic state of RL. Analogous to separate cells representing coordinates of one Euclidean space, separate NRES modalities reflect

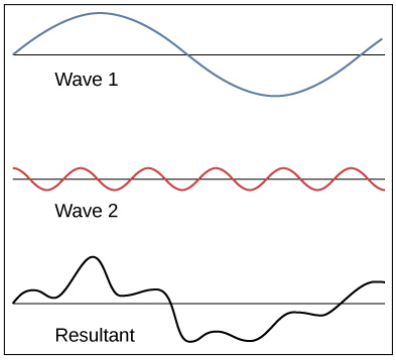


Figure 1: Two simple sinusoidal functions can be combined to a complex function by superposition. (Ling et al., 2016)

different aspects of the navigational state. The receptive fields of NRES neurons have a systematic increase from the dorsal to the ventral pole of the hippocampus (Fyhn et al., 2008; Kjelstrup et al., 2008; Solstad, 2009), allowing for NRES maps of multiple resolutions in parallel. The fully distributed representation of state thus allows for learning state representation by individual receptive fields, for different NRES resolutions and across NRES modalities in parallel. The monolithic Markov state of RL (Sutton and Barto, 2018), on the other hand, could explain difficulties for robot interaction learning (Kaelbling, 2020). The most protruding difference between AI and neural state representation lies in the distributed nature of NRES. We now explore how this can be emulated for RL systems.

Decomposing the Prediction Problem

The purpose of an *agent* in reinforcement learning is to establish a proper behavior as defined by a reward signal. The agent improves behavior based on two intertwined aspects of experience: (1) The *prediction problem* for learning the value of visiting states or state-action pairs as defined by the environment representation, and (2) The *control problem* for selecting the most appropriate action based on the value as learned by the prediction problem. In this section, we expand on the concept of the prediction problem by considering the value function as a potential field across orthogonal reward signals.

Let Orthogonal Value Functions (OVFs) be value functions of the state space $\mathbb{S}$ that adhere to mutually exclusive reward signals in $\mathbb{S}$. A relevant analogy would be to think of the value function as a potential field between different sources of energy. With multiple forces working on an object, the resultant work can be found as a linear combination of components. Similarly, a set of independent reward functions in $\mathbb{S}$ acting on agent value function can form a basis for agent value function in $\mathbb{S}$. NRES with mutually exclusive receptive fields is a good candidate for independent reward signals; with the place cell as our leading example, it is simple to visualize how agent position activates receptive fields and OVFs. Each *learner* has a simple reward shape, with a positive reward of $\mathbb{R} = +1$ upon activation of the corresponding NRES cell and $\mathbb{R} = 0$ otherwise. A separate learner form the OVF according to reward signals as defined by mutually exclusive receptive fields of $\mathbb{S}$.

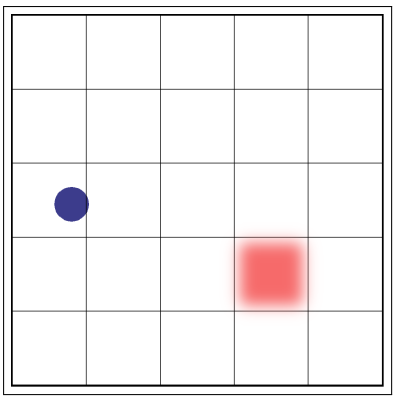


Figure 2: An agent in $N5$ allocentric place-cell representation of Euclidean space: An $N5$ representation involves that each axis is divided into 5 equal intervals. A learner could, for example, form the OVF toward cell $(4, 4)$, with a reward signal defined by the activation of the corresponding NRES cell. The reward function of this particular learner is illustrated in red for feature $s_R \in \mathbb{S}$. The current parameter configuration of the agent defines from which $s \in \mathbb{S}$ this NRES modality's value function is extracted.

Let there be K individual learners, one for every receptive field of an NRES representation $\mathbb{S}$. With mutually exclusive receptive fields, the set of learners in $\mathbb{S}$ can be considered an orthogonal basis of the value function in this representation. Value functions of $\mathbb{S}$ can be expressed as a linear combination of OVFs formed by the K learners, allowing a neoRL agent to synthesize a range of behaviors. The challenge of learning apt behavior now reduces to learning priorities between policies expressed via OVFs. Estimating scalar values based on supervised samples is a well-studied field in machine learning. However, for the sake of clarity, static priorities defined by the associated reward is used.

The Control Problem by Superposition

The motivation for learning the value function is ultimately to form an effective policy for the challenge at hand. A simple challenge in Euclidean space can be for the agent to move to one particular position, activating feature s_x. If learners use Q-learning to establishing a potential that contributes to the *Q-field* of the agent, the next action can be chosen by

$$a = \text{argmax}_a Q_{tot}(s, a)$$

where Q_{tot} is the resultant Q-field of the current situation. With a single learner as input to the agent value potential,

the agent's prediction problem becomes equivalent to that of the single learner, and the mechanism surrounding the value function of the agent simplifies to that of a monolithic agent.

For slightly more interesting challenges, multiple rewards can be expressed in the decomposed NRES representation. Each learner can be said to represent one consideration in this environment, learning the value function related to activating the corresponding NRES cell. When multiple considerations have priority, the superposition principle allows the Q-field to form over relevant OVFs.

$$Q_{tot}(s,a) = \sum_{i \in \mathbb{S}_R} Q_{\mathcal{L}i}(s,a) \tag{2}$$

where $\mathbb{S}_R$ is the set of NRES cells associated with reward and $Q_{\mathcal{L}i}(s,a)$ represent learner $\mathcal{L}_i$'s value component. The K learners in the full features set can thus be considered to be *peer learners* for the task of navigating the environment representation.

$$\mathbb{S}_R = \{s \in \mathbb{S} \;\Big|\; |\mathbb{R}_s| > 0\}$$

An elegant approach would be to consider rewards to be linked to elements of interest in the environment rather than allocentric features: Let an *Element of Interest* (ξ_i) be an instance in the environment associated with a reward. Assume for now that the priority and Euclidean parameter configuration of every element of interest in the set $\mathbb{E} = \{\xi_i\}$ is provided by the environment. Any parameter configuration is possible to map uniquely to the mutually exclusive NRES feature map $\mathbb{S}$. With element i's importance w_i proportional to the reward associated with the element activating feature s, the corresponding peer learner's contribution to the Q-field becomes:

$$Q_{tot}(s,a) = \sum_{i \in \mathbb{S}_R} w_i Q_{\mathcal{L}i}(s,a) \tag{3}$$

Isolating rewards that comes from elements of interest, i.e. abstaining from utilizing timestep rewards or other shaped rewards, the set of rewarded states is defined by the set of NRES cells occupied by an element of interest ξ_i.

$$\mathbb{S}_R = \{s \in \mathbb{S} \;\Big|\; \exists \xi_i \in \mathbb{E},\ \xi_i \in s\} \tag{4}$$

Note that an element of interest can be any element associated with a reward in a particular state set representation, decoupling the prediction problem in an environment from the rewards of one task. Experience expressed by distributed Q-fields is more general than monolithic value functions; In the neoRL approach, moving rewards or changing agent priorities during an agent's life-time does not require retraining the agent.

Experiments

Algorithms in RL learn behavior by interaction with the environment, making the environment defining for the outcome of any RL experiment. Numerous environments exist to highlight challenges for state-of-the-art reinforcement learning agents. Learning autonomous navigation in allocentric space does not seem to get much attention, as finding appropriate test-environments can be difficult. Preferably, an environment for autonomous real-world navigation learning is represented by continuous allocentric coordinates and with a complexity that requires reactive navigation. Real-time execution would be a plus since it limits the amount of training data available to the agent to a realistic order of magnitude. Physical systems generally depend on temporal aspects like inertia. Most of these qualities can be found in Karpathy's WaterWorld challenge.

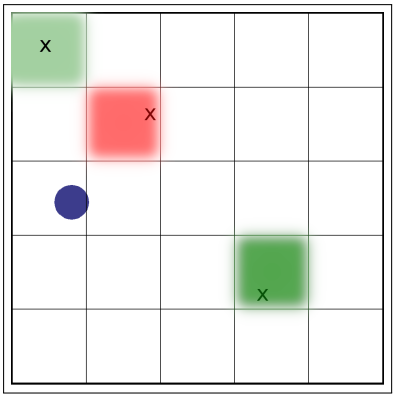


Figure 3: Element of Interest (EoI) activates desires for allocentric features according to their importance: An EoI situated in feature $(4,4)$ makes this desirable with 1.0 , another positive EoI activates feature $(1,1)$ with priority 0.5 , as represented by a green with lower saturation. An aversive element located in feature $(2,2)$ activates the corresponding learner with a negative weight $w_i < 0$.

WaterWorld

Karpathy's WaterWorld challenge as implemented in Pygame learning environment(PLE) (Tasfi, 2016) is an environment with a continuous 2D resolution, inertia dynamics and external considerations referred to as *creeps*. Creeps move with a constant speed vector, reflected when hitting a wall. Creeps have a demeanor, as illustrated by color: green creeps are desirable with [+1] reward, and red creeps are repulsive with [-1] reward upon capture. When the agent captures a creep, a new one is initialized with a random speed, position, and demeanor – causing a chaotic scenario that requires reactive navigation. When all green creeps have been captured, the board is restarted with an accompanying [+5] reward. In all experiments, a constant number of 8 creeps have been used, as illustrated in Figure 4. We find the allocentric PyGame implementation (Tasfi, 2016) of WaterWorld appropriate for RL research for real-time navigation autonomy. However, the environment is listed as unsolved

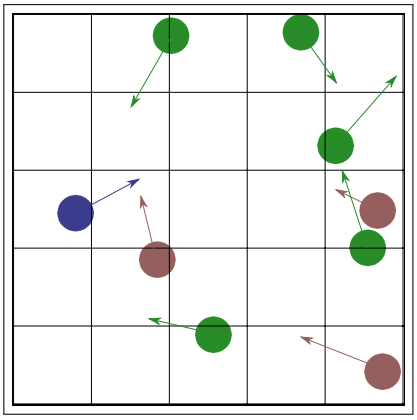


Figure 4: NRES $N5$ representation of Element-of-Interest (EoI) in the *WaterWorld* environment. Each EoI and the location of the agent represented in the PlaceCell NRES modality. Red and Green represent the demeanor of each creep, whereas Blue represents the current agent location. In addition, arrows have been drawn to illustrate the current speed vector of each element.

(OpenAI, 2020) – making comparisons to alternative solutions difficult.

Instantaneous information regarding elements-of-interest (EoI), i.e., the position and demeanor of each creep, is provided by the environment. Demeanor defines the reward associated with the creep, crucial for priority w_i associated with EoI i by equation 3. Positions are represented in 2D allocentric coordinates from the environment, allowing for extracting $\xi_i \in \mathbb{S}$ for the Place Cell NRES modality of EoI i. Basal actions affect the agent by accelerating it in the cardinal directions, [N, S, E, W].

Allocentric Position Modality, Single layer: Our primary assumption is that the agent value function in effect can be considered a potential field across OVFs, pulling the agent toward the next decision. Our first experiment explores to what degree the superposition principle holds for the value function of individual learners. We compare the accumulated score of neoRL agents based on single-res NRES to Brownian motion, i.e., an ϵ-greedy policy with $\epsilon = 1.0$. Under the convention used in Figure 4, where $N5$ signifies an NRES map with $5x5$ tiles, five different resolutions are explored from $N10$ to $N90$. All experiments were conducted over 150.000 time-steps for each neoRL agent.

