Gatsby Computational Neuroscience Unit
Faculty of Life Sciences

Principles of Optimal Learning Control in Biological and Artificial Agents

Rodrigo Carrasco-Davis

Gatsby Computational Neuroscience Unit

Faculty of Life Sciences

University College London

Supervisor: Andrew Saxe

A dissertation submitted in partial fulfillment for the degree of

Doctor of Philosophy

March 2025

Declaration

I, Rodrigo Carrasco-Davis, confirm that the work presented in my thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **For a research manuscript that has already been published** (if not yet published, please skip to section 2):

   (a) **What is the title of the manuscript?**

   (b) **Please include a link to or doi for the work:**

   (c) **Where was the work published?**

   (d) **Who published the work?**

   (e) **When was the work published?**

   (f) **List the manuscript's authors in the order they appear on the publication:**

   (g) **Was the work peer reviewed?**

   (h) **Have you retained the copyright?**

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If 'Yes', please give a link or doi**

   if 'No' please seek permission from the relevant publisher and check the box next to the below statement:

   ☐ *I acknowledge the permission of the publisher named under section 1d to include in this thesis portions of the publication listed in section 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3):

   (a) **What is the current title of the manuscript?**

   Meta-Learning Strategies through Value Maximization in Neural Networks

   (b) **Has the manuscript been uploaded to a preprint server e.g. medRxiv? If 'Yes', please please give a link or doi:**

   Yes. `https://arxiv.org/abs/2310.19919`

(c) **Where is the work intended to be published?**
Potentially to Proceedings of the National Academy of Sciences of the United States of America (PNAS)

(d) **List the manuscript's authors in the intended authorship order:**
Rodrigo Carrasco-Davis, Javier Masís, Andrew Saxe.

(e) **Stage of publication:** Preparing for submission

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4):
Rodrigo Carrasco-Davis: Formulating research ideas design of experiments, drafting manuscript, mathematical analysis, and simulations. Javier Masís: Formulating research idea, providing experimental data, draft revision. Mentoring. Andrew Saxe: Formulating research ideas and research goals. Draft revision. Mentoring.

4. **In which chapter(s) of your thesis can this material be found?**
Chapter 3.

e-Signatures confirming that the information above is accurate (this form should be co-signed by the supervisor/senior author unless this is not appropriate, e.g. if the paper was a single-author work):

Candidate: Rodrigo Carrasco-Davis
Date: March 2025

Supervisor/Senior Author signature (where appropriate): Andrew Saxe
Date: March 2025

UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **For a research manuscript that has already been published** (if not yet published, please skip to section 2):

   (a) **What is the title of the manuscript?**

   (b) **Please include a link to or doi for the work:**

   (c) **Where was the work published?**

   (d) **Who published the work?**

   (e) **When was the work published?**

   (f) **List the manuscript's authors in the order they appear on the publication:**

   (g) **Was the work peer reviewed?**

   (h) **Have you retained the copyright?**

   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If 'Yes', please give a link or doi**

   if 'No' please seek permission from the relevant publisher and check the box next to the below statement:

   ☐ *I acknowledge the permission of the publisher named under section 1d to include in this thesis portions of the publication listed in section 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3):

   (a) **What is the current title of the manuscript?**

   Uncertainty Prioritized Experience Replay

   (b) **Has the manuscript been uploaded to a preprint server e.g. medRxiv? If 'Yes', please please give a link or doi:**

   Yes. `https://arxiv.org/abs/2506.09270`

(c) **Where is the work intended to be published?**
Reinforcement Learning Conference (RLC) 2025

(d) **List the manuscript's authors in the intended authorship order:**
Rodrigo Antonio Carrasco-Davis, Sebastian Lee, Claudia Clopath, Will Dabney.

(e) **Stage of publication:** Submitted.

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4):
Rodrigo Carrasco-Davis: Drafting manuscript, simulations (arm bandit task), mathematical analysis, experiment design. Sebastian Lee: Drafted manuscript, formulating research, simulations (grid world), mathematical analysis, and experiment design. Claudia Clopath: Formulating research, experiment design, draft revision, mentoring. Will Dabney: Mathematical analysis, formulating research, simulations (Atari), draft revision, mentoring.

4. **In which chapter(s) of your thesis can this material be found?**
Chapter 5.

e-Signatures confirming that the information above is accurate (this form should be co-signed by the supervisor/senior author unless this is not appropriate, e.g. if the paper was a single-author work):

Candidate: Rodrigo Carrasco-Davis
Date: March 2025

Supervisor/Senior Author signature (where appropriate): Claudia Clopath
Date: March 2025

Abstract

Humans and other animals must continuously decide how to allocate their cognitive resources. A key aspect of this involves controlling effort throughout the learning of new tasks, known as cognitive control of learning. In machine learning, this is referred to as meta-learning and focuses on scheduling hyperparameters to improve the learning processes. Determining the optimal allocation of control requires complex computations, such as estimating future utility while accounting for uncertainties, learning capacity, task difficulty, and other resource demands. Although previous research has proposed computational mechanisms for control allocation in learning, these methods often lack interpretability, resist mathematical analysis, and rely on models that are difficult to scale. This thesis adopts a parsimonious approach to identify principles of learning control. It introduces a normative framework based on cumulative reward maximization for optimally allocating control throughout the learning process. This framework unifies and instantiates existing theories in machine learning, such as Model-Agnostic Meta-Learning, and in cognitive neuroscience, specifically the Expected Value of Control theory applied to learning systems. The thesis further explores the application of this framework to neural networks, revealing key features of optimal learning control, including inter-temporal control allocation and curriculum learning. Additionally, it provides mathematical analyses for optimal learning control, approximating time-dependent control allocation using methods from control theory. Another aspect of meta-learning is identifying learnable task components. To achieve this, the thesis introduces a novel method for estimating epistemic uncertainty to prioritize replay on the most useful experiences. The primary contribution of this thesis is a formal normative framework for understanding learning control in machine learning and cognitive neuroscience. It provides methods for optimal control and epistemic uncertainty computation, suggesting directions for further mathematical analysis. This work could establish a robust framework for understanding animal learning across lifespans and enable more efficient learning in artificial systems.

Impact Statement

The work presented in this thesis is primarily abstract and theoretical, with some practical applications in modern machine learning models and insights into observed phenomena in cognitive science. In the near future, it may contribute to the development of mathematical methods for analyzing learning agents and controlling learning processes. More broadly, the formal framework presented unifies various methods from control theory, meta-learning, and cognitive science, offering insights into optimal solutions to these problems through a common perspective. The proposed epistemic uncertainty estimator for replay could further enhance artificial learning agents and provide a framework for studying replay in biological agents. The study of learning control is inherently linked to learning itself, implying that a complete understanding of learning requires elucidating its control counterpart. This thesis presents an approach to exploring that avenue.

Acknowledgements

I'll be forever grateful for having the chance to do science for a living and to marvel at the universe every day. For that to be possible in today's world, scientists must be incredibly fortunate. In my case, I've been undeniably lucky, an outlier, thanks to the unwavering support of those around me. Because of them, I've been able to do work I'm proud of while genuinely enjoying the process. What a win...

First, I want to thank my closest collaborators, who made this thesis possible. Javier played an absolutely pivotal role, his work inspired much of the science presented here, and he always offered kind words, encouragement, and fun conversations throughout our work together. He also helped pave the way for my next steps as a researcher. Sebastian, one of the coolest people I know, not only offered an interesting project to work on together, but also shared his warm and effortless friendship. Valentina has been a delight to work with, one of the brightest minds I've met. Sarah A. gave me the chance to be part of a sci-fi-like project, filled with good laughs, likely fueled by dementia. Clementine, always radiating energy and kindness, pushed to get things done and is a close friend I know I can always count on. Will Dorrell, a great friend and a constant reminder to see life through the lens of awe and curiosity. Tom, who nourished me with deep conversations and a pristine stout. Erin, one of the most brilliant researchers I've ever encountered in nearly every sense, including her kind words and support in tough times. Stefano, an inspiration and my personal physicist.

The list is long, but one of my biggest motivations to bike to the office each day was to spend time with the incredible people at Gatsby, SWC, and the Saxe Lab. I deeply appreciate the kindness and camaraderie of Peter, Basile, Sam, Jin, Verena, Luke, Sarah E,

Tyler, Devon, Nish, Pierre, and Kevin. And without the help of I-Chun, Mike, Despina and Barry, I'd probably be lost somewhere in London.

One of the luckiest events during my time at Gatsby was the arrival of my advisor, Andrew. He has an incredible ability to transmit his excitement about ideas, sometimes speaking directly to my brain when my words fail to convey the message. He always manages to highlight the best side of research, like finding gold nuggets in the soil. He runs the lab with genuine care for its members, both in their careers and personal lives. Being part of his lab is a privilege, and if I ever find success in academia, a great deal of it will be thanks to his guidance. I also want to thank Claudia Clopath, Will Dabney, Marcus Stephenson-Jones, and Jonathan Cohen for their guidance through my research quests, and special thanks to Peter Latham for convincing me to apply to Gatsby.

Thanks to my former advisor, Prof. Pablo Estévez, who shaped me as a scientist and mentored me before I started my PhD, always keeping an eye on my journey. Pedro Ortega, a friend who helped me navigate academia and brought a bit of Chile to a foreign land. Speaking of Chile, I want to thank my friends back home, who have always been there, maintaining our bond even from afar, Pancho, Jorge, Omar, César, Charlie, Nico, José, Yves, Pamela, Felipe, Andrea, Guillermo, and Esteban.

I wouldn't be a scientist at all if it weren't for my parents and my brother. Their unconditional love, open-mindedness, and genuine curiosity about my work fuel me to push forward, to find joy in life, and, most importantly, to be kind to others. They are at the core of who I am, and I'll be forever grateful to have them in my life. I also want to thank my wife's family, who have always made me feel like I belong with them.

Finally, I want to thank my wife, Coni. I could never fully reciprocate the joy she brings into my life, but it is a blessing to try. She is both a part of me and a wonder of the cosmos. When she is around, home no longer seems like a pale blue dot, as it becomes a shiny and colorful marble, a world I could contemplate forever.

The cosmos is within us. We are made of star-stuff. We are a way for the universe to know itself.

— Carl Sagan

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DDM: Drift Difussion Model

DNN: Deep Neural Network

EVC: Expected Value of Control

GD: Gradient Descent

HJB: Hamilton-Jacobi-Bellman equation

HPM: Homotopy Perturbation Method

LDDM: Linear Drift Difussion Model

LEVC: Learning Expected Value of Control

MAML: Model Agnostic Meta-Learning

MDP: Markov Decision Process

MSE: Mean Square Error

PER: Prioritized Experience Replay

RL: Reinforcement Learning

RNN: Recurrent Neural Network

RT: Reaction Time

SNR: Signal to Noise Ratio

TD: Temporal Difference

UPER: Uncertainty Prioritized Experience Replay

Chapter 1

Introduction

1.1 What is Intelligence?

Before addressing the central topics of this thesis, meta-learning and cognitive control, it is necessary to establish several foundational concepts. This discussion progresses from general to specific, beginning with an examination of intelligence. It then explores how intelligence is acquired over a lifespan and across evolutionary time scales (learning) and, finally, how the control of learning processes and executive functions can enhance the adaptability of intelligent agents. The discussion begins with a brief exploration of intelligence as a natural phenomenon observed in living beings and machines.

What defines an intelligent agent? A common perspective is that an intelligent entity is one that can solve problems, comprehend various concepts, learn, and perhaps behave as if driven by a purpose. In nature, intelligent agents, such as humans and other animals, exhibit these characteristics. However, the precise meaning of an *intelligent* being remains complex. Intelligence is a concept that is difficult to define, yet it is widely understood in general contexts. Despite this intuitive grasp, the search for a formal definition has a long history.

The qualities that an intelligent agent must satisfy have been widely debated. A broad definition, such as the one provided by Wechsler (1958), describes it as *the aggregate or global capacity of the individual to act purposefully, to think rationally, and to deal effectively with its environment.* Later definitions incorporate additional aspects, such as multiple intelligences (Gardner, 1983), meta-cognitive capabilities (e.g., knowing what it knows, Carroll 1997), and the capacity for learning and adaptability (Sternberg and Kagan, 1986). For a historical review of the definitions and measurement of human intelligence, see Hindes et al. (2011). Most of these definitions are anthropocentric. However, discussions of intelligence in animals were also taking place, often with a focus on evolution and its influence on branching skills, which resulted in diverse intelligence capabilities among species (Vonk, 2021). While human intelligence exhibits unique features compared to the rest of the animal kingdom (Cantlon and Piantadosi, 2024), the need for a non-anthropocentric definition of intelligence that encompasses humans, other animals, and artificial agents has been recognized (Holm and Banerjee, 2024).

Following the previous discussion on definitions that include animal intelligence and in light of advances in machine learning, the debate on what intelligence is has expanded to account for observed capabilities in artificial agents. One could argue that modern artificial intelligent agents are capable of purposefully acting as designed, aiming to minimize a loss function or maximize a reward. With sufficient capacity, they can solve complex problems (Silver et al., 2021), and some have even passed a form of the Turing test (Mei et al., 2024). This raises the question of whether AI can be considered intelligent, prompting a shift toward a more general concept of intelligence. This concept is not exclusive to humans or animals but instead moves toward a more parsimonious definition applicable to complex systems capable of processing information and acting accordingly to achieve a goal. Some examples of definitions of intelligence with this broader scope include: *The intelligence of a system is a measure of its skill-acquisition efficiency over a scope of tasks, with respect to priors, experience, and generalization difficulty* (Chollet,

2019); *intelligent agents are systems that show purposeful behavior* (Palanca-Castan et al., 2021); a functional definition of intelligence that can be measured Sritriratanarak and Garcia (2023); or *the capacity for completing novel goals successfully through respective perceptual-cognitive and computational processes* (Gignac and Szodorai, 2024).

There are several other definitions in the literature, but perhaps the broadest definition of intelligence, is the one provided by Legg and Hutter (2007). The authors surveyed several definitions of intelligence in the literature, and condensed them into: *Intelligence measures an agent's ability to achieve goals in a wide range of environments.*

1.2 The study of intelligence in brains and machines

There are numerous theories on how the brain may perform specific complex computations required for intelligent behavior. Some of these theories take the form of formal mathematical frameworks that map variables of the theoretical system to actual quantities measurable in experiments, such as neural activity or behavioral data. Successful examples include the activity of dopaminergic neurons as a reward prediction error and its evolution during learning (Schultz et al., 1997), sound localization in the medial superior olive (Grothe, 2000), spatial location encoded by place cells and grid cells (Moser et al., 2008), Gabor-shaped filters in retinal ganglion cells for efficient encoding of visual stimuli (Okajima, 1998), or the navigation mechanism of a fly implemented as a ring attractor (Angelaki and Laurens, 2020).

In general, theories of brain computation study different aspects, such as specific ways of connecting neurons in the form of neural architectures (Ocko et al., 2018; Schrimpf et al., 2020), plasticity rules governing the evolution of synaptic weights throughout learning (Citri and Malenka, 2008; Stampanoni Bassi et al., 2019), the algorithms underlying decision-making (Bogacz et al., 2006; Garcia et al., 2023), biological constraints and optimization objectives (Pulvermüller et al., 2021), and biologically plausible memory

systems (Nadel and Hardt, 2011; Krotov and Hopfield, 2021; Krotov, 2023), among others.

For decades, the way researchers design experiments and analyze data in cognitive neuroscience have been influenced from mathematical concepts developed in engineering and physics (Gershman, 2021; Gershman et al., 2024). However, in recent years, neural networks and reinforcement learning frameworks have become the leading mathematical approaches for describing brain activity and behavioral experiments, perhaps due to their success in learning how to solve complex problems and exhibit sophisticated behavior (Richards et al., 2019; Botvinick et al., 2020; Saxe et al., 2021). Examples of the use of these frameworks are ubiquitous in computational neuroscience. Notable cases include grid cell patterns emerging in the activity of recurrent networks, usually trained to perform path integration (Burak and Fiete, 2009; Whittington et al., 2020) (although this can be achieved without forcing the network to path integrate as in Weber and Sprekeler 2018); modeling the visual system as a deep convolutional neural network (Schrimpf et al., 2020; Lindsay, 2021); further characterizing dopaminergic activity distribution to encode uncertainty or feature-specific rewards (Dabney et al., 2017; Lee et al., 2024a); replay in the hippocampus as a value-based process used to update agent parameters (Mattar and Daw, 2018; Jensen et al., 2024); and a theory of semantic learning described through neural network dynamics (Saxe et al., 2019).

While these theories borrow concepts from the machine learning literature, there are also clear cases where machine learning models have been inspired by neuroscience experiments (Hassabis et al., 2017). A key example is the work of Hubel and Wiesel (1959), which mapped the activity of retinal ganglion cells to specific locations and orientations in the visual field. These results were later distilled and represented in the Neo-cognitron proposed by Fukushima (1988), which eventually led to modern convolutional neural networks trained via gradient descent (Lecun et al., 1998). Further examples of neuroscience-inspired machine learning concepts include attention mechanisms (Lindsay, 2020), episodic memory (Lengyel and Dayan, 2007; Pritzel et al., 2017; Giallanza

et al., 2024), and planning mechanisms (Sutton, 1991; Tomov et al., 2023).

The kinds of questions that cognitive neuroscience and machine learning seek to answer are generally different. From a neuroscience perspective, the focus is on questions such as: *What kind of algorithm is the brain running, and how is it implemented?* In contrast, machine learning asks: *What is the best algorithm for each scenario? How can we control and predict a model's behavior?* Despite these differences, both fields converge on a common question: *Why do these algorithms work, and what are the limits of these models?* As mentioned earlier, the overlap between cognitive neuroscience and machine learning is substantial, raising the question of whether it is beneficial to blur the lines between these fields given the current state of research. Some researchers have referred to this interdisciplinary approach as NeuroAI (Zador et al., 2023), describing research that explores the intersection and commonalities between neuroscience and artificial intelligence.

Given the complexity of the brain and modern machine learning systems, answering questions about what is implemented in the brain and why it even works is challenging. One common approach is to propose models, such as specific neural networks and reinforcement learning agents, as previously mentioned, and compare them to observed data from experiments. Then, by exploring the space of design choices for these agents and then simulating the functioning of the agent, researchers can identify the configuration that best explains the observed data. This approach has been named **a deep learning framework for neuroscience** (Lillicrap and Kording, 2019; Richards et al., 2019). A promising search space is that of objective functions, as it could provide insights into the purpose of biological systems, and search space such as learning rules, and architectures may offer clues about algorithmic implementations of learning and circuit-level implementations. Most research in computational neuroscience follows this methodology, as demonstrated by the previously discussed examples. However, relying purely on simulations and ablation studies in these systems is constrained by circumstantial confounders due to the large search

space of design choices, making it difficult to distill generalizable principles of intelligence.

Beyond this approach, it has been proposed to analyze these systems mathematically, in a manner closer to physics. This consists of finding relationships between the system's variables under well-defined assumptions, thereby providing guarantees and generality to some of the conclusions drawn from the analysis (Saxe et al., 2021). This is challenging, as both biological learning agents and machine learning systems resist mathematical analysis due to their intrinsic complexity. However, there are notable successes where important phenomena observed experimentally and in simulations can be described through mathematical relationships. Examples include the explanation of the double descent phenomenon in the loss function of overparameterized networks (Liu et al., 2022), analytical learning dynamics and asymptotic behavior of neural networks and reinforcement learning agents as a function of data statistics (Saxe et al., 2019; Goldt et al., 2019; Lee et al., 2022; Bordelon et al., 2023), intrinsically firing neurons as a mechanism for stability (Latham et al., 2000), or storage capacity in Hopfield networks (Folli et al., 2017).

Both approaches offer promising avenues for gaining insight into the mechanisms of intelligence. However, it is important to distinguish the types of questions each approach can address and the corresponding answers they can provide. For instance, the simulation-based approach can help identify relevant or prominent behaviors of intelligent systems that may be explained through mathematical analysis. Conversely, mathematical results can be tested beyond their usual restrictive assumptions in real-world scenarios with larger and more complex intelligent systems, potentially validating the generality of the analysis. This thesis adopts both approaches, drawing conclusions from extensive computational simulations as well as mathematical analysis to uncover meta-learning principles in artificial and biological learning systems. Having broadly outlined the study of intelligent systems, the discussion now turns to a specific and perhaps essential feature of intelligence: *learning*.

1.3 Learning

Similarly to the concept of intelligence, most people have an intuitive grasp of what learning means. It can involve adapting to new environments, acquiring information that may be useful in the future, or changing behavior based on experience. Once again, we observe intelligent beings demonstrating the ability to learn. For instance, children learn how to speak and improve their reasoning skills during development (Feldman, 2019), people learn to play musical instruments (Hille and Schupp, 2015), dogs learning how to behave among humans (Hiby et al., 2004), and neural network models learn to detect supernova explosions and other astronomical events in images (Carrasco-Davis et al., 2021). These examples illustrate that learning has many facets and may exist in different forms. As with intelligence, learning is not straightforward to define, and researchers have debated its definition in the past.

Historically, most early discussions about the nature of learning came from psychology, as it is closely associated with human development and education (Spurlock, 2023; Renshaw and Power, 2003). According to Bruner (2004), whose ideas we briefly summarize here, theories of learning, at least in psychology, were split from the beginning into two main paradigms: *Associative learning*, which conceptualizes learning as the formation of bonds between ideas based on proximity (whether conceptual, spatial, or temporal) (Ebbinghaus , 1885; Pavlov, 1906), and *Configurationism*, which proposes that learning involves organizing information in such a way that the overall structure enables learning, rather than each individual piece of information on its own (Tolman, 1948; Krechevsky, 1975). Hence, configurationism stands in opposition to associativity in terms of the specific mechanisms of information acquisition that constitute learning.

The discussion between the two groups of researchers in each paradigm had rivalry undertones. For example, Skinner (1950, 1985) argued that a formal theory of learning might not be necessary, as associativism alone was sufficient. In contrast, Hull (1943)

advocated for a theory of associative learning by proposing mathematical expressions describing relevant learning variables. On the configurationist side, Tolman (1948) introduced the concept of a cognitive map organized based on its utility, while Krechevsky (1975) emphasized that learning was hypothesis-driven rather than a passive process, as in associativism, which relies merely on proximity. Over time, the emphasis on distinguishing between the two paradigms diminished as researchers began to consider information processing systems and language acquisition in a broader sense (Miller et al., 1960; Newell and Simon, 1972). This shift was followed by a decline in theories focusing on the basic components of learning that the earlier paradigms had attempted to capture, since language learning and other complex cognitive processes were highly abstract. Theories of learning were instead developed at the level of language itself (Chomsky, 2020), this shift perhaps revived some aspects of the earlier paradigms (Chomsky, 2013), at least within the field of psychology. See Bruner (2004); Bélanger (2011) for an extended discussion on the history of learning in psychology.

As one of the AI winters ended with the resurgence of neural network applications for solving complex problems (Toosi et al., 2021), some researchers leveraged computational models capable of adapting to solve problems, essentially, learning. From this foundation, numerous theories of learning have been proposed, including those based on neural networks and reinforcement learning, as discussed in the previous section.

Following this approach, learning in this thesis is broadly defined as *the process by which a dynamical system evolves to improve on a task or a set of tasks*, with neural networks and reinforcement learning agents serving as specific instances of this definition.

1.4 Learning to Learn

Throughout life, individuals not only learn but also *learn how to learn*. For example, at the end of a university term, students must begin preparing for exams. How should

they decide how much time and effort to allocate to studying for each course? Many factors influence this decision, including prior knowledge in each subject, the relevance of each course to their degree and future plans, the level of difficulty, and even personal engagement with the material. Humans and other animals make similar decisions daily, where the effort invested in the present influences future outcomes. This ability to manage learning is crucial for long-term planning, as it requires estimating the potential outcome of knowledge not yet acquired, an ability distinct from the direct act of studying.

Numerous empirical observations indicate that humans can effectively schedule their own learning. For instance, spontaneous matching of stimulus difficulty with skill, even among children, may facilitate learning (Kidd et al., 2012); intrinsic valuation of new information can promote exploration (Bromberg-Martin et al., 2024); and self-directed management of learning curricula has been documented (Ten et al., 2021). Additionally, evidence suggests the existence of an optimal difficulty level for rapid learning (Wilson et al., 2019), active stimulus selection based on information content in children (Raz and Saxe, 2020), and the relative benefits of mental rest and periods of boredom throughout the learning process (Agrawal et al., 2022).

More specifically, two recent empirical studies, one in rats (Masís et al., 2023) and another in humans (Masis et al., 2024), have directly investigated the role of cognitive control in learning. Although substantial evidence supports the role of cognitive control in learning management, theoretical work has largely addressed this process in a problem-specific manner (e.g., examining how control governs the development of shared or distinct representations across different environments (Sagiv et al., 2020; Ravi et al., 2021)) and has relied on normative but simplified models for tractability reasons (Masís et al., 2021). There remains a need for a comprehensive, normative, and, importantly, scalable framework to guide optimal control over an agent's learning repertoire.

In addition to psychological experiments, researchers have directly examined the brain

during tasks requiring cognitive control allocation, with evidence suggesting that specific brain regions are dedicated to computing key aspects of learning control. For example, activity in the dorsal anterior cingulate cortex (dACC) has been shown to correlate with the expected payoff associated with a task, the level of cognitive control required to achieve a goal, and the cost of engaging in the process (Shenhav et al., 2013; Klein-Flügge et al., 2016). Meanwhile, the locus coeruleus (LC), the brain's primary source of norepinephrine and a recipient of signals from the ACC, has been linked to attentional states during task learning, modulating exploration and exploitation (engagement or disengagement) while performing a task (Aston-Jones and Cohen, 2005; Breton-Provencher et al., 2021; Zhang et al., 2023; Fan et al., 2023).

While these studies provide important insights into how the brain may act optimally in certain situations, less is known about how such processes operate over extended learning timescales. Additionally, several other phenomena influence learning processes. Theories that aim to maximize cumulative reward have attempted to unify control over learning, such as the expected value of control (EVC) theory (Shenhav et al., 2013; Masís et al., 2021), and the role of neuromodulators as meta-learners (Doya, 2002; Lee et al., 2024b), which assign specific roles to neurotransmitters in the regulation of learning processes (Iigaya et al., 2018; Chantranupong et al., 2023).

Controlling learning processes is also a critical topic in machine learning (Vettoruzzo et al., 2024). The challenge of controlling learning is framed as identifying the optimal parameters for a learning agent to enhance its ability to acquire new tasks. This can take several forms, such as determining optimal initial parameters (Finn et al., 2017), developing methods for representing new data (also known as domain adaptation Li et al. 2018), maintaining long-term learning while preventing the forgetting of previously acquired tasks (Son et al., 2024; Lee et al., 2024b), and, more recently, enabling in-context learning in large language models (LLMs, Coda-Forno et al. 2023).

Most applications of these methods are numerical, and there is a lack of analytical frameworks through which such systems can be examined mathematically. However, recent advances in learning theory (Saxe et al., 2019; Goldt et al., 2020; Lee et al., 2022) and control theory (Atangana et al., 2014; Zucchet et al., 2022) are expanding the frontiers of mathematically tractable meta-learning systems. Additionally, relevant analogies between meta-learning and animal behavior have been proposed, such as the role of neurotransmitters as hyperparameters of a reinforcement learning agent (Doya, 2002; Lee et al., 2024b), the prefrontal and orbitofrontal cortex as meta-learners (Wang et al., 2018; Wang, 2021; Hattori et al., 2023), and human-like generalization capabilities in meta-learning models (Lake et al., 2017; Lake and Baroni, 2023; Binz et al., 2023).

Besides keeping track of the agent's performance, which is required for optimal cognitive control, it is necessary to estimate which aspects of the environment are useful to consider for learning. When an agent has the choice to focus on different pieces of information from the environment, and assuming there is limited capacity to focus on many things at the same time, it is beneficial to focus on aspects that can be learned and that are reward-related, as opposed to information that is random, unpredictable, and unlearnable. Such information does not promote an increase in performance or collected reward when trying to learn it (Burda et al., 2018; Mavor-Parker et al., 2022). Identifying learnable parts of the environment can be framed as an *uncertainty decomposition* problem, separating uncertainty into *epistemic* and *aleatoric* uncertainty, the learnable and unlearnable parts, respectively. This problem has been widely studied in machine learning, specifically when estimating posterior distributions of a model (Lahlou et al., 2022), or to promote exploration in reinforcement learning agents (Clements et al., 2020; Lobel et al., 2023). Being able to find epistemic knowledge in an environment while learning is necessary to improve learning performance, in a different way than simply looking at the trajectory. In a way, epistemic uncertainty prioritization is a proxy that works well to improve learning, without necessarily dealing with the complexities of considering the learning trajectory

itself, which requires further meta-knowledge from the agent, learning dynamics, and task statistics. This makes it suitable as a surrogate heuristic to control learning that can be applied to more complex models.

These connections suggest that the meta-learning framework from machine learning and the cognitive control of learning in biological agents are fundamentally related and should be explainable within a unified framework.

1.5 Thesis Structure

After the introduction and background in chapter 1 and chapter 2, respectively, the novelty of this thesis is presented concretely in three parts. chapter 3 presents a formal mathematical framework that relates the expected value of control in cognitive neuroscience with meta-learning in machine learning. This framework considers the optimal learning trajectory as the one that maximizes cumulative reward over the available time to interact with an environment, and it allows for the computation of numerical solutions of the optimal control for learning. This normative objective is highly expressive (Abel et al., 2021), and thus manages to instantiate many problems that require the adjustment of parameters or control variables outside the defined learning dynamics. The framework is applied to multiple learning settings, including curriculum learning as attention to multiple tasks while learning, and task switching for simple neural networks through weight modulation. In addition, this framework provides an alternative explanation to experimental data (Masís et al., 2023).

chapter 4 simplifies the assumptions of the meta-learning framework to obtain an approximation of the optimal control of learning. The effort of pushing the theory to obtain a mathematical expression of the optimal control relies on the need for insight beyond the experimental setting, which is the common scenario in most neural network applications. Usually, models have many design choices and parameters, which can be optimized using

the framework proposed in chapter 3. The numerical solution for the optimal control could change depending on these parameters. Having a mathematical expression provides explicit dependency of the optimal control for learning on every parameter without the need for numerical simulation, yielding a parsimonious expression to compute the control of learning and highlighting the parameters that *are relevant* versus the ones that are not. The analysis presented in chapter 4 is a novel approach that utilizes control theory perturbation methods (Atangana et al., 2014; Ganjefar and Rezaei, 2016) to approximate the normative objective: cumulative reward throughout the learning period and the optimal learning rate scheduling under an assumed cost.

In chapter 5, we move to the study of uncertainty decomposition, which, as discussed before, corresponds to a proxy or heuristic of reward-optimal control for learning. Similar to the methods in Chapters 3 and 4, this approach requires meta-cognitive information about the agent's uncertainty. The proposed uncertainty decomposition is a novel mathematical expression that relies on ensembles (Osband et al., 2016) and value distribution estimation (Dabney et al., 2017) as meta-cognitive knowledge of the agent. It is used specifically in the context of prioritized experience replay (Schaul et al., 2016), where an uncertainty-based priority scheme is implemented to select memories from the replay buffer that are high in epistemic uncertainty and low in aleatoric uncertainty. The meta-learning framework presented in chapter 3 computes the epistemic content implicitly. This point is further developed in the discussion presented in chapter 6.

Chapter 2

Background

In this chapter, a formal overview of the mathematical background utilized in this work is provided, from which the proposed meta-learning strategies framework and other control mechanisms to assist learning are derived. This work spans the use of deep neural networks (DNN) and reinforcement learning agents as dynamical systems. In general, DNN and RL-agents are used for different purposes, DNNs are usually utilized as function approximations, while RL-agents to learn policies to control a system or optimally behave within an environment. These two types of agents are integrated to create the proposed meta-learning framework, in short, a neural network (or an RL-agent) is a learning system that needs to be controlled to maximize cumulated reward through training as in an RL problem, and this controller needs to solve the optimization problem considering the learning dynamics of the DNN. The details will be presented in chapter 3.

Examining various components that enable learning, such as learning curricula, inter-temporal control allocation, and theories of their neural implementation, and revealing how to control these components to improve learning, constitutes meta-learning. In this work, it is generally assumed an underlying learning algorithm that governs the learning dynamics and trajectory of the agent, while meta-learning modifies adjacent components

to enhance this learning. One of the learning components that is further explored is the prioritize experience replay methods used in DRL, here modified to account for uncertainty of memories.

Then a formal description of cognitive control theories linked to managing learning is presented. It is followed by a review of several mechanisms from the meta-learning literature in machine learning. Using this formalization of cognitive control and meta-learning, a unified mathematical normative framework is proposed, that can instantiate multiple algorithms used in cognitive neuroscience and meta-learning in machine learning.

2.1 Learning systems

This section starts by revisiting the basic definitions of Deep Neural Networks and Reinforcement Learning Agents. These learning systems will later be described in a unified manner as dynamical systems, allowing their incorporation into the meta-learning strategies framework. This approach abstracts away the specific learning algorithms, which can then be controlled to improve learning. Although this description is quite general, it is arguably more suitable for neural networks than for RL agents. While it is possible to describe RL agent learning as a dynamical system, this perspective is less explored and more challenging because the data observed by the agents is non-i.i.d. and policy-dependent (Bordelon et al., 2023). On the other hand, neural networks are naturally translated into a dynamical system under some assumptions (Saxe et al., 2019; Goldt et al., 2019; Bordelon and Pehlevan, 2022), where the system state corresponds to the weights of the network, and the dynamical equations are defined by its learning rule (e.g., backpropagation, Hebbian rules, etc.).

2.1.1 Deep Neural Networks

DNN models are considered as functions that map an input $X \in \mathbb{R}^I$ to an output $\hat{Y} \in \mathbb{R}^O$. This function $\hat{Y} = f(X;\theta)$ is described as a function of its parameters θ, and it is usually the result of a composition of functions called *layers*, denoted here as $H_l = f_l(H_{l-1};\theta_l)$, where H_l represents the representation at layer l, and f_l is the transformation from H_{l-1} to H_l parameterized by θ_l. Hence, a neural network with L layers can be expressed as

$$\hat{Y} = f_L \circ f_{L-1} \circ \cdots \circ f_1(X) \tag{2.1}$$

where $\circ$ denotes the composition of functions, and the dependency on θ is omitted for simplicity. There are several ways to construct the layers in a neural network, such as fully connected, convolutional, or residual layers. For time-dependent data, recurrent layers and, more recently, *attentional* layers have been utilized (Alzubaidi et al., 2021).

The reason these models are called neural networks is that their building blocks are neurons, and a common way of characterizing each layer function is by its pre-synaptic neuron activity $H_{l-1} \in \mathbb{R}^{N_{l-1}}$ and post-synaptic neuron activity $H_l \in \mathbb{R}^{N_l}$, with the number of neurons in each layer given by N_{l-1} and N_l respectively. The neuron activity is then transformed by a (fully connected) layer according to

$$H_l = f_l(H_{l-1}) = \sigma_l\left(W_{l-1}H_{l-1} + b_l\right) \tag{2.2}$$

where $\sigma_l(\cdot)$ is an element-wise function, $W_l \in \mathbb{R}^{N_l \times N_{l-1}}$ are called *synaptic weights* or simply the weights of layer l, and b_l is the bias of layer l, both of which constitute the parameters of the layer, $\theta_l = \{W_l, b_l\}$. In this work, most neural networks presented have fully connected layers unless specified.

These DNNs are universal approximators when σ is a non-linear function (Hornik et al., 1989), and they can be used to *learn* mappings from X to Y directly from data. To

achieve this, it is necessary to define how close the network's estimation or output $\hat{Y}$ is to the actual target Y. This is typically referred to as the loss function, denoted as $\mathcal{L}(Y, \hat{Y})$, which takes lower values when the target and the estimation are close to each other. Examples of loss functions include Mean Squared Error (MSE), which is commonly used for regression problems, and Cross-Entropy (CE) loss, which is frequently applied to discrete outputs, such as in classification problems. In these cases, finding the set of parameters θ that adjusts the network functions to match the given dataset $(x_i, y_i)_{i=1...N}$ with N pairs of data is known as supervised learning, as the target output Y is explicitly provided to the model.

Other types of DNNs operate in an *unsupervised* setting, where no target output is given. A notable example is autoencoders, in which a neural network maps $\hat{X} = f(X; \theta)$ to *reconstruct* the input. An important consideration here is that if f and θ have sufficient capacity, they could simply learn an identity function, resulting in $\mathcal{L}(X, \hat{X}) = 0$. However, introducing a bottleneck in the autoencoder forces the model to learn a *compressed representation* of the data (Chen and Guo, 2023). This type of unsupervised loss can serve as an auxiliary task to improve generalization (Le et al., 2018) and assist DRL agents when rewards are sparse (Prakash et al., 2019).

Given a dataset, e.g., $(x_i, y_i)_{i=1...N}$, how can the network learn a mapping from X to Y? As mentioned earlier, a network output $\hat{Y}$ that is closer to the target Y results in a lower loss $\mathcal{L}(Y, \hat{Y})$. It is possible to adjust the network parameters θ by taking steps such that the loss function decreases for the given data. To achieve this, it is necessary to determine the direction in which θ should be moved to minimize the loss function. The standard approach for this is to update the parameters θ iteratively using gradient descent (GD), which follows the direction of the steepest decrease in $\mathcal{L}$ as a function of θ, given by its gradient with respect to the parameters. The gradient step updates from iteration k to

iteration $k+1$ are given by:

$$\theta^{k+1} = \theta^k - \alpha \left\langle \frac{d\mathcal{L}}{d\theta} \right\rangle_{XY} = \theta^k - \alpha \left\langle \frac{d\mathcal{L}(Y, \hat{Y})}{d\hat{Y}} \frac{d\hat{Y}}{d\theta} \right\rangle_{XY}, \tag{2.3}$$

where α denotes the hyperparameter known as the *learning rate*. To compute the update, it is necessary to evaluate these gradients using elements in the dataset $(x_i, y_i)_{i=1\ldots N}$. However, computing the gradient for the entire dataset is often infeasible. A common solution is to subsample (x_i, y_i) pairs (also called a batch) to estimate the gradient at each time step. This variation is known as stochastic gradient descent (SGD) and is given by:

$$\theta^{k+1} = \theta^k - \frac{\alpha}{B} \sum_{i=1}^{B} \frac{d\mathcal{L}(y_i, \hat{y}(x_i; \theta))}{d\theta} \tag{2.4}$$

where B represents the *batch size*. When the parameters correspond to the weights of two or more layers in a neural network, the process is referred to as *backpropagation* (Werbos, 1982; Rumelhart and McClelland, 1987), which consists of applying the chain rule to compute the loss gradient for the weights of each layer.

The specific expression utilized to iteratively update the parameters θ by using the loss gradient, as in the previous equation, is usually called an *optimizer* in machine learning (SGD being just an example of an optimizer) and a *learning rule* or *plasticity rule* in computational neuroscience. In machine learning, most optimizers rely on computing the first-order gradient of the loss function with respect to its parameters (Ruder, 2017; Schmidt et al., 2021). These methods often use adaptive learning rates to avoid getting stuck in local minima, incorporating techniques such as momentum (Nesterov, 1983) or tracking higher-order moments of the gradient, e.g., RMSProp (Graves, 2014), ADAM as the default option in most applications (Kingma and Ba, 2017) or AMSGrad (Reddi et al., 2019). Because the majority of these optimizations are non-convex, convexity of the landscape is useful in gradient-based optimization. Some methods rely on computing the

second-order derivative of the loss with respect to the parameters (Tan and Lim, 2019; Anil et al., 2021). Although these methods generally perform better than first-order methods, they are expensive to compute and do not scale well with the number of parameters in the model, making them prohibitively expensive given the current size of neural network models.

Synaptic weight changes in the brain throughout learning have been widely studied through experiments and simulations (Citri and Malenka, 2008; Magee and Grienberger, 2020; Piette et al., 2023). However, a general plasticity rule (or rules) that enables the brain to learn complex tasks has yet to be identified (Bell et al., 2024; Confavreux et al., 2024). Learning rules in the brain are subject to biological constraints; therefore, not all possible update rules for synaptic weights are suitable candidates for learning in the brain (Shervani-Tabar and Rosenbaum, 2023). For instance, researchers have argued that backpropagation, and consequently, gradient descent on synaptic weights, cannot be implemented in biological networks. The main reason is that backpropagation requires a highly specific set of computations, such as propagating the error signal backward through all layers and having access to the transposed version of weight matrices, which implies knowledge of the backward propagation of neuron activations. It is believed that this information is not accessible at the synaptic connections that need to be modified. However, some argue that even if backpropagation cannot be computed at the level of individual synaptic connections, gradient descent may still be implemented through other rules that could approximate backpropagation (Whittington and Bogacz, 2019; Lillicrap et al., 2020).

Several bioplausible alternatives to backpropagation have been proposed, with some experimentally validated. The most well-known among them are the family of *Hebbian plasticity rules*. The principle of "*neurons that fire together wire together*" suggests that synaptic weights generally increase when pre- and post-synaptic activity rises simultaneously. Unlike backpropagation, this process does not rely on external signals beyond

what is available on the corresponding synapse (Morris, 1999; Johansen et al., 2014; Fox and Stryker, 2017). A broader category of learning rules considered biologically plausible are *local learning rules*, which only require information that is local to the synapse undergoing modification (Bell et al., 2024; Confavreux et al., 2024). One example of such a rule is *predictive coding*, which states that synaptic weights are adjusted to predict upcoming information and generate an appropriate neural response. This is computed by first inferring the neural activity in the network necessary to generate the correct input and output, and then adjusting synaptic weights to produce that activity. This process requires only local variables: pre- and post-synaptic neural activity and the corresponding synaptic weight (Millidge et al., 2022; Song et al., 2024). Other rules that are non-local but still arguably biologically plausible exist. For instance, *feedback alignment* has been proposed as a biologically plausible rule. It involves backpropagating the error signal using random matrices instead of the exact transposed or reversed weight matrices required in traditional backpropagation (Lillicrap et al., 2016). Another example is the class of three-factor rules, which combine local learning mechanisms such as Hebbian learning with a rough error signal conveyed through neuromodulation, such as dopamine or norepinephrine signals (Kuśmierz et al., 2017). Most biologically plausible learning rules have performance shortcomings compared to backpropagation. Hebbian rules and feedback alignment do not achieve the same level of performance when training networks (Shervani-Tabar and Rosenbaum, 2023). On the other hand, predictive coding can perform as well as, or sometimes better than, backpropagation (Song et al., 2024), though it is more computationally expensive when training artificial networks.

Backpropagation has been highly successful in training DNN models to solve various problems, such as protein folding (Jumper et al., 2021) and design (Watson et al., 2023), text processing with LLMs (Raiaan et al., 2024), classification of astronomical data (Förster et al., 2021), and control of fusion reactions using Deep-RL (Degrave et al., 2022), among others. An important caveat of these methods is the difficulty in understanding

the logic implemented during the inference process (from input to output) to explain the network's behavior. This is why these models are often referred to as *black boxes*. Even more challenging is predicting their evolution throughout training, specifically, the path the loss function will follow, also known as the learning trajectory, and the precise weights during training. This makes designing training processes challenging and largely reliant on heuristics and brute force hyperparameter search, which have been effective in practice (Ahmed et al., 2023). In this thesis, we focus on studying simpler models where mathematical analysis is feasible, which we then support with simulations on more complex models. Further mathematical characterization of neural networks as dynamical systems will be presented in subsection 2.1.3.

2.1.2 Reinforcement Learning Agents

Reinforcement Learning (RL) is a mathematical framework to model an agent interacting with its environment, and provides methods to describe learning agents that improve their behaviour by taking actions and experiencing the consequences of their actions in this environment. In a way, RL gives both *ways to pose the problem and ways to solve it*. In this section a standard description of an RL framework is provided, for a more in-depth overview of RL methods see (Sutton and Barto, 2018).

Consider an environment modelled by a Markov Decision Process (MDP). In short, an MDP describes a process where the next state distribution is fully determined by the current state $s \in S$ and an action taken $a \in A$. The dynamics of this process are defined by $P(s', r|s, a)$ which characterizes the probability of moving to state s' and observe the scalar signal r (called reward) given that the system is on state s and action a is taken, and it is called the *state-transition function*. In an environment modeled by an MDP, the entity that takes the actions is called an Agent, and its behavior is defined by a mapping from states to actions called policy $\pi(a|s) : S \times A \to \Delta(A)$ where $\Delta(A)$ denotes the probability simplex over A.

2.1.2.1 Value-Based RL

The agent interacting with an environment could be setup to achieve a *goal*, and many goals can be indicated using the reward signal. The limits of the use of the reward signal to express tasks have been studied in Abel et al. (2021), and it has been argued that it can be used to specify preferences for artificial agents in general, and potential objectives in biological agents (Silver et al., 2021). Once the reward distribution is defined as a function of the task at hand in the environment, the optimal policy that describes the ideal agent behavior for that task $\pi^*(a|s)$ is the one that maximizes the expected discounted cumulative reward, also known as *expected return*, starting from any state s denoted

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t | s \right] \tag{2.5}$$

with

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... \tag{2.6}$$

where t indexes time, and γ denotes the discount factor, increasing the importance of future rewards when γ is increased. Assuming the agent does not have access to π^* when learning a new task, how can it find the optimal policy that gives the best possible outcome $V^*(s)$, how can the agent improve its initial policy to maximize the expected return?

One way of learning an optimal policy, among many others that will be described later, is by estimating the expected return from a state s for all available actions a, then select the action that maximizes the expected return. This form of learning is called Q-learning (Mnih et al., 2015), and it is posed formally as follows. The action-value function is

defined as

$$Q(s,a) = \mathbb{E}_\pi \left[G_t | s, a \right], \tag{2.7}$$

which is similar to the value function V_π except now it is conditioned to an action as well. Then, the optimal policy given $Q(s,a)$ is simply

$$\pi^*(a|s) = \arg\max_a Q(s,a), \quad \text{and} \quad V^*(s) = Q(s, \arg\max_a Q(s,a)). \tag{2.8}$$

As mentioned before, an agent does not necessarily comes with the knowledge of the value function given its current policy, hence it needs to learn it. Learning Q would allow the agent to define its policy based on the estimate action-value function. First, Q needs to be parameterized, which parameters θ, and initialize it from a prior distribution to start with a guess of this function. An easy way to learn Q_θ is simply by collecting trajectories of the agent exploring the environment, $\mathcal{T}_e = (s_0, a_0, r_0, s_1, a_1, r_1, ...s_{N_e}, a_{N_e}, r_{N_e})$ throughout episodes, with N_e the number of steps it takes to terminate an *episode*, and use the observed cumulated discounted reward to estimate θ. This approach, however, comes with some drawbacks, e.g. storing entire trajectories can be very costly depending on the length of the episode (not practical for open-ended environments), and the trajectories can have a large variance, as the space of possible state-action-reward observation can be very large, requiring multiple instances of trajectories to stabilize learning. A more practical approach is to take advantage of the recursive nature of the value function, and the action value function to approximate Q with Q_θ. The action-value function satisfies

the following recursive equation

$$
\begin{aligned}
Q(s_t, a_t) &= \mathbb{E}_\pi \left[G_t | s_t, a_t \right] && (2.9)\\
&= \mathbb{E}_\pi \left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... \right] && (2.10)\\
&= \mathbb{E}_\pi \left[r_{t+1} + \gamma G_{t+1} | s_t, a_t \right] && (2.11)\\
&= \mathbb{E}_\pi \left[r_{t+1} | s_t, a_t \right] + \gamma \mathbb{E}_\pi \left[G_{t+1} | s_t, a_t \right] && (2.12)\\
&= \mathbb{E}_\pi \left[r_{t+1} | s_t, a_t \right] + \gamma \sum_{s', a'} P(s'|s_t, a_t) \pi(a'|s') Q(s', a') && (2.13)
\end{aligned}
$$

taking $\pi(a|s) = \arg\max Q(s, a)$ gives

$$
Q(s_t, a_t) = \sum_r P(r|s_t, a) \cdot r + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q(s', a'). \tag{2.14}
$$

This relation is called *Bellman Equation* for the action-value function, and there are similar Bellman Equations for the value function, and their optimal versions of it (Chapter 3 in Sutton and Barto 2018). From here, a method to estimate Q_θ is by enforcing this equation in the estimation. Given a transition within a trajectory denoted by (s_{t+1}, r_t, s_t, a_t), a loss function that can be used to learn the parameters θ is

$$
\mathcal{L}(\theta) = \frac{1}{2} \left(r + \gamma \max_{a'} Q_\theta(s', a) - Q_\theta(s, a) \right)^2, \tag{2.15}
$$

and it is called *squared temporal difference error*. Then, a simple way of estimating θ iteratively is by taking gradient steps that minimize this loss

$$
\begin{aligned}
\theta^{k+1} &= \theta^k - \alpha \frac{dL}{dQ} \frac{dQ}{d\theta} && (2.16)\\
&= \theta^k + \alpha \underbrace{\left(r + \gamma \max_{a'} Q_\theta(s', a) - Q_\theta(s, a) \right)}_{\delta} \frac{dQ}{d\theta} && (2.17)
\end{aligned}
$$

with δ called the *temporal difference error* (TD-error). RL algorithms that rely on

TD-error are varied, and it has direct connections to dopamine being a TD-error signal (Schultz et al., 1997; Dabney et al., 2020; Lee et al., 2024a), and to other neurotransmitters (Doya, 2002; Shenhav et al., 2013; Lee et al., 2024b).

There are many RL algorithms that can learn how to solve a task by gradually improving their policy, and in each of them, the learning algorithm and the structure of the agent can change dramatically. In addition to this, there are many heuristics and factors to consider in order to make these agents work in practice (see subsubsection 2.1.2.2).

2.1.2.2 Deep Reinforcement Learning

Adjusting the parameters of the action-value function θ to minimize the squared temporal difference error loss is essentially a function approximation method, making it well-suited for using DNNs. Using a neural network as a function approximator in the context of RL, e.g., to approximate $Q_\theta(s, a)$, is called Deep Reinforcement Learning (DRL). Most modern applications of RL agents incorporate neural networks to approximate functions in one form or another, as discussed in the previous section.

There are several practical considerations and heuristics to improve training. A major aspect of RL training is the *exploration-exploitation trade-off*. Exploring may require sacrificing some immediate performance but could provide access to better reward sources for exploitation later. As an agent collects experiences from the environment, it might get stuck in suboptimal strategies. For instance, following $\pi(a|s) = \arg\max_a Q(s, a)$ early on may lead to poor decisions since the initial estimates of the action-value function may not accurately represent the true value function in the environment. One way to mitigate this issue is by using an ϵ-greedy policy, where the selected action from any state s is drawn as $\arg\max_{a \in A} Q_\psi(s, a)$ with probability $1 - \epsilon$ and uniformly over A otherwise. There are many other approaches to addressing this trade-off, such as rewarding curiosity-driven exploration by estimating occupancy (Lobel et al., 2023) or assessing the uncertainty of visited states (Mavor-Parker et al., 2022).

Another important aspect of training is the coupling between temporally nearby experiences and value estimation. If the agent is trained primarily on its most recent experiences, the value function approximation could overfit to the current policy, locking the estimation into a local minimum. To alleviate this issue, a common technique is to use a *replay buffer*, which stores experiences and later samples them randomly to decorrelate them temporally (Lin, 1992). This also improves gradient estimation, as a batch of transitions is sampled to compute an average update in equation 2.17.

Additionally, since the target that Q_θ is trying to estimate depends on itself, a copy of the network $\bar{\theta}$ is often maintained, such that

$$\delta = r + \gamma \max_{a'} Q_{\bar{\theta}}(s', a) - Q_\theta(s, a) \tag{2.18}$$

which stabilizes training by keeping the target fixed. This target network is then periodically updated with the current one, $\bar{\theta} = \theta$, every few iterations (Watkins and Dayan, 1992; Mnih et al., 2015).

Many other techniques and heuristics are applied to DRL agent training, such as prioritized experience replay (Schaul et al., 2016), explained in subsection 2.1.4 and further improved in subsection 5.2.1 as a contribution of this thesis, double Q-learning (van Hasselt et al., 2015), and multi-step learning instead of the single-step TD error (Sutton, 1988). Additionally, various training considerations specific to DNN models play a role in DRL. All these components interact and are typically optimized jointly, often through a grid search over all options, which is computationally expensive as it requires training multiple models (Hessel et al., 2017; Eimer et al., 2023). For a comprehensive overview of RL methods, see (Sutton and Barto, 2018).

2.1.3 Learning Dynamics

For a learning system like the ones described in the previous section, their corresponding *Learning Dynamics* describes the evolution of performance and the parameters that define the system throughout the training of a task (or tasks). Many aspects of learning affect the learning dynamics of a system, such as the loss function being optimized (Tachet et al., 2020), the initial parameters (Saxe et al., 2019; Braun et al., 2022), or task similarities in continual learning (Lee et al., 2022), among many others.

Most knowledge about the effect of each component on the learning dynamics in neural networks and RL agents is based on experience while training these systems to solve specific tasks, such as in Kanervisto et al. (2021); Ahmed et al. (2023) for deep neural networks or Hessel et al. (2017); Schwarzer et al. (2023) for DRL. Understanding the learning dynamics of learning systems is usually done experimentally by testing different design choices and reporting the resulting changes in learning trajectories. The main reason for this approach is that learning systems resist mathematical analysis in most cases making it challenging to derive learning principles, as relevant models become increasingly complex and data distributions cannot be described in simple terms to allow for mathematical analysis.

Why does learning dynamics in neural networks resist mathematical analysis? The description of performance and parameters throughout learning can be represented, for instance, by a loss function applied to the entire data distribution, $\langle \mathcal{L}(t) \rangle_D$, over training time t, and by the evolution of its parameters $\theta(t)$, which are necessary to compute the system's performance. For a simple system trained using gradient descent, the change in parameters over training time is governed by

$$\theta(t + \Delta t) = \theta(t) - \alpha \left\langle \frac{d\mathcal{L}}{d\theta} \right\rangle_D , \tag{2.19}$$

where α is the learning rate, D denotes the expectation with respect to the dataset used to

train the learning system, and Δt is an arbitrarily small time increment. Taking $\alpha \approx \Delta t$ and dividing both sides by the learning rate, then letting $\alpha = \Delta t \to 0$, the equation becomes

$$\frac{d\theta}{dt} = -\left\langle \frac{d\mathcal{L}}{d\theta} \right\rangle_D, \quad \text{s.t. } \theta(t=0) = \theta_0, \tag{2.20}$$

where θ_0 represents the initial parameters before training. This description of learning dynamics is called *Gradient Flow*, which can be thought of as the continuous counterpart to the discrete updates of gradient descent (Elkabetz and Cohen, 2021). Solving this differential equation would allow for a description of the evolution of the parameters $\theta(t)$ and the generalization error $\langle \mathcal{L}(t) \rangle$ over training time. However, in general, this differential equation cannot be solved due to two specific challenges. First, computing the expectation on the right-hand side requires evaluating a highly complex integral that depends on factors such as the model architecture, activation function, data distribution, and more. It also depends on how the training data is presented to the network, e.g., one sample at a time, in batches, and whether data points are repeated during learning, commonly referred to as online learning vs. batch training. Second, even if the expectation can be computed, the resulting expression must be in a form that allows the equation to be integrated to obtain $\theta(t)$, which is not possible in most cases.

Developing a mathematical theory of learning dynamics could help identify principles of learning that depend on different aspects of the learning system. To achieve this, theorists have simplified learning agents into models that allow for mathematical analysis while retaining some (but not all) features of the learning trajectories observed in complex models. In this thesis, Deep Linear Networks are used for this purpose, as their dynamics can be expressed in closed form while preserving certain properties observed in more complex models

2.1.3.1 Deep Linear Networks

Deep Linear Networks allow for mathematical analysis, and in the particular case of two-layer networks, they even admit a closed-form solution of their generalization error under certain assumptions (Saxe et al., 2019; Braun et al., 2022). For a two-layer linear neural network, the input-output mapping defined by the network is

$$\hat{Y} = W_2(t)W_1(t)X, \tag{2.21}$$

where $X \in \mathbb{R}^I$, $\hat{Y} \in \mathbb{R}^O$, $W_1(t) \in \mathbb{R}^{H\times I}$, and $W_2(t) \in \mathbb{R}^{O\times H}$ are the weights of the first and second layers, respectively, I, H, O being the dimensions of the input, hidden layer and output. In addition to minimizing the mean squared error, neural networks are often trained with weight regularization. This means that the loss function includes the Frobenius norm, which penalizes the magnitude of the weights in each layer. Thus, the loss function to be minimized in this case is

$$\mathcal{L} = \frac{1}{2}\|Y - \hat{Y}\|^2 + \frac{\lambda}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right). \tag{2.22}$$

Taking the gradient with respect to the weights in each layer yields

$$\frac{d\mathcal{L}}{dW_1} = -W_2^T Y X^T + W_2^T W_2 W_1 X X^T + \lambda W_1, \tag{2.23}$$

$$\frac{d\mathcal{L}}{dW_2} = -Y X^T W_1^T + W_2 W_1 X X^T W_1^T + \lambda W_2. \tag{2.24}$$

This gradient is evaluated for an arbitrary set of points X and Y in the dataset, which could correspond to a single point, a batch, or the entire dataset. From here, it is possible to take the gradient flow limit, described by equation 2.20, which requires computing the expectation over the dataset distribution. This is where linear networks become convenient for mathematical analysis, as the expectation of the derivatives remains linear

in the data, resulting in the following set of differential equations:

$$\tau_w \frac{dW_1}{dt} = W_2^T \left(\Sigma_{xy}^T - W_2 W_1 \Sigma_x\right) - \lambda W_1, \tag{2.25}$$

$$\tau_w \frac{dW_2}{dt} = \left(\Sigma_{xy}^T - W_2 W_1 \Sigma_x\right) W_1^T - \lambda W_2, \tag{2.26}$$

subject to the initial conditions $W_1(t=0)$ and $W_2(t=0)$. Here, $\Sigma_{xy}^T = \left\langle XY^T \right\rangle_D$ and $\Sigma_x^T = \left\langle XX^T \right\rangle_D$ are the input-output correlations and input-input correlation respectively, τ_w is a learning time scale for the weights that can be set arbitrarily to convert from iteration k to units of time t, and λ controls the weight regularization (see section A.2 for a complete derivation).

Even though these equations describe a linear network, which can only perform linear transformations from X to Y, **the learning dynamics are governed by a set of nonlinear and coupled differential equations** due to weight multiplication and the presence of weights from both layers in each equation. Because of this, training a neural network with two (or more) layers exhibits complex step-like dynamics that depend on the data structure. These dynamics are also present in more complex learning systems, making linear networks an appropriate surrogate model for mathematically analyzing neural network learning dynamics. These similarities have been studied analytically, as the differential equations can be solved in closed form for the generalization error (but not for the weights $W_i(t)$) in certain cases. In Saxe et al. (2019), the authors further assumed that the inputs are *decorrelated*, i.e., $\Sigma_x = I$, and that the initial weights are small. They then proceeded to solve for the time evolution of the weight product $W_2(t)W_1(t)$ in terms of the singular values of the input-output correlation matrix Σ_{xy} in closed form. This solution explains step-like transitions due to depth and knowledge acquisition time scales, among other phenomena observed experimentally in network training. This theory was further developed by Braun et al. (2022), who found a closed-form solution for the weight product while relaxing the small-weights assumption.

2.1.3.2 Beyond Linear Networks

As the work presented in this thesis focuses on describing learning agents as dynamical systems, the relevant models are those that can be expressed in such a form. These models will then be used for meta-learning in chapter 3 and approximate its optimal control in subsection 4.3.3. To achieve this, having a closed-form solution for the learning dynamics is not necessary; it is sufficient to have the differential equations that describe learning over time. Therefore, other approaches, though not yet explored in this work, could apply the methods proposed in this thesis, extending the capacity of networks to incorporate more layers and, in some cases, perform nonlinear input-output mappings. We briefly discuss these methods here:

Gated Networks: The expectation in the gradient flow limit can be computed in some cases, such as when the transformation induced by the network is linear, as in the case of linear networks discussed earlier. One way to introduce non-linear transformations while preserving the tractability afforded by linearity of the expectation in the gradient flow differential equation is by incorporating the non-linearity into the data itself. This approach is used in Gated Deep Linear Networks, where a different gating pattern is activated for each input. As a result, the network parameters can be factored out of the expectation due to linearity, yielding a differential equation similar to those found in linear networks. The non-linearity arises from the gating pattern activated for each example while maintaining tractable dynamics (Saxe et al., 2022). This framework allows the authors to study modularity and compositionality based on these expressions.

Teacher-Student Setup: In some cases, the expectation on the right-hand side of equation 2.20 can be computed in closed form for non-linear networks. One of the main challenges in doing so is describing the data generation process to then take the expectation, particularly the distribution of (x_i, y_i) pairs in the dataset. A convenient way to model this distribution is through the *teacher-student framework*, where the data

is generated by a neural network, referred to as the *teacher network*, with parameters, e.g., W_T, V_T, such that the dataset is defined by $y = V_T g(W_T x)$. A student network with a similar architecture, characterized by parameters W_S and V_S, then predicts a target $\hat{y}$ such that $\hat{y} = V_S g(W_S x)$, minimizing the error $L = \frac{1}{2}(y - \hat{y})^2$. Assuming $x \sim \mathcal{N}(0, I)$, a two-layer network, like the teacher, can capture complex relationships between x and y while still allowing the expectation for the gradient flow limit to be computed in closed form. This is achieved using certain sigmoidal activation functions (e.g., tanh and ReLU) in terms of the network's *order parameters*, which summarize the system's state and enable closed-form generalization error expressions (Saad and Solla, 1995; Goldt et al., 2019). These methods provide a theoretical foundation for analyzing the impact of task similarities on generalization error, as the teacher network parameterizes the space of tasks through its network parameters (Lee et al., 2022), as well as for studying curriculum learning (Saglietti et al., 2022). It is important to note, however, that the teacher network provides a convenient mathematical formulation of the data generation process rather than supporting the idea that it can successfully describe all complex environments. For a gentle introduction to these methods, see Carrasco-Davis and Grant (2025).

Other Field Theories: The teacher-student setup belongs to a broader family of methods classified as field theories, where complex systems are described through the time evolution of summarized variables called *order parameters*. This is achieved in various ways, primarily by taking the width of networks to infinity, as in the Neural Tangent Kernel approach (Golikov et al., 2022), or through other dynamical mean-field theories (Castellani and Cavagna, 2005; Bordelon and Pehlevan, 2022). These theories are not only applied to neural networks but have also been recently extended to RL agents (Bordelon et al., 2023). A key challenge in applying these methods to the work proposed in this thesis is the mathematical complexity involved in describing their time evolution, making control even more difficult. However, these methods could be used in combination with control theory approaches, as demonstrated in Mori et al. (2025). Nonetheless, they

remain primarily within the domain of numerical simulations, and tractability is lost when attempting to control the learning system.

2.1.4 Prioritized Experience Replay

Reinforcement learning algorithms are notoriously sample inefficient. A widely adopted practice to mitigate this issue is the use of an experience replay buffer, which stores transitions in the form of (s_t, a_t, r_t, s_{t+1}) for later learning (Mnih et al., 2015). Loosely inspired by hippocampal replay to the cortex in mammalian brains (Foster and Wilson, 2006; McNamara et al., 2014), its primary conceptual motivation is to reduce the variance of gradient-based optimization by temporally de-correlating updates, thereby improving sample efficiency. It can also serve to prevent catastrophic forgetting by maintaining transitions from different time scales. The effectiveness of this buffer can often be improved further by *prioritising* some transitions at the point of sampling rather than selecting uniformly. Formally, when transition i is placed into replay, it is given a priority p_i. The probability of sampling this transition during training is given by:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \tag{2.27}$$

where α is a hyper-parameter called *prioritisation exponent* ($\alpha = 0$ corresponds to uniform sampling). Schaul et al. (2016) introduced *prioritized experience replay*, which most often uses the absolute TD-error $|\delta_i|$ of transition i, as $p_i = |\delta_i| + \epsilon$ where a small ϵ constant ensures transitions with zero error still have a chance of being sampled. Another form of prioritization, known as rank-based prioritisation, is to use $p_i = 1/\text{rank}(i)$ where $\text{rank}(i)$ is the rank of the experience in the buffer when ordered by $|\delta_i|$. Sampling transitions non-uniformly from the replay buffer will change the observed distribution of transitions, biasing the solution of value estimates. To correct this bias, the error used for each update is re-weighted by an importance weight of the form $w_i \propto (NP(i))^{-\beta}$, where N is the size

of the buffer and β controls the correction of bias introduced by important sampling ($\beta = 1$ corresponds to a full correction).

The key intuition behind PER is that transitions on which the agent previously made inaccurate predictions should be replayed more often than transitions on which the agent already has low error. While this heuristic is reasonable and has enjoyed empirical success, TD-errors can be insufficiently distinct from the irreducible aleatoric uncertainty; considering instead uncertainty measures more explicitly, this form of prioritisation can be significantly improved.

2.2 Expected Value of Control Theory

In a natural environment, possible actions may incur a cost to be executed. This cost could involve resources such as energy, the difficulty of overriding default or automatic behavior, or simply the opportunity cost of doing something else. In any case, these costs must be considered when making decisions.

The Expected Value of Control (EVC) theory (Shenhav et al., 2013) provides a computational framework in which the cost of taking actions, here understood as control, is explicitly formulated, extending the commonly used RL framework, which may instead discount costs from the reward. It is hypothesized that the quantities necessary for action selection, based on both value and cost, are tracked by the dorsal anterior cingulate cortex (dACC) and lateral prefrontal cortex (LPFC). The following is a summary of key aspects relevant to this thesis.

2.2.1 Computational Framework

Formally, the EVC theory extends the commonly used RL framework described in subsection 2.1.2 by separating the perceived reward into the cost of taking an action and the reward from the environment. Here, we illustrate this theory by decomposing the

perceived reward $\bar{r}$ and deriving the Bellman optimality equation for the action-value function $Q_\pi(s,a)$. The Bellman equation for the action-value function, considering the perceived reward $\bar{r}$ (following the notation introduced in subsection 2.1.2), is:

$$Q_\pi(s,a) = \sum_{s',\bar{r}} p(s',\bar{r}|s,a) \left[\bar{r} + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right]. \tag{2.28}$$

Here, the perceived reward is defined as the actual reward collected from the environment, r, minus the cost of executing an action or control, $C(a)$. Thus, we have $\bar{r} = r - C(a)$. We assume that the cost of each action is deterministic, meaning that the randomness in the perceived reward arises solely from the actual reward r. Consequently, the expectation is taken over r through $p(s', r \mid s, a)$, yielding:

$$Q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) \left[r - C(a) + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right] \tag{2.29}$$

$$= \sum_{s',r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right] - C(a). \tag{2.30}$$

Finally, taking the optimal policy (Bellman optimality equation for Q_*) gives the expected value of control

$$EVC(s,a) = Q_*(s,a) = \sum_{s',r} p(s',r|s,a) \left[r + \gamma \max_{a'} Q_*(s',a')\right] - C(a). \tag{2.31}$$

Then, the optimal control a^* for a given state s is $a^*(s) = \arg\max_a EVC(s,a)$ as in standard Q-learning. This same formulation is converted to a continuous time version and control of dynamical (learning) systems in chapter 3, formally derived in section 3.4.

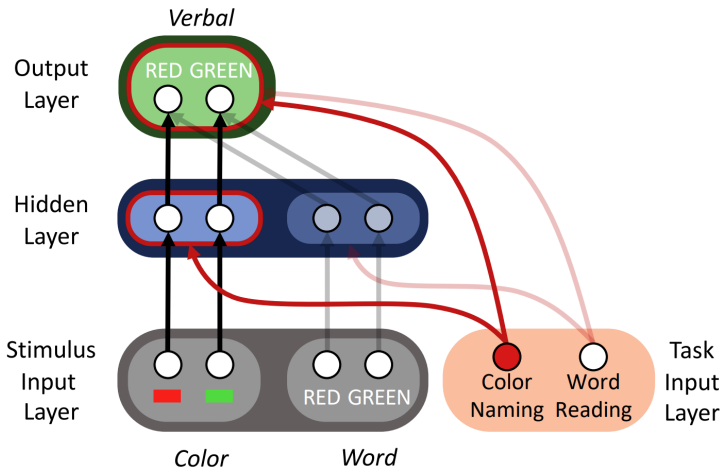

Figure 2.1: Stroop Model, **taken from Musslick et al. (2020)**.

2.2.2 Cognitive Functions and Neural Implementation

The EVC theory provides a normative framework for describing optimal decision-making and control under specific costs. To solve this problem, certain processes can be identified as necessary. These processes are *Regulation*, *Specification*, and *Monitoring*. Each of these processes has been mapped to different brain regions through experimental observations. To illustrate the role of each process, the Stroop task is used as an example in which cognitive control is required.

The Stroop model is a classic setup consisting of a task and an agent, designed to reflect the challenges associated with cognitive control (Cohen et al., 1990). The task involves reporting the cued feature of a given stimulus. In one instance of the task, words with colored fonts are presented as inputs, and the agent must respond by identifying either the written word or the color of the font. When the written word corresponds to a color, human subjects exhibit longer response times when asked to name the font color and shorter response times when asked to read the word. This discrepancy is exacerbated when the written word and the font color are incongruent (e.g., the word "Red" written in blue font, Stroop 1935; MacLeod 1991).

The effects observed in the Stroop task can be reproduced by a neural network model (see figure 2.1). The network's input consists of the word identity and the font color, which are processed through a shared hidden representation, followed by an output layer that specifies the estimated response. One assumption is that the pathway from the stimulus to reading the word is stronger than the pathway required to report the font color, as individuals spend significantly more time reading words than naming colors over their lifespan. As a result, when the model is tasked with naming the font color, the color pathway competes with the stronger word pathway, interfering with the correct response. However, as observed in experiments, humans are capable of naming the colors, implying that they must be able to regulate the strength of these pathways to resolve interference. In the neural network model, this regulation is achieved by modulating the activity of the hidden neurons, amplifying the signal of the relevant pathway based on the cued task (naming the word or the color). This mechanism enables the model to override the default tendency to read the word and instead engage in the controlled behavior of reporting the font color.

Using this model and task, the processes of *Monitoring*, *Specification*, and *Regulation* can be instantiated. Next, each process is defined, and matched to a component of the Stroop model, and existing evidence for its neural implementation in the brain is presented.

