The University of Cape Town

Faculty of Science

Department of Mathematics and Applied Mathematics

February 2022

Applied Mathematics Masters Thesis

Submitted for the degree of Master of Science

Active Inference in Multi-Objective Dynamic Environments

by

Rowan Hodson

HDSROW001

Supervisors: Associate Professor Jonathan Shock, Dr Ryan Smith and Professor Mark Solms

Declaration of Authorship

I, Rowan Hodson, declare that this thesis titled *Active Inference in multi-objective dynamic environments*, and the work presented in it are my own. I confirm that:

- This work was done while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this has always been clearly attributed.
- Where I have used figures and/or diagrams from the work of others, the source is always given. With the exception of such figures/diagrams, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed by candidate

Signed:_____

Date: 18/02/2022

Abstract

Artificial Intelligence holds the promise of not only creating intelligent entities, but also unlocking the mysteries of our brains, and the nature of the subjective consciousness that accompanies them. Many paradigms of artificial intelligence are attempting to push the boundaries of the field, in order to catch a glimpse of the secrets behind general intelligence and the nature of the human mind. A less-explored, yet promising paradigm is that of Active Inference - a theory which details a first-principled explanation of how agents use action and perception to successfully operate within an external environment. Much work has been done to explore the framework's viability in modelling scenarios both related to neural process theory and more classical agent-based machine learning. However, due to the relative recency of the theory, there are still many areas of comparison and evaluation to explore. This dissertation aims to investigate Active Inference's algorithmic capacity to solve more complex decision-based environments. Specifically, with varying degrees of complexity, I make use of a dynamic environment with a multi-objective reward function to investigate the Active Inference agent's ability to learn and plan while balancing exploration and exploitation, and compare this to other Bayesian Machine Learning algorithms. In doing so, I investigate some novel approaches and additions to Active Inference's algorithmic structure which include a dynamic preference distribution, a two-tiered hierarchical approach to the state space (using model-free Reinforcement Learning to solve the lower level), and the introduction of the Propagated Parameter Belief Search algorithm - a modification to Active Inference which allows the agent to perform more complex counterfactual reasoning.

Acknowledgements

Here follows an expression of my gratitude to those who were significant figures in my life over the past two years.

Jonathan Shock - who has supported me now for many years - long before becoming my supervisor. You are, and always will be, a foundational figure in my academic life, and I hope we will continue working together in the years to come. Thank you for all the individual attention you have shown me during your time as my supervisor. You truly are a rare find.

Ryan Smith, who has been the best Active Inference mentor a boy could have ever ask for! A year ago, I hit an immense stroke of luck when you decided to take an interest in me and my work, and I am almost certain that much of what I have done to date would not have happened otherwise.

Mark Solms, who, over the last four years, has continuously put his faith in me and supported my academic trajectory in every possible way. You were the start of everything and I will never forget that. May your homeostatic imperatives remain forever (on average) fulfilled!

To Kyle Levin for providing a sense of company hard to find, despite being thousands of miles away for much of the time we've known each other.

To my parents who, without question, supported me, along a drastic and risky life-change. Many parents would not have done so, and I hope that one day you get to see the fully-realised result of that support.

Contents

1 Introduction

The theory of Active Inference is deeply-rooted in an amalgamation of preceding theories and concepts. In its essence, it provides both a theoretical and practical explanation as to how autonomous agents use perception and action to infer hidden states, minimise prediction error and thus reduce their surprisal and uncertainty with respect to these hidden states. The fundamental quantity which measures such surprisal and uncertainty is the variational Free Energy functional (Friston et al., 2006) around which, have formed large and intricate theoretical and practical frameworks. In their most fundamental form, these frameworks describe a process by which an agent, defined by the concept of a Markov Blanket (Kirchhoff et al., 2018), acts to better approximate the external state of the world (Friston et al., 2006), whether by changing its own internal state or by changing the state of its external environment. Based on this core concept, the Active Inference framework emerged, presenting a biologically plausible algorithmic approach to agent-based learning and planning. While the fundamental mathematical formulation of Active Inference is similar to other agent-based decision processes such as Bayes-Adaptive Learning (Duff and Barto, 2002) and Reinforcement Learning (Sutton and Barto, 2018), inherent to Active Inference's formalisation is a unique approach to the exploration versus exploitation trade-off, which is an ongoing topic of research in many areas of machine learning (Berger-Tal et al., 2014). This, along with its basis in biological plausibility (Friston, 2013; Colombo and Wright, 2021), makes Active Inference an attractive lens through which to approach Artificial Intelligence research.

Over the past few years, active Inference has solidified itself as a notable mechanism by which to implement intelligent agents to solve complex tasks in both fully and partially observable environments (Sajid et al., 2021; Friston et al., 2021; Fountas et al., 2020; Millidge, 2021; Hesp et al., 2021a). In addition to its attractive quality of biological plausibility, the framework comes equipped with naturally-derived state and parameter exploration schemes. This is in contrast to many other machine learning methods, which require the explicit addition of such schemes in order to achieve exploratory behavior. Despite much work being done to articulately analyse and develop the field, there remain many areas to explore in order to advance our understanding of its viability and capacity as a machine-learning paradigm. Particularly, theoretical and investigative analysis is needed to ascertain how it compares to other algorithms, particularly when operating in a partially-observable environment under model uncertainty. Historically, Bayes-adaptive Reinforcement Learning methods (Duff and Barto, 2002; Ross et al., 2007) have been used to navigate such environments, and, as of yet, no in-depth comparison of these methods to Active Inference has been done. In addition to this, the Active Inference algorithm presents a framework with great potential for expansion, both with respect to its integration with other machine learning techniques, and in terms of expanding upon the core ideas behind its use of belief propagation and inference.

Inline with its quality of biological plausibility (Friston, 2013), the framework naturally lends itself to effectively representing scenarios of *living* entities navigating an environment representative of some real-world feature. In this regard, the possibilities are endless, and significant opportunity exists to explore the application of the Active Inference framework to such 'real-world' models.

The work presented in this dissertation aims to investigate some of these opportunities and flesh out our understanding of the Active Inference algorithm. In particular, we analyse how it compares to other similar model-based Bayesian methods, the potential ways in which it can be integrated with other machine learning techniques and ways which the algorithm, particularly the Sophisticated Inference algorithm (Friston et al., 2021) can be extended to enhance agent behavior, in a manner that remains congruent with the core concepts of inference and belief propagation. While we cannot claim that the investigation herein offers completely conclusive answers to theses factors mentioned above, we hope that it provides an iterative progression to the field of Active Inference, and acts as a bedrock upon which further research can be undertaken.

1.1 Research Questions

The aims of this dissertation can be articulated by four research questions:

1. How do Active Inference and Sophisticated Inference compare to Bayesian Reinforcement Learning in a partially observable, dynamic, context-dependent environment?

2. To what extent, and in what manner, do the *epistemic* and *novelty* terms affect agent behavior in a complex dynamic environment?

3. Can model-free Reinforcement Learning be integrated with Active Inference in ways that offer both biological plausibility and computational efficiency?

4. Can the belief propagation mechanism of Sophisticated Inference be applied to novelty to create a complex nested belief structure which encourages more intelligent behavior?

1.2 Overview

Section 2 acts as the literature review for this dissertation and explores the background concepts and theories which act as the foundations of Active Inference. To start, we summarise the methods of Bayesian Inference in Sections 2.1 and 2.2, with a focus on variational inference, which acts as the mathematical foundation of Active Inference. Following this, Sections 2.3 and 2.4 define two fundamental structures - the MDP and the POMDP, both of which act as the core structures upon which decision-based algorithms are created. Section 2.5 gives a brief overview of several Reinforcement Learning methods, focusing on Bayesian Reinforcement Learning, due to its prevalence in the practical work of this thesis. Subsequently, we unpack and review predictive coding and detail how it relates to concepts of Variational Inference and variational Free Energy in section 2.6. This section serves as the foundation upon which we build an intuition, derivation and analysis of the Free Energy Principle and its basis in a first-principled account of the emergence of biological systems (Friston et al., 2006). Following this, Section 2.8 formally presents Active Inference and the Expected Free Energy functional (Friston et al., 2015; Parr and Friston, 2019), detailing its various decompositions and the concepts behind the terms which constitute them. A specific focus of this section is on the latest developments of Active Inference, many of which aim to bring its scalability more inline with other artificial intelligence agents. As part of this discussion, we present Sophisticated Inference - describing its differences to

the original Active Inference algorithm and its unique feature of forming beliefs about how its actions might influence its future beliefs. We end Section 2 with a brief discussion of how subjective feeling (or affect) can potentially be represented by an Active Inference agent.