Allocentric Position Modality, Multiple resolutions: Our second experiment explores how integrating experience across multiple state spaces affect neoRL performance. An interpretation of the progressive increase for receptive fields in the ventral direction of the hippocampus is that different NRES maps exist with different resolutions. We adopt this view in experiment 2, where we let the neoRL agent combine value function across multiple NRES state representations. In this experiment we assess whether the neoRL agent is capable of forming apt policies by integrating experience across multiple state spaces. We compare the proficiency of a multi-res neoRL agent that learns over $\{N3, N7, N23\}$ NRES state spaces to three single-res agents by $N3$, $N7$, and $N23$ NRES. The neoRL agent layout is illustrated in Figure 5. Prime numbers are used as the resolution for each layer, minimizing the potential for overlapping boundaries. The resulting 587 learners in the multi-res agent learn in parallel by off-policy learning. In this setup, the contribution of each learner is inversely proportional to the size of its receptive field.

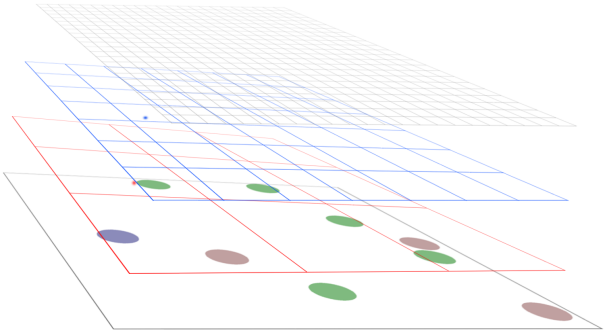


Figure 5: Illustration of multiple state representation in the decision agent, where each tile represent the objective of its respective learner. [Red] N3 representation [Blue] N7 representation [Black] N23 representation.

One approach of measuring the proficiency of the agent is as the per-timestep average reward across parallel runs. We are interested in real-time learning efficiency and initialize a neoRL agent with no priors at the beginning of each run. A per-timestep average across 100 independent runs provides information about the transient timecourse in navigation capabilities. Note that every run starts with a separate neoRL agent with no prior experience. All experiments are conducted on an average desktop computer, with one run taking somewhat under one hour on a single CPU core.

Results

Results are reported as real-time execution of agents as they learn, without any previous experience at the task. Reported resolution for each experiment adheres to the convention from Figure 2, dividing each axis of the Euclidean space into N steps. The x-axis of all plots represents the number of time steps since the beginning of a run, i.e., the real-time execution in time-steps since initiation of the agent.

Allocentric Position Modality, Single layer A distributed representation of the Markov state is plausible for neoRL agents. Figure 6 shows the accumulated score of

neoRL agents with NRES Place Cell representations from $N10$ to $N90$. All neoRL agents perform better than control. Brownian motion seems incapable of achieving a single board reset since the accumulated score fluctuates around 0 for the length of the experiment. All neoRL agents are capable of accumulating a significant amount of experience, verifying that OVF can function as a basis for synthezing successful behavior.

A strong correlation between NRES resolution and proficiency at the task can also be observed in Figure 6. The immediate proficiency at the task can be seen from the steepness of the curve. Agents based on lower NRES resolution initially learn quicker than agents with higher NRES resolution. However, neoRL agents based on lower NRES resolutions seem to saturate at a lower proficiency. For these particular runs, with 8 creeps and during a 150.000 time step interval, the $N50$ representation appears to achieve the highest score. Although this number is task-specific, it is worth noting how all neoRL agents are comparable in learning speed. Despite $N70$ NRES having almost 50 times the dimensionality of $N10$[1], the two neoRL agents based on these representations are comparable in learning. This effect requires further attention.

Allocentric Position Modality, Multiple resolutions Combining the value potential from multiple representations of state can significantly increase navigation performance. The transient proficiency of the neoRL agent in the four experiments, $N3$, $N7$, $N23$, and multi-res $\{N3, N7, N23\}$, is presented in Figure 7. Each curve is the result of a per-timestep average over 100 independent runs. These results verify without any doubt that neoRL agents benefit from combining experience across multiple NRES feature sets. With the algebraic sum of the per-timestep proficiency of the three mono-res agents shown in grey, we see that the multi-res neoRL agent learns quicker, to higher proficiency, than the sum of its parts.

The superposition principle for behavior across state spaces seems to alleviate the curse of dimensionality: The almost 6-fold increase in the number of states (from $7^2 = 49$ to $3^2 + 7^2 + 23^2 = 290$ states) resulted in a 3.5-factor increase in received reward without increasing training time. Figure 7 shows that learning happens as fast or possibly a little faster for the multi-res agent than for the $N7$ mono-res agent. This effect could be defining for real-world interaction learning and requires further attention.

Discussion

Navigation autonomy is plausible in real-time by RL agents with an emulated neural representation of space. NRES-Oriented RL (neoRL) agents are possible due to developed theory on orthogonality in the value domain, allowing for behavior synthesis across multiple learners.

Whereas neural systems are capable of autonomous navigation, modern technology is not. The most protruding difference between these systems is how state is represented. Digital RL systems require a monolithic state concept, whereas neural systems work by patterns of activation. The Markov state in RL holds enough information to uniquely define the probability distribution of the next state (Sutton and Barto, 2018). The Markov decision process works well with deep function approximation, and RL agents supported by deep learning have mastered a selection of board games. However, deep RL agents require much training, do not generalize, and are neither incremental nor compositional (Kaelbling, 2020). With deep RL appearing to struggle with real-world interaction learning, we have looked elsewhere for inspiration. Evidence suggests that Neural Representation of Euclidean Space (NRES) represent Euclidean coordinates by activation patterns on the per-neuron level. An NRES set $\mathbb{S}$ with mutually exclusive receptive fields provides a set of orthogonal reward signals of $\mathbb{S}$. Utilizing these signals as reward signal for independent learners, the set of Orthogonal Value Functions (OVFs) form a basis for any reward function of $\mathbb{S}$. Experiments verify that NRES-Oriented RL (neoRL) agents are capable of forming skilled navigation while learning.

Considering this work as a plausibility study for neoRL navigation, we see at least three important directions for further study. Firstly, a thorough mathematical analysis on the relevance of orthogonality could be key for proper understanding of neoRL capabilities. Specifically, deriving the equations for how singular reward functions cause orthogonal value functions can cause a better understanding of behavior synthesis. In experiment 2, we have seen how different state-space representations of the same parameter set can improve performance. We believe the same to be possible for state spaces across different parameter spaces. Secondly, the priority w_i in Equation 3 remains static in this work but allows for a dynamic weighing of OVF based on importance. Directly learning the association between element i and global reward $\mathbb{R}$ would make neoRL learning comply to the reward hypothesis, and be an important continuation of this work. Lastly, all experiments conducted on the neoRL framework have yet been with the WaterWorld environment. The WaterWorld represents a quite general task in a highly general Euclidean space across undefined parameters. Many would find it more interesting with a tangible demonstration in a more specific Euclidean space, e.g., navigation of the joints' angles in a robot manipulator task. A most important next step would be to demonstrate neoRL navigation for other Euclidean spaces, e.g., for maritime autonomy, (learned) autonomous driving, or for adaptive control of robot manipulators.

[1]The $N10$ representation is comprised of 100 receptive fields, whereas the finer $N70$ resolutions have 4900 receptive fields.

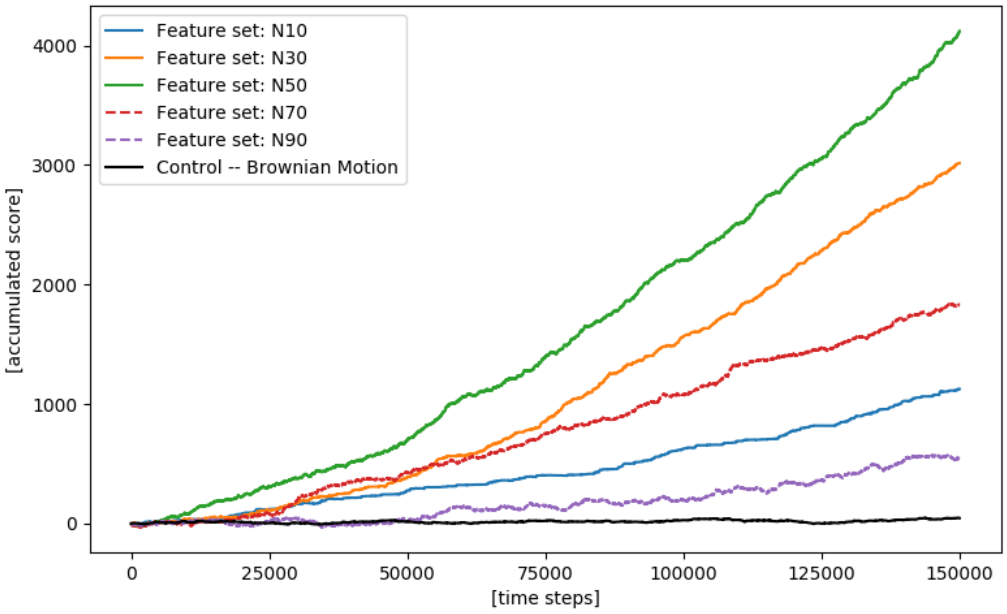


Figure 6: Accumulated Reward by peer agents with elements of interest for runs with grid coding resolutions, $N10 - N90$ over 150.000 time steps. Brownian motion in black is believed to be comparable to a first run of an untrained Deep RL agent.

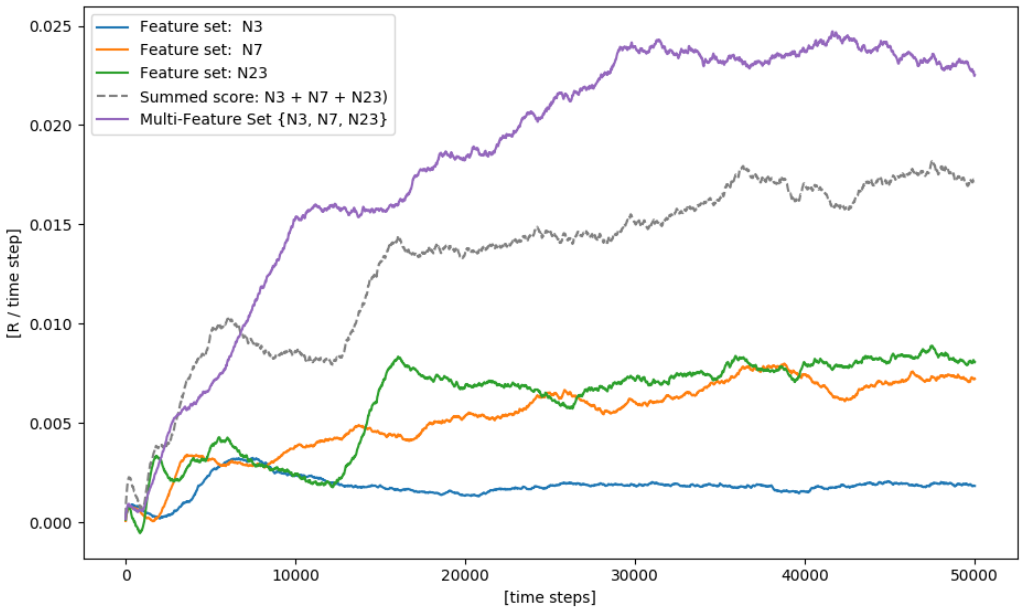


Figure 7: The neoRL agent is capable of incorporating experience from multiple state sets for navigation. A neoRL agent with experience from all three layers seen in Fig. 5 (purple) performs better than neoRL agents based on the individual NRES layer (blue, orange, green). The grey line represents the algebraic sum of the mono-res agents, highlighting that the multi-res neoRL agent performs better than the sum of its parts. Each curve is a presentation of the per-timestep average of 100 independent runs.