Specification: The decision of whether control should be allocated to a task and to what extent, defining both the identity (which tasks) and intensity (how much effort) assigned to each task. The EVC theory attributes this process specifically to the dACC. For example, in the Stroop model, specification involves determining whether any control should be applied to the task, which depends on the cued task (word or color), and how much control is required.

Regulation: This refers to the implemented control signal at a lower level compared to specification. It is similar in that both the identity and intensity need to be defined,

but regulation involves the actual control signal that affects information processing of the system at hand. In the Stroop model, regulation corresponds to the control implementation from the IPFC to the hidden layer, modulating each neural pathway depending on the current task. The regulation from the IPFC depends on the executive function of specification from the dACC.

Monitoring: Specification and regulation depend on how well the task at hand is being performed. In the Stroop model, for example, long reaction times or incorrect responses when cued to report the color could indicate a lack of control. This may signal the need for increased control, which would be conveyed to the dACC for later use in the specification process and subsequent application through regulation.

Jointly: The specification process determines the optimal control signal. The identity and intensity of control are modulated based on monitored task performance while the control signal is implemented through the regulation system. These processes must account for intrinsic features of the information processing system, such as the agent's capacity to solve the task. Examples include speed-accuracy trade-offs (Bogacz et al., 2006), task uncertainty (Yu and Dayan, 2005; Yu et al., 2009), default behavior override (Piray and Daw, 2021), and exploration-exploitation trade-offs (Li et al., 2012), among others (Botvinick et al., 2001; Holmes and Cohen, 2014). For a comprehensive review of EVC theory, its relationship to brain architectures, and supporting experimental evidence, see Shenhav et al. (2013).

2.3 Meta-Learning

As mentioned in the introduction, meta-learning is also referred to as learning to learn and is a fundamental concept in machine learning. In this field, meta-learning techniques are typically employed to optimize components of the learning system that are traditionally hand-selected during model design (also known as meta-parameters). One approach to

achieving this involves formulating the meta-learning problem as a bilevel optimization problem (Franceschi et al., 2018; Hospedales et al., 2020), which separates the standard optimization of a model (inner loop) from the optimization of the meta-parameter (outer loop). This can be formally defined as follows:

Consider the parameters of a learning model, W (e.g., the weights of a neural network), and a meta-parameter g, which could represent hyperparameters such as the learning rate (Franceschi et al., 2018; Baik et al., 2020), initial weights (Finn et al., 2017), a parameterization of learning rules (Metz et al., 2019), a loss function (Bechtle et al., 2021; Raymond et al., 2023), or even the learning curriculum (Stergiadis et al., 2021; Zhang et al., 2022). Typically, the parameters of the learning model are found by minimizing an objective function $\mathcal{L}$:

$$W^* = \arg\min_W \mathcal{L}(W, g, D_{\text{train}}), \tag{2.32}$$

where D_{train} represents the training data distribution. This equation is solved through an optimization process, such as gradient descent, and is referred to as the *inner loop*. Given the solution of the inner loop, W^*, the *outer loop* is defined as

$$g^* = \arg\min_g \mathcal{L}_{\text{meta}}(W^*, g, D_{\text{meta}}), \tag{2.33}$$

where $\mathcal{L}_{\text{meta}}$ is a *meta-objective*, and D_{meta} is the data distribution used to evaluate the meta-loss. The outer loop is dependent on the solution of the inner loop and can take various forms. For example, the inner loop may produce a set of optimal parameters for different tasks, while the outer loop optimizes the meta-parameter to achieve high performance across all tasks, considering the specific solutions obtained for each task.

Not all meta-learning is formulated in this manner, and multiple perspectives exist on how to conceptualize it. For instance, the term meta-learning is also applied in cases

where solutions emerge from models trained on large distributions of tasks. If an agent is capable of discovering general strategies across a distribution of tasks, thereby enabling the resolution of each task individually, the agent is said to have meta-learned the task distribution (see subsection 3.3.2 for a detailed discussion on emergent meta-learning). For a comprehensive review of meta-learning methods, see (Hospedales et al., 2020; Vettoruzzo et al., 2024).

2.3.1 Model Agnostic Meta-Learning

A specific algorithm closely related to the proposed method presented in this thesis is Model-Agnostic Meta-Learning (MAML, Finn et al. 2017). MAML can be understood as a bilevel optimization problem, where the meta-parameters to be optimized are the initial parameters of the inner loop. These initial parameters are learned such that, when the model is presented with a new task, adaptation occurs rapidly.

Consider a model parameterized by a function $f(W)$, trained on a task $\mathcal{T}_i$ from a family of tasks $\mathcal{T}$. The training process is typically performed using gradient updates, defined as follows:

$$W' = W - \alpha \frac{d\mathcal{L}\left(f(W, \mathcal{T}_i)\right)}{dW}, \tag{2.34}$$

which constitutes the *inner loop*. To find the optimal initial parameters that enhance adaptation across tasks in $\mathcal{T}$, the meta-objective (solved in the *outer loop*) is defined as:

$$\min_W \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}\left(f(W', \mathcal{T}_i)\right) = \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}\left(f\left(W - \alpha \frac{d\mathcal{L}\left(f(W, \mathcal{T}_i)\right)}{dW}, \mathcal{T}_i\right)\right). \tag{2.35}$$

In other words, the goal is to find the initial parameters W that minimize the loss after one update on each of the available tasks. This results in initial conditions W that enable rapid improvement of task-specific losses with subsequent updates. In practice,

this meta-parameter is optimized in the outer loop using gradient descent updates:

$$W^* = W - \beta \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}\left(f(W', \mathcal{T}_i)\right). \tag{2.36}$$

MAML is one of the most widely used meta-learning algorithms in machine learning, as it can be generalized to probabilistic settings (Yoon et al., 2018), has convergence guarantees (Nichol et al., 2018; Fallah et al., 2020), and has been applied in various domains (Griva et al., 2023).

2.4 Uncertainty Estimation

Uncertainty is a fundamental concept in statistics. Within machine learning, it has predominately been studied in supervised learning, particularly with Bayesian methods (Lahlou et al., 2022; Narimatsu et al., 2023). Various aspects of the task setting such as bootstrapping and non-stationarity make uncertainty estimation a significantly more challenging problem in RL compared to neural networks; nevertheless, it has featured more prominently in recent work, including for use in generalization (Jiang et al., 2023), as reward bonuses in exploration (Nikolov et al., 2019), and to guide safe actions (Lütjens et al., 2019; Kahn et al., 2017). This section presents key concepts related to uncertainty that are relevant to this work, particularly those concerning the distinction between aleatoric and epistemic uncertainty.

2.4.1 Bootstrapped DQN

The concept behind bootstrapping is to approximate a posterior distribution by sampling a prediction from an ensemble of estimators, where each estimator is initialized randomly and observes a distinct subset of the data (Tibshirani, 1994; Bickel and Freedman, 1981). In RL, Osband et al. (2016) introduced a protocol known as *bootstrapped DQN* for

deep exploration, whereby bootstrapping is used to approximate the posterior of the action-value function, from which samples can be drawn. Each agent within an effective ensemble, parameterized by ψ, is randomly initialized and trained using a different subset of experiences via random masking. A sample estimate of the posterior distribution, denoted as $\psi \sim P(\psi|D)$ (D being training data), is obtained by randomly selecting one of the agents from the ensemble. This work utilizes extensions of the bootstrapped DQN approach for epistemic uncertainty measurements, particularly ensemble disagreement.

2.4.2 Distributional RL

Learning quantities beyond the mean return has been a long-standing programme of RL research, with particular focus on the return variance (Sobel, 1982). A yet richer representation of the return is sought by more recent methods known collectively as *distributional RL* (Bellemare et al., 2023), which aims to learn not just the mean and variance, but the entire return distribution. This section examines a specific class of distributional RL methods: those that model the quantiles of the distribution, specifically QR-DQN (Dabney et al., 2017). A more comprehensive discussion of the distributional RL literature can be found in Bellemare et al. (2023).

In QR-DQN, the distribution of returns, for example from taking action a in state s and subsequently following policy π, $\eta^\pi(s,a)$ is approximated as a *quantile representation* (Bellemare et al., 2023), that is, as a uniform mixture of Diracs, and trained through *quantile regression* (Koenker and Hallock, 2001). For such a distribution, $\hat{\nu} = \frac{1}{m}\sum_{i=1}^{m} \delta_{\theta_{\tau_i}}$, with learnable quantile values θ_{τ_i} and corresponding quantile targets $\tau_i = \frac{2i-1}{2m}$, the quantile regression loss for target distribution ν is given by

$$\mathcal{L}_{\text{QR}} = \sum_{i=1}^{m} \mathbb{E}_{Z\sim\nu}[\rho_{\tau_i}(Z - \theta_{\tau_i})], \tag{2.37}$$

where $\rho_\tau(u) = u(\tau - \mathbb{1}_{u<0})$ and $\mathbb{1}$ is the indicator function. By leveraging the so-called

distributional Bellman operator and the standard apparatus of a DQN model, QR-DQN prescribes a temporal difference deep learning method for minimising the above loss function and learning an approximate return distribution function via quantile regression.

Distributional RL in itself does not (so far) permit a natural decomposition of uncertainties into epistemic and aleatoric (Clements et al., 2020; Chua et al., 2018; Charpentier et al., 2022); rather the variance of the learned distribution will converge on what can reasonably be thought of as the aleatoric uncertainty. In subsection 5.2.1, previous techniques that combine distributions with ensembles are extended to construct estimates of both epistemic and aleatoric uncertainties. Both approaches for characterizing epistemic uncertainty can be understood within an excess risk framework, which is outlined below.

2.4.3 Direct Epistemic Uncertainty Prediction

A clear and formal representation of uncertainty is used to describe uncertainty, where total uncertainty is defined as the sum of epistemic and aleatoric components, with epistemic uncertainty interpreted as excess risk. This notion was introduced by Xu and Raginsky (2022) and later extended by Lahlou et al. (2022); their framing is adapted to the present setting. Consider the **total uncertainty** $\mathcal{U}(s,a)$ of an action-value predictor $Q_\psi(s,a)$, for a given state s and action a as:

$$\mathcal{U}(Q_\psi, s, a) = \int \left(\Theta(s', r) - Q_\psi(s,a)\right)^2 P(s', r|s, a) ds' dr, \tag{2.38}$$

where $\Theta(s', r)$ is the Q-learning target as in equation 2.15. Then, the **aleatoric uncertainty** $\mathcal{A}(s,a)$, is given by the total uncertainty (as defined above) of a Bayes-optimal predictor Q^*_ψ (see Lahlou et al. (2022)):

$$\mathcal{A}(s,a) = \mathcal{U}(Q^*_\psi, s, a). \tag{2.39}$$

Note that this quantity is independent of any learned predictor and is a function of the data only. The **epistemic uncertainty** $\mathcal{E}(Q_\psi, s, a)$, which is computed for a given predictor, is defined as the total uncertainty of the predictor minus the aleatoric uncertainty:

$$\mathcal{E}(Q_\psi, s, a) = \mathcal{U}(Q_\psi, s, a) - \mathcal{A}(s, a), \tag{2.40}$$

where $\mathcal{E}(Q_\psi, s, a)$ is the squared distance between the true mean and estimate mean. The proof goes as follows:

$$\begin{aligned}
\mathcal{U}(Q_\psi, s, a) &= \int \left(\Theta(s', r) - Q_\psi(s, a)\right)^2 P(s', r|s, a) ds' dr && (2.41)\\
&= \mathbb{E}_{s',r}\left[\left(\Theta(s', r) - Q_\psi(s, a)\right)^2\right] && (2.42)\\
&= \mathbb{E}_{s',r}\left[\Theta(s', r)^2\right] - 2Q_\psi(s, a)\mathbb{E}_{s',r}\left[\Theta(s', r)\right] + Q_\psi(s, a)^2 && (2.43)\\
&= \mathbb{V}_{s',r}\left[\Theta(s', r)\right] + \mathbb{E}_{s',r}\left[\Theta(s', r)\right]^2 - 2Q_\psi(s, a)\mathbb{E}_{s',r}\left[\Theta(s', r)\right] + Q_\psi(s, a)^2 && (2.44)\\
&= \underbrace{\mathbb{V}_{s',r}\left[\Theta(s', r)\right]}_{\text{aleatoric } \mathcal{A}(s,a)} + \underbrace{\left(Q_\psi(s, a) - \mathbb{E}_{s',r}\left[\Theta(s', r)\right]\right)^2}_{\text{epistemic } \mathcal{E}(Q_\psi,s,a)} && (2.45)
\end{aligned}$$

Concretely, this decomposition can be useful in instances where you want to estimate epistemic uncertainty, but doing so directly is significantly more difficult than estimating total and aleatoric uncertainty, which is often the case. In subsection 5.2.1, a method is provided to estimate quantities in this manner, which is later used to prioritize transitions in the replay buffer.

2.4.4 Ensembles of Distributions

Using an ensemble of distributional RL agents gives us a concrete prescription for computing epistemic uncertainty as well as aleatoric uncertainty. This approach was first

formalised by Clements et al. (2020), who define learned aleatoric and epistemic uncertainty quantities as a decomposition of the variance of the estimation from the ensemble (here defined as total uncertainty $\hat{\mathcal{U}}$) of distributional RL agents:

$$\hat{\mathcal{U}}(s,a) = \mathbb{V}_{\tau,\psi}\left[\theta_\tau(s,a;\psi)\right] = \hat{\mathcal{E}}(s,a) + \hat{\mathcal{A}}(s,a) \tag{2.46}$$

where

$$\hat{\mathcal{A}}(s,a) = \mathbb{V}_\tau[\mathbb{E}_\psi(\theta_\tau(s,a;\psi))], \quad \hat{\mathcal{E}}(s,a) = \mathbb{E}_\tau[\mathbb{V}_\psi(\theta_\tau(s,a;\psi))], \tag{2.47}$$

and s, a are state and action, $\psi \sim P(\psi|D)$ are the model parameters of each agent in the ensemble, D denotes the data distribution, and θ_τ is the value of the τ^{th} quantile. $\mathbb{V}$ and $\mathbb{E}$ are variance and expectation operators respectively. Intuitively, $\hat{\mathcal{E}}$ measures epistemic uncertainty as the expected disagreement (variance) in quantile estimations across the ensemble, while $\hat{\mathcal{A}}$ takes the average estimation across the ensemble for each quantile of the distribution, and computes the variance of this averaged distribution. Clements et al. (2020) stop short of using a *bona fide* ensemble to estimate these quantities, opting instead for a two-sample approximation in the agent they present. However Jiang et al. (2023) go on to use ensemble methods more explicitly, as we do in this work.

2.4.5 Other methods

Direct Variance Estimation

Distributional RL provides a framework for computing statistics of the return beyond the mean. Efforts to compute such quantities in RL date back to Sobel (1982), who derived Bellman-like operators for higher order moments of the return in MDPs that can be used to indirectly estimate variance. This has since been extended to a greater set of

problem settings and models (Prashanth and Ghavamzadeh, 2016; Tamar et al., 2016; White and White, 2016). More recently methods have also been developed to directly estimate variance (Tamar et al., 2012); arguably the simplest such scheme for TD(0) learning is the following update rule for the action-value variance $\hat{\mathcal{A}}(s,a)$ at state s,a (re-estated from Sherstan et al. (2018) for state and action):

$$\hat{\mathcal{A}}_{t+1}(s,a) \leftarrow \hat{\mathcal{A}}_t(s,a) + \bar{\alpha}\bar{\delta}_t, \tag{2.48}$$

where

$$\bar{\delta}_t \leftarrow \bar{r}_{t+1} + \bar{\gamma}_{t+1}\hat{\mathcal{A}}_t(s',a') - \hat{\mathcal{A}}_t(s,a), \tag{2.49}$$

$$\bar{r}_{t+1} \leftarrow \delta_t^2, \tag{2.50}$$

$$\bar{\gamma}_{t+1} \leftarrow \gamma_{t+1}^2; \tag{2.51}$$

δ_t is the temporal difference error of on the mean value estimate, and $\bar{\alpha}$ is the variance learning rate. $\bar{r}$ can be thought of as a 'meta' reward for the variance estimate. This update corresponds to simply regressing on the square of the mean estimate error in a standard regression problem (single state, no concept of discounting) like in the bandit experiments shown in section 5.3.

Bayesian methods

A more comphrehensive Bayesian approach to the reinforcement learning problem can be formulated via so-called Bayes-adaptive Markov decision processes (BAMDPs) (White, 1969), where an agent continuously updates a belief distribution over underlying Markov decision processes. Solutions to BAMDPs are Bayes' optimal in the sense that they optimally trade off exploration and exploitation to maximise expected return. However, in all but the smallest environments and settings, learning over this entire belief distribution is intractable (Brunskill, 2012; Asmuth et al., 2012).

Posterior sampling, which can be viewed as the analogue of Thompson sampling for MDPs, has been a popular method to approximate the full Bayesian posterior e.g. via ensembles (Osband et al., 2016) or dropout (Gal and Ghahramani, 2016); extensions include provision of pseudo priors (Osband et al., 2018, 2023). While these approaches have been successful in some settings, they have few guarantees. A different line of work includes using methods such as meta-learning to reason on and train the approximate posterior (Humplik et al., 2019; Zintgraf et al., 2020).

With regards to the discussions on epistemic and aleatoric uncertainty, the above methods can give the model access to a distribution over parameters that can be sampled and operated on (e.g. to calculate variance). They do not however, Bayes optimal or not, lead *per se* to a decomposition into epistemic and aleatoric uncertainty.

State Counts

Another category of methods that are frequently used in reinforcement learning and related paradigms like bandits is based around notions of counts e.g. of state visitation. Such counts can be used to construct intervals/bounds on confidence of learned quantities. This is the foundation of well established exploration methods in tabular settings called upper confidence bounds (Auer, 2002). In function approximation settings, much of the focus has been on constructing accurate *pseudo* counts that incorporate state similarities (Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017). Despite the well demarcated distinction between count-based methods and those that address the Bayesian posterior above, with access to any mean-zero unit-variance distribution, an ensemble of mean-predictors of that distribution can be used to estimate pseudo-counts (Lobel et al., 2023). As a result, it is generally possible to convert a Bayesian posterior into pseudo-counts.

Model-Based Estimation

A set of methods that is further removed from those used in our work, but are often motivated by similar questions consists of learning a model of the environment. Down-

stream quantities like the prediction error of the environment model can be used as proxies for uncertainty or novelty e.g. for exploration bonuses. Much of this work falls under the domain of intrinsic motivation (Barto, 2013). Some of the methods in this area e.g. curiosity (Pathak et al., 2017) attempt implicitly to make the distinction between epistemic uncertainty and aleatoric uncertainty to avoid the noisy TV problem.

Beyond the prioritisation variable

Altering the prioritized experience replay is not confined to changing the prioritization variable. In Zha et al. (2019), the replay policy is adapted through gradient optimization. Balaji et al. (2020) introduces a regularization technique, enhancing continual learning by storing a compressed network activity version for replay. Additional methods encompass the utilization of sub-buffers storing transitions at multiple time scales (Kaplanis et al., 2020), replay for sparse rewards (Andrychowicz et al., 2017; Nair et al., 2018), and employing diverse sampling strategies (Pan et al., 2022). Further endeavors are aiming to understand the effects of PER in RL (Liu and Zou, 2017; Fedus et al., 2020).

Chapter 3

Meta-Learning Strategies through Value Maximization

The work presented in this chapter is primarily based on *Meta-Learning Strategies through Value Maximization in Neural Networks*, posted in October 2023. Available at `http://arxiv.org/abs/2310.19919` (arXiv:2310.19919 [cs, q-bio]) by Rodrigo Carrasco-Davis, Javier Masís, and Andrew M. Saxe. The text has been modified to conform to the thesis format and includes additional results in section 3.7.

3.1 Introduction

Deploying a learning system requires making many considered decisions about hyperparameters, architectures, and dataset properties. As learning systems have grown more complex, so have these decisions about how to learn. One approach to managing this complexity is to place these decisions under the control of the agent and meta-learn them. Building on this strategy, a range of meta-learning algorithms have been developed that are capable of fast adaptation to new tasks within a distribution (Finn et al., 2017; Nichol et al., 2018),

continual learning (Parisi et al., 2019), and multitasking (Crawshaw, 2020). Meta-learning methods target diverse aspects of a learning system: they can adapt hyperparameters (Franceschi et al., 2018; Baik et al., 2020; Zucchet and Sacramento, 2022); learn weight initializations well-suited to a task distribution (Finn et al., 2017; Baik et al., 2020); manage different modules or architectural components (Andreas et al., 2017); enhance exploration (Gupta et al., 2018; Liu et al., 2021); and order tasks into a suitable curriculum (Stergiadis et al., 2021; Zhang et al., 2022). While this prior work has shown that meta-learning can bring important performance benefits, algorithms are often hand-designed for a specific intervention and a large gap remains in our theoretical understanding of how meta-learning operates (see section A.1 for an extended discussion on related work).

The aim of this chapter is to develop a normative framework for investigating optimal meta-strategies in neural networks, implemented in biological and artificial agents. A core difficulty in computing optimal strategies is the complexity of optimizing through the learning process. To tackle this problem, the inner-loop learning dynamics are simplified using simpler tractable network models. Specifically, meta-learning dynamics in deep linear networks are studied, as these exhibit complex non-linear dynamics that share properties with observed non-linear network dynamics (Saxe et al., 2019; Braun et al., 2022). By examining this problem in a reduced setting, optimal meta-learning strategies are derived under various control designs and meta-learning scenarios. The focus is on questions pertinent to the cognitive control literature, such as learning effort allocation, task switching, and attention to multiple tasks. The Expected Value of Control Theory (EVC Shenhav et al. (2013, 2017); Musslick et al. (2020); Masís et al. (2021)) has proposed answers to these questions. It posits that higher-level areas in the brain perform executive functions (cognitive control) over lower-level areas to maximize the cumulative return. The framework presented serves as a formal and computationally tractable example of the EVC theory, taking into account the impact of the control signal on future learning dynamics.

Main contributions

• A computationally tractable *learning effort* framework[1] is developed to study diverse and complex meta-learning interventions that normatively maximize value throughout learning.

• Learning dynamics are fully solved as a function of control variables for simple models, and this solution is used to derive efficient optimization procedures that maximize discounted performance throughout learning.

• Meta-learning algorithms such as Model Agnostic Meta-Learning (Finn et al., 2017) and Bilevel Programming (Franceschi et al., 2018) are expressed within this framework, allowing for the study of the impact of approximations on their performance.

• Optimized control strategies are computed for a range of settings spanning continual learning, multi-tasking, and curriculum learning, then these normative strategies are examined.

• Due to the framework's normative goal of maximizing expected return, qualitative connections are drawn to phenomena in cognitive neuroscience such as task engagement, mental effort, and cognitive control (Shenhav et al., 2013, 2017; Lieder et al., 2018; Masís et al., 2021). More specifically, behavioral observation in rats are qualitatively reproduced (Masis et al., 2024).

3.2 Learning Effort Framework

The framework is first defined in a general manner before introducing a simple example in subsection 3.2.1. The generality of this description allows the framework to be applicable to a variety of different settings of interest, spanning machine learning (section 3.3) and

[1]Python package at `https://github.com/rodrigcd/neuromod` for reproducibility.

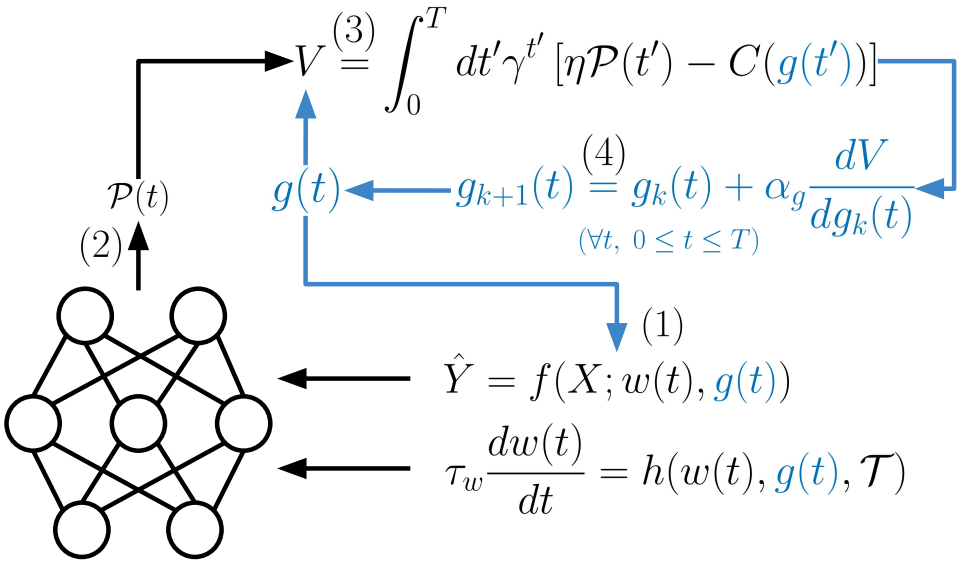

Figure 3.1: Learning effort framework. A neural network is under the influence of a control signal $g(t)$. This control signal is optimized iteratively by initializing $g(t)$, then: (1) Solving learning dynamics in equation 3.1; (2) Computing the performance $\mathcal{P}(t)$; (3) Integrating performance and control cost to compute the exact cumulative return V in equation 3.3; (4) Taking the gradient of V with respect to the control signal $g(t)$ and update as in equation 3.4, then go back to (1).

cognitive control (section 3.5, section 3.6 and section 3.7).

Consider a learning model trained on a task $\mathcal{T}$ for a period of time T. Two equations define the learning model. The input-output mapping f and learning dynamics h are defined as

$$\hat{Y} = f(X; w(t), g(t)), \quad \tau_w \frac{dw(t)}{dt} = h(w(t), g(t), \mathcal{T}) \tag{3.1}$$

respectively. In the first equation, f is a continuously differentiable function, X represents the input, and $\hat{Y}$ the output. Here, $w(t)$ denotes the parameters of the learning model (e.g., weights in a neural network) during training, with $T \geq t \geq 0$. The term $g(t)$ is introduced as an *effort signal* (or *control signal*), which is chosen by the meta-learning optimization. This vector of control signals can model a number of interventions in the learning system and is selected to maximize cumulative learning performance. The

learning dynamics equation describes the evolution of the parameters during training and is given by a differential equation over the parameters of the learning model. The function h is a continuously differentiable function, and the evolution of the learned parameters $w(t)$ (starting at $w(0)$) may depend on the control signal $g(t)$ and task parameters $\mathcal{T}$.

Given this setup, the control signal $g(t)$ can be understood as a meta parameter that can be chosen in different ways and influences the network's input-output map and learning behavior. In the context of cognitive neuroscience literature, this could take the form of controlled attention or neural activity modulation. To determine the selection of $g(t)$, a task performance metric is defined during the learning period, $\mathcal{P}(t)$ (e.g., mean squared error during regression). Furthermore, it is assumed that using the control signal $g(t)$ incurs a cost, represented by a cost function $C(g(t))$. This formulation is commonly used in control theory to describe factors such as the energy resources required to exert control or the mental effort allocated to sustain engagement in a task. At any time during the learning of the task $\mathcal{T}$, an instant reward rate is considered, defined as $R(t) = \eta\mathcal{P}(t)$, where η is a constant that converts task performance $\mathcal{P}(t)$ into reward per unit time. The instant net reward rate is then defined as the difference between scaled performance and the cost of control:

$$v(t) = R(t) - C(g(t)). \tag{3.2}$$

The expected return or value function at the start of training can then be written as the cumulative discounted reward from learning and performing the task from time $t = 0$ to $t = T$, with a discount factor $1 \geq \gamma > 0$,

$$V = \int_0^T dt\gamma^t v(t) = \int_0^T dt\gamma^t \left[\eta\mathcal{P}(t) - C(g(t))\right]. \tag{3.3}$$

This value function measures performance across the entire learning period. Finally, it is posited that the objective of meta-learning is to select $g(t)$ to maximize the value function

in equation 3.3. To approximate an optimal $g(t)$, gradient steps are taken as

$$g_{k+1}(t) = g_k(t) + \alpha_g \frac{dV}{dg(t)}, \tag{3.4}$$

for every $0 \geq t \geq T$, where k denotes the iteration index. The optimal $g(t)$ thus depends on a complex interplay of past and future values of the control signal and their interaction with the entire trajectory of learning. Computing the gradient in equation 3.4 is, in general, computationally intractable. In the remainder of this paper, learning models and settings are carefully selected to exhibit rich dynamics while maintaining partial analytical tractability of the learning dynamics, enabling efficient computation of the full control signal over time. Further details on the implemented algorithm and estimation of involved quantities can be found in Algorithm 1 and section 4.1.

By appropriate choice of how $g(t)$ influences the network and learning dynamics, this general framework can accommodate a variety of possible interventions on a learning system. Some interventions correspond to other meta-learning algorithms such as Multi-Step MAML and Bilevel Programming (section 3.3), also providing a formal mathematical connection (subsection 3.3.1). The results in subsequent sections investigate several scenarios, where the experiments are variations on the influence of the control signal over the learning dynamics, keeping the rest of the framework as is.

3.2.1 Single Neuron Example

After describing the general framework, attention is now turned to a simple case to illustrate it, while still exhibiting complex emergent solutions. A *single neuron learning model* trained on a *two-Gaussians regression task* is considered, where the control signal functions as a *weight gain modulation*. This case provides insights into the dependence of the optimal control signal on task parameters and learning model hyperparameters.

Two Gaussians regression task: A dataset of examples $i = 1, \cdots, P$ is drawn as

Algorithm 1 Learning Effort Optimization

Input: A learning system (a input-output mapping equation, and a learning dynamics equation as in equation 3.1), a task $\mathcal{T}$, learning period T, initialize $g_0(t_i)$ for every $i = 0, ..., N$ (with $t_0 = 0$ and $t_N = T$), reward conversion η, control cost $C(g(t))$, parameters $w(t_0)$, number of gradient updates on the control signal N_k, control learning rate α_g.

for $k = 0$ **to** N_k **do**
  set $V = 0$
  for $i = 0$ **to** N **do**
    Compute $w(t_i)$ for every i using the parameters updates in equation 4.4.
    Compute $Y_i = f(X; w(t_i), g_k(t_i))$
    Compute $\mathcal{P}(t_i)$, and $R(t_i) = \eta\mathcal{P}(t_i)$ (e.g $\mathcal{P}(t_i) = -\langle\mathcal{L}(t_i)\rangle_{XY}$)
    Compute $v(t_i) = R(t_i) - C(g_k(t_i))$
    $V \leftarrow V + v(t_i) \cdot \delta t$
  end for
  for $i = 0$ **to** N **do**
    $g_{k+1}(t_i) = g_k(t_i) + \alpha_g \frac{dV}{dg_k(t_i)}$
  end for
end for

Output: Optimized control signal g_{N_k}.

follows: A label y_i is first sampled as either $+1$ or -1 with probability $1/2$. The input x_i is then sampled from a Gaussian $x_i \sim \mathcal{N}(y_i \cdot \mu_x, \sigma_x^2)$. The task is to predict y_i from the value of x_i. The intrinsic difficulty of the task is controlled by how much the Gaussians overlap, controlled by the relative value of μ_x and σ_x.

Single neuron learning model: The input-output mapping of the single neuron model is $\hat{y}_i = x_i \cdot w(t)\,[1 + g(t)]$, $w(t)$ is the learned weight parameter, and $g(t)$ is the control signal which acts as a multiplicative gain. The learning dynamics of $w(t)$ are given by gradient descent on the loss function $\mathcal{L} = \frac{1}{2}(y_i - \hat{y}_i)^2 + \frac{\lambda}{2}w(t)^2$. Taking the gradient flow limit (small learning rate as in Saxe et al. 2019; Elkabetz and Cohen 2021), the average learning dynamics for the weight is described by

$$\tau_w \frac{dw}{dt} = -\left\langle \frac{\partial \mathcal{L}}{\partial w} \right\rangle = \mu_x \tilde{g}(t) - w(t)\left(\left\langle x^2 \right\rangle \tilde{g}^2(t) + \lambda\right) \tag{3.5}$$

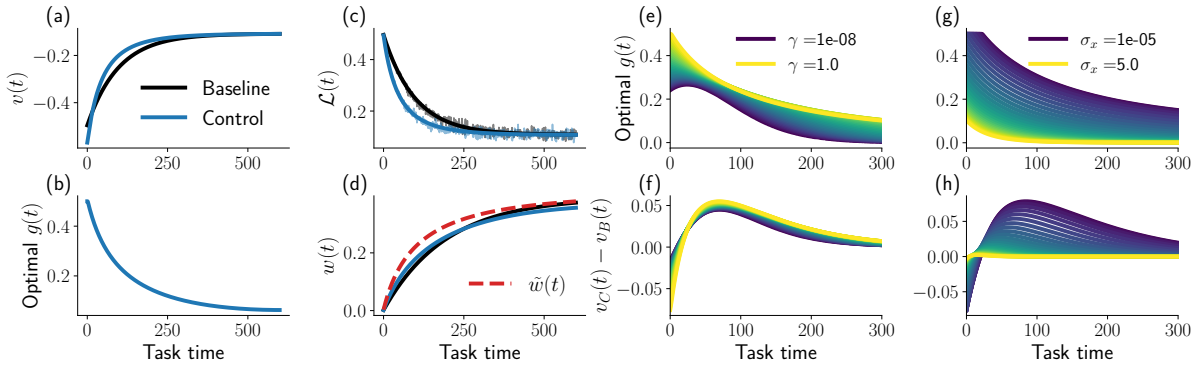

Figure 3.2: Results in single neuron model throughout the learning period $0 \geq t \geq T$. **(a)** Instant net reward $v(t)$. **(b)** Loss $\langle \mathcal{L}(t) \rangle$ for theoretical predictions (solid) and simulations using SGD (shaded). **(c)** Optimal control signal decreases through learning (Baseline $g(t) = 0$). **(d)** Weight $w(t)$ through learning for control and baseline case, $\tilde{w}(t) = w(t) \cdot (1 + g(t))$. Dependence of optimal control signal on task parameters. **(e)** and **(g)**: optimal $g(t)$ when varying discount factor γ and noise level σ_x respectively. **(f)** and **(h)**: Difference between instant net rewards $v(t)$ between control and baseline when varying γ and σ_x respectively. Longer time horizons and less noisy tasks recruit more control.

where $\tilde{g}(t) = 1 + g(t)$, $\langle \cdot \rangle$ denotes expectation over the data distribution, and τ_w is the learning time scale of the weight. This gradient depends on $g(t)$, making the learning dynamics of $w(t)$ dependent on the control signal. This tractability allows us to compute average dynamics and the necessary gradient efficiently.

Control signal optimization: The performance and control measure for this model were defined as $\mathcal{P}(t) = -\langle \mathcal{L}(t) \rangle$, $C(g(t)) = \beta g(t)^2$, meaning smaller loss leads to better performance and exerting control has a cost that is monotonic in the control signal magnitude, with cost per unit of control β. Note that if $g(t) = 0$ for all $T \geq t \geq 0$, then $C(g(t)) = 0$, and $\hat{y}_i = x_i \cdot w(t)$, which means that the weight is learned purely by gradient descent with no influence from the control signal, this is called the *Baseline* model. Having $\mathcal{P}(t)$ and $C(g(t))$, the value function in equation 3.3 is computed to find the optimal $g(t)$ by gradient ascent following equation 3.4 (Algorithm 1). In essence, this setting considers a simple learning scenario in which an agent can adjust the gain of the weights in a neural network that otherwise learns via gradient descent.

Results: In figure 3.2a, the difference in instant net reward $v(t)$ is shown for the baseline case ($g(t) = 0$ for every t) and the control case (optimizing $g(t)$). The optimal meta-learning strategy that maximizes expected return in equation 3.3 allocates more control at the beginning of the learning period (figure 3.2b) at the cost of some instant reward, resulting in faster learning. This is demonstrated by the lower loss observed in the control case in figure 3.2c. The control signal $g(t)$ influences the instant net reward rate both at the present time t and at future times $t' > t$. The instant change in net reward rate $v(t)$ is driven by both the immediate effect on the effective weight $\tilde{w}(t) = w(t) \cdot (1+g(t))$ (figure 3.2d) and the cost function $C(g(t))$, making the effective weight $\tilde{w}(t)$ closer to the solution at early stages.

As expected, increasing the discount factor γ leads to higher levels of control, since future net reward will contribute more to the cumulative expected return, compensating the cost of increasing $g(t)$ (figure 3.2e,f). Increasing the intrinsic noise of the task σ_x reduces the overall optimal control (figure 3.2g,h). Because it is not possible to overcome this noise, the use of control will generate a cost that cannot be compensated by boosting learning. This inter-temporal choice of allocating effort based on the prospect of future reward has been widely studied in psychology and neuroscience (Masís et al., 2021; Keidel et al., 2021; Frömer et al., 2021; Masís et al., 2023) (section A.1), and it is specifically applied to the experiment presented in (Masís et al., 2023) in section 3.7, and naturally arises from maximizing the discounted cumulative performance in equation 3.3. For more parameter variations see subsection A.9.1.

3.3 Instantiating Meta-Learning

The normative objective in equation 3.3 and its maximization through gradient steps on the control signal $g(t)$ can describe various meta-learning algorithms. This section highlights connections to two well-established approaches: Model-Agnostic Meta-Learning (MAML; Finn et al. 2017) and Bilevel Programming (Franceschi et al., 2018).

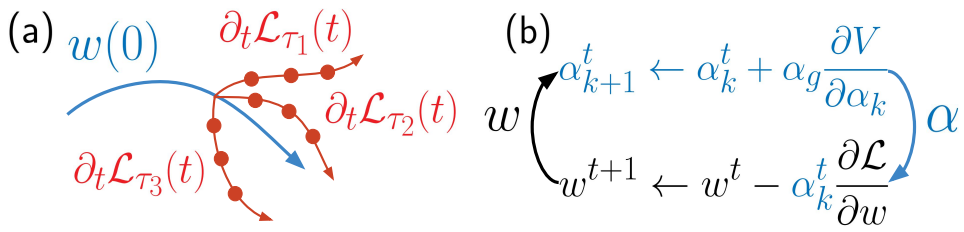

Figure 3.3: **(a)**: Multi-step MAML. **(b)**: Learning rate optimization as in Bilevel Programming.

MAML is a specific instance of this framework in which the initial weights $W_1(0)$ and $W_2(0)$ in the deep linear network serve as the control signal $g(t)$. Defining performance as the average loss per task indexed by τ, $\mathcal{P}(t) = \sum_\tau \langle \mathcal{L}_\tau(t) \rangle$, leads to a meta-objective in MAML when considering only a single step ahead in the value function, i.e., $V_{\text{MAML}} = \mathcal{P}(\delta t)$, where δt represents the time after one gradient update on $g(t)$ (see subsection 3.3.1). This framework also supports optimization over multiple gradient steps, thereby enabling a computationally tractable version of Multi-Step MAML by simplifying the neural network model (figure 3.1b).

To simulate Multi-Step MAML, a two-layer linear network (as in section A.2) was trained on five binary regression tasks, each involving different pairs of digits from MNIST (subsection A.9.2). Results in figure 3.4a,b indicate that the standard MAML loss, V_{MAML}, varies depending on the number of steps considered during the optimization of initial weights. Specifically, V_{MAML} decreases when a few steps ahead are included, enhancing the ability to optimize the learning dynamics. However, beyond a certain number of steps, V_{MAML} begins to increase, suggesting a trade-off where immediate performance is sacrificed to optimize longer-term dynamics, as depicted in figure 3.4b. These multi-step results are made possible by the tractability of this setting. Notably, one-step MAML can substantially underperform compared to Multi-Step MAML.

Additionally, hyperparameters of the network were optimized throughout training. Bilevel Programming provides a means to compute this, with the primary distinction being

the reverse-hypergradient method used to update the meta-parameters (control signal) (Franceschi et al., 2017, 2018). This framework provides a similar application of hyperparameter optimization using gradient descent algorithm in equation 3.4, also incorporating features with intuitive interpretations within a normative framework, such as the discount factor γ and control cost C (subsection A.9.4). The learning rate was optimized over time to maximize cumulative reward in equation 3.3, while γ and β were varied, similar to the single-neuron example, to illustrate their normative significance in hyperparameter optimization (figure 3.1c). The results revealed qualitatively similar behavior to the single-neuron model: longer time horizons and lower costs for increasing the learning rate led to greater control recruitment.

This work provides additional utility to meta-learning algorithms by interpreting them within a normative value-based framework. However, it is important to note that a subset of meta-learning phenomena falls outside the explicit scope of this framework. Specifically, emergent meta-learning agents, in which no outer loop or explicit meta-variable is optimized, are not directly described here (Wang et al. 2017). This topic is further discussed in subsection 3.3.2. For further results in meta-learning simulations for MAML and Bilevel-Programming see subsection A.9.2.

3.3.1 Formal Connection with MAML

The formal description of MAML is as follows. Consider a set of tasks $\mathcal{T}_i$ and a function f_θ parametrized by θ. Each task $\mathcal{T}_i$ has an associated loss function, denoted as $\mathcal{L}_{\tau_i}(f_\theta)$. The parameters after a single gradient step on the loss for a specific task are given by

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}(f_\theta), \tag{3.6}$$

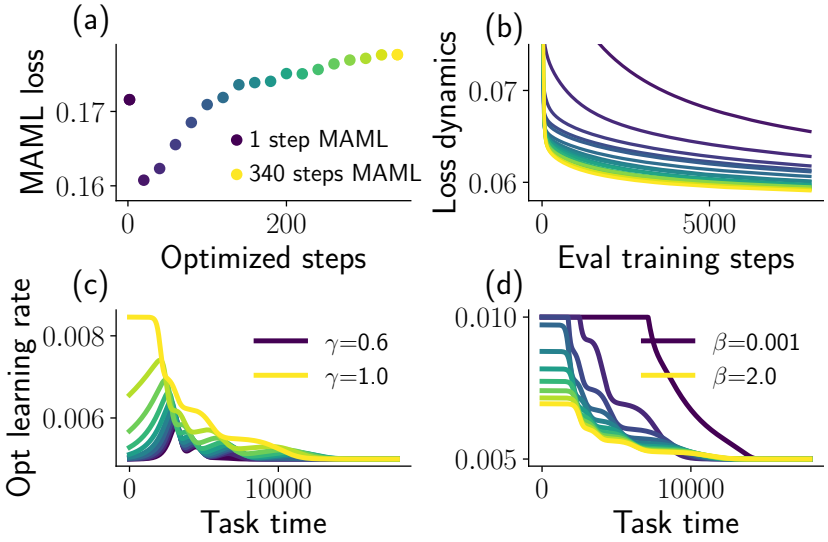

Figure 3.4: **(a)**: Single step MAML loss $V = \mathcal{P}(\delta t)$ when considering more steps in the learning dynamics. **(b)**: Resulting learning dynamics from initial parameters found with Multi-Step MAML. **(c) and (d)**: Optimal learning rate when varying discount factor γ and cost coefficient β.

where α is the learning rate of the inner loop. The meta-objective is then formulated as

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i} \left(f_{\theta_i'} \right) = \min_{\theta} \mathcal{L}_{\text{MAML}}, \tag{3.7}$$

which is minimized via stochastic gradient descent on the model parameters:

$$\theta = \theta - \alpha_M \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\tau_i}(f_{\theta_i'}), \tag{3.8}$$

where α_M is the learning rate of the outer loop or meta-iteration. This optimization minimizes the loss function *of future update steps* across all available tasks, yielding a parameter set θ that can be rapidly adapted to solve specific tasks within the task distribution.

To align this framework with MAML, the control signal is set as $g = \theta$, which, in the two-layer linear network, corresponds to the initial weights: $g = \theta = (W_1(t = 0), W_2(t = 0))$.

The network's performance is then defined as

$$\mathcal{P}(t_i) = -\sum_{\mathcal{T}_i \sim p(\mathcal{T})} \langle \mathcal{L}_{\mathcal{T}_i}(t_i) \rangle \,, \tag{3.9}$$

where $\langle \mathcal{L}_{\mathcal{T}_i}(t_i) \rangle$ represents the loss function at time t_i after training the initial parameters under task $\mathcal{T}_i$. The cumulative reward can then be expressed as

$$V \approx \sum_{i=1}^{N} \delta t \gamma^{t_i} \left[\eta \mathcal{P}(t_i) - C(g) \right]. \tag{3.10}$$

Setting $\eta = 1$, $\gamma = 1$, and $C = 0$ for all g, recovers Multi-Step MAML (Ji et al., 2020). When $N = 1$ (i.e., considering only one step), the formulation reduces to standard MAML optimization:

$$\max_g V = -\mathcal{P}(t_1) = -\sum_{\mathcal{T}_i \sim p(\mathcal{T})} \langle \mathcal{L}_{\mathcal{T}_i}(t_1) \rangle = -\mathcal{L}_{\text{MAML}}, \tag{3.11}$$

where at t_1, only one update step is considered from the initial parameters $g = (W_1(t = 0), W_2(t = 0))$. Finally, maximizing the value in the previous equation using Algorithm 1 is equivalent to optimizing standard MAML. Extending this to multiple time steps, as in Eq. (3.10), recovers Multi-Step MAML.

3.3.2 Meta-learning beyond control

The framework is based on the premise that at least one free parameter does not evolve according to a predefined learning rule, as described by the learning dynamics in equation 3.23. This framework can be adapted to learn any free parameters not governed by the learning dynamics, such as control signals or hyperparameters, to maximize value.

However, a significant portion of the meta-learning literature focuses on scenarios where no parameters are explicitly trained to optimize an outer-loop loss or meta-learning objective. Instead, meta-learning emerges naturally from training on a large distribution of tasks or

a diverse set of tasks (Wang et al., 2017; Team et al., 2021). In these cases, no distinct control signal or set of parameters is explicitly optimized in an outer loop. Instead, all parameters are trained across all tasks, leading to solutions capable of one-shot learning in previously unseen tasks. The underlying intuition is that solving a broad task distribution with a fixed set of parameters forces the model to develop strategies that generalize across tasks. This results in "meta-solutions" that either generalize well or can be learned within a few trials.

Another class of meta-learning algorithms follows a "memory-based" approach (Ritter et al., 2018; Genewein et al., 2023). These models utilize stored memory of past experiences to improve task performance, as exemplified by "Neural Episodic Control" (Pritzel et al., 2017). In this paradigm, meta-learning emerges through the use of memory rather than being explicitly controlled, as in the current framework.

Additionally, the learning dynamics of neural networks trained on action-value or value functions are often more complex and difficult to describe (Bordelon et al., 2023; Patel et al., 2024) than those of the regression problem examined in this study. The framework could be extended in future work to account for these types of models. However, it is likely that the framework will be most insightful in cases where a subset of parameters is optimized toward a distinct meta-learning objective within an outer loop.

3.4 Expected Value of Control

In EVC theory (Shenhav et al., 2013), a model is proposed to account for cognitive control allocation by estimating the action-value function (or signal-value in this case) for each possible control signal, explicitly incorporating the cost of taking an action. It also suggests that the dorsal anterior cingulate cortex (dACC) is responsible for integrating and computing most of the quantities required in EVC theory, including the expected payoff of a controlled process, the amount of control needed, and the cost associated with

executing control.

The EVC is posed in a reinforcement learning setting where the control cost is made explicit, and the EVC quantity is the action-value (Q-value function). Starting from the bellman equation for q_π as

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) \left[\bar{r} + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right], \tag{3.12}$$

where s denotes the current state, a the action taken $\bar{r} = r - C(a)$ with $C(a)$ the control cost (a being the control signal), r the reward received from the environment (see equation 1 and 2 in Shenhav et al. (2013)), and π a given policy (which will end up being the control signal $g(t)$ later on), then

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) \left[r - C(a) + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right], \tag{3.13}$$

$$EVC(s,a) = q_\pi(s,a) = \sum_{s',r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right] - C(a). \tag{3.14}$$

The expected value of control (EVC) is the action-value function where the cost of taking actions (control) is explicitly incorporated. It is now shown that this quantity is equivalent to the cumulative reward presented in equation 3.3 in the main text under certain conditions. First, the state $s = w(t)$ is indexed with time learning time, assuming a non-stochastic policy for the control signal at each state, $a = g(t)$, for that particular state. From one step to the next, a small change in time δt is considered (making the state transition deterministic, $p(s' = w(t + \delta t)|s = w(t)) = 1$). Additionally, reward and

cost are redefined in terms of reward per time unit, yielding

$$EVC(s = w(t), a = g(t)) = \sum_r p(r|w(t), g(t)) \left[r\delta t \right. \tag{3.15}$$

$$\left. + \gamma^{\delta t} EVC(s = w(t + \delta t), a = g(t + \delta t))\right] - C(g(t))\delta t \tag{3.16}$$

$$= \delta t \left(\mathbb{E}\left[r|w(t), g(t)\right] - C(g(t))\right) \tag{3.17}$$

$$+ \gamma^{\delta t} EVC(s = w(t + \delta t), a = g(t + \delta t)) \tag{3.18}$$

$$= \delta t \left(\mathbb{E}\left[r|w(t), g(t)\right] - C(g(t))\right) \tag{3.19}$$

$$+ \gamma^{\delta t} \delta t \left(\mathbb{E}\left[r|w(t + \delta t), g(t + \delta t) - C(g(t + \delta t))\right]\right)$$

$$+ \gamma^{2\delta t} EVC(s = w(t + 2\delta t), a = g(t + 2\delta t)). \tag{3.20}$$

The EVC term in the previous equation can be unrolled until the termination of the task at time T, where $r = 0$ for any $t > T$, indexing time as $t_0 = 0$, $t_N = T$, and $t_{i+1} = t_i + \delta t$. This gives

$$EVC = \sum_{i=0}^{N} \delta t \gamma^{t_i} \left[\mathbb{E}\left[r|t, g(t)\right] - C(g(t_i))\right] \tag{3.21}$$

which is equation equation 4.3. Taking the limit $\delta t \to 0$ recovers the integral form in equation 3.3.

3.4.1 Purpose of the control cost

Adding a control cost term to the optimization is standard in the control theory literature to describe, for example, energy consumption or depletion of some resource when applying control. In general, this cost is minimized. In this work, however, the control cost serves additional purposes.

First, it limits the space of control signals to avoid trivial solutions. For example, taking

$C = 0$ when optimizing the learning rate in the single neuron case results in the trivial solution of choosing a value such that the weight reaches the solution after one step. In the case of MAML, $C = 0$ is required to demonstrate the equivalence to the learning effort framework. Another example of $C = 0$ giving useful control signals occurs in the gain modulation case. Intuitively, because the gain modulation speeds up learning, the optimal action would be to use an extremely large gain to reach the solution weights quickly. However, the value of the control signal also affects the prediction in $Y = f(X; w(t), g(t))$, potentially increasing the loss. Simulations were conducted to verify this (not included) in the same setting as the one in effort allocation in section 3.6, except that $C = 0$ and with no restrictions on the size of G. The resultant gain modulation was qualitatively similar to the case when $C \neq 0$, but was much more concentrated and less smooth (figure not included). In this case, the cost does not benefit performance, but the goal is to study the nature of these solutions under different cost assumptions. For example, as in the task engagement case in section 3.5, where the cost function has a significant impact on the optimal control signal.

The second function of the cost is to describe mental effort when performing cognitively demanding tasks. The control cost introduced in the framework is intended to account for limited attention or a sense of fatigue. While the specific meaning of the control cost is not assumed yet, some theories of effort feeling include metabolic resource depletion (a controversial hypothesis Hagger et al. 2016; Randles et al. 2017), the opportunity cost of performing another task in the environment (Agrawal et al., 2022), reflecting a bottleneck in information processing in the brain (Musslick et al., 2020), or computation time in the presence of uncertainty (Gershman and Burke, 2023). Links to these theories can be directly tested using the framework.

3.5 Engagement Modulation

Next, the focus shifts to the question of which tasks, among many, should be engaged with over time. The model is provided with control over its *engagement* in a set of available tasks or classes in a classification problem during learning.

Selecting the optimal control signal in this setting involves improving multi-task capabilities and estimating optimal curriculum. Consider a set of N_τ datasets, and a loss function $\mathcal{L}(\hat{Y}_\tau, Y_\tau)$, where $\hat{Y}_\tau$ is the estimation of a model and Y_τ is the required target for dataset τ. The average loss for a set of datasets is $\mathcal{L} = \sum_{\tau=1}^{N_\tau} \mathcal{L}(\hat{Y}_\tau, Y_\tau) + \mathcal{R}(W)$ which is used to measure the performance $\mathcal{P}(t) = -\langle \mathcal{L}(t) \rangle$ only, and assuming that weights are updated via gradient descent on the auxiliary loss $\mathcal{L}_{\text{aux}} = \sum_{\tau=1}^{N_\tau} \psi_\tau(t) \mathcal{L}(\hat{Y}_\tau, Y_\tau) + \mathcal{R}(W)$, where $\psi_\tau(t)$ are control signals, called *engagement coefficients*, and $\mathcal{R}(W)$ is a weight decay regularizer. This auxiliary loss is used solely to obtain learning dynamics equations that explicitly depend on the engagement coefficients $\psi_\tau(t)$. Assuming that the network receives inputs from all datasets simultaneously (concatenated in X) and has specific outputs allocated to each dataset (concatenated in Y), as schematized in figure 3.5a, the learning dynamics equations for the weights can be derived as a function of $\psi_\tau(t)$, yielding

$$\begin{aligned} \tau_w \frac{dW_1}{dt} &= \sum_\tau \psi_\tau(t) W_{2\tau}^T \left(\Sigma_{xy\tau}^T - W_{2\tau} W_1 \Sigma_x \right) - \lambda W_1, \\ \tau_w \frac{dW_2}{dt} &= \sum_\tau \psi_\tau(t) \left(\Sigma_{xy\tau}^T - W_{2\tau} W_1 \Sigma_x \right) W_1^T - \lambda W_2, \end{aligned} \tag{3.22}$$

where $W_{2\tau}$ denotes the weights of the neurons for the output to dataset τ and $\Sigma_{xy\tau}$ is $\left\langle X Y_\tau^T \right\rangle$, both padded with zeros to preserve dimension (see section A.4).

Each of the $\psi_\tau(t)$ modulates the amount of learning of each dataset. The auxiliary loss to get a learning dynamics is to avoid the trivial solution of $\psi_\tau = 0$ to minimize the loss. The optimal $\psi_\tau(t)$ can be found throughout learning by computing $\mathcal{P}(t)$, using $C(\psi(t)) = \beta \|\mu_\psi - \bar{\psi}(t)\|^2$ ($\bar{\psi} = (\psi_1(t), \psi_2(t), ...)$), then taking gradient steps on $\psi_\tau(t)$ to

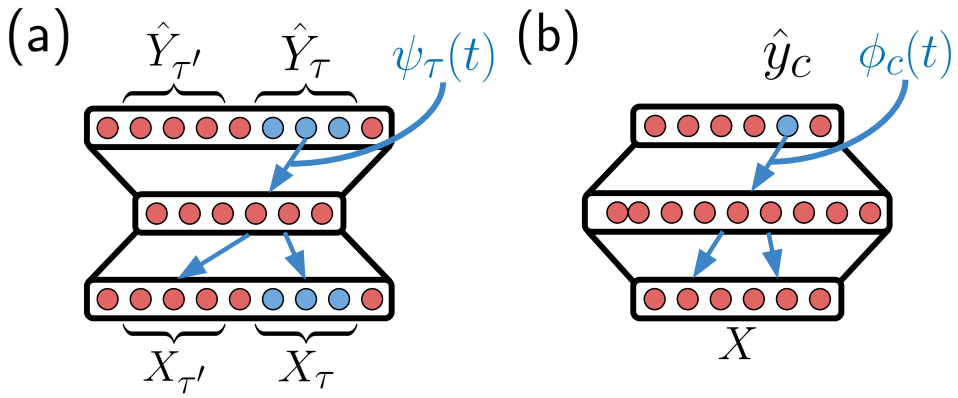

Figure 3.5: **(a)**: Task engagement, where the control signal determines the optimal amount of *engagement* through time to multiple regression tasks. **(b)**: Category assimilation, where a model is trained to learn a classification task and can control the *engagement* on each class c throughout training.

maximize V. Taking $\mu_\psi = 0$ means that to learn a dataset τ ($\psi_\tau(t) > 0$) the agent must pay a cost, here named as *active engagement*. For $\mu_\psi = 1$, the agent must pay a cost to increase or suppress the learning signal from a specific dataset relative to a baseline, here named as *attentive engagement*. In these cases, each of the elements in $\bar{\psi}(t)$ are forced to stay in a certain range independently. Finally, it is possible to force $\bar{\psi}(t)$ to be of a fixed norm by making the cost $C(\bar{\psi}(t)) = \beta \left(\|\bar{\psi}(t)\|^2 - \Psi \right)^2$, such that there is a fixed overall amount of engagement to distribute, here named as *vector engagement*. For category engagement, which is focusing on particular subclasses in a classification problem, a similar set of equations can be derived (see section A.5), where the engagement on class c through learning is denoted by $\phi_c(t)$ (figure 3.1b). The meta-learning tasks used to train this model are the following:

Task engagement: Given a set of N_τ datasets and a total training period of T, the engagement modulation model described in section 3.5 was trained. The idea of this task is to estimate the optimal learning curriculum (order of datasets presented in the neural network training) that maximizes expected return V during the time period T. In this task, three binary MNIST classification datasets were used, specifically the digits $(0, 1)$,

$(7, 1)$ and $(8, 9)$ ordered by difficulty (easier to harder according to linear separability, see section A.4).

Category engagement: For a classification task, there might be a better set of classes to learn during different stages of training. The category engagement modulation model was trained (described in section 3.5 and section A.5) to estimate the optimal *engagement* or *attention* to each of the categories in a classification task (Semantic and MNIST datasets) through learning. In addition, the gain modulation model (next Section) was trained in this same setting using a *neuron basis* (see section A.3).

3.5.1 Results

Task engagement: The neural network has access to all inputs and targets for three datasets simultaneously, as described in section 3.5, each of them a different binary regression problem from MNIST. Each dataset used was chosen to vary on the level of difficulty to learn: the pair of numbers (0, 1) is easier to classify than (7, 1) and (8, 9) (based on the lowest loss achievable with linear regression in subsection A.9.7). An engagement coefficients $\psi_\tau(t)$ was assigned to each dataset, that maximizes the expected return in equation 3.3. Learning curves and the evolution of engagement coefficients are depicted in figure 3.6; the baseline case corresponds to simultaneous training on all datasets at the same time ($\psi_\tau(t) = 1$ and $C(\psi(t)) = 0$). In the *attentive engagement* agent, where $\mu_\psi(t) = 1$ (shown in figure 3.6a and b), the agent just needs to pay a cost to either amplify or suppress engagement on a dataset. In this setting, the agents amplify the engagement of all of the datasets, effectively increasing the learning rate per dataset, and achieving a lower $\mathcal{L}_C(t)$ compared to $\mathcal{L}_B(t)$. The order of learning each of the datasets goes from easier to harder, it is in the same order as in the *active engagement* and *vector engagement*, and none of the datasets are engaged with $\psi_\tau(t) < 1$, avoiding forgetting of early amplified datasets. In the case of *active engagement*, where $\mu_\psi = 0$ in $C(\psi(t)) = \beta\|\mu_\psi - \bar{\psi}(t)\|^2$ (shown in figure 3.6c and d), the agent must pay a cost to learn any of the tasks ($\psi_\tau(t) > 0$).

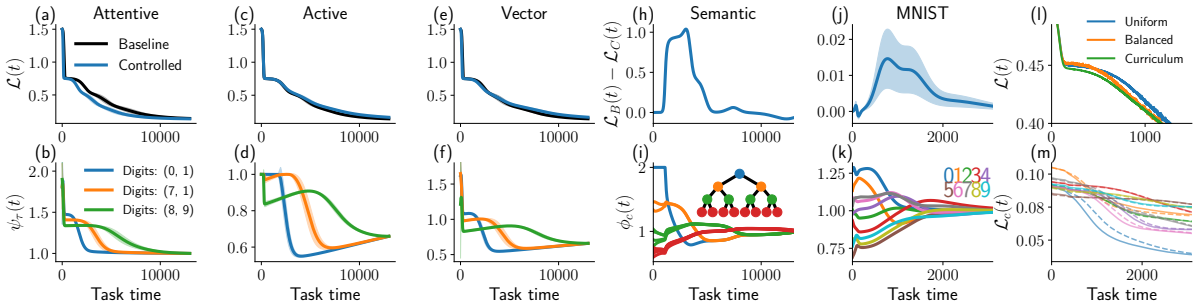

Figure 3.6: Results for task engagement experiment. **(a)**, **(c)** and **(e)**: $\mathcal{L}(t)$ for baseline and control case for Attentive, Active and Vector engagement. **(b)**, **(d)** and **(f)**: Engagement coefficients $\psi_\tau(t)$ for each of the binary classification tasks Attentive, Active and Vector engagement. Mean and standard deviations from 5 independent trainings. **(h)** and **(j)**: Results for category engagement task, improvement in the loss function when using control for MNIST and Semantic dataset respectively. **(i)** and **(k)**: Optimal category engagement coefficients for MNIST and Semantic datasets. **(l)**: Class proportion experiment. **Uniform**: Loss when using uniform distribution for the abundance of classes in each batch. **Balanced**: Loss on a balanced batch, but using the inferred curriculum of classes in the batch to train. **Curriculum**: Loss on curriculum batch when using the curriculum. **(m)**: Loss per class using control (solid lines) and baseline (dashed lines).

By distributing the learning between the tasks, the agent is capable of reaching $\mathcal{L}_C(t)$ close to $\mathcal{L}_B(t)$ as shown in the top panel of figure 3.6, without the need of fully engaging on all of the datasets at every time step. None of these datasets are fully disengaged at any point, possibly as a mechanisms to avoid catastrophic forgetting (Kirkpatrick et al. 2017) of datasets previously engaged during training. The engagement coefficients in the *vector* case behave similarly. Since the control signal in this case is forced to keep a constant size of Ψ, the agent is not able to fully engage in all of the datasets, and distributes this *attention* resource on each dataset from easier to harder, as in the *active* case. The meta-learning strategy found in our setting of keep re-visiting previous tasks to keep performance is well studied in psychology (Ericsson and Harwell, 2019; Eglington and Pavlik Jr, 2020), and it is also related to memory *replay* theories as a value-based mechanism that avoids catastrophic forgetting (Mattar and Daw, 2018; Agrawal et al., 2022).

Category engagement: In some classification tasks, it might be better to learn some

categories first and others later during training. The engagement modulation model to control *engagement* and *attention* was trained on categories of a classification dataset. In figure 3.6, the results of this model trained on the Semantic dataset, and MNIST dataset classifying all digits are shown. The engagement coefficients $\phi_c(t)$ describe the focus on class c in the classification problem, which basically scales the error signal for that specific class through training (see section A.5). figure 3.6h and j show the improvement in the loss when optimizing for categoric assimilation coefficients for both datasets. figure 3.6i and k depict the engagement coefficients per class $\phi_c(t)$. In the Semantic task, the engagement coefficients are clustered depending on the level of the hierarchy for the respective output. Higher coefficients are spent on categories in higher levels of the hierarchy, as well as earlier during learning. Because β is high for this experiment ($\beta = 5.0$), the cost of deviating from a control vector of size C is high (where C is the number of classes); therefore the amplification of engagement in some categories goes along with suppression for other categories to keep the control with constant size. For the MNIST dataset, each $\phi_c(t)$ corresponds to a specific digit, and the order of assimilation that maximizes value shows a consistent order of digits among different runs, being ordered as $(0, 1, 7, 6, 4, 2, 3, 9, 8, 5)$, which is roughly the same as the average linear separation per digit (see subsection A.9.7). As in the task engagement results, results show that it is optimal to assimilate easier elements first, allocating higher $\phi_c(t)$ and more concentrated in the early stages of learning. More difficult categories are assimilated later, allocating a smaller maximum $\phi_c(t)$ compared to easier classes, but with sustained engage over time. The benefits of learning from easier to harder aspects of tasks have been shown in cognitive science (Krueger and Dayan, 2009; Wilson et al., 2019) and machine learning (Parisi et al., 2019; Saglietti et al., 2022; Zhang et al., 2022), the engagement and category engagement experiments within our normative framework resemble these findings on task difficulty curriculum. The engagement level per class amplifies the error signal of learning a particular class through time, which can be roughly controlled by modifying

the proportion of classes in the batch through training. To show this, a baseline agent (no control, only backpropagation) was trained on MNIST, and used $\phi_c(t)$ to modify the proportion of classes in the batch throughout the training (section A.5). This gives a better curriculum than sampling each class uniformly to populate the batch, as shown in figure 3.6l and m.

3.6 Gain Modulation

Motivated by studies of neuromodulation (Lindsay and Miller, 2018; Ferguson and Cardin, 2020), this section presents a model with a *learning effort control signals* as gain modulation $G_1(t) \in \mathbb{R}^{H\times I}$ and $G_2(t) \in \mathbb{R}^{O\times H}$ modulate the gain of each layers weights as $\tilde{W}_i(t) = (\mathbb{1} + G_i(t)) \circ W_i(t) = \tilde{G}_i(t) \circ W_i(t)$ where $\circ$ denotes element-wise multiplication. This control signal will modify the input-output mapping of the network to $\hat{Y} = \tilde{W}_2(t)\tilde{W}_1(t)X$. Given the control signals, the weights are learned using gradient descent, yielding the learning dynamics equations

$$\begin{aligned}
\tau_w \frac{dW_1}{dt} &= \left(\tilde{W}_2^T \Sigma_{xy}^T\right) \circ \tilde{G}_1 - \left(\tilde{W}_2^T \tilde{W}_2 \tilde{W}_1 \Sigma_x\right) \circ \tilde{G}_1 - \lambda W_1, \\
\tau_w \frac{dW_2}{dt} &= \left(\Sigma_{xy}^T \tilde{W}_1^T\right) \circ \tilde{G}_2 - \left(\tilde{W}_2 \tilde{W}_1 \Sigma_x \tilde{W}_1^T\right) \circ \tilde{G}_2 - \lambda W_2.
\end{aligned} \tag{3.23}$$

The control signal $G_i(t)$ effect is *similar* to a time-varying learning rate, except (1) it is weight specific (i.e. with coupling between the elements of the control matrix), (2) it does not change the weight decay rate which is originally controlled by λ and τ_w, and (3) $G_i(t)$ also changes the input-output mapping. Solving the learning dynamics gives $\mathcal{P}(t) = -\langle L(t) \rangle$, using $C(G(t)) = \exp\left(\beta\left(\|G_1(t)\|_F^2 + \|G_2(t)\|_F^2\right)\right) - 1$, to then estimate $dV/dG_i(t)$ as in section 4.1, and find the control trajectory that maximizes cumulative reward in equation 3.3 (derivation of learning dynamics in a two-layer linear network given a control signal $G(t)$ is provided in section A.3). In addition, a non-linear network

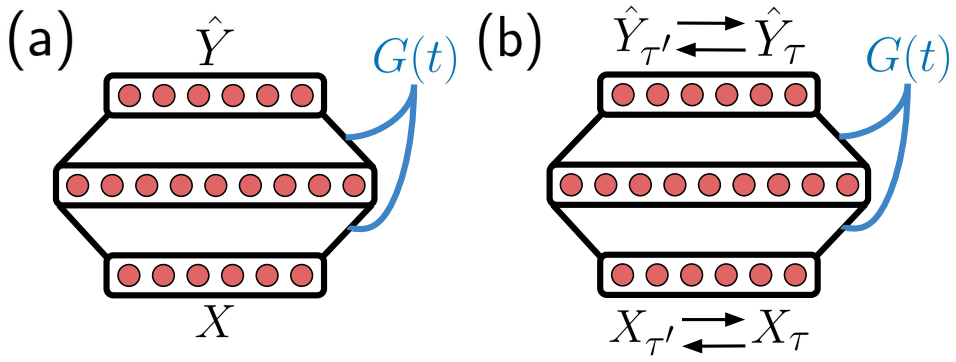

Figure 3.7: **(a)**: Effort allocation, where the control signal (gain modulation of weights) is computed to maximize value throughout the learning of a single task. **(b)**: Task switching, where the gain modulation model is trained to switch tasks repeatedly and the control signal is computed throughout the switches.

was simulated using approximations (see section A.6). The meta-learning tasks used to train this model are the following:

Effort Allocation: The gain modulation model was trained separately on each of the three datasets for a time period of T, and estimate the control signal that maximizes the expected return V in equation 3.3.

Task Switch: Two different Gaussian datasets are defined (section A.10). The network was sequentially trained on each dataset for a time period T_s. The expected reward V is computed for the whole training period $T > T_s$ of the gain modulation model, and maximized through gradient updates on $G_i(t)$.

3.6.1 Results

Effort Allocation: This setting is similar to the single neuron setting of subsection 3.2.1, but with a two layers network instead of just one neuron, where every weight in the network has its own gain signal as described in equation 3.23 and schematized in figure 3.7a. The results of the baseline training and controlled training using gain modulation are presented

in figure 3.8. In the gain modulation model, the results show the same qualitative behavior as in the single-neuron model when varying parameters of the learning model and control optimization. The control signal that maximizes expected return reduces the instant net reward rate by the use of control in the early stages of learning, to get better performance later as shown in the lower loss for the controlled case (figure 3.8a and b). Through both optimizing the learning and minimizing $C(G(t))$ at the same time, the gain modulation is not only more efficient by getting a more sparse solution (L_1 norm in figure 3.8b), using fewer weights than when no control is used, it also learns faster (figure 3.8c, more details in subsection A.9.5). There are times during learning when it is more effective to apply control. As can be seen in the L_2 norm of the control matrices $G_1(t)$ and $G_2(t)$, and the absolute value of the time derivative of the loss $d_t\mathcal{L}(t) = |d\mathcal{L}(t)/dt|$ for the baseline and control case (figure 3.8d), the control signal is larger early in training and near the stages of learning when the increase in performance ($d_t\mathcal{L}(t)$) is larger (figure 3.8a). The control signal shifts the peaks in $d_t\mathcal{L}(t)$ earlier in learning, leading to better performance and higher reward earlier, compensating for the momentarily increased cost of control. Similar results are obtained when training on the other two datasets (see figure A.7 and figure A.8 in subsection A.9.5). Neuromodulators are known to be involved in high-level executive tasks such as engagement in learning (Shenhav et al., 2013, 2017; Lieder et al., 2018; Grossman et al., 2022), and some of them are believed to act as gain modulation (Lindsay et al., 2020; Ferguson and Cardin, 2020) (see section A.1). This model provides a testable and tractable setting in which different control influences over the learning system can be evaluated against neuromodulator signals from experiments, where the subject performs and learns tasks to maximize cumulative reward.