Section 3 describes our approach of implementing and testing several algorithms in two different environments. Specifically, we detail four different testing iterations which each test a different set of configurations of the Active Inference agents and the environment - the aim being to compare Active Inference and Bayesian Reinforcement Learning algorithms, as well as analyse the effect of removing or including the epistemic imperative from the Active Inference Agents in environments where there is model uncertainty. In addition to this, we describe the formulation and comparative testing of a new algorithm, Propagated Parameter Belief search, which extends the 'sophisticated' aspect of Sophisticated Inference by including the concept of parameter novelty in the agent's nested beliefs about beliefs.

Section 4 displays the results of these four testing iterations and in Section 5 we discuss the causes and meaning of these results, the reasons for differing agent behavior, and the reasons behind, and implications of, the PPBS algorithm's behavior.

Finally, Section 6 concludes this dissertation. Here we address each of the four core research questions, discuss to what extent they were answered, and what those answers mean for the field of Active Inference. In the process of this We describe possible future work which could potentially validate and extend the findings and hypotheses presented in this thesis.

2 A review and analysis of background material

2.1 Bayesian Inference

Bayesian inference is an approach to statistical modelling which is characterised by a prior distribution over some hypothesis space. When (observed) data (D) is received, this, combined with the prior, is used to calculate a posterior (H) over said hypothesis space. Bayes' theorem encapsulates this process, combining the likelihood of the data, the prior, and a normalising constant.

$$p(H|D) = \frac{p(D|H)p(H)}{p(D)} \tag{1}$$

In particular, Bayesian Inference holds two key characteristics which are relevant to the topic of this dissertation: Its ability to represent and update uncertainty over the parameters of a model and, in turn, its usefulness in representing the modelling of beliefs about the causes/states (hypotheses) of observations (data).

As an example of this belief-based approach to hidden causes of observations, let us imagine a very simple organism which can only detect the size of vague shapes in its environment. From these observations, it needs to infer to what extent the size of a shape indicates danger (let us imagine that it represents this degree of danger in numerical fashion, with lower numbers meaning less dangerous). The organism has a prior belief over the probability of the degree of danger in the environment: $p(s)\ ,\ s \in \mathcal{R}$, as well as a model of the probability of an observation conditioned on a degree of danger: $p(o|s)\ ,\ o \in \mathbb{R}$. This construction can be described as a generative model, which the agent/organism uses to infer a posterior belief over unknown states (degree of danger). Thus, when an organism receives an observation (vague shape), this inference can naturally be represented by Bayes' theorem.

$$p(state|observation) = \frac{p(observation|state)p(state)}{p(observation)} \quad \forall states \tag{2}$$

In addition to this concept, Bayes' theorem can also be used to represent inference, and, by extension, learning of model parameters. For example, let ϕ be a value or vector representing the parameters of the likelihood model. Then,

$$p(\phi|o) = \frac{p(o|s,\phi)p(s)}{p(o|\phi)} \quad \forall s \forall \phi \tag{3}$$

2.2 Variational Inference

While Bayes theorem offers a convenient way to perform inference, a problem often emerges when evaluating the denominator of the equation - sometimes referred to as the model evidence or the marginal likelihood. Using states and observations as our variables, the calculation of this marginalisation requires a summation over all possible states in the joint density:

$$p(o) = \int p(o,s)\ ds \tag{4}$$

when dealing with complex distributions, this integral is often intractable and so approximation methods are required to determine the posterior, $p(s|o)$. Variational inference is one such method of approximation and functions by converting an inference problem into an optimisation problem (Bishop, 2006; Blei et al., 2016). To achieve this, the method introduces a family of approximate distributions, $q(s) \in Q$, with the objective being to find the specific q which best approximates the true posterior. Crucial to variational inference is the concept of the Kullback-Liebler divergence (Kullback and Leibler, 1951) which measures the asymmetric similarity between two probability distributions. The KL-divergence between two distributions is at a minimum when the two distributions are identical.

$$D_{kl}(q||p) = \mathbb{E}_q \left[\ln \frac{q(s)}{p(s)} \right] \geq 0 \tag{5}$$

With this set-up, we can express the following optimisation problem:

$$q^*(s) = \underset{q(s) \in Q}{\arg\min}\, \mathbb{E}_q \left[\ln \frac{q(s)}{p(s|x)} \right] \tag{6}$$

Deriving an expression which we can use to minimise the KL-divergence between the approximate posterior and the true posterior requires a few steps. Starting with our original terms used for Bayesian inference:

$$p(o, s) = p(s|o)p(o)$$
$$\ln p(o) = \ln p(o, s) - \ln p(s|o)$$

and, for any $q(s) \in Q$:

$$\begin{aligned}
q(s) \ln p(o) &= q(s) \ln p(o, s) - q(s) \ln p(s|o) \\
\int q(s) \ln p(o)\, ds &= \int q(s) \ln p(o, s)\, ds - \int q(s) \ln p(s|o)\, ds \\
\ln p(o) &= \int q(s) \ln p(o, s)\, ds - \int q(s) \ln p(s|o)\, ds - \int q(s) \ln q(s)\, ds + \int q(s) \ln q(s)\, ds \\
\ln p(o) &= \underbrace{\int q(s) \ln \frac{p(o, s)}{q(s)}\, ds}_{\text{ELBO}} + \int q(s) \ln \frac{q(s)}{p(s|o)}\, ds
\end{aligned} \tag{7}$$

Notice that the third term in the last line in Equation 1 is the KL-divergence between the approximating distribution $q(s)$ and the true posterior $p(s|o)$. The second term is known in machine learning literature as the Evidence Lower Bound (ELBO) (Bishop, 2006). Its name derives from the fact that because the KL-divergence between the approximate distribution and the true posterior (third term) is always greater or equal to zero, the ELBO is always less than or equal to $\ln p(o)$, the model evidence, thus acting as a lower bound to it. Because $\ln p(o)$ does not depend on $q(s)$, maximising the ELBO (second term) with respect to $q(s)$ minimises the KL-divergence between the approximate and true posterior (third term). This implicitly equates to finding a member of the family Q which best approximates said true posterior. As discussed later on, the Evidence Lower Bound is an important concept in the formalisation of the Free Energy Principle, and underpins its mathematical structure.

An important assumption that is often made in Variational Inference is the independence of latent variables (in our example, states). Realistically, the hidden state s can be decomposed into a set of latent variables,

$$\mathbf{s} = s_1, ..., s_m$$

We then make the assumption that q factorises with respect to this partition,

$$q(\mathbf{s}) = \prod_{i=1}^{M} q_i(s_i) \tag{8}$$

This concept of factorisation has its basis in Mean Field Theory (Parisi, 1988) and, as we will see, this simplifying assumption is another important factor in the theory of the Free Energy Principle, offering additional tractability to the minimisation of Variational Free Energy.

2.3 Markov Decision Process

A Markov Decision process (MDP) (Bellman, 1958) is a mathematical framework to model sequential decision-making. Its dynamics can be mathematically expressed by the tuple:

$$(S, A, R, T, \gamma)$$

- S is a set of states (assumed to be finite)
- $A(s)$ is a finite set of actions which can be taken in state $s \in S$
- $R(s, a, s') = p(r|s, a, s)$ is the reward function defining the reward received upon taking action a in state s and transitioning to state s'.
- $T(s'|s, a) = p(s'|s, a)$ is a function which models the Transition dynamics between states as a function of action.
- $\gamma \in [0, 1]$ is a discount factor which determines the comparative present weighting of future rewards.