References

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1.

Bicanski, A. and Burgess, N. (2020). Neuronal vector coding in spatial cognition. *Nature Reviews Neuroscience*, pages 1–18.

Fyhn, M., Hafting, T., Witter, M. P., Moser, E. I., and Moser, M.-B. (2008). Grid cells in mice. *Hippocampus*, 18(12):1230–1238.

Fyhn, M., Molden, S., Witter, M. P., Moser, E. I., and Moser, M.-B. (2004). Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264.

Heinrich, J. and Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.

Høydal, Ø. A. (2020). *Allocentric vector coding in the medial entorhinal cortex*. Unpublished PhD thesis, Kavli Insitute of Systems Neuroscience / Center of Neural Computation.

Kaelbling, L. P. (2020). The foundation of efficient robot learning. *Science*, 369(6506):915–916.

Kandel, E. R. and Tauc, L. (1965). Heterosynaptic facilitation in neurones of the abdominal ganglion of aplysia depilans. *The Journal of Physiology*, 181(1):1.

Kjelstrup, K. B., Solstad, T., Brun, V. H., Hafting, T., Leutgeb, S., Witter, M. P., Moser, E. I., and Moser, M.-B. (2008). Finite scale of spatial representation in the hippocampus. *Science*, 321(5885):140–143.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Latombe, J.-C. (2012). *Robot motion planning*, volume 124. Springer Science & Business Media.

Ling, S. J., Sanny, J., Moebs, W., Friedman, G., Druger, S. D., Kolakowska, A., Anderson, D., Bowman, D., Demaree, D., Ginsberg, E., et al. (2016). *University Physics Volume 1*. OpenStax.

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

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

OpenAI (2020). OpenAI, Snake v0.

Ramón y Cajal, S. (1911). *Histologie du système nerveux de l'homme et des vertébrés*, volume 2. A. Maloine.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Solstad, T. (2009). *Neural representations of Euclidean space*. PhD thesis, Kavli Insitute of Systems Neuroscience / Center of Neural Computation.

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

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems.

Tasfi, N. (2016). Pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment.

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

Tesauro, G. (1994). Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219.

Van Seijen, H., Fatemi, M., Romoff, J., Laroche, R., Barnes, T., and Tsang, J. (2017). Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5392–5402.

Whitlock, J. R., Sutherland, R. J., Witter, M. P., Moser, M.-B., and Moser, E. I. (2008). Navigating from hippocampus to parietal cortex. *Proceedings of the National Academy of Sciences*, 105(39):14755–14762.

Wiener, N. (1948). *Cybernetics: Control and communication in the animal and the machine*. Wiley.

Wiering, M. A. and Van Hasselt, H. (2008). Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):930–936.

Appendix B.

Navigating Conceptual Space;
a new take on Artificial General Intelligence.

Per R. Leikanger.
In International Conference on Articial General Intelligence, pages 116–126. Springer, 2021.

Navigating Conceptual Space; A New Take on AGI

Per Roald Leikanger( )

UiT - The Arctic University of Norway, Tromsø, Norway
Per.R.Leikanger@uit.no

Abstract. Edward C. Tolman found reinforcement learning unsatisfactory for explaining intelligence and proposed a clear distinction between learning and behavior. Tolman's ideas on latent learning and cognitive maps eventually led to what is now known as conceptual space, a geometric representation where concepts and ideas can form points or shapes. Active navigation between ideas – reasoning – can be expressed directly as purposive navigation in conceptual space. Assimilating the theory of conceptual space from modern neuroscience, we propose autonomous navigation as a valid approach for emulated cognition. However, achieving autonomous navigation in high-dimensional Euclidean spaces is not trivial in technology. In this work, we explore whether neoRL navigation is up for the task; adopting Kaelbling's concerns for efficient robot navigation, we test whether the neoRL approach is general across navigational modalities, compositional across considerations of experience, and effective when learning in multiple Euclidean dimensions. We find neoRL learning to be more resemblant of biological learning than of RL in AI, and propose neoRL navigation of conceptual space as a plausible new path toward emulated cognition.

1 Introduction

Edward C. Tolman first proposed cognitive maps for explaining the mechanism behind rats taking shortcuts and what he referred to as latent learning [25]. Tolman was not satisfied with behaviorists' view that goals and purposes could be reduced to a hard-wired desire for reward [4]. Experiments showed that unrewarded rats would perform better than the fully rewarded group when later *motivated* by reward [26]. Arguing that a reinforcement signal was more important for behavior than for learning, Tolman proposed the existence of a cognitive model of the environment in the form of a *cognitive map*. The mechanisms behind neural representation of Euclidean space (NRES) has later been identified for a range of navigational modalities by electrophysical measurements [3]. Further, the NRES mechanism has been implied for navigating *conceptual space* [5], a Euclidean representation where betweenness and relative location makes sense for explaining concepts [7]. Results from theoretical neuroscience indicate NRES' role in social navigation [17], temporal representation [6], and reasoning [2]. Cognitive maps for representing thought have received much attention in

B. Goertzel et al. (Eds.): AGI 2021, LNAI 13154, pp. 116–126, 2022.
https://doi.org/10.1007/978-3-030-93758-4_13

neurophysiology in the recent five years [2,5,17]. Navigating conceptual space as an analogy of thought could explain generalization and reasoning based on locality [7].

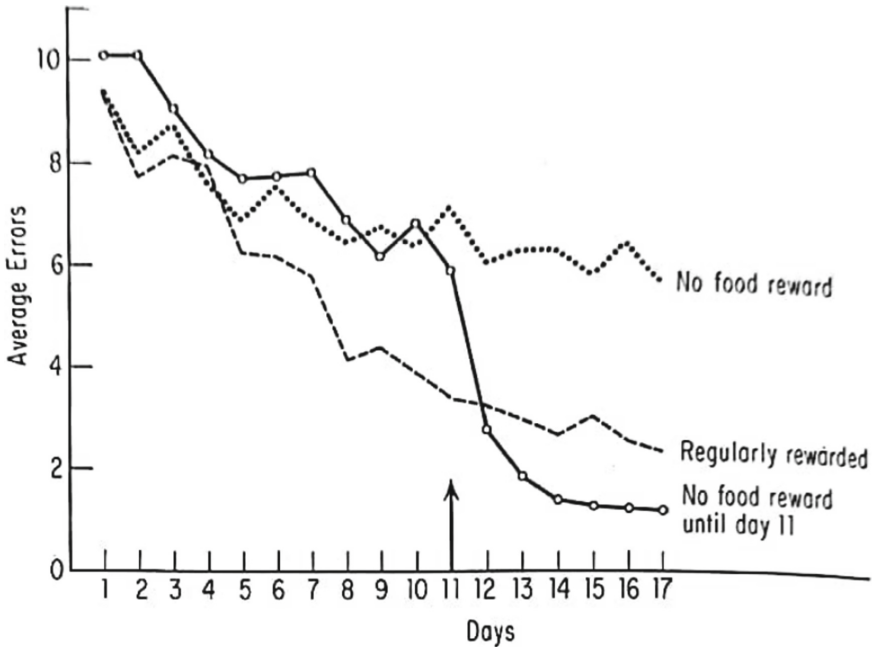


Fig. 1. Evidence for latent learning by Tolman and Honzik (1930). (after [26], from *Systems and theories of psychology* [4]).

Autonomous navigation is difficult to reproduce in technology. Autonomous operation implies a decision *agent* capable of forming decisions based on own desires and experience. A well-renowned approach to establish experience-based behavior is reinforcement learning (RL) from AI. Via trial and error based on a scalar reward signal $\mathbb{R}$, a decision *agent* is capable of adapting behavior according to the accumulation of $\mathbb{R}$. Considering robot path planning as Euclidean navigation, we look toward robot learning for inspiration on autonomous navigation. However; whereas RL powered by deep function approximation has been demonstrated for playing board games at an expert level, requirements to sample efficiency combined with high Markov dimensionality in temporal systems makes deep RL difficult in navigation learning [10]. Leslie Kaelbling (2020) points out key challenges for efficient robot learning, apparently concerned with the current direction of deep RL. Navigation has to be efficient (require few interactions for learning new behaviors), general (applicable to situations outside one's direct experience), and compositional/incremental (compositional with earlier knowledge, incremental with earlier considerations). The current state-of-the-art deep RL for robotics struggles on all three points [9].

Inspired by neural navigation capabilities, Leikanger (2019) has developed an NRES-oriented RL (neoRL) architecture for online navigation [12]. Via orthogonal value functions (OVF) formed by off-policy learning toward each cell of an

NRES representation, the neoRL architecture allows for a distinction between learning and behavior. Inspired by animal psychology, the neoRL framework allows purposive behavior to form based on the desire for anticipated reward [13]. However, navigating a multi-dimensional conceptual space of unknown dimensionality in real-time would be impossible for any current learning algorithm. In this work, we adopt Kaelbling's three concerns when testing whether neoRL navigation allows for autonomous navigation in high-dimensional Euclidean space.

2 Theory

Central to all navigation is knowledge of one's current navigational state. Information about relative location, orientation, and heading to objects that can block or otherwise affect the path is crucial for efficient navigation. When such knowledge is represented as vectors relative to one's current configuration, neuroscientists refer to this representation as being *egocentric*. When represented relative to some external reference frame, coordinates are referred to as being *allocentric*. Vectors can be expressed as Cartesian coordinates, e.g. the vector $\vec{a} = [1.0, 3.0]$ represent a point or displacement in a plane, one unit size from the origin along the first dimension, and three units along a second dimension. Vectors can also be represented in polar coordinates $\vec{a} = [r, \varphi]$, a point with distance r from the origin in the allocentric direction φ. In order to apply RL for navigation, all this information must be represented according to the Markov property; each instance of agent state must contain enough data to define next-state distribution [20]. Combined with temporal dynamics, the number of such instances becomes prohibitively expensive for autonomous navigation by RL [10]. Neural state representation, on the other hand, appears to be fully distributed across individual neurons and parts of the hippocampal formation [18]. NRES coding for separate navigational modalities (as should be represented in separate Euclidean spaces) have been located in different structures in the hippocampal formation [3]. Navigational state representation for the only system capable of true autonomous navigation seems to be decomposed across multiple NRES modalities. This section introduces theory and considerations on how state is represented in the animal and the learning machine, an important inspiration for neoRL mechanisms for navigation and problem solving.