Task Switch: The task is schematized in Fig. figure 3.7b. In figure 3.8e, each peak in the loss is a task switch (every 1800 time steps), and as expected, the baseline loss $\mathcal{L}_B(t)$ is higher than the loss with control $\mathcal{L}_C(t)$ almost at every point throughout learning. After each switch, the control signal manages to iteratively drive the learning dynamics to

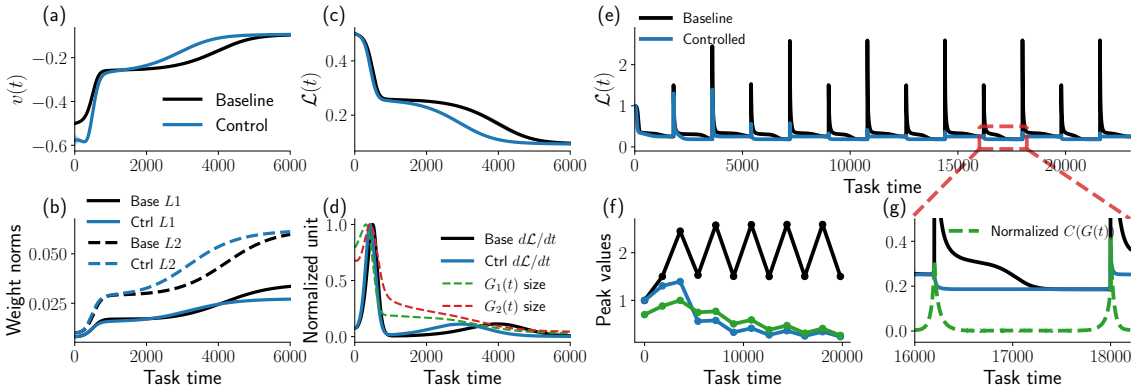

Figure 3.8: Results of the gain modulation model trained on an MNIST classification task. **(a)**: Instant net reward $v(t)$, baseline vs controlled. **(b)**: L1 and L2 norms of the weights. **(c)**: Loss $\mathcal{L}(t)$ throughout learning. **(d)**: normalized $d_t\mathcal{L}(t)$, and normalized L2 norm of the control signal $G_1(t)$ and $G_2(t)$. **(e)**: Results on the task switch meta-task. Comparison of $\mathcal{L}(t)$ for the baseline and control case. **(f)**: Values of $\mathcal{L}(t)$ at switch times, along with the normalized cost of control $C(t)$ at switch times (green line). **(g)**: Zoom of $\mathcal{L}(t)$ in the top panel, along with the normalized cost of control.

places in parameter space W where each switch is less costly (figure 3.8f). Since the linear network is over-parametrized, the drive to adjust for the next task can be done without meaningfully changing the solution for the current task. The control signal starts acting before the switch (figure 3.8g) to amortize the loss peak at the time of the switch, and to speed up the approach of the weight to the solution, skipping the plateau in the loss. In addition, the sparsity of the weights is higher compared to the baseline case, the cost of using control to switch is transferred to the size of the weights, making it easier to move the effective weight $\tilde{W}(t)$ by a large amount when changing $G(t)$ (See subsection A.9.6). This setting poses meta-learning and gain modulation as a neural implementation of task/context switching, which could be compared with behaviors in real scenarios where the switch is cued or periodic (Puigbò et al. 2020; Ben-Iwhiwhu et al. 2022).

3.7 Direct Comparison with Behavioral Data

In this section, the learning effort framework is applied to a behavioral experiment in which a trade-off between control and learning speed has been proposed (Masís et al., 2023). First, a description of the behavioral task and main findings is provided, along with the learning model proposed by the authors. Then, a model adapted to the learning effort framework is presented, named *epistemic noise control model*. While retaining some features of the model in Masís et al. (2023), the adapted model does not necessarily describe an implementation, such as the recurrent neural network and drift-diffusion model as in the original work. Instead, it characterizes the temporal trade-off of applying control while sacrificing some instantaneous reward rate.

To account for key findings in the behavioral task, the adapted model includes the capacity to control the **epistemic noise** (meaning the noise that can be reduced by learning), which is done by increasing the exposure time to the stimulus on each trial reducing the instantaneous reward rates, the noise in the error signal, and increaasing learning speed. The ability to reproduce many of the findings from the behavioral experiment using a simpler model suggests a fundamental trade-off in controlling learning.

3.7.1 Managing Learning during Decision Making in Rats

In the work presented by Masís et al. (2023), rats were exposed to a binary classification task (figure 3.9). In each trial, the animal was presented with one of two possible stimuli, each corresponding to a different abstract image. The rat could collect a reward by licking a port depending on the image presented, one image corresponded to the left port, and the other to the right port. After the rat achieved a certain level of proficiency in performance, a randomly transformed version of the stimulus (effectively a new stimulus) was introduced, and training continued for 10 additional sessions from that point.

Main Findings: Upon the presentation of the new pair of images (after achieving

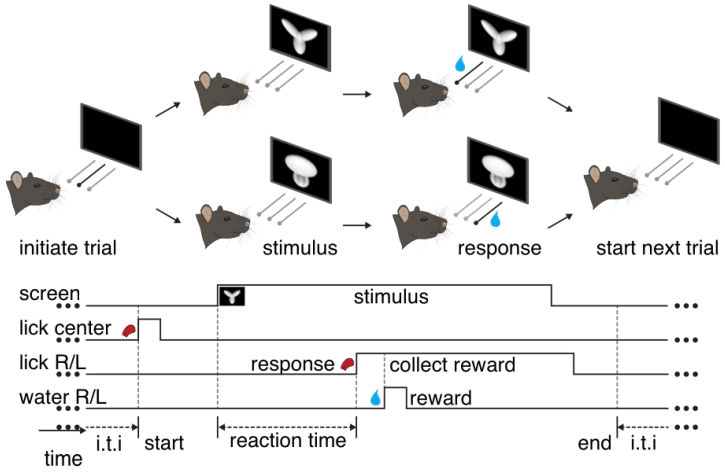

Figure 3.9: Schematics of the binary classification task in rats. Figure from Masís et al. (2023).

proficiency with the previous pair), rats were conditioned to exhibit a specific reaction time, forming two groups: **high RT** and **low RT** at the beginning of learning. Accuracy, inferred SNR (using the RNN and LDDM model explained later), instantaneous reward, and cumulative reward were measured on average for each group throughout the entire session. Rats in the high RT group exhibited faster learning compared to those in the low RT group, improving their accuracy and inferred SNR more rapidly during learning (figure 3.12a and b). However, because the reaction times were longer at the start of learning the new stimulus for the high RT group, their instantaneous reward rate was lower compared to the low RT group. Despite this, the group with faster learning, while initially sacrificing some reward rate, was ultimately able to obtain higher cumulative rewards than the low RT group (figure 3.12c and d). These results were also reproduced when grouping rats by their natural initial reaction time, without the need for conditioning (Figure 7 in (Masís et al., 2023)).

Transparent Stimulus Experiment: A variation of the experiment is also presented, in which the learnability of the task is modified. To achieve this, the authors introduced a pair of completely transparent stimuli to a subset of proficient rats from the base experiment

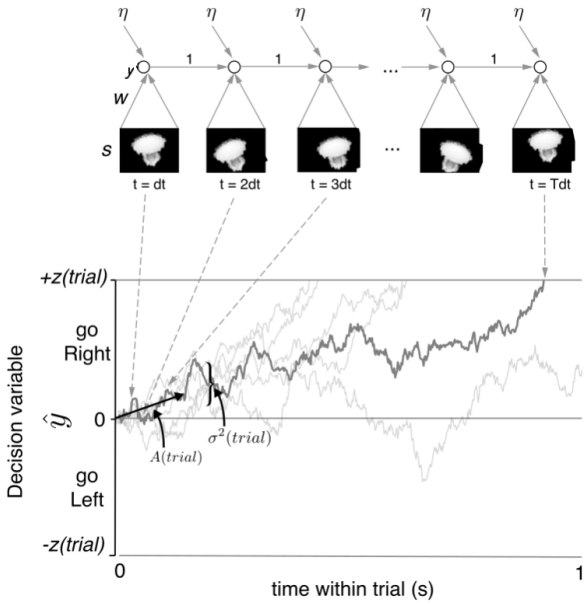

Figure 3.10: Schematics of the RNN generating the decision variable signal. Once the decision variable $\hat{y}$ reaches a threshold $z(\text{trial})$, the model decides right or left depending on the sign of $\hat{y}$. A consequence of increasing the threshold $z(\text{trial})$ can be seen graphically: as the threshold increases, the time it takes to reach the threshold increases (longer RTs), while collecting more evidence from the stimulus, improving the accuracy and speeding up learning. Figure from Masís et al. (2023).

with visible stimulus. In other words, the rats could not see the images, making the task impossible to learn. These rats followed an optimal strategy, which involved minimizing reaction time as much as possible. Since the stimulus could not be learned, the probability of choosing correctly remained fixed and could not improve. Therefore, reacting as soon as the stimulus was presented maximized both the instantaneous reward rate and cumulative reward. This behavior contrasts with that of rats in the non-transparent condition, whose reaction times started high and gradually decreased back to proficiency levels (figure 3.13a and b).

Original Model: To explain these behavioral findings, the authors proposed a model combining a Recurrent Neural Network (RNN) and a Linear Drift Diffusion Model (LDDM)

(Bogacz et al., 2006). The model is able to make binary decisions through a drift diffusion process. Within each trial, the model receives an image input corresponding to one of the two images, modeled as a Gaussian input $x(t) \sim \mathcal{N}(\mu y dt, \sigma_x^2 dt)$, where y is the image identity, μ regulates the signal of y carried in x, and σ_x is the irreducible overlap between the stimuli (aleatoric uncertainty). This is very similar to the data generation process used in the single neuron example in subsection 3.2.1, with the notation matched to this work. The decision variable $\hat{y}(t)$ within one trial time t is governed by

$$\hat{y}(t + dt) = \hat{y}(t) + w(\text{trial})x(t) + \eta(t), \tag{3.24}$$

where $\eta(t) \sim \mathcal{N}(0, \sigma_0^2 dt)$ is the output noise (epistemic uncertainty, as this can be overcome by learning), and w is a scalar gain parameter that is trained through gradient descent from trial to trial as

$$w(\text{trial} + 1) = w(\text{trial}) - \lambda \frac{\partial L}{\partial w}, \text{ with } L(w, y) = \max(0, 1 - \hat{y}y). \tag{3.25}$$

The decision variable described in equation 3.24 can be viewed as the output of a recurrent neural network, where the forward weight is w, and the recurrent weight feeds the previous output $\hat{y}(t)$ to the next one $\hat{y}(t + dt)$. The DDM aspect of this model is that the decision on the image identity (licking left or right) is made once $\hat{y}(t)$ reaches a threshold z (See figure 3.10).

Given the network is linear and the data is Gaussian, the authors are able to provide closed-form dynamics of the agent, for instance, showing the trade-off between learning speed and instantaneous reward rate. The hypothesis in this experiment is that rats "approximately maximize total reward over the full learning epoch", and that they can control their own learning process by scheduling the decision threshold $z(\text{trial})$, as it influences the trial accuracy, reaction time, and learning speed. Using the RNN and LDDM formulation, it is possible to provide a *globally optimal threshold scheduling* $z^*(t)$

that maximizes

$$z^*(t) = \arg\max_{z(t)} \int_0^T RR(t)dt \tag{3.26}$$

with $RR(t)$ denoting the reward rate, and T the amount of time available within an epoch to learn. Note that this integral follows a similar rationale to the optimization goal described in chapter 3. An optimal policy for the threshold can be optimized through gradient descent, which gives an exponential-like solution. Heuristics for surrogate optimal threshold scheduling are also provided. Optimized threshold scheduling compared with a baseline threshold reproduces many of the behavioral observations, analogous to the ones shown in figure 3.12 and figure 3.13. In the next section, an alternative model controlling epistemic uncertainty is provided.

3.7.2 Epistemic Noise Control

The reason to provide an alternative model that can reproduce the data is to suggest that **the problem of immediate reward vs. costly control and better future rewards is a fundamental problem when controlling learning**. What this means exactly is that controlling aspects of learning can create a trade-off between learning progress and immediate reward, and that the optimal control will behave similarly in many instances of learning and control, exerting more control at the start of learning and reducing control later. This is further argued in chapter 4. Support for this hypothesis is given, in this particular application, by the simplified nature of the following model, as the DDM is removed, and there are no within-trial dynamics. The main aspect that presents this trade-off is the capacity to improve learning speed by paying a cost (in this case, instant reward rate). In the following model, this control is simply done by being able to reduce epistemic noise (hence, improving learning updates) while reducing the instantaneous reward rate due to longer times taken on each learning update. As an extra advantage,

the simplicity of the model might allow for mathematical analysis of the optimal control, perhaps utilizing the methods developed in section 4.3.

Formally, the model can be presented as follows: A single-weight network is trained to solve a similar task as the original model. The model needs to report the target sign $y = \pm 1$ with $P(y = 1) = P(y = -1) = 1/2$ given an input $x(t) \sim \mathcal{N}(\mu_x y, \sigma_x^2)$. The decision variable is defined by $\hat{y} = \text{sign}(w(t)x + \frac{\eta}{(1+g(t))})$, where $\eta \sim \mathcal{N}(0, \sigma_0^2)$, and the control signal $g(t)$ is introduced. Increasing $g(t)$ will decrease the output noise, hence increasing the carried signal from y to $\hat{y}$. This can be quantified by the signal-to-noise ratio

$$\text{SNR}(t) = \frac{\mu_x^2 w^2(t)}{w^2(t)\sigma_x^2 + \frac{\sigma_0^2}{(1+g(t))^2}} \xrightarrow{g \text{ large}} \frac{\mu_x^2}{\sigma_x^2}. \tag{3.27}$$

Therefore, learning is necessary as long as there is output noise η that needs to be overcome. This can be understood as reducing *epistemic uncertainty* since it can be reduced by learning, and σ_x being aleatoric uncertainty, which is intrinsic to the data generation process. Given the decision variable $\hat{y}$ and the data distribution, the accuracy $a(t)$ is

$$a(t) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{\mu_x w(t)}{\sqrt{2(w^2(t)\sigma_x^2 + \frac{\sigma^2}{(1+g(t))^2})}}\right)\right]. \tag{3.28}$$

The weight $w(t)$ is trained to maximize this accuracy

$$\frac{dw}{dt} = \alpha \frac{da(t)}{dw} - \lambda w \tag{3.29}$$

where α is the learning rate, and λ controls a square norm regularizer. It can be shown that the accuracy derivative $da(t)/dw$ is always larger than 0 under some conditions,

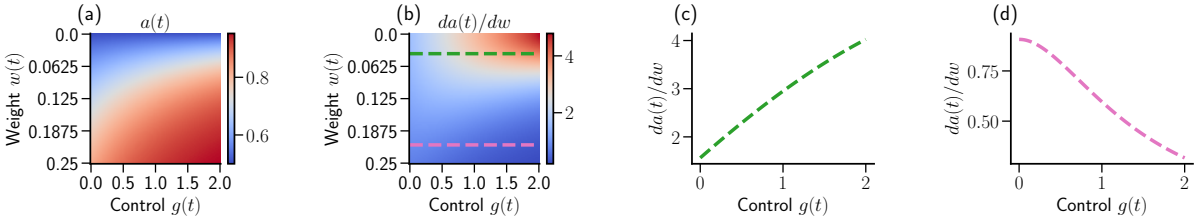

Figure 3.11: Accuracy $a(t)$ and accuracy gradient $da(t)/dw$ as functions of the weight $w(t)$ and control signal $g(t)$. **(a)**: Accuracy improves through two mechanisms: increasing the signal of x via the weight $w(t)$ to overcome epistemic noise η, and directly reducing epistemic noise η. **(b)**: The gradient of accuracy increases with $g(t)$ for small $w(t)$, as it enhances the SNR of the gradient update (see green line in **(b)** and **(c)**). For large $w(t)$, where $a(t)$ saturates, the effect of control diminishes (see pink line in **(b)** and **(d)**).

e.g. $\sigma_0^2 > 0$ (Shown numerically in figure 3.11 and proven in section A.7). The control signal influences the weight dynamics. As both the weight $w(t)$ and control $g(t)$ increase, accuracy improves because $w(t)$ helps overcome output noise, while $g(t)$ reduce the noise (figure 3.11a). The control signal can increase the accuracy gradient $da(t)/dw$ when $w(t)$ is small enough that accuracy has not yet saturated. By reducing the epistemic error η, control increases the signal-to-noise ratio (SNR), thereby accelerating learning (figure 3.11b, c). When $w(t)$ is sufficiently large, increasing $g(t)$ decreases the always positive $da(t)/dw$, because reducing noise further does not contribute to the accuracy $a(t)$ when it is already saturated (figure 3.11b, d).

The instantaneous reward rate can be defined as

$$iRR(t) = a(t) \cdot \frac{\xi}{1 + \beta g(t)}, \tag{3.30}$$

making the reward proportional to accuracy (obtaining a reward of ξ with probability $a(t)$, otherwise 0). The rate at which rewards are obtained is weighted by $1/(1 + \beta g(t))$, meaning that an increase in control extends the time required to make a decision and collect a reward at each step, while simultaneously reducing epistemic noise, as described earlier. The trade-off between the time invested in each training step to reduce noise and

the acceleration of learning creates an **inter-temporal choice of control allocation**, where learning speed can be increased at the cost of reducing the iRR. Notably, increasing control $g(t)$ can be interpreted as an explicit cost within the learning effort framework defined in equation 3.3, by considering the first-order effect of increasing $g(t)$, giving

$$iRR(t) \approx a(t)\xi - a(t)\xi\beta g(t) = \mathcal{P}(t) - C(t). \tag{3.31}$$

where $\mathcal{P}(t)$ and $C(t)$ are performance and cost as defined in section 3.2. This is similar to an opportunity cost of the reward lost by waiting the extra amount of time $\beta g(t)$ to make a decision. Finally, to find the optimal noise control scheduling g^* throughout learning,

$$g^*(t) = \arg\max_{g(t)} \int_0^T iRR(t)dt \tag{3.32}$$

which can be solved using gradient ascent as explained in the learning effort framework in section 3.2.

3.7.3 Simulation Results

By optimizing the noise control schedule $g(t)$ through gradient ascent, it is possible to qualitatively reproduce some behavioral findings in Masís et al. (2023). The optimized agent is referred to as the **high-control** agent, while the baseline agent with no control ($g(t) = 0$) is called the **low-control** agent, corresponding to a baseline level of control (Figure 3.12).

The high-control agent exhibits a control profile that starts high at the beginning of training (figure 3.13d). This is explained by the effect of control on the accuracy gradient, which accelerates learning. Control then decreases as its impact becomes less useful once the task is learned through $w(t)$. This leads to a faster improvement in accuracy and signal-to-noise ratio (SNR) compared to the low-control case (figure 3.12e and f).

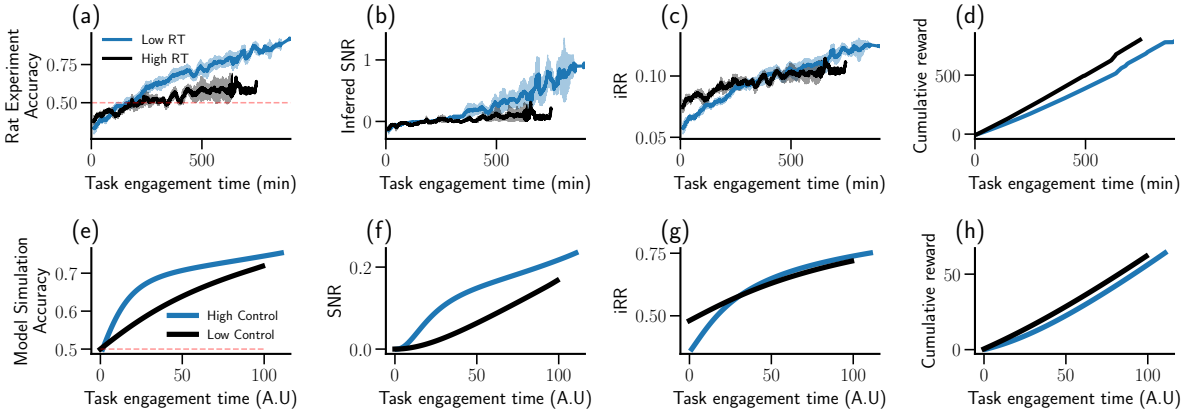

Figure 3.12: **(a) to (d)**: Original behavioral results observed in the rat experiment, adapted from Masís et al. (2023). **(e) to (h)**: Results from optimizing control (high control) with respect to a baseline (low control).

Since high control results in longer reaction times (here, longer stimulus exposures reduce epistemic noise η), the instantaneous reward rate of the high-control agent is lower at the start of training. However, the total collected reward throughout training is larger for the high-control agent, as maximizing total reward is the objective of the control signal $g(t)$, as expressed in equation 3.32 (figure 3.12g and h).

This cumulative reward is maximized over a fixed number of trials. However, because increased control affects reaction time, each trial in the integral takes slightly longer for the high-control agent than for the low-control agent. As a result, the total task engagement time is longer for the high-control agents. These simulations qualitatively match the behavioral observations (first row of plots in figure 3.12).

The optimal control of epistemic noise can also be simulated for a transparent stimulus by simply increasing the value of σ_x^2 and then computing the optimal control using gradient ascent. For a transparent stimulus (not fully transparent, as $\mu_x > 0$), since the error rate cannot be decreased, the optimal way to maximize reward (through task time, not trials) is to answer as quickly as possible on each trial, thereby minimizing control (figure 3.13). The control exerted, in the base task with a visible stimulus, will

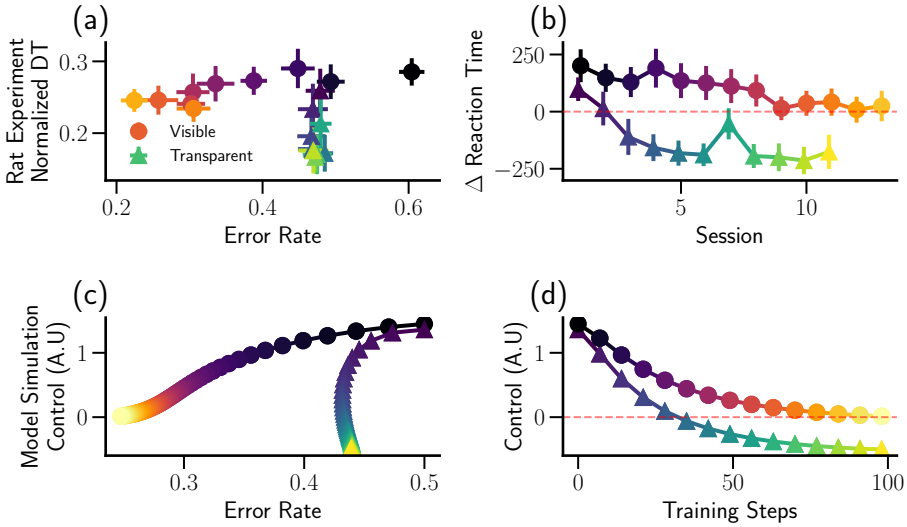

Figure 3.13: **(a) and (b)**: Behavioral results from rat experiments adapted from Masís et al. (2023). **(a)**: Normalized decision time vs. error rate for rats. For transparent stimuli, rats reduce their decision time more than for visible stimuli, without decreasing their error rate, as they cannot discriminate the stimulus. **(b)**: When presented with a new stimulus, rats' reaction times start high as they learn the task and return to the same reaction times once proficient with a visible pair of stimuli. When faced with a transparent stimulus, reaction times decrease further.**(c) and (d)**: Simulated optimal control exhibits similar reaction time behavior to the experimental results for the visible and transparent stimulus tasks.

start high to boost learning (leading to high initial RTs in the case of rats, lowering their initial instantaneous reward rate) and will then decrease to levels comparable to proficient performance, which in the simulation corresponds to $g(t) = 0$ (figure 3.13b and d). In the transparent stimulus condition, control is minimized to the extent that is perceptually and mechanically possible for the rat. In the model, this was enforced simply by including a lower bound on control. Note that the optimal control for the transparent stimulus is not minimized at all training steps (figure 3.13c and d). This is because the initial control signal for starting the gradient ascent iterations to maximize cumulative reward is the optimal control in the base task with visible stimulus, which starts with high control. This initial condition simulates a control prior over the optimal control in the base task

with a visible stimulus before performing the task with a transparent stimulus, as in the case of rats.

3.7.4 Interpretation of Cognitive Control

Cognitive control is generally defined as a mechanism that overrides habitual or default behavior to pursue context-dependent goals (Shenhav et al., 2013; Cohen, 2017). Considering this definition, the optimal cognitive control allocation will depend on the task at hand, and it will have an impact on behavioral variables, e.g., reaction times. In addition, the *amount* of cognitive control is not directly measured but inferred by observing surrogate behavioral or neural signals, such as reaction times, pupil size, or activity in related brain regions.

In the *epistemic noise control* model, we have assumed that the amount of cognitive control increases reaction times while reducing the amount of noise in the response, thereby increasing accuracy. This is the same interpretation as in the original work (Masís et al., 2023; Masis et al., 2024). This interpretation diverges from observations in other experiments, where a higher level of cognitive control is associated with lower RTs but also lower accuracy. Both instances of cognitive control are consistent with the speed-accuracy trade-off (Heitz, 2014), which has been modeled in the past, e.g., using DDMs (Bogacz et al., 2006). Below we argue why, in our model, more control leads to higher RTs, and how this relates to decision noise.

There are multiple experiments in the literature where cognitive control is associated with lower reaction times, intuitively understood as the subject putting more effort into the task (higher levels of attention), thereby being able to solve each trial faster. Some of these experiments are related to the Gratton effect, where stimulus-conflicting trials show higher RTs, and two conflicting or hard trials in a row produce lower RTs and improved performance on the second trial. This can be explained as using the first trial to properly

adjust cognitive control or attention to the current difficulty of the set of trials (Gratton et al., 1992; Botvinick et al., 2001; Ullsperger et al., 2005; Kool et al., 2010). Note that this reduction in RTs is compared with conflictive trials, and in most cases, in easier non-conflictive stimulus trials, the reaction times are even lower.

Another set of experiments is related to time-constrained tasks, where cognitive control is associated with faster reaction times (sacrificing performance as well), as responding quickly is required to obtain reward (Forstmann et al., 2008; Ulrich et al., 2015; Miletić and van Maanen, 2019; Stone et al., 2025). In both of these examples (and perhaps other instances), cognitive control is associated with lower RTs.

One way to understand the conflict between the standard interpretation in the literature (e.g., the Gratton Effect) and the epistemic noise model is by noticing that RTs increase in conflicting trials not only due to the conflict itself, but also because it is often reward-optimal, which is the goal of cognitive control according to the EVC theory. In principle, the subject could answer very quickly regardless of the conflict on a given trial, but it is better to operate in a regime of higher RTs and higher accuracy within the speed–accuracy trade-off (SAT). This reflects a direct trade-off between effort and reward, as in some cases the subject could choose to respond randomly instead of taking the time to resolve the conflict. Evidence for this comes from (Kool et al., 2010) (experiment 6), where subjects reduced their reaction times on harder trials but adopted a slower (higher RT) strategy when higher rewards were offered. Similarly, (Alfers et al., 2025) showed that manipulating reward and time constraints can elicit a range of speed–accuracy behaviors within the same subject, depending on their current goals.

It is speculated that there is an intrinsic operating point in the SAT for every subject on a given task, and deviations from this operating point require cognitive control. These trade-offs translate directly into a reward–time cost trade-off, and depending on the task and the intrinsic urgency of a subject, as measured in (Yau et al., 2021) and observed

in (Masis et al., 2024), individuals may settle at different operating points. This view aligns with the more general definition of cognitive control, which involves operating beyond or overriding default behavior, in this case, the default operating point in the SAT (Piray and Daw, 2021). In the context of the Gratton effect, the observation that reaction times can be further reduced for a second consecutive conflictive trial suggests that cognitive control can be further adjusted, perhaps starting from a state closer to the default operating point on the first trial.

Finally, the epistemic noise modeling is discussed in relation to decision noise in a DDM, as in the original model. In the proposed framework, epistemic noise is reduced by the control signal, but *it is not equivalent to the decision noise of a DDM*. Epistemic noise models the probabilistic overlap between stimuli, which can be reduced through training (increasing the SNR via the weights). This differs from the standard decision noise in the drift-diffusion process, which is integrated over time (Bogacz et al., 2006; Ulrich et al., 2015). The two concepts are related: epistemic noise in the proposed model defines the final expected SNR when making a decision, while decision noise is integrated throughout the decision process. Moreover, the SNR in a DDM is not solely determined by noise but also by the threshold and the drift rate. In this sense, epistemic noise can be seen as a collapsed representation of the resulting SNR given the drift, threshold, and decision noise. This creates a degeneracy in the model, since similar behavioral effects can be obtained by modulating decision noise, threshold, or drift rate, making the influence of control ambiguous. However, this is by design: the goal of the model is to capture the relation between control and learning, where the latter improves as epistemic noise is reduced through control. Disentangling the influence of control on a trial-by-trial basis in standard DDM modeling remains an open question requiring further research (Lee and Sewell, 2024; Stone et al., 2025).

In summary, the epistemic noise model proposed in this section diverges from the standard literature in three main aspects. First, the notion of epistemic noise differs from decision

noise during evidence accumulation; instead, it represents the separation of stimuli at decision time, as determined by the collapsed DDM dynamics. Second, higher cognitive control is assumed to be associated with longer reaction times, an assumption that requires further study, in addition to acknowledging the degeneracy of control effects resulting from abstracting away from the full DDM model. Third, the goal of the model is not to describe within-trial dynamics, but rather the intertemporal allocation of control, where control signals influence both the learning trajectory and trial-by-trial performance (accuracy). Consequently, the expected value of control must incorporate the future effects of control through improved learning in order to compute an optimal control signal, which is the effect the proposed model is capturing.

3.8 Discussion

This work presents a flexible and computationally tractable *learning effort* framework for studying optimal meta-learning with neural network dynamics across various settings, where control signals influence both learning and performance. The framework optimizes control signals based on a fully normative objective: the discounted cumulative performance throughout learning. The primary goal of this framework is to facilitate the evaluation of potential interventions in engineered systems and provide formal foundations for cognitive control theories in neuroscience. While a limitation of this approach is the reliance on linear network models, approximations of non-linear network dynamics are explored in section A.6. This work aims to contribute to a deeper understanding of how agents should act to maximize their learning abilities based on their own learning prospects.

The learning effort framework is sufficiently flexible to address various questions about meta-learning strategies by making slight modifications to the original setting while maintaining the underlying conceptual structure. The primary connection is through

the expected value of control (EVC) theory (Shenhav et al., 2013; Keidel et al., 2021; Frömer et al., 2021; Masís et al., 2021; Musslick and Cohen, 2021), formally explained in section 3.4, but applied specifically to learning agents as dynamical systems. Additionally, it has been proposed that the dorsal anterior cingulate cortex (dACC) plays a role in the integration and computation of most of the quantities required in EVC theory. This claim is supported by neuroimaging experiments (Botvinick et al., 2001) and is consistent with other theoretical frameworks (Botvinick and Cohen, 2014). **Using this framework, optimal solutions to the EVC problem can be efficiently computed, with the added capability of accounting for the impact of control on complex learning dynamics throughout training. This approach has the potential to facilitate comparisons with experimental data, including behavioral studies as exemplified in section 3.7 and neural recordings.**

One of the emergent solutions identified in this work is that **it is generally advantageous to allocate resources during the early stages of learning to achieve higher rewards later**. This intertemporal choice of allocating effort based on the prospect of future reward has been widely studied in psychology and neuroscience (Masís et al., 2021; Keidel et al., 2021; Frömer et al., 2021) (section A.1). An example studied within the scope of the learning effort framework is presented in (Masís et al., 2023) and explained in section 3.7, where it is hypothesized that rats regulate their reaction times to enhance learning speed in a classification task. Longer reaction times in the early stages of learning result in a lower instantaneous reward rate but accelerate performance improvements across sessions, ultimately leading to a higher cumulative reward throughout the learning process. This work provides an alternative model within the learning effort framework that qualitatively reproduces some of these findings, specifically the epistemic noise control mechanism described in subsection 3.7.2. This phenomenon resembles a learning dynamic version of the *marshmallow test* experiment (Mischel, 2014), in which children decide between consuming one marshmallow immediately or waiting 20 minutes to receive

a second marshmallow, a decision requiring self-control. A key feature of this setting is the influence of the control signal on decision-making and its impact on learning dynamics throughout the task. This introduces an additional level of abstraction related to metacognition and the agents' understanding of their own learning capabilities (Son and Sethi, 2006, 2010; Musslick et al., 2020). The concept of allocating resources at the beginning of learning to achieve higher rewards later is further discussed in subsection 4.2.3.

Another emergent solution is that **easier tasks/categories should receive more resources in the early stages of learning compared to the harder ones, which require sustained effort throughout training**. As mentioned in the main text, this strategy has been observed in cognitive science (Krueger and Dayan, 2009; Wilson et al., 2019) and machine learning (Parisi et al., 2019; Saglietti et al., 2022; Zhang et al., 2022). One possible reason for not disengaging in a task or category at any point could be to avoid catastrophic forgetting or interference between tasks/categories, since they all share the hidden layer, perhaps having a stronger interference effect in harder categories. It is hypothesized that this phenomenon can be posed as a memory replay system, and a framework is provided where this is a value-based mechanism, as indicated by other work (Mattar and Daw, 2018; Agrawal et al., 2022).

In addition, the fact that **increasing the cost of control reduces control allocation** has also been observed in cognitive science and is denoted as avoidance of cognitive demand (Kool et al., 2010; Kool and Botvinick, 2014; Westbrook and Braver, 2015). For instance, strategies that require high cognitive demand (the cost term C in the framework) will be naturally avoided even if this strategy is optimal to solve the task. However, as shown in Experiment number 6 in (Kool et al., 2010), subjects will engage in the optimal high cognitive demand strategy if they are paid to solve the task efficiently. This, in an abstract way, increases the reward in the framework (described by η in equation 3.3), which is equivalent to decreasing β. Higher engagement in control is observed when β is decreased as shown in this experiment, and it has been described in (Kool and Botvinick,

2014) as a labor-leisure trade-off. **The learning effort framework presented in this thesis has already been applied to explain cognitive fatigue, where the cost of control is not explicitly stated, but it emerges as an opportunity cost of overriding previous knowledge (Li et al., 2024)**, which has been proposed as a normative framework to explain cognitive fatigue and boredom in Agrawal et al. (2022).

In this work, specific instances of neural implementations of a control signal have been used, gain modulation (section 3.6) and error modulation (named engagement modulation in this work, section 3.5). Neuromodulators are known to be involved in high-level executive tasks such as engagement in learning (Shenhav et al., 2013, 2017; Lieder et al., 2018; Grossman et al., 2022), and some of them act as gain modulation (Lindsay et al., 2020; Ferguson and Cardin, 2020) (section A.1). Previous work has attempted to improve the learning of artificial agents using gain modulation, such as the Stroop model (Cohen et al., 1990, 2004) or attention mechanisms (Lindsay et al., 2020). Another prominent approach to neuromodulators and cognitive control is Kenji Doya's theory of neuromodulation as a meta-learning mechanism (Doya, 2002; Lee et al., 2024b), where each neuromodulator is assigned to a specific function in a reinforcement learning setting. Dopamine (Westbrook and Braver, 2016) is proposed to be the error signal in reward prediction (perhaps tasks engagement modulation $\psi_\tau(t)$), serotonin is the time scale of reward prediction (the discount factor γ in the setting), noradrenaline (Cohen et al., 1990; Aston-Jones and Cohen, 2005; Shenhav et al., 2013) controls randomness in action selection (also believed to activate different neural paths as in (Cohen et al., 2004), as the gain modulation setting), and acetylcholine (Yu and Dayan, 2005; Ren et al., 2022) controls the speed of updates (as the learning rate, which is optimized later in analytical form in section 4.3). In summary, different instances of the learning effort framework could be tested against neural recordings to validate normative theories of neuromodulators such as the one described here (Doya, 2002; Shenhav et al., 2013).

All of the work has been simulated in linear networks. While these models induce a linear

mapping from the input to the prediction, the learning dynamics are non-linear, showing behavior that resembles those from more complex non-linear networks, while providing a simpler surrogate to perform the optimization. A first approach to non-linear network control is provided (section A.6) by approximating the gradient flow of a non-linear network using a first-order Taylor expansion around the mean of the data distribution, then maximizing reward using the gain modulation model. Since the network dynamics of the non-linear network are approximately linear for small weights (and $\tanh(\cdot)$ non-linear activation function), the control obtained when optimizing expected return still speeds up learning in the non-linear network. Given the necessary equations from the learning model, further analysis can be done on, for example, linear recurrent networks, which can be used for complex decoding depending on the properties of the recurrent connections (Bondanelli and Ostojic, 2020). Closed-form non-linear network dynamics approaches such as the teacher-student settings (Goldt et al., 2019; Ye and Bors, 2022; Lee et al., 2022), and mean-field theory (Mignacco et al., 2020; Bordelon and Pehlevan, 2022; Bordelon et al., 2023) are promising directions to extend the framework to non-linear networks and reinforcement learning dynamics.

Chapter 4

Fundamentals of Optimal Control of Learning

In this chapter, the problem of controlling learning is formulated in its most fundamental terms to facilitate mathematical analysis. The discussion begins with a discrete decision-making problem characterized by well-defined learning trajectories and progresses toward a continuous control problem, where learning is modeled as a dynamical system. This approach is motivated by the significant challenge of mathematically characterizing learning dynamics in neural networks and reinforcement learning, as previously mentioned. To date, such characterization has been achieved only for a limited subset of cases, as discussed in subsection 2.1.3. Moreover, effective control of learning requires not only an understanding of learning dynamics but also an analysis of how control interventions influence the learning process, adding to the complexity of the problem.

The next section formally outlines the difficulties associated with controlling learning. These challenges arise from the need to regulate learning through its dynamical description, where control applied at each time step influences the learning trajectory at all future points. A more tractable alternative involves assuming a predefined learning curve and

subsequently adjusting the hyperparameters that shape its trajectory. An example of this approach is found in the Expected Value of Control for Learning (LEVC) framework proposed by Masís et al. (2021). In that study, a simple model is simulated to investigate the properties of optimal control, and this thesis provides mathematical explanation for the observed simulation results.

Subsequently, an extension of the LEVC model is introduced, where the effects of controlling learning are examined in the context of a single task. This scenario represents a trade-off between maximizing immediate rewards and incurring a cost to enhance future performance. This trade-off gives rise to a fundamental control strategy, described as **learn first, do later**. This principle is presented as a conjecture in section 3.2, where a mathematical justification is provided for this widely observed strategy, alongside an analysis of the challenges in determining the optimal policy even in this simplified setting. The proposed system can be viewed as a discrete counterpart of the continuous problem initially described in the learning effort framework in chapter 3.

Finally, in section 4.3, the time-continuous learning system is analyzed to explore the possibility of controlling learning dynamics, a problem that is challenging to solve in closed form. To address this, the homotopy perturbation method is implemented to solve the Hamilton-Jacobi-Bellman (HJB) equation governing the learning process. This method enables the approximation of the optimal value function and control over time, yielding a mathematical expression that facilitates analysis. Additionally, it provides a formulation for **online control that maximizes value during learning**. This approach introduces a novel analytical framework applicable to a broad range of problems involving control and meta-learning, including those described in chapter 3.

4.1 The Challenge of Controlling Learning

A general formulation of the problem of controlling learning was introduced in section 3.2 and is restated here in a slightly modified form for convenience, facilitating subsequent mathematical analysis (refer to section 3.2 for notation). The model's input-output mapping and the learning dynamics equation, governed by parameters $w(t)$, are given as follows:

$$\hat{Y} = f(X; w(t), g(t)), \quad \frac{dw(t)}{dt} = h(w(t), g(t), \mathcal{T}). \tag{4.1}$$

where the time scale τ_w is omitted for simplicity. Given the model inference process and its associated learning dynamics, the problem of controlling learning at each time step t can be reduced to maximizing the following integral:

$$V(t) = \int_0^T dt\gamma^t v(t) = \int_t^T dt\gamma^t \left[\eta(t)\mathcal{P}(t) - C(g(t))\right]. \tag{4.2}$$

Here, $\eta(t)$ is time-dependent, as the problem may specify that task performance is evaluated at a particular learning stage, typically at the end of the learning process. However, the subject can also be queried at any point during learning, meaning $\eta(t) \neq 0$ for $0 \leq t \leq T$.

A naive approach to solving this problem would involve computing the derivative of the value function in equation 4.2 with respect to the control signal, $dV/dg(t)$, which is used to compute gradient steps in equation 3.4. The optimal control could then be determined by setting this derivative to zero. Before computing the derivative, the integral is converted into a Riemann sum for simplicity, primarily to circumvent formalities related to differentiating an integral, yielding:

$$V \approx \sum_{i=0}^{N_T} \delta t\gamma^{t_i} \left[\eta\mathcal{P}(t_i) - C(g(t_i))\right] \tag{4.3}$$

such that $t_0 = 0$ and $t_{N_T} = T$, where δt is a small time constant (in practice, equal to the learning rate of the weights trained using SGD in the deep linear network). In this case, performance is a function of the loss $\langle \mathcal{L}(t) \rangle$, which depends on the parameters at each time step, $w(t)$, and the control signal, $g(t)$, from the input-output relation in equation 3.1. The parameters evolve according to the learning dynamics in equation 4.1, which depends on the control signal. By discretizing this differential equation, it can be rewritten as:

$$w(t_i) = w(t_{i-1}) + \delta t \frac{dw(t_{i-1})}{dt} \tag{4.4}$$

where the last term in the rhs depends on the control signal $g(t_i)$ as well. Given these equations, the dependencies of the parameters and loss function can be written explicitly as

$$w(t_i) = w(g(t_{i-1}), g(t_{i-2}), ..., g(t_0), w(t_0)) \tag{4.5}$$

$$\langle \mathcal{L}(t_i) \rangle = \mathcal{L}(w(t_i), g(t_i)). \tag{4.6}$$

Making these dependencies explicit in equation 4.3 and replacing $\mathcal{P}(t_i) = -\mathcal{L}(w(t_i), g(t_i))$, the gradient of the approximated integral with respect to $g(t_j)$ is given by

$$\frac{dV}{dg(t_j)} = \sum_{i=1}^{N} \delta t \gamma^{t_i} \frac{d}{dg(t_j)} \left[-\eta \mathcal{L}(w(t_i), g(t_i)) - C(g(t_i)) \right], \tag{4.7}$$

$$= \underbrace{-\delta t \gamma^{t_j} \left[\eta \frac{\partial \mathcal{L}(w(t_j), g(t_j))}{\partial g(t_j)} + \frac{\partial C(g(t_j))}{\partial g(t_j)} \right]}_{\text{instant variation}} \underbrace{- \sum_{i=j+1}^{N} \delta t \gamma^{t_i} \eta \frac{\partial \mathcal{L}(w(t_i), g(t_i))}{\partial w(t_i)} \frac{\partial w(t_i)}{\partial g(t_j)}}_{\text{future variation}}. \tag{4.8}$$

Finally, the optimal control signal is the one that satisfies

$$dV/dg(t_i) = 0 \tag{4.9}$$

for every $0 \leq t_i \leq T$. Note that the optimal $g^*(t_j)$ such that $dV/dg(t_j) = 0$ depends on every other $g^*(t_{i\neq j})$. Because the entire control signal $g(t)$ is planned at time step $t = 0$, then for $\gamma = 0$ the integral only considers the term $v(0)dt$, leading to no control. This is different from what is expected from an agent in common settings of reinforcement learning, where the agent makes a decision at each time step, and $\gamma = 0$ is equivalent to maximizing each $v(t)$ independently. In this setting, $\gamma > 0$ small is virtually equivalent to maximizing instantaneous net reward. From equation 4.8, it is direct that $\gamma \to 0$ makes the sum (future variations t_i with $i \geq j + 1$) vanish, therefore leaving the gradient only as a function of the loss and control cost at time t_j (meaning maximizing instantaneous reward rate $v(t)$). In general, the learning dynamics of w still depends on g, therefore the optimal $g^*(t_j)$ will depend on $g^*(t_i)$ with $i < j$, which can be solved using dynamic programming (Bertsekas, 2012), and can be solved in closed form under some linearity assumptions such as in the Linear Quadratic Regulator (LQR, Chacko et al. 2024).

The instantaneous variation in equation 4.8 shows the immediate effect of varying control, showing the change instantly in performance and cost of control. This instantaneous variation competes with the future variation term, which describes how the control signal is going to change the performance at all future times. This is generally a complicated problem, as it requires integration of the learning dynamics, which in most learning systems is challenging as discussed in subsection 2.1.3, while keeping track of the effect of control on these dynamics. As shown later, even if the learning dynamics are linear, interesting control signals of learning will have a non-linear relationship with the parameters/state of the system $w(t)$. One way to avoid this issue is by simply assuming learning curves here denoted as $a(t)$, which are closed-form time signals that give performance throughout training, without the assumption of learning dynamics on their parameters, hence bypassing the complex dependency on the control signal, and replacing it with simpler surrogates.

4.2 A Principle of Optimal Control of Learning

As mentioned in the previous section, complex learning dynamics and the effect of control on it resists mathematical analysis. A way to alleviate this problem is by assuming a performance curve $a(t)$ with t indexing learning time, in the optimization objective. This would be describing $a(t) = \eta\mathcal{P}(t)$. This approach has been taken by other researchers, such as in optimal allocation of time to different learning tasks (Son and Sethi, 2006, 2010) and the expected value of control of learning theory (Masís et al., 2021). From this work, two key ideas are taken:

- The authors Son and Sethi (2006, 2010) argue that well-behaved learning curves $a(t)$ are shaped by **sigmoid or exponential** functions (or a mix of these), based on the assumption that learning curves are **monotonic** (learning always improves with allocated time) and **bounded** (there is a maximum achievable performance). The authors are able to solve for optimal time allocation on tasks with different learnabilities based on these assumptions.

- In the expected value of control of learning by Masís et al. (2021), a simple example of two tasks is simulated as an instance of a problem of control allocation for learning. In the simulations, the authors found that there are mainly two optimal learning strategies: either stick with a learnable task for all the available time, or simply harvest available reward without learning.

By mixing these two contributions and assumptions, it is possible to explain the simulation findings by Masís et al. (2021), and in addition, show a subset of general control strategies observed in most numerical simulations in the learning effort framework applications from chapter 3, which reflects an **intertemporal choice of control allocation** given the assumptions of monotonicity of learning curves.

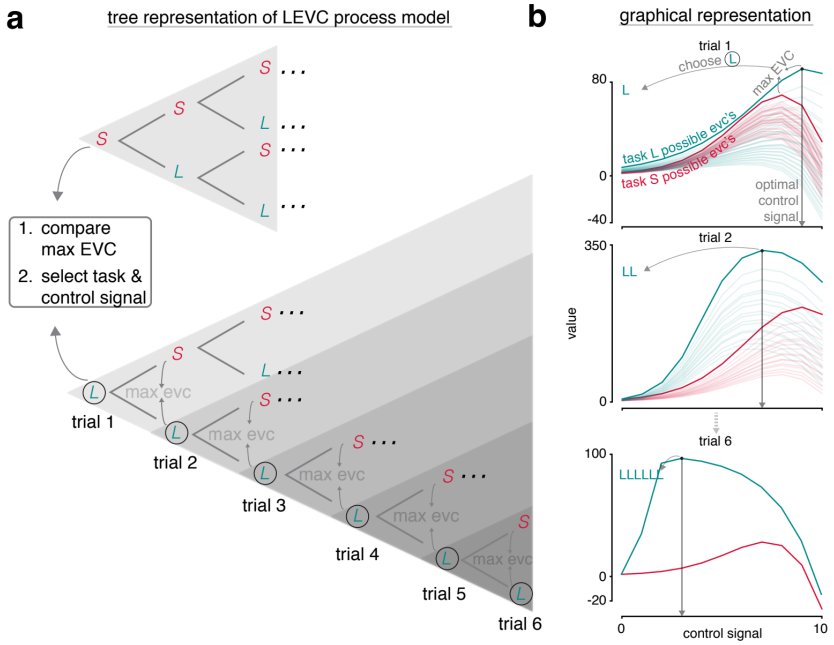

Figure 4.1: (a) At each time step, the agent could choose between the static task and the learnable task by comparing the EVC value, the trajectories made by the decisions form a tree. (b) EVC as a function of the control signal for different trajectories. **Figure taken from Masís et al. (2021)**

4.2.1 EVC for Learning

In this extension of the EVC theory, it is proposed to account for an evolving level of competence (learning) of an agent executing a task. A specific instance of a learning paradigm is used to explain the features of this extension. At each time step, an agent is able to choose the execution between two tasks. One of them is static (task S), meaning this task cannot be improved upon when being executed, and the other task is learnable (task L). The performance on task L can be improved with experience, and the learning is described as increasing the automaticity (equation 6 in Masís et al. (2021), here named u) on performing the task, which is written as

$$u_L = \alpha_L n_{\text{trials } L} + u_{0_L}, \tag{4.10}$$

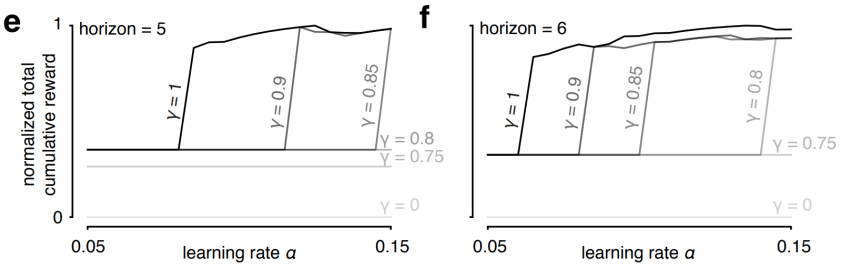

Figure 4.2: LEVC for different values of task horizon, learning rates α_L and γ. **Figure taken from Masís et al. (2021)**

where u_{0_L} is the initial ability, $n_{\text{trials } L}$ is the number of trials already taken on task L, and α_L is the learning rate on task L. Then, the accuracy on task L, a_L, is defined as a sigmoid function

$$a_L = \frac{1}{1 + \exp\left(-r_L\left(g - \frac{b}{r_L}\right)\right)}, \quad r_L = \frac{u_L}{D_f} \tag{4.11}$$

with r_L being the ratio between the automaticity on task u_L and the task difficulty D_f, g the control signal, and b a fixed bias term. Increasing automaticity on the task reduces the amount of control needed to achieve the same accuracy.

In this simple setup, the control of choosing either task L or S must now include the learning trajectory of L. This could help to explain the *effort paradox* observed in experiments, where subjects decide to take on tasks that are more cognitively demanding despite having equal rewards (Cacioppo and Petty, 1982; Inzlicht et al., 2018). The specific influence of learning on future values removes the need for an intrinsic value of control or learning, instead being explicitly about future improved rewards due to control and learning.

To obtain the optimal control policy that maximizes value while considering learning, the authors simulated all possible paths at each time step, then chose the task S or L that had better accumulated rewards from the simulated paths. The authors found through

numerical simulations that the **optimal strategies in this setting were either to only execute the learnable task L or to only execute the static task S.** In addition, the authors show how the decision of choosing either the learnable task or the static task depends on the discount factor and learning rates. For higher learning rates, the value of choosing the learnable task increases as rewards from better performance can be collected earlier. Similarly, the value of the learnable task increases with higher γ as future rewards are more relevant. A similar effect can be found when increasing the time horizon to learn the task (see figure 4.2).

4.2.2 Optimal Control Identity of EVC for Learning

In this section, a theoretical explanation is provided that shows why the optimal strategies for the EVC model for learning presented by (Masís et al., 2021) are either sticking with the learnable task L or the static task S. Based on this derivation, a similar setup is presented to show that it is generally optimal to exert effort at the start of learning to harvest the reward coming from the improved performance later.

For the static task S, its accuracy $a_S(t)$ is described as in equation 4.11, except that the automaticity u_S of the static task is constant as it does not change with the number of trials when task S is chosen. The accuracy $a_S(t)$ is also subject to the control signal g. Hence, the control decision is twofold: the **identity** of the task to solve, here denoted as *decision* $d = L$ or $d = S$, and the **intensity** of the control g for the task chosen.

For the learning task L, since the learning curve is defined as a sigmoid, it is easy to show

that

$$a_L(n,g) = \frac{1}{1+\exp\left(-\frac{u_L(n)}{D_f}g+b\right)}, \tag{4.12}$$

$$> \frac{1}{1+\exp\left(-\frac{u_L(n-1)}{D_f}g+b\right)}, \tag{4.13}$$

$$= a_L(n-1,g), \tag{4.14}$$

meaning that for the same amount of control g, the accuracy, and therefore the reward from the learnable task after engaging from one step, will be larger. In a similar way, it can be shown that

$$a_L(n,g_n) = a_L(n-1,g_{n-1}) \implies g_n < g_{n-1}, \tag{4.15}$$

this is, to achieve the same accuracy one step later compared to the previous step, the control needed is smaller. This is the role of automaticity, as increasing automaticity through learning reduces the amount of control needed to perform the task. In this formulation, the amount of steps left to pick tasks is denoted by N.

The optimal agent should decide either $d = S$ or $d = L$ based on its value function which here written as

$$Q(d=S,n,N) \quad \text{and} \quad Q(d=L,n,N), \tag{4.16}$$

where n is the number of trials in the learnable task, and N is the number of steps left to pick tasks. Now each of the choices is assumed at the last step $N = 1$, and studied the consequences for the previous choice in backwards manner. First, note that at the last

decision step

$$Q(d = S, n, N = 1) = a_S(g_S) - C(g_S) \quad \text{and} \tag{4.17}$$

$$Q(d = L, n, N = 1) = a_L(n, g_{L,n}) - C(g_{L,n}). \tag{4.18}$$

Choosing $d = S$ implies that $Q(d = S, n, N = 1) > Q(d = L, n, N = 1)$, then the previous choice at $N = 2$ is governed by

$$Q(d = S, n, N = 2) = a_S(g_S) - C(g_S) + \gamma Q(d = S, n, N = 1), \tag{4.19}$$

$$Q(d = L, n - 1, N = 2) = a_L(n - 1, g_{L,n-1}) - C(g_{L,n-1}) + \gamma Q(d = S, n, N = 1). \tag{4.20}$$

Then in step $N = 2$, the optimal decision is $d = L$ if

$$a_S(g_S) - C(g_S) \leq a_L(n - 1, g_{L,n-1}) - C(g_{L,n-1}), \tag{4.21}$$

but this is a contradiction as it is assumed in the next decision $N = 1$ that $d = S$ and $Q(d = S, n, N = 1) > Q(d = L, n, N = 1)$, meaning

$$a_L(n, g_{L,n}) - C(g_{L,n}) \leq a_S(g_S) - C(g_S) \leq a_L(n - 1, g_{L,n-1}) - C(g_{L,n-1}), \tag{4.22}$$

which is not possible since

$$a_L(n, g_{L,n}) - C(g_{L,n}) \geq a_L(n - 1, g_{L,n-1}) - C(g_{L,n-1}) \tag{4.23}$$

as previously noted in equations 4.14 showing that accuracy increases for larger n and equation 4.15 showing that the optimal g decreases therefore the cost decreases as well. This means choosing $d = L$ a step before $d = S$ is a contradiction. A symmetric argument can be done show it is suboptimal to choose $d = S$ and then switch to $d = L$ in the next step. **Hence, the optimal strategy is either sticking with the learnable task** L

or sticking with the static task S. Note that this contradiction applies recursively to any time step before in step left for learning $N = 1$, then it applies generally for any other step.

Given the optimal strategies, it is possible to write, for instance, the value of the static task

$$Q(d = S, n, N) = \sum_{k=0}^{N} \gamma^k \left(a_S(g_S) - C(g_S)\right) \tag{4.24}$$

$$= \left(a_S(g_S) - C(g_S)\right) \frac{1 - \gamma^{N+1}}{1 - \gamma} \tag{4.25}$$

meaning the optimal signal g^* can be computed *greedily* by solving

$$\frac{da_S(g)}{dg} = \frac{dC(g)}{dg}. \tag{4.26}$$

This means that the intensity or the use of control does not affect the trajectory of learning. This is also true even for the learning task, since the learning curve a_L improves depending on n and not on g. Hence for a given trajectory for $a_L(n = k_1, g_{k_1})$, $a_L(n = k_2, g_{k_2})$ ..., the optimal g at each step only depends on the current accuracy and cost as in previous equation 4.26, as g does not change the trajectory.

4.2.3 Conjecture: Learn First, Do Later

When the control signal does affect future trajectories, the calculation for the optimal control is no longer greedy, as in the previous case, but rather resembles the problem expressed in section 4.1, more specifically in the decomposition of the gradient of the value function in equation 4.8, separating the instantaneous effect of control versus the future effect on dynamics. As noted in the numerical simulations in chapter 3, solving the control signal in closed form when the control affects the entire future trajectory is challenging, and this can be solved numerically using gradient methods like in the proposed learning

effort framework, or later using perturbation methods shown in subsection 4.3.3. However, in simple settings, it is possible to recognize general features of the optimal control of learning, simply by studying the optimality of trajectories, aided also by numerical optimization.

Next, the abstract model to depict the EVC for learning theory is extended to a case where the control affects the learning dynamics of an agent. In the EVC for learning described in subsection 4.2.2, the number of trials executing the learning task would modify the performance on that task, while keeping the unlearnable task constant, therefore producing the two strategies found: only engaging in the learning task L or the static task S, but with no influence by the control signal.

In this extended model, the agent faces a single task, and the decision is either to engage in **Active Learning**, which accelerates the progress in performance (learning) in the single task by incurring a cost, or to disengage and incur a **Passive Engagement** strategy, where task performance improves slower (or not at all) with no cost. The amount of control then is how much of the available time is allocated to **Active** versus **Passive** engagement. This approach makes the control signal affect the future performance of both options, which is fundamentally different from Son and Sethi (2006, 2010); Masís et al. (2021) calculations because in their case, the control changes the value of the chosen task and keeps the rest of the tasks with the same performance as those have not been engaged, allowing for extended mathematical analysis.

The phenomena this extended model attempts to describe is the situation where an agent faces a new task and needs to decide if it is worth engaging in learning, and how much effort throughout learning needs to be put into this task. It might be the case that, when the available time is not enough, or the cost of engaging in learning is too high, it might not be worth even learning the task. Perhaps collecting a reward passively (or perhaps collecting no reward at all) could be the optimal strategy. If it is worth engaging in

learning, then it would make sense to put effort at the start to have more time to collect reward without the cost of actively engaging in learning. This has been observed in many of the simulations in chapter 3. It has been hinted at by Son and Sethi (2006, 2010) in their own simplified setting, and it is aligned with the idea of controlled to automatic behavior when learning a task (Bogacz et al., 2006; Foroughi et al., 2017; Masís et al., 2023; Masis et al., 2024).

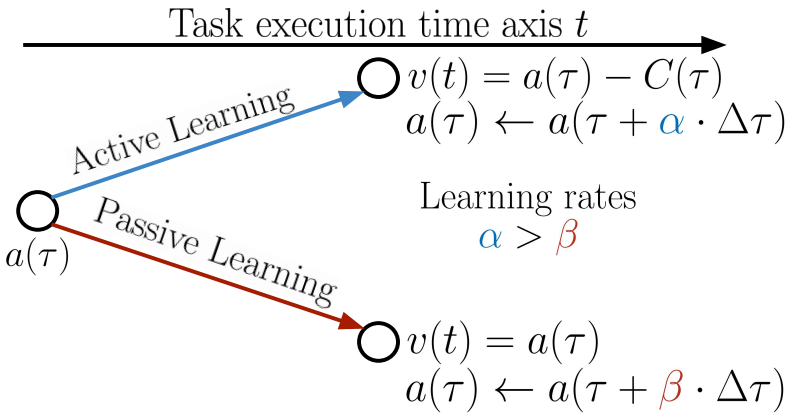

Figure 4.3: Active vs Passive learning problem. At each time step, the agents picks either to actively learn vs passively learn. When learning actively, learning rate is higher but incurs in a cost, when learning passively, learning rate with no cost ($\alpha > \beta$).

Formally, the active versus passive learning model can be stated as follows: An agent is facing a learning task described by a learning curve $a(\tau)$, where τ specifies how much progress in the task has been made. At the start of learning, the performance of the agent is denoted by $a(\tau_0)$. From the start, at each time step t, the agent can either engage in active learning or passive learning. Both forms of learning improve performance, but at different rates. When engaging in active learning, the argument of the learning curves is increased by $\alpha \cdot \Delta\tau$, and $\beta \cdot \Delta\tau$ for passive learning, where the learning rate is higher for active learning than passive learning $\alpha > \beta$, and $\Delta\tau$ denotes a small amount of learning progress. In addition, at each step the agent collects a reward of $v(t) = a(\tau) - C(\tau)$ if active learning is selected, where $C(\tau)$ denotes the cost of engaging in active learning. On the other hand, the reward for passive learning is simply the current performance

$a(\tau)$ with no cost. The agent then needs to plan when to engage in active learning (See figure 4.3), such that it maximizes the cumulative reward throughout learning, denoted by

$$V = \sum_{i=0}^{N-1} v(t_i). \tag{4.27}$$

To study the optimal policy, consider first a large constant cost C_P such that the optimal policy is simply engaging in passive learning throughout the entire time. Then,

$$V^* = \sum_{i=0}^{N-1} a(\tau_0 + i \cdot \beta\Delta\tau). \tag{4.28}$$

Now, consider decreasing C_P until only one step of active learning is worth performing C_A. Assuming that the learning curve is a monotonically increasing function (which is an abstract ideal feature of learning as indicated by Ritter and Schooler 2001; Son and Sethi 2006, 2010), meaning $a(\tau_1) \leq a(\tau_2)$ if $\tau_1 \leq \tau_2$, then it is easy to see that the active learning step must be executed at the start of learning, obtaining the following optimal value function:

$$V^* = a(\tau_0) - C + \sum_{i=1}^{N-1} a(\tau_0 + \alpha\Delta\tau + (i-1) \cdot \beta\Delta\tau). \tag{4.29}$$

This is because, if the active learning step is applied at any other point in time T_A

$$V = a(\tau_0) - C + \sum_{i=1}^{T_A-1} a(\tau_0 + i \cdot \beta\Delta\tau) + \sum_{i=T_A}^{N-1} a(\tau_0 + (i \cdot \beta + \alpha)\Delta\tau). \tag{4.30}$$

Splitting the sum in V^* at the same time point

$$V^* = a(\tau_0) - C + \sum_{i=1}^{T_A-1} a(\tau_0 + (\alpha + (i-1)\cdot\beta)\Delta\tau) + \sum_{i=T_A}^{N-1} a(\tau_0 + (\alpha + i\cdot\beta)\Delta\tau) \quad (4.31)$$

$$> a(\tau_0) - C + \sum_{i=1}^{T_A-1} a(\tau_0 + i\cdot\beta\Delta\tau) + \sum_{i=T_A}^{N-1} a(\tau_0 + (i\cdot\beta + \alpha)\Delta\tau) \quad (4.32)$$

$$= V \quad (4.33)$$

where the inequality is given by the monotonicity of $a(\tau)$ and that $\alpha > \beta$ in the first sum, showing that $V^* > V$, hence **the optimal policy is engaging in active learning as early as possible**. Consider decreasing C_A further such that two steps of active learning are now worth it, a similar argument applies concentrating the second active learning step after the first one. For a given number of optimal active learning steps N_A, the optimal policy becomes

$$V^* = -C\cdot N_A + \sum_{i=0}^{N_A-1} a(\tau_0 + i\cdot\alpha\Delta\tau) + \sum_{i=N_A}^{N} a(\tau_0 + N_A\alpha\Delta\tau + i\cdot\beta\Delta\tau). \quad (4.34)$$

For a given cost C, the problem becomes finding the optimal N_A, that is, the amount of time spent engaging in active learning. From this value function, it is easy to see how the amount of time on active learning influences the reward collected during passive learning, expressing the same problem of control allocation described in section 4.1. This problem is simpler to study in the continuous limit of learning, then maximizing the value by taking its derivative with respect to the point in time to switch from active to passive learning, here named δ. The continuous-time version of the same problem can be written as

$$V(\delta) = \int_0^{\delta} a(\alpha t + \tau_0)dt - \delta C + \int_{\delta}^{T} a(\beta t + \delta\alpha + \tau_0)dt. \quad (4.35)$$

Using the Leibniz Integral Rule, the derivative of the value function with respect to δ is

$$\frac{dV}{d\delta} = 0 = a(\alpha t + \tau_0) - a(\delta(a+\beta) + \tau_0) - C \\ + \frac{\alpha}{\beta}\left[a(\beta T + \delta\alpha + \tau_0) - a(\delta(\alpha+\beta) + \tau_0)\right]. \tag{4.36}$$

For non-linear learning curves or further including other aspects such as the time discount factor γ or control dependent cost $C(\delta)$, this equation becomes quite challenging and perhaps unsolvable[1]. The *Learn First, Do Later* conjecture is numerically verified with numerical simulations in subsection 4.2.4.

The proposed model is simple enough that it allows for a basic understanding of the rough shape of optimal learning control. To move to a continuous-time control setting, with a continuous amount of control, consider now $N_{\mathcal{T}}$ possible engagement levels. At each step, the agent needs to pick a learning rate corresponding to each strategy; it is no longer only α and β from active and passive learning, but rather a set of strategies with a continuous level of learning rate $g(t)$. If at each time step t a different engagement level $g(t)$ is assigned, then the original problem presented in equation 4.35 becomes

$$V_g = \int_0^T \left[a\left(\tau_0 + \int_0^t g(t')dt'\right) - C(g(t))\right] dt, \tag{4.37}$$

which resembles closely the problem of optimal learning rate scheduling presented in section 4.3, as the learning rate affects the entire learning dynamics when integrated to compute the value function. It is not possible to verify the conjecture for the last expression, but from numerical simulations in chapter 3 and further analytical methods presented in section 4.3, the empirical optimal control seems to hold generally. This connects the discrete binary decision problem to the more complex continuous control problem further studied in the next sections.

[1] I believe for some simple learning functions, e.g. $a(t) = r\left(1 - \frac{1}{t}\right)$ this equation might be solvable.

4.2.4 Numerical Verifications

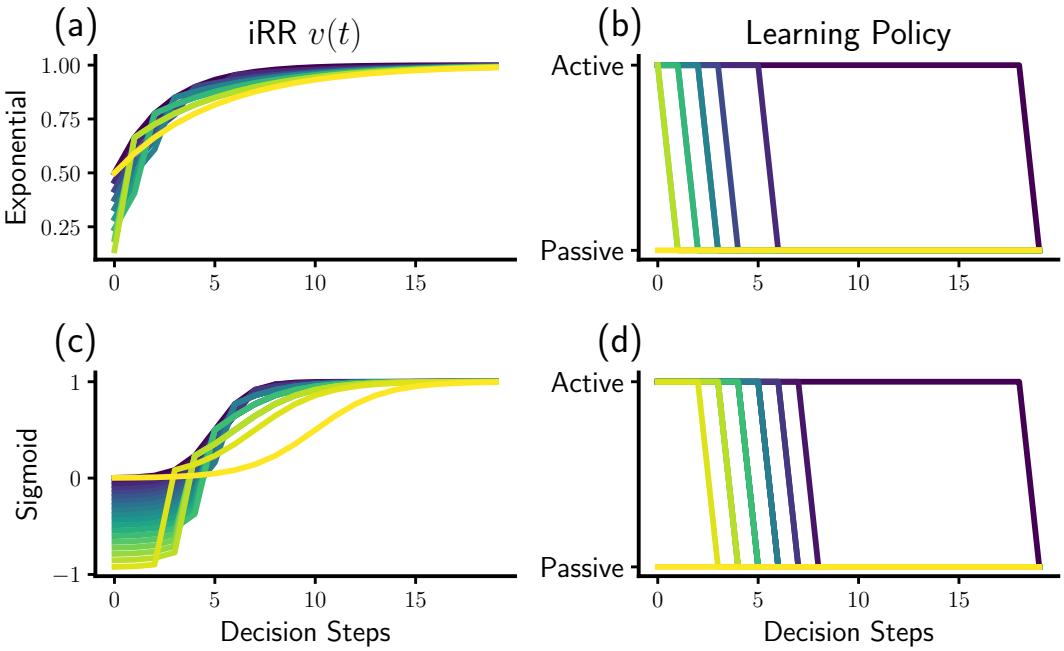

Figure 4.4: Numerical Verification of Learning First conjecture. **(a) and (c)**: Instant reward rate $v(t)$ for exponential and sigmoid learning curve for the optimal policy with increasing cost of control C (low to high cost from blue to yellow). **(b) and (d)**: Brute force search optimal policy of active vs passive learning. All of them engage in active learning first, then switch to passive learning, or simply not engage at all in the task.

To numerically verify the conjecture, a model with 20 steps to choose between active and passive learning was simulated for two learning trajectories that are monotonically increasing functions of learning time, as suggested by Son and Sethi (2006, 2010). These two learning curves are an exponential improvement and a sigmoid (see figure 4.4). To verify the conjecture numerically, all possible combinations of decisions of active and passive learning were simulated for the 20 steps (meaning 2^{20} possible trajectories). Then, gradually decreasing the cost C (higher cost from blue to yellow in figure 4.4) would also gradually increase the amount of control. Without constraining the structure of the optimal policy, all of them were fully engaged in active learning at the start and then switched to passive learning until C was large enough so that it disengaged from active

learning completely.