At each discrete time-step t, an agent receives a signal representing what state it is in, $s_t \in S$ and proceeds to select an action, $a_t \in A(s_t)$. In the next time-step, $t + 1$, the agent transitions to a new state s_{t+1} and receives a reward $r_{t+1} \in \mathbb{R}$. Action selection in a MPD is determined by a policy $\pi(s)$ which is a functional mapping between states and actions: $\pi : S \times A \to [0, 1]$.
A fundamental property of MDPs is their natural ability to construct recurrence relations. From this, it is evident that the information available at each time-step encapsulates the information from all previous time-steps. Implicit in this is the assumption that all future time-steps only depend on the current state. This property can be formally defined as,

$$p(s_{t+1}, r_{t+1}|s_0, a_0, ..., s_{t-1}, a_{t-1}, s_t, a_t) = p(s_{t+1}, r_{t+1}|s_t, a_t) \tag{9}$$

The objective function of an MDP for any given time-step can naturally be defined in terms of an expectation of the reward accumulated over some time-step horizon (potentially infinite),

$$G_t = \mathbb{E}\left[\sum_{k=0}^{T} \gamma^t r_{t+k+1})\right] \tag{10}$$

More specifically, the expected return from any given state at any timestep, t, for a certain policy, can be better articulated via a recursive state-value function.

$$\upsilon_\pi(s) = \mathbb{E}_\pi\left[G_t | S_t = s\right] = \sum_{r,s,a} p(r|s,a,s')p(s'|a,s)\pi(a|s)\left[r + \gamma\upsilon_\pi(s')\right] \tag{11}$$

Thus, the *value* of a state s in this context, under a specific policy π, is the expected cumulative reward when starting in state s and following π onward.
Optimising the objective function of an MDP from a given initial state can therefore be expressed as finding a policy (there may be more than one) which maximises the value function of this state,

$$G_t^*(s) = \arg\max_\pi \upsilon_\pi(s) \tag{12}$$

2.4 Partially Observable Markov Decision Process

An assumption which is made in the Markov Decision Process presented above, is that the signal provided by the environment at each timestep has a one-to-one mapping with some state. However when modelling many real-world problems this assumption does not hold. Often the signal is noisy, incomplete, or has probabilistic mappings to many different states. The issue of decoding a noisy signal channel in a Markov Process was first introduced by Drake (Drake, 1962), and later officially expanded to the concept of a Hidden Markov Process (Baum and Petrie, 1966). It is worth noting here that a Markov Process or Hidden Markov Process is similar to a Markov *Decision* Process, but does not have an entity that makes actions. Rather it is an autonomous process that evolves irrespective of some external input. To make the connection between the Markov *decision* Process and the Hidden Markov Process, Sondik (Sondik, 1971) presented his thesis on optimal control of partially observable Markov Processes. From this emerged the framework of the Partially Observable Markov Decision Process (POMDP).
Unsurprisingly, POMDPs share much of the same formulation as MDPs, with the difference being the introduction of *observations* (as well as *states*) and a distribution which serves to probabilistically map states to observations. The tuple, presented in the section above, then becomes:

$$(S, A, R, T, \gamma, Z, \mathcal{O})$$

- Z is the finite set of observations
- $\mathcal{O} = p(z|s) \;\; z \in Z, \; s \in S$

Thus, the POMDP can be viewed as a generalisation of the MDP, with the MDP being the special case where there is a bijection between the observation and state sets.

In practice, determining a value function for a POMDP, as we did with an MPD, is not so straightforward. This stems from the fact that an agent interacting with a POMDP does not (for the most part) deterministically know which state it is in. How then could an agent be able to implement an accurate policy if action choices are conditioned on states?

A way to circumvent this problem is for an agent to occupy a belief state, $b \in B$ where this belief specifies the probability of being in each state at a given timestep. The value function can thus be formulated as a function of this belief state, rather than the set of true states.

$$\upsilon_\pi(b) = \sum_{r,b,a} p(r|b,a,s')p(s'|a,b)\pi(a|b)\left[r + \gamma\upsilon_\pi(b')\right] \tag{13}$$

2.5 Reinforcement Learning

Reinforcement Learning (Sutton and Barto, 2018) is a set of algorithmic techniques and frameworks for solving partial and fully-observable Markov Decision Processes. Central to Reinforcement Learning is a decision-making entity - an agent which receives numerical rewards, whether positive or negative, over its discrete timestep trajectory. From this reward-based feedback, the agent learns to value certain states and/or actions in those states.

Although a Reinforcement Learning agent learns via feedback, it is not actually a true form of supervised learning, nor is it a type of unsupervised learning (Sutton and Barto, 2018). It's learning cannot be considered *supervised* because it doesn't explicitly receive 'guiding' examples as to which states or state/action pairs are positive or negative. Rather it must rely on its own experience of a feedback signal which it uses to construct its own evaluation of the MDP dynamics.

It might be tempting then to see Reinforcement Learning as a type of unsupervised learning, a process whereby the hidden structure of data is modelled. However, while this might form part of what the agent achieves, this description does not fully cover the main objective of a Reinforcement Learning agent, which is simply to maximise a cumulative reward signal.

Throughout this review, I will explicitly be focusing on value function formalisations of Reinforcement Learning, rather than policy gradient methods. Although this latter set of techniques makes up some of the more cutting-edge Reinforcement Learning research today, I do not involve them in the practical work of this dissertation.

Reinforcement Learning can generally be divided into two types: Model-free and Model-based. These categories, or some combination thereof, encompass all formulations and algorithmic design of the field. The practical work of this dissertation makes use of these two categories, and so here I present a review of the main concepts and practices inherent to both.

2.5.1 Model-free Reinforcement Learning

In the discussion on MDPs, it was implicitly assumed that an agent navigating such a setup makes use of its knowledge of the MDP dynamics to be able to predict state-transitions and rewards at future time-steps. However, in many problem types, this setup is unrealistic and computationally complex. The model-free paradigm of reinforcement learning posits a framework which foregoes the requirement

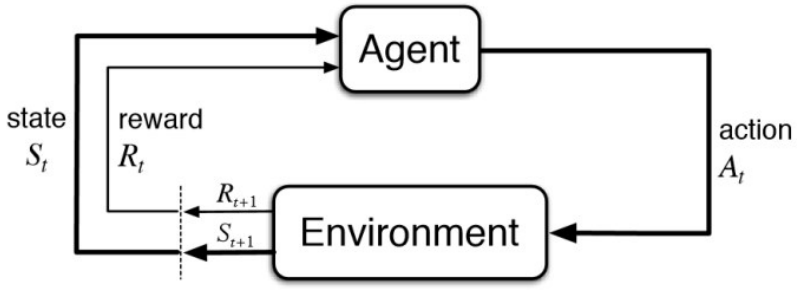

Figure 1: (Sutton and Barto, 2018) Reinforcement Learning flow, defined by an agent that takes action A_t in state S_t, transitions to state S_{t+1} and receives reward R_{t+1}.

that the agent has such prior knowledge. However, the problem remains that an agent must still learn the implicit or explicit value of states and policies in order to 'successfully' operate within an MDP environment The simplest version of such a Model-free approach is pure trajectory sampling. Given that an agent knows which state it is currently in, it can perform online 'rollouts' of certain action combinations defined by some policy. At the end of such a trajectory, the agent propagates the cumulative discounted reward it received over the course of the sampling trajectory back to states it visited along the way. This can be done many times, taking the average of the sampled trajectories to estimate the value functions of states under a policy.

This type of approach is known as a Monte Carlo method (Metropolis and Ulam, 1949), which is a set of statistical estimation techniques which require only sampled experience to formulate a hypothesis over data.

When implementing Monte Carlo methods in an MPD, it is particularly useful to estimate the value of a state-action pair, rather than just the value of a state. The value of a state alone is sufficient in scenarios where the agent knows the model dynamics, as it can then effectively look ahead and choose which action leads to the greatest expected reward, given its knowledge of the distribution over states it can transition to. However, if the agent cannot predict the distribution over next states, it loses its ability to look ahead and predict such expectations. Defining value in terms of states and actions solves this dependency on model knowledge. Although the agent cannot predict which potential states it will transition to given an action, it can take some action in a given state, receive a reward signal, and so attribute value to that state-action pair. The value function defined in Section 2.3, thus becomes a state-action value function,

$$q_\pi(s, a)$$

One way to achieve optimal control in this setting is to use an iterative approach of greedy policy selection, whereby the agent deterministically selects the action with maximum estimated value,

$$\pi(s) = \arg\max_a q(s, a) \tag{14}$$

Due to the policy improvement theorem (Sutton and Barto, 2018) this approach is guaranteed to converge to an optimal policy given that each state-action pair is visited an infinite number of times.