2.1 Neural Representation of Euclidean Space

The first identified NRES neuron was the *place cell* [16]; O'Keefe and Dostrovsky discovered that specific neurons in the hippocampus became active whenever the animal traversed a specific location in the test environment. Reflecting the allocentric position of the animal, the individual place cell could be thought of as a geometric *feature detector* on the animal's location; the place cell is active whenever the animal is located within the *receptive field* of the cell. Other NRES cells have later been identified, expressing information in various parameter spaces. Identified NRES modalities for navigation includes: one's allocentric location

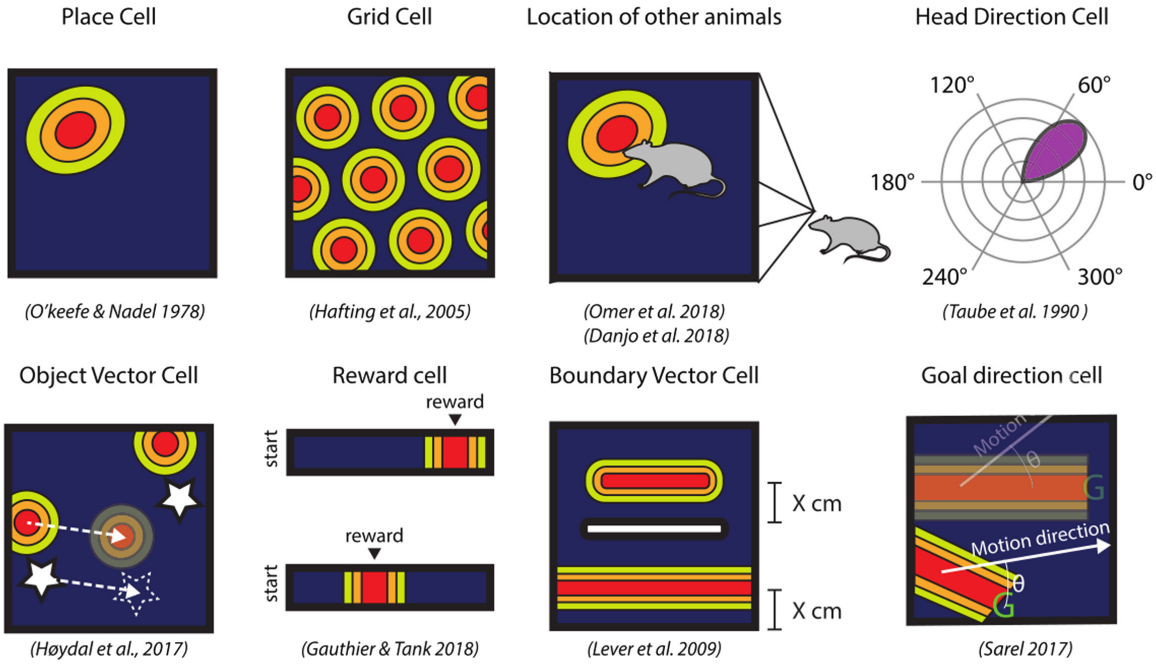


Fig. 2. Some identified NRES modalities of importance for navigation, with reference to the original publication. All NRES modalities could be important for autonomous spatial navigation. The place cell and the object vector cell will be of particular importance in the examples and experiments of this text. (Illustration adopted from [1]

[16], allocentric polar vector coordinates to external objects [8], and one's current heading [24]. A selection of relevant NRES modalities is listed in Table 1 or in Fig. 2. A more comprehensive study on NRES modalities in neurophysiology has been composed by Bugress and Bicanski [3].

Table 1. Neural representation for different Euclidean spaces of importance for navigation: Head-direction cells reflect the current allocentric (*ac.*) angle of the head (a scalar parameter). The place cell and border cell respond to a proximal allocentric location (2D). The remaining NRES reflect conditions represented in other Euclidean spaces – listed as NRES modalities.

	Location	Tuning	Direction	NRES modality	
Place cell	Ac.	[proximal] 2D	–	Current position	[16]
Border cell	Ac.	[proximal] 2D	–	Location of borders	[19]
Object vector cell	Polar c.	[spectrum] 2D	Ac.	Location of objects	[8]
Boundary vector cell	Polar c.	[spectrum] 2D	Ac.	Location of boundaries	[14]
Head-direction cell	–	[angular] 1D	Ac.	Head direction	[24]
Speed cell	–	[rate code] 1D	–	Current velocity	[11]

Neuroscientists assume that populations of NRES neurons map Euclidean vectors by neural patterns of activation. A simple mapping could be formed by

a population of NRES cells responding to mutually exclusive receptive fields. One could visualize this representation as a chessboard; exactly one cell (tile) would be satisfied for any point on the board. Referred to as *one-hot encoding*[1] in computing sciences, a mutually exclusive map structure is defined by the resolution and the geometric coverage of the map's tiles. This intuitive map is appropriate for demonstrative purposes: All examples and experiments in this text are encoded by a comprehensive one-hot mapping as illustrated in Fig. 3, where, e.g., $N13$ signifies a 13×13 tile set in $\Re^2$.

2.2 Autonomous Navigation by neoRL Agents

One can separate navigation into two distinct aspects; the desired location – the objective of the interaction – and how this objective can be reached. When both aspects are governed by one's own inclinations and *experience*, we refer to this as an autonomous operation. A most accomplished approach to experience-based behavior is RL from AI; a decision *agent* can be thought of as an algorithmic entity that learns how better to reach an objective by trial and error. The decision process of the agent can be summarized by 3 signals: the *state* of the system before the interaction, the *action* with which the agent interacts with the system, and a *reward* signal that reflects the success of the operation with regard to an objective. Experience can be expressed via the *value function*, reflecting the expected total reward from this state and forward under the current policy. Since behavior (policy) is based on the current value function, and the value function is defined under one policy, an alternating iterative improvement is required for learning. The resulting asymptotic progress is slow, requiring many interactions by RL learning. Although RL has proven effective for solving a range of algorithmic tasks, autonomous control for robotics remains a challenge [10]. Even RL powered by deep function approximation (deep RL) has limited applicability for online interaction learning in Euclidean spaces [9].

A neoRL agent, on the other hand, is composed by a set of sub-agents learning how to achieve different NRES cells for the corresponding NRES representation [12]. The whole set of learning processes constitutes the (latent) learning aspect of the agent; behavior can later be harvested as a weighted sum over the OVFs according to priorities [13]. Learning OVFs as general value functions [21] with $\mathbb{R}$ defined by NRES cell activation, the value function of the whole neoRL agent resembles Kurt Lewin's *fieldt theory* of learning [15]. Leikanger (2021) demonstrated how emulated NRES for agent state allows for autonomous navigation in a single Euclidean space [13], however, multi-modal navigation and combining experience across NRES modalities remains to be tested. As multi-modal NRES capabilities would bring neoRL state representation closer to navigational state representation in the brain, compositionality across NRES modalities would be important for making neoRL a plausible candidate for conceptual navigation.

[1] Note for computing scientists: NRES is not concerned with the Markov state. Any similarity to RL coarse coding and CMAC can therefore be considered to be an endorsement of these AI techniques, not grounds for direct comparison.

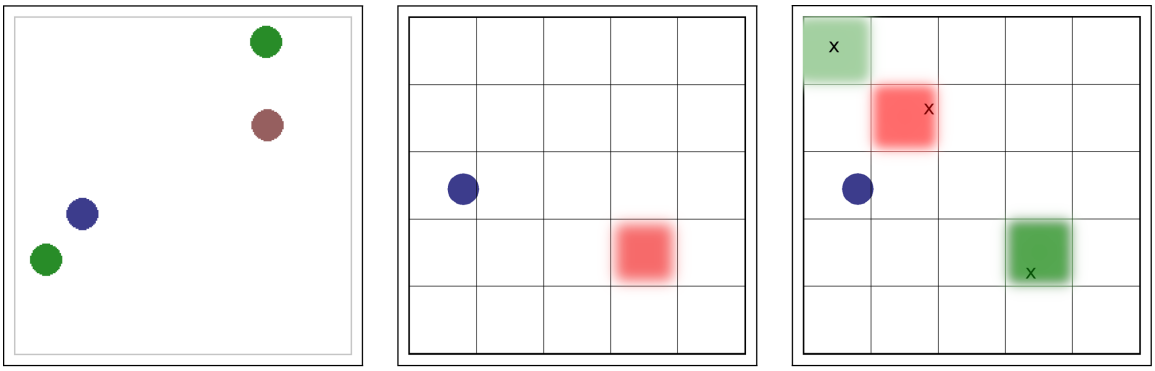


Fig. 3. (A) The allocentric WaterWorld environment: Blue entity is governed by inertia dynamics, with a desire for green ($\mathbb{R} = +1$) and aversion for red ($\mathbb{R} = -1$). (B) An $N5$ mapping of NRES: Each axis is divided into $N = 5$ equal intervals, resulting in $N^2 = 25$ NRES cells. An OVF represents the value function toward one NRES activation. (C) Learned NRES maps can form behaviors via anticipated reward: When an NRES tile contains an element associated with reward, the corresponding OVF is weighted accordingly. Anticipated rewards are illustrated using the same colors as in (A); one aversive NRES cell in red and two desirable NRES cells associated with various anticipation are represented in shades of green. (Color figure online)

3 Multi-modal neoRL Navigation

Adopting Kaelbling's three concerns for Euclidean navigation, we next explore how neoRL navigation scales with increasing (Euclidean) dimensionality. First, it is crucial that NRES-oriented navigation can operate based on different Euclidean spaces; with little knowledge of the form or meaning of conceptual spaces, neoRL must be capable of navigation by other information than location. Further, we are interested in how neoRL navigation scales with additional parameters or across multiple NRES modalities. Any exponential increase in training time with additional states would make conceptual navigation infeasible. NeoRL navigation must be *general* across NRES modalities, *compositional* across conceptual components, without any significant decline in learning *efficiency*. In this section, we explore neoRL capabilities for hi-dimensional navigation by experiments inspired by Kaelbling's concerns for efficient robot navigation.

All experiments are conducted in the allocentric version of the WaterWorld environment [23], illustrated in Fig. 3A. An agent controls the movement of the self (blue dot), with a set of actions that accelerate the object in the four directions N, S, E, W. Three objects of interest move freely in a closed section of a Euclidean plane. When the agent encounters an object, it is replaced by a new object with a random color, location, and speed vector. Green objects are desirable with an accompanying reward $\mathbb{R} = +1.0$, and red objects should be avoided with $\mathbb{R} = -1.0$. No other rewards exist in these experiments, making $\mathbb{R}$ a decent measure of an agent's navigation capabilities. Note that the agent must catch the last green in a board full of red before the board can be reset and continue beyond (on average) 1.5 points per reset.