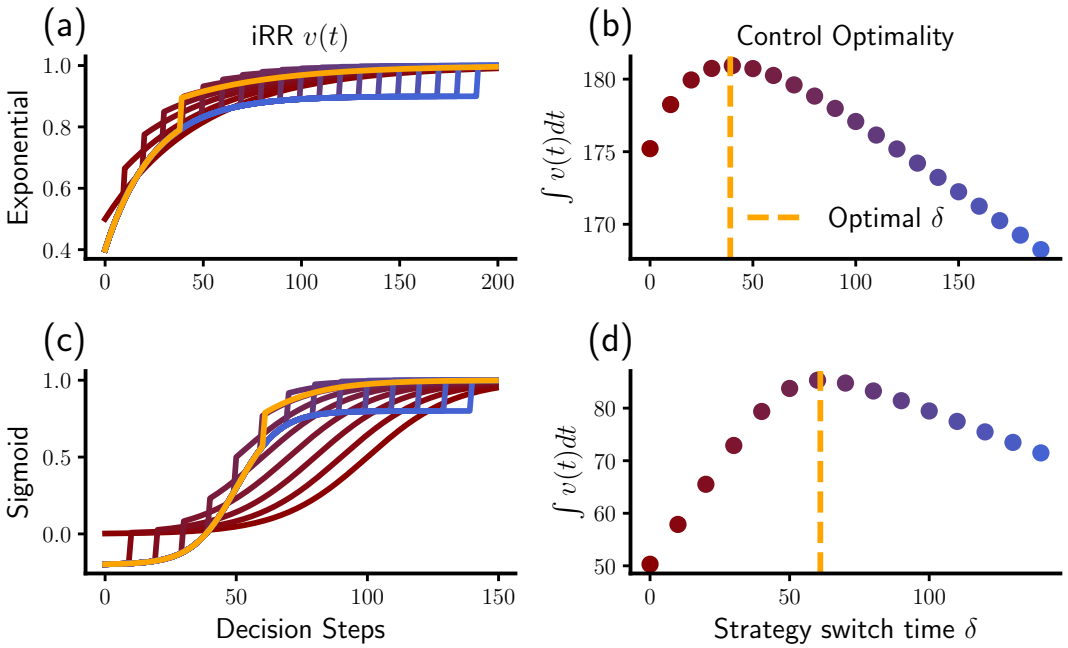

Figure 4.5: Optimal Switch time between Active and Passive Learning. **(a) and (c)**: Instant reward rate $v(t)$ for varying strategy switch time δ, for exponential and sigmoid learning curve respectively. **(b) and (d)**: Cumulative reward $\int v(t)dt$ for every value of strategy switch time δ.

For a fixed control cost C, it is possible to look for the strategy switch time, from active learning to passive learning, within the trajectories that satisfy the conjecture. The results for both types of learning curves show that finding the switching time δ is a concave maximization problem (or convex minimization), where there is an unequivocal optimal switch time. Too early of a switch will not allow the agent to learn enough, and too late of a switch will not improve performance enough to pay off the cost of investing in active learning.

4.3 Analytical Approximation of Optimal Control

The work presented in this section is primarily based on *An Analytical Theory of Cognitive Control of Learning*, presented as a contributed talk at COSYNE 2025 [2], by Valentina Njaradi*, Rodrigo Carrasco-Davis* and Andrew Saxe, (Thanks to Javier Masís and Peter Latham for their useful input). Results have been extended from the abstract to conform to the thesis.

The problem of optimal control that maximizes cumulative reward, solved numerically in chapter 3, is restated here and approached in a simplified manner to find an analytical solution to the optimal control, previously only achievable through numerical simulations. In this section, the assumption of a learning curve used in section 4.2 is dropped; now, the focus is to work directly with the learning model as a dynamical system, which parameters evolve through backpropagation and are influenced by the control signal. The target is also changed to *minimizing cumulative risk through learning*, which is equivalent to maximizing reward (simply by a change of sign). Here, the formal optimization problem is stated.

Given a learning system that faces a new task $\mathcal{T}$ that can be learned within a period of time $0 \leq t \leq T$, with parameters W that evolve according to some dynamical system characterized by h and with initial parameters W_0, where performance is measured based on a loss function L (e.g., mean square error). The instantaneous risk is discounted by γ, the control signal is denoted as $g(t)$, and deviations from a default value of control will incur a cost $C(g)$. Then, the optimal control signal $g^*(t)$ is the one that minimizes the following cumulative risk $R(g)$:

$$\min_g R(g) = \int_0^T dt\gamma^t \left[L(W(t), g(t)) + C(g(t))\right], \tag{4.38}$$

$$\text{s.t. } \frac{W(t)}{dt} = h(g(t), W(t), t), \text{ and } W(0) = W_0. \tag{4.39}$$

[2] https://www.world-wide.org/cosyne-25/analytical-theory-cognitive-control-a57c6e06/

This is analogous to the problem stated in the learning effort framework introduced in section 3.2. Most of the methods to solve this problem rely on numerical simulations, and methods that allow for mathematical analysis need linearity assumptions, e.g. LQR (Chacko et al., 2024).

To approximate this solution, a relatively standard process (at least in control theory) is applied to the control of a learning system. First, the Hamilton-Jacobi-Bellman equation is derived from the optimization problem, which is a partial differential equation (PDE) describing the evolution of cumulative risk throughout time, denoted by $R(W,t)$, under the optimal control $g^*(t)$. An expression of the optimal control as a function of the cumulative risk is also provided. This PDE is non-linear, and for interesting non-linear cases, it does not allow for an analytical solution. To solve this equation, the solution is approximated using the Homotopy Perturbation Method (Atangana et al., 2014), described in the following sections. This solution allows for an approximation of the optimal control $g^*(t)$ as a function of the state parameters $W(t)$, in principle allowing for a closed-loop online control.

4.3.1 Hamilton-Jacobi-Bellman Equation of Learning

The optimal cumulated risk function $R(W,t)$ is defined as

$$R(W,t) = \min_g \left\{ \int_t^T d\tau \gamma^{\tau - t} \left[L(W(\tau), g(\tau)) + C(g(\tau)) \right] \right\} \tag{4.40}$$

where the weights change according the dynamics defined in equation 4.39. This expression can be defined recursively, similar to a standard Bellman equation in RL, as follows:

$$R(W,t) = \min_g \left\{ \int_t^{t+\Delta t} ...d\tau + \int_{t+\Delta t}^T ...d\tau \right\}, \tag{4.41}$$

$$= \min_g \left\{ \int_t^{t+\Delta t} ...d\tau + \gamma^{\Delta t} R(W(t+\Delta t), t+\Delta t) \right\}. \tag{4.42}$$

From here, assuming Δt small, it is possible to Taylor expand the last term giving

$$R(W(t+\Delta t), t+\Delta t) = R(W,t) + \frac{\partial R}{\partial W}\frac{dW}{dt}\Delta t + \frac{\partial R}{\partial t}\Delta t. \tag{4.43}$$

Replacing this in the equation above

$$R(W,t) = \min_g \left\{ \int_t^{t+\Delta t} ...d\tau + \gamma^{\Delta t}\left(R(W,t) + \frac{\partial R}{\partial W}\frac{dW}{dt}\Delta t + \frac{\partial R}{\partial t}\Delta t \right) \right\}, \tag{4.44}$$

$$\frac{R(W,t) - \gamma^{\Delta t} R(W,t)}{\Delta t} = \min_g \left\{ \int_t^{t+\Delta t} ...d\tau + \frac{\partial R}{\partial W}\frac{dW}{dt} + \frac{\partial R}{\partial t} \right\}, \tag{4.45}$$

then taking the limit of $\Delta t \to 0$, the limit on the left side of the equation becomes $-\log(\gamma)$, the integral on the right side with the infinitesimal limits becomes the argument evaluated at t, and taking the time derivative of the cumulated risk out of the min as it does not depend on the control, gives

$$-\frac{\partial R}{\partial t} = \log(\gamma) R(W,t) + \min_g \left\{ L(W(t), g(t)) + C(g(t)) + \frac{\partial R}{\partial W}\frac{dW}{dt} \right\} \tag{4.46}$$

$$\text{s.t. } R(w,T) = 0. \tag{4.47}$$

This equation is called the **Hamilton-Jacobi-Bellman equation**, which is a partial differential equation in time and state parameters. To proceed from this equation, the argument of the $\min(\cdot)$ operator, which is called **the Hamiltonian**, needs to be minimized by solving for the optimal control that minimizes this Hamiltonian. The boundary condition $R(w,T) = 0$ is imposed as it is the time limit to learn the task, and no risk is collected after time T. Depending on the loss L, cost function C, and learning dynamics dW/dt, the control that minimizes the Hamiltonian might be solvable in closed form, and replacing this value back into the equation gives a PDE purely in terms of the cumulative risk, which could be solved in some cases. For instance, the Riccati equation

used in the LQR controller is the resulting HJB equation when assuming a quadratic cumulative risk and linear influence of control over the learning dynamics (Chacko et al., 2024). In the next sections, the dynamics, loss function, and control cost are instantiated with a few examples to solve for the optimal control and obtain a PDE explicitly in terms of the cumulative risk, to then solve it using the Homotopy Perturbation Method.

4.3.2 Optimal Learning Rate Scheduling

Here, the task to be solved by the learning system, the loss function, the learning dynamics, and the cost of control are instantiated to proceed with the mathematical analysis. In this case, the control will describe the optimal learning rate as a function of time.

The task is simply a gaussian discrimination task as the one used in the single neuron example described in subsection 3.2.1, here re-stated for convenience. A dataset of examples $i = 1, \cdots, P$ is drawn as follows: A label y_i is first sampled as either $+1$ or -1 with probability $1/2$. The input x_i is then sampled from a Gaussian $x_i \sim \mathcal{N}(y_i \cdot \mu_x, \sigma_x^2)$. The task is to predict y_i from the value of x_i. The intrinsic difficulty of the task is controlled by how much the Gaussians overlap, controlled by the relative value of μ_x and σ_x.

For simplicity, the learning agent used here will be a single-neuron network; however, the results presented here apply to a multi-dimensional input and output of a single-layer network. The prediction from this single neuron model is defined by $\hat{y} = wx$, with w being the neuron weight, in this case, the state parameter of the dynamical system. The loss function is simply the MSE between the target and prediction, and the dynamics of the weight are defined by gradient descent. The updates are regulated by the learning rate, which in this setup is the control signal, and the control cost is a square penalty of

non-zero control values. Next, the learning model is stated formally,

$$\min_g R(g) = \int_0^T dt\gamma^t \left[L(w(t)) + C(g(t))\right], \tag{4.48}$$

$$\text{s.t. } \frac{w(t)}{dt} = -(\alpha + g(t)) \left\langle \frac{dL}{dw} \right\rangle_{xy}, W(0) = W_0, \tag{4.49}$$

$$L(w(t)) = \frac{1}{2} \left\langle (y - \hat{y})^2 \right\rangle_{xy}, \text{ and } C(g(t)) = \frac{\beta}{2} g^2(t). \tag{4.50}$$

where α is a baseline learning rate, and β is a coefficient regulating the cost of increasing control. Note that while the learning dynamics are linear in the state space, the effect of the control signal (learning rate) has a non-linear effect on the learning dynamics, making LQR not applicable in this setup. This non-linearity is given by the product of the control signal over the state variable in the gradient flow dynamics,

$$\frac{dw}{dt} = -(\alpha + g) \left\langle \frac{dL}{dw} \right\rangle_{xy}, \tag{4.51}$$

$$= (\alpha + g) \left(\mu_x - w(\mu_x^2 + \sigma_x^2)\right), \tag{4.52}$$

$$= (\alpha + g) \left(\mu_x - w\xi^2\right). \tag{4.53}$$

The loss function in this case does not depend on the control signal, so it can be removed from the $\min(\cdot)$ operator in the HJB-equation, and replacing dW/dt with this specific learning dynamics and cost function the equation (dropping the time dependency for simplicity) becomes

$$-\frac{\partial R}{\partial t} = \log(\gamma) R(w,t) + L(w) + \min_g \underbrace{\left\{ \frac{\beta}{2} g^2(t) - (\alpha + g) \frac{\partial R}{\partial w} \left\langle \frac{dL}{dw} \right\rangle_{xy} \right\}}_{H}. \tag{4.54}$$

In this particular form of the Hamiltonian H, it is possible to find the control that

minimizes the Hamiltonian by

$$0 = \frac{dH}{dg}, \tag{4.55}$$

$$0 = \beta g - \frac{\partial R}{\partial w} \left\langle \frac{dL}{dw} \right\rangle_{xy}, \tag{4.56}$$

$$g^* = \frac{1}{\beta} \frac{\partial R}{\partial w} \left\langle \frac{dL}{dw} \right\rangle_{xy}. \tag{4.57}$$

This control signal is a minimum (hence the problem being convex on the control signal) due to

$$\frac{d^2 H}{dg^2} = \beta > 0. \tag{4.58}$$

This convexity might be related to the optimal strategy switch discussed in subsection 4.2.4, specifically connected through equation 4.37, but it requires further analysis. Replacing the optimal control signal in the HJB-equation

$$\log(\gamma) R(w,t) + \frac{\partial R}{\partial t} + L(w) - \alpha \frac{\partial R}{\partial w} \left\langle \frac{dL}{dw} \right\rangle - \frac{1}{2\beta} \left(\frac{\partial R}{\partial w} \left\langle \frac{dL}{dw} \right\rangle \right)^2 = 0 \tag{4.59}$$

This is a non-linear PDE on the cumulative risk $R(w,t)$. Solving this equation would allow for a solution to the optimal control problem through equation 4.57.

4.3.3 Homotopy Perturbation Method

The solution to the HJB equation for optimal learning rate scheduling can be solved using the homotopy perturbation method (HPM) (He, 1999; Liao, 2003; Roy and Maiti, 2023), which was previously applied to HJB equations in Atangana et al. (2014), although not applied to learning systems yet. Here, a simplified version of the HPM is described as used to solve the HJB equation.

A homotopy G is a continuous function that describes a *continuous deformation* or

interpolation between two functions f and h. For instance, a linear homotopy is given by

$$G : (x, p) \to (1 - p)f(x) + ph(x). \tag{4.60}$$

The parameter p is called the *embedding parameter*, and it controls how far the interpolation is from each of the functions. To solve complex equations using a homotopy, first, a homotopy is constructed between the equation that needs to be solved and a simple equation or initial guess of the solution. Then, the solution to the problem is defined as an infinite polynomial over the embedding parameter p, and the coefficients of this polynomial that describe the solution are solved for. More concretely, assuming a non-linear partial differential equation

$$\frac{df}{dt} = F(f, x, t, df/dx), \tag{4.61}$$

a homotopy is constructed, interpolating between the equation that needs to be solved and an *initial guess function* f_g

$$(1 - p)\left(\frac{df}{dt} - \frac{df_g}{dt}\right) + p\left(\frac{df}{dt} - F(f, x, t, df/dx)\right) = 0. \tag{4.62}$$

Next, the solution to this equation is defined as a polynomial over the embedding parameter

$$f(x, t, p) = \sum_{i=0}^{\infty} p^i f_i(x, t). \tag{4.63}$$

where the functions f_i are called *modes*. Traversing the homotopy from $p = 0$ to $p = 1$ moves the solution between the guess function and the actual equation. As $p \to 1$, higher-order modes start to become relevant to the sum in the solution $f(x, t)$, increasing the complexity of the polynomial as f approaches the solution to the equation. In general, the infinite sum cannot be written in closed form, and a finite number of modes are computed to approximate a solution. To find each mode, the polynomial is replaced in the

homotopy equation, and the coefficients are solved recursively by equating coefficients of each power of p, starting from mode p^0, obtaining $f_0 = f_g$, then solving for the coefficient p^1, and so on for the rest of the coefficients (shown concretely later in subsection 4.3.4 when solving for the optimal learning rate). Naturally, adding more modes to the sum increases the quality of the approximation, and some convergence guarantees have been provided analytically in some cases (Turkyilmazoglu, 2010; Ayati and Biazar, 2015); however, the applicability of this method to PDEs in general is mostly based on trial and error. Using this method, it is possible to approximate the solution to the optimal learning rate scheduling described in equation 4.59, where the modes are functions of the statistics of the task and hyperparameters of the model, such as the discount factor γ or cost of control β.

4.3.3.1 Padé Approximation of Homotopy Modes

To further improve the approximation capabilities of the HPM, an extra step is included to express the sum over the modes as a Padé Approximation. This converts the polynomial over the embedding parameter p (which resembles a Taylor expansion) and converts it to a ratio of polynomials. It has been argued that the Padé Approximation is overall better than a Taylor expansion for highly non-linear functions and can maintain the approximation beyond the convergence radius of the Taylor series (Apresyan, 1979), and it has recently been shown effective when solving HJB equations (Ganjefar and Rezaei, 2016). Here is how the approximation is computed.

Given a function $f(x)$, a Padé approximation is a fraction of polynomials of the form

$$\text{Pade}_{[M/N]} = \frac{a_0 + a_1 x + a_2 x^2 + ... + a_N x^N}{1 + b_0 + b_1 x + b_2 x^2 + ... + b_M x^M}, \tag{4.64}$$

and the coefficients a_i and b_i computed by enforcing the derivative condition

$$\frac{d^n f(p)}{dp^n} = \frac{d^n \text{Pade}_{[M/N]}}{dp^n} \qquad 0 \leq n \leq M+N+1, \tag{4.65}$$

giving an approximation such that

$$f(p) - \text{Pade}_{[M/N]} = O(p^{M+N+1}). \tag{4.66}$$

In the particular case of applying it to solve the HJB equation, the modes resulting from the HPM are approximated using the Padé approximation around $p = 0$, hence from the value of the first guess to the true solution to the non-linear PDE. In the next section, the HPM and the Padé approximation are applied to solve for the optimal learning rate scheduling described by equation 4.59.

4.3.4 Approximated Solution to the HJB-equation

To solve the HJB-equation that describe the cumulative risk under optimal learning rate scheduling, first a homotopy is constructed similarly to equation 4.62 assuming a guess function $f_g = 0$, obtaining

$$(1-p)\frac{\partial R}{\partial t} + p\left(\frac{\partial R}{\partial t} + \log(\gamma)R(w,t) + L(w) - \alpha\frac{\partial R}{\partial w}\left\langle\frac{dL}{dw}\right\rangle - \frac{1}{2\beta}\left(\frac{\partial R}{\partial w}\left\langle\frac{dL}{dw}\right\rangle\right)^2\right) = 0. \tag{4.67}$$

Now writing the solution as a polynomial over the embedding parameter

$$R(w,t,p) = \sum_{i=0}^{\infty} p^i R_i(w,t) \tag{4.68}$$

Because the optimal control depends linearly on a partial derivative of $R(w,t)$ as described in equation 4.57, the optimal control signal will also be described by a sum of modes of

the form

$$g(t) = \sum_{i=0}^{\infty} p^i g_i(t), \text{ where } g_i = \frac{1}{\beta}\frac{\partial R_i}{\partial w}\left\langle \frac{dL}{dw} \right\rangle_{xy}. \tag{4.69}$$

Replacing the sum of modes $R(w,t,p)$ in the HJB-equation, the coefficients for each power of p is collected and equated to 0 such that the sum of modes solves the equation. For instance, solving for a few modes:

Coefficient of p^0:

$$\frac{\partial R_0}{\partial t} = 0. \tag{4.70}$$

Then, the first control mode $g_0 = 0$.

Coefficient of p^1:

$$\frac{\partial R_1}{\partial t} - \frac{\partial R_0}{\partial t} + \frac{\partial R_0}{\partial w} + \log(\gamma)R_0 + L(w) - \alpha\frac{\partial R_0}{\partial t}\left\langle \frac{dL}{dw} \right\rangle - \frac{1}{2\beta}\left(\frac{\partial R_0}{\partial t}\left\langle \frac{dL}{dw} \right\rangle\right) = 0. \tag{4.71}$$

Replacing $R_0 = 0$

$$\frac{\partial R_1}{\partial t} + L(w) = 0, \tag{4.72}$$

then integrating through time (partially integrating as R depends explicitly on w and t), enforcing $R_1(w,T) = 0$, and computing the optimal control for mode g_1

$$R_1(w,t) = (T-t)L(w) \rightarrow g_1(t) = \frac{1}{\beta}\frac{\partial R_1}{\partial w}\left\langle \frac{dL}{dw} \right\rangle = \frac{T-t}{\beta}\left\langle \frac{dL}{dW} \right\rangle^2 \tag{4.73}$$

Coefficient of p^2: The non-zero terms are

$$\frac{\partial R_2}{\partial t} + \log(\gamma) R_1 - \alpha \frac{\partial R_1}{\partial w} \left\langle \frac{dL}{dw} \right\rangle = 0, \tag{4.74}$$

$$\frac{\partial R_2}{\partial t} = (T-t) \left[\alpha \left\langle \frac{dL}{dw} \right\rangle^2 - \log(\gamma) L(w) \right]. \tag{4.75}$$

Time integrating (s.t $R_2(w,T) = 0$), then expanding the gradient in terms of the task statistics μ_x and ξ^2

$$R_2(w,t) = \frac{(T-t)^2}{2} \left[\log(\gamma) L(w) - \alpha \left\langle \frac{dL}{dw} \right\rangle^2 \right], \tag{4.76}$$

$$= \frac{(T-t)^2}{2} \left[\log(\gamma) L(w) - \alpha \left(\xi^2 w - \mu_x \right)^2 \right]. \tag{4.77}$$

Then the second mode of control can be computed

$$g_2(t) = \frac{1}{\beta} \frac{\partial R_2}{\partial w} \left\langle \frac{dL}{dw} \right\rangle = \frac{(T-t)^2}{\beta} \left\langle \frac{dL}{dw} \right\rangle^2 \left(\frac{\log(\gamma)}{2} - \alpha \xi^2 \right). \tag{4.78}$$

This can continue recursively for the rest of the modes. Writing the truncated version of the optimal control signal in terms of the first 4 modes and $p = 1$ gives

$$\begin{aligned} g^*(t) =& \frac{(T-t)}{\beta} \left\langle \frac{\partial L}{\partial w} \right\rangle^2 \\ &+ \frac{(T-t)^2}{\beta} \left\langle \frac{dL}{dw} \right\rangle^2 \left(\frac{\log(\gamma)}{2} - \alpha \xi^2 \right) \\ &+ \frac{(T-t)^3}{6\beta^2} \left\langle \frac{\partial L}{\partial w} \right\rangle^2 \left(4\xi^2 \alpha^2 \beta - 4\xi^2 \alpha \beta \ln(\gamma) - 4\xi^2 \left(\frac{\partial L}{\partial w} \right)^2 + \beta \ln^2(\gamma) \right) \\ &+ ... \end{aligned} \tag{4.79}$$

Before moving on to the Padé Approximation, it is important to note that from this expression it is already possible to obtain some insight into the behavior of control with

respect to some of the parameters of the task, which corroborates the correctness of the solution. For instance, the control signal is reduced as time t approaches the time limit T to learn a task. Also, it is proportional to the expected gradient squared, meaning that control is less necessary when the gradient approaches 0, because the task is close to being learned. Finally, the control intensity will decrease with the size of the control cost coefficient β. The rest of the parameters are challenging to interpret. Fortunately, the Padé approximation under a few modes gives a better approximation and a still interpretable solution, where the effect of γ and α can be analyzed. It is important to note that the solution with the first few modes does not necessarily behave optimally in all domains of parameters, and it can easily diverge in cases where the available time to learn is enough for the agent to converge. However, there are other cases where the homotopy method (with no Padé approximation) follows the simulation closely (non-convergent learning of the agent) and follows a sweep of values as well (see subsection C.1.2). Further modes are provided in subsection C.1.1.

This solution in equation 4.79 is an *online controller*, meaning that in principle, it should be possible to apply it online while learning. However, some of the quantities needed to compute the optimal control are statistics of the task, such as μ_x and ξ^2, which, in the case of having access to them, allow the optimal weight to be computed in closed form. Regardless of this, it might be possible to apply it online by estimating each statistic $\mu_x \approx \hat{\mu_x}$ and $\xi \approx \hat{\xi}$, e.g., using batches to estimate these online.

Next, an example of Padé computation is given for $M = 1$ and $N = 1$. First, the control

signal is written as a polynomial of p,

$$g^*(t) = \sum_{i=0}^{\infty} p^i g_i \tag{4.80}$$

$$= p \underbrace{\frac{(T-t)}{\beta}\left\langle \frac{\partial L}{\partial w} \right\rangle^2}_{g_1}$$

$$+ p^2 \underbrace{\frac{(T-t)^2}{\beta}\left\langle \frac{dL}{dw} \right\rangle^2 \left(\frac{\log(\gamma)}{2} - \alpha\xi^2 \right)}_{g_2}$$

$$+ p^3 \frac{(T-t)^3}{6\beta^2}\left\langle \frac{\partial L}{\partial w} \right\rangle^2 \left(4\xi^2\alpha^2\beta - 4\xi^2\alpha\beta\ln(\gamma) - 4\xi^2\left(\frac{\partial L}{\partial w}\right)^2 + \beta\ln^2(\gamma) \right)$$

$$+ ... \tag{4.81}$$

then approximating up to mode p^2, using $\text{Pade}_{[M=1,N=1]}$,

$$pg_1 + p^2 g_2 - \frac{ap}{1+bp} = O(p^3). \tag{4.82}$$

Solving for a and b,

$$(pg_1 + p^2 g_2)(1+bp) - ap = 0 \tag{4.83}$$

$$pg_1 + p^2 g_2 + p^2 g_1 b + p^3 b g_2 - ap = 0 \tag{4.84}$$

$$\implies g_1 = a, \text{ and } b = \frac{-g_2}{g_1}. \tag{4.85}$$

By equation coefficients of powers of p. The Pade approximation is then

$$g(t) = \text{Pade}_{[M=1,N=1]} = \frac{1}{\beta}\frac{\left\langle \frac{dL}{dW} \right\rangle^2}{\frac{1}{T-t} - \frac{\log(\gamma)}{2} + \alpha\xi^2} = \frac{1}{\beta}\frac{(\mu_x - w\xi^2)^2}{\frac{1}{T-t} - \frac{\log(\gamma)}{2} + \alpha\xi^2}. \tag{4.86}$$

In general, the Padé approximation requires that the number of homotopy modes $K \leq M + N + 1$ (K starting from 0). Importantly, this simplified expression exhibits well-

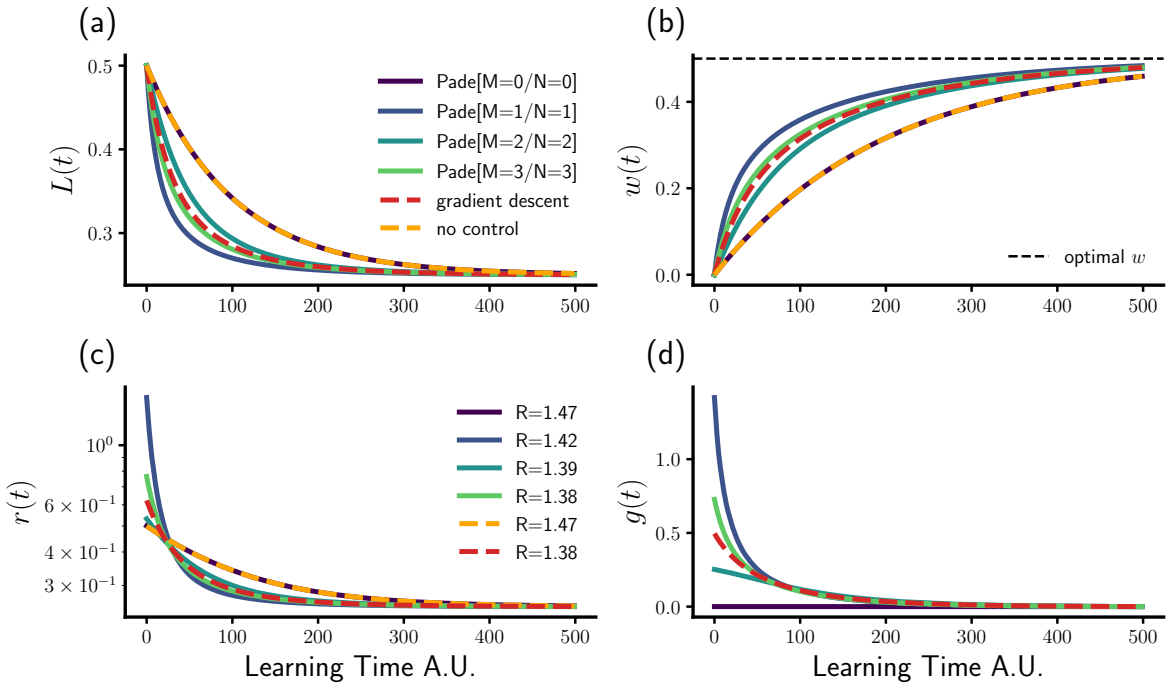

Figure 4.6: Padé approximation of learning rate control problem. **(a)**: Loss function under different levels of Padé approximation. **(b)**: Weight evolution. **(c)**: Instant reward rate $r(t)$, and cumulative reward $R = \int r dt$, being improved with number of modes considered in the Padé approximation. **(d)**: Optimal control signal approximation.

behaved characteristics compared to higher-order Padé approximations (see figure 4.6), as a satisfactory approximation of the control signal is achieved with only a few homotopy modes. Using the same number of modes without Padé approximation results in a divergent solution (see subsection C.1.2). While the first-order approximation provided in equation 4.86 is sufficiently simple for inspection, higher-order Padé approximation modes are highly complex and were computed automatically using a symbolic library called `SymPy` [3]. A higher-order Padé approximation is provided in section C.2, which is considerably more challenging to interpret.

The expression from equation 4.86 provides expected relationships between the task variables and the amount of control exerted. Similarly to the homotopy modes, control

[3] https://www.sympy.org/en/index.html

will decrease with a decrease in the performance gradient $\left\langle \frac{dL}{dW} \right\rangle^2$, and with the amount of time available to learn the task $T - t$. As $0 \leq \gamma \leq 1$, a decrease in the discount factor will also decrease the amount of control. A higher base learning rate α (that is, learning that does not require effort, similar to the passive learning strategy proposed in subsection 4.2.3) will decrease the amount of control needed, as the improvement in performance will occur automatically through the baseline (no control) dynamics (see figure 4.7). These trade-offs are expected from a normative perspective and can be connected back to cognitive neuroscience literature through the connections made in chapter 3. To confirm that these relationships are correct, the first-order approximation and a higher-order approximation $\text{Pade}_{[M=3,N=3]}$ were compared against the optimal control computed numerically using gradient steps as in the learning effort framework (see Algorithm 1). The comparison is performed in the space of cumulative risk, as it is the solution of the HJB equation, and the integrated amount of control exerted during the learning period.

As expected, including more modes in the Padé approximation improves the match with numerical simulation. However, the first-order approximation $\text{Pade}_{[M=1,N=1]}$ follows similar trends when varying the hyperparameters of the learning system, providing a reasonable trade-off between the quality of the approximation and interpretability. Additionally, more complex models also follow the same relationships of hyperparameters of the learning system optimized numerically using the gradient descent method of the learning effort framework (see figure C.5 in subsection C.1.2).

4.4 Discussion

This chapter provides a thorough process to study the control of learning. Starting from a very simple setup proposed by others, as in (Son and Sethi, 2006, 2010), it gradually adds complexity to transition to a more complex, continuous-time control optimization problem.

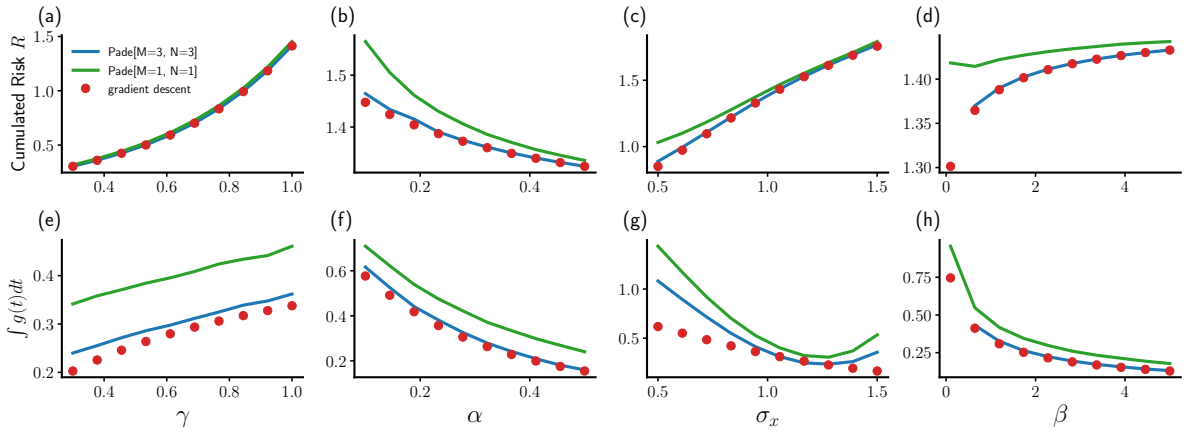

Figure 4.7: Homotopy Pade approximation parameter sweep for gradient descent based solution, first order pade approximation ($[M = 1, N = 1]$) and higher order Pade approximation ($[M = 3, N = 3]$). **First row**: Cumulated Risk R at the start of learning. **Second row**: Integrated amount of control $\int g(t)dt$. **From left to right**: Variations of learning system and task hyperparameter. Discount factor γ, base learning rate α, task difficulty or aleatoric noise σ_x and cost of control β.

It explores potential avenues for further analysis and provides a mathematical explanation of observations made in simulations. One of the main insights from these simple models is the conjecture discussed in subsection 4.2.3, *learn first, do later*. The description of this conjecture is rather simple, but it has the potential to describe phenomena in cognitive neuroscience, as previously discussed in chapter 3, through the connection to the expected value of control theory (EVC, Shenhav et al. 2013; Masís et al. 2021) or the role of neurotransmitters in meta-learning (Doya, 2002; Lee et al., 2024b). In addition, it relates to other kinds of relations in machine learning, such as the exploration-exploitation trade-off (further discussed in chapter 5) or annealing strategies Nakamura et al. (2021).

While the models presented are rather simple, the optimal solution describing control over learning is not trivial. Even for simple settings, e.g., controlling the learning rate, the influence of control over state variables, in this case, the weights of a network, is non-linear, hence making it challenging for mathematical analysis. In addition to the homotopy perturbation method, other tools from control theory, such as the extended

Kalman filter (Urrea and Agramonte, 2021) and non-linear system identification (Nelles, 2001), might offer further analytical tools to manipulate the problem of optimal control over learning. However, not many of these methods dedicate their efforts to optimize over a complete period of time, as in equation 4.39; rather, the performance evaluation is done at a fixed time step, or in a greedy manner.

Further systems that are being explored at the moment are two-layer linear networks with optimal learning rate scheduling and multi-task attention control, where the optimal control must choose how to allocate its attention to each task, similar to some of the experiments presented in section 3.5. However, for the two-layer linear network, the analytical approximations are not close to what is observed in simulation, diverging most of the time. For the multi-task setting, adding non-linear constraints to the space of possible control vectors as attention is challenging, as it makes the HJB equation unsolvable in most cases. Other alternatives to tackle these technical issues are, for instance, describing the learning dynamics of the two-layer linear network in terms of its *order parameters*, so the differential equation describing the evolution of the weights only changes for a few variables that summarize the system, instead of keeping track of all the weights of the network. While these two system descriptions are equivalent in expectation, the dynamical system in order parameters is smoother and usually less complex. Additionally, for the multi-task approach, one interesting constraint that could be added to the control signal (now a vector where each entry is attention to each of the tasks) is to force it to live on a sphere, so that paying attention to one task necessarily incurs an opportunity cost of not attending another task. Imposing this constraint could be possible by splitting the HJB equation into different conditions that project the control back to the permitted control space. These ideas are currently being tested, but as indicated throughout section 4.3, trying these avenues requires extensive algebra and math tricks, both done manually and through automatic symbolic computation, which in some cases can limit the quality of the approximation. Beyond applying this method

to control of learning, further efforts could be made to study convergence guarantees of the homotopy perturbation methods in machine learning problems. Nevertheless, the potential gain of having a mathematical theory of control of learning is important, as it could shed light on fundamental trade-offs and formal principles of control of learning.

Chapter 5

Uncertainty Prioritized Experience Replay

The work presented in this chapter is primarily based on *Uncertainty Prioritized Experience Replay*, Accepted to RLC 2025. Available at ICLR 2025 OpenReview[1], by Rodrigo Carrasco-Davis*, Sebastian Lee*, Claudia Clopath, Will Dabney (* equal contribution). The text has been modified to conform to the thesis format.

5.1 Introduction

DRL has proven highly effective across a diverse array of problems, consistently yielding state-of-the-art results in control of dynamical systems (Nian et al., 2020; Degrave et al., 2022; Weinberg et al., 2023), abstract strategy games (Mnih et al., 2015; Silver et al., 2016), continual learning (Khetarpal et al., 2022; Team et al., 2021), and multi-agent learning (OpenAI et al., 2019; Baker et al., 2020). It has also been established as a foundational theory for explaining phenomena in cognitive neuroscience (Botvinick et al.,

[1] https://openreview.net/forum?id=aAxzDb0nlO

2020; Subramanian et al., 2022). Nonetheless, a significant drawback of these methods pertains to their inherent *sample inefficiency* whereby accurate estimations of value and policy necessitate a substantial demand for interactions with the environment.

Sample inefficiency has been mitigated through the use of, among other methods, prioritized experience replay (PER, Schaul et al. 2016, see subsection 2.1.4). PER is an extension of Experience Replay (Lin, 1992), which uses a memory buffer populated with past agent transitions to improve training stability through the temporal de-correlation of data used in parameter updates. Subsequently, PER extends this approach by sampling transitions from the buffer with probabilities proportional to their absolute TD-error, thereby allowing agents to *prioritize* learning from pertinent data. PER has been widely adopted as a standard technique in DRL; however, despite significantly better performance over uniform sampling in most cases, it is worth noting that PER can encounter limitations under specific task conditions and agent designs. The most prominent example of such a limitation is related to the so-called *noisy TV* problem (Burda et al., 2018), a thought experiment at the heart of the literature around exploration in RL. Just as novelty-based exploration bonuses can trap agents in noisy states, PER is susceptible to frequently replaying transitions involving high levels of randomness (e.g. in reward or transition dynamics) even if they do not translate to meaningful learning and thus are not useful for solving the task.

To combat this issue, a combination of epistemic and aleatoric uncertainty measures is proposed (Clements et al., 2020; Alverio et al., 2022; Lahlou et al., 2022; Liu et al., 2023; Jiang et al., 2023), originally used to promote exploration, under an information gain criterion for use in replay prioritization. Epistemic uncertainty, the uncertainty reducible through learning, is the key quantity of interest. However, this needs to be appropriately "calibrated", which, it is shown, both empirically and with justification from Bayesian inference, can be done effectively by dividing the epistemic uncertainty estimate by an aleatoric uncertainty estimate. Intuitively, the need for this kind of calibration

can be seen by considering the following game: the aim is to estimate the mean of two distributions; the ground truth is that both distributions have identical means but different variances, and the current estimates for both distributions are the same, i.e., the epistemic uncertainty on the mean is the same for both distributions. However, if a new sample from either distribution is offered to refine the estimate, one would choose to sample the distribution with lower variance since this is more likely to be informative. In addition to arguing for this novel prioritization variable, candidate methods involving distributions of ensembles (in the vein of Clements et al. 2020) to estimate these quantities are also provided. A comprehensive background of the used methods is provided in section 2.4 and its relation to PER and exploration methods is discussed in subsection 5.5.1.

The primary contributions are as follows:

1. In section 5.2, a novel approach for estimating epistemic uncertainty is presented, building upon an existing uncertainty formalization introduced by Clements et al. (2020) & Jiang et al. (2023). This extension incorporates information about the target value that the model aims to estimate, thereby accounting for bias in the estimator.

2. A prioritization variable is derived using estimated uncertainty quantities, finding a specific functional form derived from a concept called *information gain*, showing that both epistemic and aleatoric uncertainty should be considered for prioritization.

3. In section 5.3, the advantages of the proposed epistemic uncertainty prioritization scheme are illustrated through two interpretable toy models—a bandit task and a grid world.

4. In section 5.4, the effectiveness of this method is demonstrated on the Atari-57 benchmark (Bellemare et al., 2013), where it significantly outperforms baseline models based on a combination of PER, QR-DQN, and ensemble agents.

5.2 Proposed Method: Uncertainty Prioritized Experience Replay

In this section, a new method for estimating epistemic uncertainty is introduced, which arises from a decomposition of the total uncertainty as defined by the average error over both the ensemble and quantiles (subsection 2.4.4). This decomposition is in the vein of Clements et al. (2020); however, the proposed estimator considers distance from the target in addition to the disagreement within the ensemble, thereby allowing us to handle—among others—model bias. An expression for prioritization variables based on the concept of *information gain* is derived, which trades off epistemic and aleatoric uncertainty with a view to maximizing learnability from each sampled transition. This method is named Uncertainty Prioritized Experience Replay (UPER). Importantly, the prioritized replay algorithm itself is not changed (subsection 2.1.4), but just the variable p_i used to prioritize in equation 2.27, replacing the TD-error by the information gain.

5.2.1 Uncertainty from Distributional Ensembles

The definitions given in equation 2.46 arise from a decomposition of $\mathbb{V}_{\psi,\tau}[\theta_\tau(s,a;\psi)]$, where ψ and τ index the quantile and ensemble, respectively (see Clements et al. (2020) for details). This quantity does not explicitly consider how far estimates are from targets, but rather how consistent the estimates are among the quantiles and members of the ensemble. A modified concept of total uncertainty $\hat{\mathcal{U}}_\delta$, named *target total uncertainty*, is proposed, simply defined as the average squared error to the target Θ over the quantiles and ensemble, which can be decomposed as:

$$\hat{\mathcal{U}}_\delta = \mathbb{E}_{\tau,\psi}[(\Theta(s',r) - \theta_\tau(s,a;\psi))^2] = \underbrace{\delta^2_\Theta(s,a) + \hat{\mathcal{E}}(s,a)}_{\hat{\mathcal{E}}_\delta(s,a)} + \hat{\mathcal{A}}(s,a); \tag{5.1}$$

where $\delta^2_\Theta(s,a) = (\Theta(s',r) - \mathbb{E}_{\tau,\psi}[\theta_\tau(s,a;\psi)])^2$, and the *target epistemic uncertainty* $\hat{\mathcal{E}}_\delta(s,a) = \delta^2_\Theta(s,a) + \hat{\mathcal{E}}(s,a)$ is introduced. The proof is as follows. Dropping the dependency on the transition (s',r,s,a) in the target total uncertainty for simplicity, it can be decomposed into:

$$\mathbb{E}_{\tau,\psi}[(\Theta - \theta_\tau(\psi))^2] = \int_\psi \frac{1}{N}\sum_\tau^N (\Theta - \theta_\tau(\psi))^2 P(\psi|D)d\psi, \tag{5.2}$$

$$= \int_\psi \frac{1}{N}\sum_\tau^N \left[\Theta - \theta_\tau(\psi) \pm \mathbb{E}_\psi(\theta_\tau(\psi))\right]^2 P(\psi|D)d\psi, \tag{5.3}$$

$$= \int_\psi \frac{1}{N}\sum_\tau^N \Big[(\Theta - \mathbb{E}_\psi(\theta_\tau(\psi)))^2 + (\mathbb{E}_\psi(\theta_\tau(\psi)) - \theta_\tau(\psi))^2 \tag{5.4}$$

$$+2\left(\Theta - \mathbb{E}_\psi(\theta_\tau(\psi))\right)\left(\mathbb{E}_\psi(\theta_\tau(\psi)) - \theta_\tau(\psi)\right)\Big] P(\psi|D)d\psi, \tag{5.5}$$

$$= \int_\psi \frac{1}{N}\sum_\tau^N \left(\Theta - \mathbb{E}_\psi(\theta_\tau(\psi))\right)^2 P(\psi|D)d\psi \tag{5.6}$$

$$+ \underbrace{\frac{1}{N}\sum_\tau^N \int_\psi \left(\mathbb{E}_\psi(\theta_\tau(\psi)) - \theta_\tau(\psi)\right)^2 P(\psi|D)d\psi}_{\hat{\mathcal{E}} \text{ in equation 2.47}}, \tag{5.7}$$

the term in equation 5.5 is zero when integrating over ψ. Finally, the term in equation 5.6 is

$$\int_\psi \frac{1}{N}\sum_\tau^N \left(\Theta - \mathbb{E}_\psi(\theta_\tau(\psi))\right)^2 P(\psi|D)d\psi = \Theta^2 - 2\mathbb{E}_{\psi,\tau}\left(\theta_\tau(\psi)\right) + \mathbb{E}_\tau\left(\mathbb{E}_\psi\left[\theta_\tau(\psi)\right]^2\right) \tag{5.8}$$

$$= \underbrace{\left(\Theta - \mathbb{E}_{\psi,\tau}\left[\theta_\tau(\psi)\right]\right)^2}_{\text{Distance to the target } \delta^2_\Theta} + \underbrace{\mathbb{V}_\tau\left(\mathbb{E}_\psi\left[\theta_\tau(\psi)\right]\right)}_{\hat{\mathcal{A}} \text{ in equation 2.46}}, \tag{5.9}$$

obtaining the proposed uncertainty decomposition

$$\hat{\mathcal{U}}_\delta = \mathbb{E}_{\tau,\psi}[(\Theta(s',r) - \theta_\tau(s,a;\psi))^2] = \delta^2_\Theta(s,a) + \hat{\mathcal{E}}(s,a) + \hat{\mathcal{A}}(s,a); \tag{5.10}$$

Note that in order to construct ensemble disagreement estimates or estimates of the total uncertainty $\hat{\mathcal{U}}_\delta$, independence among the ensemble is assumed, which is facilitated by masking and random initialization akin to bootstrapped DQN. Through the lens of the DEUP formulation from subsection 2.4.3, this decomposition suggests a modified definition of epistemic uncertainty that considers the distance to the target δ^2_Θ as well as the disagreement in estimation within the ensemble $\hat{\mathcal{E}}$ from Clements et al. (2020) and Jiang et al. (2023). To see why this extra term can be useful, consider the following pathological example: all members of an ensemble are initialized equally; the variance among the ensemble, and the resulting epistemic uncertainty estimate without this additional error term, will be zero. A more subtle generalization of this would be if inductive biases from other parts of the learning setup (architecture, learning rule, etc.) lead to characteristic learning trajectories in which individual members of the ensemble effectively collapse with no variance. In essence, $\hat{\mathcal{E}}$ assesses ensemble disagreement without including the estimation offset. The use of pseudo-counts (Lobel et al., 2023) presents a similar problem: while epistemic uncertainty does scale with the number of visits to a state, it does not necessarily encode the true distance between the estimation and target values. Pseudo-counts bear the additional disadvantage of being task-agnostic, i.e., ignoring context, which makes them brittle under any change in the underlying MDP. A simulation is provided in section 5.3 where the advantage of using $\hat{\mathcal{E}}_\delta$ instead of $\hat{\mathcal{E}}$ to prioritize replay is shown.

5.2.2 Prioritising using Information Gain

Having arrived at suitable methods for estimating both epistemic and aleatoric uncertainty, it remains to establish a functional form for the prioritization variable, denoted $p_i = h(\mathcal{E}(s_i, a_i), \mathcal{A}(s_i, a_i))$. The most straightforward approach is to directly use $p_i = \hat{\mathcal{E}}_\delta$; however, in practical applications, this does not yield satisfactory results. One intuition for this, which will be made more concrete in later passages, is that the magnitude of

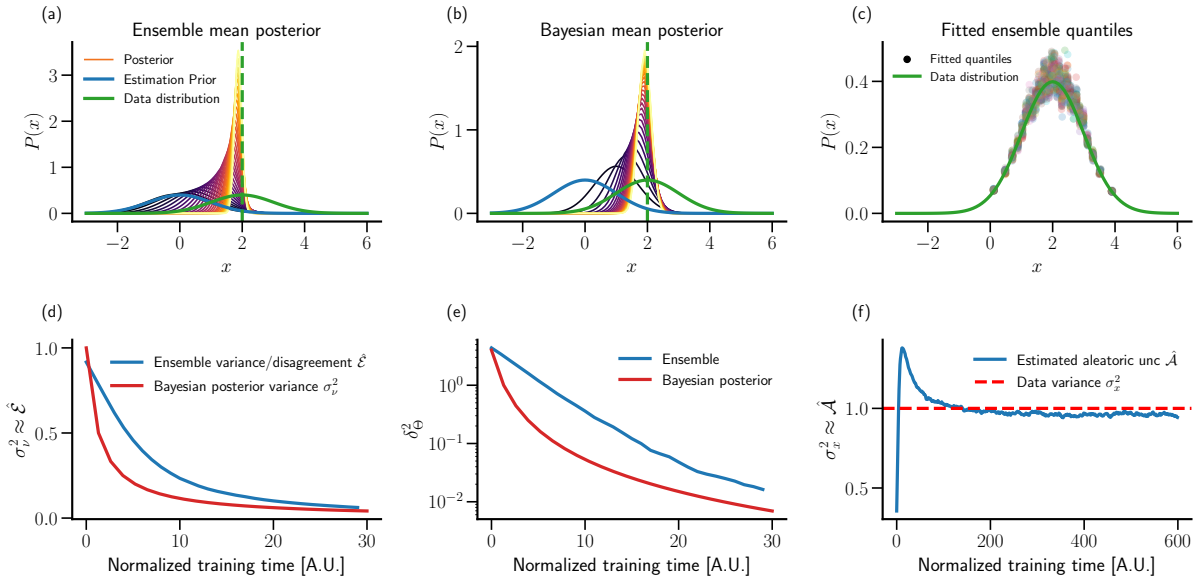

Figure 5.1: **Variances in the information gain can be approximated by epistemic and aleatoric uncertainty in the information gain**: **(a) and (b)** Evolution during training of the posterior of the mean using an ensemble (gaussian fitted to members of the ensemble at each step) and an ideal Gaussian respectively. Training progresses from purple to yellow. **(c)**: Fitted ensemble quantiles to true data distribution. **(d)**: Ensemble disagreement (equivalent to variance of the posterior estimated with ensemble as $\hat{\mathcal{E}}$ in equation 2.47) and true posterior variance σ_ν^2 from ideal Gaussian. **(e)**: Distance to the target true value δ_Θ. **(f)**: Data variance σ_x^2 approximated with $\mathcal{A}$ in equation 2.47. Training time was scaled to show a match between Gaussian posterior and uncertainty measures.

epistemic uncertainty does not in itself determine how easily reducible that uncertainty is.

It is informative therefore to also consider the aleatoric uncertainty, since this indicates the fidelity of the data, and hence how readily it can be used to reduce the epistemic uncertainty (this is demonstrated experimentally in subsection 5.3.1 and section B.3, and expounded upon in section B.2).

We take inspiration from the idea of *information gain* to determine h. For the purpose of this explanation, consider a hypothetical dataset of points $x_i \sim \mathcal{N}(\mu_x, \sigma_x^2)$. The objective is to estimate the posterior distribution $P(\nu|x_i) \propto P(x_i|\nu)P(\nu)$ with a prior distribution $\nu \sim \mathcal{N}(\mu, \sigma^2)$. Following the observation of a single sample x_i, the posterior distribution

becomes a Gaussian with variance $\sigma_\nu^2 = \frac{\sigma^2\sigma_x^2}{\sigma_x^2+\sigma^2}$. To quantify the information gained by incorporating the sample x_i when computing the posterior, the difference in entropy between the prior distribution and the posterior is measured as

$$\Delta\mathcal{H} = \mathcal{H}\left(P(\nu)\right) - \mathcal{H}\left(P(\nu|x_i)\right), \tag{5.11}$$

From here, $\sigma^2 = \hat{\mathcal{E}}_\delta$ is considered as a form of epistemic uncertainty, since the ensemble disagreement is reduced by sampling more points, and $\sigma_x^2 = \hat{\mathcal{A}}$ is considered as aleatoric uncertainty corresponding to the variance of the ensemble average distribution, giving the irreducible noise of the data, obtaining a prioritization variable

$$p_i = \Delta\mathcal{H}_\delta = \frac{1}{2}\log\left(1 + \frac{\hat{\mathcal{E}}_\delta(s,a)}{\hat{\mathcal{A}}(s,a)}\right). \tag{5.12}$$

For a detailed derivation of the information gain see subsection B.2.1, and a comprehensive exploration of other functional forms of prioritization variables based on uncertainty, please refer to section 2.4.

5.3 Motivating Examples

In this section, the epistemic uncertainty estimators and the information gain criterion is employed in simple and interpretable toy models to highlight their potential as experience replay prioritization variables.

5.3.1 Conal Bandit

A multi-armed bandit task is devised in which each arm has the same expected reward but with increasing noise level per arm, forming a *cone* as shown from left to right in figure 5.2a. The memory buffer in this experiment has one transition per arm, and after sampling one arm, the observed reward is replaced in the buffer for the respective

transition (as done in the toy example in the original PER paper (Schaul et al., 2016)). Specifically, let n_a denote the number of arms; then the reward distribution r for arm a is defined as:

$$r(a) = \bar{r} + \eta \cdot \sigma(a), \quad \sigma(a) = a \cdot \sigma_{\max}/(n_a - 1) + \sigma_{\min}; \tag{5.13}$$

where $\bar{r}$ represents the expected reward, $\sigma(a)$ is the reward standard deviation associated with arm a, $\sigma_{\max}$ and $\sigma_{\min}$ are constant, and η is sampled from a centred, unit-variance Gaussian.

The choice of employing noisy arms serves the purpose of demonstrating that the TD-errors will inherently include the sample noise, regardless of whether the reward estimation for each arm $Q(a) = \mathbb{E}_j[\theta_j(a)]$ approximates the target value $\bar{r}$. Results for the bandit task using different variables to prioritize learning are depicted in figure 5.2b for $n_a = 5$, $\bar{r} = 2$, $\sigma_{\max} = 2$, and $\sigma_{\min} = 0.1$ (details in section B.3).

Four relevant prioritisation schemes are shown in this section (see section B.3 for other prioritisation schemes): TD-error (standard PER): $p_i = \frac{1}{N_e} \sum_{\psi} |r_i - Q(a_i; \psi)|$; Inverse count: $p_i = 1/\sqrt{1 + C}$, where C denotes the number of times an arm has been sampled to update the reward estimate; Information gain (UPER): $p_i = \Delta \mathcal{H}_\delta$; True distance to target: $p_i = \mathcal{E}^* = |\bar{r} - Q(a_i)|$.

Prioritizing with epistemic uncertainty measures, such as UPER or inverse counts (a proxy for epistemic uncertainty), leads to improved training speed and final true Mean Squared Error (true MSE, averaged across all arms, between the estimated reward and the true mean reward), compared to $p_i = |\delta_i|$ (PER), as illustrated in figure 5.2b. Throughout the paper, we highlight that the TD-error includes aleatoric uncertainty, corresponding to the arm variance in this scenario, which is irreducible through learning (see section B.1 for more details). Therefore, the TD-error tends to over-sample arms with high variance compared with UPER, to the cost of not sampling the low variance arm. This is demonstrated in figure 5.2c.

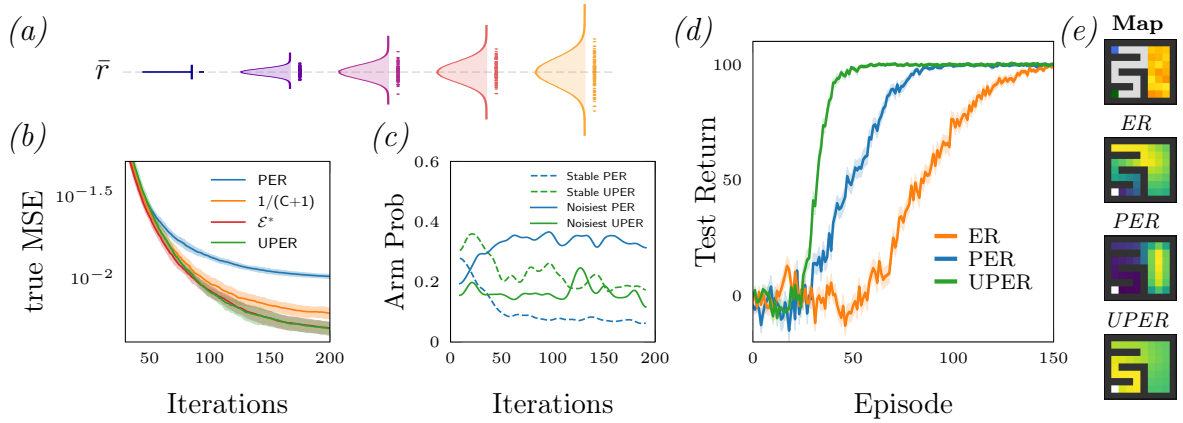

Figure 5.2: **Conal Bandit.** *(a)* multi-armed bandit task constructed such that each arm has identical mean payoff but increasing variance. *(b)* true MSE (average error across arms, between estimated reward and the true reward mean) over 200 iterations (each of 1000 steps) using different quantities to prioritise transitions from the replay buffer: absolute value of the TD error $|\delta|$ (PER), inverse counts (C being the number of visits to the respective arm), information gain $\Delta\mathcal{H}_\delta$ (UPER), and an oracle epistemic uncertainty $\mathcal{E}^*$ measured as the distance from the estimated mean to the true mean. *(c)* arm replaying selection probabilities for the stablest (dashed) and noisiest (solid) arms in the conal bandit; the key intuition is that prioritising by TD-error over-samples noisier arms, while prioritising using UPER places importance on learn-ability and leads to greater selection of stable arms. Results averaged across 10 seeds. **Noisy Gridworld.** *(d)* 300 seeds return on a test episode throughout training of an agent on the noisy gridworld, with the shaded region being stared error on the mean. *(e)* in the **Map**, blue denotes the starting state, green is the goal state, and yellow are the non-zero variance immediate rewards. Below, sampling heatmaps where yellows are highly sampled and blues are scarcely sampled: uniform experience replay (ER) leads to sampling more from early parts of a trajectory since these fill the buffer first; replay based on TD error (PER) leads to a pathological sampling of the noisy part of the gridworld; replay using UPER leads to greater sampling of later parts of the trajectory.

Using inverse counts as the prioritization variable (similar to Lobel et al. (2023)) outperforms TD-error (as designed in the task) but not UPER. The reason is that, although each initial estimated Q-value per arm is equidistant from the true mean, the learning speed for each arm diminishes with the variance of the respective arm. Inverse counts do not account for this variance-dependent decay in learning speed, so the number of updates

per arm will not reflect the distance of the estimation to the true target, whereas UPER (prioritizing by $\hat{\mathcal{E}}_\delta$ and inverse $\hat{\mathcal{A}}$) tends to sample arms with high aleatoric uncertainty less frequently and is also based on the distance to the targets as defined in equation 5.10.

The distance between the estimated mean and the true mean, denoted as $\mathcal{E}^*$ (accessible due to the task design), is equivalent to the epistemic uncertainty in the DEUP formulation, as derived in subsection 2.4.3. This distance is the ideal prioritization variable to which there is not access in general. Notably, using UPER, which prioritizes based on information gain, yields results comparable to prioritizing directly based on the true distance. These results show UPER as a promising modification to TD-error-based prioritized replay.

To emphasize the significance of incorporating the target value when utilizing the target epistemic uncertainty $\hat{\mathcal{E}}_\delta$ for replay prioritization, modifications to the conal bandit task were introduced by assigning distinct mean rewards per arm, denoted as $\bar{r} \rightarrow \bar{r}(a)$ (see simulation details in section B.3, figure B.3). In the original conal bandit task, all arms shared the same mean reward $\bar{r}$, resulting in an equal initial distance expectation from $Q(a)$ to each arm. This uniformity dampened the performance improvement when considering the target distance δ_Θ in $\hat{\mathcal{E}}_\delta$ with respect to $\hat{\mathcal{E}}$. By introducing varying mean rewards per arm, denoted as $r(a)$, the relevance of information about the target value becomes important. This adjustment highlights the advantage of employing the proposed target epistemic uncertainty $\hat{\mathcal{E}}_\delta$ over merely considering ensemble disagreement $\hat{\mathcal{E}}$.

5.3.2 Noisy Gridworld

To move toward the full reinforcement learning (RL) problem, this section considers a tabular gridworld. Inspired by ideas from planning within dynamic programming methods (Moore and Atkeson, 1993), the goal is to explore uncertainty-guided prioritized replay. Under this framework, "direct" reinforcement learning through interactions with the environment (sometimes referred to as control) is typically supplemented by "indirect"

learning of a model from stored experiences (sometimes referred to as planning). In this case, learning is purely model-free, yet it retains the distinction between offline and online learning. In many ways, these methods serve as a precursor to the use of experience replay buffers in deep reinforcement learning.

When making updates on stored data offline (whether for planning or other purposes), similar questions arise regarding criteria for prioritization. Notably, prioritized sweeping, which favors high-error samples in memory, was an early extension to the Dyna models that exemplify this learning protocol (Sutton, 1991).

In figure 5.2e (Map), a gridworld is constructed where the agent encounters highly noisy states with random rewards early in the episode, while a single deterministic state with a much larger reward is located at the end of the maze. figure 5.2d demonstrates that this simple task can be solved without additional planning steps. However, experience replay (ER), which samples uniformly, improves sample efficiency. This improvement is further enhanced by prioritized experience replay (PER), which prioritizes transitions based on temporal-difference (TD) error. The best performance is achieved by uncertainty-prioritized experience replay (UPER), which prioritizes transitions based on the information gain criterion and the inverse of state visitation counts, a reliable proxy for epistemic uncertainty in this tabular setting.

As shown by the heatmaps in figure 5.2e, PER tends to over-sample noisy states, whereas UPER prioritizes novel states toward the end of the trajectory. Full details of the experimental setup and hyperparameters can be found in section B.4.

5.4 Deep RL: Atari

In the final set of experiments, the proposed method is applied in a deep reinforcement learning setting, specifically using the Atari benchmark (Bellemare et al., 2013). The agent is an ensemble of QR-DQN distributional predictors ($N = 10$), where experience replay is

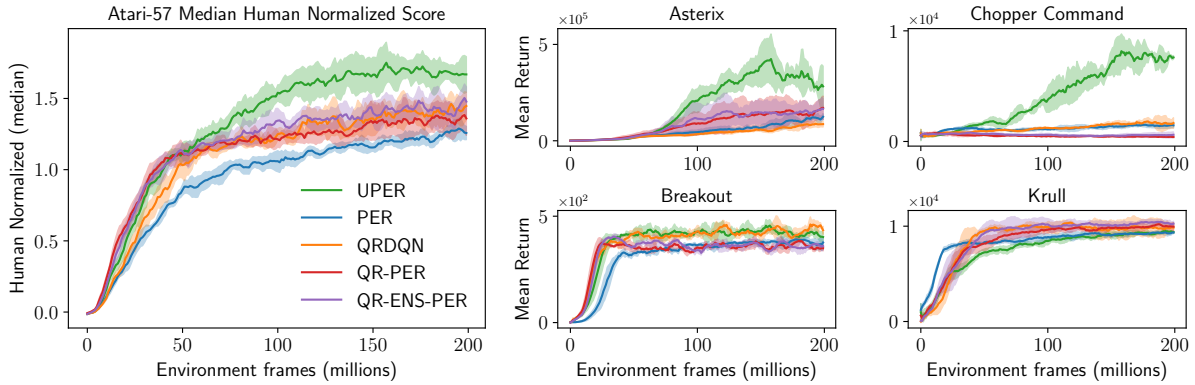

Figure 5.3: (**Left**) Comparing Uncertainty Prioritized Experience Replay (UPER) with Prioritized Experience Replay (PER) and QR-DQN on the full Atari-57 benchmark. Median human normalized score for UPER is significantly higher than baselines throughout the learning trajectory. (**Right**) Example of per-game performance, with vastly superior performance on e.g. Asterix and Chopper Command; cases in which UPER is worse are far less extreme, for instance Breakout and Krull (this is shown graphically in figure B.11 and figure B.12). All results are averages over 3 seeds.

prioritized using the information gain criterion (UPER, as described in subsection 5.2.2). This method is compared against a vanilla QR-DQN agent (Dabney et al., 2017) with uniform prioritization and the original PER agent (Schaul et al., 2016).

To demonstrate that the performance improvement is not solely due to the quantile regression method or the ensemble, additional comparisons are conducted. Specifically, a QR-DQN agent with TD-error prioritization (QR-PER) and an ensemble of QR agents with TD-error prioritization (QR-ENS-PER) are trained. A summary of the empirical results is presented in figure 5.3, with further ablations and details provided in section B.5.

Except for the additional hyperparameters associated with the ensemble of distributional prediction heads and a commonly used configuration for the Adam optimizer ($\epsilon = 0.01/(\text{batch\_size})^2$), the network architecture and all hyperparameters in UPER remain identical to those in QR-DQN (Dabney et al., 2017). Similarly, the PER, QR-DQN, and QR-PER baselines follow the implementations of Dabney et al. (2017) and Schaul et al. (2016), respectively, while QR-ENS-PER is identical to UPER except for the prioritization

variable, which is based on TD-error.

Concretely for the UPER agent, the target epistemic uncertainty is computed using $\hat{\mathcal{E}}_\delta(s,a) = \hat{\mathcal{U}}_\delta(s,a) - \hat{\mathcal{A}}(s,a)$. Then for a given transition i the total uncertainty is given by

$$\hat{\mathcal{U}}_\delta = \mathbb{E}_{\tau,\tau',\psi}\left[\left(r_i + \gamma\theta_{\tau'}(s_i', a_i'; \bar{\psi}) - \theta_\tau(s_i, a_i; \psi)\right)^2\right], \tag{5.14}$$

where τ (τ') are the quantiles of the online (target) network ψ ($\bar{\psi}$). The aleatoric uncertainty estimate is given by $\hat{\mathcal{A}}(s,a)$ in equation 2.47. Based on these estimates, the UPER priority variable is constructed using the uncertainty ratio discussed in subsection 5.2.2, i.e., equation 5.12. Since both UPER and QR-ENS-PER are ensemble-based agents, a random mask $m \in \mathbb{R}^N$ is stored for each transition in the replay buffer, where each element is sampled as $m_i \sim \mathcal{B}(0.5)$. When a transition is sampled for learning, gradients are propagated only for heads whose corresponding mask element is equal to 1. This follows the approach proposed by (Osband et al., 2016) and serves to decorrelate the learning trajectories of the ensemble members, which is essential for obtaining valid uncertainty estimates.

As shown in figure 5.3, the median performance of UPER across games is significantly better than that of other prioritization schemes, demonstrating that the observed performance improvement is not solely due to the quantile regression technique or the ensemble. Notably, UPER outperforms its closest comparison, QR-ENS-PER, which differs from UPER only in its use of TD-error for prioritization (see figure B.11). In most games where UPER does not yield a performance improvement, such as Krull, Q*bert, or H.E.R.O., the performance difference is not statistically significant. This is further illustrated in the per-game panels in figure B.12 and the asymmetry of the bar plots in section B.5.

Table 5.1: Computational Cost (seconds per iteration)

Architecture	CPU	GPU
QR-DQN-ENS	28.40 ± 0.26	20.74 ± 0.43
QR-DQN	17.80 ± 0.13	18.49 ± 0.68
DQN	18.34 ± 0.09	18.39 ± 0.56

5.5 Discussion

In this study, epistemic uncertainty measures are proposed to guide the prioritization of transitions from the replay buffer. Through both mathematical analysis and carefully designed experiments, it is demonstrated that the commonly applied TD-error criterion can incorporate aleatoric uncertainty, leading to the over-sampling of noisy transitions. Prioritizing transitions based on a principled function of epistemic and aleatoric uncertainty, formulated as information gain, mitigates these effects.

To construct this function, the concept of epistemic uncertainty from Clements et al. (2020) is extended to incorporate the distance to the target, resulting in performance advantages in both toy settings and complex problems, such as the Atari 57 benchmark. A potential concern when estimating these auxiliary quantities is the increased computational cost in deep learning settings. However, sharing lower-level representations across multiple heads, combined with efficient implementations, can significantly reduce this overhead.

To illustrate this, an experiment was conducted on a lower-capacity GPU to compare the training times of DQN, QR-DQN, and QR-DQN + ensemble networks in the Pong environment. The time per iteration is reported in Table 5.1.

The comparable training times can be attributed to effective batch processing facilitated by GPU parallelization. In the proposed implementation, each agent in the ensemble is represented by a distinct output head within the network architecture. By extending the batch dimension to $(\text{batch}, \text{action}, \text{quantiles}, \text{ensemble})$, the parallelization capacity of the GPU is leveraged while remaining within its operational limits for the QR-DQN ensemble

network. Further details of this experiment, as well as the computer architecture used, are provided in subsection B.5.2. It is important to note that this analysis does not aim to compare the computational cost of sampling with a priority variable versus uniform sampling, as this aspect has already been addressed in the original PER paper and is known to have a negligible impact.

While this implementation focuses on distributional reinforcement learning, a widely used class of methods, exploring alternative forms of uncertainty estimation, such as pseudo-counts (Lobel et al., 2023), in combination with different functional forms beyond information gain presents a promising research direction. These approaches could benefit not only prioritization schemes but also other aspects of the reinforcement learning problem, such as exploration (see section B.2 and section B.3).

The framework introduced in this work, which combines epistemic and aleatoric uncertainties through information gain, is not limited to reinforcement learning. In principle, these concepts can be extended to other learning systems. A substantial body of literature explores efficient data selection to improve learning in paradigms such as supervised learning (Hüllermeier and Waegeman, 2021; Zhou et al., 2022), continual learning (Henning et al., 2021; Li et al., 2021), and active learning (Nguyen et al., 2022). Additionally, this work has the potential to provide alternative insights into replay events in biological agents (Daw et al., 2005; Mattar and Daw, 2018; Liu et al., 2019; Antonov et al., 2022).

5.5.1 Replay and Exploration Alternatives

Exploration. While UPER does not explicitly promote exploration through a reward bonus for unexplored or uncertain states, it leverages methods from this field to estimate epistemic and aleatoric uncertainty (Clements et al., 2020) and prioritize transitions in the replay buffer based on information gain.

A fundamental challenge faced by reinforcement learning agents is the exploration-

exploitation trade-off (Osband et al., 2016; O'Donoghue, 2023), wherein agents must balance two competing objectives when selecting actions: acquiring new information about the environment (exploration) and maximizing immediate reward based on current knowledge (exploitation). Both replay sampling and exploration strategies influence the data used to refine value function estimation. The former determines which past experiences contribute to value updates, while the latter governs the experiences that populate the replay buffer.

Many exploration strategies have been developed around the concepts of intrinsic reward (Oudeyer and Kaplan, 2007) and episodic memory (Savinov et al., 2019; Badia et al., 2020). However, these approaches are vulnerable to pathological behaviors, such as the noisy TV problem. Later variants have been designed to mitigate these issues and often emphasize obtaining reliable and meaningful estimates of counts and novelty (Ostrovski et al., 2017; Bellemare et al., 2016; Burda et al., 2018; Lobel et al., 2023), dynamics (Stadie et al., 2015; Pathak et al., 2017), uncertainty (Mavor-Parker et al., 2022), and related quantities. Many of these considerations are directly relevant to the challenge of constructing effective measures for replay prioritization.