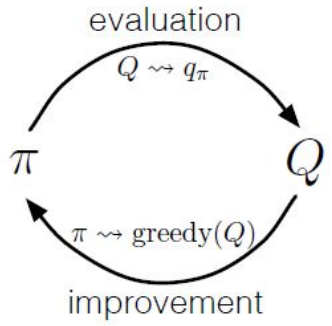

Figure 2: (Sutton and Barto, 2018) Policy improvement theorem.

This proof of policy improvement can be formulated via greedy policy selection:

$$\begin{aligned} q_{\pi_t}(s, \pi_{t+1}(s)) &= q_{\pi_t}(s, \arg\max_a q_{\pi_t}(s, a)) \\ &= \max_a q_{\pi_t}(s, a) \\ &\geq q_{\pi_k}(s, \pi_t(s)) \end{aligned} \tag{15}$$

Algorithmically, optimal policy convergence is assured by alternating between policy evaluation, which involves Monte Carlo trajectory sampling to estimate the value of $q(s, a)$, and policy iteration, which can be achieved via some greedy (or ϵ-greedy) policy improvement. This cycle is shown in figure 2.

2.5.2 Q-Learning

A crucial extension to model-free Reinforcement Learning was the idea of truncating sampling trajectories with 'bootstrapped' value functions. This technique is broadly known as Temporal-Difference (TD) Learning and, similar to the recursive property of Bellman's equation as shown in Section 2.3, involves using the value function of some future state to encompass the rest of the cumulative reward along a trajectory. In non-TD methods, updating a value function involves evaluating a full return, and incrementally moving the current iteration of the value function in the direction this return:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[G_t - Q(S_t, A_t) \right] \tag{16}$$

where G_t is the cumulative reward along a trajectory and $\alpha \in [0, 1]$ is the step-size parameter. In TD implementations, this full cumulative reward is replaced by 'true' reward up until some number of steps, followed by a boostrapped value function. For a one-step TD return this can be formulated as,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right] \tag{17}$$

Although there are many forms of Temporal-Difference Learning, here I will focus on a sub-category, known as Q-Learning, as this pertains most to the methodology of this dissertation. Q-Learning is a one-step (though it can be generalised to n-steps) *off-policy* form of TD control. It is described as an off-policy method, as the policy the agent learns does not necessarily rely on the policy actually

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

Figure 3: (Sutton and Barto, 2018) Q-Learning algorithm.

implemented by the agent. The update equation for one-step Q-learning is defined as:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right] \tag{18}$$

Importantly, although the state-action value function is updated based on greedy action evaluation, the agent does not necessarily choose this greedy action. Thus, the Q-Learning algorithm is considered off-policy, as shown in Figure 3. In general, model-free Reinforcement Learning carries with it the advantage of not having to learn a model - which can be intractable, computationally expensive and memory inefficient.

2.5.3 Model-Based Reinforcement Learning

The second pillar of Reinforcement Learning and the paradigm which is most generally aligned with solving MDPs is model-based techniques, where an agent utilises its knowledge of the environment to construct estimate value functions and plan rewarding trajectories. As we will see, much of the concepts presented in this dissertation are inherently model-based, with an agent using its model of the environment to generate predictions and plan trajectories.

Perhaps the most fundamental theoretical form of model-based Reinforcement Learning presents itself in Dynamic Programming which is a collection of algorithms used to determine optimal policies in an MDP given perfect knowledge of the environment (Sutton and Barto, 2018). Similar to the classical optimisation setup shown in Section 2.3, policies are evaluated in Dynamic Programming via determining the value of a state through a recursive update scheme. This update for a state-value function, under a given policy, is given by:

$$v_{t+1}(s) = \sum_{a,s',r} \pi(s|a) p(s', r|s, a) \left[r + \gamma v_t(s')\right] \quad \forall s \in S \tag{19}$$

In the context of the policy improvement theorem, this represents the evaluation step, with the improvement step defined by a greedy update to the policy function:

$$\pi_{t+1}(s) = \arg\max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_t(s')\right] \quad \forall s \in S \tag{20}$$

There are many slight variations to this general concept of policy evaluation and improvement, and it is important to note that they all involve offline 'update sweeps' of the states in the environments, given a full *distribution model* (Sutton and Barto, 2018), rather than using only online trajectories as discussed in the section above.

Another method which can be used in model-based approaches is a *sample model*, which, conditioned on an action, returns a random sample of a next reward and state to the agent, rather than a full distribution over states and rewards. Although this might seem similar to a Monte Carlo approach, the crucial difference is that this is an *offline* technique which involves no actual interaction with the environment. Interesting factors arise when planning is done *online*, amidst actual interaction with the environment. In this scenario, the model is constantly being updated, and computational resources must be split between decision making and model learning (Sutton and Barto, 2018).

An interesting technique, which model-based Reinforcement Learning (and all model-based methods in general) can implement, is the process of data generation as a means to augment a training data set. An original form of this appeared in Sutton's Dyna algorithm (Sutton, 1991), which combined both policy optimisation and a predictive model - which generated learning data. This central idea has been used in foundational works (Krizhevsky et al., 2017) and has solidified itself as a crucial element of model-based schemes (Ha and Schmidhuber, 2018; Kurutach et al., 2018; Clavera et al., 2018). Importantly, the mechanism and general concept is entirely analogous to that of the *generative model* which is a core theme of this dissertation.

2.5.4 Bayesian Reinforcement Learning

Much investigation into Bayesian machine learning methods has been conducted to date, with many such methods emerging as effective ways to incorporate prior information into a problem, in order to perform inference over unknown variables (Ghavamzadeh et al., 2015). Naturally, Bayesian machine learning is applied to problems involving *uncertainty*, where new information is combined with prior beliefs to formulate a new posterior belief about some unknown factor(s). In particular, Bayesian methods have been shown to be an effective paradigm with which to approach the navigation of **POMDPs** (Poupart and Vlassis, 2008).

Bayesian Reinforcement Learning is a paradigm where Bayesian methods are used to either frame the problem with respect to uncertainty over the *solution-space* (model-free), or uncertainty over the *parameter-space* (model-based). A significant advantage of framing such problems in a Bayesian way is that it effectively side-steps the issue of exploration vs. exploitation. This is due to the fact that Bayesian methods have the capability to represent uncertainty over states/parameters/solutions as *belief* states. Given, and with respect to, these belief states, optimal solutions can be found. (Ghavamzadeh et al., 2015). The downside of such an approach is its sensitivity to the initial prior information incorporated into the system, as all belief states are initially entirely based upon this prior (Guez et al., 2012). Thus, an integral, and often difficult, aspect of Bayesian Reinforcement Learning

is the design and incorporation of effective prior information.

Model-based Bayesian RL offers a particularly interesting approach to modelling uncertainty over parameters. As an example, given a setup where the transition model, $p(s'|s,a,\theta)$ is unknown , with θ being the parameters of this transition model, the Bayesian agent can represent this uncertainty with respect to its *beliefs* about θ.

Given that $b \in B$ where $b(\theta) = p(\theta)$, the transition model becomes:

$$p(s'|s,b,a) = \int_{\theta} p(s'|s,a,\theta)b(\theta)d\theta \tag{21}$$

Here the expectation of theta with respect to belief b has been taken (it has been integrated out), and so θ does not appear in the resulting probability density. Thus the model is effectively *known* with respect to belief b, and exploration of θ is not necessary. Beliefs themselves are updated upon receiving data (in this case data about about transitions):

$$b' = b(\theta|s,a,s') \tag{22}$$

With the model being framed as known, with respect to b, the problem can be formulated as a Markov Decision Process, and Bellman's equation can be used to determine the optimal value function for each state/belief pair.

$$\upsilon^*(s,b) = \arg\max_a \sum_{s',r} p(s',r|s,a,b)\left[r + \gamma\upsilon(s',b^{s,a,s'})\right] \quad \forall s \in S \tag{23}$$

We note here that Equation 10 is a specific form of the general value function shown in Equation 3. This is not unexpected, as a POMDP can be formulated as *belief* MDP, which is exactly how Bayesian RL problems are constructed - the difference being, that not all Bayesian RL problems are partially observed.