The PLE [22] version of WaterWorld reports the Cartesian coordinates of the agent and elements of interest; thinking of the Euclidean plane in Fig. 3A as representing location facilitates later discussion. A direct NRES encoding of this information will be referred to as place cell (PC) NRES modality in the remainder of this text. One can also compute a simple object vector cell (OVC) interpretation by vector subtraction:

$$\vec{o^i}_{\mathrm{OVC}} = \vec{o^i}_{\mathrm{PC}} - \vec{s}_{\mathrm{PC}}$$

where $\vec{s}$ is the location of self and $\vec{o^i}$ is the location of object i in PC or OVC reference frame. Note that this OVC interpretation allows for a modality similar to OVC with the self in the center and allocentric direction to external objects, but not with polar coordinates as reported for OVC [8]. However, the two Cartesian representations of location still give different points of view due to different reference frames. Information is encoded in NRES maps as described in Sect. 2.1; the neoRL agent is organized across multiple NRES maps of different resolutions as described in [13]. Multi-res NRES modalities cover resolutions given by primes up to $N13$, i.e., with layers $N2, N3, N5, N7, N11$, and $N13$. For more on multi-resolution neoRL agents and the mechanism behind policy from parallel NRES state spaces, see [13]. All execution runs smoothly on a single CPU core, and the agent starts with no priors other than described in this section. Referring to the NRES modalities as PC and OVC for WaterWorld is only syntactical to facilitate later discussions; 2D Euclidean coordinates are general and can represent any parameter pair.

Learning efficiency is compared by considering the transient proficiency of the agent as measured by the reward received by the agent during $0.2s$ intervals. Any end-of-episode reward is disabled in the WaterWorld settings; the only received reward is $\mathbb{R} = +1$ when encountering green elements and $\mathbb{R} = -1$ when encountering red elements. The simple reward structure makes accumulated $\mathbb{R}$ a direct measure of how well the agent performed during one run. However, observing the transient proficiency – real-time learning efficiency – of the agent requires further analysis: in all experiments, a per-interval average or received reward is computed over 100 independent runs with additional smoothing by a Butterworth low-pass filter. All runs are conducted in isolation; the agent is initiated before each run and deleted after the run – without any accumulation of experience between runs. The x-axis of every plot represents minutes since agent initiation. The y-axis represents proficiency as computed by the per-interval average of received reward, scaled to reflect $[\mathbb{R}/s]$. Proficiency thus measures how many more desirable (green) encounters happen per second than unwanted (red) ones.

3.1 NeoRL Navigation: NRES Generality

First, we examine the generality of the neoRL architecture by comparing navigational proficiency for an agent exposed to a PC modality to one exposed to an OVC modality. We are interested in the generality of neoRL navigation; can neoRL navigate the PC modality by different Euclidean information, and at what cost?

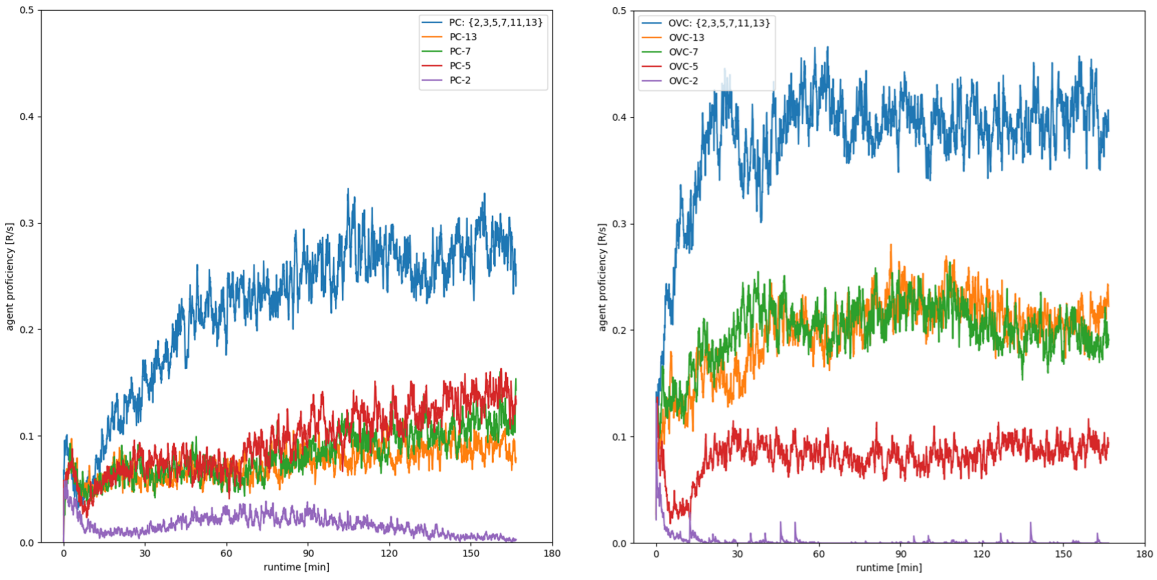


Fig. 4. The neoRL architecture is general across NRES modalities: (A) an original place cell (PC) NRES modality, implemented by applying NRES code directly on an allocentric location of the agent or elements of interest. (B) an emulated object vector cell (OVC) NRES modality, implemented by vector subtraction. OVC is centered on the self with an allocentric representation of other objects.

Results are presented in Fig. 4: agent proficiency from the original PC modality (Fig. 4A) can be compared with agent proficiency when navigating by the OVC modality (Fig. 4B). The immediate proficiency of several mono-resolution neoRL agents is plotted alongside the proficiency of a multi-resolution neoRL agent. There is no loss in sample efficiency when utilizing the OVC modality compared to PC modality. The multi-res neoRL agent performs better than mono-res neoRL agents for both the PC and the OVC modality. NeoRL navigation performs well across both aspects of experience, indicating that the neoRL architecture is general across navigational modalities.

3.2 NeoRL Navigation: NRES Compositionality

Secondly, we are interested in how neoRL scales with additional navigational information. Experiment 1 showed how a neoRL agent is capable of reactive navigation based on an auxiliary NRES modality. In this experiment, we explore the benefit of combining experience across more than one NRES modality. A multi-modal neoRL agent is exposed to both the PC and the OVC modality from experiment 1, effectively doubling the number of NRES states for the agent to consider. We are anxious about how well the neoRL architecture scales with the additional information, both for final proficiency and learning time.

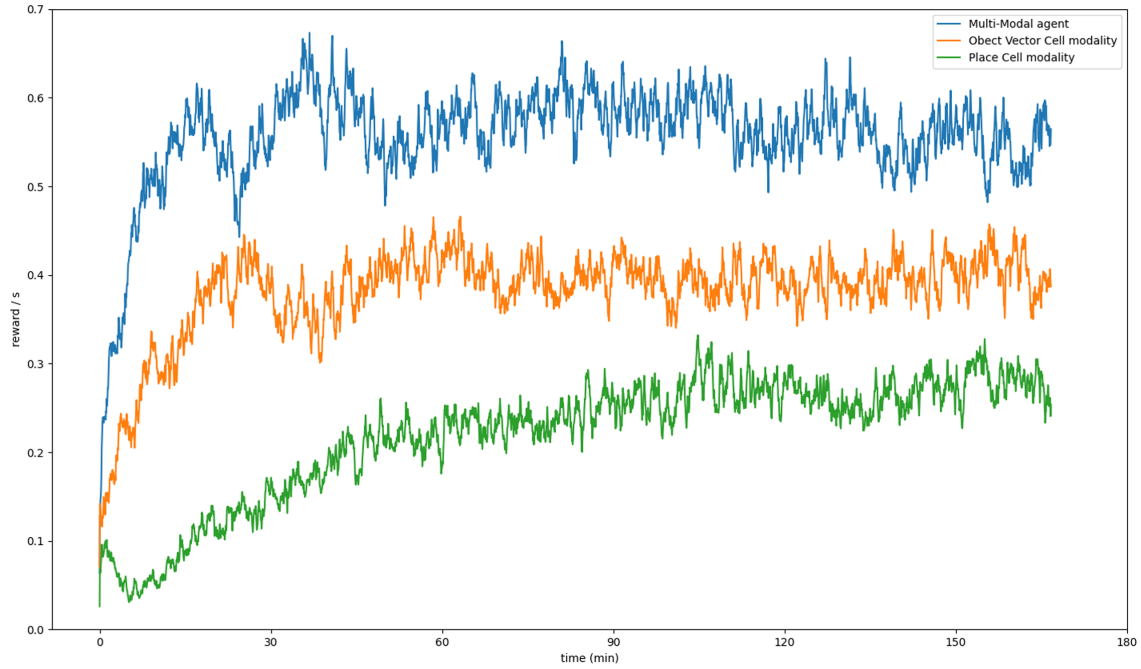


Fig. 5. Multi-modal neoRL navigation leads to higher proficiency and quicker learning than mono-modal agents, despite having twice as many NRES states.

Compare the proficiency of the neoRL agent when exposed to PC, OVC, and multi-modal information in Fig. 5. Combining information across multiple NRES modalities significantly improves navigational performance. The final proficiency of the multi-modal neoRL agent approaches $0.55[\mathbb{R}/s]$ while the PC neoRL agent barely reaches $0.29[\mathbb{R}/s]$. The multi-modal neoRL agent reaches final proficiency after 15 min, whereas the PC neoRL agent uses more than 160 min. In terms of learning efficiency, i.e., how fast the agent reaches final proficiency, and in terms of trained performance, the multi-modal neoRL agent performs better than both mono-modal neoRL agents.

4 Discussion

Contrary to RL in AI, neoRL navigation learns quicker, to higher proficiency, when more information is available to an agent. The neoRL agent is capable of multi-modal navigation, making multi-dimensional Euclidean navigation by a digital agent plausible.

Moving on from reinforcement learning and classical behaviorism, Tolman made a clear distinction between learning and performance after his latent learning experiments (see Fig. 1). Observing how an animal could learn facts about the world that could subsequently be used in a flexible manner, Tolman proposed what he called purposive behaviorism. When motivated by the promise of reward, the animal could utilize latent knowledge to form beneficial behavior toward that objective. Mechanisms underlying orientation have further been implied in cognition, a conceptual space where ideas are represented as points in a multi-dimensional Euclidean space. Technological advances have allowed new evidence

from modern neuroscience, supporting Tolman's hypotheses on cognitive maps' involvement in thought. Inferring that active navigation of such a space corresponds to reasoning and problem solving, we here propose autonomous navigation of conceptual space as an interesting new approach to artificial general intelligence. However, navigating conceptual space – with high dimensionality, an unknown form, and possibly an evolving number of Euclidean dimensions, is no trivial challenge for technology. Based on neural representation of space, the neoRL architecture is distributed and concurrent in learning, capable of separating between latent learning and purposive behavior, and a good candidate for emulated cognition by autonomous navigation of conceptual space.

Adopting Kaelbling's concerns for efficient robot learning to account for multi-modal navigation, we have methodically tested neoRL navigation in the WaterWorld environment. Firstly, it is crucial that neoRL navigation can operate in other Euclidean spaces than its primary navigation modality. Our first experiment verifies that the neoRL architecture is general across Euclidean spaces; a neoRL agent that navigates by the location modality is compared to one exposed to a relative-vector representation of external objects. Both NRES modalities perform admirably at this task, indicating that neoRL navigation is not restricted to one NRES modality. Secondly, we explore how neoRL navigation scales with additional NRES modalities; an agent based on both a place-cell and an object-vector-cell representation is compared to the two mono-modal neoRL agents from experiment 1. Navigation, both in training efficiency and final proficiency, improves significantly when more information is available to the agent. High-dimensional Euclidean navigation appears to be plausible with neoRL technology, formed by the basic principles from neuroscience and NRES.