PER. Since its introduction by Schaul et al. (2016), various efforts have been made to understand and improve prioritized experience replay. The integration of uncertainty-related information has often been explored in conjunction with strategies for managing the exploration-exploitation trade-off. For instance, Sun et al. (2020) propose sampling frequently visited states more often to reduce uncertainty in well-explored regions. Conversely, Alverio et al. (2022) prioritize uncertain states to encourage exploration, using epistemic uncertainty estimated as the standard deviation across an ensemble of next-state predictors. This approach is combined with other techniques to enhance sample efficiency.

Another method, presented by Lobel et al. (2023), employs a pseudo-count approximation to estimate state visit frequencies, fostering exploration through an intrinsic reward

mechanism. During training, transitions are prioritized based on these counts. However, this approach does not extend prioritization to learning the actual value network, which is the primary focus of this study. The method proposed by Lobel et al. (2023) enables the estimation of epistemic uncertainty independently of the reward signal's sparsity or density, making it particularly appealing in sparse-reward environments. However, using pseudo-counts to estimate epistemic uncertainty may not always align well with the true uncertainty in value estimation (Osband et al., 2018). As discussed in subsection 5.3.1, the frequency of visits to a given state-action pair does not necessarily reflect the error between the estimated and true value. Moreover, as explained in subsection 5.2.2 and demonstrated through simulations in subsection 5.3.1, both epistemic and aleatoric uncertainty should be considered when constructing an effective prioritization scheme.

Chapter 6

General Discussion

In this thesis, the learning effort framework is presented (see chapter 3). This formal framework provides a way to formulate the problem of controlling the learning process of a system by considering the influence of a control signal on a dynamical system that describes learning. The framework is general enough to instantiate other machine learning algorithms, such as MAML (Finn et al., 2017), and solves the expected value of control in the regulation of a learning system (Shenhav et al., 2013; Masís et al., 2021).

The framework is applied to a variety of settings, where control assumes different forms and the agent is exposed to various tasks, all governed by the same optimization process using gradient descent on the control signal. One of the applications of the learning effort framework involves a simulation of rats solving a binary classification task (Masís et al., 2023). In this context, an alternative model is proposed, incorporating highly abstract variables that impose trade-offs between immediate rewards and improved learning at the cost of early rewards. The optimal control for this model aligns with the reaction times observed in rats, suggesting that control can be understood as the inhibition of default or impulsive behavior. Rats that exerted greater control (demonstrated by longer reaction times during early learning) learned faster and accumulated higher cumulative rewards on the task.

The proposed *epistemic noise control* model described in subsection 3.7.2, developed within the learning effort framework, conceptualizes control as a reduction of epistemic uncertainty, achieved by incurring a cost in terms of reduced instantaneous reward rate. This model collapses the evidence collection process and it is analogous to extending the duration of stimulus observation in each trial (see section 3.7). The ability of this proposed model, in addition to the original model introduced by the authors of the experiment, to replicate behavioral data suggests a fundamental dilemma in controlling learning: early rewards versus improved learning at a cost. It is important to note that the relation between control, reaction time, and noise in this model differs from the more widely accepted view in the literature, where higher cognitive control is generally associated with shorter reaction times and, depending on the experimental design, may also be linked to increased accuracy (Gratton et al., 1992; Ullsperger et al., 2005; Ulrich et al., 2015). The connection between this mainstream interpretation of cognitive control and the epistemic noise control model is discussed further in subsection 3.7.4.

Most models discussed in chapter 3, exhibit an optimal control signal that is stronger early in learning and gradually diminishes as the model acquires proficiency in the task, thereby requiring less control. In chapter 4, the thesis focuses on examining simplified instances of learning control, starting from previously proposed models for this problem. An analysis based on policy optimality in a reinforcement learning framework reveals the underlying reasons for the numerically observed optimal strategy. Specifically, most strategies allocate more control at the beginning of learning, a pattern encapsulated in the conjecture referred to as *learn first, do later* (subsection 4.2.3). While this idea is intuitive, it can be analytically justified through optimality arguments and fundamental properties of learning trajectories, such as monotonicity, i.e., the more time allocated to a task, the greater the improvement in performance (Ritter and Schooler, 2001; Son and Sethi, 2006, 2010; Masís et al., 2021).

The *learn first, do later* conjecture is tightly related to the concept of *automaticity*,

which corresponds to the transition of an agent from a more controlled, effortful behavior when learning, to a more automatic, efficient performance that is minimally affected by task-unrelated information (Hélie and Cousineau, 2014). Automaticity effects can be found in many aspects of learning, e.g., classical conflict in automatic response for the Stroop task (Cohen et al., 1990; Musslick et al., 2020), automaticity as memory retrieval (Logan, 1988; Rickard, 1997; Hélie et al., 2010), or motor skill automaticity (Wulf et al., 2001; Poldrack et al., 2005; Christiansen et al., 2020). Common hallmarks of the automaticity phenomenon in learning can be found across applications, such as a reduction in mean RTs and stability of RTs, saturated performance on the task (for different RTs), and less susceptibility to distractions (Moors and De Houwer, 2006). The presented conjecture uses a normative objective to show that it is optimal to exert highly controlled engagement on a task when the agent can improve on the task and collect more reward with practice. However, as this conjecture is still highly abstract to allow for a proof of the optimal learning policy, other aspects of the learning process regarding experimental setups, such as trial dynamics, performance, and agents' reaction times, are not yet described by this setup.

Although the discrete models used for the conjecture are relatively simple, they can be interpreted as discrete approximations of a fully time-continuous model, where the control signal takes a range of continuous values, as in the learning effort framework. While the optimality arguments may not directly apply to the continuous and more complex formulation of the control problem, they could serve as useful approximations for analyzing more intricate control problems and learning systems.

The optimality argument used to support the conjecture does not apply to the time-continuous version of the problem. However, it is still possible to extend the formal analysis. To achieve this, subsection 4.3.3 introduces the homotopy perturbation method (He, 1999; Liao, 2003) to approximate an optimal control signal, particularly in the context of learning rate scheduling. This method relies on deriving the Hamilton-Jacobi-Bellman

equation (Atangana et al., 2014) for the optimization problem and provides optimality guarantees along with an expression for the control signal as a function of the value. The resulting formulation leads to a partial differential equation that, in general, cannot be solved in closed form.

To address this, a homotopy is constructed, meaning that a new equation is formulated to interpolate between the full Hamilton-Jacobi-Bellman equation and a simpler, solvable approximation. The value function and control signal are then derived in terms of modes, where higher-order modes yield increasingly accurate approximations in a Taylor expansion-like manner. However, these modes tend to diverge rapidly, as they form polynomials in the embedding parameter of the homotopy. To mitigate this issue, a final step involves applying a Padé approximation to the resulting control, which enhances the quality of the estimation (Ganjefar and Rezaei, 2016).

The Padé approximation provides a concise analytical expression for the control signal, offering insight into how control depends on model parameters and task statistics. Additionally, this control is computed online, meaning that it functions as a closed-loop feedback mechanism that can be applied to the dynamical system during learning. The dependence on model parameters, such as time discount and task difficulty, follows expected relationships, supporting the approximation method as a valuable analytical tool for studying optimal control of learning.

If theoretical models are too abstract to match experimental settings, why is it still necessary to perform mathematical analysis of simplified abstract models? Many numerical models describing an agent performing a task can serve as surrogate systems for biological agents. These models allow researchers to test different setups through simulations by varying architecture, optimization objectives, learning rules, and other design parameters. The results can then be compared with experimental measurements to validate the model and provide a mechanistic explanation of the agent's internal processing in an experiment,

which is arguably a powerful approach to *understanding* how agents work (Lillicrap and Kording, 2019; Richards et al., 2019; Botvinick et al., 2020; Niu et al., 2024).

However, the simulation approach alone is not sufficient; mathematical analysis is also necessary for several reasons, a view that has been supported in the literature (Eliasmith and Trujillo, 2014; Saxe et al., 2021). One advantage of mathematical analysis is the potential access to guaranteed gold-standard solutions, which is often not the case with numerical methods. Although simulated agents and biological agents in experiments may not reach this theoretical optimality, it is valuable to understand why this occurs and what alternative methods or heuristics could approximate the gold standard. Another advantage of mathematical analysis is the availability of transparent expressions that make it possible to identify the variables most relevant for optimal control without running simulations. Such expressions also allow researchers to systematically explore the space of relevant variables and their effects on the optimal solution, providing guarantees of functionality and strengthening the conclusions drawn. In contrast, attributing changes in simulations can be difficult, even when controlling for a single variable, because the model may be operating in a special regime determined by other, unaltered variables. As noted earlier, theoretical models often differ substantially even from simple computational models, which themselves can be far removed from the behavior of biological agents. Bridging the gap between theoretical, computational, and real agents remains an important research objective, and it is essential to acknowledge that each approach operates at a different level of abstraction in explaining and describing intelligent systems (Levenstein et al., 2023; Shankar et al., 2025).

In summary, both grounded simulations validated through experimental data and mathematical analysis in simplified, tractable models are needed. These complementary approaches provide best-case scenarios for understanding and advancing theories of learning and control.

Finally, in chapter 5, the thesis develops a new method to estimate epistemic uncertainty in environmental transitions. This concept is closely related to the epistemic control model presented in section 3.7, and more generally to the overall meta-learning framework in chapter 3. In this context, epistemic uncertainty refers to the error that can be reduced through learning (Lahlou et al., 2022), thereby providing a proxy for prioritizing replay based on this uncertainty. The proposed method, termed Uncertainty Prioritized Experience Replay (UPER, chapter 5), leverages this idea by using epistemic uncertainty as a guiding signal. Rather than relying on detailed representations of the learning trajectory, which would demand additional meta-knowledge about the agent, its learning dynamics, and the statistics of the task, this approach offers a simpler heuristic to guide learning. In practice, prioritizing replay through epistemic uncertainty serves as a practical surrogate for controlling learning in more complex models, improving performance without requiring explicit modeling of the entire trajectory.

The uncertainty estimator is based on distributed reinforcement learning (Dabney et al., 2017), which approximates the underlying data distribution, and ensembles (Osband et al., 2016; Dwaracherla et al., 2022), which offer a practical means of sampling a posterior distribution of estimators. Compared to previous approaches, this estimation method accounts for the bias toward the target value that needs to be estimated, rather than solely considering uncertainty. More importantly, the experiments conducted aim to isolate the effect of the prioritization variable. In contrast, prior work using uncertainty estimators has often included confounding factors in the agent design, preventing the isolation of uncertainty's effect. Results from multiple experiments, including the Atari-57 benchmark, indicate that prioritizing replay based on information gain enhances the agent's overall learning performance.

6.1 Future Ideas

Learning effort framework

After thoroughly examining the capacity of the formal framework to represent different control setups, a natural direction for future research is to apply it to specific systems in the brain. One clear example is an adaptation of the Stroop task, where control not only resolves interference between task representations but also accounts for its impact on learning (Cohen et al., 1990; Musslick et al., 2020). Similarly, this framework could be applied to prioritized replay, potentially in a system consolidation setting, as explored in Sun et al. (2023), to further investigate the types of content that are stored or consolidated.

Computing the optimal control is challenging and not biologically plausible, yet humans and other animals appear capable of estimating these quantities (Ten et al., 2021; Masís et al., 2023; Masis et al., 2024), suggesting that heuristics may be employed to allocate control for learning. Prototypes of episodic memory-based control estimation have been tested, in which control is estimated based on prior learning experiences, similar to other approaches like Pritzel et al. (2017) and Ritter et al. (2018). However, this approach is not included in the thesis, as it requires further analysis.

Another potential direction is leveraging the general properties of learning trajectories, as indicated by Ritter and Schooler (2001); Son and Sethi (2006, 2010); Masís et al. (2021), to develop a *model-based control of learning estimator*. Given the common characteristics of learning trajectories, it may be possible to constrain the space of potential optimal control signals, thereby simplifying the problem.

Beyond extending the applications of the meta-learning framework to find optimal control policies under different scenarios, this framework also has the potential to be applied to phenomena in cognitive neuroscience, as discussed throughout the thesis. Examples include the concept of automaticity in learning (Hélie and Cousineau, 2014), engagement

based on learning difficulty (Wilson et al., 2019), curriculum learning (Kidd et al., 2012; Raz and Saxe, 2020; Ten et al., 2021), exploration heuristics such as intrinsic motivation (Bromberg-Martin et al., 2024), and speed–accuracy trade-offs, as described in section 3.7.

Further work is necessary to use the framework to understand psychological phenomena, but concrete predictions can already be derived from the framework and its mathematical analysis. First, the amount of control or effort in learning is predicted to decrease with more delayed rewards or with higher future reward discounting. Previous experiments have accounted for reward time discounting (Frederick et al., 2002; Kable and Glimcher, 2007; Shadmehr et al., 2010), but in most of this work, discounting is studied in environments where improvement or learning on a task is not explicitly involved. Other research has controlled for delayed reward and feedback (a manipulation similar to changing the discount rate when the delay is fixed), showing that such delays impair learning (Yin et al., 2018). Furthermore, delaying reward has been shown to reduce goal-directed actions (Perez and Urcelay, 2025), suggesting a reduction of control with delayed rewards, although not specifically referring to control of learning. Further experimentation is required to account for the interaction between controlled learning, delayed rewards, and time discounting.

Another prediction from the meta-learning framework, derived from its connection with the EVC theory, is the reduced amount of control over learning when the cost of control increases. Regardless of how control is defined in the framework (as discussed in subsection 3.4.1), the specific form of the control cost function will affect the optimal control signal and, consequently, its impact on learning.

It has been shown that exerting effort (e.g., through attentional state) can improve learning (Yu and Dayan, 2005; Eldar et al., 2013; Gottlieb, 2012). Perhaps the phenomenon most closely relating attentional states and learning is the flow state, in which subjects report full absorption in a task along with a sense of control and enjoyment. Studies have

shown that enhanced feedback on learning progress helps maintain task engagement and cognitive control, thereby improving learning outcomes (Biasutti, 2017; Lu et al., 2025). Interestingly, the analytical results from section 4.3 for the optimal control derived using the homotopy perturbation method are proportional to the squared gradient of the loss function with respect to the weights, and are therefore larger when learning progress is greater.

In some cases, optimal engagement in learning a task can be regulated by its difficulty, i.e., by adapting the task's difficulty to keep the subject engaged. So far, the meta-learning framework has modeled task difficulty as irreducible noise, which reduces the bound on maximum potential performance and slows down learning. As shown in simulations, increasing this notion of difficulty reduces the amount of control, as it is not worth investing effort in a task that cannot be learned. This is in contrast to what has been reported in human subjects, where there appears to be a sweet spot in difficulty that promotes task engagement, hence it is necesary to find an appropiate definition of task difficulty within the learning effort (Wilson et al., 2019).

Some evidence suggests that the exertion of effort itself may improve learning, as in (Jarvis et al., 2022). However, in that case the effort was physical and did not necessarily alter the information-processing mechanisms relevant to the task. Instead, effort placed the subject in a state that indirectly enhanced learning. Additionally, it has been shown that the perception of physical effort correlates with the perception of mental labor, suggesting that effort *in general* may influence information processing and learning (Bustamante et al., 2023).

In the meta-learning framework, the cost of control is additive to the reward obtained from the environment, which is also a feature of the EVC theory. Interestingly, it has been suggested that reward delay can be functionally equivalent to an increase in the effort required to obtain a reward, leading to similar behavioral outcomes despite being

encoded in different brain regions: reward delay in the ventral striatum and ventromedial prefrontal cortex, and effort in the anterior cingulate cortex (Prévost et al., 2010).

An appropriate experimental setting to test the meta-learning framework should account for the effect of control on learning dynamics while decoupling the motivation to exert control for immediate trial-by-trial improvements. For instance, applying cognitive control to reduce reaction time (RT) or improve accuracy on a single trial may incidentally improve learning, but this does not imply that control is exerted because of its contribution to learning improvement. In such cases, control is applied merely to optimize performance on an individual trial.

Along these lines, recent experimental setups appear promising for testing the meta-learning effort framework and examining the effect of cognitive control on learning. In a pre-registered experiment (Sandbrink et al., 2024), subjects could either *observe* (check whether the hidden state of the trial is favorable) or *bet* (collect the reward if the hidden state of the trial is favorable without observing it). This experiment was designed to measure exploration–exploitation trade-offs in humans when the environment is volatile.

To adapt this paradigm for testing the proposed meta-learning framework, it would be necessary to introduce a form of reward improvement on each trial instead of volatility. For example, the action of *bet* could be replaced with *collect* (receive a reward), and a new action *improve* could be added (increase the eventual reward from the *collect* action, or increase the improvement rate; no immediate reward, or even a negative reward, is given). This modified setup closely resembles the framework used for mathematical analysis in subsection 4.2.3. For a fixed number of trials, the cost of control would be the opportunity cost of choosing the *improve* action, since no reward is collected in contrast to the *collect* action. In this way, the process of *learning* is abstracted and replaced by *improvement*.

This experimental design could manipulate the number of available trials in a session (time available for improvement), improvement dynamics, cost of improvement, improvement

monitoring, and reward volatility. Each of these factors would influence the optimal control signal, defined as the number of times a subject chooses to *improve* instead of *collect*. Importantly, while this setting could, in principle, provide all the meta-cognitive information necessary to compute the optimal control signal, it remains to be tested whether humans and animals can actually gather and utilize such information in natural learning scenarios. In real-world tasks, the estimation of improvement must be based on their own learning dynamics, rather than a monotonic, externally defined reward improvement detached from the learning process.

An example of a direct measure of control allocation for learning is provided in (Masis et al., 2024), where it is shown that human subjects choose to deliberate longer on each trial (a costly action due to opportunity cost) when the task is perceived as learnable, and deliberate for a shorter time when the task is not learnable. The author proposed a model based on a drift–diffusion model (DDM), similar to the one described in subsection 3.7.1. A comparable approach can be taken to test the proposed framework, now by modifying the model in subsection 3.7.2 with the considerations discussed in subsection 3.7.4.

Analytical Methods

The extent to which the homotopy perturbation method can be applied depends on the complexity of the Hamilton-Jacobi-Bellman equation, which varies across different learning systems and control implementations. Some explored approaches (not included in this thesis) involve describing control as a vector of scalars, where each entry corresponds to a learning rate specific to the output of a single-layer network. Another extension that has been investigated is the learning rate scheduling of a two-layer linear network, although no positive results have been obtained thus far.

An alternative to the standard gradient flow description of the two-layer network is to use the *order parameters* formulation, as in a teacher–student setup (Goldt et al., 2019; Lee et al., 2022; Carrasco-Davis and Grant, 2025). Since the results from the approximation

using the homotopy method approach is weight dependent, it is difficult to interpret. By incorporating order parameters, which summarize the state of the learning system, it may be possible to explain optimal control as a function of summary statistics of the initial conditions of the system. This perspective could also provide insights into better strategies for setting hyperparameters in neural networks. Additionally, incorporating nonlinear constraints on the control signal could lead to interesting applications, though these constraints are mathematically challenging to analyze. For instance, enforcing the control signal to lie on a sphere, where each axis projection represents attention to a specific task, could model a scenario in which attending to one task comes at the expense of learning others. This setting is conceptually similar to one of the cases discussed in section 3.5. While this approach shows promise, it involves intensive algebraic computations and requires further study.

Uncertainty Prioritized Experience Replay

A key contribution of this work is the proposed estimator for epistemic and aleatoric uncertainty (see section 5.2). Most methods for uncertainty estimation in reinforcement learning are applied in exploration-exploitation trade-off scenarios (Pathak et al., 2017; Boldt et al., 2019; Clements et al., 2020; Lobel et al., 2023). Consequently, a natural next step is to evaluate this estimator, for example, as an intrinsic motivation mechanism in tasks that require extensive exploration.

A well-established theory of prioritized replay suggests that events are replayed based on their value following a temporal difference (TD) update (Mattar and Daw, 2018). This theory could be tested using the learning effort framework, as it follows the same fundamental premise, but in this case, the learning system is a neural network. Furthermore, the theory of prioritized replay could be extended to account for the uncertainty associated with the value obtained from updates. This extension could broaden the set of predictions derived from the theory, incorporating aspects such as risk aversion and

confidence estimation in replay content.

UPER is not limited to RL applications. Evidence from neuroscience suggests that dopaminergic neurons encode a distribution of value functions (Dabney et al., 2017; Lee et al., 2024a). Additionally, probabilistic approaches have proposed mechanisms by which the brain may compute posterior distributions (Pouget et al., 2013; De Martino et al., 2013; Sohn and Narain, 2021), which are necessary for uncertainty estimation. A promising direction is then studying potential mechanisms of biological neural networks to estimate uncertainty, specifically how the separation between epistemic and aleatoric uncertainty could be computed.

A broader follow-up question to the work presented in this thesis is how the brain develops tools for meta-cognitive abilities. For example, part of the information required to compute optimal control within the learning effort framework includes the learning dynamics of the agent, an integration of learning over time, and noise estimations of the data. In the uncertainty-prioritized replay application, the separation between epistemic and aleatoric uncertainty requires computing ensemble disagreement. These quantities are properties of the learning system itself and thus represent a form of meta-cognition.

Meta-cognition is the ability of an agent to *be aware of it's own cognitive processes* and to use that information to regulate them (Fleur et al., 2021). It is required to appropriately control learning and to estimate epistemic uncertainty, as previously discussed. Meta-cognition has been broadly studied in cognitive neuroscience. In a review by Fleming (2024), the author seeks to unify the concept of confidence in neuroscience, mostly focused on the visual or motor systems at a *subpersonal-level* representation of uncertainty, with the concept of meta-cognition in cognitive science, which corresponds to beliefs and the agent's knowledge about its own performance, at a *personal-level*, also called *propositional confidence* (Pouget et al., 2016).

This propositional confidence can be constructed through an inferential process: first,

computing a posterior distribution of possible stimuli based on a measurement (sensory uncertainty), then computing the probability of each scenario and set of actions given that sensory uncertainty. Hence, propositional confidence corresponds to the probability that each scenario, or a specific action, is the correct one to achieve a goal. This propositional confidence is not fully determined by sensory uncertainty, as it can also be corrected by a *self-model*, which provides knowledge of the inner workings of the agent's own perceptions and mental processes (Nelson, 1990; Olop et al., 2024).

This work provides a useful framework to understand how an agent could decompose epistemic and aleatoric uncertainty. For instance, sensory uncertainty could correspond to an estimate of aleatoric uncertainty. Knowledge about how learning works, i.e., a self-model, such as improvement with practice, and the existence of noisy distractions, such as a *noisy TV*, could provide knowledge of epistemic uncertainty. This information is then integrated to generate propositional confidence, in the form of an appropriate policy for exploring a noisy environment.

Similar mechanisms could be applied to the meta-learning framework when inferring optimal control over learning. Some of the required information is available at the sensory level, e.g., stimulus noise or task features, while other information is at the self-model level, such as the effect of control over learning dynamics, presumably learned through multiple learning opportunities or by instruction (Olop et al., 2024), understanding the opportunity costs of engaging in a task (Agrawal et al., 2022), weighting against aversion to cognitive load (Kool and Botvinick, 2018), or even considering the opportunity cost of the time required to compute an optimal policy for a given trial or overall task (Gershman and Burke, 2023). Further testing of both the meta-learning framework and the uncertainty decomposition of experiences requires considering the agent's access to its own meta-cognitive information.

While it is reasonable to assume that complex intelligence models might achieve meta-

cognition due to their large capacity and the pressure of solving problems that require knowledge of itself, it remains unclear how such abilities emerge from training or what computational principles underlie meta-cognition. Models of the allocation of control in learning have been proposed, such as a hierarchical RL model where higher levels in the hierarchy correspond to more abstract meta-cognitive information (Silvetti et al., 2023), learning control based on episodic memory generalization (Giallanza et al., 2024), or rule-based control guided by instructions or imitation (Fleur et al., 2021; Olop et al., 2024).

Broader Speculations

The learning effort framework provides a means to compute optimal control that maximizes cumulative reward throughout learning. This control is presumably exerted by high-level executive functions, which is reasonable given that it requires more complex computations compared to the simpler process of learning. This suggests that, since higher-level executive functions develop later, both in developmental and evolutionary time scales, any form of control implemented in the brain may be tightly constrained by the pre-existing dynamics of neural networks.

This hypothesis is further supported by the use of control to resolve ambiguities, as seen in the Stroop task, and to mitigate intrinsic computational challenges in neural networks, such as interference between task representations (Musslick and Masís, 2023) or stability vs flexibility trade-offs in attentional stated (Dreisbach et al., 2024). **Under this perspective, control may generally exist to overcome the limitations of the neural substrate in implementing computations**, an idea previously explored by Musslick et al. (2020); Musslick and Masís (2023).

The primary discussion has focused on the control of learning, specifically the optimization problem presented in chapter 3. This can be further generalized or framed as the *effect of a relevant variable on a dynamical system*, meaning that control is not necessarily

restricted to learning. Alternative formulations could include optimizing an architecture that remains advantageous throughout an agent's lifespan or meta-optimizing inputs to a network such that the induced activity remains useful over a relevant time frame, an approach directly related to replay. In other words, aspects beyond control could be examined within a meta-learning framework, where the dynamical system resembles a brain-like structure with its respective time scales. For instance, shorter time scales, perhaps within decision-making, could correspond to the control of decision-making as discussed in this work. Longer time scales in the meta-objective could help elucidate advantages in development occurring throughout an organism's lifespan, and even longer time scales could be considered to study emergent evolutionary advantages. Complex brain functions, such as replay (Mattar and Daw, 2018; Thompson et al., 2024), specific architectures or modularity (Russo et al., 2014; Saxe et al., 2022), or preparatory periods in the motor cortex (Churchland et al., 2010), may simply emerge as by-products of developmental or evolutionary solutions. These solutions are constrained by the fact that computations are implemented in a neural network, which is subject to a given dynamics defined by its physical constraints. Notably, the objective function that maximizes reward collected within a given time frame is highly expressive (Abel et al., 2021; Silver et al., 2021). It can be leveraged to construct additional meta-objectives operating on longer time scales or even fitness-like objectives akin to those in evolutionary computation.

The learning effort framework provides a tool for studying the control of learning and could be applied to other phenomena in cognitive neuroscience and machine learning by offering a naturalistic normative framework for analyzing features observed in biological systems. **The study of learning control is entangled with learning itself, meaning that learning will never be fully understood without elucidating its control counterpart.**

References

David Abel, Will Dabney, Anna Harutyunyan, Mark K Ho, Michael Littman, Doina Precup, and Satinder Singh. On the Expressivity of Markov Reward. In *Advances in Neural Information Processing Systems*, volume 34, pages 7799–7812. Curran Associates, Inc., 2021. URL https://papers.neurips.cc/paper/2021/hash/4079016d940210b4ae9ae7d41c4a2065-Abstract.html.

Mayank Agrawal, Marcelo G. Mattar, Jonathan D. Cohen, and Nathaniel D. Daw. The temporal dynamics of opportunity costs: A normative account of cognitive fatigue and boredom. *Psychological Review*, 129(3):564–585, 2022. ISSN 1939-1471. doi: 10.1037/rev0000309. Place: US Publisher: American Psychological Association.

Shams Forruque Ahmed, Md. Sakib Bin Alam, Maruf Hassan, Mahtabin Rodela Rozbu, Taoseef Ishtiak, Nazifa Rafa, M. Mofijur, A. B. M. Shawkat Ali, and Amir H. Gandomi. Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, 56(11):13521–13617, November 2023. ISSN 1573-7462. doi: 10.1007/s10462-023-10466-8. URL https://doi.org/10.1007/s10462-023-10466-8.

Tobias Alfers, Georg Gittler, Esther Ulitzsch, and Steffi Pohl. Assessing the Speed–Accuracy Tradeoff in Psychological Testing Using Experimental Manipulations. *Educational and Psychological Measurement*, 85(2):357–383, April 2025. ISSN

0013-1644. doi: 10.1177/00131644241271309. URL https://doi.org/10.1177/00131644241271309. Publisher: SAGE Publications Inc.

Julian Alverio, Boris Katz, and Andrei Barbu. Query The Agent: Improving sample efficiency through epistemic uncertainty estimation, October 2022. URL http://arxiv.org/abs/2210.02585. arXiv:2210.02585 [cs].

Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, March 2021. ISSN 2196-1115. doi: 10.1186/s40537-021-00444-8. URL https://doi.org/10.1186/s40537-021-00444-8.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural Module Networks, July 2017. URL http://arxiv.org/abs/1511.02799. arXiv:1511.02799 [cs].

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Dora E Angelaki and Jean Laurens. The head direction cell network: attractor dynamics, integration within the navigation system, and three-dimensional properties. *Current Opinion in Neurobiology*, 60:136–144, February 2020. ISSN 0959-4388. doi: 10.1016/j.conb.2019.12.002. URL https://www.sciencedirect.com/science/article/pii/S0959438819301370.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable

Second Order Optimization for Deep Learning, March 2021. URL http://arxiv.org/abs/2002.09018. arXiv:2002.09018 [cs].

Georgy Antonov, Christopher Gagne, Eran Eldar, and Peter Dayan. Optimism and pessimism in optimised replay. *PLOS Computational Biology*, 18(1):e1009634, January 2022. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009634. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009634. Publisher: Public Library of Science.

L. A. Apresyan. Pade approximants (review). *Radiophysics and Quantum Electronics*, 22(6):449–466, June 1979. ISSN 1573-9120. doi: 10.1007/BF01081220. URL https://doi.org/10.1007/BF01081220.

John Asmuth, Lihong Li, Michael L. Littman, Ali Nouri, and David Wingate. A Bayesian Sampling Approach to Exploration in Reinforcement Learning, May 2012. URL http://arxiv.org/abs/1205.2664. arXiv:1205.2664 [cs].

Gary Aston-Jones and Jonathan D. Cohen. An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance. *Annual Review of Neuroscience*, 28:403–450, 2005. ISSN 0147-006X. doi: 10.1146/annurev.neuro.28.061604.135709.

Abdon Atangana, Aden Ahmed, and Suares Clovis Oukouomi Noutchie. On the Hamilton-Jacobi-Bellman Equation by the Homotopy Perturbation Method. *Abstract and Applied Analysis*, 2014(1):436362, 2014. ISSN 1687-0409. doi: 10.1155/2014/436362. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2014/436362. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2014/436362.

Peter Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002. ISSN ISSN 1533-7928. URL https://www.jmlr.org/papers/v3/auer02a.html.

Zainab Ayati and Jafar Biazar. On the convergence of Homotopy perturbation method. *Journal of the Egyptian Mathematical Society*, 23(2):424–428, July 2015. ISSN 1110-256X. doi: 10.1016/j.joems.2014.06.015. URL https://www.sciencedirect.com/science/article/pii/S1110256X14000923.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, and Charles Blundell. Never Give Up: Learning Directed Exploration Strategies, February 2020. URL http://arxiv.org/abs/2002.06038. arXiv:2002.06038 [cs, stat].

Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-Learning with Adaptive Hyperparameters. In *Advances in Neural Information Processing Systems*, volume 33, pages 20755–20765. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc//paper/2020/hash/ee89223a2b625b5152132ed77abbcc79-Abstract.html.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent Tool Use From Multi-Agent Autocurricula, February 2020. URL http://arxiv.org/abs/1909.07528. arXiv:1909.07528 [cs, stat].

Yogesh Balaji, Mehrdad Farajtabar, Dong Yin, Alex Mott, and Ang Li. The Effectiveness of Memory Replay in Large Scale Continual Learning, October 2020. URL http://arxiv.org/abs/2010.02418. arXiv:2010.02418 [cs].

Andrew G. Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pages 17–47. Springer Science + Business Media, New York, NY, US, 2013. ISBN 978-3-642-32374-4 978-3-642-32375-1. doi: 10.1007/978-3-642-32375-1_2.

Sarah Bechtle, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic

Righetti, Gaurav Sukhatme, and Franziska Meier. Meta-Learning via Learned Loss, January 2021. URL http://arxiv.org/abs/1906.05374. arXiv:1906.05374 [cs].

David Bell, Alison Duffy, and Adrienne Fairhall. Discovering plasticity rules that organize and maintain neural circuits, November 2024. URL https://www.biorxiv.org/content/10.1101/2024.11.18.623688v1. Pages: 2024.11.18.623688 Section: New Results.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013. ISSN 1076-9757. doi: 10.1613/jair.3912. URL https://jair.org/index.php/jair/article/view/10819.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying Count-Based Exploration and Intrinsic Motivation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

Marc G Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023. URL http://www.distributional-rl.org.

Eseoghene Ben-Iwhiwhu, Jeffery Dick, Nicholas A. Ketz, Praveen K. Pilly, and Andrea Soltoggio. Context meta-reinforcement learning via neuromodulation. *Neural Networks*, 152:70–79, August 2022. ISSN 0893-6080. doi: 10.1016/j.neunet.2022.04.003. URL https://www.sciencedirect.com/science/article/pii/S0893608022001368.

Dimitri P Bertsekas. *Dynamic programming and optimal control*. 2012. ISBN 978-1-886529-08-3 978-1-886529-43-4 978-1-886529-44-1. OCLC: 863688177.

Michele Biasutti. Flow and Optimal Experience. In *Reference Module in Neuroscience and Biobehavioral Psychology*. Elsevier, January 2017. ISBN 978-0-12-809324-5. doi: 10.

1016/B978-0-12-809324-5.06191-5. URL https://www.sciencedirect.com/science/article/pii/B9780128093245061915.

Peter J. Bickel and David A. Freedman. Some Asymptotic Theory for the Bootstrap. *The Annals of Statistics*, 9(6):1196–1217, November 1981. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176345637. URL https://projecteuclid.org/journals/annals-of-statistics/volume-9/issue-6/Some-Asymptotic-Theory-for-the-Bootstrap/10.1214/aos/1176345637.full. Publisher: Institute of Mathematical Statistics.

Marcel Binz, Ishita Dasgupta, Akshay Jagadish, Matthew Botvinick, Jane X. Wang, and Eric Schulz. Meta-Learned Models of Cognition, April 2023. URL http://arxiv.org/abs/2304.06729. arXiv:2304.06729 [cs].

Rafal Bogacz, Eric Brown, Jeff Moehlis, Philip Holmes, and Jonathan D. Cohen. The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, 113(4):700–765, 2006. ISSN 1939-1471. doi: 10.1037/0033-295X.113.4.700. Place: US Publisher: American Psychological Association.

Annika Boldt, Charles Blundell, and Benedetto De Martino. Confidence modulates exploration and exploitation in value-based learning. *Neuroscience of Consciousness*, 2019(1):niz004, January 2019. ISSN 2057-2107. doi: 10.1093/nc/niz004. URL https://doi.org/10.1093/nc/niz004.

Giulio Bondanelli and Srdjan Ostojic. Coding with transient trajectories in recurrent neural networks. *PLOS Computational Biology*, 16(2):e1007655, February 2020. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007655. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007655. Publisher: Public Library of Science.

Blake Bordelon and Cengiz Pehlevan. Self-Consistent Dynamical Field Theory of Kernel Evolution in Wide Neural Networks, October 2022. URL http://arxiv.org/abs/2205.09653. arXiv:2205.09653 [cond-mat, stat].

Blake Bordelon, Paul Masset, Henry Kuo, and Cengiz Pehlevan. Dynamics of Temporal Difference Reinforcement Learning, July 2023. URL http://arxiv.org/abs/2307.04841. arXiv:2307.04841 [cond-mat, stat].

M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen. Conflict monitoring and cognitive control. *Psychological Review*, 108(3):624–652, July 2001. ISSN 0033-295X. doi: 10.1037/0033-295x.108.3.624.

Matthew Botvinick, Jane X. Wang, Will Dabney, Kevin J. Miller, and Zeb Kurth-Nelson. Deep Reinforcement Learning and Its Neuroscientific Implications. *Neuron*, 107(4): 603–616, August 2020. ISSN 0896-6273. doi: 10.1016/j.neuron.2020.06.014. URL https://www.sciencedirect.com/science/article/pii/S0896627320304682.

Matthew M. Botvinick and Jonathan D. Cohen. The Computational and Neural Basis of Cognitive Control: Charted Territory and New Frontiers. *Cognitive Science*, 38(6):1249–1285, 2014. ISSN 1551-6709. doi: 10.1111/cogs.12126. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12126. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12126.

Lukas Braun, Clémentine Carla Juliette Dominé, James E. Fitzgerald, and Andrew M. Saxe. Exact learning dynamics of deep linear networks with prior knowledge. October 2022. URL https://openreview.net/forum?id=lJx2vng-KiC.

Vincent Breton-Provencher, Gabrielle T. Drummond, and Mriganka Sur. Locus Coeruleus Norepinephrine in Learned Behavior: Anatomical Modularity and Spatiotemporal Integration in Targets. *Frontiers in Neural Circuits*, 15, June 2021. ISSN 1662-5110. doi: 10.3389/fncir.2021.638007. URL https://www.frontiersin.org/journals/

neural-circuits/articles/10.3389/fncir.2021.638007/full. Publisher: Frontiers.

Ethan S. Bromberg-Martin, Yang-Yang Feng, Takaya Ogasawara, J. Kael White, Kaining Zhang, and Ilya E. Monosov. A neural mechanism for conserved value computations integrating information and rewards. *Nature Neuroscience*, 27(1):159–175, January 2024. ISSN 1546-1726. doi: 10.1038/s41593-023-01511-4. URL https://www.nature.com/articles/s41593-023-01511-4. Publisher: Nature Publishing Group.

Jerome Bruner. A Short History of Psychological Theories of Learning. *Daedalus*, 133 (1):13–20, 2004. ISSN 0011-5266. URL https://www.jstor.org/stable/20027892. Publisher: The MIT Press.

Emma Brunskill. Bayes-optimal reinforcement learning for discrete uncertainty domains. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, volume 3 of *AAMAS '12*, pages 1385–1386, Richland, SC, June 2012. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-3-0.

Yoram Burak and Ila R. Fiete. Accurate Path Integration in Continuous Attractor Network Models of Grid Cells. *PLOS Computational Biology*, 5(2):e1000291, February 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000291. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000291. Publisher: Public Library of Science.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by Random Network Distillation, October 2018. URL http://arxiv.org/abs/1810.12894. arXiv:1810.12894 [cs, stat].

Laura A. Bustamante, Temitope Oshinowo, Jeremy R. Lee, Elizabeth Tong, Allison R. Burton, Amitai Shenhav, Jonathan D. Cohen, and Nathaniel D. Daw. Effort Foraging

Task reveals positive correlation between individual differences in the cost of cognitive and physical effort in humans. *Proceedings of the National Academy of Sciences*, 120 (50):e2221510120, December 2023. doi: 10.1073/pnas.2221510120. URL https://www.pnas.org/doi/10.1073/pnas.2221510120. Publisher: Proceedings of the National Academy of Sciences.

Paul Bélanger. Three Main Learning Theories. In *Theories in Adult Learning and Education*, pages 17–34. Verlag Barbara Budrich, 1 edition, 2011. ISBN 978-3-86649-362-9. doi: 10.2307/j.ctvbkjx77.6. URL https://www.jstor.org/stable/j.ctvbkjx77.6.

John T. Cacioppo and Richard E. Petty. The need for cognition. *Journal of Personality and Social Psychology*, 42(1):116–131, 1982. ISSN 1939-1315. doi: 10.1037/0022-3514.42.1.116. Place: US Publisher: American Psychological Association.

Jessica F. Cantlon and Steven T. Piantadosi. Uniquely human intelligence arose from expanded information capacity. *Nature Reviews Psychology*, 3(4):275–293, April 2024. ISSN 2731-0574. doi: 10.1038/s44159-024-00283-3. URL https://www.nature.com/articles/s44159-024-00283-3. Publisher: Nature Publishing Group.

R. Carrasco-Davis, E. Reyes, C. Valenzuela, F. Förster, P. A. Estévez, G. Pignata, F. E. Bauer, I. Reyes, P. Sánchez-Sáez, G. Cabrera-Vives, S. Eyheramendy, M. Catelan, J. Arredondo, E. Castillo-Navarrete, D. Rodríguez-Mancini, D. Ruz-Mieres, A. Moya, L. Sabatini-Gacitúa, C. Sepúlveda-Cobo, A. A. Mahabal, J. Silva-Farfán, E. Camacho-Iñiguez, and L. Galbany. Alert Classification for the ALeRCE Broker System: The Real-time Stamp Classifier. *The Astronomical Journal*, 162(6):231, November 2021. ISSN 1538-3881. doi: 10.3847/1538-3881/ac0ef1. URL https://dx.doi.org/10.3847/1538-3881/ac0ef1. Publisher: The American Astronomical Society.

Rodrigo Antonio Carrasco-Davis and Erin Grant. A primer on analytical learning dynamics of nonlinear neural networks. January 2025. URL https://openreview.

net/forum?id=fJCgF02C87&referrer=%5BAuthor%20Console%5D(%2Fgroup%3Fid%3DICLR.cc%2F2025%2FBlogPosts%2FAuthors%23your-submissions).

John B. Carroll. Psychometrics, intelligence, and public perception. *Intelligence*, 24(1): 25–52, January 1997. ISSN 0160-2896. doi: 10.1016/S0160-2896(97)90012-X. URL https://www.sciencedirect.com/science/article/pii/S016028969790012X.

Tommaso Castellani and Andrea Cavagna. Spin-glass theory for pedestrians. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(05):P05012, May 2005. ISSN 1742-5468. doi: 10.1088/1742-5468/2005/05/P05012. URL https://dx.doi.org/10.1088/1742-5468/2005/05/P05012.

Sanjay Joseph Chacko, Neeraj P.c., and Rajesh Joseph Abraham. Optimizing LQR controllers: A comparative study. *Results in Control and Optimization*, 14:100387, March 2024. ISSN 2666-7207. doi: 10.1016/j.rico.2024.100387. URL https://www.sciencedirect.com/science/article/pii/S2666720724000171.

Lynne Chantranupong, Celia C. Beron, Joshua A. Zimmer, Michelle J. Wen, Wengang Wang, and Bernardo L. Sabatini. Dopamine and glutamate regulate striatal acetylcholine in decision-making. *Nature*, 621(7979):577–585, September 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06492-9. URL https://www.nature.com/articles/s41586-023-06492-9. Publisher: Nature Publishing Group.

Bertrand Charpentier, Ransalu Senanayake, Mykel Kochenderfer, and Stephan Günnemann. Disentangling Epistemic and Aleatoric Uncertainty in Reinforcement Learning, June 2022. URL http://arxiv.org/abs/2206.01558. arXiv:2206.01558 [cs].

Shuangshuang Chen and Wei Guo. Auto-Encoders in Deep Learning—A Review with New Perspectives. *Mathematics*, 11(8):1777, January 2023. ISSN 2227-7390. doi: 10.3390/math11081777. URL https://www.mdpi.com/2227-7390/11/8/1777. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On Lazy Training in Differentiable Programming, January 2020. URL http://arxiv.org/abs/1812.07956. arXiv:1812.07956 [cs, math].

François Chollet. On the Measure of Intelligence, November 2019. URL http://arxiv.org/abs/1911.01547. arXiv:1911.01547 [cs].

Noam Chomsky. 4. A Review of B. F. Skinner's Verbal Behavior. In Ned Block, editor, *Volume I Readings in Philosophy of Psychology, Volume I*, pages 48–64. Harvard University Press, October 2013. ISBN 978-0-674-59462-3. doi: 10.4159/harvard.9780674594623.c6. URL https://www.degruyter.com/document/doi/10.4159/harvard.9780674594623.c6/html?lang=en.

Noam Chomsky. *Syntactic Structures*. De Gruyter Mouton, May 2020. ISBN 978-3-11-231600-9. doi: 10.1515/9783112316009. URL https://www.degruyter.com/document/doi/10.1515/9783112316009/html?lang=en.

Lasse Christiansen, Malte Nejst Larsen, Mads Just Madsen, Michael James Grey, Jens Bo Nielsen, and Jesper Lundbye-Jensen. Long-term motor skill training with individually adjusted progressive difficulty enhances learning and promotes corticospinal plasticity. *Scientific Reports*, 10(1):15588, September 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-72139-8. URL https://www.nature.com/articles/s41598-020-72139-8. Publisher: Nature Publishing Group.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models, November 2018. URL http://arxiv.org/abs/1805.12114. arXiv:1805.12114 [cs, stat].

Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Stephen I. Ryu, and Krishna V. Shenoy. Cortical Preparatory Activity: Representation of Movement or First Cog in a Dynamical Machine? *Neuron*, 68(3):387–400, November 2010. ISSN

0896-6273. doi: 10.1016/j.neuron.2010.09.015. URL https://www.cell.com/neuron/abstract/S0896-6273(10)00757-9. Publisher: Elsevier.

Ami Citri and Robert C. Malenka. Synaptic Plasticity: Multiple Forms, Functions, and Mechanisms. *Neuropsychopharmacology*, 33(1):18–41, January 2008. ISSN 1740-634X. doi: 10.1038/sj.npp.1301559. URL https://www.nature.com/articles/1301559. Publisher: Nature Publishing Group.

William R. Clements, Bastien Van Delft, Benoît-Marie Robaglia, Reda Bahi Slaoui, and Sébastien Toth. Estimating Risk and Uncertainty in Deep Reinforcement Learning, September 2020. URL http://arxiv.org/abs/1905.09638. arXiv:1905.09638 [cs, stat].

Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matt Botvinick, Jane Wang, and Eric Schulz. Meta-in-context learning in large language models. *Advances in Neural Information Processing Systems*, 36:65189–65201, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/cda04d7ea67ea1376bf8c6962d8541e0-Abstract-Conference.html.

Jonathan D. Cohen. Cognitive control: Core constructs and current considerations. In *The Wiley handbook of cognitive control*, pages 3–28. Wiley Blackwell, Hoboken, NJ, US, 2017. ISBN 978-1-118-92054-1 978-1-118-92048-0 978-1-118-92047-3. doi: 10.1002/9781118920497.ch1.

Jonathan D. Cohen, Kevin Dunbar, and James L. McClelland. On the control of automatic processes: A parallel distributed processing account of the Stroop effect. *Psychological Review*, 97(3):332–361, 1990. ISSN 1939-1471. doi: 10.1037/0033-295X.97.3.332. Place: US Publisher: American Psychological Association.

Jonathan D. Cohen, Gary Aston-Jones, and Mark S. Gilzenrat. A Systems-Level Perspective on Attention and Cognitive Control: Guided Activation, Adaptive Gating, Conflict

Monitoring, and Exploitation versus Exploration. In *Cognitive neuroscience of attention*, pages 71–90. The Guilford Press, New York, NY, US, 2004. ISBN 978-1-59385-048-7.

David Cohn, Zoubin Ghahramani, and Michael Jordan. Active Learning with Statistical Models. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL https://papers.nips.cc/paper_files/paper/1994/hash/7f975a56c761db6506eca0b37ce6ec87-Abstract.html.

Basile Confavreux, Everton J. Agnes, Friedemann Zenke, Henning Sprekeler, and Tim P. Vogels. Balancing complexity, performance and plausibility to meta learn plasticity rules in recurrent spiking networks, June 2024. URL https://www.biorxiv.org/content/10.1101/2024.06.17.599260v1. Pages: 2024.06.17.599260 Section: New Results.

Michael Crawshaw. Multi-Task Learning with Deep Neural Networks: A Survey, September 2020. URL http://arxiv.org/abs/2009.09796. arXiv:2009.09796 [cs, stat].

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional Reinforcement Learning with Quantile Regression, October 2017. URL http://arxiv.org/abs/1710.10044. arXiv:1710.10044 [cs, stat].

Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577(7792):671–675, January 2020. ISSN 1476-4687. doi: 10.1038/s41586-019-1924-6. URL https://www.nature.com/articles/s41586-019-1924-6. Publisher: Nature Publishing Group.

Nathaniel D. Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, 8(12):1704–1711, December 2005. ISSN 1546-1726. doi: 10.1038/nn1560. URL https://www.nature.com/articles/nn1560. Number: 12 Publisher: Nature Publishing Group.

Benedetto De Martino, Stephen M. Fleming, Neil Garrett, and Raymond J. Dolan. Confidence in value-based choice. *Nature Neuroscience*, 16(1):105–110, January 2013. ISSN 1546-1726. doi: 10.1038/nn.3279. URL https://www.nature.com/articles/nn.3279. Publisher: Nature Publishing Group.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602 (7897):414–419, February 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04301-9. URL https://www.nature.com/articles/s41586-021-04301-9. Number: 7897 Publisher: Nature Publishing Group.

Tristan Deleu, David Kanaa, Leo Feng, Giancarlo Kerg, Yoshua Bengio, Guillaume Lajoie, and Pierre-Luc Bacon. Continuous-Time Meta-Learning with Forward Mode Differentiation, March 2022. URL http://arxiv.org/abs/2203.01443. arXiv:2203.01443 [cs].

Li Deng. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6):141–142, November 2012. ISSN 1558-0792. doi: 10.1109/MSP.2012.2211477. Conference Name: IEEE Signal Processing Magazine.

Kenji Doya. Metalearning and neuromodulation. *Neural Networks*, 15(4):495–506, June 2002. ISSN 0893-6080. doi: 10.1016/S0893-6080(02)00044-8. URL https://www.sciencedirect.com/science/article/pii/S0893608002000448.

Gesine Dreisbach, Sebastian Musslick, and Senne Braem. Flexibility and stability can be

both dependent and independent. *Nature Reviews Psychology*, 3(9):636–636, September 2024. ISSN 2731-0574. doi: 10.1038/s44159-024-00348-3. URL https://www.nature.com/articles/s44159-024-00348-3. Publisher: Nature Publishing Group.

Vikranth Dwaracherla, Zheng Wen, Ian Osband, Xiuyuan Lu, Seyed Mohammad Asghari, and Benjamin Van Roy. Ensembles for Uncertainty Estimation: Benefits of Prior Functions and Bootstrapping, June 2022. URL http://arxiv.org/abs/2206.03633. arXiv:2206.03633 [cs, stat].

Hermann Ebbinghaus (1885). Memory: A Contribution to Experimental Psychology. *Annals of Neurosciences*, 20(4):155–156, October 2013. ISSN 0972-7531. doi: 10.5214/ans.0972.7531.200408. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4117135/.

Luke G. Eglington and Philip I. Pavlik Jr. Optimizing practice scheduling requires quantitative tracking of individual item performance. *npj Science of Learning*, 5 (1):1–10, October 2020. ISSN 2056-7936. doi: 10.1038/s41539-020-00074-4. URL https://www.nature.com/articles/s41539-020-00074-4. Number: 1 Publisher: Nature Publishing Group.

Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in Reinforcement Learning and How To Tune Them, June 2023. URL http://arxiv.org/abs/2306.01324. arXiv:2306.01324 [cs].

Eran Eldar, Jonathan D. Cohen, and Yael Niv. The effects of neural gain on attention and learning. *Nature Neuroscience*, 16(8):1146–1153, August 2013. ISSN 1546-1726. doi: 10.1038/nn.3428. URL https://www.nature.com/articles/nn.3428. Number: 8 Publisher: Nature Publishing Group.

Chris Eliasmith and Oliver Trujillo. The use and abuse of large-scale brain models. *Current Opinion in Neurobiology*, 25:1–6, April 2014. ISSN 0959-4388. doi: 10.1016/

j.conb.2013.09.009. URL https://www.sciencedirect.com/science/article/pii/S095943881300189X.

Omer Elkabetz and Nadav Cohen. Continuous vs. Discrete Optimization of Deep Neural Networks, December 2021. URL http://arxiv.org/abs/2107.06608. arXiv:2107.06608 [cs].

K. Anders Ericsson and Kyle W. Harwell. Deliberate Practice and Proposed Limits on the Effects of Practice on the Acquisition of Expert Performance: Why the Original Definition Matters and Recommendations for Future Research. *Frontiers in Psychology*, 10, 2019. ISSN 1664-1078. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2019.02396.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the Convergence Theory of Gradient-Based Model-Agnostic Meta-Learning Algorithms. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 1082–1092. PMLR, June 2020. URL https://proceedings.mlr.press/v108/fallah20a.html. ISSN: 2640-3498.

Haoxue Fan, Taylor Burke, Deshawn Chatman Sambrano, Emily Dial, Elizabeth A. Phelps, and Samuel J. Gershman. Pupil Size Encodes Uncertainty during Exploration. *Journal of Cognitive Neuroscience*, 35(9):1508–1520, September 2023. ISSN 0898-929X. doi: 10.1162/jocn_a_02025. URL https://ieeexplore.ieee.org/abstract/document/10301964. Conference Name: Journal of Cognitive Neuroscience.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting Fundamentals of Experience Replay. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3061–3071. PMLR, November 2020. URL https://proceedings.mlr.press/v119/fedus20a.html. ISSN: 2640-3498.

Heidi M. Feldman. How Young Children Learn Language and Speech. *Pediatrics In Review*, 40(8):398–411, August 2019. ISSN 0191-9601. doi: 10.1542/pir.2017-0325. URL https://doi.org/10.1542/pir.2017-0325.

Katie A. Ferguson and Jessica A. Cardin. Mechanisms underlying gain modulation in the cortex. *Nature Reviews Neuroscience*, 21(2):80–92, February 2020. ISSN 1471-0048. doi: 10.1038/s41583-019-0253-y. URL https://www.nature.com/articles/s41583-019-0253-y. Number: 2 Publisher: Nature Publishing Group.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, July 2017. URL http://arxiv.org/abs/1703.03400. arXiv:1703.03400 [cs].

Stephen M. Fleming. Metacognition and Confidence: A Review and Synthesis. *Annual Review of Psychology*, 75(Volume 75, 2024):241–268, January 2024. ISSN 0066-4308, 1545-2085. doi: 10.1146/annurev-psych-022423-032425. URL https://www.annualreviews.org/content/journals/10.1146/annurev-psych-022423-032425. Publisher: Annual Reviews.

Timo Flesch, Keno Juechems, Tsvetomira Dumbalska, Andrew Saxe, and Christopher Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110(7):1258–1270.e11, April 2022. ISSN 0896-6273. doi: 10.1016/j.neuron.2022.01.005. URL https://www.sciencedirect.com/science/article/pii/S0896627322000058.

Damien S. Fleur, Bert Bredeweg, and Wouter van den Bos. Metacognition: ideas and insights from neuro- and educational sciences. *npj Science of Learning*, 6(1): 13, June 2021. ISSN 2056-7936. doi: 10.1038/s41539-021-00089-5. URL https://www.nature.com/articles/s41539-021-00089-5. Publisher: Nature Publishing Group.

Viola Folli, Marco Leonetti, and Giancarlo Ruocco. On the Maximum Storage Capacity of the Hopfield Model. *Frontiers in Computational Neuroscience*, 10, January 2017. ISSN 1662-5188. doi: 10.3389/fncom.2016.00144. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2016.00144/full. Publisher: Frontiers.

Cyrus K. Foroughi, Ciara Sibley, and Joseph T. Coyne. Pupil size as a measure of within-task learning. *Psychophysiology*, 54(10):1436–1443, 2017. ISSN 1469-8986. doi: 10.1111/psyp.12896. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.12896. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/psyp.12896.

Birte U. Forstmann, Gilles Dutilh, Scott Brown, Jane Neumann, D. Yves von Cramon, K. Richard Ridderinkhof, and Eric-Jan Wagenmakers. Striatum and pre-SMA facilitate decision-making under time pressure. *Proceedings of the National Academy of Sciences*, 105(45):17538–17542, November 2008. doi: 10.1073/pnas.0805903105. URL https://www.pnas.org/doi/full/10.1073/pnas.0805903105. Publisher: Proceedings of the National Academy of Sciences.

David J. Foster and Matthew A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, March 2006. ISSN 1476-4687. doi: 10.1038/nature04587. URL https://www.nature.com/articles/nature04587. Number: 7084 Publisher: Nature Publishing Group.

Kevin Fox and Michael Stryker. Integrating Hebbian and homeostatic plasticity: introduction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1715):20160413, March 2017. doi: 10.1098/rstb.2016.0413. URL https://royalsocietypublishing.org/doi/10.1098/rstb.2016.0413. Publisher: Royal Society.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward

and Reverse Gradient-Based Hyperparameter Optimization, December 2017. URL http://arxiv.org/abs/1703.01785. arXiv:1703.01785 [stat].

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel Programming for Hyperparameter Optimization and Meta-Learning, July 2018. URL http://arxiv.org/abs/1806.04910. arXiv:1806.04910 [cs, stat].

Shane Frederick, George Loewenstein, and Ted O'Donoghue. Time Discounting and Time Preference: A Critical Review. *Journal of Economic Literature*, 40(2):351–401, June 2002. ISSN 0022-0515. doi: 10.1257/002205102320161311. URL https://www.aeaweb.org/articles?id=10.1257/002205102320161311.

R. Frömer, H. Lin, C. K. Dean Wolf, M. Inzlicht, and A. Shenhav. Expectations of reward and efficacy guide cognitive control allocation. *Nature Communications*, 12(1): 1030, February 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-21315-z. URL https://www.nature.com/articles/s41467-021-21315-z. Publisher: Nature Publishing Group.

Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119–130, January 1988. ISSN 0893-6080. doi: 10.1016/0893-6080(88)90014-7. URL https://www.sciencedirect.com/science/article/pii/0893608088900147.

F. Förster, G. Cabrera-Vives, E. Castillo-Navarrete, P. A. Estévez, P. Sánchez-Sáez, J. Arredondo, F. E. Bauer, R. Carrasco-Davis, M. Catelan, F. Elorrieta, S. Eyheramendy, P. Huijse, G. Pignata, E. Reyes, I. Reyes, D. Rodríguez-Mancini, D. Ruz-Mieres, C. Valenzuela, I. Álvarez Maldonado, N. Astorga, J. Borissova, A. Clocchiatti, D. De Cicco, C. Donoso-Oliva, L. Hernández-García, M. J. Graham, A. Jordán, R. Kurtev, A. Mahabal, J. C. Maureira, A. Muñoz-Arancibia, R. Molina-Ferreiro, A. Moya, W. Palma, M. Pérez-Carrasco, P. Protopapas, M. Romero, L. Sabatini-Gacitua, A. Sánchez, J. San Martín, C. Sepúlveda-Cobo, E. Vera, and J. R. Vergara. The Automatic Learning

for the Rapid Classification of Events (ALeRCE) Alert Broker. *The Astronomical Journal*, 161(5):242, April 2021. ISSN 1538-3881. doi: 10.3847/1538-3881/abe9bc. URL https://dx.doi.org/10.3847/1538-3881/abe9bc. Publisher: The American Astronomical Society.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR, June 2016. URL https://proceedings.mlr.press/v48/gal16.html. ISSN: 1938-7228.

Soheil Ganjefar and Sara Rezaei. Modified homotopy perturbation method for optimal control problems using the Padé approximant. *Applied Mathematical Modelling*, 40 (15):7062–7081, August 2016. ISSN 0307-904X. doi: 10.1016/j.apm.2016.02.039. URL https://www.sciencedirect.com/science/article/pii/S0307904X16301147.

Samuel Liebana Garcia, Aeron Laffere, Chiara Toschi, Louisa Schilling, Jacek Podlaski, Matthias Fritsche, Peter Zatka-Haas, Yulong Li, Rafal Bogacz, Andrew Saxe, and Armin Lak. Striatal dopamine reflects individual long-term learning trajectories, December 2023. URL https://www.biorxiv.org/content/10.1101/2023.12.14.571653v1. Pages: 2023.12.14.571653 Section: New Results.

Howard Gardner. *Frames Of Mind*. Basic Books, 1983. ISBN 978-0-465-02508-4. Google-Books-ID: _hWdAAAAMAAJ.

Tim Genewein, Grégoire Delétang, Anian Ruoss, Li Kevin Wenliang, Elliot Catt, Vincent Dutordoir, Jordi Grau-Moya, Laurent Orseau, Marcus Hutter, and Joel Veness. Memory-Based Meta-Learning on Non-Stationary Distributions, May 2023. URL http://arxiv.org/abs/2302.03067. arXiv:2302.03067 [cs, stat].

Samuel J. Gershman. Dopamine, Inference, and Uncertainty. *Neural Computation*, 29

(12):3311–3326, December 2017. ISSN 0899-7667. doi: 10.1162/neco_a_01023. URL https://doi.org/10.1162/neco_a_01023.

Samuel J. Gershman. Just looking: The innocent eye in neuroscience. *Neuron*, 109(14): 2220–2223, July 2021. ISSN 0896-6273. doi: 10.1016/j.neuron.2021.05.022. URL https://www.cell.com/neuron/abstract/S0896-6273(21)00375-5. Publisher: Elsevier.

Samuel J. Gershman and Taylor Burke. Mental control of uncertainty. *Cognitive, Affective, & Behavioral Neuroscience*, 23(3):465–475, June 2023. ISSN 1531-135X. doi: 10.3758/s13415-022-01034-8. URL https://doi.org/10.3758/s13415-022-01034-8.

Samuel J. Gershman, John A. Assad, Sandeep Robert Datta, Scott W. Linderman, Bernardo L. Sabatini, Naoshige Uchida, and Linda Wilbrecht. Explaining dopamine through prediction errors and beyond. *Nature Neuroscience*, 27(9):1645–1655, September 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01705-4. URL https://www.nature.com/articles/s41593-024-01705-4. Publisher: Nature Publishing Group.

Tyler Giallanza, Declan Campbell, and Jonathan D. Cohen. Toward the Emergence of Intelligent Control: Episodic Generalization and Optimization. *Open Mind*, 8: 688–722, May 2024. ISSN 2470-2986. doi: 10.1162/opmi_a_00143. URL https://doi.org/10.1162/opmi_a_00143.

Gilles E. Gignac and Eva T. Szodorai. Defining intelligence: Bridging the gap between human and artificial perspectives. *Intelligence*, 104:101832, May 2024. ISSN 0160-2896. doi: 10.1016/j.intell.2024.101832. URL https://www.sciencedirect.com/science/article/pii/S0160289624000266.

Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems*, volume 32. Cur-

ran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/cab070d53bd0d200746fb852a922064a-Abstract.html.

Sebastian Goldt, Madhu S. Advani, Andrew M. Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher–student setup*. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124010, December 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc61e. URL https://dx.doi.org/10.1088/1742-5468/abc61e. Publisher: IOP Publishing and SISSA.

Eugene Golikov, Eduard Pokonechnyy, and Vladimir Korviakov. Neural Tangent Kernel: A Survey, August 2022. URL http://arxiv.org/abs/2208.13614. arXiv:2208.13614 [cs].

Jacqueline Gottlieb. Attention, Learning, and the Value of Information. *Neuron*, 76(2):281–295, October 2012. ISSN 0896-6273. doi: 10.1016/j.neuron.2012.09.034. URL https://www.cell.com/neuron/abstract/S0896-6273(12)00888-4. Publisher: Elsevier.

Gabriele Gratton, Michael G. H. Coles, and Emanuel Donchin. Optimizing the use of information: Strategic control of activation of responses. *Journal of Experimental Psychology: General*, 121(4):480–506, 1992. ISSN 1939-2222. doi: 10.1037/0096-3445.121.4.480. Place: US Publisher: American Psychological Association.

Alex Graves. Generating Sequences With Recurrent Neural Networks, June 2014. URL http://arxiv.org/abs/1308.0850. arXiv:1308.0850 [cs].

William H. Greene. *Econometric Analysis*. Prentice Hall, 2000. ISBN 978-0-13-013297-0. Google-Books-ID: YiC7AAAAIAAJ.

Aikaterini I. Griva, Achilles D. Boursianis, Lazaros A. Iliadis, Panagiotis Sarigiannidis, George Karagiannidis, and Sotirios K. Goudos. Model-Agnostic Meta-Learning Techniques: A State-of-The-Art Short Review. In *2023 12th International Confer-*

ence on Modern Circuits and Systems Technologies (MOCAST), pages 1–4, June 2023. doi: 10.1109/MOCAST57943.2023.10176893. URL https://ieeexplore.ieee.org/document/10176893.

Cooper D. Grossman, Bilal A. Bari, and Jeremiah Y. Cohen. Serotonin neurons modulate learning rate through uncertainty. *Current Biology*, 32(3):586–599.e7, February 2022. ISSN 0960-9822. doi: 10.1016/j.cub.2021.12.006. URL https://www.sciencedirect.com/science/article/pii/S0960982221016821.

Benedikt Grothe. The evolution of temporal processing in the medial superior olive, an auditory brainstem structure. *Progress in Neurobiology*, 61(6):581–610, August 2000. ISSN 0301-0082. doi: 10.1016/S0301-0082(99)00068-4. URL https://www.sciencedirect.com/science/article/pii/S0301008299000684.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-Reinforcement Learning of Structured Exploration Strategies, February 2018. URL http://arxiv.org/abs/1802.07245. arXiv:1802.07245 [cs].

Martin S. Hagger, Nikos L. D. Chatzisarantis, Hugo Alberts, Calvin Octavianus Anggono, Cédric Batailler, Angela R. Birt, Ralf Brand, Mark J. Brandt, Gene Brewer, Sabrina Bruyneel, Dustin P. Calvillo, W. Keith Campbell, Peter R. Cannon, Marianna Carlucci, Nicholas P. Carruth, Tracy Cheung, Adrienne Crowell, Denise T. D. De Ridder, Siegfried Dewitte, Malte Elson, Jacqueline R. Evans, Benjamin A. Fay, Bob M. Fennis, Anna Finley, Zoë Francis, Elke Heise, Henrik Hoemann, Michael Inzlicht, Sander L. Koole, Lina Koppel, Floor Kroese, Florian Lange, Kevin Lau, Bridget P. Lynch, Carolien Martijn, Harald Merckelbach, Nicole V. Mills, Alexej Michirev, Akira Miyake, Alexandra E. Mosser, Megan Muise, Dominique Muller, Milena Muzi, Dario Nalis, Ratri Nurwanti, Henry Otgaar, Michael C. Philipp, Pierpaolo Primoceri, Katrin Rentzsch, Lara Ringos, Caroline Schlinkert, Brandon J. Schmeichel, Sarah F. Schoch, Michel Schrama, Astrid Schütz, Angelos Stamos, Gustav Tinghög, Johannes Ullrich, Michelle vanDellen, Supra

Wimbarti, Wanja Wolff, Cleoputri Yusainy, Oulmann Zerhouni, and Maria Zwienenberg. A Multilab Preregistered Replication of the Ego-Depletion Effect. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science*, 11(4): 546–573, July 2016. ISSN 1745-6924. doi: 10.1177/1745691616652873.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-Inspired Artificial Intelligence. *Neuron*, 95(2):245–258, July 2017. ISSN 0896-6273. doi: 10.1016/j.neuron.2017.06.011. URL https://www.cell.com/neuron/abstract/S0896-6273(17)30509-3. Publisher: Elsevier.

Ryoma Hattori, Nathan G. Hedrick, Anant Jain, Shuqi Chen, Hanjia You, Mariko Hattori, Jun-Hyeok Choi, Byung Kook Lim, Ryohei Yasuda, and Takaki Komiyama. Meta-reinforcement learning via orbitofrontal cortex. *Nature Neuroscience*, 26(12):2182–2191, December 2023. ISSN 1546-1726. doi: 10.1038/s41593-023-01485-3. URL https://www.nature.com/articles/s41593-023-01485-3. Publisher: Nature Publishing Group.

Ji-Huan He. Homotopy perturbation technique. *Computer Methods in Applied Mechanics and Engineering*, 178(3):257–262, August 1999. ISSN 0045-7825. doi: 10.1016/S0045-7825(99)00018-3. URL https://www.sciencedirect.com/science/article/pii/S0045782599000183.

Richard P. Heitz. The speed-accuracy tradeoff: history, physiology, methodology, and behavior. *Frontiers in Neuroscience*, 8, June 2014. ISSN 1662-453X. doi: 10.3389/fnins.2014.00150. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2014.00150/full. Publisher: Frontiers.

Christian Henning, Maria Cervera, Francesco D' Angelo, Johannes von Oswald, Regina Traber, Benjamin Ehret, Seijin Kobayashi, Benjamin F. Grewe, and João Sacramento. Posterior Meta-Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 14135–14149. Curran As-

sociates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/761b42cfff120aac30045f7a110d0256-Abstract.html.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining Improvements in Deep Reinforcement Learning, October 2017. URL http://arxiv.org/abs/1710.02298. arXiv:1710.02298 [cs].

E. F. Hiby, N. J. Rooney, and J. W. S. Bradshaw. Dog training methods: their use, effectiveness and interaction with behaviour and welfare. *Animal Welfare*, 13(1): 63–69, February 2004. ISSN 0962-7286, 2054-1538. doi: 10.1017/S0962728600026683. URL https://www.cambridge.org/core/journals/animal-welfare/article/abs/dog-training-methods-their-use-effectiveness-and-interaction-with-behaviour-and-wel B219F8FBE35C05DA269D02F310E38B66.

Adrian Hille and Jürgen Schupp. How learning a musical instrument affects the development of skills. *Economics of Education Review*, 44:56–82, February 2015. ISSN 0272-7757. doi: 10.1016/j.econedurev.2014.10.007. URL https://www.sciencedirect.com/science/article/pii/S0272775714000995.

Yvonne Hindes, Mike R. Schoenberg, and Donald H. Saklofske. Intelligence. In Jeffrey S. Kreutzer, John DeLuca, and Bruce Caplan, editors, *Encyclopedia of Clinical Neuropsychology*, pages 1329–1335. Springer, New York, NY, 2011. ISBN 978-0-387-79948-3. doi: 10.1007/978-0-387-79948-3_1061. URL https://doi.org/10.1007/978-0-387-79948-3_1061.

Halfdan Holm and Soumya Banerjee. Intelligence in animals, humans and machines: a heliocentric view of intelligence? *AI & SOCIETY*, April 2024. ISSN 1435-5655. doi: 10.1007/s00146-024-01931-1. URL https://doi.org/10.1007/s00146-024-01931-1.

Philip Holmes and Jonathan D. Cohen. Optimality and Some of Its Discontents:

Successes and Shortcomings of Existing Models for Binary Decisions. *Topics in Cognitive Science*, 6(2):258–278, 2014. ISSN 1756-8765. doi: 10.1111/tops.12084. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/tops.12084. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tops.12084.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-Learning in Neural Networks: A Survey, November 2020. URL http://arxiv.org/abs/2004.05439. arXiv:2004.05439 [cs].

D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591, 1959. ISSN 1469-7793. doi: 10.1113/jphysiol.1959.sp006308. URL https://onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1959.sp006308. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1959.sp006308.

C. L. Hull. *Principles of behavior: an introduction to behavior theory.* Principles of behavior: an introduction to behavior theory. Appleton-Century, Oxford, England, 1943. Pages: x, 422.

Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A. Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference, October 2019. URL http://arxiv.org/abs/1905.06424. arXiv:1905.06424 [cs].

Sébastien Hélie and Denis Cousineau. The cognitive neuroscience of automaticity: Behavioral and brain signatures. In *Advances in cognitive and behavioral sciences*, Psychology

research progress, pages 141–159. Nova Science Publishers, Hauppauge, NY, US, 2014. ISBN 978-1-62948-890-5 978-1-62948-893-6.

Sébastien Hélie, Jennifer G. Waldschmidt, and F. Gregory Ashby. Automaticity in rule-based and information-integration categorization. *Attention, Perception, & Psychophysics*, 72(4):1013–1031, 2010. ISSN 1943-393X. doi: 10.3758/APP.72.4.1013. Place: US Publisher: Psychonomic Society.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, March 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05946-3. URL https://doi.org/10.1007/s10994-021-05946-3.

Kiyohito Iigaya, Madalena S. Fonseca, Masayoshi Murakami, Zachary F. Mainen, and Peter Dayan. An effect of serotonergic stimulation on learning rates for rewards apparent after long intertrial intervals. *Nature Communications*, 9(1):2477, June 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-04840-2. URL https://www.nature.com/articles/s41467-018-04840-2. Publisher: Nature Publishing Group.

Michael Inzlicht, Amitai Shenhav, and Christopher Y. Olivola. The Effort Paradox: Effort Is Both Costly and Valued. *Trends in Cognitive Sciences*, 22(4):337–349, April 2018. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2018.01.007. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(18)30020-2. Publisher: Elsevier.

Huw Jarvis, Isabelle Stevenson, Amy Q. Huynh, Emily Babbage, James Coxon, and Trevor T.-J. Chong. Effort Reinforces Learning. *Journal of Neuroscience*, 42(40):7648–7658, October 2022. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2223-21.2022. URL https://www.jneurosci.org/content/42/40/7648. Publisher: Society for Neuroscience Section: Research Articles.

Kristopher T. Jensen, Guillaume Hennequin, and Marcelo G. Mattar. A recurrent network model of planning explains hippocampal replay and human behavior. *Nature Neuroscience*, 27(7):1340–1348, July 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01675-7. URL https://www.nature.com/articles/s41593-024-01675-7. Publisher: Nature Publishing Group.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Theoretical Convergence of Multi-Step Model-Agnostic Meta-Learning, July 2020. URL http://arxiv.org/abs/2002.07836. arXiv:2002.07836 [cs, math, stat] version: 2.

Yiding Jiang, J. Zico Kolter, and Roberta Raileanu. On the Importance of Exploration for Generalization in Reinforcement Learning, June 2023. URL http://arxiv.org/abs/2306.05483. arXiv:2306.05483 [cs].

Joshua P. Johansen, Lorenzo Diaz-Mataix, Hiroki Hamanaka, Takaaki Ozawa, Edgar Ycu, Jenny Koivumaa, Ashwani Kumar, Mian Hou, Karl Deisseroth, Edward S. Boyden, and Joseph E. LeDoux. Hebbian and neuromodulatory mechanisms interact to trigger associative memory formation. *Proceedings of the National Academy of Sciences*, 111 (51):E5584–E5592, December 2014. doi: 10.1073/pnas.1421304111. URL https://www.pnas.org/doi/10.1073/pnas.1421304111. Publisher: Proceedings of the National Academy of Sciences.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*,

596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2. Publisher: Nature Publishing Group.

Joseph W. Kable and Paul W. Glimcher. The neural correlates of subjective value during intertemporal choice. *Nature Neuroscience*, 10(12):1625–1633, December 2007. ISSN 1546-1726. doi: 10.1038/nn2007. URL https://www.nature.com/articles/nn2007. Publisher: Nature Publishing Group.

Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-Aware Reinforcement Learning for Collision Avoidance, February 2017. URL http://arxiv.org/abs/1702.01182. arXiv:1702.01182 [cs].

Anssi Kanervisto, Christian Scheller, Yanick Schraner, and Ville Hautamäki. Distilling Reinforcement Learning Tricks for Video Games. In *2021 IEEE Conference on Games (CoG)*, pages 01–04, August 2021. doi: 10.1109/CoG52621.2021.9618997. URL https://ieeexplore.ieee.org/document/9618997. ISSN: 2325-4289.

Christos Kaplanis, Claudia Clopath, and Murray Shanahan. Continual Reinforcement Learning with Multi-Timescale Replay, April 2020. URL http://arxiv.org/abs/2004.07530. arXiv:2004.07530 [cs, stat].

Kristof Keidel, Qëndresa Rramani, Bernd Weber, Carsten Murawski, and Ulrich Ettinger. Individual Differences in Intertemporal Choice. *Frontiers in Psychology*, 12, 2021. ISSN 1664-1078. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2021.643670.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards Continual Reinforcement Learning: A Review and Perspectives, November 2022. URL http://arxiv.org/abs/2012.13490. arXiv:2012.13490 [cs].

Celeste Kidd, Steven T. Piantadosi, and Richard N. Aslin. The Goldilocks Effect: Human

Infants Allocate Attention to Visual Sequences That Are Neither Too Simple Nor Too Complex. *PLOS ONE*, 7(5):e36399, May 2012. ISSN 1932-6203. doi: 10.1371/journal.pone.0036399. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0036399. Publisher: Public Library of Science.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/10.1073/pnas.1611835114. Publisher: Proceedings of the National Academy of Sciences.

Miriam C. Klein-Flügge, Steven W. Kennerley, Karl Friston, and Sven Bestmann. Neural Signatures of Value Comparison in Human Cingulate Cortex during Decisions Requiring an Effort-Reward Trade-off. *Journal of Neuroscience*, 36(39):10002–10015, September 2016. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0292-16.2016. URL https://www.jneurosci.org/content/36/39/10002. Publisher: Society for Neuroscience Section: Articles.

Roger Koenker and Kevin F. Hallock. Quantile Regression. *Journal of Economic Perspectives*, 15(4):143–156, December 2001. ISSN 0895-3309. doi: 10.1257/jep.15.4.143. URL https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143.

Wouter Kool and Matthew Botvinick. A labor/leisure tradeoff in cognitive control. *Journal of Experimental Psychology: General*, 143(1):131–141, 2014. ISSN 1939-2222. doi: 10.1037/a0031048. Place: US Publisher: American Psychological Association.

Wouter Kool and Matthew Botvinick. Mental labour. *Nature Human Behaviour*, 2 (12):899–908, December 2018. ISSN 2397-3374. doi: 10.1038/s41562-018-0401-9. URL https://www.nature.com/articles/s41562-018-0401-9. Publisher: Nature Publishing Group.

Wouter Kool, Joseph T. McGuire, Zev B. Rosen, and Matthew M. Botvinick. Decision making and the avoidance of cognitive demand. *Journal of Experimental Psychology: General*, 139(4):665–682, 2010. ISSN 1939-2222. doi: 10.1037/a0020198. Place: US Publisher: American Psychological Association.

I. Krechevsky. 'Hypothesis' Versus 'Chance' in The Pre-Solution Period In Sensory Discrimination Learning *. In *A Cognitive Theory of Learning*. Routledge, 1975. ISBN 978-1-003-31656-5. Num Pages: 12.

Dmitry Krotov. A new frontier for Hopfield networks. *Nature Reviews Physics*, 5(7): 366–367, July 2023. ISSN 2522-5820. doi: 10.1038/s42254-023-00595-y. URL https://www.nature.com/articles/s42254-023-00595-y. Publisher: Nature Publishing Group.

Dmitry Krotov and John Hopfield. Large Associative Memory Problem in Neurobiology and Machine Learning, April 2021. URL http://arxiv.org/abs/2008.06996. arXiv:2008.06996 [q-bio].

Kai A. Krueger and Peter Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, March 2009. ISSN 0010-0277. doi: 10.1016/j.cognition.2008.11.014. URL https://www.sciencedirect.com/science/article/pii/S0010027708002850.

Łukasz Kuśmierz, Takuya Isomura, and Taro Toyoizumi. Learning with three factors: modulating Hebbian plasticity with errors. *Current Opinion in Neurobiology*, 46:

170–177, October 2017. ISSN 0959-4388. doi: 10.1016/j.conb.2017.08.020. URL https://www.sciencedirect.com/science/article/pii/S0959438817300612.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct Epistemic Uncertainty Prediction. Technical Report arXiv:2102.08501, arXiv, April 2022. URL http://arxiv.org/abs/2102.08501. arXiv:2102.08501 [cs, stat] type: article.

Brenden M. Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06668-3. URL https://www.nature.com/articles/s41586-023-06668-3. Publisher: Nature Publishing Group.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, January 2017. ISSN 0140-525X, 1469-1825. doi: 10.1017/S0140525X16001837. URL https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/building-machines-that-learn-and-think-like-people/A9535B1D745A0377E16C590E14B94993.

P. E. Latham, B. J. Richmond, S. Nirenberg, and P. G. Nelson. Intrinsic Dynamics in Neuronal Networks. II. Experiment. *Journal of Neurophysiology*, 83(2):828–835, February 2000. ISSN 0022-3077. doi: 10.1152/jn.2000.83.2.828. URL https://journals.physiology.org/doi/full/10.1152/jn.2000.83.2.828. Publisher: American Physiological Society.

Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Advances in Neural Information Processing Systems*, volume 31. Curran Asso-

ciates, Inc., 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/2a38a4a9316c49e5a833517c45d31070-Abstract.html.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256. doi: 10.1109/5.726791. URL https://ieeexplore.ieee.org/document/726791. Conference Name: Proceedings of the IEEE.

Ping-Shien Lee and David K. Sewell. A revised diffusion model for conflict tasks. *Psychonomic Bulletin & Review*, 31(1):1–31, February 2024. ISSN 1531-5320. doi: 10.3758/s13423-023-02288-0. URL https://doi.org/10.3758/s13423-023-02288-0.

Rachel S. Lee, Yotam Sagiv, Ben Engelhard, Ilana B. Witten, and Nathaniel D. Daw. A feature-specific prediction error model explains dopaminergic heterogeneity. *Nature Neuroscience*, 27(8):1574–1586, August 2024a. ISSN 1546-1726. doi: 10.1038/s41593-024-01689-1. URL https://www.nature.com/articles/s41593-024-01689-1. Publisher: Nature Publishing Group.

Sebastian Lee, Stefano Sarao Mannelli, Claudia Clopath, Sebastian Goldt, and Andrew Saxe. Maslow's Hammer for Catastrophic Forgetting: Node Re-Use vs Node Activation, May 2022. URL http://arxiv.org/abs/2205.09029. arXiv:2205.09029.

Sebastian Lee, Samuel Liebana Garcia, Claudia Clopath, and Will Dabney. Lifelong Reinforcement Learning via Neuromodulation, August 2024b. URL http://arxiv.org/abs/2408.08446. arXiv:2408.08446 [cs].

Shane Legg and Marcus Hutter. Universal Intelligence: A Definition of Machine Intelligence, December 2007. URL http://arxiv.org/abs/0712.3329. arXiv:0712.3329 [cs].

Máté Lengyel and Peter Dayan. Hippocampal Contributions to Control: The Third Way. In *Advances in Neural Information Processing Systems*, volume 20. Curran

Associates, Inc., 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/1f4477bad7af3616c1f933a02bfabe4e-Abstract.html.

Daniel Levenstein, Veronica A. Alvarez, Asohan Amarasingham, Habiba Azab, Zhe S. Chen, Richard C. Gerkin, Andrea Hasenstaub, Ramakrishnan Iyer, Renaud B. Jolivet, Sarah Marzen, Joseph D. Monaco, Astrid A. Prinz, Salma Quraishi, Fidel Santamaria, Sabyasachi Shivkumar, Matthew F. Singh, Roger Traub, Farzan Nadim, Horacio G. Rotstein, and A. David Redish. On the Role of Theory and Modeling in Neuroscience. *Journal of Neuroscience*, 43(7):1074–1088, February 2023. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1179-22.2022. URL https://www.jneurosci.org/content/43/7/1074. Publisher: Society for Neuroscience Section: Viewpoints.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: meta-learning for domain generalization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 3490–3497, New Orleans, Louisiana, USA, February 2018. AAAI Press. ISBN 978-1-57735-800-8.

Fang Li, Mingbo Li, Wenyu Cao, Yang Xu, Yanwei Luo, Xiaolin Zhong, Jianyi Zhang, Ruping Dai, Xin-Fu Zhou, Zhiyuan Li, and Changqi Li. Anterior cingulate cortical lesion attenuates food foraging in rats. *Brain Research Bulletin*, 88(6):602–608, September 2012. ISSN 0361-9230. doi: 10.1016/j.brainresbull.2012.05.015. URL https://www.sciencedirect.com/science/article/pii/S0361923012001074.

Honglin Li, Payam Barnaghi, Shirin Enshaeifar, and Frieder Ganz. Continual Learning Using Bayesian Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):4243–4252, September 2021. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.3017292. URL https://ieeexplore.ieee.org/document/9181489. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Qianyi Li and Haim Sompolinsky. Statistical Mechanics of Deep Linear Neural Networks: The Backpropagating Kernel Renormalization. *Physical Review X*, 11(3):031059, September 2021. doi: 10.1103/PhysRevX.11.031059. URL https://link.aps.org/doi/10.1103/PhysRevX.11.031059. Publisher: American Physical Society.

Qianyi Li and Haim Sompolinsky. Globally Gated Deep Linear Networks, December 2022. URL http://arxiv.org/abs/2210.17449. arXiv:2210.17449 [physics].

Yujun Li, Rodrigo Carrasco-Davis, Younes Strittmatter, Stefano Sarao Mannelli, and Sebastian Musslick. A meta-learning framework for rationalizing cognitive fatigue in neural systems. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 46 (0), 2024. URL https://escholarship.org/uc/item/8pn5q3kx.

Shijun Liao. *Beyond Perturbation: Introduction to the Homotopy Analysis Method*. Chapman and Hall/CRC, New York, October 2003. ISBN 978-0-429-20861-4. doi: 10.1201/9780203491164.

Falk Lieder, Amitai Shenhav, Sebastian Musslick, and Thomas L. Griffiths. Rational metareasoning and the plasticity of cognitive control. *PLOS Computational Biology*, 14(4):e1006043, April 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006043. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006043. Publisher: Public Library of Science.

Timothy P. Lillicrap and Konrad P. Kording. What does it mean to understand a neural network?, July 2019. URL http://arxiv.org/abs/1907.06374. arXiv:1907.06374 [cs].

Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1):13276, November 2016. ISSN 2041-1723. doi: 10.1038/ncomms13276.

URL https://www.nature.com/articles/ncomms13276. Publisher: Nature Publishing Group.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, June 2020. ISSN 1471-0048. doi: 10.1038/s41583-020-0277-3. URL https://www.nature.com/articles/s41583-020-0277-3. Publisher: Nature Publishing Group.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3):293–321, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992699. URL https://doi.org/10.1007/BF00992699.

Grace W. Lindsay. Attention in Psychology, Neuroscience, and Machine Learning. *Frontiers in Computational Neuroscience*, 14, April 2020. ISSN 1662-5188. doi: 10.3389/fncom.2020.00029. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2020.00029/full. Publisher: Frontiers.

Grace W. Lindsay. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *Journal of Cognitive Neuroscience*, 33(10):2017–2031, September 2021. ISSN 0898-929X. doi: 10.1162/jocn_a_01544. URL https://doi.org/10.1162/jocn_a_01544.

Grace W Lindsay and Kenneth D Miller. How biological attention mechanisms improve task performance in a large-scale visual system model. *eLife*, 7:e38105, October 2018. ISSN 2050-084X. doi: 10.7554/eLife.38105. URL https://doi.org/10.7554/eLife.38105. Publisher: eLife Sciences Publications, Ltd.

Grace W. Lindsay, Daniel B. Rubin, and Kenneth D. Miller. A unified circuit model of attention: Neural and behavioral effects, July 2020. URL https://www.biorxiv.org/

content/10.1101/2019.12.13.875534v2. Pages: 2019.12.13.875534 Section: New Results.

Evan Z. Liu, Aditi Raghunathan, Percy Liang, and Chelsea Finn. Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6925–6935. PMLR, July 2021. URL https://proceedings.mlr.press/v139/liu21s.html. ISSN: 2640-3498.

Fanghui Liu, Johan A. K. Suykens, and Volkan Cevher. On the Double Descent of Random Features Models Trained with SGD, October 2022. URL http://arxiv.org/abs/2110.06910. arXiv:2110.06910 [stat].

Qi Liu, Yanjie Li, Shiyu Chen, Ke Lin, Xiongtao Shi, and Yunjiang Lou. Distributional reinforcement learning with epistemic and aleatoric uncertainty estimation. *Information Sciences*, 644:119217, October 2023. ISSN 0020-0255. doi: 10.1016/j.ins.2023.119217. URL https://www.sciencedirect.com/science/article/pii/S0020025523008022.

Ruishan Liu and James Zou. The Effects of Memory Replay in Reinforcement Learning, October 2017. URL http://arxiv.org/abs/1710.06574. arXiv:1710.06574 [cs, stat].

Yunzhe Liu, Raymond J. Dolan, Zeb Kurth-Nelson, and Timothy E.J. Behrens. Human Replay Spontaneously Reorganizes Experience. *Cell*, 178(3):640–652.e14, July 2019. ISSN 00928674. doi: 10.1016/j.cell.2019.06.012. URL https://linkinghub.elsevier.com/retrieve/pii/S0092867419306403.

Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping Coins to Estimate Pseudocounts for Exploration in Reinforcement Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 22594–22613. PMLR, July 2023. URL https://proceedings.mlr.press/v202/lobel23a.html. ISSN: 2640-3498.

Gordon D. Logan. Toward an instance theory of automatization. *Psychological Review*,

95(4):492–527, 1988. ISSN 1939-1471. doi: 10.1037/0033-295X.95.4.492. Place: US Publisher: American Psychological Association.

Hairong Lu, Dimitri Van der Linden, and Arnold B. Bakker. The neuroscientific basis of flow: Learning progress guides task engagement and cognitive control. *NeuroImage*, 308: 121076, March 2025. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2025.121076. URL https://www.sciencedirect.com/science/article/pii/S1053811925000783.

Björn Lütjens, Michael Everett, and Jonathan P. How. Safe Reinforcement Learning with Model Uncertainty Estimates, March 2019. URL http://arxiv.org/abs/1810.08700. arXiv:1810.08700 [cs].

Colin M. MacLeod. Half a century of research on the Stroop effect: An integrative review. *Psychological Bulletin*, 109(2):163–203, 1991. ISSN 1939-1455. doi: 10.1037/0033-2909.109.2.163. Place: US Publisher: American Psychological Association.

Jeffrey C. Magee and Christine Grienberger. Synaptic Plasticity Forms and Functions. *Annual Review of Neuroscience*, 43(1):95–117, 2020. doi: 10.1146/annurev-neuro-090919-022842. URL https://doi.org/10.1146/annurev-neuro-090919-022842. _eprint: https://doi.org/10.1146/annurev-neuro-090919-022842.

Vincent Mai, Kaustubh Mani, and Liam Paull. Sample Efficient Deep Reinforcement Learning via Uncertainty Estimation, May 2022. URL http://arxiv.org/abs/2201.01666. arXiv:2201.01666 [cs].

Javier Alejandro Masis, Sebastian Musslick, and Jonathan D. Cohen. Learning expectations shape cognitive control allocation, August 2024. URL https://osf.io/d2cbg.

Javier Masís, Travis Chapman, Juliana Y Rhee, David D Cox, and Andrew M Saxe. Strategically managing learning during perceptual decision making. *eLife*, 12:e64978,

February 2023. ISSN 2050-084X. doi: 10.7554/eLife.64978. URL https://doi.org/10.7554/eLife.64978. Publisher: eLife Sciences Publications, Ltd.

Javier Alejandro Masís, Sebastian Musslick, and Jonathan Cohen. The Value of Learning and Cognitive Control Allocation. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2021. URL https://escholarship.org/uc/item/7w0223v0.

Marcelo G. Mattar and Nathaniel D. Daw. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11):1609–1617, November 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0232-z. URL https://www.nature.com/articles/s41593-018-0232-z. Publisher: Nature Publishing Group.

Augustine Mavor-Parker, Kimberly Young, Caswell Barry, and Lewis Griffin. How to Stay Curious while avoiding Noisy TVs using Aleatoric Uncertainty Estimation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 15220–15240. PMLR, June 2022. URL https://proceedings.mlr.press/v162/mavor-parker22a.html. ISSN: 2640-3498.

Colin G. McNamara, Álvaro Tejero-Cantero, Stéphanie Trouche, Natalia Campo-Urriza, and David Dupret. Dopaminergic neurons promote hippocampal reactivation and spatial memory persistence. *Nature Neuroscience*, 17(12):1658–1660, December 2014. ISSN 1546-1726. doi: 10.1038/nn.3843. URL https://www.nature.com/articles/nn.3843. Number: 12 Publisher: Nature Publishing Group.

Qiaozhu Mei, Yutong Xie, Walter Yuan, and Matthew O. Jackson. A Turing test of whether AI chatbots are behaviorally similar to humans. *Proceedings of the National Academy of Sciences*, 121(9):e2313925121, February 2024. doi: 10.1073/pnas.2313925121. URL https://www.pnas.org/doi/10.1073/pnas.2313925121. Publisher: Proceedings of the National Academy of Sciences.

Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Meta-

Learning Update Rules for Unsupervised Representation Learning, February 2019. URL http://arxiv.org/abs/1804.00222. arXiv:1804.00222 [cs].

Francesca Mignacco, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification. In *Advances in Neural Information Processing Systems*, volume 33, pages 9540–9550. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6c81c83c4bd0b58850495f603ab45a93-Abstract.html.

Steven Miletić and Leendert van Maanen. Caution in decision-making under time pressure is mediated by timing ability. *Cognitive Psychology*, 110:16–29, May 2019. ISSN 0010-0285. doi: 10.1016/j.cogpsych.2019.01.002. URL https://www.sciencedirect.com/science/article/pii/S0010028518302445.

George A. Miller, Eugene Galanter, and Karl H. Pribram. *Plans and the structure of behavior*. Plans and the structure of behavior. Henry Holt and Co, New York, NY, US, 1960. doi: 10.1037/10039-000. Pages: x, 222.

Beren Millidge, Anil Seth, and Christopher L. Buckley. Predictive Coding: a Theoretical and Experimental Review, July 2022. URL http://arxiv.org/abs/2107.12979. arXiv:2107.12979 [cs].

Walter Mischel. *The Marshmallow Test: Mastering self-control*. The Marshmallow Test: Mastering self-control. Little, Brown and Co, New York, NY, US, 2014. ISBN 978-0-316-23087-2 978-0-316-23085-8. Pages: 328.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN

1476-4687. doi: 10.1038/nature14236. URL https://www.nature.com/articles/nature14236. Number: 7540 Publisher: Nature Publishing Group.

Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, October 1993. ISSN 1573-0565. doi: 10.1007/BF00993104. URL https://doi.org/10.1007/BF00993104.

Agnes Moors and Jan De Houwer. Automaticity: A Theoretical and Conceptual Analysis. *Psychological Bulletin*, 132(2):297–326, 2006. ISSN 1939-1455. doi: 10.1037/0033-2909.132.2.297. Place: US Publisher: American Psychological Association.

Francesco Mori, Stefano Sarao Mannelli, and Francesca Mignacco. Optimal Protocols for Continual Learning via Statistical Physics and Control Theory, March 2025. URL http://arxiv.org/abs/2409.18061. arXiv:2409.18061 [cs].

R. G. M Morris. D.O. Hebb: *The Organization of Behavior*, Wiley: New York; 1949. *Brain Research Bulletin*, 50(5):437, November 1999. ISSN 0361-9230. doi: 10.1016/S0361-9230(99)00182-3. URL https://www.sciencedirect.com/science/article/pii/S0361923099001823.

Edvard I. Moser, Emilio Kropff, and May-Britt Moser. Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience*, 31(Volume 31, 2008):69–89, July 2008. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.31.061307.090723. URL https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.31.061307.090723. Publisher: Annual Reviews.

Sebastian Musslick and Jonathan D. Cohen. Rationalizing constraints on the capacity for cognitive control. *Trends in Cognitive Sciences*, 25(9):757–775, September 2021. ISSN 1364-6613. doi: 10.1016/j.tics.2021.06.001. URL https://www.sciencedirect.com/science/article/pii/S1364661321001480.

Sebastian Musslick and Javier Masís. Pushing the Bounds of Bounded Optimality and Rationality. *Cognitive Science*, 47(4):e13259, 2023. ISSN 1551-6709. doi: 10.1111/cogs.13259. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.13259. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.13259.

Sebastian Musslick, Matthew M Botvinick, Amitai Shenhav, and Jonathan D Cohen. A computational model of control allocation based on the Expected Value of Control. page 6.

Sebastian Musslick, Andrew Saxe, Abigail Novick Hoskin, Yotam Sagiv, Daniel Reichman, Giovanni Petri, and Jonathan D. Cohen. On the Rational Boundedness of Cognitive Control: Shared Versus Separated Representations, November 2020. URL https://osf.io/jkhdf.

Lynn Nadel and Oliver Hardt. Update on Memory Systems and Processes. *Neuropsychopharmacology*, 36(1):251–273, January 2011. ISSN 1740-634X. doi: 10.1038/npp.2010.169. URL https://www.nature.com/articles/npp2010169. Publisher: Nature Publishing Group.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming Exploration in Reinforcement Learning with Demonstrations, February 2018. URL http://arxiv.org/abs/1709.10089. arXiv:1709.10089 [cs].

Kensuke Nakamura, Bilel Derbel, Kyoung-Jae Won, and Byung-Woo Hong. Learning-Rate Annealing Methods for Deep Neural Networks. *Electronics*, 10(16):2029, January 2021. ISSN 2079-9292. doi: 10.3390/electronics10162029. URL https://www.mdpi.com/2079-9292/10/16/2029. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

Hiromi Narimatsu, Mayuko Ozawa, and Shiro Kumano. Collision Probability Matching Loss for Disentangling Epistemic Uncertainty from Aleatoric Uncertainty. In *Pro-*

ceedings of The 26th International Conference on Artificial Intelligence and Statistics, pages 11355–11370. PMLR, April 2023. URL https://proceedings.mlr.press/v206/narimatsu23a.html. ISSN: 2640-3498.

Oliver Nelles. *Nonlinear System Identification*. Springer, Berlin, Heidelberg, 2001. ISBN 978-3-642-08674-8 978-3-662-04323-3. doi: 10.1007/978-3-662-04323-3. URL http://link.springer.com/10.1007/978-3-662-04323-3.

Thomas O. Nelson. Metamemory: A Theoretical Framework and New Findings. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 26, pages 125–173. Academic Press, January 1990. doi: 10.1016/S0079-7421(08)60053-5. URL https://www.sciencedirect.com/science/article/pii/S0079742108600535.

Y. Nesterov. A method for solving the convex programming problem with convergence rate O(1/k^2). *Proceedings of the USSR Academy of Sciences*, 1983. URL https://www.semanticscholar.org/paper/A-method-for-solving-the-convex-programming-problem-Nesterov/8d3a318b62d2e970122da35b2a2e70a5d12cc16f.

Allen Newell and Herbert A. Simon. *Human problem solving*. Human problem solving. Prentice-Hall, Oxford, England, 1972. Pages: xiv, 920.

Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1): 89–122, January 2022. ISSN 1573-0565. doi: 10.1007/s10994-021-06003-9. URL https://doi.org/10.1007/s10994-021-06003-9.

Rui Nian, Jinfeng Liu, and Biao Huang. A review On reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139: 106886, August 2020. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2020.106886. URL https://www.sciencedirect.com/science/article/pii/S0098135420300557.

Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms, October 2018. URL http://arxiv.org/abs/1803.02999. arXiv:1803.02999 [cs].

Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause. Information-Directed Exploration for Deep Reinforcement Learning, March 2019. URL http://arxiv.org/abs/1812.07544. arXiv:1812.07544 [cs, stat].

Qian Niu, Junyu Liu, Ziqian Bi, Pohsun Feng, Benji Peng, Keyu Chen, Ming Li, Lawrence KQ Yan, Yichao Zhang, Caitlyn Heqi Yin, Cheng Fei, Tianyang Wang, Yunze Wang, Silin Chen, and Ming Liu. Large Language Models and Cognitive Science: A Comprehensive Review of Similarities, Differences, and Challenges, December 2024. URL http://arxiv.org/abs/2409.02387. arXiv:2409.02387 [cs].

Johan S. Obando-Ceron and Pablo Samuel Castro. Revisiting Rainbow: Promoting more Insightful and Inclusive Deep Reinforcement Learning Research, May 2021. URL http://arxiv.org/abs/2011.14826. arXiv:2011.14826 [cs].

Samuel Ocko, Jack Lindsey, Surya Ganguli, and Stephane Deny. The emergence of multiple retinal cell types through efficient coding of natural movies. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/d94fd74dcde1aa553be72c1006578b23-Abstract.html.

Brendan O'Donoghue. Efficient Exploration via Epistemic-Risk-Seeking Policy Optimization, June 2023. URL http://arxiv.org/abs/2302.09339. arXiv:2302.09339 [cs].

K. Okajima. The Gabor function extracts the maximum information from input local signals. *Neural Networks*, 11(3):435–439, April 1998. ISSN 0893-6080. doi: 10.1016/

S0893-6080(98)00008-2. URL https://www.sciencedirect.com/science/article/pii/S0893608098000082.

Joosep Olop, Mikk Granström, and Eve Kikas. Students' metacognitive knowledge of learning-strategy effectiveness and their recall of teachers' strategy instructions. *Frontiers in Education*, 9, May 2024. ISSN 2504-284X. doi: 10.3389/feduc.2024.1307485. URL https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2024.1307485/full. Publisher: Frontiers.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, and others. Dota 2 with Large Scale Deep Reinforcement Learning, December 2019. URL http://arxiv.org/abs/1912.06680. arXiv:1912.06680 [cs, stat].

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep Exploration via Bootstrapped DQN. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8d8818c8e140c64c743113f563cf750f-Abstract.html.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 8626–8638, Red Hook, NY, USA, 2018. Curran Associates Inc.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic Neural Networks, May 2023. URL http://arxiv.org/abs/2107.08924. arXiv:2107.08924 [cs].

Georg Ostrovski, Marc G. Bellemare, Aäron Oord, and Rémi Munos. Count-Based Exploration with Neural Density Models. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2721–2730. PMLR, July 2017. URL https://proceedings.mlr.press/v70/ostrovski17a.html. ISSN: 2640-3498.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, 1, 2007. ISSN 1662-5218. URL https://www.frontiersin.org/articles/10.3389/neuro.12.006.2007.

Nicolas Palanca-Castan, Beatriz Sánchez Tajadura, and Rodrigo Cofré. Towards an interdisciplinary framework about intelligence. *Heliyon*, 7(2), February 2021. ISSN 2405-8440. doi: 10.1016/j.heliyon.2021.e06268. URL https://www.cell.com/heliyon/abstract/S2405-8440(21)00373-X. Publisher: Elsevier.

Yangchen Pan, Jincheng Mei, Amir-massoud Farahmand, Martha White, Hengshuai Yao, Mohsen Rohani, and Jun Luo. Understanding and mitigating the limitations of prioritized experience replay. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 1561–1571. PMLR, August 2022. URL https://proceedings.mlr.press/v180/pan22a.html. ISSN: 2640-3498.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71, May 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.01.012. URL https://www.sciencedirect.com/science/article/pii/S0893608019300231.

Nishil Patel, Sebastian Lee, Stefano Sarao Mannelli, Sebastian Goldt, and Andrew Saxe. The RL Perceptron: Generalisation Dynamics of Policy Learning in High Dimensions, December 2024. URL http://arxiv.org/abs/2306.10404. arXiv:2306.10404 [cs].

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven Exploration by Self-supervised Prediction, May 2017. URL http://arxiv.org/abs/1705.05363. arXiv:1705.05363 [cs, stat].

Ivan P. Pavlov. The Scientific Investigation of the Psychical Faculties or Processes in the Higher Animals. *Science*, 24(620):613–619, November 1906. doi: 10.1126/science.24.620.

613. URL https://www.science.org/doi/10.1126/science.24.620.613. Publisher: American Association for the Advancement of Science.

Omar David Perez and Gonzalo P. Urcelay. Delayed rewards weaken human goal directed actions. *npj Science of Learning*, 10(1):36, June 2025. ISSN 2056-7936. doi: 10.1038/s41539-025-00325-2. URL https://www.nature.com/articles/s41539-025-00325-2. Publisher: Nature Publishing Group.

Charlotte Piette, Nicolas Gervasi, and Laurent Venance. Synaptic plasticity through a naturalistic lens. *Frontiers in Synaptic Neuroscience*, 15, December 2023. ISSN 1663-3563. doi: 10.3389/fnsyn.2023.1250753. URL https://www.frontiersin.org/journals/synaptic-neuroscience/articles/10.3389/fnsyn.2023.1250753/full. Publisher: Frontiers.

Payam Piray and Nathaniel D. Daw. Linear reinforcement learning in planning, grid fields, and cognitive control. *Nature Communications*, 12(1):4942, August 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25123-3. URL https://www.nature.com/articles/s41467-021-25123-3. Number: 1 Publisher: Nature Publishing Group.

Russell A. Poldrack, Fred W. Sabb, Karin Foerde, Sabrina M. Tom, Robert F. Asarnow, Susan Y. Bookheimer, and Barbara J. Knowlton. The Neural Correlates of Motor Skill Automaticity. *Journal of Neuroscience*, 25(22):5356–5364, June 2005. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.3880-04.2005. URL https://www.jneurosci.org/content/25/22/5356. Publisher: Society for Neuroscience Section: Behavioral/Systems/Cognitive.

Alexandre Pouget, Jeffrey M. Beck, Wei Ji Ma, and Peter E. Latham. Probabilistic brains: knowns and unknowns. *Nature Neuroscience*, 16(9):1170–1178, September 2013. ISSN 1546-1726. doi: 10.1038/nn.3495. URL https://www.nature.com/articles/nn.3495. Publisher: Nature Publishing Group.

Alexandre Pouget, Jan Drugowitsch, and Adam Kepecs. Confidence and certainty: distinct probabilistic quantities for different goals. *Nature Neuroscience*, 19(3):366–374, March 2016. ISSN 1546-1726. doi: 10.1038/nn.4240. URL https://www.nature.com/articles/nn.4240. Publisher: Nature Publishing Group.

Bharat Prakash, Mark Horton, Nicholas R. Waytowich, William David Hairston, Tim Oates, and Tinoosh Mohsenin. On the use of Deep Autoencoders for Efficient Embedded Reinforcement Learning. In *Proceedings of the 2019 Great Lakes Symposium on VLSI*, GLSVLSI '19, pages 507–512, New York, NY, USA, May 2019. Association for Computing Machinery. ISBN 978-1-4503-6252-8. doi: 10.1145/3299874.3319493. URL https://doi.org/10.1145/3299874.3319493.

L. A. Prashanth and Mohammad Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning*, 105(3): 367–417, December 2016. ISSN 1573-0565. doi: 10.1007/s10994-016-5569-5. URL https://doi.org/10.1007/s10994-016-5569-5.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural Episodic Control, March 2017. URL http://arxiv.org/abs/1703.01988. arXiv:1703.01988 [cs, stat].

Charlotte Prévost, Mathias Pessiglione, Elise Météreau, Marie-Laure Cléry-Melin, and Jean-Claude Dreher. Separate Valuation Subsystems for Delay and Effort Decision Costs. *Journal of Neuroscience*, 30(42):14080–14090, October 2010. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2752-10.2010. URL https://www.jneurosci.org/content/30/42/14080. Publisher: Society for Neuroscience Section: Articles.

J.-Y. Puigbò, X. D. Arsiwalla, M. A. González-Ballester, and P. F. M. J. Verschure. Switching Operation Modes in the Neocortex via Cholinergic Neuromodulation. *Molecular Neurobiology*, 57(1):139–149, January 2020. ISSN 1559-1182. doi: 10.1007/s12035-019-01764-w. URL https://doi.org/10.1007/s12035-019-01764-w.

Friedemann Pulvermüller, Rosario Tomasello, Malte R. Henningsen-Schomers, and Thomas Wennekers. Biological constraints on neural network models of cognitive function. *Nature Reviews Neuroscience*, 22(8):488–502, August 2021. ISSN 1471-0048. doi: 10.1038/s41583-021-00473-5. URL https://www.nature.com/articles/s41583-021-00473-5. Publisher: Nature Publishing Group.

Mohaimenul Azam Khan Raiaan, Md. Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access*, 12:26839–26874, 2024. ISSN 2169-3536. doi: 10.1109/ACCESS.2024.3365742. URL https://ieeexplore.ieee.org/document/10433480. Conference Name: IEEE Access.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-Learning with Implicit Gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/072b030ba126b2f4b2374f342be9ed44-Abstract.html.

Daniel Randles, Iain Harlow, and Michael Inzlicht. A pre-registered naturalistic observation of within domain mental fatigue and domain-general depletion of self-control. *PLOS ONE*, 12(9):e0182980, September 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0182980. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0182980. Publisher: Public Library of Science.

Sachin Ravi, Sebastian Musslick, Maia Hamin, Theodore L. Willke, and Jonathan D. Cohen. Navigating the Trade-Off between Multi-Task Learning and Learning to Multitask in Deep Neural Networks, January 2021. URL http://arxiv.org/abs/2007.10527. arXiv:2007.10527 [cs, stat].

Christian Raymond, Qi Chen, Bing Xue, and Mengjie Zhang. Learning Symbolic Model-

Agnostic Loss Functions via Meta-Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13699–13714, November 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2023.3294394. URL https://ieeexplore.ieee.org/document/10177983. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Gal Raz and Rebecca Saxe. Learning in Infancy Is Active, Endogenously Motivated, and Depends on the Prefrontal Cortices. *Annual Review of Developmental Psychology*, 2(Volume 2, 2020):247–268, December 2020. ISSN 2640-7922. doi: 10.1146/annurev-devpsych-121318-084841. URL https://www.annualreviews.org/content/journals/10.1146/annurev-devpsych-121318-084841. Publisher: Annual Reviews.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond, April 2019. URL http://arxiv.org/abs/1904.09237. arXiv:1904.09237 [cs].

Chi Ren, Kailong Peng, Ruize Yang, Weikang Liu, Chang Liu, and Takaki Komiyama. Global and subtype-specific modulation of cortical inhibitory neurons regulated by acetylcholine during motor learning. *Neuron*, 110(14):2334–2350.e8, July 2022. ISSN 0896-6273. doi: 10.1016/j.neuron.2022.04.031. URL https://www.sciencedirect.com/science/article/pii/S0896627322004081.

Peter D. Renshaw and Colin Power. The Process of Learning and Human Development. In John P. Keeves, Ryo Watanabe, Rupert Maclean, Peter D. Renshaw, Colin N. Power, Robyn Baker, S. Gopinathan, Ho Wah Kam, Yin Cheong Cheng, and Albert C. Tuijnman, editors, *International Handbook of Educational Research in the Asia-Pacific Region: Part One*, pages 351–363. Springer Netherlands, Dordrecht, 2003. ISBN 978-94-017-3368-7. doi: 10.1007/978-94-017-3368-7_25. URL https://doi.org/10.1007/978-94-017-3368-7_25.

Blake A. Richards, Timothy P. Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz,

Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, Colleen J. Gillon, Danijar Hafner, Adam Kepecs, Nikolaus Kriegeskorte, Peter Latham, Grace W. Lindsay, Kenneth D. Miller, Richard Naud, Christopher C. Pack, Panayiota Poirazi, Pieter Roelfsema, João Sacramento, Andrew Saxe, Benjamin Scellier, Anna C. Schapiro, Walter Senn, Greg Wayne, Daniel Yamins, Friedemann Zenke, Joel Zylberberg, Denis Therien, and Konrad P. Kording. A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11):1761–1770, November 2019. ISSN 1546-1726. doi: 10.1038/s41593-019-0520-2. URL https://www.nature.com/articles/s41593-019-0520-2. Publisher: Nature Publishing Group.

Timothy C. Rickard. Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, 126(3):288–311, 1997. ISSN 1939-2222. doi: 10.1037/0096-3445.126.3.288. Place: US Publisher: American Psychological Association.

F. E. Ritter and L. J. Schooler. Learning Curve, The. In Neil J. Smelser and Paul B. Baltes, editors, *International Encyclopedia of the Social & Behavioral Sciences*, pages 8602–8605. Pergamon, Oxford, January 2001. ISBN 978-0-08-043076-8. doi: 10.1016/B0-08-043076-7/01480-7. URL https://www.sciencedirect.com/science/article/pii/B0080430767014807.

Samuel Ritter, Jane X. Wang, Zeb Kurth-Nelson, Siddhant M. Jayakumar, Charles Blundell, Razvan Pascanu, and Matthew Botvinick. Been There, Done That: Meta-Learning with Episodic Recall, July 2018. URL http://arxiv.org/abs/1805.09692. arXiv:1805.09692 [stat].

Tapas Roy and Dilip K. Maiti. An optimal and modified homotopy perturbation method for strongly nonlinear differential equations. *Nonlinear Dynamics*, 111(16):15215–15231, August 2023. ISSN 1573-269X. doi: 10.1007/s11071-023-08662-w. URL https://doi.org/10.1007/s11071-023-08662-w.

Sebastian Ruder. An overview of gradient descent optimization algorithms, June 2017. URL http://arxiv.org/abs/1609.04747. arXiv:1609.04747 [cs].

David E. Rumelhart and James L. McClelland. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pages 318–362. MIT Press, 1987. ISBN 978-0-262-29140-8. URL https://ieeexplore.ieee.org/document/6302929. Conference Name: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations.

R. Russo, H. J. Herrmann, and L. de Arcangelis. Brain modularity controls the critical behavior of spontaneous activity. *Scientific Reports*, 4(1):4312, March 2014. ISSN 2045-2322. doi: 10.1038/srep04312. URL https://www.nature.com/articles/srep04312. Publisher: Nature Publishing Group.

David Saad and Sara A. Solla. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225–4243, October 1995. doi: 10.1103/PhysRevE.52.4225. URL https://link.aps.org/doi/10.1103/PhysRevE.52.4225. Publisher: American Physical Society.

Yotam Sagiv, Sebastian Musslick, Yael Niv, and Jonathan D. Cohen. Efficiency of learning vs. processing: Towards a normative theory of multitasking, July 2020. URL http://arxiv.org/abs/2007.03124. arXiv:2007.03124 [q-bio].

Luca Saglietti, Stefano Sarao Mannelli, and Andrew Saxe. An Analytical Theory of Curriculum Learning in Teacher-Student Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(11):114014, November 2022. ISSN 1742-5468. doi: 10.1088/1742-5468/ac9b3c. URL http://arxiv.org/abs/2106.08068. arXiv:2106.08068 [cond-mat, stat].

Kai Jappe Sandbrink, Christopher Summerfield, and Laurence Hunt. Learning the

value of control with Deep RL. January 2024. doi: 10.17605/OSF.IO/9EWT8. URL https://osf.io/9ewt8. Publisher: OSF.

Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic Curiosity through Reachability, August 2019. URL http://arxiv.org/abs/1810.02274. arXiv:1810.02274 [cs, stat].

Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, January 2021. ISSN 1471-0048. doi: 10.1038/s41583-020-00395-8. URL https://www.nature.com/articles/s41583-020-00395-8. Publisher: Nature Publishing Group.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, June 2019. doi: 10.1073/pnas.1820226116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1820226116. Publisher: Proceedings of the National Academy of Sciences.

Andrew M. Saxe, Shagun Sodhani, and Sam Lewallen. The Neural Race Reduction: Dynamics of Abstraction in Gated Networks, July 2022. URL http://arxiv.org/abs/2207.10430. arXiv:2207.10430 [cs].

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay, February 2016. URL http://arxiv.org/abs/1511.05952. arXiv:1511.05952 [cs].

Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers, August 2021. URL http://arxiv.org/abs/2007.01547. arXiv:2007.01547 [cs].

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kai-

lyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like?, January 2020. URL https://www.biorxiv.org/content/10.1101/407007v2. Pages: 407007 Section: New Results.

Wolfram Schultz, Peter Dayan, and P. Read Montague. A Neural Substrate of Prediction and Reward. *Science*, 275(5306):1593–1599, March 1997. doi: 10.1126/science.275.5306.1593. URL https://www.science.org/doi/10.1126/science.275.5306.1593. Publisher: American Association for the Advancement of Science.

Max Schwarzer, Johan Obando-Ceron, Aaron Courville, Marc Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, Better, Faster: Human-level Atari with human-level efficiency, November 2023. URL http://arxiv.org/abs/2305.19452. arXiv:2305.19452 [cs].

Reza Shadmehr, Jean Jacques Orban de Xivry, Minnan Xu-Wilson, and Ting-Yu Shih. Temporal Discounting of Reward and the Cost of Time in Motor Control. *Journal of Neuroscience*, 30(31):10507–10516, August 2010. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1343-10.2010. URL https://www.jneurosci.org/content/30/31/10507. Publisher: Society for Neuroscience Section: Articles.

Sohan Shankar, Yi Pan, Hanqi Jiang, Zhengliang Liu, Mohammad R. Darbandi, Agustin Lorenzo, Junhao Chen, Md Mehedi Hasan, Arif Hassan Zidan, Eliana Gelman, Joshua A. Konfrst, Jillian Y. Russell, Katelyn Fernandes, Tianze Yang, Yiwei Li, Huaqin Zhao, Afrar Jahin, Triparna Ganguly, Shair Dinesha, Yifan Zhou, Zihao Wu, Xinliang Li, Lokesh Adusumilli, Aziza Hussein, Sagar Nookarapu, Jixin Hou, Kun Jiang, Jiaxi Li, Brenden Heinel, XianShen Xi, Hailey Hubbard, Zayna Khan, Levi Whitaker, Ivan Cao, Max Allgaier, Andrew Darby, Lin Zhao, Lu Zhang, Xiaoqiao Wang, Xiang Li, Wei Zhang, Xiaowei Yu, Dajiang Zhu, Yohannes Abate, and Tianming Liu. Bridging Brains and Machines: A Unified Frontier in Neuroscience, Artificial Intelligence, and Neuromorphic

Systems, July 2025. URL http://arxiv.org/abs/2507.10722. arXiv:2507.10722 [q-bio].

Amitai Shenhav, Matthew M. Botvinick, and Jonathan D. Cohen. The Expected Value of Control: An Integrative Theory of Anterior Cingulate Cortex Function. *Neuron*, 79(2):217–240, July 2013. ISSN 0896-6273. doi: 10.1016/j.neuron.2013.07.007. URL https://www.sciencedirect.com/science/article/pii/S0896627313006077.

Amitai Shenhav, Sebastian Musslick, Falk Lieder, Wouter Kool, Thomas L. Griffiths, Jonathan D. Cohen, and Matthew M. Botvinick. Toward a Rational and Mechanistic Account of Mental Effort. *Annual Review of Neuroscience*, 40(1):99–124, 2017. doi: 10.1146/annurev-neuro-072116-031526. URL https://doi.org/10.1146/annurev-neuro-072116-031526. _eprint: https://doi.org/10.1146/annurev-neuro-072116-031526.

Craig Sherstan, Dylan R Ashley, Brendan Bennett, Kenny Young, Adam White, Martha White, and Richard S Sutton. Comparing Direct and Indirect Temporal-Difference Methods for Estimating the Variance of the Return. 2018.

Navid Shervani-Tabar and Robert Rosenbaum. Meta-learning biologically plausible plasticity rules with random feedback pathways. *Nature Communications*, 14(1): 1805, March 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-37562-1. URL https://www.nature.com/articles/s41467-023-37562-1. Publisher: Nature Publishing Group.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/

nature16961. URL https://www.nature.com/articles/nature16961%7D. Number: 7587 Publisher: Nature Publishing Group.

David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, October 2021. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103535. URL https://www.sciencedirect.com/science/article/pii/S0004370221000862.

Massimo Silvetti, Stefano Lasaponara, Nabil Daddaoua, Mattias Horan, and Jacqueline Gottlieb. A Reinforcement Meta-Learning framework of executive function and information demand. *Neural Networks*, 157:103–113, January 2023. ISSN 0893-6080. doi: 10.1016/j.neunet.2022.10.004. URL https://www.sciencedirect.com/science/article/pii/S0893608022003884.

B. F. Skinner. Are theories of learning necessary? *Psychological Review*, 57(4):193–216, 1950. ISSN 1939-1471. doi: 10.1037/h0054367. Place: US Publisher: American Psychological Association.

B. F. Skinner. Cognitive science and behaviourism. *British Journal of Psychology*, 76(3):291–301, 1985. ISSN 2044-8295. doi: 10.1111/j.2044-8295.1985.tb01953.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2044-8295.1985.tb01953.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2044-8295.1985.tb01953.x.

Matthew J. Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, December 1982. ISSN 0021-9002, 1475-6072. doi: 10.2307/3213832. URL https://www.cambridge.org/core/journals/journal-of-applied-probability/article/abs/variance-of-discounted-markov-decision-processes/AA4549BFA70081B27C0092F4BF9C661A. Publisher: Cambridge University Press.

Hansem Sohn and Devika Narain. Neural implementations of Bayesian inference. *Current Opinion in Neurobiology*, 70:121–129, October 2021. ISSN 0959-4388. doi: 10.1016/j.conb.2021.09.008. URL https://www.sciencedirect.com/science/article/pii/S0959438821001082.

Jaehyeon Son, Soochan Lee, and Gunhee Kim. When Meta-Learning Meets Online and Continual Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2024.3463709. URL https://ieeexplore.ieee.org/document/10684017. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Lisa K. Son and Rajiv Sethi. Metacognitive Control and Optimal Learning. *Cognitive Science*, 30(4):759–774, 2006. ISSN 1551-6709. doi: 10.1207/s15516709cog0000_74. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0000_74. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog0000_74.

Lisa K. Son and Rajiv Sethi. Adaptive Learning and the Allocation of Time. *Adaptive Behavior*, 18(2):132–140, April 2010. ISSN 1059-7123. doi: 10.1177/1059712309344776. URL https://doi.org/10.1177/1059712309344776. Publisher: SAGE Publications Ltd STM.

Yuhang Song, Beren Millidge, Tommaso Salvatori, Thomas Lukasiewicz, Zhenghua Xu, and Rafal Bogacz. Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature Neuroscience*, 27(2):348–358, February 2024. ISSN 1546-1726. doi: 10.1038/s41593-023-01514-1. URL https://www.nature.com/articles/s41593-023-01514-1. Publisher: Nature Publishing Group.

Kassandra R. Spurlock. Review of Learning styles, classroom instruction, and student achievement. *Education Review*, 30, April 2023. ISSN 1094-5296. doi: 10.14507/er.v30.3713. URL https://edrev.asu.edu/index.php/ER/article/view/3713.

Warisa Sritriratanarak and Paulo Garcia. On a Functional Definition of Intelligence, December 2023. URL http://arxiv.org/abs/2312.09546. arXiv:2312.09546 [cs].

Bradly C. Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing Exploration In Reinforcement Learning With Deep Predictive Models, November 2015. URL http://arxiv.org/abs/1507.00814. arXiv:1507.00814 [cs, stat].

Mario Stampanoni Bassi, Ennio Iezzi, Luana Gilio, Diego Centonze, and Fabio Buttari. Synaptic Plasticity Shapes Brain Connectivity: Implications for Network Topology. *International Journal of Molecular Sciences*, 20(24):6193, January 2019. ISSN 1422-0067. doi: 10.3390/ijms20246193. URL https://www.mdpi.com/1422-0067/20/24/6193. Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

Emmanouil Stergiadis, Priyanka Agrawal, and Oliver Squire. Curriculum Meta-Learning for Few-shot Classification, December 2021. URL http://arxiv.org/abs/2112.02913. arXiv:2112.02913 [cs].

R. Sternberg and J. Kagan. Intelligence Applied: Understanding and Increasing Your Intellectual Skills. 1986. URL https://www.semanticscholar.org/paper/Intelligence-Applied%3A-Understanding-and-Increasing-Sternberg-Kagan/f42576591b3a52b092a97522835226eecee10339.

Caleb Stone, Jason B. Mattingley, and Dragan Rangelov. Neural mechanisms of metacognitive improvement under speed pressure. *Communications Biology*, 8(1): 223, February 2025. ISSN 2399-3642. doi: 10.1038/s42003-025-07646-3. URL https://www.nature.com/articles/s42003-025-07646-3. Publisher: Nature Publishing Group.

J. R. Stroop. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18(6):643–662, 1935. ISSN 0022-1015. doi: 10.1037/h0054651. Place: US Publisher: Psychological Review Company.

Ajay Subramanian, Sharad Chitlangia, and Veeky Baths. Reinforcement learning and its connections with neuroscience and psychology. *Neural Networks*, 145:271–287, January 2022. ISSN 0893-6080. doi: 10.1016/j.neunet.2021.10.003. URL https://www.sciencedirect.com/science/article/pii/S0893608021003944.

Peiquan Sun, Wengang Zhou, and Houqiang Li. Attentive Experience Replay. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5900–5907, April 2020. ISSN 2374-3468. doi: 10.1609/aaai.v34i04.6049. URL https://ojs.aaai.org/index.php/AAAI/article/view/6049. Number: 04.

Weinan Sun, Madhu Advani, Nelson Spruston, Andrew Saxe, and James E. Fitzgerald. Organizing memories for generalization in complementary learning systems. *Nature Neuroscience*, 26(8):1438–1448, August 2023. ISSN 1546-1726. doi: 10.1038/s41593-023-01382-9. URL https://www.nature.com/articles/s41593-023-01382-9. Publisher: Nature Publishing Group.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, August 1988. ISSN 1573-0565. doi: 10.1007/BF00115009. URL https://doi.org/10.1007/BF00115009.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, July 1991. ISSN 0163-5719. doi: 10.1145/122344.122377. URL https://dl.acm.org/doi/10.1145/122344.122377.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 978-0-262-03924-6.

Remi Tachet, Mohammad Pezeshki, Samira Shabanian, Aaron Courville, and Yoshua Bengio. On the Learning Dynamics of Deep Neural Networks, December 2020. URL http://arxiv.org/abs/1809.06848. arXiv:1809.06848 [cs].

Aviv Tamar, Dotan Di Castro, and Shie Mannor. Policy gradients with variance related

risk criteria. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1651–1658, Madison, WI, USA, June 2012. Omnipress. ISBN 978-1-4503-1285-1.

Aviv Tamar, Dotan Di Castro, and Shie Mannor. Learning the Variance of the Reward-To-Go. *Journal of Machine Learning Research*, 17(13):1–36, 2016. ISSN 1533-7928. URL http://jmlr.org/papers/v17/14-335.html.

Hong Hui Tan and King Hann Lim. Review of second-order optimization techniques in artificial neural networks backpropagation. *IOP Conference Series: Materials Science and Engineering*, 495(1):012003, April 2019. ISSN 1757-899X. doi: 10.1088/1757-899X/495/1/012003. URL https://dx.doi.org/10.1088/1757-899X/495/1/012003. Publisher: IOP Publishing.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: a study of count-based exploration for deep reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 2750–2759, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-Ended Learning Leads to Generally Capable Agents, July 2021. URL http://arxiv.org/abs/2107.12808. arXiv:2107.12808 [cs].

Alexandr Ten, Pramod Kaushik, Pierre-Yves Oudeyer, and Jacqueline Gottlieb. Humans monitor learning progress in curiosity-driven exploration. *Nature Communications*, 12 (1):5972, October 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-26196-w. URL https:

//www.nature.com/articles/s41467-021-26196-w. Publisher: Nature Publishing Group.

Emmett J. Thompson, Lars Rollik, Benjamin Waked, Georgina Mills, Jasvin Kaur, Ben Geva, Rodrigo Carrasco-Davis, Tom George, Clementine Domine, William Dorrell, and Marcus Stephenson-Jones. Replay of procedural experience is independent of the hippocampus, June 2024. URL https://www.biorxiv.org/content/10.1101/2024.06.05.597547v1. Pages: 2024.06.05.597547 Section: New Results.

Bradley Efron Tibshirani, R. J. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, New York, May 1994. ISBN 978-0-429-24659-3. doi: 10.1201/9780429246593.

Edward C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948. ISSN 1939-1471. doi: 10.1037/h0061626. Place: US Publisher: American Psychological Association.

Momchil S. Tomov, Pedro A. Tsividis, Thomas Pouncy, Joshua B. Tenenbaum, and Samuel J. Gershman. The neural architecture of theory-based reinforcement learning. *Neuron*, 111(8):1331–1344.e8, April 2023. ISSN 0896-6273. doi: 10.1016/j.neuron.2023.01.023. URL https://www.cell.com/neuron/abstract/S0896-6273(23)00073-9. Publisher: Elsevier.

Amirhosein Toosi, Andrea G. Bottino, Babak Saboury, Eliot Siegel, and Arman Rahmim. A Brief History of AI: How to Prevent Another Winter (A Critical Review). *PET Clinics*, 16(4):449–469, October 2021. ISSN 1556-8598, 1879-9809. doi: 10.1016/j.cpet.2021.07.001. URL https://www.pet.theclinics.com/article/S1556-8598(21)00053-5/abstract. Publisher: Elsevier.

Mustafa Turkyilmazoglu. Convergence of the homotopy analysis method, June 2010. URL http://arxiv.org/abs/1006.4460. arXiv:1006.4460 [math-ph].

Markus Ullsperger, Lauren M. Bylsma, and Matthew M. Botvinick. The conflict adaptation

effect: It's not just priming. *Cognitive, Affective, & Behavioral Neuroscience*, 5 (4):467–472, December 2005. ISSN 1531-135X. doi: 10.3758/CABN.5.4.467. URL https://doi.org/10.3758/CABN.5.4.467.

Rolf Ulrich, Hannes Schröter, Hartmut Leuthold, and Teresa Birngruber. Automatic and controlled stimulus processing in conflict tasks: Superimposed diffusion processes and delta functions. *Cognitive Psychology*, 78:148–174, May 2015. ISSN 0010-0285. doi: 10.1016/j.cogpsych.2015.02.005. URL https://www.sciencedirect.com/science/article/pii/S0010028515000195.

Claudio Urrea and Rayko Agramonte. Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation. *Journal of Sensors*, 2021(1):9674015, 2021. ISSN 1687-7268. doi: 10.1155/2021/9674015. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/9674015. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/9674015.

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning, December 2015. URL http://arxiv.org/abs/1509.06461. arXiv:1509.06461 [cs].

Anna Vettoruzzo, Mohamed-Rafik Bouguelia, Joaquin Vanschoren, Thorsteinn Rögnvaldsson, and KC Santosh. Advances and Challenges in Meta-Learning: A Technical Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7): 4763–4779, July 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2024.3357847. URL https://ieeexplore.ieee.org/document/10413635/?arnumber=10413635. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Jennifer Vonk. Nonhuman Intelligence. In Todd K. Shackelford and Viviana A. Weekes-Shackelford, editors, *Encyclopedia of Evolutionary Psychological Science*, pages 5432–5440. Springer International Publishing, Cham, 2021. ISBN 978-3-319-

19650-3. doi: 10.1007/978-3-319-19650-3_3110. URL https://doi.org/10.1007/978-3-319-19650-3_3110.

Jane X Wang. Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences*, 38:90–95, April 2021. ISSN 2352-1546. doi: 10.1016/j.cobeha.2021.01.002. URL https://www.sciencedirect.com/science/article/pii/S2352154621000024.

Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn, January 2017. URL http://arxiv.org/abs/1611.05763. arXiv:1611.05763 [cs, stat].

Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6):860–868, June 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0147-8. URL https://www.nature.com/articles/s41593-018-0147-8. Publisher: Nature Publishing Group.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8 (3):279–292, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992698. URL https://doi.org/10.1007/BF00992698.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8. URL https:

//www.nature.com/articles/s41586-023-06415-8. Publisher: Nature Publishing Group.

Simon Nikolaus Weber and Henning Sprekeler. Learning place cells, grid cells and invariances with excitatory and inhibitory plasticity. *eLife*, 7:e34560, February 2018. ISSN 2050-084X. doi: 10.7554/eLife.34560. URL https://doi.org/10.7554/eLife.34560. Publisher: eLife Sciences Publications, Ltd.

David Wechsler. *The measurement and appraisal of adult intelligence, 4th ed.* The measurement and appraisal of adult intelligence, 4th ed. Williams & Wilkins Co, Baltimore, MD, US, 1958. doi: 10.1037/11167-000. Pages: ix, 297.

David Weinberg, Qian Wang, Thomas Ohlson Timoudas, and Carlo Fischione. A Review of Reinforcement Learning for Controlling Building Energy Systems From a Computer Science Perspective. *Sustainable Cities and Society*, 89:104351, February 2023. ISSN 2210-6707. doi: 10.1016/j.scs.2022.104351. URL https://www.sciencedirect.com/science/article/pii/S2210670722006552.

Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical Report, University of North Carolina at Chapel Hill, USA, October 1995.

Paul J. Werbos. Applications of advances in nonlinear sensitivity analysis. In R. F. Drenick and F. Kozin, editors, *System Modeling and Optimization*, pages 762–770, Berlin, Heidelberg, 1982. Springer. ISBN 978-3-540-39459-4. doi: 10.1007/BFb0006203.

Andrew Westbrook and Todd S. Braver. Cognitive effort: A neuroeconomic approach. *Cognitive, Affective, & Behavioral Neuroscience*, 15(2):395–415, June 2015. ISSN 1531-135X. doi: 10.3758/s13415-015-0334-y. URL https://doi.org/10.3758/s13415-015-0334-y.

Andrew Westbrook and Todd S. Braver. Dopamine does double duty in motivating cognitive effort. *Neuron*, 89(4):695–710, February 2016. ISSN 0896-6273. doi:

10.1016/j.neuron.2015.12.029. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4759499/.

D. J. White. Review of Bayesian Decision Problems and Markov Chains. *Journal of the Royal Statistical Society. Series A (General)*, 132(1):106–107, 1969. ISSN 0035-9238. doi: 10.2307/2343761. URL https://www.jstor.org/stable/2343761. Publisher: [Royal Statistical Society, Wiley].

Martha White and Adam White. A Greedy Approach to Adapting the Trace Parameter for Temporal Difference Learning, October 2016. URL http://arxiv.org/abs/1607.00446. arXiv:1607.00446 [cs, stat].

James C. R. Whittington and Rafal Bogacz. Theories of Error Back-Propagation in the Brain. *Trends in Cognitive Sciences*, 23(3):235–250, March 2019. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2018.12.005. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(19)30012-9. Publisher: Elsevier.

James C. R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E. J. Behrens. The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation. *Cell*, 183(5):1249–1263.e23, November 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.10.024. URL https://www.sciencedirect.com/science/article/pii/S009286742031388X.

Robert C. Wilson, Amitai Shenhav, Mark Straccia, and Jonathan D. Cohen. The Eighty Five Percent Rule for optimal learning. *Nature Communications*, 10(1):4646, November 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-12552-4. URL https://www.nature.com/articles/s41467-019-12552-4. Publisher: Nature Publishing Group.

Gabriele Wulf, Nancy McNevin, and Charles H. Shea. The automaticity of complex

motor skill learning as a function of attentional focus. *The Quarterly Journal of Experimental Psychology A: Human Experimental Psychology*, 54A(4):1143–1154, 2001. ISSN 1464-0740. doi: 10.1080/02724980143000118. Place: United Kingdom Publisher: Taylor & Francis.

Aolin Xu and Maxim Raginsky. Minimum Excess Risk in Bayesian Learning. *IEEE Transactions on Information Theory*, 68(12):7935–7955, December 2022. ISSN 1557-9654. doi: 10.1109/TIT.2022.3176056. URL https://ieeexplore.ieee.org/document/9780255. Conference Name: IEEE Transactions on Information Theory.

Yvonne Yau, Thomas Hinault, Madeline Taylor, Paul Cisek, Lesley K. Fellows, and Alain Dagher. Evidence and Urgency Related EEG Signals during Dynamic Decision-Making in Humans. *Journal of Neuroscience*, 41(26):5711–5722, June 2021. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2551-20.2021. URL https://www.jneurosci.org/content/41/26/5711. Publisher: Society for Neuroscience Section: Research Articles.

Fei Ye and Adrian G. Bors. Lifelong Teacher-Student Network Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6280–6296, October 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3092677. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Hang Yin, Yu Wang, Xukai Zhang, and Peng Li. Feedback delay impaired reinforcement learning: Principal components analysis of Reward Positivity. *Neuroscience Letters*, 685:179–184, October 2018. ISSN 0304-3940. doi: 10.1016/j.neulet.2018.08.039. URL https://www.sciencedirect.com/science/article/pii/S0304394018305846.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Asso-

ciates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e1021d43911ca2c1845910d84f40aeae-Abstract.html.

Angela J. Yu and Peter Dayan. Uncertainty, Neuromodulation, and Attention. *Neuron*, 46(4):681–692, May 2005. ISSN 0896-6273. doi: 10.1016/j.neuron.2005.04.026. URL https://www.sciencedirect.com/science/article/pii/S0896627305003624.

Angela J. Yu, Peter Dayan, and Jonathan D. Cohen. Dynamics of attentional selection under conflict: Toward a rational Bayesian account. *Journal of Experimental Psychology: Human Perception and Performance*, 35(3):700–717, 2009. ISSN 1939-1277. doi: 10.1037/a0013553. Place: US Publisher: American Psychological Association.

Anthony Zador, Sean Escola, Blake Richards, Bence Ölveczky, Yoshua Bengio, Kwabena Boahen, Matthew Botvinick, Dmitri Chklovskii, Anne Churchland, Claudia Clopath, James DiCarlo, Surya Ganguli, Jeff Hawkins, Konrad Körding, Alexei Koulakov, Yann LeCun, Timothy Lillicrap, Adam Marblestone, Bruno Olshausen, Alexandre Pouget, Cristina Savin, Terrence Sejnowski, Eero Simoncelli, Sara Solla, David Sussillo, Andreas S. Tolias, and Doris Tsao. Catalyzing next-generation Artificial Intelligence through NeuroAI. *Nature Communications*, 14(1):1597, March 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-37180-x. URL https://www.nature.com/articles/s41467-023-37180-x. Publisher: Nature Publishing Group.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence, June 2017. URL http://arxiv.org/abs/1703.04200. arXiv:1703.04200 [cs, q-bio, stat].

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience Replay Optimization. pages 4243–4249, 2019. URL https://www.ijcai.org/proceedings/2019/589.

Ji Zhang, Jingkuan Song, Lianli Gao, Ye Liu, and Heng Tao Shen. Progressive Meta-Learning With Curriculum. *IEEE Transactions on Circuits and Systems for Video*

Technology, 32(9):5916–5930, September 2022. ISSN 1558-2205. doi: 10.1109/TCSVT.2022.3164190. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

Yuxiao Zhang, Yan Chen, Yushi Xin, Beibei Peng, and Shuai Liu. Norepinephrine system at the interface of attention and reward. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 125:110751, July 2023. ISSN 0278-5846. doi: 10.1016/j.pnpbp.2023.110751. URL https://www.sciencedirect.com/science/article/pii/S0278584623000374.

Xinlei Zhou, Han Liu, Farhad Pourpanah, Tieyong Zeng, and Xizhao Wang. A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications. *Neurocomputing*, 489:449–465, June 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.10.119. URL https://www.sciencedirect.com/science/article/pii/S0925231221019068.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning, February 2020. URL http://arxiv.org/abs/1910.08348. arXiv:1910.08348 [cs].

Nicolas Zucchet and João Sacramento. Beyond backpropagation: implicit gradients for bilevel optimization, May 2022. URL http://arxiv.org/abs/2205.03076. arXiv:2205.03076 [cs, math].

Nicolas Zucchet, Simon Schug, Johannes von Oswald, Dominic Zhao, and João Sacramento. A contrastive rule for meta-learning, October 2022. URL http://arxiv.org/abs/2104.01677. arXiv:2104.01677.

Appendix A

Meta-Learning Strategies through Value Maximization in Neural Networks

A.1 Further related work

In recent years, several meta-learning algorithms have been proposed to solve a variety of meta-learning tasks, such as fast-adaptation (Finn et al., 2017; Nichol et al., 2018), continual-learning (Parisi et al., 2019), and multi-tasking (Crawshaw, 2020; Sagiv et al., 2020; Ravi et al., 2021; Musslick et al., 2020). Because these tasks have different goals, the specific design of the meta-learning algorithm used to solve each task differs.

One popular application for meta-learning algorithms is reinforcement learning (RL). RL agents use a policy to choose actions to maximize the expected return. The policy is usually based on a value function (or action-value), linking particular actions to values, that agents learn to estimate through experience (Mnih et al., 2015; Wang et al., 2017). For the agent to act optimally, the policy requires a good estimation of the value function. For single tasks, this is typically not hard, but agents struggle when they must solve more than one task. To aid this difficulty, researchers implement meta-learning strategies, such as

enhancing exploration (Gupta et al., 2018; Liu et al., 2021), re-using experiences through a memory buffer (Ritter et al., 2018), exposing the agent to a large number of tasks from a task distribution (Wang et al., 2017; Team et al., 2021), and choosing the order of those tasks carefully (Stergiadis et al., 2021; Zhang et al., 2022) with the hope that the agent performs well on each of these tasks in the task distribution. Indeed, several techniques improve the performance of reinforcement learning agents (Hessel et al., 2017; Obando-Ceron and Castro, 2021; Kanervisto et al., 2021), but there are two main issues with these approaches. First, these techniques are usually designed manually and specifically for the tasks at hand. Second, how the learning dynamics depend on these techniques remains unclear. The combined effects of the agent-environment interaction dynamics and the value estimation during training make analyzing learning dynamics on these models remarkably challenging. A technique that could autonomously generate a meta-learning strategy *a priori* by leveraging analysis of the learning dynamics would address these issues and potentially improve the understanding, performance, and flexibility of these types of models.

Learning dynamics has been widely studied in the context of neural network training, where the goal in these cases is to minimize a loss function. In particular, deep linear networks (Saxe et al., 2019; Zenke et al., 2017; Li and Sompolinsky, 2021; Braun et al., 2022), gated linear networks (Saxe et al., 2022; Li and Sompolinsky, 2022), have been useful to analyze learning dynamics, due to their mathematical tractability and still complex (non-linear) learning dynamics. Having access to the learning dynamics allows us to test the learning system under different conditions (tasks, architectures, hyperparameters, etc) and draw conclusions, either from mathematical analysis or simulations on how these conditions affect learning during the training period. Further techniques to describe learning dynamics exist, which have their drawbacks in terms of mathematical and computational tractability, or required limits in input or hidden dimensionality to obtain closed-form differential equations describing the dynamics. Some of these frameworks

are the teacher-student settings (Goldt et al., 2019; Ye and Bors, 2022), and mean-field theory for neural networks (Mignacco et al., 2020; Bordelon and Pehlevan, 2022), recently applied to temporal difference learning (Bordelon et al., 2023). These methods present a promising direction to extend our framework to non-linear networks and reinforcement learning dynamics.