Due to the fact that model-based Bayesian RL can be construed as an MDP, it can algorithmically be constructed as any normal model-based Reinforcement Learning scheme, with all elements discussed in Section 2.5.3 being applicable to ways in which this can be implemented.

While Bayesian methods offer a principled approach to the exploration/exploitation dilemma, via the construction of a belief MDP, issues arise upon the introduction of both model uncertainty and a partial observability (Katt et al., 2018). In light of these issues, the Bayes-adaptive POMDP framework emerged (Ross et al., 2007). As the name suggests, this set of methods uses the foundation of the Bayes-adaptive approach for MDPs (Duff and Barto, 2002) and extends its functionality to the partially observed domain, by allowing the agent to model its own uncertainty over its model of the environment dynamics. When navigating a fully observable MDP, an agent can learn by registering and storing the number of times it witnesses specific environment dynamics, upon interacting with said environment. As is implied in Equation 9, this could take the form of an agent increasing its belief that some state, s transitions to another state s' upon observing this specific transition actually happening, with its confidence in this transition belief increasing the more such observations it makes. This increase of belief about the transition can be represented by incrementing a count, $\phi_{sa}^{s'}$, with the actual belief over parameters being a Dirichlet distribution which uses such counts as its concentration parameters.

However, when operating within a POMDP, the agent does not fully observe the state-space and thus, in many cases, has uncertainty as to what transitions between states actually takes place (implicitly due to its uncertainty over $\mathcal{O}$ as presented in Section 2.4). This creates a scenario where learning is difficult, due to agents in POMDPs often having inaccurate beliefs about the environmental dynamics that they sample. To take this uncertainty into account, the BAPOMDP framework incorporates the agent's beliefs over model parameters into the hidden state, forming a *hyper-state* space, $S^* = \langle S, T, \mathcal{O} \rangle$, with the state-transition and state-observation counts given by $\phi_{sa}^{s'}$ and θ_{sa}^{z} respectively. Thus, the space of T and $\mathcal{O}$ is formally defined as:

$$T = \{\phi \in N^{|S|^2|A}|\forall(s,a) \in S \times A, \sum_{s' \in S} \phi_{sa}^{s'}\}$$
$$\mathcal{O} = \{\theta \in N^{|S|A|Z}|\forall(s,a) \in S \times A, \sum_{z \in Z} \theta_{sa}^{z}\}$$

Therefore, given the definitions $T_{\phi}^{sas'} = \frac{\phi_{sa}^{s'}}{\sum_{s \neq s'} \phi_{sa}^{s'}}$ and $\mathcal{O}_{\phi}^{s'az} = \frac{\theta_{sa}^{z}}{\sum_{z' \neq s'} \phi_{sa}^{z'}}$, the environment dynamic probabilities are given as:

$$p(s', \phi', \theta', z|s, \phi, \theta, a) = T_{\phi}^{sas'} \mathcal{O}_{\theta}^{s'az} I_{\{\phi'\}}(\mathcal{U}(\phi, s, a, s')) I_{\{\theta'\}}(\mathcal{U}(\theta, s', a, z)) \tag{24}$$

where $\mathcal{U}$ is a function which increases the count of ϕ and θ upon the agent receiving data (observations). While these equations are rather hard to look at, the concept is simple: Given an initial observation and belief over counts ϕ and θ, the agent can, in theory, compute all (countably infinite) hyper-state states conditioned on this initial belief. Thus, the model becomes *known* with respect to its priors, with $\mathcal{O}$ and T updated upon the agent receiving new data when interacting with the environment.
While this represents belief states in a POMDP in a mathematically precise way, convergence is only assured with respect to the agent's initial prior (Katt et al., 2018). However, despite this, the framework has shown good convergence properties in practice (Ross et al., 2007; Erik et al., 2015; Katt et al., 2018).
The following section explores the field of Predictive Processing, and aims to analyse the material so as to create an intuitive presentation of the connection between it and the Free Energy Principle.

2.6 Predictive Coding

The idea that humans (and by extension other biological organisms) use a type of statistical inference for perception dates back the ideas of Helmholtz (Sikl, 2001), whose key hypothesis was that systems which experience sensory input attempt to model the causes of said input (Clark, 2013). From this premise emerged bodies of work in the fields of machine learning, neuroscience and philosophy of mind (Dayan et al., 1995; Rao and Ballard, 1999; Friston et al., 2006; Hohwy, 2013)
The field of predictive coding (and predictive processing) is vast and multi-faceted, and in this text we will mainly focus on the general form of it as it pertains to generative models and error minimisation. In its broadest sense, a generative model can be understood as a statistical model which aims to encode the relationship between the values of two or more random variables in an efficient way (Harshvardhan

et al., 2020). As it is referred to throughout this dissertation, it can be more specifically viewed as a model which captures the causal relationships between hidden states and observations which are generated by such hidden states. True to their name, generative models are used to generate statistical predictions over data. Hinton and Zemel (1993) and Dayan et al. (1995) used these ideas to develop machine learning architectures with such generative models, the fundamental purpose of which was to avoid the need for supervised learning when learning to perform complex inference. The main idea behind this was that a generative model could be used to generate synthetic training data. Importantly, in their formalisation of the 'Helmholtz Machine', Dayan et al. used the definition of Helmholtz Free Energy to derive a process now known as variational free energy minimisation, whereby a *recognition model* approximates an intractable posterior distribution over latent variables. Although it was not explicitly stated as such in this work, this implementation had its basis as a form of Variational Inference.

Alongside, and in an extension to, the concepts of the generative model and statistical perceptual inference, the foundations of the neuroscientifc theory of predictive coding was established (Srinivasan et al., 1982; Mumford, 1992; Rao and Ballard, 1999). While these earlier works focused more on the functioning of specific cortical structures, the theory has since developed into a broad field which aims to act as a unifying theory of general cortical function (Friston, 2003, 2005, 2010; Clark, 2013). The essence of this theory is that the function of the brain is to minimise the error resulting from a mismatch between predicted and received sensory input. Naturally, the prediction here is constructed via an equipped (or learned) generative model of some form. In an extension to this, the theory accounts for a multi-layered approach to predictive coding, describing the brain as inherently hierarchical, with each layer of the hierarchy making predictions about the layer immediately below it (Clark, 2015; Millidge, 2021; Rao and Ballard, 1999), where 'below' in this case is not necessarily a spatial description, but rather a description of the order in which data is processed. The error resulting from a prediction mismatch, in this case, is used to update the generative model components (predictive estimator in Roa & Ballard's work) of the layer making the prediction. In turn, this update subsequently causes a mismatch between the information at the now current layer, and the prediction from the layer above, and so generates another error signal to that higher layer, resulting in a repeat of the process (Figure 4).

Following from Roa and Ballard's seminal paper (1999) which posited this hierarchical structure, predictive coding theory has since developed into a framework which implements inference and learning via variational Bayes (Millidge, 2021; Friston, 2003, 2005, 2010; Friston et al., 2015).

The picture presented here is of a brain which uses deep hierarchical predictive models to represent its external environment. Thus perception is not simply a 'feed-forward' process of 'experiencing' raw sensory input, but rather a top-down inference, modulated by said real sensory data. Due to the nature of top-down predictions being generated from 'within' rather than from 'without', perception can be seen as a sort of controlled hallucination (Clark, 2013; Seth et al., 2020).