In this work, we have collected evidence from theoretical neuroscience and the psychology of learning to propose a new direction toward emulated cognition. We have shown how online autonomous navigation is feasible by the neoRL architecture; still, the most interesting steps toward conceptual navigation in machines remain. What are the implications of autonomous navigation of conceptual space for AGI? Could latent spaces from deep networks be used for neoRL navigation? Should desires (elements of interest) propagate across NRES modalities based on associativity? Many important questions are yet to be asked. In showing that neoRL is up for the task of multi-modal navigation, we hereby propose a novel approach to AGI and present a plausible first step toward conceptual navigation in machines.

References

1. Behrens, T.E., et al.: What is a cognitive map? organizing knowledge for flexible behavior. Neuron **100**(2), 490–509 (2018)
2. Bellmund, J.L., Gärdenfors, P., Moser, E.I., Doeller, C.F.: Navigating cognition: spatial codes for human thinking. Science **362**(6415) (2018)
3. Bicanski, A., Burgess, N.: Neuronal vector coding in spatial cognition. Nat. Rev. Neurosci. **21**, 1–18 (2020)

4. Chaplin, J.P.: Systems and Theories of Psychology. Holt, Rinehart and Winston, New York (1961)
5. Constantinescu, A.O., O'Reilly, J.X., Behrens, T.E.: Organizing conceptual knowledge in humans with a gridlike code. Science **352**(6292), 1464–1468 (2016)
6. Eichenbaum, H.: Time cells in the hippocampus: a new dimension for mapping memories. Nat. Rev. Neurosci. **15**(11), 732–744 (2014)
7. Gärdenfors, P.: Conceptual Spaces: The Geometry of Thought. MIT press, Cambridge (2000)
8. Høydal, Ø.A., Skytøen, E.R., Andersson, S.O., Moser, M.B., Moser, E.I.: Object-vector coding in the medial entorhinal cortex. Nature **568**(7752), 400–404 (2019)
9. Kaelbling, L.P.: The foundation of efficient robot learning. Science **369**(6506), 915–916 (2020)
10. Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: a survey. Int. J. Rob. Res. **32**(11), 1238–1274 (2013)
11. Kropff, E., Carmichael, J.E., Moser, M.B., Moser, E.I.: Speed cells in the medial entorhinal cortex. Nature **523**(7561), 419–424 (2015)
12. Leikanger, P.R.: Modular RL for real-time learning. In: The 3rd Conference on Cognitive and Computational Neuroscience (2019)
13. Leikanger, P.R.: Decomposing the prediction problem; autonomous navigation by neoRL agents. In: ALIFE 2021: The 2021 Conference on Artificial Life (2021)
14. Lever, C., Burton, S., Jeewajee, A., O'Keefe, J., Burgess, N.: Boundary vector cells in the subiculum of the hippocampal formation. J. Neurosci. **29**(31), 9771–9777 (2009)
15. Lewin, K.: Field theory and learning (1942)
16. O'Keefe, J., Dostrovsky, J.: The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. Brain Res. **34**, 171–175 (1971)
17. Schafer, M., Schiller, D.: Navigating social space. Neuron **100**(2), 476–489 (2018)
18. Solstad, T.: Neural representations of Euclidean space. PhD thesis, Kavli Insitute of Systems Neuroscience/Center of Neural Computation (2009)
19. Solstad, T., Boccara, C.N., Kropff, E., Moser, M.B., Moser, E.I.: Representation of geometric borders in the entorhinal cortex. Science **322**(5909), 1865–1868 (2008)
20. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, Cambridge (2018)
21. Sutton, R.S., et al.: Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In: The 10th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 761–768. International Foundation for Autonomous Agents and Multiagent Systems (2011)
22. Tasfi, N.: Pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment, Accessed 01 Sept 2020
23. Tasfi, N.: Waterworld in pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment, Accessed 04 Apr 2021
24. Taube, J.S., Muller, R.U., Ranck, J.B.: Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. J. Neurosci. **10**(2), 420–435 (1990)
25. Tolman, E.C.: Cognitive maps in rats and men. Psychol. Rev. **55**(4), 189 (1948)
26. Tolman, E.C., Honzik, C.H.: Degrees of hunger, reward and non-reward, and maze learning in rats. University of California Publications in Psychology (1930)

Appendix C.

Towards neoRL networks;

the emergence of purposive graphs.

Per R. Leikanger.
Submitted for the 5th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM2022).

Towards neoRL networks; the emergence of purposive graphs.

Per R. Leikanger

UiT – Norges Artigste Universitet
Per.Leikanger@uit.no

Abstract

The neoRL framework for purposive AI implements latent learning by emulated cognitive maps, with general value functions (GVF) expressing operant desires toward separate states. The agent's expectancy of reward, expressed as learned projections in the considered space, allows the neoRL agent to extract purposive behavior from the learned map according to the reward hypothesis. We explore this allegory further, considering neoRL modules as nodes in a network with desire as input and state-action Q-value as output; we see that action sets with Euclidean significance imply an interpretation of state-action vectors as Euclidean projections of desire. *Autonomous desire* from neoRL nodes within the agent allows for deeper neoRL behavioral graphs. Experiments confirm the effect of neoRL networks governed by autonomous desire, verifying the four principles for purposive networks. A neoRL agent governed by purposive networks can navigate Euclidean spaces in real-time while learning, exemplifying how modern AI still can profit from inspiration from early psychology.

Keywords: Tolman, purposive AI, GVF, autonomous navigation, neoRL

1 Behavioristic AI by neoRL nodes

Thorndike's *law-an-effect* in functionalist psychology have been reported as an important inspiration for Reinforcement Learning (RL) in AI [6]. Thorndike considered the reinforcement of randomly encountered reflexes as a plausible explanation for simple behavior and acquired reflexes. The law-of-effect represents an essential first step toward the study of behavior becoming a natural science, quickly replaced by *behaviorism* for explaining advanced policies and human behavior. Edward C. Tolman (1866-1959) further proposed that behavior is separate from learning, attempting to explain observations where an animal could express different behavior as a function of varying motivation [9]. Combining Tolman's *latent learning* with *operant conditioning* from E. C. Skinner, considering policies as being *operant* toward an objective, the *neoRL* framework allows for purposive behaviorism for Euclidean navigation [1]. By expressing latent learning as a set of operant reflexes in the environment, i.e., with a set of general value functions (GVF) [7] trained by mutually exclusive conditionals as reward signals, agent purpose becomes an expression of the parameter configurations where operant value functions are extracted. Considering a set of conditionals inspired by the 2014 Nobel Price in neuroscience, the discovery of place cells and other mechanisms behind state representation for neural navigation, Leikanger (2019) demonstrated how operant neoRL subagents could apply for autonomous navigation. Latent learning by a set of operant GVFs combined with elements-of-interest, projections associated with reward in the considered Euclidean space, allows for autonomous navigation governed by the purpose of attaining reward. See my thesis [4] for more on the theory behind NRES-oriented RL (neoRL) agents, scheduled to be presented in a public online[1] PhD defence only days before RLDM.

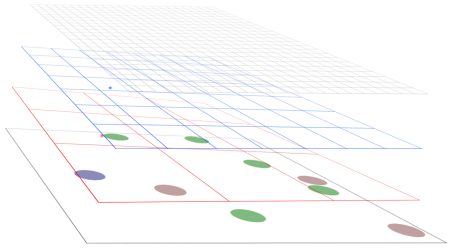


Figure 1: **The first multi-map neoRL agent ;** The neoRL agent is capable of expressing latent learning across several representations of the same Euclidean space, forming agent value function as a weighted sum of operant value from all NRES maps. (figure from [1])

A neoRL learning module can be considered as a behavioral node in a purposive network; early results for neoRL navigation explored the effect of considering multiple state spaces in parallel for the neoRL agent. In some ways analogous to the Hybrid Reward Architecture [10], the neoRL navigation agent combines several learners that establish GVFs toward separate concerns [2]. From applying the superposition principle in the value domain, the neoRL agent is capable of combining value function from many learners in one state space [1], across multiple representations of the same state space [2], or across information represented in orthogonal Euclidean spaces – thus capable of fully decomposing the state space to simpler considerations [3]. The behavioral node consists of three parts; first, the latently learned cognitive map formed by GVF on operant desires toward NRES cells. Operant desires are trained by off-policy GVF, expressing latent learning on how to accomplish different conditionals in this environment representation. Second, the neoRL agent is governed by purpose – mental projections of parameter configurations associated with reward are expressed as elements-of-interest in the NRES map. These free-ranging representations of desire in the considered space are mapped to NRES nodes, activating GVFs corresponding to the associated valence – the element's expectancy of reward upon achievement. Note that the same neoRL node, containing latent knowledge in one NRES representation, can be harvested by different sets of elements-of-interest. Third, the value function can be extracted by elements-of-interest from the digital analogy to Tolman's cognitive map – resulting in an actionable Q-vector output of the neoRL node. When mutually exclusive NRES receptive fields are used for latent learning, the GVF components become operant toward that NRES cell – Operant Value-function Components (OVC). The singular (orthogonal) value component can be combined with others to form the full value function of the agent, further implying that multiple modalities can be learned in parallel and combined to one agent value function [3]. Figure 2 shows the aggregation of the value function for the neoRL agent in [2]. The location and valence of elements-of-interest can be considered as inputs to the neoRL behavioral node, and the superposition of weighted OVC establishes an output of the neoRL node.

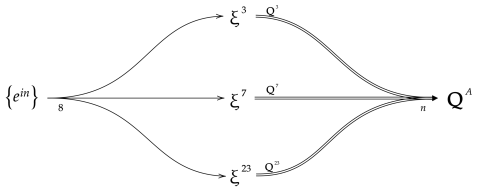


Figure 2: **A schematic representation of fig. 1,** the neoRL agent from [2]. Agent value function is formed from the combined value function Q^N from each of the three ξ^N neoRL nodes, where $N \in \{3, 7, 23\}$ [2].

[1] Information about the streamed PhD defence will be posted on *www.neoRL.net*

The purposive AI expressed by the neoRL agent in figure 2 can well be considered as a single-layered behavioral network with a single output state-action value function. Figure 2 presents a functional representation of the multi-map navigational agent from figure 1. The individual neoRL sub-nodes ξ^3, ξ^7, ξ^{23} are trained by latent learning, and purposive state-action value functions $\{Q^3, Q^7, Q^{23}\}$ can be extracted for the digital analogy to a cognitive map by purposive elements $\{e^{in}\}$. The network would have an input, the full set of elements of interest $\{e^{in}\}$; the purposive network would have a latent state, formed by latent learning expressed as off-policy operant desires; the neoRL network would have an actionable state-action Q-vector as output. When further assuming a Euclidean significance for the action set, as with $\mathbb{A} = \{N, S, E, W\}$ in [2], the state-action value could be interpreted a Euclidean *desire-vector*;

$$\vec{d} = \vec{\sum} \mathbf{Q}^{in} \quad ,$$

where $\vec{\sum}$ represents the vector sum. The valence of the desire vector $\vec{d}$ should express the combined valence across $\{e^{in}\}$.

$$\vec{e^{out}} = \vec{d}$$
$$e^{out}_{\psi} = \sum e^{in}_{\psi} \ ,$$

where $\vec{e^i}$ signifies the coordinate and e^i_ψ represents the valence of purposive element i. A functional schematic of the neoRL module is illustrated in figure 3; a single output-desire can be formed from any number of input elements $\{e^{in}\}$. Different sets of purposive elements-of-interest $\{e^{in}\}$ can establish different output desire e^{out} and actionable state-action values Q^N from the same cognitive map. Earlier networks of the neoRL node could be seen as a behavioral analog to a one-layered perceptron [5]. The enclosed theory allows for multi-layered neoRL networks governed by autonomous desire.