In this work, we propose a new framework called *learning effort*, where we combine the goal of maximizing value, with neural network learning dynamics, by making the regression loss during training proportional to the reward of the learning system (in this case, a neural network). This has several advantages. First, in practice, this choice makes the problem of estimating value equivalent to estimating the learning dynamics. Because the loss function during training can be obtained from fully solving the learning dynamics, then the reward throughout training is also solved (similar to Zenke et al. 2017). Second, taking advantage of the partial tractability of linear networks, and approximating non-linear network dynamical equations, we are able to draw conclusions on how some parameters interact with the learning dynamics when maximizing the value. Using this framework, we define a control signal, that at a cost, can modify the learning dynamics to maximize value during training. Furthermore, any network parameter that is not subject to the learning dynamics could be chosen as this control signal to maximize value. Previous work has addressed these questions by assuming a learning trajectory, where the functional form is fixed, hence not considering possible changes in the trajectory due to the time-varying control (Son and Sethi, 2006, 2010). Other work along these lines improves this by training complex learning systems and learning the value function to estimate the optimal control signal (Musslick et al.; Lieder et al., 2018). Some meta-learning algorithms such as MAML (Finn et al., 2017) and bilevel optimization methods (Franceschi et al., 2018; Andreas et al., 2017) can be encapsulated under our framework as explained in Section 3.3 in the main manuscript.

We used this framework to investigate different kinds of intervention of the control signal

in the learning process, finding what are optimal strategies when facing a meta-learning task, including when the use of the control signal is costly. Using this setting, we can ask questions such as how to optimally allocate control to speed up learning while minimizing the use of it, how to train the network to switch tasks quickly, or what is the optimal level of attention to a set of tasks, or even if it is worth to learn a specific task given the cost of engaging on learning it, all by maximizing value during learning. Related work has used similar bi-level optimization, but with different loss functions on each optimization level (Zucchet and Sacramento, 2022). We provide the implementation for the work presented here in `https://anonymous.4open.science/r/neuromod-6A3C`.

We further suggest that this same framework could be useful to analyze phenomena in cognitive neuroscience, and that there is a correspondence of our *learning effort* framework and the Expected Value of Control Theory (Shenhav et al., 2013; Musslick et al., 2020; Masís et al., 2021).

A.2 Two-Layer linear network dynamics

Taking $\hat{Y} = W_2 W_1 X$, where $X \in \mathbb{R}^I$, $\hat{Y} \in \mathbb{R}^O$, $W_1(t) \in \mathbb{R}^{H \times I}$ and $W_2(t) \in \mathbb{R}^{O \times H}$ are the first and second layer weights (dropping time dependency of the weights to simplify notation), the loss function is

$$\begin{aligned}
\mathcal{L} =& \frac{1}{2}\|Y - \hat{Y}\|^2 + \frac{\lambda}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right), && \text{(A.1)}\\
=& \frac{1}{2}\operatorname{Tr}\left(\left(Y - \hat{Y}\right)\left(Y - \hat{Y}\right)^T\right) + \frac{\lambda}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right), && \text{(A.2)}\\
=& \frac{1}{2}\left[\operatorname{Tr}\left(YY^T\right) - 2\operatorname{Tr}\left(YX^T W_1^T W_2^T\right) + \operatorname{Tr}\left(W_2 W_1 X X^T W_1^T W_2^T\right)\right] + \frac{\lambda}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right). && \text{(A.3)}
\end{aligned}$$

Taking the derivative of $\mathcal{L}$ with respect to the weights W_2 and W_1, in general $\frac{\partial}{\partial W}\operatorname{Tr}\left(AW^T B\right) =$

BA and $\frac{\partial}{\partial W}\operatorname{Tr}\left(AWBW^TC\right) = A^TC^TWB^T + CAWB$, then we get

$$\frac{\partial \mathcal{L}}{\partial W_1} = -W_2^T Y X^T + W_2^T W_2 W_1 X X^T + \lambda W_1, \tag{A.4}$$

$$\frac{\partial \mathcal{L}}{\partial W_2} = -Y X^T W_1^T + W_2 W_1 X X^T W_1^T + \lambda W_2. \tag{A.5}$$

Updating the weights using gradients steps (layers $i = \{1, 2\}$, gradient step iteration index k, and sample index b from a batch with size B) gives

$$W_i(t_{k+1}) = W_i(t_{k+1}) - \alpha \frac{1}{B} \sum_{b=1}^{B} \frac{\partial \mathcal{L}(Y_b, X_b)}{\partial W_i}. \tag{A.6}$$

Taking the gradient flow limit $\alpha \to 0$, number of samples given to the model per unit of time goes to infinity, converting the average to an expectation over samples (Saxe et al., 2019; Elkabetz and Cohen, 2021),

$$\tau_w \frac{dW_i}{dt} = -\left\langle \frac{\partial \mathcal{L}}{\partial W_i} \right\rangle, \tag{A.7}$$

obtaining

$$\tau_w \frac{dW_1}{dt} = W_2^T \left(\Sigma_{xy}^T - W_2 W_1 \Sigma_x\right) - \lambda W_1, \tag{A.8}$$

$$\tau_w \frac{dW_2}{dt} = \left(\Sigma_{xy}^T - W_2 W_1 \Sigma_x\right) W_1^T - \lambda W_2. \tag{A.9}$$

Note that, both equations, the input-output mapping ant the learning dynamics are valid simultaneously, then describing the learning system as forward and backward happening at the same time.

A.3 Gain modulation

In this model, all weights are multiplied by a gain term, which we will optimize to maximize the expected return. The input-output equation of the neural network with gain modulation is $\hat{Y} = \left(W_2 \circ \tilde{G}_2\right)\left(W_1 \circ \tilde{G}_1\right) X = \tilde{W}_2\tilde{W}_1 X$ and $\tilde{G}_i = (\mathbb{1}_i + G_i)$, where $\mathbb{1}$ is a matrix of ones, and $\mathbb{1}_i$, G_i having the same shape as W_i (again dropping time dependence for weights and gain for notation simplicity), then the loss is

$$\mathcal{L} = \frac{1}{2}\left[\mathrm{Tr}\left(YY^T\right) - 2\,\mathrm{Tr}\left(YX^T\tilde{W}_1^T\tilde{W}_2^T\right) + \mathrm{Tr}\left(\tilde{W}_2\tilde{W}_1XX^T\tilde{W}_1^T\tilde{W}_2^T\right)\right] + \frac{1}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right). \tag{A.10}$$

In general

$$\frac{\partial}{\partial W}\mathrm{Tr}\left(A\tilde{W}^TB\right) = (BA)\circ G \tag{A.11}$$

$$\frac{\partial}{\partial W}\mathrm{Tr}\left(A\tilde{W}B\tilde{W}^TC\right) = \left(A^TC^T\tilde{W}B + CA\tilde{W}B^T\right)\circ G. \tag{A.12}$$

Following the same procedure as in section A.2, we can derive the learning dynamics equations for the weights when using gain modulation:

$$\begin{aligned}\tau_w\frac{dW_1}{dt} &= \left(\tilde{W}_2^T\Sigma_{xy}^T\right)\circ\tilde{G}_1 - \left(\tilde{W}_2^T\tilde{W}_2\tilde{W}_1\Sigma_x\right)\circ\tilde{G}_1 - \lambda W_1,\\ \tau_w\frac{dW_2}{dt} &= \left(\Sigma_{xy}^T\tilde{W}_1^T\right)\circ\tilde{G}_2 - \left(\tilde{W}_2\tilde{W}_1\Sigma_x\tilde{W}_1^T\right)\circ\tilde{G}_2 - \lambda W_2.\end{aligned} \tag{A.13}$$

The G_2 and G_1 that optimize value can be computed iterating

$$G_i^{k+1}(t_i) = G_i^k(t_i) + \alpha_g\frac{dV}{dG_i(t_i)} \tag{A.14}$$

using algorithm 1.

A.3.1 Control basis

For a set of weights $W_i() \in \mathbb{R}^{m\times n}$, then $G_i(t) \in \mathbb{R}^{m\times n}$ such that $\tilde{W}_i(t) = W_i(t) \circ (\mathbb{1} + G_i(t))$, we can write the control signal as a projection on a basis

$$G_i(t) = \sum_b \nu_i^b(t) G_i^b. \tag{A.15}$$

Note that the time dependency of $G_i(t)$ comes from ν_i^b which is the variable it is optimized, instead of $G_i(t)$ directly.

Neuron Basis: Take b indexing a row (or column) of a matrix $\in \mathbb{R}^{m\times n}$, then G_i^b has row (or column) as 1s, and 0 everywhere else. For example, if b indexes the rows of G_i^b, then

$$G_i^b = \text{row } b \rightarrow \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{A.16}$$

This is called neuron basis since $\nu_i^b(t)$ will end up multiplying all of the weights connecting a specific neuron b. For example, in the previous G_i^b where the rows are 1s, using this gain modulation on the second layer W_2, means that $\nu_i^b(t)$ will multiply the output b of the layer. Changing this by columns means modulation of the input weights per each hidden unit. In this case, the iterations on the control signal to maximize the expected return in equation 3.3, following algorithm 1 is

$$\nu_i^{k+1}(t_i) = \nu_i^k(t_i) + \alpha_g \frac{dV}{d\nu_i(t_i)}. \tag{A.17}$$

This procedure was used in the category assimilation task on MNIST and the Semantic Datasetm but with the specific restriction of using a *neuron basis* for $G_2(t)$ as in equation A.15, while keeping $G_1(t) = 0$ (no gain modulation on the first layer). The *neuron basis* on the output neurons allows the control signal to change the gain on specific output neurons, therefore changing the gain of the learning signal from a specific category in a classification task. See the results of this specific model in Appendix A.9.8.

A.4 Dataset Engagement modulation

In the engagement modulation, the auxiliary loss used to derive the learning dynamics equation, in this case, is given by

$$\mathcal{L}_{\text{aux}} = \sum_{\tau=1}^{N_\tau} \psi_\tau(t) \mathcal{L}(\hat{Y}_\tau, Y_\tau) + \frac{\lambda}{2} \left(\|W_2\|_F^2 + \|W_1\|_F^2 \right), \tag{A.18}$$

where N_τ is the number of available datasets, $\psi_\tau(t)$ are the engagement coefficients for dataset τ, $\hat{Y}_\tau$ and Y_τ are the predictions and target for dataset τ. The network can simultaneously try to solve all of the dataset at the same time since the inputs and outputs per task are concatenated as $X^T = [X_1^T, ..., X_\tau^T, ..., X_{N_\tau}^T] \in \mathbb{R}^I$ and $Y^T = [Y_1^T, ..., Y_\tau^T, ..., Y_{N_\tau}^T] \in \mathbb{R}^O$. Then, taking the gradient with respect to the weights, and taking the gradient flow limit we obtain equation 3.22. In this equation, $\Sigma_x = \left\langle X X^T \right\rangle$ can be expressed as the statistics of each tasks following

$$\Sigma_x = \begin{bmatrix} \Sigma_1 & \dots & \langle X_1 \rangle \langle X_\tau \rangle^T & \dots & \langle X_1 \rangle \langle X_{N_\tau} \rangle^T \\ \vdots & \ddots & & & \\ \langle X_\tau \rangle \langle X_1 \rangle^T & & \Sigma_\tau & & \\ \vdots & & & \ddots & \\ \langle X_{N_\tau} \rangle \langle X_1 \rangle^T & & & & \Sigma_{N_\tau} \end{bmatrix}, \tag{A.19}$$

with $\Sigma_\tau = \left\langle X_\tau X_\tau^T \right\rangle$. Since the target Y_τ is only correlated to the input X_τ, the input-output correlation matrix $\Sigma_{xy\tau} \in \mathbb{R}^{I\times O}$ is

$$\Sigma_{xy\tau} = \underbrace{\begin{bmatrix} 0 & \dots 0 \dots & \langle X_1 \rangle \langle Y_\tau \rangle^T & \dots 0 \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots 0 \dots & \langle X_\tau Y_\tau \rangle^T & \dots 0 \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots 0 \dots & \langle X_{N_\tau} \rangle \langle Y_\tau \rangle^T & \dots 0 \dots & 0 \end{bmatrix}}_{\text{output size O}} . \tag{A.20}$$

The rows of $W_{2\tau} \in \mathbb{R}^{O\times H}$ are replaced with zeros for outputs not contributing to $\hat{Y}_\tau$. From here, the $\psi_\tau(t)$ that optimize value can be computed iterating

$$\psi_\tau^{k+1}(t_i) = \psi_\tau^k(t_i) + \alpha_g \frac{dV}{d\psi_\tau(t_i)} \tag{A.21}$$

using algorithm 1.

A.5 Category engagement modulation

This derivation is similar to the engagement modulation, but the engagement coefficient scales the error coming from each of the categories from a classification problem. The loss function is just the mean square error between the labels and the output of the network. The auxiliary loss used to derive the learning dynamics equations can be written as

$$\mathcal{L}_{\text{aux}} = \frac{1}{2}\sum_{c=1}^{C} \left[\phi_c(y_c - \hat{y}_c)\right]^2 + \frac{\lambda}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right), \tag{A.22}$$

$$= \frac{1}{2}\left[\mathrm{d}\left(\phi\right)\left(Y - \hat{Y}\right)\right]^2 + \frac{\lambda}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right) \tag{A.23}$$

with $\mathrm{d}(\phi) = \mathrm{diag}(\phi)$ and $\phi = [\phi_1, ..., \phi_c, ..., \phi_C]^T$. Then, deriving the learning dynamics equation by learning the weights using backpropagation (as in Appendix 3.5) we obtain

$$\begin{aligned}\tau_w \frac{dW_1}{dt} &= W_2^T \mathrm{d}(\phi)^2 \Sigma_{xy}^T - W_2^T \mathrm{d}(\phi)^2 W_2 W_1 \Sigma_x - \lambda W_1, \\ \tau_w \frac{dW_2}{dt} &= \mathrm{d}(\phi)^2 \Sigma_{xy}^T W_1^T - \mathrm{d}(\phi)^2 W_2 W_1 \Sigma_x W_1^T - \lambda W_1. \end{aligned} \tag{A.24}$$

The reason for this slight variation compared to the task engagement model, is because for the category engage case, we assume we do not have access to $\left\langle X_c X_c^T \right\rangle$ or $\left\langle X_c Y_c^T \right\rangle$ which are class-specific quantities of the dataset. From here, the $\phi_c(t)$ that optimize value can be computed iterating

$$\phi_c^{k+1}(t_i) = \phi_c^k(t_i) + \alpha_g \frac{dV}{d\phi_c(t_i)} \tag{A.25}$$

using algorithm 1.

Class proportion experiment: We trained a neural network (same architecture as in this section), and modifying the proportion of classes through time using the category engagement inferred from the optimization. The number of elements per class $b_c(t_i)$ in a batch of size B used in this experiment is

$$b_c(t_i) = \frac{\phi_c(t_i) B}{C} \tag{A.26}$$

with i indexing the SGD iteration on training the weights W_1 and W_2.

A.6 Non-linear Two-layer Network

As a first approach to applying this same learning effort framework in non-linear networks, we approximated the dynamics by Taylor expanding the non-linearities around the mean to get equations depending on first and second moment of the data distribution. Consider

a neural network of the form $\hat{Y} = W_2 f(W_1 X)$ where f is a non-linear function ($\tanh(\cdot)$ used in simulations). Following the same setting and procedure as in Appendix section A.2, the loss function can be written as

$$\mathcal{L} = \frac{1}{2}\left[\mathrm{Tr}\left(YY^T\right) - 2\,\mathrm{Tr}\left(Y^T W_2 f(W_1 X)\right) + \mathrm{Tr}\left(f(W_1 X)^T W_2^T W_2 f(W_1 X)\right)\right] \tag{A.27}$$

$$+ \frac{1}{2}\left(\|W_2\|_F^2 + \|W_1\|_F^2\right). \tag{A.28}$$

Taking the derivative of $\mathcal{L}$ with respect to W_i, updating by gradient descend and taking the gradient flow limit as in equations A.6 and A.7, we obtain

$$\tau_w \frac{dW_2}{dt} = \left\langle Y f(W_1 X)^T - W_2 f(W_1 X) f(W_1 X)^T \right\rangle_{XY} - \lambda W_2, \tag{A.29}$$

$$\tau_w \frac{dW_1}{dt} = \left\langle \mathrm{diag}(f') W_2^T Y X^T - \left(W_2 \mathrm{diag}(f')\right)^T \left(W_2 f(W_1 X)\right) X^T \right\rangle_{XY} - \lambda W_1, \tag{A.30}$$

with $f' = f'(W_1 X)$ is the element-wise application of the non-linear function derivative on $W_1 X$, and $\mathrm{diag}(f')$ a diagonal matrix with f' in the diagonal. Now we take the Taylor expansion of f around the mean of the data distribution,

$$f\left(W_1 X\right) \approx f\left(W_1 \left\langle X \right\rangle\right) + J(W_1 \left\langle X \right\rangle)(X - \left\langle X \right\rangle) \;\text{ with }\; J = W_1 \mathrm{diag}(f'(W_1 \left\langle X \right\rangle)). \tag{A.31}$$

Replacing this expansion in equations A.29 and A.30, using $f' = f'(W_1 \left\langle X \right\rangle)$ then taking the expectation $\left\langle \cdot \right\rangle_{XY}$ we obtain

$$\begin{aligned}
\tau_w \frac{dW_2}{dt} \approx & \left\langle Y \right\rangle f\left(W_1 X\right)^T + \left[\Sigma_{xy}^T + \left\langle Y \right\rangle \left\langle X \right\rangle^J\right] J^T \\
& - W_2 \left[f\left(W_1 X\right) f\left(W_1 X\right)^T + J \Sigma_x J^T + J \left\langle X \right\rangle \left\langle X \right\rangle^T J^T\right] - \lambda W_2 \\
\tau_w \frac{dW_1}{dt} \approx & \mathrm{diag}(f') \left[W_2^T \Sigma_{xy}^T - W_2^T W_2 f\left(W_1 X\right) X^T W_2^T W_2 J \Sigma_x W_2^T W_2 J \left\langle X \right\rangle \left\langle X \right\rangle^T\right] - \lambda W_1.
\end{aligned} \tag{A.32}$$

In case of using gain modulation, $\hat{Y} = \tilde{W}_2 f\left(\tilde{W}_1 X\right)$ as in Section A.3. Executing same steps as for the case without control (see Appendix A.3), the approximated learning dynamics for the gain modulation case is

$$\begin{aligned}
\tau_w \frac{dW_2}{dt} &\approx \left(\langle Y\rangle f\left(\tilde{W}_1 X\right)^T + \left[\Sigma_{xy}^T + \langle Y\rangle \langle X\rangle^J\right] J^T\right) \circ \tilde{G}_2 \\
&\quad - \left(\tilde{W}_2\left[f\left(\tilde{W}_1 X\right) f\left(\tilde{W}_1 X\right)^T + J\Sigma_x J^T + J\langle X\rangle \langle X\rangle^T J^T\right]\right) \circ \tilde{G}_2 - \lambda W_2, \\
\tau_w \frac{dW_1}{dt} &\approx \left(\mathrm{diag}(f')\left[\tilde{W}_2^T \Sigma_{xy}^T - \tilde{W}_2^T \tilde{W}_2 f\left(\tilde{W}_1 X\right) X^T \tilde{W}_2^T \tilde{W}_2 J\Sigma_x \tilde{W}_2^T \tilde{W}_2 J\langle X\rangle \langle X\rangle^T\right]\right) \circ \tilde{G}_1 - \lambda W_1,
\end{aligned} \tag{A.33}$$

where $f' = f'(\tilde{W}_1 \langle X\rangle)$ and $J = \tilde{W}_1 \mathrm{diag}(f'(\tilde{W}_1 \langle X\rangle))$. The G_2 and G_1 that optimize value can be computed iterating

$$G_i^{k+1}(t_i) = G_i^k(t_i) + \alpha_g \frac{dV}{dG_i(t_i)} \tag{A.34}$$

using algorithm 1. The obtained $G_i(t)$ from this optimization process are plugged into $\hat{Y} = \tilde{W}_2 f\left(\tilde{W}_1 X\right)$, then trained using SGD to check how much of improvement we get using the computed control signal inferred using the approximated dynamics. The results for this model are depicted in Figure A.14.

A.7 Closed form Accuracy and Gradient

Here, the accuracy and gradient of the epistemic noise control model are derived in closed form. Consider the decision variable $\hat{y} = \mathrm{sign}(w(t)x + \eta)$, given that $P(y = 1) = P(y = -1) = 1/2$, $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $\eta \sim \mathcal{N}(0, \sigma_0^2)$, the probability of the model answering

correctly is given by

$$a(t) = P(y(xw + \frac{\eta}{(1+g(t)^2)}) > 0), \tag{A.35}$$

$$= P((xw + \frac{\eta}{(1+g(t)^2}))) > 0|y=1)P(y=1) \tag{A.36}$$

$$+ P((xw + \frac{\eta}{(1+g(t)^2)}) < 0|y=-1)P(y=-1), \tag{A.37}$$

$$= P((xw + \frac{\eta}{(1+g(t)^2)}) > 0|y=1), \tag{A.38}$$

$$= \frac{1}{2}\left[1 + \text{erf}\left(\frac{\mu_x w(t)}{\sqrt{2}\sqrt{w^2(t)\sigma_x^2 + \frac{\sigma_0^2}{(1+g)^2}}}\right)\right], \tag{A.39}$$

where the error function comes from a truncated integral of $xw+\eta \sim \mathcal{N}(\mu_x w(t), w^2(t)\sigma_x^2 + \sigma_0^2/(1+g)^2$. The gradient of the accuracy with respect to the weight w gives

$$\frac{da(t)}{dw} = \frac{1}{\sqrt{\pi}} e^{-\frac{\text{SNR}}{2}} \cdot \frac{d}{dw}\left[\frac{\mu_x w(t)}{\sqrt{2}\sqrt{w^2(t)\sigma_x^2 + \frac{\sigma_0^2}{(1+g)^2}}}\right] \tag{A.40}$$

$$= \frac{1}{\sqrt{\pi}} e^{-\frac{\text{SNR}}{2}} \left[\frac{\mu_x}{\sqrt{2}\sqrt{w^2(t)\sigma_x^2 + \frac{\sigma_0^2}{(1+g)^2}}} - \frac{\mu_x \sigma_x^2 w^2(t)}{\sqrt{2}\left(\sqrt{w^2(t)\sigma_x^2 + \sigma_0^2/(1+g)^2}\right)^3}\right]. \tag{A.41}$$

From here, it is easy to show that the difference on the right-hand side is larger than zero for $\sigma_0^2 > 0$, otherwise, the gradient is zero, therefore, there is no need for learning.

A.8 Dataset Details

Correlated gaussians: Toy dataset with correlated gaussian inputs. We sample y_1 as ± 1 with probability 1/2. Then sample $y_2 = y_1(1-2\xi)$ with $\xi \sim \text{Ber}(p)$, if $p = 1/2$ then the labels are independent. We generate the input $x_i \sim \mathcal{N}(y_i\mu_i, \sigma_i^2)$, then taking $X = [x_1, x_2]^T$ and $Y = [y_1, y_2]^T$. From this data distribution process, we can analytically

compute $\Sigma_x = \langle XX^T \rangle$, $\Sigma_{xy} = \langle XY^T \rangle$ as $\Sigma_y = \langle YY^T \rangle$

$$\Sigma_x = \begin{bmatrix} \mu_1^2 + \sigma_1^2 & \mu_1\mu_2(1-2p) \\ \mu_1\mu_2(1-2p) & \mu_1^2 + \sigma_1^2 \end{bmatrix}, \quad \Sigma_{xy} = \begin{bmatrix} \mu_1 & \mu_1(1-2p) \\ \mu_2(1-2p) & \mu_2 \end{bmatrix} \text{ and } \Sigma_y = \begin{bmatrix} 1 & 1-2p \\ 1-2p & 1 \end{bmatrix}. \tag{A.42}$$

$\langle X \rangle = 0$ and $\langle Y \rangle = 0$.

Hierarchical concepts: This is a semantic learning dataset used in (Saxe et al., 2019; Braun et al., 2022) to study learning dynamics when learning a hierarchy of concepts. This task allows linear neural networks to present *rich* learning (opposite to *lazy* learning, see (Chizat et al., 2020; Flesch et al., 2022)) when using a small weight initialization. The *rich* regime shows a step-like learning dynamics where each step represents a learning of a different hierarchy level. In this dataset, Σ_x and Σ_{xy} have a close form, for example, for a hierarchy of 3 levels,

$$\Sigma_x = I_4, \quad \Sigma_{xy} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \Sigma_y = \Sigma_{xy}^T \Sigma_{xy} \tag{A.43}$$

where I_n is the identity matrix of size n. $\langle X \rangle_j = \frac{1}{I} \sum_{j=1}^{I} (\Sigma_x)_{ji}$ and $\langle Y \rangle_j = \frac{1}{I} \sum_{j=1}^{I} (\Sigma_{xy})_{ji}$. Samples from this dataset are simply X as a random column from the identity matrix and Y as the corresponding column from Σ_{xy}.

MNIST: This is a classification task of hand-written digits (Deng, 2012). Images from this dataset were reduced to 5×5 and flattened. We estimated the correlation matrices,

$\hat{\Sigma}_x$, $\hat{\Sigma}_{xy}$ and $\hat{\Sigma}_y$ by taking the expectation over the samples in the training set.

A.9 Additional results

Here we present extra Figures and discussion to some of the results from the main text.

A.9.1 Single Neuron Model

We did several runs varying some hyperparameters of the system to see the effect on the optimal control signal and the improvement in the instant reward rate $v(t)$.

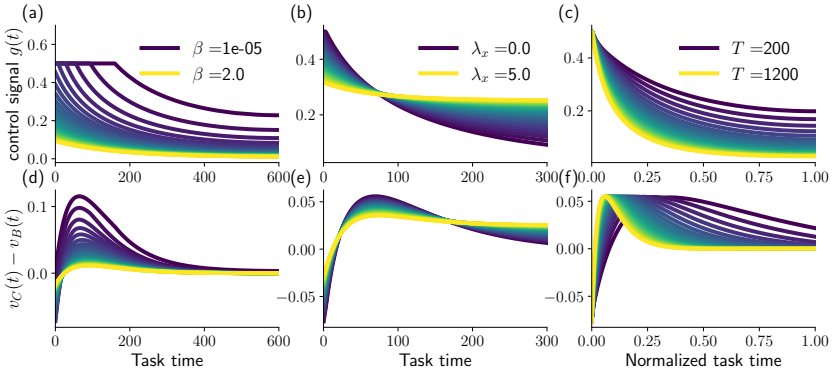

Figure A.1: Results of hyperparameter variations for the single neuron case. **(a) and (d)**: optimal control signal $g(t)$ and difference between instant net reward using control $v_C(t)$ and the baseline (no control) net reward $v_B(t)$, for the variations of the cost coefficient β. Panels **(b) and (e)** and **(c) and (f)** are same results but varying regularization coefficient λ and available time to learn T.

A.9.2 Extended results for Meta-Learning

Our optimization framework is related to other meta-learning algorithms in the machine learning literature. Here we provide a formal description of the relation between our framwork and two well-established meta-learning algorithms, *Model-Agnostic Meta-learning for Fast Adaptation of Deep Networks* (MAML, Finn et al. 2017) and *Bilevel Programming*

for Hyperparameter Optimization and Meta-Learning (Franceschi et al., 2018), as well as simulations details of the results shown in Figure 3.4 in Section 3.3.

We highlight that there are scalable meta-learning algorithm methods in the literature (Rajeswaran et al., 2019; Deleu et al., 2022) which are able to meta-learn variables in state-of-the-art architectures. However, these methods rely on simplifying the meta objective, restricting the meta-variables (making them smaller for tractability), or using extra iterative processes to approximate gradients. This is fundamentally different from what we are able to achieve in our framework. In most experiments of our paper, each step in our outer loop considers the entire inner loop learning trajectory and computes the gradient of the meta-loss at all time steps to capture the effect of the meta-parameters across the entire learning trajectory, not just the last step as in the referenced work. In addition, our meta-parameters can be as complex as the (size of the network) times (inner loop training iterations), in other words, our meta-variables also depend on time, increasing the complexity of the optimization problem. We are solving the full complex meta-learning problem (which is the desired target in both references), by considering a simpler model, instead of approximating our computation. This will provide insight on the *ideal* meta-objective in complex non-linear learning dynamics which is intractable in large state-of-the-art learning architectures.

A.9.3 Model Agnostic Meta-Learning

Simulation results are shown in Figure A.2 and A.3 for the MAML equivalent with the learning effort framework presented in subsection 3.3.1, and works as follows: We created a set of tasks $\mathcal{T}_i$ with pairs of MNIST numbers, (0, 1), (7, 1), (8, 9), (3, 8) and (5, 3) (see App. A.8). Then we picked a range of time-steps to consider during the optimization of the initial weights g (control signal) made with a from a `linspace` starting from 1 to 340 (including) every 20 steps (except between 1 to 20 where there is a difference of 19), we call these *Optimized steps* as in Figure A.2 and A.3. We evaluate how good is the loss

dynamics starting from the optimized initial conditions g throughout 8000 updates steps on each task as shown in Figure A.2 and A.3 as *eval steps*. We obtain the cumulative loss eval steps in Figure A.3 by integrating these dynamics throughout the evaluation time after optimizing initial conditions.

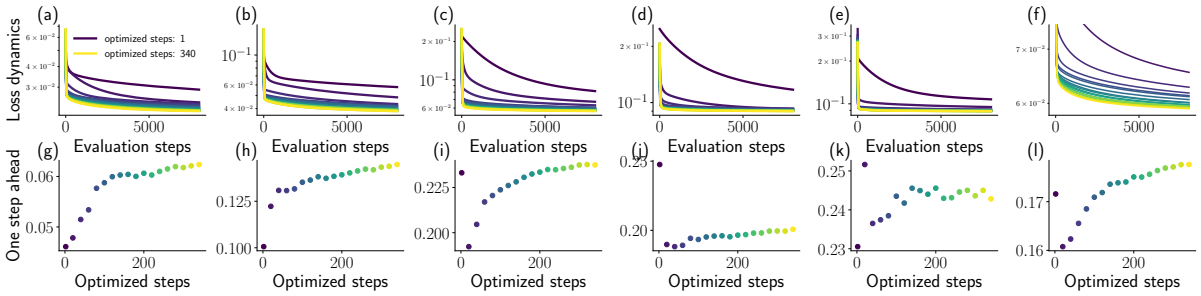

Figure A.2: Simulating multi-step MAML with the *learning effort* framework: The first row is the loss dynamics evaluated through 8000 updates steps, starting from the optimized parameters found by MAML, the color code shows the number of optimized steps considered in the meta-objective (1 step is standard MAML). The second row shows the one step ahead when considering different number of optimized steps (same color code), in other words, it is the first loss value from the curves in the first row. From **(a)** to **(e)** columns, the results for the binary regression tasks for MNIST pairs (0, 1), (7, 1), (8, 9), (3, 8) and (5, 3) is shown respectively, with the last column **(f)** being the average across tasks $\mathcal{L}_{MAML}$.

In Figure A.2, it is shown that considering more time-steps in the optimization (*Optimized steps*) is beneficial for the resulting dynamics. The more steps are considered, the faster the learning after optimizing for initial parameters. Something important to note is that there is a qualitative difference in the optimization when considering 1 steps, vs considering multiple steps. As mentioned in section 3.3, considering only one step ahead is a myopic optimization initial condition. Immediately after considering a few steps ahead, the loss dynamics in the evaluation steps is improve very quickly (Figure A.3), and the one step ahead loss (the one considered when optimizing only one step) is reduced even more after considering a few more optimized steps, as shown in Figure A.2l, presumably because the gradient over the initial parameters can see the dynamics after one step ahead, perhaps finding better solutions through looking at more steps ahead. Then, there is a

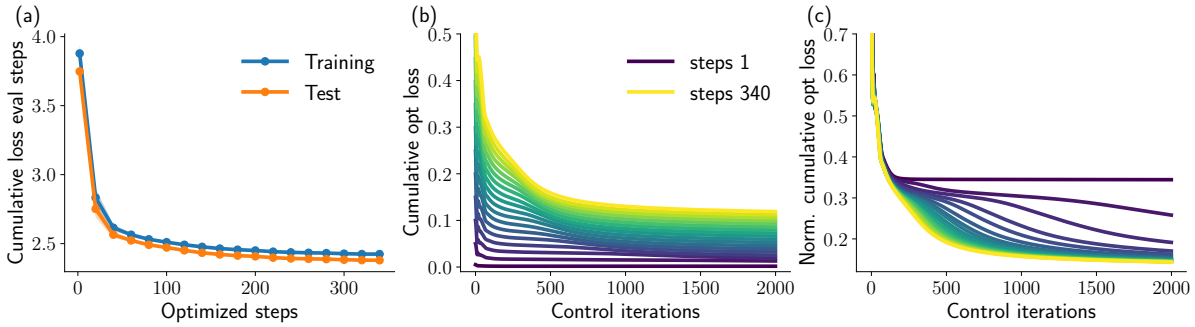

Figure A.3: Optimization results on Multi-step MAML. **(a)**: Cumulative loss eval steps (integral of curves in Figure A.2) evaluated on training and test sets in MNIST pairs. **(b)**: Actual loss considered during the optimization vs control iterations (finding the best initial conditions $W_1(0)$ and $W_2(0)$). **(c)**: Same as **(b)** but normalized by its maximum for visualization purposes.

transition, where after considering around 180 steps ahead (Figure A.2l), the one-step ahead loss increases while improving the loss dynamics in the evaluation steps even more, sacrificing immediate loss for a better overall cumulative dynamics. This hypothesis is supported by looking at Figure A.3c where optimizing one step ahead converges in a few iterations over the initial parameters g. In contrast, when considering more steps, the optimization goes beyond this plateau and the speed at which this plateau is skipped increases with the number of optimized steps considered. The myopic solution is not able to optimize further since it doesn't have information of the learning dynamics, which is provided when considering multiple steps as shown by these results. In Table A.2 we summarize the parameters used for the simulation.

A.9.4 Bilevel Programming

In Bilevel Programming (Franceschi et al., 2018) which unifies gradient-based hyper-parameter optimization and meta-learning algorithms. They show an approximated bilevel programming method and the conditions to guarantee convergence to the exact problem. The setting described in this work show meta-learning problems as an inner and

outer objective L_λ and $E(w, \lambda)$ respectively, where w are the parameters of the model and λ are the hyperparameters (equations 3 and 4 in (Franceschi et al., 2018)), and the approximated problem corresponds to doing a few update steps only in the inner objective, meaning the loss of the inner loop is not fully minimized. The first difference between our setting and Bilevel Programming is the method they use to optimize the meta-parameters called reverse-hypergradient method (Franceschi et al., 2017). Another difference, is that we extend the bilevel optimization setting to have a normative meaning through the inclusion of a control cost, discount factor and we further simplify it by using the average learning dynamics obtained from using the gradient flow limit in the two-layer linear network. As an example of this, we find the optimal learning rate $\alpha(t)$ throughout learning that maximizes the value function for the semantic task, to give more intuition on the impact of learning rate changes for complex step-like learning curves. In our implementation, we find the optimal learning rate using a surrogate variable $\rho(t)$ that facilitates the optimization. We use the learning effort framework to train a two layer linear network on the semantic dataset, just by modifying the learning dynamics as

$$\tau_w \frac{dW_1}{dt} = (1 + \rho(t)) \left[W_2^T \left(\Sigma_{xy}^T - W_2 W_1 \Sigma_x \right) - \lambda W_1 \right], \tag{A.44}$$

$$\tau_w \frac{dW_2}{dt} = (1 + \rho(t)) \left[\left(\Sigma_{xy}^T - W_2 W_1 \Sigma_x \right) W_1^T - \lambda W_2 \right]. \tag{A.45}$$

We define $\rho(t) = g(t)$ as our control signal (effort signal), therefore with $\rho = 0$ we recover the baseline learning dynamics of a network trained with SGD, and the effective learning rate is $\alpha(t) = (1 + \rho(t))\alpha$ with α the baseline learning rate. We set $\eta = 1$ and the cost $C(\rho(t)) = \beta \left(\rho - \text{offset}\right)^2$. The optimal learning rates $\alpha(t)$ and average training loss resulting from the control signal are depicted in Figure A.4.

As mentioned in Section 3.3, the control signal as the learning rate presents qualitatively the same behavior as in the single neuron case as shown in FIgure A.4. More control is allocated when γ increases due to pay off in the future, and more control is used when β

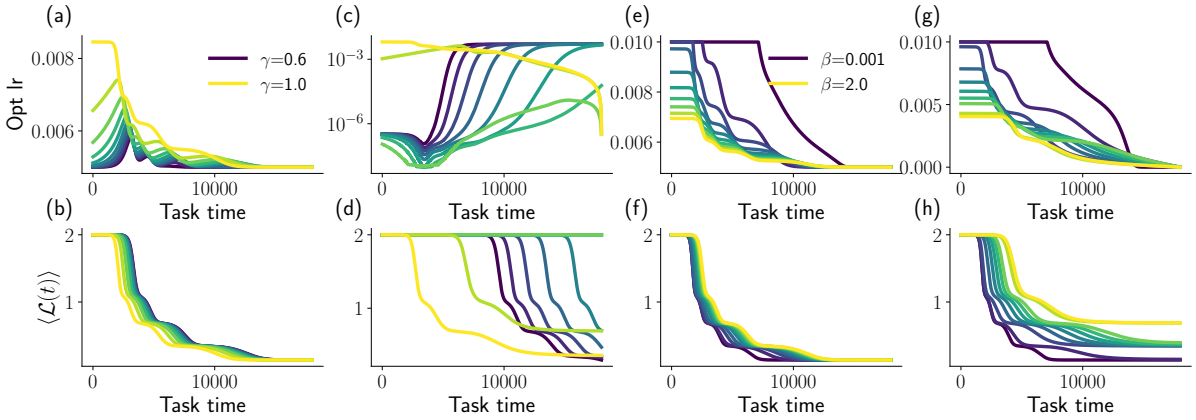

Figure A.4: Learning rate optimization. Top row: Optimal learning rates $\alpha(t)$. Bottom row: Resulting average dynamics. **(a)**, **(b)**, **(c)** and **(d)**: results when varying γ, **(a)** and **(b)** when the offset = 0 (baseline dynamics has 0 cost), **(c)** and **(d)** when the offset= -1 (any dynamics is costly). **(e)**, **(f)**, **(g)** and **(h)**: Results for varying β.

is decreased. The step-like shape in these plots is from learning each step in the hierarchy of the semantic task. Something important to notice is the results in the last column of Figure A.4, where depending on the cost of using control, the optimal solution is to learn some, but not all of the levels of the hierarchy. A higher cost of control leads to fewer levels learned in the hierarchy, meaning that deeper and harder levels in the structure might not be worth it if it is too costly for learning effort. The parameters used for this simulation are shown in Table A.3.

In Figure A.1 we show the results for these runs varying the cost coefficient β (Figures A.1a and A.1d), the regularization coefficient for the L_2 norm of the weights λ (Figures A.1b and A.1e), and different available time to learn the task T (Figures A.1c and A.1f). Increasing β leads to less control. Note that for $\beta = 0$ the amount of control does not explode, since changing $g(t)$ after learning the task will alter the input-output mapping function, leading to an increase in the loss. Increasing λ leads to an overall similar amount of control, but the control is sustained longer for higher λ, this is due to the high cost of increasing the size of the weight $w(t)$, which is absorbed by $g(t)$ to get closer the optimal

solution w^*, by making $\tilde{w}(t) = w(t)(1 + g(t))$ closer to w^*. To allow comparison, we normalized the time axis for the variations of available time, this parameter does not seem to change the control signal or instant reward rate within the time span to learn the task.

A.9.5 Effort Allocation

Here we present results and analysis of the gain modulation model trained on single datasets. Figures A.5 and A.6 depict the results of the effort allocation using gain modulation trained on the gaussian and semantic datasets respectively. As mentioned in Section 3.6, in all cases the learning of the dataset is speed up by the learning effort control signal. Most of the control is exerted in early stages of learning (to compensate with reward in later high reward stages of training), with peaks around times when the improvement in the loss is higher due to the weight learning dynamics, decaying to zero at the end of the given time frame. The L_2 norm of the weights is roughly higher when using control throughout the training, but it converges to the same value for the baseline and control case. Different are the trajectories for the L_1 norm (measuring sparsity), where the set of weights gets more sparse when using control, to minimize the cost of using the control signal while keeping the effect of it over the weights still high. This can be explicitly seen when training in all of the dataset when inspecting the weights and control evolution through time, shown in Figures A.7, A.8 and A.9 for the gaussian, semantic and MNIST datasets respectively. There is a cluster of weights that move closer to zero compared to the baseline training, and a few weights (near the number of non-zero gain coefficients in $G_i(t)$) become larger with time. Given that the input-output mapping is linear $\left(\hat{Y} = \tilde{W}_2(t)\tilde{W}_1(t)X\right)$, the solution for the linear regression problem (taking $\lambda = 0$ for simplicity) is given by $W^* = \tilde{W}_2\tilde{W}_1 = \Sigma_{xy}^T\Sigma_x^{-1}$ for both the baseline and the gain modulation case, and both cases reach the global solution for the linear regression problem (Fig. 3.8c). In the controlled case, because the instant net reward $v(t)$ also considers the cost of having $G(t) \neq 0$, the purpose of the control is to change

the learning dynamics and reach a solution such that $W^* = \tilde{W}_2\tilde{W}_1$ and $G(t) = 0$. Since regularization is considered in the backpropagation dynamics, the weights for the baseline and controlled case reach the same L_2 norm, but the weights when optimizing control are in general more sparse, as shown in the L_1 norms throughout learning in Fig. 3.8b. The reason for this is the over-parametrized nature of the network, having more parameters W than the ones needed to solve the linear regression.

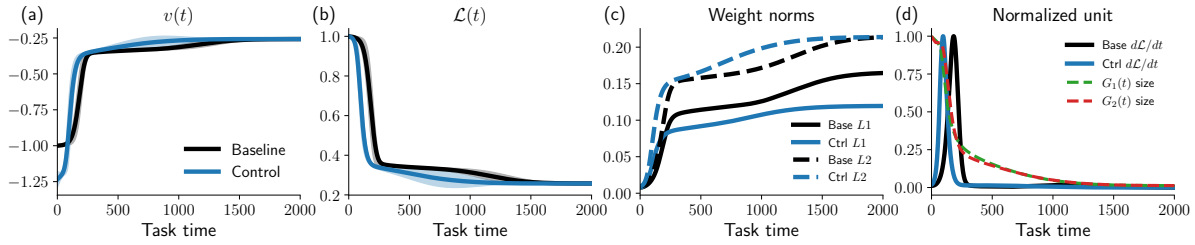

Figure A.5: Results of the gain modulation model trained on the gaussian dataset. **(a)**: Instant net reward $v(t)$, baseline vs controlled. **(b)**: Loss $\mathcal{L}(t)$ throughout learning. **(c)**: L1 and L2 norms of the weights. **(d)**: normalized $d_t\mathcal{L}(t)$, and normalized L2 norm of the control signal $G_1(t)$ and $G_2(t)$.

In the particular case of the effort allocation model trained on the semantic dataset, we can see two other extra features. First, because we initialized the network with small weights ($\sim 10^{-4}$), we can see step-like transitions in the loss through time $\mathcal{L}(t)$, a regime known as *rich learning* (Chizat et al., 2020; Flesch et al., 2022), where each step corresponds to learning one of the hierarchical concepts in the dataset, from highest to lowest. In this regime, the control signal is able to skip the plateaus when learning each level on the hierarchy. In addition, the control signal is the highest just the first step and decays exponentially until the end of training. We infer that the effect on the dynamics by the control signal is not merely scaling the learning rate, but since each neuron has its own independent gain modulation, the control signal can guide the complex learning dynamics to avoid facing step-like transitions in the loss function.

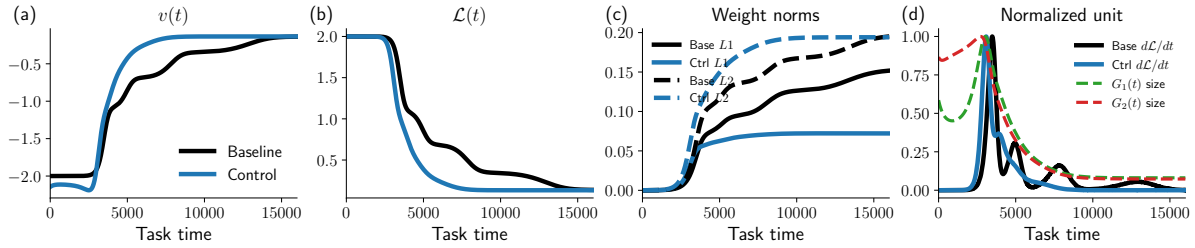

Figure A.6: Results of the gain modulation model trained on the semantic dataset. **(a)**: Instant net reward $v(t)$, baseline vs controlled. **(b)**: Loss $\mathcal{L}(t)$ throughout learning. **(c)**: L1 and L2 norms of the weights. **(d)**: normalized $d_t\mathcal{L}(t)$, and normalized L2 norm of the control signal $G_1(t)$ and $G_2(t)$.

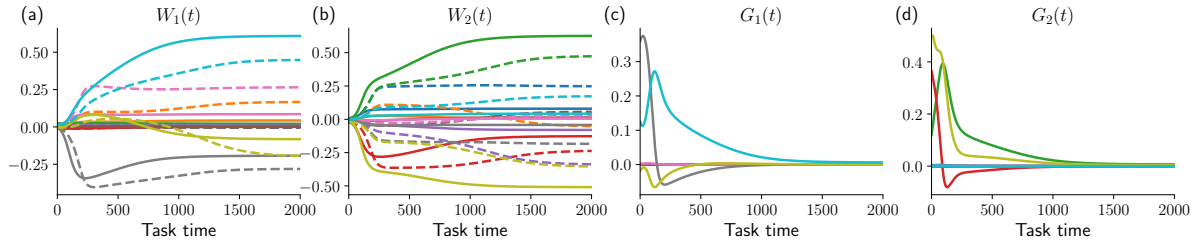

Figure A.7: Weight and control signal evolution through training in the gaussian dataset using the effort allocation task from Section 3.6. **(a)**: First layer weights $W_1(t)$ (baseline and control training depicted with solid and dashed lines respectively). **(b)**: Second layer weights $W_2(t)$. **(c)**: First layer gain modulation $G_1(t)$. **(d)**: Second layer gain modulation $G_2(t)$. Each color corresponds to a specific weight, and colors match between plots of weights and the control signal.

A.9.6 Task Switch

In Figures A.10 and A.11, additional results for the gain modulation model trained on the task switch are presented. In Figure A.10, note that the weight norm for the controlled case through the switches are larger. The cost of switching is transferred to the weights by making them larger, so the use of the control signal is less costly when switching. This can also be seen in Figure A.11, where the control signal is large only for a few weights, which in the long terms are the ones that become larger, reducing also the size of control needed to change the effective input-output transformation $\hat{Y} = \tilde{W}_2\tilde{W}_1X$. In

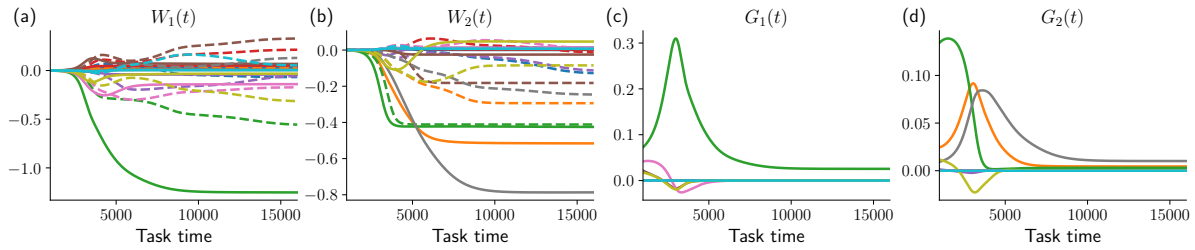

Figure A.8: Weight and control signal evolution through training in the smantic dataset using the effort allocation task from Section 3.6. **(a)**: First layer weights $W_1(t)$ (baseline and control training depicted with solid and dashed lines respectively). **(b)**: Second layer weights $W_2(t)$. **(c)**: First layer gain modulation $G_1(t)$. **(d)**: Second layer gain modulation $G_2(t)$. Each color corresponds to a specific weight, and colors match between plots of weights and the control signal.

addition, the weights when using control are grouped in two clusters, the ones influenced by the control signal which have larger absolute values, and the rest are pushed near zero (opposite to what is seen in the baseline case, with weights spread around zero). The gain modulation allocates resources for every switch in only a few weights, while the rest of the weights are pushed closer to zero to avoid interfering with the inference process through when switching.

A.9.7 Task Engagement

The set of datasets for the task engagement experiment was chosen based on how hard is to solve them with linear regression. In Figure A.12a the best achievable loss when classifying MNIST digits using linear regression is depicted for pairs of digits. The pairs used in the task engagement experiments, $(0, 1)$, $(7, 1)$ and $(8, 9)$ have corresponding 0.02, 0.036, and 0.055 optimal loss $\mathcal{L}^*$ respectively, therefore ordered from easiest to harder according to this metric.

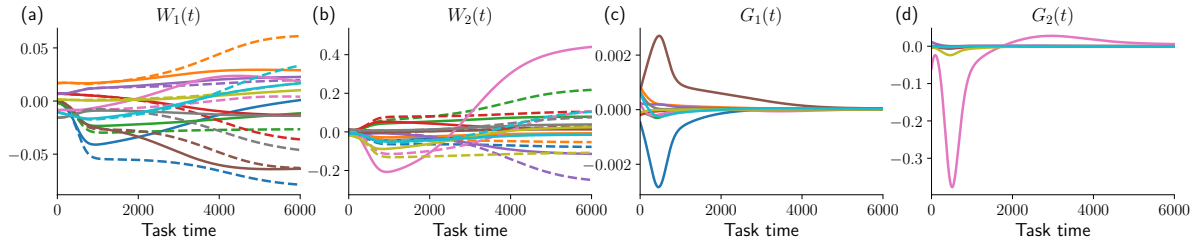

Figure A.9: Weight and control signal evolution through training in the MNIST dataset using the effort allocation task from Section 3.6. **(a)**: First layer weights $W_1(t)$ (baseline and control training depicted with solid and dashed lines respectively). **(b)**: Second layer weights $W_2(t)$. **(c)**: First layer gain modulation $G_1(t)$. **(d)**: Second layer gain modulation $G_2(t)$. Each color corresponds to a specific weight, and colors match between plots of weights and the control signal.

A.9.8 Category Assimilation

By taking the average across columns of the matrix in Figure A.12a, we obtained the average minimum loss per digit when compared to any other digit, shown in Figure A.12b (color bar with normalized values). When training the engagement modulation on the category assimilation task (Results in Section 3.5), the order of learning numbers is roughly the same as the difficulty in terms of linear separability. The easiest digits are focused on first, then harder ones. According to the linear separability metric, the order from easier to hardest are 0, 6, 1, 4, 7, 9, 2, 3, 8, 5, similar to what is depicted in Figure 3.6.

In addition to the engagement modulation model trained on the category assimilation task, we trained a restricted gain modulation model using the *neuron base* as described in Appendix A.3.1. In this model, the gain modulation for the first layer is disabled, then the only extra parameter adjusted to maximize expected return in equation 3.3 are the coefficients of the base ν_2^b. These coefficients scale the response of output neurons in the second layer, scaling the error signal, but also the value of the error signal itself (since it is changing the mapping as well). The results are similar in terms of order of the order of digits engaged through learning as shown in Figure A.13, but the effect in the loss

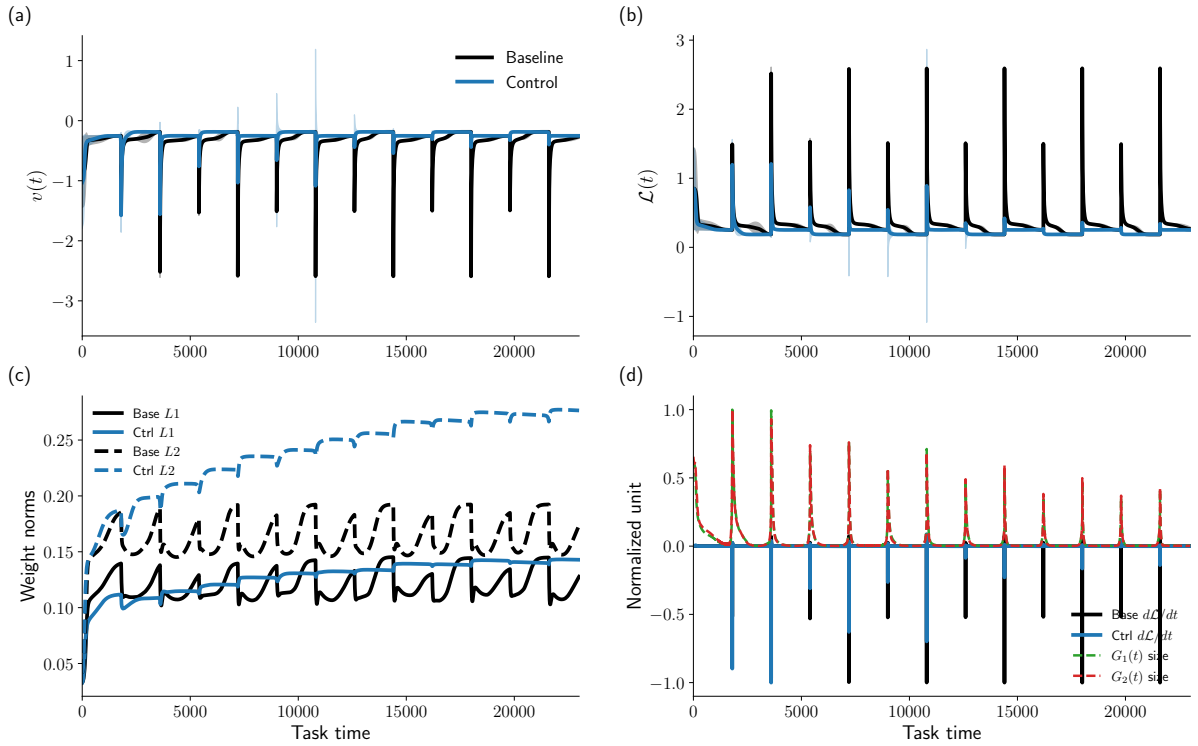

Figure A.10: Additional results of the gain modulation model trained on the Task Switch. **(a)**: Instant net reward $v(t)$, baseline vs controlled. **(b)**: Loss $\mathcal{L}(t)$ throughout learning. **(c)**: L1 and L2 norms of the weights. **(d)**: normalized $d_t\mathcal{L}(t)$, and normalized L2 norm of the control signal $G_1(t)$ and $G_2(t)$.

function is smaller because of the influence on the error signal (the value itself, not the scaling). This effect can be made explicit when deriving the learning dynamics equations for the weights (taking $G_1(t) = 0$) (for example for W_2), giving

$$\tau_w \frac{dW_2}{dt} = \left[\left(\Sigma_{xy}^T - \tilde{W}_2 W_1 \Sigma_x\right) W_1^T\right] \circ \tilde{G}_2(t) \tag{A.46}$$

$$= \nu \left[\left(\underbrace{\Sigma_{xy}^T - \nu W_2 W_1 \Sigma_x}_{\text{error signal}}\right) W_1^T\right] \tag{A.47}$$

where ν is a diagonal matrix with the coefficients $\nu_2^b(t)$ in the diagonal. Both the *learning rate per output* (the ν outside the square parenthesis) and the error signal are influenced

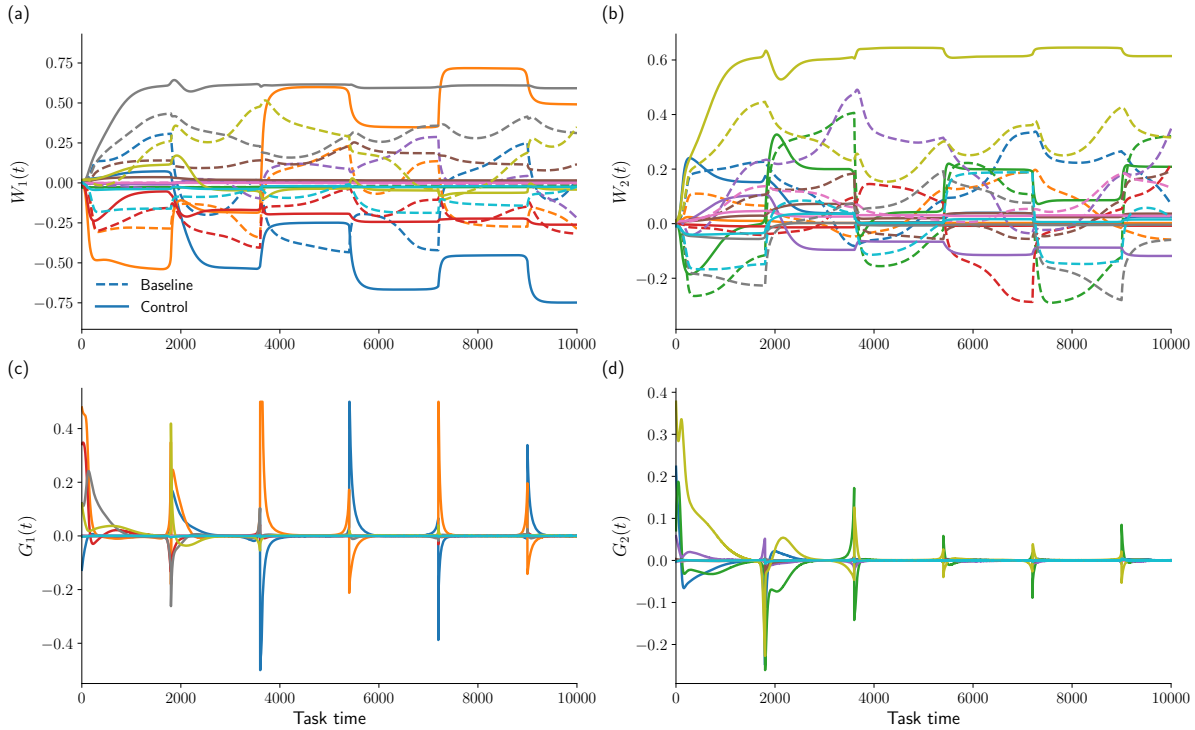

Figure A.11: Weight and control signal evolution through training in Task Switches. **(a)**: First layer weights $W_1(t)$. **(b)**: Second layer weights $W_2(t)$. **(c)**: First layer gain modulation $G_1(t)$. **(d)**: Second layer gain modulation $G_2(t)$. Each color corresponds to a specific weight, and colors match between plots of weights and the control signal.

by these coefficients. In the case of the engagement modulation explained in section A.5, the coefficients just scale the error signal in equation A.23.

A.9.9 Non-linear network

We used the gain modulation model (Effort Allocation experiment) to train a non-linear network on the gaussian dataset. We used a Taylor expansion around the mean to obtain an approximated equation for the non-linear dynatmics as explained in Appendix A.6. Because the non-linear function chosen is $\tanh(\cdot)$, and we initialize the network using small weights, the learning dynamics of the non-linear network are near linear at the beginning of the training, so the estimated weights and the real ones from SGD training

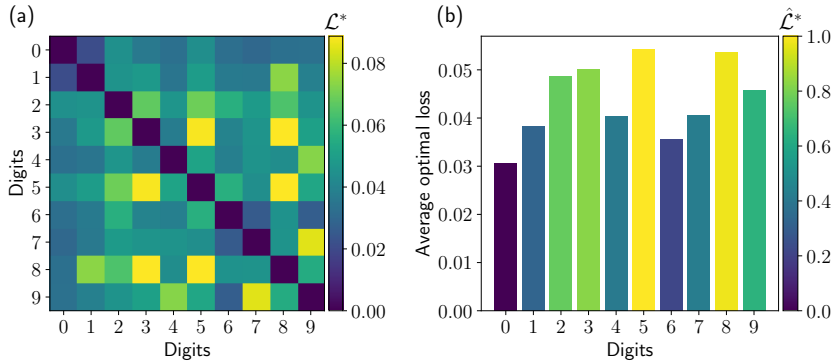

Figure A.12: **(a)**: Minimum error achievable $\mathcal{L}^*$ when classifying between 2 digits (rows and columns) using a linear classifier. **(b)**: Average of minimum error achievable per digit across all digits (average per row), color bar shows this normalized quantity.

are close as shown in Figure A.14 (panels (c), (d), (g), (h)). This is useful to estimate the control signal since most of the control is exerted in the early stages of learning, as depicted in Figures A.14i and A.14j (and in all linear networks testes throughout this work). The obtained control signal using the approximated dynamics is still able to improve training of the real non-linear network using SGD and gain modulation as shown in Figures A.14a and A.14b.

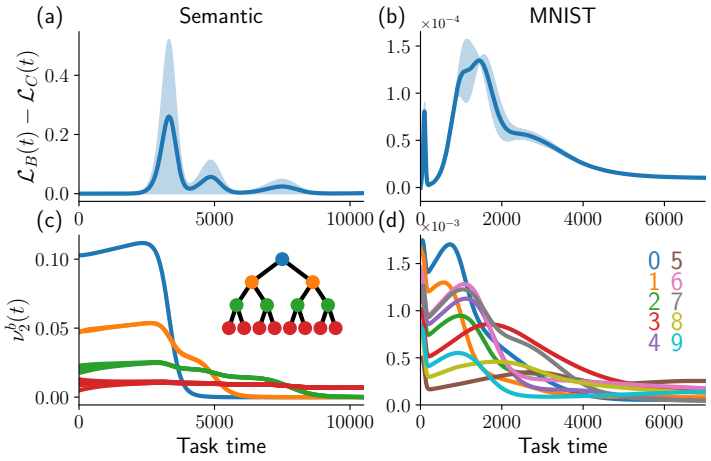

Figure A.13: Results for category assimilation task using the *neuron base* when restricting the gain modulation model. **(a) and (b)**: Improvement in the loss function when using control for MNIST and Semantic dataset respectively. **(c) and (d)**: Optimal category engagement coefficients for MNIST and Semantic respectively.

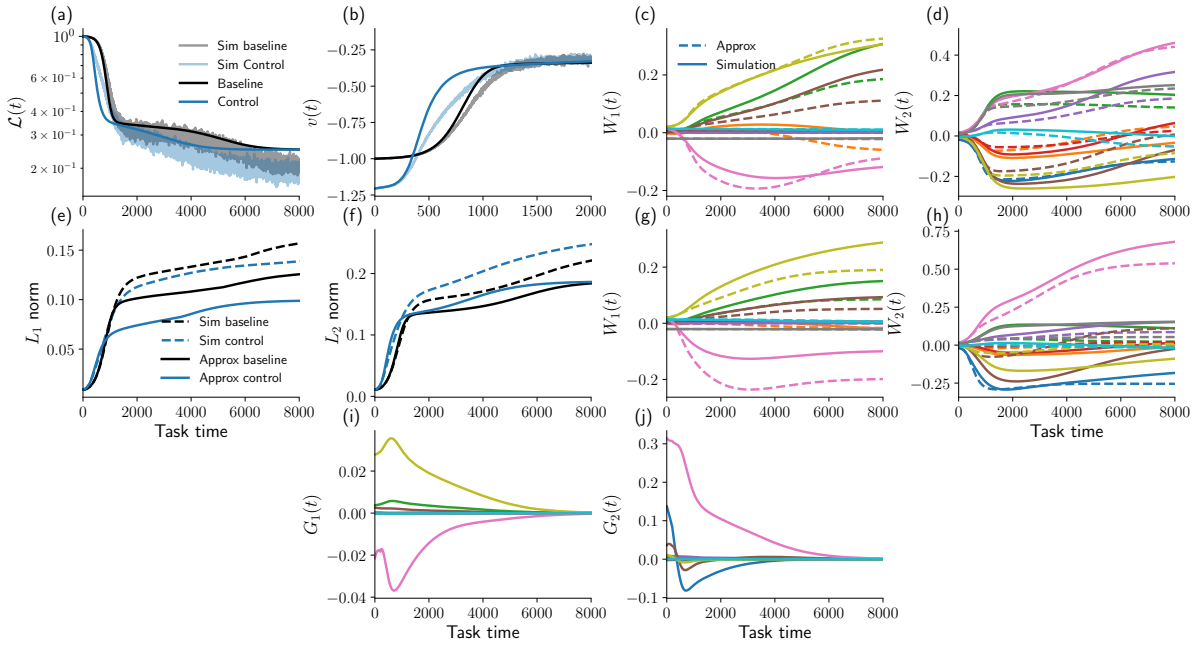

Figure A.14: Results of the learning effort framework in a non-linear network using a linear approximation. **(a) and (b)**: $\mathcal{L}(t)$ and $v(t)$ respectively. Solid lines are numerical solutions to the approximated learning dynamic for the baseline case in equation A.32 and the control case in equation A.33. Simulation training the real non-linear network using SGD shown in shaded lines, using the inferred gain modulation in the real non-linear network when using control. **(c) and (d)**: $W_1(t)$ and $W_2(t)$ respectively. In the baseline training, comparing the numerical solution of the approximated dynamics in equation A.32 (solid lines) with the weights from the simulation in the real non-linear network trained using SGD (dashed lines). **(e) and (f)**: L_1 and L_2 norms of the weights through training. Solid lines are numerical solutions to the approximated learning dynamic for the baseline case in equation A.32 and the control case in equation A.33. Simulation training the real non-linear network using SGD shown in dashed lines, using the inferred gain modulation in the real non-linear network when using control. **(g) and (h)**: $W_1(t)$ and $W_2(t)$ respectively. In the control case training, comparing the numerical solution of the approximated dynamics in equation A.33 (solid lines) with the weights from the simulation in the real non-linear network trained using SGD and using the gain modulation computed to maximize expected return (dashed lines). **(i) and (j)**: Control signals for first and second layer $G_1(t)$ and $G_2(t)$ respectively.

A.10 Simulation Parameters

Table A.1: Optimization Parameters for the single neuron example in Section 3.2.1, shown in Figures 3.2 and A.1. Every other variable is constant when varying each particular value of the parameter sweep.

Parameters	notation	Value
Network learning rate	α	0.001
Regularization coefficient	λ	0.1
Control lower bound	g_{MIN}	0
Control upper bound	g_{MAX}	0.5
Discount factor	γ	0.99
Reward conversion	η	1.0
Control cost coefficient	β	0.3
Control learning rate	α_g	10.0
Control gradient updates	K	700
Weight time scale	τ_w	1.0
Available time (A.U.)	T	600
Mean of Gaussians	μ	2.0
Intrinsic Noise	σ_x	1.0
Batch Size	B	128
γ values sweep	`10 ** (np.linspace(-8, 0, num=30, endpoint=True))`	
β values sweep	`np.linspace(1e-5, 2, num=30, endpoint=True)`	
σ_x values sweep	`np.linspace(1e-5, 5, num=30, endpoint=True)`	
λ values sweep	`np.linspace(0, 5, num=30, endpoint=True)`	
T values sweep	`[200 + i*50 for i in range(21)]`	

Table A.2: Parameters used for MAML simulation in App. A.9.2. The distribution of tasks where 5 MNIST pairs, (0, 1), (7, 1), (8, 9), (3, 8) and (5, 3), see App. A.8. Weights initialized from a gaussian distribution centered at 0 with standard deviation of 0.01

Parameters	notation	value
Network learning rate	α	0.005
Hidden units	H	40
Regularization coefficient	λ	0
Control lower bound	g_{MIN}	not bounded
Control upper bound	g_{MAX}	not bounded
Discount factor	γ	1.0
Reward conversion	η	1.0
Control cost coefficient	β	0
Control learning rate	α_g	0.005 with ADAM
Control gradient updates	K	2000
Weight time scale	τ_w	1.0
Optimized steps	T	from 1 to 340 every 20

Table A.3: Parameters used for learning rate $\alpha(t)$ optimization in App. A.9.4. γ and β where varied independently, keeping the default values when varying the other.

Parameters	notation	value
Network learning rate	α	0.005
Hidden units	H	40
Regularization coefficient	λ	0
Control lower bound	g_{MIN}	-1
Control upper bound	g_{MAX}	1
Reward conversion	η	1.0
Control learning rate	α_g	0.005 with ADAM
Control gradient updates	K	800
Weight time scale	τ_w	1.0
Available time (A.U.)	T	18000
default discount factor	γ	1.0
default cost coefficient	β	1.0
γ values sweep	`np.linspace(0.6, 1.0, num=10, endpoint=True)`	
β values sweep	`np.linspace(1e-3, 2, num=10, endpoint=True)`	

Table A.4: Optimization Parameters for results in Section 3.5. Dataset parameters are the following: For the **Attentive, Active and Vector** models, the three datasets used are MNIST digits, being $(0, 1)$, $(7, 1)$ and $(8, 9)$, each a binary classification task with a batch size of 256, images reshaped to (5×5) and flattened. **Eng. MNIST**: all digits were used with a batch size of 256, images reshaped to (5×5) and flattened. **Eng. Semantic**: Batch size of 32 and 4 hierarchy levels described in A.8. For every dataset, an extra 1 was concatenated to the input to account for the bias when multiplying by the weights.