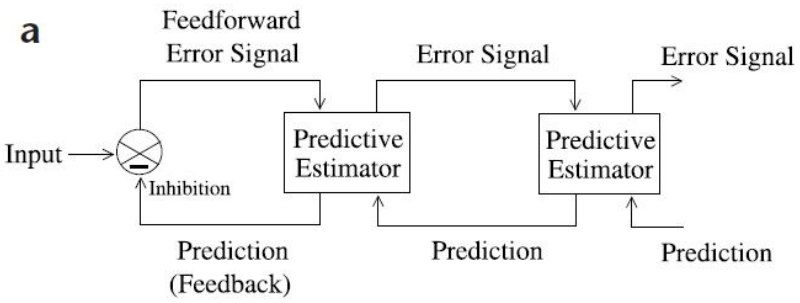

Figure 4: (Rao and Ballard, 1999). The general architecture of the hierarchical network used for predictive coding. At each level a predictive estimator attempts to predict incoming input. This prediction is then compared to the true input. The error from this comparison is propagated to the next layer of the hierarchy, where it acts as the input for further prediction/input comparison. Although, in their paper, Rao and Ballard constructed this architecture to represent neural activity, the initial input can be seen as some general type of initial data,, such as photons entering the retina, or training examples to a neural network.

2.6.1 A Basic Predictive Coding Example

Before continuing, let us display, via example, how prediction error minimisation might be implemented in a Bayesian fashion. This example closely follows some of the text from Bogacz's tutorial (2017). To do this, we return to the simple organism introduced in Section 2.4. As we have seen, the organism uses a Bayesian setup to model its beliefs about the true state of its environment, given observation signals it receives. Although generative models can take many forms, this example makes use of the type of generative model motivated by the theory of Bayesian inference and learning (Friston, 2005). For simplicity, let's imagine that both its prior over states and its likelihood can be modelled with two Normal distributions.

$$\begin{aligned} p(s) &= \mathcal{N}(s; \mu, \Sigma_s) \\ p(o|s) &= \mathcal{N}(o; v(s), \Sigma_o) \end{aligned} \tag{25}$$

Here $v(s)$, the mean of the likelihood distribution, is some functional mapping between the degree of danger, $s \in \mathbb{R}$ and size, $o \in \mathbb{R}$, of the observed shape. As previously stated, the objective of the organism is to try determine the probability over potential hidden causes of the observation it receives. Initially let's assume that the organism only cares about estimating the most likely cause of the observation, rather than the distribution over all causes. This can then be calculated in terms of the aptly-named *maximum a posteriori* method, which simply involves maximising the numerator of Bayes' formula. The denominator (normalising constant) can be ignored as we are trying to find a single, maximum value of s, rather than a whole distribution, and so normalisation is unnecessary. We thus seek $s^* \in S$ such that,

$$s^* = \arg\max_s p(o|s)p(s) \tag{26}$$

substituting in the two normal distributions defined above and taking the logarithm of the term, as it is easier to work with and has the same maximum point,

$$F = \ln p(o|s)p(s) = \ln\left[\mathcal{N}(o; v(s), \Sigma_o)\right] + \ln\left[\mathcal{N}(s; \mu, \Sigma_s)\right]$$

$$= \ln \frac{1}{\sqrt{2\pi\Sigma_o}} exp\left(-\frac{(o - v(s))^2}{2\Sigma_o}\right) + \ln \frac{1}{\sqrt{2\pi\Sigma_s}} exp\left(-\frac{(s-\mu)^2}{2\Sigma_s}\right) \tag{27}$$

$$= -\frac{1}{2}\left(\frac{(o - v(s))^2}{\Sigma_o} + \frac{(s-\mu)^2}{\Sigma_s}\right) + C$$

where C is a term encompassing all the constants that will vanish upon taking the derivative. We now have a term we can maximise with respect to s in the direction of its gradient,

$$\frac{dF}{ds} = \frac{o - v(s)}{\Sigma_o} v'(s) + \frac{\mu - s}{\Sigma_s} \tag{28}$$

The two terms in Equation 12 clearly show off the intuitive concept that maximising the numerator, $p(o|s)p(s)$, involves simultaneously moving s toward the mean of the prior, μ, and in a direction which is congruent with the observation, o. Another striking observation is that they represent a weighted measurement of the difference between expected and actual values (the expected values here being the two means μ and $v(s)$). Thus, these terms can be thought of as representing the weighted prediction errors, with $\mu - s$ being the difference between expected and estimated s and $o - v(s)$ the difference between the actual observation received and the observation expected, conditioned on the estimated state. Therefore, it is evident that maximising F in this scenario is equivalent to minimising prediction error for a specific observation.

2.6.2 Variational Free Energy

In the example above, a few key assumptions were made. The first is that the organism is equipped with a generative model that is already known and accurate at representing the generative process. Often this is an unrealistic assumption, and in general, it is important for our organism to be equipped with the mechanisms to update its models of the world. Learning here takes the form of updating some parameter θ (or set thereof), which defines the shape of a generative density, so as to better fit the observed data.

$$p(s, o; \theta) = p(o|s; \theta)p(s; \theta) \tag{29}$$

A second assumption is that the organism only cares about finding a single $s \in S$ - the mode of the posterior distribution. This point-based calculation is an example of *estimation* rather than full *inference*, which involves approximating the whole posterior. As we saw in the section on Variational Inference, it is often useful to use some family of approximating densities to achieve such 'full' inference. Although the coupling of the concepts behind predictive coding and approximate inference was intro-

duced by Dayan et al. (1995), the idea of explicitly representing the predictive coding mechanism as a Variational Inference problem was formalised by Friston (2003; 2005; 2008). Here Friston showed that predictive coding has a natural Bayesian underpinning, and that the minimisation of prediction error can be achieved via the minimisation of variational free energy. We define variational free energy as:

$$\mathcal{F} = D_{kl}\left[q(s|o;\phi)||p(o,s;\theta)\right] \tag{30}$$

Where q is the approximating density. Notice that in order to approximate $p(o,s)$, q is driven to assign probability mass to posterior states based on the prior over these states $p(s)$ and the likelihood of the observations, $p(o|s)$ as is analogous with the gradient of F in the predictive coding example above.
In these earlier works, Friston proposed using the Expectation-Maximisation algorithm (Dempster et al., 1977) to weave both inference and learning into the same predictive coding objective function. Although the **EM** algorithm is used to find a point-estimate (maximum likelihood or maximum a posteriori), as we will see, it is equivalent to variational Bayesian inference under the Dirac delta distribution, and conceptually shows the validity of the use of Variational Inference in variational free energy minimisation.
In the **E**-step in this algorithm, the function is optimised (in this case minimised) with respect to ϕ, while holding θ constant. This is essentially learning the approximating distribution with respect to the current generative model, and can be seen as representing *inference*. The **M** step involves holding ϕ constant and optimising $\mathcal{F}$ with respect to θ. By doing so, the generative model is modified to implicitly assign greater probability mass to the observation, o. This step can be viewed as representing *learning*. To make this process more intuitive, we separate the generative model into a prior and likelihood.

$$\begin{aligned}
\mathcal{F} &= D_{kl}\left[q(s|o;\phi)||p(s|o;\theta)p(o;\theta)\right] \\
&= \int q(s|o;\phi)\ln\frac{q(s|o;\phi)}{p(s|o;\theta)}ds - \int q(s|o;\phi)\ln p(o;\theta)ds \\
&= D_{kl}\left[q(s|o;\phi)||p(s|o;\theta)\right] - \ln p(o;\theta)
\end{aligned} \tag{31}$$

In this form, it is more clear what the **EM** algorithm achieves. The **E** step minimises $\mathcal{F}$ by changing ϕ to move q closer to the posterior, $p(s|o;\theta)$, while the **M** step minimises $\mathcal{F}$ with respect to θ by increasing model evidence, $p(o)$. To link this to Equation 1 in Section 2.2, we note that $\mathcal{F}$ is exactly equivalent to the negative Evidence Lower Bound.

$$\begin{aligned}
\mathcal{F} &= -\int q(s|o;\phi)\ln\frac{p(o,s;\theta)}{q(s|o;\phi)}ds \\
\ln p(o;\theta) &= -\mathcal{F} + \int q(s|o;\phi)\ln\frac{q(s|o;\phi)}{p(s|o;\theta)}ds \\
\ln p(o;\theta) &= \int q(s|o;\phi)\ln\frac{p(s,o;\theta)}{q(s|o;\phi)}ds + \int q(s|o;\phi)\ln\frac{q(s|o;\phi)}{p(s|o;\theta)}ds
\end{aligned} \tag{32}$$