2 Experiments

A comprehensive environment for research on autonomous navigation is the PLE implementation [8] of Karpathy's WaterWorld environment; An agent controls acceleration of the self in the four Cardinal directions $[N, S, E, W]$, i.e., [up, down, right, left]. Three objects move around in the Euclidean plane according to predefined mechanics. Encountering an object replaces it with a new object with a random color, location, and speed vector. Green objects are desirable with a positive reward $\mathbb{R} = +1.0$, and red objects are repulsive with a negative valence of -1.0. Capturing the last green object resets all remaining (red) objects by the same reset mechanisms. The only reward comes from encountering objects, making the accumulation of $\mathbb{R}$ an objective measure for navigational capabilities and its time course an indication of real-time learning capabilities.

Rewards are only received sparsely, with discrete $+1.0$ or -1.0 steps after encountering objects in the WaterWorld environment; measuring the immediate proficiency of the agent by accumulated $\mathbb{R}$ can be a challenge. Capturing the transient time course of agent skill can be done by averaging independent runts; the enclosed experiments average 100 separate runs to measure transient navigational proficiency, i.e., across 100 separate agents. No pre-training or other precursors are available for the agents, making all navigation happen live as the agent gathers experience in the environment for the first time. Curves are presented with minutes along the x-axis, signifying the wall-clock time since the beginning of each run.

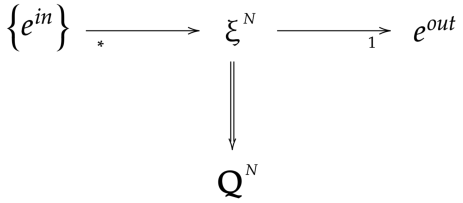
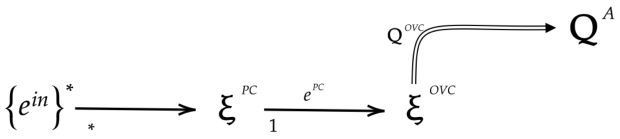
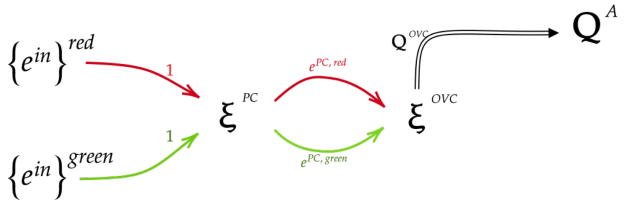


Figure 3: **A neoRL learning module with one input and two output;** actions with a Euclidean significance implies state-action vectors to be representable in the same Euclidean space; e^{out} can be used as input for compatible neoRL nodes.

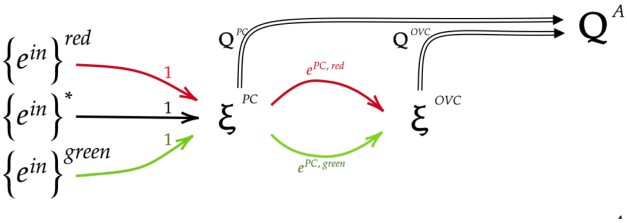
We shall explore four aspects in the WaterWorld environment; first, we challenge the basic principle of propagating purpose by the layout illustrated in figure 4**a**. The first neoRL node, ξ^{PC}, forms a purposive desire based on all reported objects from WaterWorld; a single desire vector $\vec{d}$ with accompanying valence e_ψ propagates to the compatible neoRL node ξ^{OVC} as autonomous desire e^{PC}. *Experiment **[b]*** explores the effect of extracting separate desires from the same learned cognitive map. A purposive desire vector from the neoRL node comes from extracting latent knowledge from considered coordinates; the agent extracts two purposive vectors from ξ^{PC}, one from desirable objects $e^{in_{green}}$ and a separate from aversive objects $e^{in_{red}}$. The output from multiple neoRL nodes can be combined for the policy-forming value function in the neoRL framework; *experiment **[c]*** explores the effect of aggregating value function from multiple depths of the neoRL net. The output desire from a neoRL node ξ^N can be applied to any compatible neoRL node ξ^M, including itself: *experiment **[d]*** explores recurrent connections for neoRL nodes. Collaborative experience is explored with and without recursive desires, as illustrated in figure 4c and 4d. All results are reported in figure 4e.



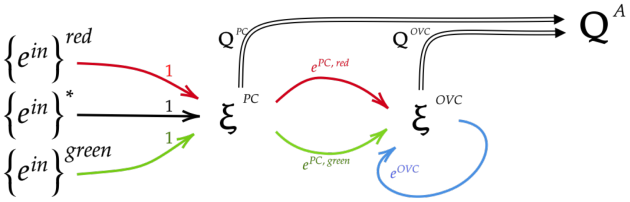
(a) Aspect 1: a single desire vector e^{PC} as input to ξ^{OVC}.



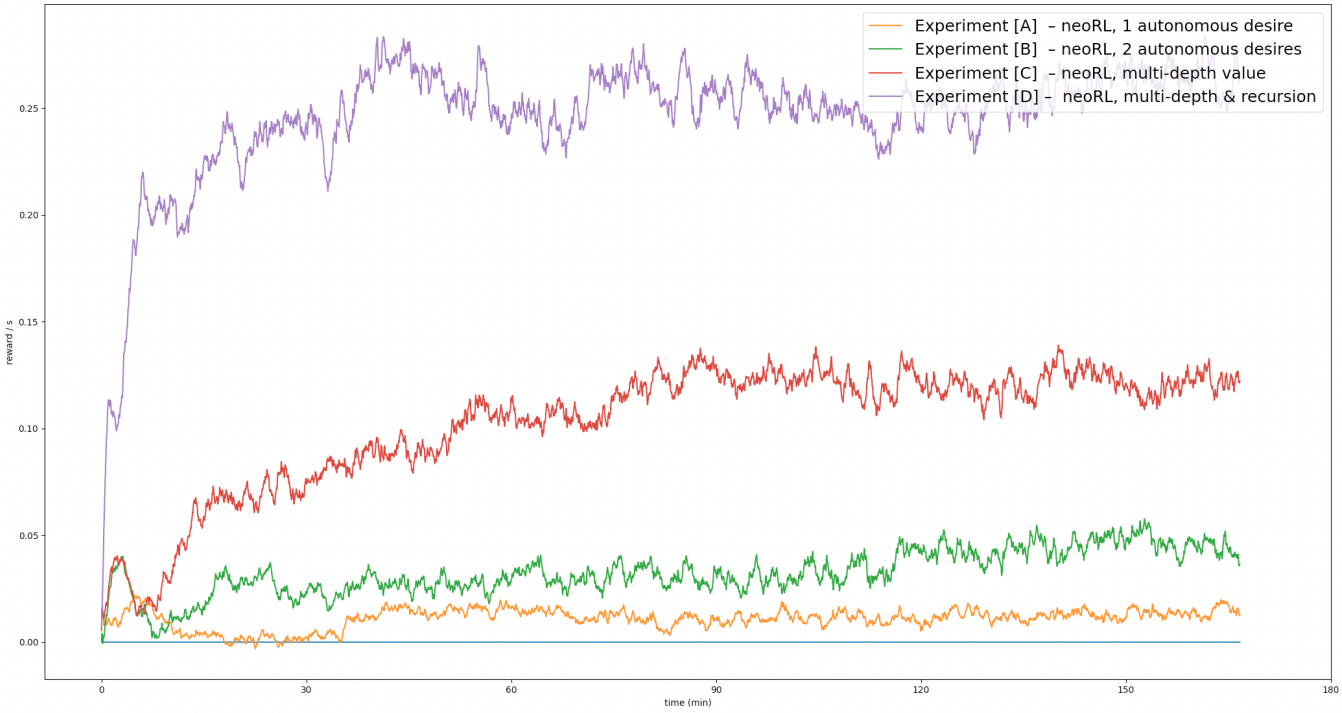
(b) Aspect 2: separate desire extraction; $e^{PC_{red}}$ and $e^{PC_{green}}$.



(c) Aspect 3: joint value function from sequential neoRL nodes.



(d) Aspect 4: recursive desires for neoRL autonomy.



(e) Transient proficiency of neoRL agent A-D.

Figure 4: [**Top**] Illustrations of the neoRL architecture tested in experiment A-D. [**a**] A first attempt on desire from experience; neoRL node ξ^{PC} forms a single desire e^{PC} for value-generating neoRL node ξ^{OVC}. [**b**] Latent knowledge can be extracted separately for separate classes for desire; experiment B forms two desire-vectors $e^{PC_{red}}$ and $e^{PC_{green}}$ from ξ^{OVC} – grouping according to valence. [**c**] The value function output from neoRL node ξ^{PC} and node ξ^{OVC} contribute equally to agent value function. [**d**] Recursive desires are possible for neoRL nodes: the ξ^{OVC} is governed by three elements-of-interest, $e^{PC_{red}}$, $e^{PC_{green}}$, and recurrent desire e^{OVC}. [**Down**] Results from the four experiments: [**e**] Purposive neoRL networks allows for purposive autonomy by deep and/or recurrent desires.

3 Discussion

The neoRL agent navigates continuous space by projections of desire, vectors of purpose associated with an agent's expectancy or reward. When actions have a Euclidean significance, purposive Q-vectors can form autonomous projections of desire, experience-based inferences that can establish input to deeper neoRL nodes. Experiments demonstrate how deeper or recurrent desires are crucial for navigational proficiency, suggesting purposive neoRL networks as a plausible approach to autonomous navigation.

This work explores four principles of purposive neoRL networks. *First*, experience-based desire vectors can shape purposive navigation in Euclidean space. The neoRL network from illustration 4a improves navigational proficiency over time; however, considering all objectives under one, i.e., forming a single desire vector based on all red *and* green objects, becomes too simple for proficient navigation. *Second*, a single neoRL node can generate different e^{out} desires by considering different sets of objectives $\{e^{in}\}$. Experiment **b** demonstrates the effect of separating desires according to valence, resulting in the increased performance by the neoRL navigational agent. The simplicity and clarity expressed by separation of desires, as illustrated in figure 4b, facilitates explainability of the trained solution – a crucial element if traditional AI is to be applied for desire classification. *Third*, the neoRL agent can base agent value function on any neoRL node in the network. Agents extracting purpose from multiple depths of the neoRL network, as illustrated in 4c, becomes better navigators than more superficial agents. *Fourth*, desire vectors e^{out} from one neoRL node can form objectives for any compatible neoRL node – including itself. Experiment [d] explores recursive desires, where the output of ξ^{OVC} – desire vector e^{OVC} – establish an additional purpose for neoRL node ξ^{OVC}. The proficiency of the recursive agent from 4d is reported as curve [D], showing how recursive desires drastically improve the agent's navigational proficiency. Results can be examined in figure 4e and in real-time video demonstrations at *www.neoRL.net* .