Parameters	notation	Attentive	Active	Vector	Eng MNIST	Eng Semantic
Network learning rate	α	0.005	0.005	0.005	0.05	0.005
Hidden units	H	20	20	20	50	30
Regularization coefficient	λ	0.0	0.0	0.0	0.0	0.0
Eng. Coef. lower bound	ψ_{MIN}, ϕ_{MIN}	0.0	0.0	0.0	0.0	0.0
Eng. Coef. upper bound	ψ_{MAX}, ϕ_{MAX}	2.0	1.0	2.0	2.0	2.0
Discount factor	γ	0.99	0.99	0.99	0.99	0.99
Reward conversion	η	1.0	1.0	1.0	1.0	1.0
Control cost coefficient	β	0.1	0.1	0.1	5.0	5.0
Control learning rate	α_g	1.0	1.0	1.0	1.0	1.0
Control gradient updates	K	800	800	800	600	600
Weight time scale	τ_w	1.0	1.0	1.0	1.0	1.0
Available time (A.U.)	T	13000	13000	13000	30000	18000

Table A.5: Optimization Parameters for results in Section 3.6 and Appendix A.6. Dataset parameters are the following: **MNIST**: Batch size, 32; reshape size, (5×5); Digits, $(1, 3)$. **Gaussian and Non-linear**: Batch size, 32; $\mu_1 = 3$, $\mu_2 = 1$, $\sigma_1 = 1$, $\sigma_2 = 1$, $p = 0.8$. **Semantic**: Batch size, 32; hierarchy levels, 4. **Task Switch**: Gaussian 1: $\mu_1 = 3$, $\mu_2 = 1$, $\sigma_1 = 1$, $\sigma_2 = 1$, $p = 0.8$; Gaussian 2: $\mu_1 = -2$, $\mu_2 = 2$, $\sigma_1 = 1$, $\sigma_2 = 1$, $p = 0.2$, switch every 1800 iterations. For every dataset, an extra 1 was concatenated to the input to account for the bias when multiplying by the weights.

Parameters	notation	MNIST	Gaussians	Semantic	Task Switch	Non-linear
Network learning rate	α	0.005	0.005	0.005	0.005	0.001
Hidden units	H	50	6	30	8	8
Regularization coefficient	λ	0.01	0.01	0.01	0.001	0.0
Control lower bound	g_{MIN}	-0.5	-0.5	-0.5	-0.5	-0.5
Control upper bound	g_{MIN}	0.5	0.5	0.5	0.5	0.5
Discount factor	γ	0.99	0.99	0.99	0.99	0.99
Reward conversion	η	1.0	1.0	1.0	1.0	1.0
Control cost coefficient	β	0.3	0.3	0.3	0.3	0.3
Control learning rate	α_g	10.0	10.0	10.0	1.0	10.0
Control gradient updates	K	1000	1000	1000	1000	500
Weight time scale	τ_w	1.0	1.0	1.0	1.0	1.0
Available time (A.U.)	T	16000	16000	16000	23000	10000
Results		Figure 3.8	Figure A.5	Figure A.6	Figure 3.8	Figure A.14

Appendix B

Uncertainty Prioritized Experience Replay

B.1 Uncertainty decomposition in quantile regression

Here we provide some extra intuition on the difference between MSE curves when prioritising by total uncertainty $\mathcal{U}$, td-error $|\delta|$, estimated epistemic uncertainty $\hat{\mathcal{E}}_\delta$ and true epistemic uncertainty $\mathcal{E}^*$. Let's start by considering a single agent trained using quantile regression as explained in subsection 2.4.2. Consider the expected squared error of all quantiles indexed by τ and the target distribution Z, also defined in subsection 5.2.1 as $\mathcal{U}$:

$$\mathcal{U}^2 = \mathbb{E}_{\tau, r\sim Z}\left[(r-\theta_\tau)^2\right] = \mathbb{E}_r\left[r^2\right] - 2\mathbb{E}_r[r]\mathbb{E}_\tau\left[\theta_\tau\right] + \mathbb{E}_\tau\left[\theta_\tau^2\right], \tag{B.1}$$

$$= \mathbb{V}_r\left[r\right] + \bar{r}^2 - 2\bar{r}Q(a) + Q(a)^2 + \mathbb{V}_\tau\left[\theta_\tau\right], \tag{B.2}$$

$$= \underbrace{(\bar{r}-Q(a))^2}_{(\mathcal{E}^*)^2} + \underbrace{\mathbb{V}_r\left[r\right]}_{\text{Target variance}} + \underbrace{\mathbb{V}_\tau\left[\theta_\tau\right]}_{\text{Estimation variance}}. \tag{B.3}$$

The first term is the true epistemic uncertainty $\mathcal{E}^*$, second term and third term are the variance from the target, and the estimation variance. When using the total uncertainty as priority variable $p_i = \mathcal{U}$, the target and estimation uncertainty will be considered in the priority, therefore oversampling the noisiest arm as shown in the sampling probabilities depicted in Figures B.4 and B.5. When using the TD-error $p_i = |\delta_i|$, consider the expected squared TD-error

$$\mathbb{E}_r\left[\delta^2\right] = \left[(r - \mathbb{E}_\tau\left[\theta_\tau\right])^2\right], \tag{B.4}$$

$$= \underbrace{(\bar{r} - Q(a))^2}_{(\mathcal{E}^*)^2} + \underbrace{\mathbb{V}_r\left[r\right]}_{\text{Target variance}} . \tag{B.5}$$

Therefore, the TD-error does not prioritise by estimation variance, but it includes the target variance. Eventually, the target variance will be equal to the estimation variance, but from the start of the training, this is not true. Hence, the TD-error will also oversample the noisiest arm, but less compared to prioritising by total uncertainty $\mathcal{U}$. In practice, we do not have direct to $\mathbb{V}_r\left[r\right]$, in fact this is a quantity we are trying to estimate by using quantile regression. We have implicit access to the true distance $\mathcal{E}^*$ (epistemic uncertainty) through the decomposition $\mathcal{U} = \mathcal{E} + \mathcal{A}$ as explain in subsection 2.4.3, which is used to estimate epistemic uncertainty as in section 5.2. Prioritising using information gain achieve similar results compare to the direct use of $\mathcal{E}^*$ to prioritise replay. For further discussion about epistemic uncertainty ratios, refer to B.2.3.

B.2 Prioritisation Quantities based on Uncertainty

B.2.1 Information gain derivation

Given the setup in subsection 5.2.2, consider a hypothetical dataset of points $x_i \sim \mathcal{N}(\mu_x, \sigma_x^2)$. Our objective is to estimate the posterior distribution of the mean after observing one sample $P(\nu|x_i) \propto P(x_i|\nu)P(\nu)$ with a prior distribution of the mean $\nu \sim \mathcal{N}(\mu, \sigma^2)$.

Following the observation of a single sample x_i, the posterior distribution is Gaussian with variance $\sigma_\nu^2 = \frac{\sigma^2\sigma_x^2}{\sigma_x^2+\sigma^2}$. Knowing that the entropy of a Gaussian random variable is $\mathcal{H}(P(\nu)) = 1/2\log(2\pi e\sigma^2)$, we proceed to compute the information gain (or entropy reduction) of the posterior distribution as

$$\Delta\mathcal{H} = \mathcal{H}(P(\nu)) - \mathcal{H}(P(\nu|x_i)) \tag{B.6}$$

$$= \frac{1}{2}\log\left(2\pi e\sigma^2\right) - \frac{1}{2}\log\left(2\pi e\left(\frac{\sigma^2\sigma_x^2}{\sigma_x^2+\sigma^2}\right)\right) \tag{B.7}$$

$$= \frac{1}{2}\log\left(1+\frac{\sigma^2}{\sigma_x^2}\right). \tag{B.8}$$

We consider $\sigma^2 = \hat{\mathcal{E}}_\delta$ as a form of epistemic uncertainty that can be reduced by sampling more points, and $\sigma_x^2 = \hat{\mathcal{A}}$ as aleatoric uncertainty, which is the underlying irreducible noise of the data, giving a prioritisation variable

$$p_i = \Delta\mathcal{H}_\delta = \frac{1}{2}\log\left(1+\frac{\hat{\mathcal{E}}_\delta(s,a)}{\hat{\mathcal{A}}(s,a)}\right). \tag{B.9}$$

As discussed in the main text, other form of priority variables p_i can be effective in some settings. We extend the discussion about uncertainty ratios in the following sections, and show empirical results in the arm bandit task in B.3.

B.2.2 Variance as Uncertainty Estimation

To justify our choice of $\sigma^2 = \hat{\mathcal{E}}$ and $\sigma_x^2 = \hat{\mathcal{E}}$ in the information gain described in equation 5.12, we train an ensemble of distribution regressors to learn the mean from Gaussian samples ($\mu_x = 2$, $\sigma_x = 1$). This ensemble is compared to the Bayesian posterior distribution of the mean (Gaussian prior, likelihood, and posterior) as detailed in subsection 5.2.2. The ensemble, composed of 50 distribution quantile regressors, is initialized with the same prior as the Bayesian model – a unit variance Gaussian centered at 0 – by sampling 50 values from this prior and setting the initial mean of each quantile

regressor accordingly. Both the ensemble and Bayesian models are trained using samples from the data distribution. The ensemble training process follows the method described in the paper, and where each regressor is updated with a probability of 0.5 to introduce ensemble variability. The updates are performed using quantile regression as outlined in subsection 2.4.2. At each time step, the ensemble's estimated posterior is computed by averaging the means of all regressors and calculating the variance of these means.

figure 5.1 (a) and (b) illustrate the posterior evolution of both models from the same starting prior, given more samples. Both posteriors exhibit similar trends (the Bayesian model converges faster to the mean, due to the use of TD-updates with a smaller learning rate in the ensemble). In the Bayesian model, posterior sharpness is quantified by its variance, σ_ν^2, whereas for the ensemble, it corresponds to the epistemic uncertainty $\hat{\mathcal{E}}$ from equation 2.47. Both measures converge to zero, but at different rates figure 5.1d. The aleatoric uncertainty of the data, by definition the variance σ_x^2, is well approximated by $\hat{\mathcal{A}}$ from equation 2.47, and shown in figure 5.1f. The slight underestimation of the variance is a known issue in quantile regression, as quantiles often fail to capture lower probability regions (figure 5.1c), leading to an underestimation of the distribution's variance. Our contribution to prioritization involves incorporating the distance to the target δ_Θ from equation 5.10 (figure 5.1e). This approach prioritizes transitions not only based on the reduction in posterior variance but also on the regressor's proximity to the target.

B.2.3 Uncertainty Ratios

Having arrived at various methods for estimating epistemic and aleatoric uncertainty using distributional reinforcement learning, we now consider how to construct prioritisation variables from these estimates. Naively, one might consider prioritising directly using the epistemic uncertainty estimate; but neglecting the inherent noise or aleatoric uncertainty entirely ignores the 'learnability' of the data. Many methods in related learning domains can be interpreted as incorporating both uncertainties, including Kalman learning (Welch

and Bishop, 1995; Gershman, 2017), active learning (Cohn et al., 1994), weighted least-squares regression (Greene, 2000), and corresponding extensions in deep learning and reinforcement learning (Mai et al., 2022). To gain an intuition on how the choice of functional form might impact our particular use-case of prioritisation for various magnitudes of epistemic and aleatoric uncertainty.

$\mathcal{E}/\mathcal{A}$ has desirable properties. For instance under Bayesian learning of Gaussian distributions, $\log(1+\mathcal{E}/A)$ maximises information gain (see subsection 5.2.2), but discontinuities around very low noise must be dealt with—for instance by adding small constants to the denominator. Normalising instead with the total uncertainty is another way of handling the discontinuities. $\mathcal{E}^2/\mathcal{U}$ in particular corresponds to maximising reduction in variance under Bayesian learning in the same Gaussian setting. Both of these forms have the advantage over e.g. $\mathcal{E}/\mathcal{A}$ of preferring low epistemic uncertainty for equal ratios of epistemic and aleatoric uncertainties, i.e. they are not constant along the diagonal of the phase diagram. More generally, it is difficult to say *a priori* which functional form is optimal. Many factors, including the data distributions, model and learning rule will play a role. Further discussion on these considerations can be found in subsection B.2.4. These trade-offs are also borne out empirically in the experimental section 5.3 & section 5.4 below.

B.2.4 Bias as Temperature

Lahlou et al. (2022) and others make an equivalence between excess risk and epistemic uncertainty. Concretely, if $f^*(x)$ is the Bayes optimal predictor, the excess risk is defined as:

$$\mathrm{ER}(f,x) = R(f,x) - R(f^*,x), \tag{B.10}$$

where R is the risk and $R(f^*,x)$ can be thought of as the aleatoric uncertainty.

One possible issue arises in overstating the connection between excess risk and epistemic

uncertainty. Consider the case where there is model mis-specification, and f^* is not in the model class; then assuming the model class is fixed (as is standard), then the lower bound of $\mathrm{ER}(f, x)$ is non-zero. Stated differently, it is *not* fully reducible, which is often viewed as a central property of epistemic uncertainty. For some applications this distinction may not be important; there is some non-zero lower bound to the epistemic uncertainty but the ordering and correlations are intact under this equivalence. But it could also play a significant role. For us in particular, adopting this equivalence has two related consequences:

1. The model mis-specification acts as a temperature for our prioritisation distribution;
2. The ratio, or more generally the functional form of our prioritisation variable, can offset this temperature.

To make the above equation fully reducible, we would need to further subtract a term capturing the difference between the Bayes predictor, and the best predictor in the model class i.e. the model bias or mis-specification term. Let us denote this term by C, and assume it constant over the domain. And let us denote the fully reducible uncertainty by η. In the case where we use the excess risk, the prioritisation of sample i is given by

$$\text{[Vanilla]} \qquad p_i = \frac{\eta_i + C}{\sum_i (\eta_i + C)} = \frac{\eta_i + C}{NC + \sum_i \eta_i}. \tag{B.11}$$

It is easy to see how C acts as a temperature. In the limit of large C we get a uniform distribution over samples. Similarly if $C = 0$ we recover the 'true' distribution for reducible uncertainty.

It is of course hard to measure this model mis-specification term. In large networks we can assume the capacity is unlikely to be restrictive, but perhaps other parts of the training regime could play a part. Importantly, the above holds true not just for model mis-specification, but also if there is any systematic error in the epistemic uncertainty

estimate (i.e. think of C as an error on the epistemic uncertainty estimate).

B.2.5 Prioritisation Distribution Entropy

Assuming the above effect is significant, might a different functional form (as discussed in subsection B.2.3) for prioritisation alleviate the impact? Consider the following additional options:

$$[\mathcal{E}/\mathcal{U}] \qquad p_i = \frac{\frac{\eta_i+C}{\eta_i+C+\beta_i}}{\sum_i \frac{\eta_i+C}{\eta_i+C+\beta_i}}; \tag{B.12}$$

$$[\mathcal{E}^2/\mathcal{U}] \qquad p_i = \frac{\frac{(\eta_i+C)^2}{\eta_i+C+\beta_i}}{\sum_i \frac{(\eta_i+C)^2}{\eta_i+C+\beta_i}}; \tag{B.13}$$

and more generally,

$$[\mathcal{E}^m/\mathcal{U}] \qquad p_i = \frac{\frac{(\eta_i+C)^m}{\eta_i+C+\beta_i}}{\sum_i \frac{(\eta_i+C)^m}{\eta_i+C+\beta_i}}. \tag{B.14}$$

In the limit of large C all of these forms tend to a uniform distribution. However, at what rate? And is there anything else interesting we can say?

Consider the following toy problem:

- Populate "replay" buffer with N samples;
- Each sample's reducible uncertainty is sampled from ρ_η;
- Each sample's reducible uncertainty is sampled from ρ_β;
- C is constant over the samples.

We can plot as a function of C the entropy of the prioritisation distribution for the functional forms above. Such a plot is shown for various choices of ρ_η, ρ_β in figure B.1. Clearly, as C increases the entropy in the distribution increases and saturates at some maximum entropy. There is some variation in the entropy ordering depending on the

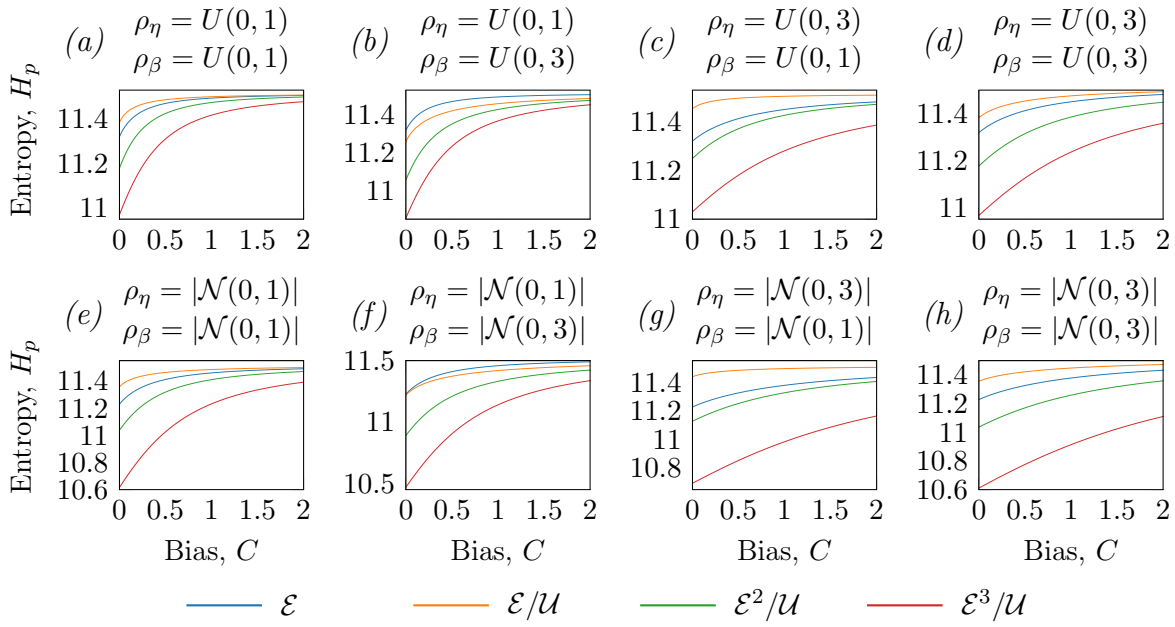

Figure B.1: **Ratios can reduce entropy of distribution under bias.**

exact ρ_η, ρ_β distributions; in some instances the vanilla form is lower entropy than $\mathcal{E}/\mathcal{U}$, but in general the entropy remains lower for longer (as a function of C) when the exponent in the nominator is higher. This is not a particularly surprising result, but lends support to the idea that a higher order function of $\mathcal{E}$ in a ratio form is desirable for prioritisation.

B.2.6 Relation to $\mathcal{E}$ under 0 Bias

Now let us consider a more interesting measure. Ordinarily, or naively—in the sense that this is the first order approach—we want our prioritisation variable to be the vanilla prescription; and ideally we would want C to be 0. We can measure the difference, which we denote δ_i to this ideal for each functional form as a function of C. This plot is show for various choices of ρ_η, ρ_β in figure B.2.

In general, the standard $\mathcal{E}/\mathcal{U}$ ratio is poor, it has systematically higher mean and variance of error. Beyond that, a clear trade-off emerges: as you increase the exponent m, then for high C there is lower deviation from the 'correct' distribution for priority. This is related

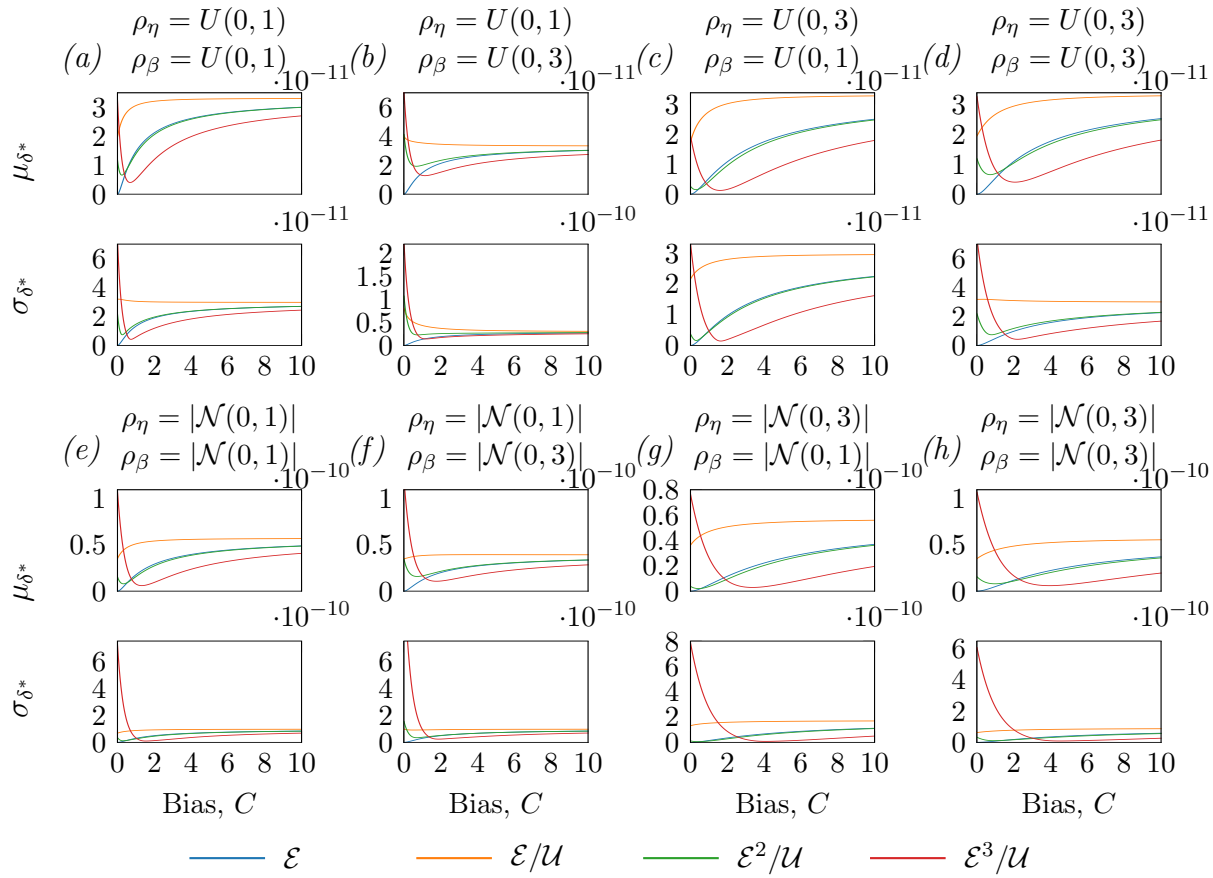

Figure B.2: $\mathcal{E}^2/\mathcal{U}$ **closely approximates** E **for non-trivial bias.**

to maintaining lower entropy and tending to a uniform distribution more slowly. However, for lower C you are likely to be more wrong, catastrophically so. This trade-off for $m = 3$ is effectively crossed when the red line intersects with the blue in these plots. The point at which this intersection happens will be a function of various things, primarily the underlying distributions—in this case ρ_η, ρ_β.

Interestingly however, for $m = 2$ there is very fast convergence of $\mathcal{E}^2/\mathcal{U}$ and $\mathcal{E}$ as a function of C. So while $m = 3$ has a very stark trade-off, $m = 2$ is less extreme: For low C it may make you more wrong but generally you will have very similar average error by this metric to the vanilla case; all the while the entropy of the distribution will be much lower and more informative (as shown in figure B.1). This toy model is clearly very simplistic,

not least the lack of variation in C over the samples; but future work could be dedicated to understanding these trade-offs more formally in the context of prioritized replay.

B.2.7 Off-setting Bias with TD Term

Leaving aside the ratio forms, the consequences of the temperature effect may differ depending on the choice of epistemic uncertainty estimate we use. The methods we discuss in section 2.4 all effectively use the equivalence of excess risk and epistemic uncertainty, and so do not explicitly consider the possibility of model bias. The possible exception is the method resulting from the expansion of the average error over the quantiles and ensemble in subsection 5.2.1. The main difference between this decomposition and that of Clements et al. (2020) is a term that encodes the distance from the target:

$$\delta_{\Theta}^{2} = \left(\Theta - \mathbb{E}_{\psi,i}\left[\theta_i(\psi)\right]\right)^2 . \tag{B.15}$$

This term *could* guard against two possible shortcomings of the decomposition in Clements et al. (2020):

1. Consider the pathological case in which each ensemble is initialised identically, then each quantile will have zero variance and the epistemic uncertainty measure from Clements et al. (2020) will be zero. Even if there is independence at initialisation, there may be characteristic learning trajectories or other systematic biases that push the ensemble together and lead to an underestimate in epistemic uncertainty. Here, the term above—if treated as part of the epistemic uncertainty—can continue to drive learning in ways we want.

2. However, it could be that the ensemble behaves nicely and the metric over the ensemble from Clements et al. (2020) is principally a good one, *but* that there is significant model bias. This could also be captured by the term above but would need to be *subtracted* from the total error in order to get a fully reducible measure

for epistemic uncertainty (as per the argument discussed above).

Which of the two problems is more pronounced is difficult to know *a priori*, and could be an avenue for future work. Empirically, the performance of the UPER agent in section 5.4 suggests that the former is the greater effect—at least on the atari benchmark with the model architecture and learning setting used.

B.3 Arm-Bandit Task

The hyperparameters used in the Arm-Bandit Task shown in section 5.3 are shown below:

- Number of train steps: 20^5
- Learning rate annealing: $0.005 \cdot 2^{-\text{iters}/40000}$.
- Init variance estimation: uniformly sampled from to 0.1
- Number of agents in the ensemble: 30
- $\alpha = 0.7$, β is annealing from 0.5 to 1 in 0.4 to 1 in proportional prioritisation as in the original work by Schaul et al. (2016).
- n arms: $n_a = 5$, $\bar{r} = 2$, $\sigma_{\max} = 2$ and $\sigma_{\min} = 0.1$.
- Number of quantiles: 30.
- Quantiles initialized as uniform distribution between -1 and 1. For the main results in figure 5.2, θ_τ are initialized randomly between -1 and 1, then sorted to describe a cumulative distribution.
- Each agent in the ensemble is updated with probability $1/2$ on each step.
- For the shifted arm experiment, the mean reward per arm $\bar{r}(a) = 3, 2.75, 2.5, 2.25, 2$ for arms $1, 2, 3, 4$ and 5.

figure B.3 show the mean squared error from the estimated $Q(a) = \mathbb{E}_{j,\psi}\left[\theta_j(\psi)\right]$ to the true mean, where ψ denotes agents in the ensemble case. figure B.5 and figure B.5 show the probability of sampling each arm from the memory buffer throughout the training, and the mean square error from the estimated arm value $Q(a)$ to the true arm value $\bar{r}$ (the same for every arm). In addition, we depict the evolution of uncertainty quantities for all prioritisation variables for the arm bandit task in figure B.6.

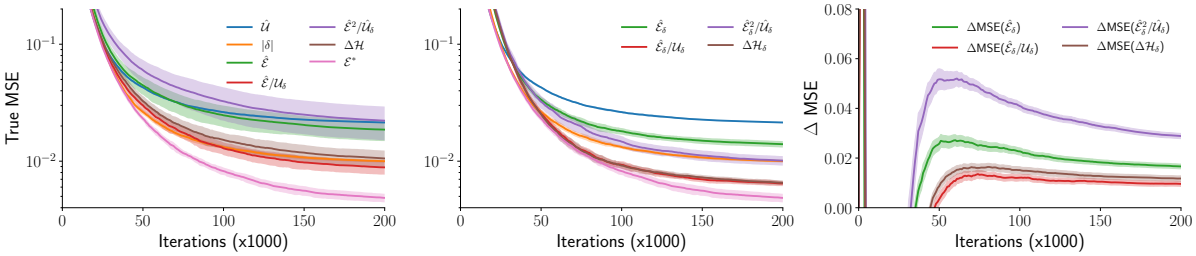

Figure B.3: Comparison of MSE for different prioritisation scheemes. Left panel, shows ratios and information gain based on epistemic uncertainty $\hat{\mathcal{E}}$ proposed by Clements et al. (2020). Middle panel, shows ratios and information gain based on our proposed *target epistemic uncertianty* $\hat{\mathcal{E}}_\delta$. Right panel, different in MSE between curves in the left panel and right panel for the shifted arm task. For instance, $\Delta\text{MSE}(\hat{\mathcal{E}}_\delta) = \text{MSE}(\hat{\mathcal{E}}) - \text{MSE}(\hat{\mathcal{E}}_\delta)$, showing that our proposed $\hat{\mathcal{E}}_\delta$ is in general better for prioritisation in the arm-bandit task. Averaged across 10 seeds.

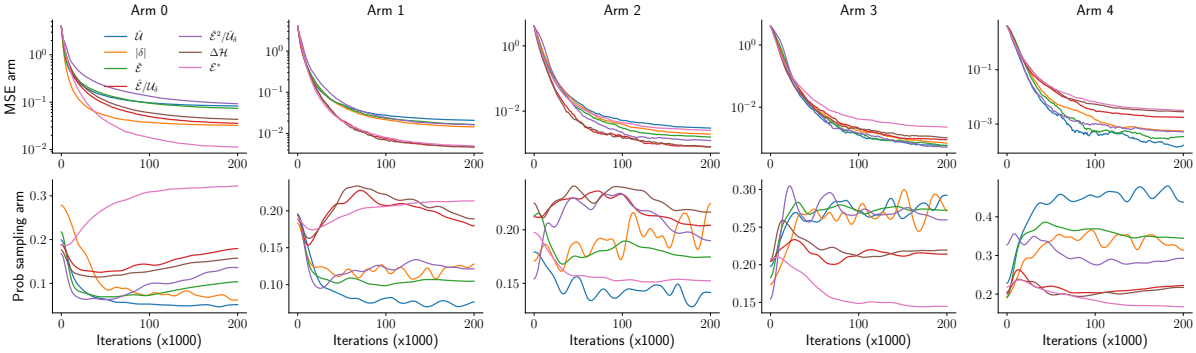

Figure B.4: Comparison of MSE for different prioritisation scheemes using $\hat{\mathcal{E}}$ based prioritisation. Total uncertainty $\mathcal{U}$ and TD-error prioritisation tend to oversample high variance arms compared to epistemic uncertainty prioritisation.

B.4 Gridworld Experiments

The hyperparameters used in figure 5.2 are listed below:

- Learning rate: 0.1
- Discount factor, γ: 0.9
- Exploration co-efficient, ϵ: 0.95
- Buffer capacity: 10,000

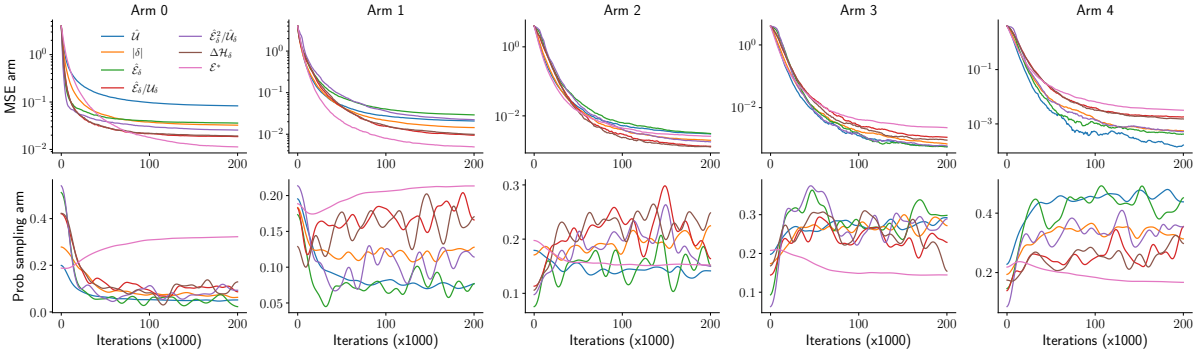

Figure B.5: Comparison of MSE for different prioritisation scheemes using $\hat{\mathcal{E}}_\delta$ based prioritisation. Total uncertainty $\mathcal{U}$ and TD-error prioritisation tend to oversample high variance arms compared to epistemic uncertainty prioritisation.

- Episode timeout: 1000 steps
- Random reward distribution: $\mathcal{N}(0, 2)$

For every 10 steps of 'direct' interaction and learning from the environment, the agent makes 5 updates with 'indirect' learning from the buffer replay. The data shown in the plots consists of 100 repeats and is smoothed over a window of 10.

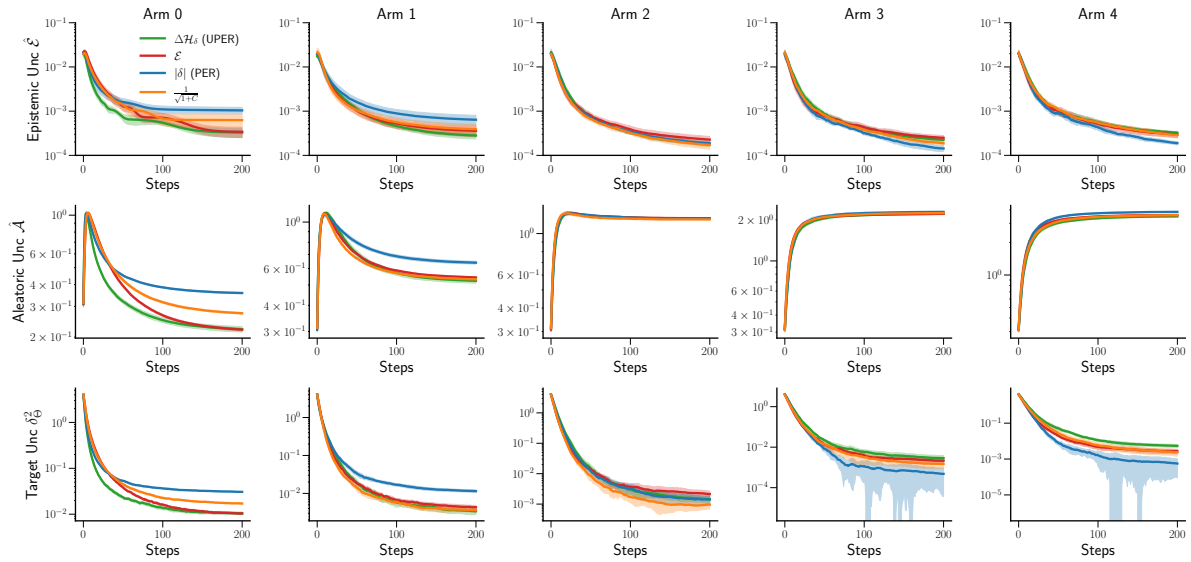

Figure B.6: Epistemic uncertainty $\hat{\mathcal{E}}$ and target uncertainty δ^2_Θ decrease more rapidly for lower noise arm (first column), for UPER compared to other methods. The inclusion of aleatoric uncertainty in the prioritization variable, as utilized in the information gain formula, aims to sample transitions with high epistemic uncertainty for its reduction, while also avoiding transitions with high aleatoric uncertainty with less learnable content. This rationale is reflected in the ratio presented in the derived $\Delta\mathcal{H}_\delta$, and shown its effect in the sampling probabilities plotted in figure B.5. The TD-error tends to oversample noisier transitions, resulting in less frequent updates for the least noisy arm, consequently leading to higher levels of epistemic and target uncertainty for that arm.

B.5 Atari Experiments

Cumulated training improvement of UPER over PER, QRDQN, QR-PER and QR-ENS-PER are shown in figure B.8 to figure B.11. The accumulate percent improvement $C_{\text{UPER/PER}}$, (same for $C_{\text{UPER/QRDQN}}$ and the rest), is computed as

$$C_{\text{UPER/PER}} = \frac{\sum_t [\text{UPER}_{\text{human}}(t) - \text{PER}_{\text{human}}(t)]}{\sum_t \text{PER}_{\text{human}}(t)} \cdot 100 \tag{B.16}$$

where t indexes training time, and $\text{UPER}_{\text{human}}$ (same for $\text{PER}_{\text{human}}$ and $\text{QRDQN}_{\text{human}}$) denotes human normalized performance.

For the baseline experiments we use the same implementations as those of the original papers, including hyperparemeter specifications. For our UPER method, we performed a limited hyperparameter sweep over 3 key hyperparameters: learning rate and ϵ for the optimizer, and the priority exponent. The sweep ranged 3×10^{-5} to 5×10^{-5} for the learning rate, 6.1×10^{-7} to 3.125×10^{-4} for ϵ and 0.6 to 1 for the priority exponent. We chose values for our final experiments based on average performance over 2 seeds across a sub-selection of 5 Atari games (chopper command, asterix, gopher, space invaders, and battlezone).

B.5.1 QR Models Ablation

To demonstrate the effectiveness of the information gain prioritization, and to confirm that the performance improvement stems from our proposed prioritization variable, we compared UPER to identical QR-DQN ensemble agents, maintaining the same architecture but altering only the prioritization variable. The results are presented in figure B.7. UPER outperforms alternative approaches such as QR-DQN-PER, which uses the TD-error to prioritize (as previously shown in figure 5.3), QR-ENS-EPI, which directly prioritizes

using epistemic uncertainty as defined in equation 5.14, and QR-ENS-UNI, which uses uniform sampling. These findings highlight the significance of both epistemic uncertainty and aleatoric uncertainty in prioritizing replay, as included in the information gain term. Additionally, these results confirm that the performance improvement can be solely attributed to the prioritization variable, as the QR-DQN ensemble architecture employed in each agent remains constant.

B.5.2 Computational cost

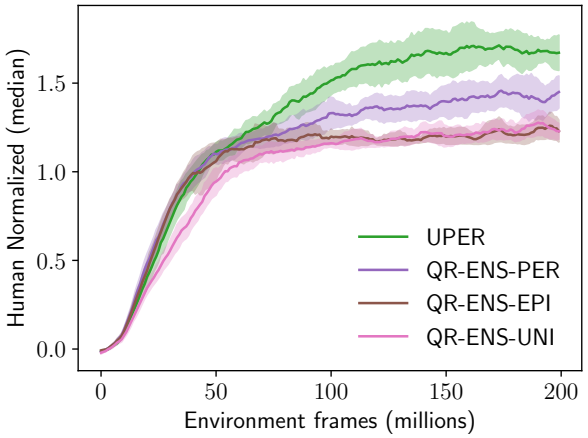

Figure B.7: Comparison of ablated prioritization variables. Median Human Normalized Score for QR-DQN ensembles, where only the prioritization variable is changed. UPER, PER, EPI, and UNI use the information gain in equation 5.12, the TD-error, target epistemic uncertainty in equation 5.14, and uniform sampling, respectively.

For the main Atari-57 benchmark results, average clock time training for PER, QR-DQN, and UPER (standard DQN, distributed RL agent, and ensemble of distributed RL agents) are $\approx$ 150 hours, $\approx$ 149 hours, and $\approx$ 162 hours respectively, all implemented in JAX running in Tesla V100 NVIDIA Tensor Cores.

To generate Table 5.1, we conducted experiments on a laptop equipped with an i5-10500H CPU (2.50GHz) and a 6GB NVIDIA GeForce RTX 3060 Mobile/Max-Q (not the same architecture as the main results in the paper, which uses Tesla V100 NVIDIA Tensor Cores). We ran 40 iterations of Pong for each model, using the last 20 iterations to avoid initialization and buffer filling times. The experiments were conducted on both CPU and GPU using different network architectures. In each iteration, the agent processed 1000 frames and performed one batch update of 64 transitions, with 4

frames per iteration. For all these runs, we used the publicly available implementation of DQN Zoo by DeepMind. Table 5.1 shows the time it takes for each iteration (1000 frames and a batch update) in seconds, along with standard deviations. There are two main conclusions from this experiment. First, most of the time consumed during each iteration is spent running the game engine (the 1000 frames per iteration), which is typically run on the CPU. This is evident from the small difference in time between QR-DQN and DQN in both the CPU and GPU cases. This difference could be larger in favor of the GPU if the batch size is increased and the frames per iteration are reduced. Second, we are significantly leveraging the parallelization capabilities of GPUs, as shown by the reduced times for the QR-DQN-ENS model (the architecture needed for UPER) when comparing GPU to CPU performance. The 2-second gap per iteration when comparing QR-DQN-ENS with QR-DQN and DQN is further reduced by utilizing V100 GPUs, as demonstrated by the training times reported in the main Atari-57 experiment.

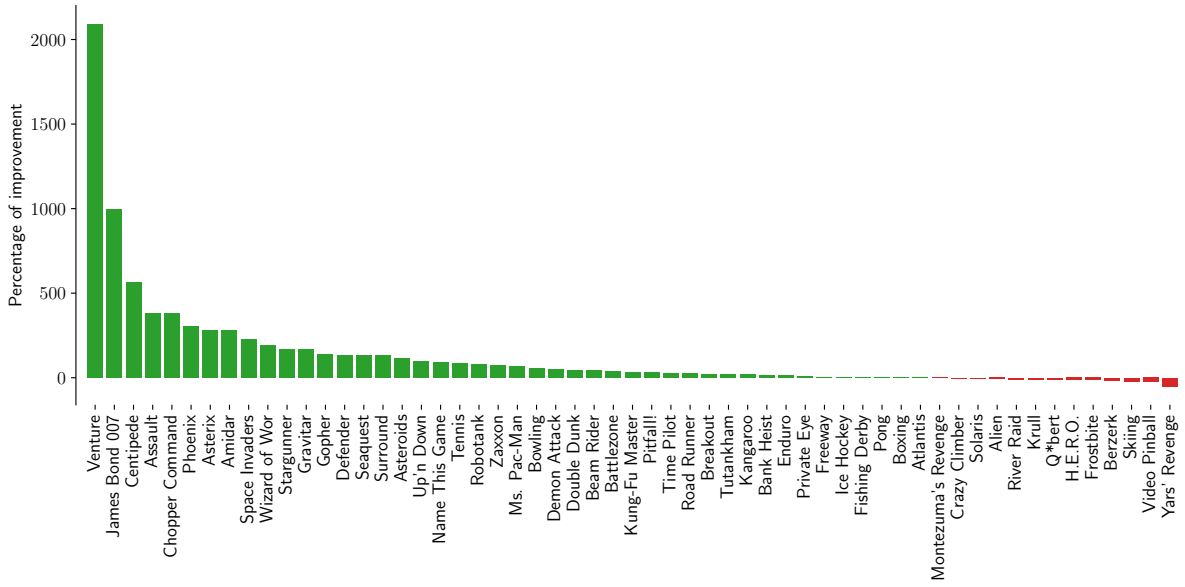

Figure B.8: Cumulated training improvement of UPER over PER defined as $C_{\text{UPER/PER}}$.

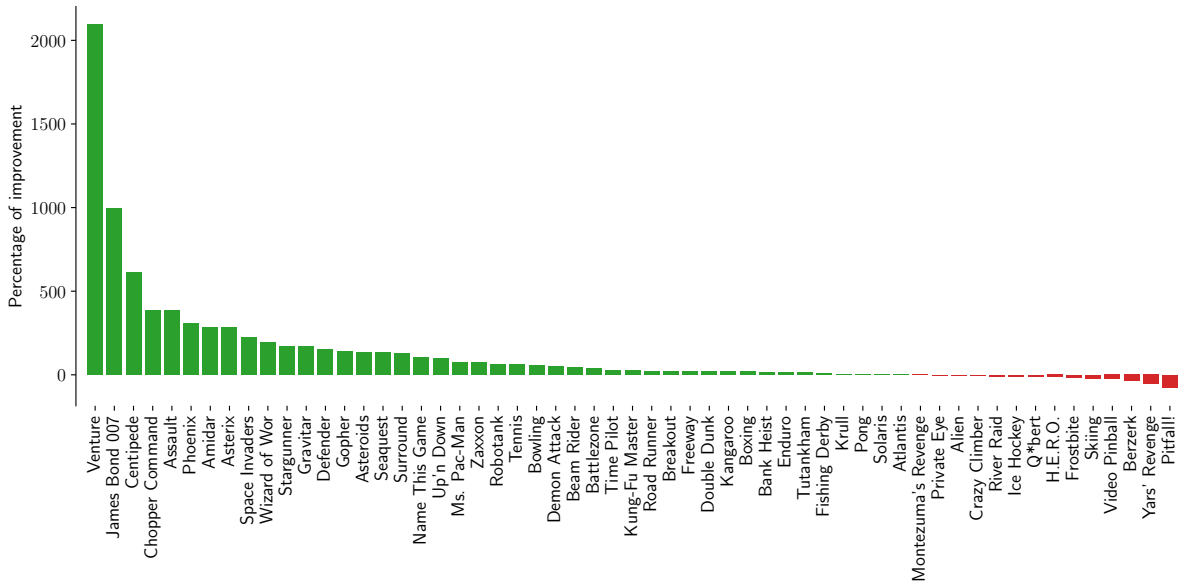

Figure B.9: Cumulated training improvement of UPER over QR-DQN defined as $C_{\text{UPER/QR-DQN}}$.

B.6 C51

To help assess whether the UPER methodology would also work when used in conjunction with other deep learning algorithms beyond QR-DQN, we performed a smaller scale set of experiments using the C51 algorithm **?**. We selected 5 Atari games in which ablations from Hessel et al. (2017) suggested vanilla PER was ineffective or even detrimental. Results on these 5 games comparing an ensemble C51 agent with PER vs an ensemble C51 agent with UPER are shown in figure B.13. Our method is significantly better on 4 games and similar in the fifth.

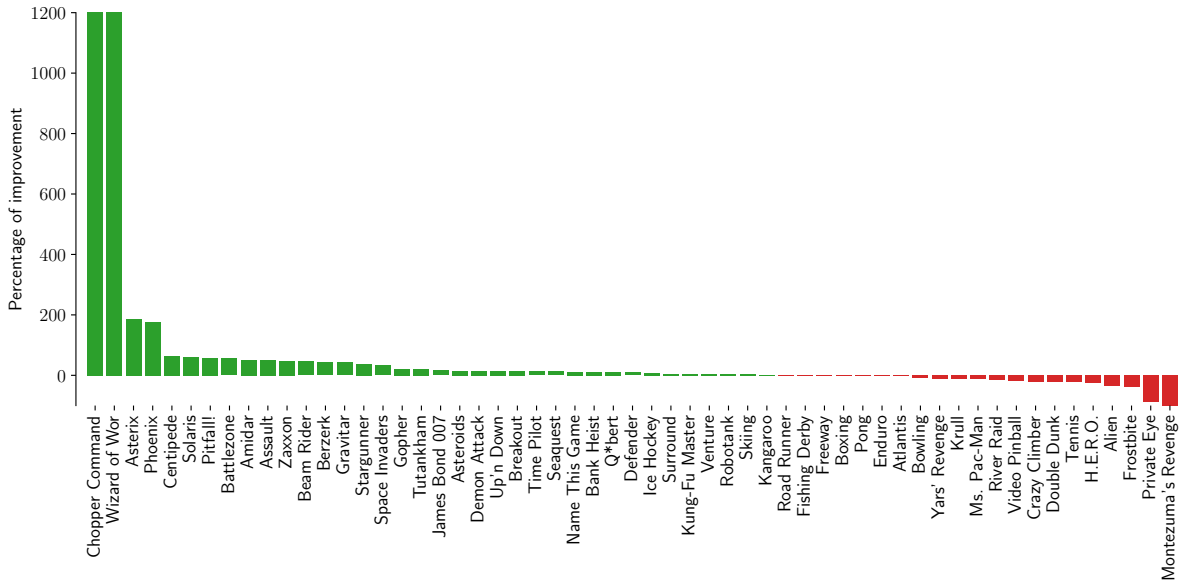

Figure B.10: Cumulated training improvement of UPER over QR-PER defined as $C_{\text{UPER/QR-PER}}$.

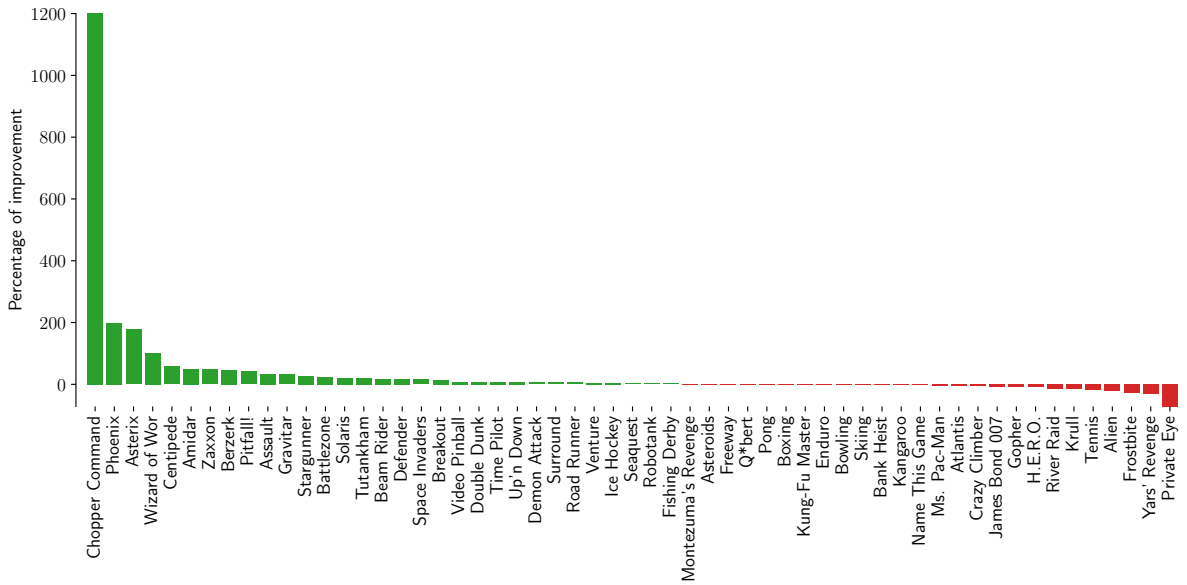

Figure B.11: Cumulated training improvement of UPER over QR-ENS-PER defined as $C_{\text{UPER/QR-ENS-PER}}$.

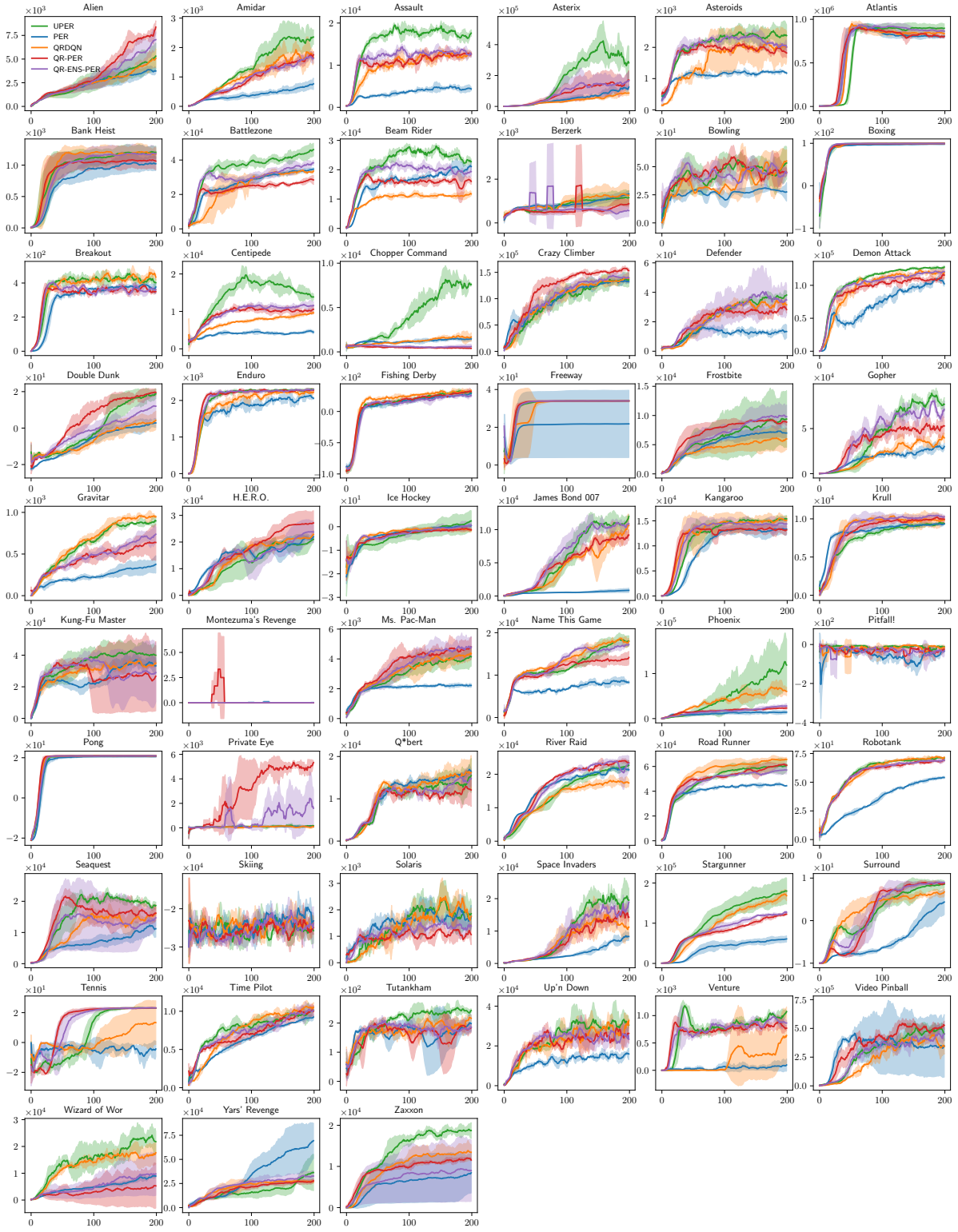

Figure B.12: Average performance and corresponding standard deviation for all games across 3 seeds.

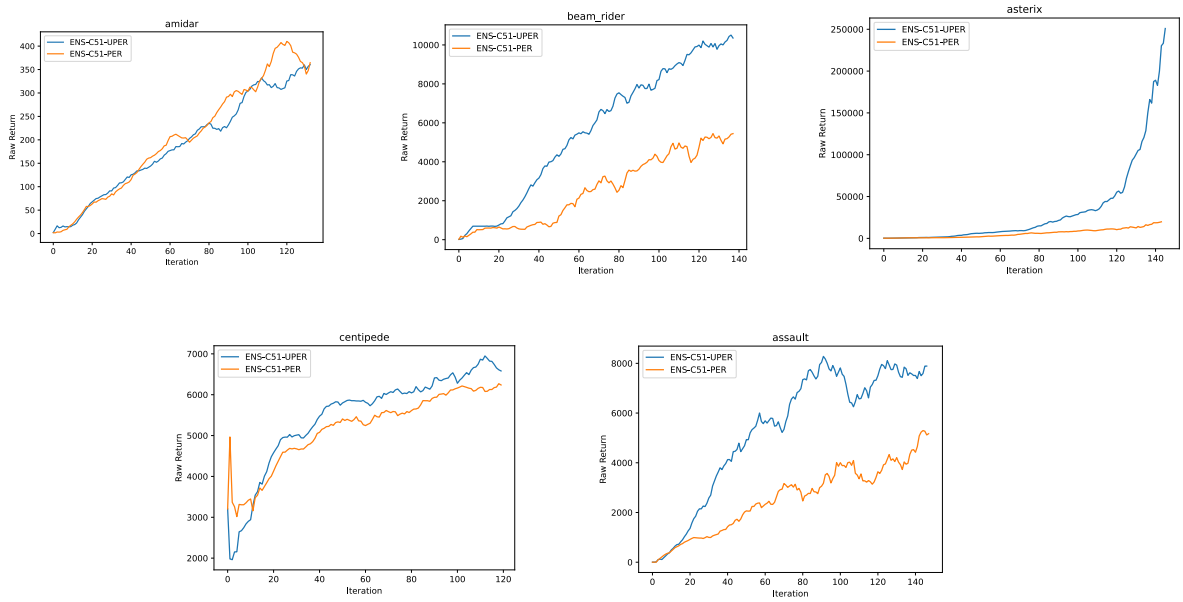

Figure B.13: Performance of an ensemble C51 agent with PER vs ensemble C51 agent with PER for 5 Atari games. Average across 2 seeds.

Appendix C

Homotopy Method solutions

C.1 Homotopy method

C.1.1 Mode Expressions

$$
\begin{aligned}
g_0 &= 0, \\
g_1 &= \frac{(T-t)(\xi^2 w - \mu_x)^2}{\beta}, \\
g_2 &= \frac{(T-t)^2(\xi^2 w - \mu_x)^2(-\xi^2\alpha + \log(\gamma)/2)}{\beta}, \\
g_3 &= \frac{(T-t)^3(\xi^2 w - \mu_x)^2(4\xi^4\alpha^2\beta - 4\xi^2\alpha\beta\log(\gamma) - 4\xi^2(\xi^2 w - \mu_x)^2 + \beta\log(\gamma)^2)}{6\beta^2}, \\
g_4 &= \frac{(T-t)^4(\xi^2 w - \mu_x)^2(-8\xi^6\alpha^3\beta + 12\xi^4\alpha^2\beta\log(\gamma) + 40\xi^4\alpha(\xi^2 w - \mu_x)^2 - 6\xi^2\alpha\beta\log(\gamma)^2 - 16\xi^2(\xi^2 w - \mu_x)^2\log(\gamma) + \beta\log(\gamma)^3)}{24\beta^2}, \\
&\vdots \\
g_i &= (T-t)^i(\xi^2 w - \mu_x)^2 \cdot \mathrm{coef}(i, t, \beta, \alpha, \mu_x, \xi^2, T, \gamma)
\end{aligned}
$$

C.1.2 Mode stability and Parameter Sweep

In this section, additional results are presented for the homotopy approximation without the Padé step, as explained in subsection 4.3.3. figure C.1 illustrates the divergence of

the homotopy modes for the same setup described in subsection 4.3.3.

figure C.2 displays the base homotopy modes without Padé, exemplifying a regime where the vanilla homotopy approximation is effective. This occurs specifically when base learning does not have enough time to converge and the optimal learning trajectory does not deviate significantly from the optimal one.

Convergence as modes are added is shown in figure C.3, and a hyperparameter sweep is depicted in figure C.4. Finally, figure C.5 demonstrates that certain relationships between the hyperparameters of the learning system and the task distribution influence the optimal control in a manner similar to that observed in the toy models.

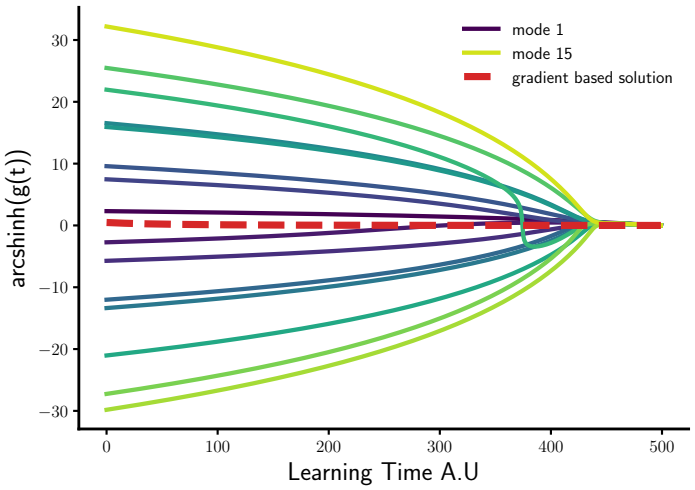

Figure C.1: Divergence of base homotopy modes without Pade approximation (arcsinh works as a log scale but including negative numbers).

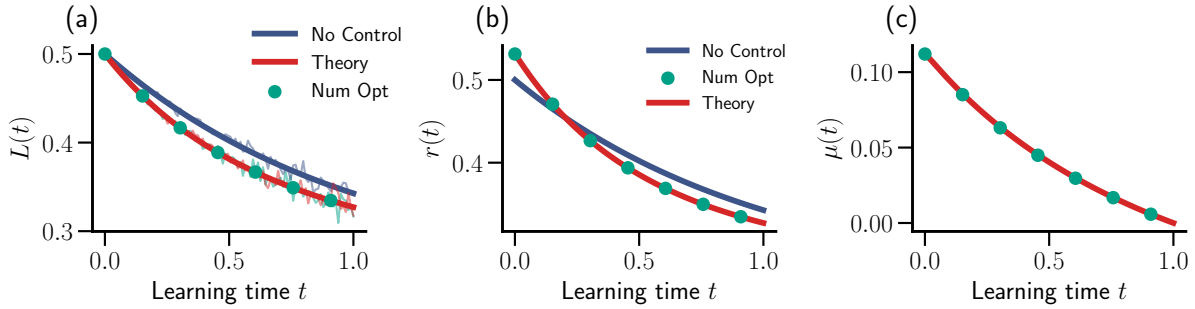

Figure C.2: Base Homotopy with no Pade Approximation for a simplified regime where learning has no converged. **(a)**: Loss function. **(b)**: Instant reward rate. **(c)**: Control signal (control called μ instead of g)

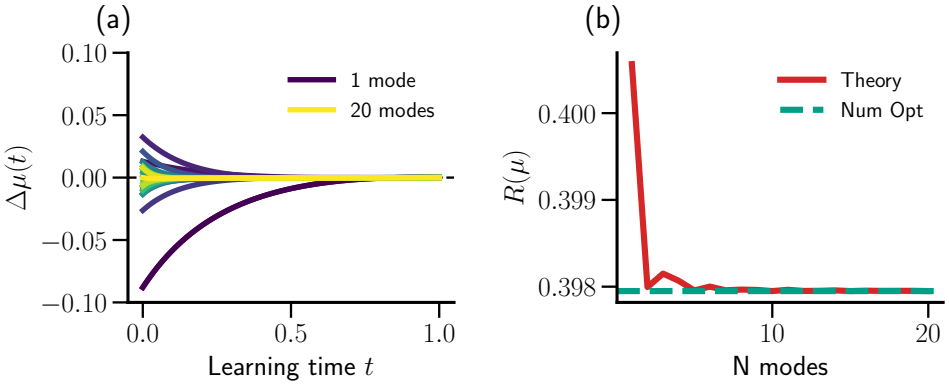

Figure C.3: Convergence of homotopy approximation as modes are included. **(a)**: Difference between analytical approximation and numerical control. (control called μ instead of g). **(b)**: Approximation of optimal risk as modes are included.

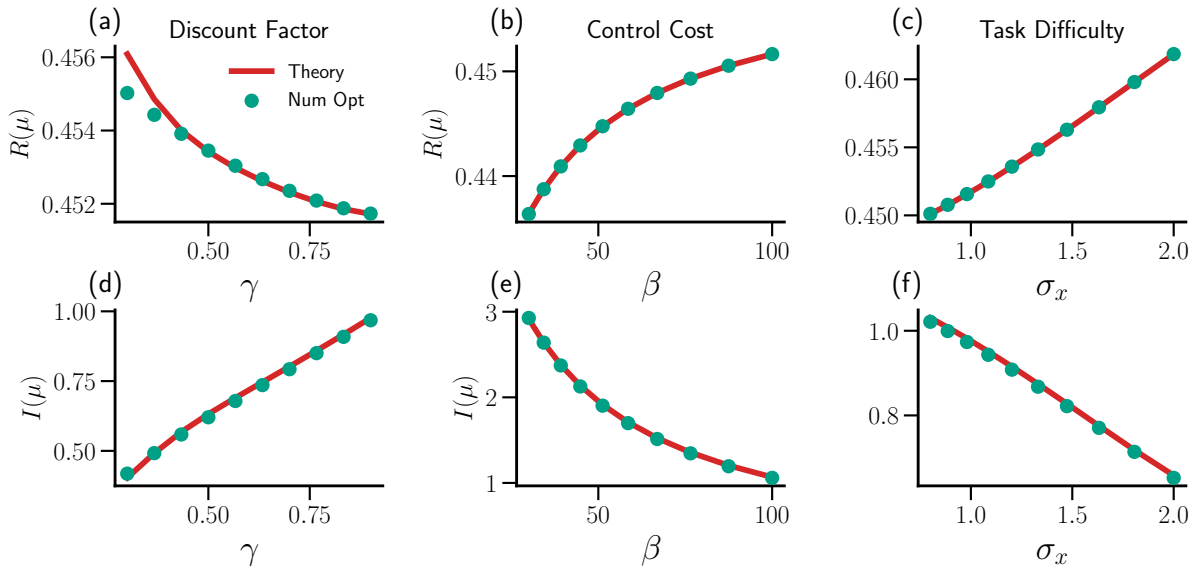

Figure C.4: Base Homotopy Parameter Sweep, similar to figure 4.7, except under a working regime where base learning does not enough time to converge.

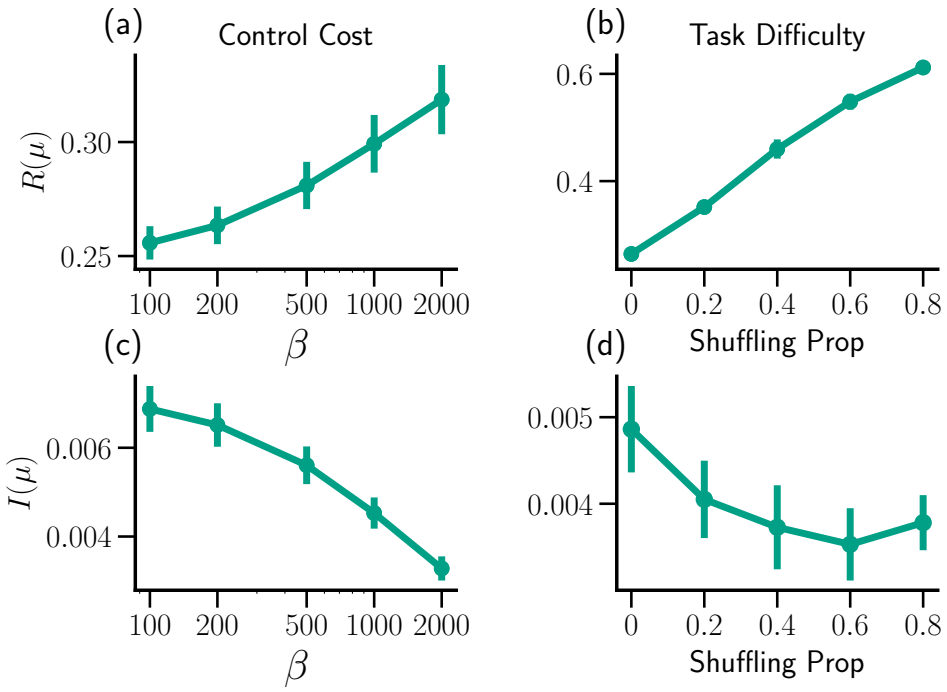

Figure C.5: Non-linear network simulation parameter sweep of learning rate control. Learning rate scheduling (here noted as μ) was optimized using the learning effort framework. $R(\mu)$ denoting cumulated risk, and $I(\mu)$ cumulated control as in figure 4.7. Task difficulty was controlled by randomly shuffling the labels in the training set, simulating the type of aleatoric noise that cannot be overcome by learning.

C.2 Padé Approximation

Padé[M=2/N=2]

```
\frac{\frac{4 \left(<x^{2}>\right)^{2} \alpha \left(- T + t\right)^{2} \left(<x^{2}> w - <xy>\right)^{4}}{\beta \left(8 \left(<x^{2}>\right)^{3} w^{2} - 16 \left(<x^{2}>\right)^{2} <xy> w + 4 \left(<x^{2}>\right)^{2} \alpha^{2} \beta + 8 <x^{2}> <xy>^{2} - 4 <x^{2}> \alpha \beta \log{\left(\gamma \right)} + \beta \log{\left(\gamma \right)}^{2}\right)} - \frac{\left(- T + t\right) \left(<x^{2}> w - <xy>\right)^{2}}{\beta}}{- \frac{\left(- T + t\right) \left(24 \left(<x^{2}>\right)^{4} \alpha w^{2} - 48 \left(<x^{2}>\right)^{3} <xy> \alpha w + 8 \left(<x^{2}>\right)^{3} \alpha^{3} \beta - 8 \left(<x^{2}>\right)^{3} w^{2} \log{\left(\gamma \right)} + 24 \left(<x^{2}>\right)^{2} <xy>^{2} \alpha + 16 \left(<x^{2}>\right)^{2} <xy> w \log{\left(\gamma \right)} - 12 \left(<x^{2}>\right)^{2} \alpha^{2} \beta \log{\left(\gamma \right)} - 8 <x^{2}> <xy>^{2} \log{\left(\gamma \right)} + 6 <x^{2}> \alpha \beta \log{\left(\gamma \right)}^{2} - \beta \log{\left(\gamma \right)}^{3}\right)}{2 \left(8 \left(<x^{2}>\right)^{3} w^{2} - 16 \left(<x^{2}>\right)^{2} <xy> w + 4 \left(<x^{2}>\right)^{2} \alpha^{2} \beta + 8 <x^{2}> <xy>^{2} - 4 <x^{2}> \alpha \beta \log{\left(\gamma \right)} + \beta \log{\left(\gamma \right)}^{2}\right)} + 1 + \frac{\left(- T + t\right)^{2} \left(64 \left(<x^{2}>\right)^{6} w^{4} - 256 \left(<x^{2}>\right)^{5} <xy> w^{3} + 112 \left(<x^{2}>\right)^{5} \alpha^{2} \beta w^{2} + 384 \left(<x^{2}>\right)^{4} <xy>^{2} w^{2} - 224 \left(<x^{2}>\right)^{4} <xy> \alpha^{2} \beta w + 16 \left(<x^{2}>\right)^{4} \alpha^{4} \beta^{2} - 88 \left(<x^{2}>\right)^{4} \alpha \beta w^{2} \log{\left(\gamma \right)} - 256 \left(<x^{2}>\right)^{3} <xy>^{3} w + 112 \left(<x^{2}>\right)^{3} <xy>^{2} \alpha^{2} \beta + 176 \left(<x^{2}>\right)^{3} <xy> \alpha \beta w \log{\left(\gamma \right)} - 32 \left(<x^{2}>\right)^{3} \alpha^{3} \beta^{2} \log{\left(\gamma \right)} + 16 \left(<x^{2}>\right)^{3} \beta w^{2} \log{\left(\gamma \right)}^{2} + 64 \left(<x^{2}>\right)^{2} <xy>^{4} - 88 \left(<x^{2}>\right)^{2} <xy>^{2} \alpha \beta \log{\left(\gamma \right)} - 32 \left(<x^{2}>\right)^{2} <xy> \beta w \log{\left(\gamma \right)}^{2} + 24 \left(<x^{2}>\right)^{2} \alpha^{2} \beta^{2} \log{\left(\gamma \right)}^{2} + 16 <x^{2}> <xy>^{2} \beta \log{\left(\gamma \right)}^{2} - 8 <x^{2}> \alpha \beta^{2} \log{\left(\gamma \right)}^{3} + \beta^{2} \log{\left(\gamma \right)}^{4}\right)}{12 \beta \left(8 \left(<x^{2}>\right)^{3} w^{2} - 16 \left(<x^{2}>\right)^{2} <xy> w + 4 \left(<x^{2}>\right)^{2} \alpha^{2} \beta + 8 <x^{2}> <xy>^{2} - 4 <x^{2}> \alpha \beta \log{\left(\gamma \right)} + \beta \log{\left(\gamma \right)}^{2}\right)}}
```