As noted in Section 2.2, the ELBO is a tractable bound which can be maximised to achieve inference. Although the approximating distribution can theoretically be a member of any arbitrary family of distributions (see (Friston, 2003), for the use of a Gaussian distribution with Laplace approximation), for brevity, we model it as a Dirac delta density, with all probability mass assigned to a single point,

and all other points having a probability of 0.

$$q(s|o;\phi) = \delta(s - \phi) \tag{33}$$

where

$$\delta(x) = \begin{cases} +\infty, x = 0. \\ 0, x \neq 0. \end{cases}$$

and

$$\int_{-\infty}^{\infty} d(x) = 1$$

Given that the entropy of the delta density is 0, $\mathcal{F}$ in this case simplifies as follows:

$$\begin{aligned} \mathcal{F} &= (s|o;\phi) \ln q(s|o;\phi ds_{\text{entropy}} - \int q(s|o;\phi) \ln p(o,s;\theta) ds \\ &= 0 - \int q(s|o;\phi) \ln p(o,s;\theta) ds \\ &= -\int q(s|o;\phi) \ln p(o,s;\theta) ds \end{aligned} \tag{34}$$

Modelling the generative model as Guassian, we have

$$\int q(s|o;\phi) \ln p(o,s;\theta) ds = -\int \delta(s - \phi) \ln \left[\mathcal{N}(o, v(s,\theta_1), \Sigma_o)\mathcal{N}(s,\theta_2,\Sigma_s)\right] ds \tag{35}$$

Under the delta density, the expectation simplifies to the sum of the logarithms of the two Normal distributions for the value of $s = \phi;\ \phi \in S$

$$\mathcal{F} = -\ln \left[\mathcal{N}(o; v(\phi,\theta_1), \Sigma_o)\right] - \ln \left[\mathcal{N}(\phi;\theta_2,\Sigma_s)\right] \tag{36}$$

This is exactly the same form as was presented in the simple predictive coding example above, and we can minimise prediction error by performing gradient descent on the derivative of $\mathcal{F}$ with respect to ϕ (**E**-step). It is also important to note here that prediction error can also be minimised via minimising the variational free energy with respect to θ (**M**-step). This learning of the generative model can be thought of as achieving an implicit, more long-term, prediction error minimisation, via the minimisation of surprising observations.

Although the use of a delta density results in exactly the same form as MAP optimisation, this special case conceptually shows off the unification of variational free energy and prediction error minimisation. Furthermore, it is evident that the minimisation of variational free energy can be framed as a Variational Inference technique that results in both the approximation of a true posterior, the maximisation of model evidence and the minimising of prediction error.

2.6.3 Hierarchical predictive coding

A problem that often arises in Bayesian inference techniques is constructing an adequate prior. Although Variational Inference often converges to good solutions (Blei et al., 2016). The speed at which it converges is heavily correlated with the shape and accuracy of the initial prior distribution. Looking at this from another perspective, a valid question is how a brain might construct priors 'de novo'. Friston (2003; 2005) postulates that a way for the brain to represent the process of empirical Bayes, is via hierarchical layers, whereby estimates over latent variables at one level act as priors to a subordinate level. The generative model can then be expressed as follows

$$p(s_0, s_1, ..., s_n) = p(s_n) \prod_{i=0}^{n-1} p(s_i|s_{i+1}) \tag{37}$$

Here s_n is some non-empirical 'deepest level' prior and $s_0 = o$ is the actual observation received from the environment at the first level of the hierarchy. In this sense, inferences about hidden states at lower levels of the hierarchy act as observations to higher levels. This is similar to the model presented by Rao and Ballard, where each higher-level layer in the network acted to predict the layer one level below. Crucial to understanding the usefulness of such a set-up is the fact that the likelihood of some latent variable at level i, $p(s_i|p_{i+1}; \theta)$ acts as the prior for the layer below, $p(s_{i-1}|s_i; \theta)p(s_i; \theta)$, therefore the likelihood parameters for any given level are implicitly learned upon the process of learning the prior for the subordinate level.

2.7 The Free Energy Principle

The Free Energy Principle presents an amalgamating neuroscientific theory, taking ideas from non-equilibrium thermodynamics, predictive coding, Bayesian Inference and agent-based Markov process control (Friston et al., 2006; Friston, 2010, 2019). As its core thesis, the Free Energy Principle postulates that in order for a system to maintain a non-equilibrium steady state with respect to its external environment, it must encode an approximation of the dynamics of that environment, at least insofar as those dynamics interact with the system. This idea can be viewed as a generalisation of Conant and Ashby's thesis (Conant and Ashby, 1970) which states that "every good regulator of a system must become a model of that system." (Millidge, 2021). Effectively encoding dynamics of an external environment can ultimately be achieved in two ways: By updating one's model of those dynamics, or by altering the external dynamics to fit your model. As is evident, the type of predictive coding discussed in the previous section describes a process by which the former is achieved - updating a model in response to evidence, rather than selectively sampling evidence.

In order to formulate the concepts which make up the Free Energy Principle, it is important to explore how one defines what a system is in this paradigm. This is done via the use of a Markov Blanket (Pearl, 1989), which defines an information boundary with respect to connected random variables. When this is coupled with the dynamics of a non-equilibrium steady state, the intuitive picture of a self-organising system which performs free energy minimisation to resist thermodynamic dissipation emerges. In terms of specifically describing the functioning of brains (at a slightly higher level of abstraction), the Free Energy Principle serves as an explanation for how brains might maximise model evidence (or minimise

surprise). As we have seen, the variational free energy (or negative evidence lower bound) can offer a tractable way to do this as, unlike the negative log-evidence, it is a function of sensory data and internal brain states, both of which the brain can effectively encode and evaluate (Friston, 2009). In doing so, the brain becomes a (biased) model of its external environment, constantly seeking to capture both the prevalent external states, as well as proofs for its own preferential existence.

2.7.1 Non-Equilibrium Steady States (NESS)

To begin to define the mathematical underpinnings of the **FEP**, we introduce a simple Langevin stochastic differential equation which is effective at modelling the evolution of some set of degrees of freedom. Importantly, Langevin differential equations are inherently Markovian, in that the dynamics at any point in time only rely on the values of the elements in the set at that point. For clarity, in this context, we can specifically view the elements of this set as states $x = [x_0, ..., x_n]$. The differential equation can then most simply be written as

$$\frac{dx}{dt} = f(x) + \mathcal{W} \tag{38}$$

Here, f is some non-linear, differentiable, deterministic function and $\mathcal{W}$ is the additive Gaussian white noise which represents fast time-scale fluctuations in the dynamics (this term is what inherently makes the system stochastic). At each arbitrary time point i, $\mathcal{W}$ can be expressed as $\mathcal{W} = \mathcal{N}(x; 0, 2\Gamma)$, with the co-variance between any two arbitrarily close time-points being 0: $\text{Cov}(\mathcal{W}_i, \mathcal{W}_j) = 0 \ \ \forall \ i \neq j$. By absorbing $\mathcal{W}$ into f we can represent this equation in terms of a probability density, $p(x, t) \equiv p(x_t)$. To model the evolution of this differential equation, we express it as a Fokker-Planck equation

$$\frac{dp(x,t)}{dt} = -\nabla_s \left[f(x,t)p(x,t) - \nabla_x \Gamma \nabla_s p(x,t) \right] \tag{39}$$

From this point, we postulate that the dynamics of this setup tend toward a gradient of zero in the limit, $lim_{t\to\infty} p(x,t) \to p^*(x)$. We thus say that $p^*(x)$ is a steady-state density such that the dynamics of s no longer become a function of the time trajectory. At the core of the dynamics of the **FEP** is the concept of a **NESS**. While an equilibrium steady state describes a system where transitions between states are in equilibrium and symmetrical with respect to time, a **NESS** is a state where, although the distribution over states remains constant, there is still *directionality* to the dynamics with respect to time. More intuitively, **NESS** are occupied by systems which require input energy and/or matter for the steady state to be maintained, whereas (most) steady states at equilibrium are closed systems. Biological systems can all be considered non-equilibrium steady states, where the transition to an equilibrium steady state would be equivalent to death. By this definition, we wish to show that $dp*(x)/dt = 0$, and we hope to do so in terms of mathematical objects that are useful to work with in this context. In essence, one wants to find a useful representation of f.