Note that no comparison has been made with alternative approaches for control; this work is only concerned with uncovering the basic principles of purposive neoRL nets for behavioral AI. Still, any attempt on finding RL or AI solutions capable of allocentric Euclidean navigation in real-time has failed. Further work could involve finding and comparing alternative approaches for real-time autonomous navigation in the WaterWorld environment. Likewise, this work involves no search for optimal parameters for neoRL navigation. Experiment [c] explores collaborative experience with 1:1 weight ratio between ξ^{PC} and ξ^{OVC}, and experiment [d] only explores a unitary feedback loop $r = -1.0$. It is left for further work to explore network architecture theory or find data-driven methods for parameter adaptation. We have barely scratched the surface of autonomous navigation by purposive networks, proposing neoRL networks as a plausible new approach toward navigational autonomy.

References

[1] Per R. Leikanger. Modular RL for real-time learning. In *The 3rd Conference on Cognitive and Computational Neuroscience*. Cognitive and Computational Neuroscience, 2019.

[2] Per R. Leikanger. Decomposing the Prediction Problem; Autonomous Navigation by neoRL Agents. volume ALIFE 2021: The 2021 Conference on Artificial Life of *ALIFE 2021: The 2021 Conference on Artificial Life*, 07 2021. 30.

[3] Per R. Leikanger. Navigating conceptual space; a new take on AGI. In *International Conference on Artificial General Intelligence*, pages 116–126. Springer, 2021.

[4] Per R. Leikanger. *Autonomous navigation in the (animal and the) machine*. PhD thesis preprint @ arχiv, UiT, 2022.

[5] Frank Rosenblatt. The perceptron, a perceiving and recognizing automaton project para. Technical report, 1957.

[6] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[7] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[8] Norman Tasfi. Waterworld in pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment.

[9] Edward Chace Tolman and Charles H Honzik. Degrees of hunger, reward and non-reward, and maze learning in rats. *University of California Publications in Psychology*, 1930.

[10] Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5392–5402, 2017.

Bibliography

[1] Caltech archives. Photo of richard feynman's blackboard, february 1988, at the time of his death. https://archives.caltech.edu/pictures/1.10-29.jpg. Accessed: 2022-01-01.

[2] Dmitriy Aronov, Rhino Nevers, and David W Tank. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722, 2017.

[3] Christopher G Atkeson et al. Using local trajectory optimizers to speed up global optimization in dynamic programming. *Advances in neural information processing systems*, pages 663–663, 1994.

[4] Jonathan Baxter, Andrew Tridgell, and Lex Weaver. Knightcap: a chess program that learns by combining td (lambda) with game-tree search. *arXiv preprint cs/9901002*, 1999.

[5] Mark F Bear, Barry W Connors, and Michael A Paradiso. *Neuroscience: Exploring the brain*. Lippincott Williams & Wilkins Publishers, 2007.

[6] Timothy EJ Behrens, Timothy H Muller, James CR 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.

[7] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[8] RE Bellman. A markov decision process. *Journal of Mathematical Mechanics*, 1957. (according to [66]).

[9] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[10] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

[11] Jacob LS Bellmund, Peter Gärdenfors, Edvard I Moser, and Christian F Doeller. Navigating cognition: Spatial codes for human thinking. *Science*, 362(6415), 2018.

[12] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[13] Andrej Bicanski and Neil Burgess. Neuronal vector coding in spatial cognition. *Nature Reviews Neuroscience*, pages 1–18, 2020.

[14] James Patrick Chaplin. *Systems and theories of psychology*. Holt, Rinehart and Winston, 1961.

[15] Jack Clark and Dario Amodei. Faulty reward functions in the wild. `https://openai.com/blog/faulty-reward-functions/`, 2016.

[16] Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

[17] Charles Darwin. *On the origin of species by means of natural selection, : or the preservation of favoured races in the struggle for life*. J. Murray, (1. ed.). edition, 1859.

[18] Sachin S Deshmukh and James J Knierim. Influence of local objects on hippocampal representations: Landmark vectors and memory. *Hippocampus*, 23(4):253–267, 2013.

[19] Clayton W Dodge. *Euclidean geometry and transformations*. Courier Corporation, 2004.

[20] Russell A Epstein, Eva Zita Patai, Joshua B Julian, and Hugo J Spiers. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504–1513, 2017.

[21] Howard Whitley Eves. *Foundations and fundamental concepts of mathematics*. Courier Corporation, 1997.

[22] Richard Feinman. Q&A september 26th, 1985. "can computers think". https://www.youtube.com/watch?v=ipRvjS7q1DI . Accessed: 2022-03-03.

[23] P. Foldiak and D. Endres. Sparse coding. *Scholarpedia*, 3(1):2984, 2008. revision #145589.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[25] Charles AE Goodhart. Problems of monetary management: the uk experience. In *Monetary theory and practice*, pages 91–121. Springer, 1984.

[26] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation. *arXiv preprint arXiv:1610.00633*, 1, 2016.

[27] David Hilbert. Mathematical problems. *Bulletin of the American Mathematical Society*, 8(10):437–479, 1902.

[28] Ronald A Howard. Dynamic programming and markov processes. 1960.

[29] Øyvind Arne Høydal, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May-Britt Moser, and Edvard I Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404, 2019.

[30] Alex Irpan. Deep reinforcement learning doesn't work yet. `https://www.alexirpan.com/2018/02/14/rl-hard.html`, 2018.

[31] William James. *The Principles of Psychology*. Henry Holt and Company, 1890.

[32] Leslie Pack Kaelbling. The foundation of efficient robot learning. *Science*, 369(6506):915–916, 2020.

[33] ER Kandel, JH Schwartz, and TM Jessell. Principles of neuroscience, 4th, 2000.

[34] Eric R Kandel and Leon Tauc. Heterosynaptic facilitation in neurones of the abdominal ganglion of aplysia depilans. *The Journal of Physiology*, 181(1):1, 1965.

[35] Andrej Karpathy. Waterworld. https://cs.stanford.edu/people/karpathy/reinforcejs/waterworld.html. Accessed: 2022-01-01.

[36] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[37] Andrey Nikolaevich Kolmogorov. The representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. *Doklady Akademii Nauk SSSR*, 108(2):179–182, 1956.

[38] 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.

[39] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[40] Per R. Leikanger. Modular RL for real-time learning. In *The 3rd Conference on Cognitive and Computational Neuroscience*. Cognitive and Computational Neuroscience, 2019.

[41] Per R. Leikanger. Decomposing the Prediction Problem; Autonomous Navigation by neoRL Agents. volume ALIFE 2021: The 2021 Conference on Artificial Life of *ALIFE 2021: The 2021 Conference on Artificial Life*, 07 2021. 30.

[42] Per R. Leikanger. Navigating conceptual space; a new take on AGI. In *International Conference on Artificial General Intelligence*, pages 116–126. Springer, 2021.

[43] Per R. Leikanger. Towards neoRL networks; the emergence of purposive graphs, *Submitted* to RLDM2022. , @arXiv : 2202.12622.

[44] Colin Lever, Stephen Burton, Ali Jeewajee, John O'Keefe, and Neil Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31):9771–9777, 2009.

[45] Tingguang Li, Weitao Xi, Meng Fang, Jia Xu, and Max Q-H Meng. Learning to solve a rubik's cube with a dexterous hand. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1387–1393. IEEE, 2019.

[46] Yitao Liang, Marlos C Machado, Erik Talvitie, and Michael Bowling. State of the art control of atari games using shallow reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 485–493. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[47] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[48] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

[49] Richard E. Meyer. *The Oxford handbook of cognitive psychology*, chapter Problem Solving. Oxford University Press, 2013.

[50] John Stuart Mill. On liberty, ed. david spitz. *New York: Norton*, 54:82, 1975.

[51] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[52] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[53] Hai Nguyen and Hung La. Review of deep reinforcement learning for robot manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 590–595. IEEE, 2019.

[54] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971.

[55] Santiago Ontanón, Gabriel Synnaeve, Alberto Uriarte, Florian Richoux, David Churchill, and Mike Preuss. A survey of real-time strategy game ai research and competition in starcraft. *IEEE Transactions on Computational Intelligence and AI in games*, 5(4):293–311, 2013.

[56] The Nobel Price. The nobel prize in physiology or medicine 2014. https://www.nobelprize.org/prizes/medicine/2014/summary/, 2014.

[57] Santiago Ramón y Cajal. *Histologie du système nerveux de l'homme et des vertébrés*, volume 2. A. Maloine, 1911.

[58] Edmund T Rolls, Alessandro Treves, and Edmund T Rolls. *Neural networks and brain function*, volume 572. Oxford university press Oxford, 1998.

[59] Frank Rosenblatt. The perceptron, a perceiving and recognizing automaton project para. Technical report, 1957.

[60] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[61] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[62] Duane P Schultz and Sydney Ellen Schultz. *Modern psychology: A history*. Wadsworth Cengage Learning, 2012.

[63] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[64] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[65] Trygve Solstad, Charlotte N Boccara, Emilio Kropff, May-Britt Moser, and Edvard I Moser. Representation of geometric borders in the entorhinal cortex. *Science*, 322(5909):1865–1868, 2008.

[66] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[67] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[68] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[69] Frode Svartdal. *Psykologi: en introduksjon*. Gyldendal akademisk, 2011.

[70] Norman Tasfi. Waterworld in pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment.

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

[72] Karl Halvor Teigen. *En psykologihistorie*. Fagbokforl., 2019.

[73] Gerald Tesauro. Td-gammon: A self-teaching backgammon program. In *Applications of Neural Networks*, pages 267–285. Springer, 1995.

[74] Gerald Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[75] Edward L Thorndike. Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, 2(4):i, 1898.

[76] Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

[77] Edward C Tolman, Benbow F Ritchie, and D Kalish. Studies in spatial learning. ii. place learning versus response learning. *Journal of experimental psychology*, 36(3):221, 1946.

[78] Edward Chace Tolman and Charles H Honzik. Degrees of hunger, reward and non-reward, and maze learning in rats. *University of California Publications in Psychology*, 1930.

[79] John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 42(5):674–690, 1997.

[80] Harm van Seijen, Mehdi Fatemi, Joshua Romoff, and Romain Laroche. Separation of concerns in reinforcement learning. *arXiv preprint arXiv:1612.05159*, 2016.

[81] Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5392–5402, 2017.

[82] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[83] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[84] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

[85] John B Watson. Psychology as the behaviorist views it. *Psychological review*, 20(2):158, 1913.

[86] John Broadus Watson. Behaviorism, rev. 1930.

[87] John Broadus Watson and William McDougall. *The battle of behaviorism: An exposition and an exposure*. WW Norton & Company, 1929.

[88] Marco A Wiering and Hado Van Hasselt. Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):930–936, 2008.

[89] Thomas Wolbers and Jan Wiener. Challenges for identifying the neural mechanisms that support spatial navigation: The impact of spatial scale. *Frontiers in human neuroscience*, 8:571, 08 2014.

[90] Herman OA Wold. On prediction in stationary time series. *The Annals of Mathematical Statistics*, 19(4):558–567, 1948.

[91] Stephen P Wood, Jesse Chang, Thomas Healy, and John Wood. The potential regulatory challenges of increasingly autonomous motor vehicles. *Santa Clara L. Rev.*, 52:1423, 2012.

[92] Daniel Zhang, Saurabh Mishra, Erik Brynjolfsson, John Etchemendy, Deep Ganguli, Barbara Grosz, Terah Lyons, James Manyika, Juan Carlos Niebles, Michael Sellitto, et al. The ai index 2021 annual report. *arXiv preprint arXiv:2103.06312*, 2021.