To do this, we use the Helmoltz decomposition (Friston and Ao, 2012; Friston, 2019) to represent the dynamics of states at **NESS** as a linear combination of solenoidal (divergence-free) and dissipative (irrotational) flows

$$\frac{dx}{dt} = (\Gamma(x) - Q(x))\nabla_s \ln p^*(x) \tag{40}$$

where Γ and Q are the dissipative and solenoidal flows respectively. Notice that the dissipative flow here is equivalent to half the variance of $\mathcal{W}$, the Gaussian white noise introduced in Equation 13. Conceptually, the solenoidal flow, which can be viewed as representing the input of energy or matter into the system, can be interpreted as counteracting the dissipative flow, allowing the system to maintain a steady distribution over states. Substituting the Helmholtz decomposition into the original Fokker-Planck equation, we have

$$\begin{aligned}
\frac{dp(x,t)}{dt} &= -\nabla_s \left[f(x,t)p(x,t) - \nabla_x \Gamma \nabla_x p(x,t) \right] \\
&= -\nabla_x \left[(\Gamma(x) - Q(x)) \frac{\nabla_x p^*(x)}{p^*(x)} p^*(x) + \Gamma(s) \nabla_x p^*(x) \right] \\
&= -\nabla_x \left[(\Gamma(x) - Q(x)) \nabla_x p^*(x) + \Gamma(x) \nabla_x p^*(x) \right] \\
&= -\Gamma \nabla_x^2 p^*(x) + \Gamma \nabla_x^2 p^*(x) + \nabla_x Q \nabla_s p^*(x) \\
&= \nabla_s Q(x) \nabla_s p^*(x) = 0
\end{aligned} \tag{41}$$

The last line follows from the fact that we assume that $Q = -Q^T$ and that Q is orthogonal to the gradient of the density (Millidge, 2021). For a full proof of this, see (Friston (2019), Appendix B). We now have a way to describe the dynamics of the non-equilibrium steady state in terms of solenoidal and irrotational flows. This, as we will see, becomes useful when attempting to describe the flow of states within the context of a Markov Blanket.

2.7.2 Markov Blankets

How should the boundary between the 'internal' and 'external' be formalised? At its deepest level this can almost be viewed as a philosophical problem of forming abstractions. At some point one needs to define a level of abstraction that represents some meaningful object we can work with and which provides explanatory power. In the case of the Free Energy Principle, this abstraction is the Markov Blanket (Pearl, 1989). Given a set of random variables, a Markov Blanket defines the minimum subset needed to define a specific random variable in that set. Practically, this allows one to define, for each random variable, which other random variables it is conditionally independent of and, conversely, which it is conditionally dependent on. To link this to the previous section, we define x to be the set of states $x = [\epsilon, o, \alpha, \mu]$, where

- ϵ = External States
- o = Observation States
- α = Active States
- μ = Internal States

Figure 5 shows how these states are grouped in terms of their conditional dependencies. Here the sensory and active states are blanket states relative to the internal states, and can be expressed in terms of conditional dependency as

$$p^*(x) = p^*(\mu, o, \alpha, \epsilon) = p^*(\epsilon|b)p^*(\mu|b)p^*(b) \tag{42}$$

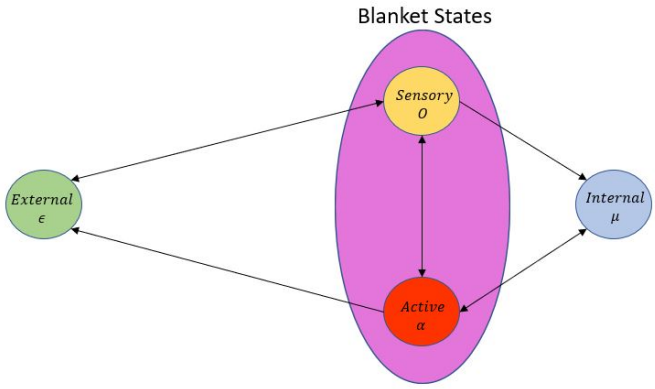

Figure 5: Markov Blanket. The states in the pink oval are the blanket states, which mediate interaction between external - ϵ and internal - μ states

where $b = [o, \alpha]$. The actual dynamics of this setup can be formalised in terms of the *Marginal Flow Lemma* (Friston, 2019), which states that the (Helmholtz decomposed) flow of a subset of states, when averaged under the complement of a second subset (marginalised), can be expressed in terms of the gradients of the logarithm of the marginal density over that second subset (for the full proof, see Appendix B in (Friston, 2019). This lemma is accompanied by a corollary which restricts the interaction of flows depending on conditional independence, as presented in Equation 15. In the context of the Markov Blanket flows, this yields a system of equations describing the flow of each state as a function of the states it directly interacts with.

$$\begin{aligned} f_\epsilon(\epsilon, b) &= (Q_{\epsilon\epsilon} - \Gamma_{\epsilon\epsilon})\nabla_\epsilon \ln p(\epsilon, b) \\ f_o(\epsilon, b) &= (Q_{oo} - \Gamma_{oo})\nabla_o \ln p(\epsilon, b) + Q_{sa}\nabla_a \ln p(\epsilon.b) \\ f_\mu(\epsilon, b) &= (Q_{\mu\mu} - \Gamma_{\mu\mu})\nabla_\mu \ln p(\mu, b) \\ f_\alpha(\mu, b) &= (Q_{\alpha\alpha} - \Gamma_{\alpha\alpha})\nabla_\alpha \ln p(\mu, b) + Q_{as}\nabla_s \ln p(\mu, b) \end{aligned} \tag{43}$$

We further define two other groupings of states, particular states $\pi = [\mu, o, \alpha]$ and autonomous states, $a = [\mu, \alpha]$. Particular states are the subset including internal - μ, sensory - o and action -α states, and are grouped to indicate the states that are specifically part of the system implementing the Markov Blanket. The autonomous states are those states which do not receive direct input from external states, ϵ. Of interest, is the flow of how particular states interact with external states. External states here are synonymous with hidden states previously referenced throughout this text. At the center of understanding this is the evaluation of the actual **NESS** density $\ln p^*(\pi)$ for particular states, and its role in defining the entropy of external states in relation to these particular states and vice versa (mutual information). Conceptually, mutual information here is subject to two opposing constraints. On the one hand, low entropy of particular states is proportional to low average uncertainty about those states given external states, and thus there is high mutual information. Contrasting this, if mutual information is too low, then the meaningful differentiation between particular and external states dis-

appears, which would theoretically result in the dissipation of the Markov Blanket altogether (Friston, 2019). These opposing constraints can be derived by noting that surprisal (otherwise known as Shannon information) $-\ln p^*(\pi)$ is equal to the expected surprisal under ϵ, $-\ln p^*(\pi) = \mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\pi)]$ as $\ln p^*(\pi)$ does not depend on ϵ. With some expansion, we are able to represent $\ln p^*(\pi)$ in terms of *Inaccuracy* and *Complexity*.

$$\begin{aligned} -\ln p^*(\pi) &= \mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\pi)] \\ &= \mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\epsilon|\pi) - \ln p^*(\epsilon,\pi)] \\ &= \mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\epsilon|\pi) - \ln p^*(\pi|\epsilon) -^* (\epsilon)] \\ &= \underbrace{\mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\pi|\epsilon)]}_{\text{Inaccuracy}} + \underbrace{D_{kl}[p^*(\epsilon|\pi)||p^*(\epsilon)]}_{\text{Complexity}} \end{aligned} \tag{44}$$

These two terms are best understood through the lens of Bayesian statistics, where *Complexity* is the measure of the divergence between a prior and posterior over external states, and Inaccuracy the expected likelihood of particular states under the posterior of external states. Minimising inaccuracy corresponds to maximising this likelihood, while minimising complexity corresponds to minimising the difference between prior and posterior beliefs over external states. The minimisation of both these terms naturally leads to the minimisation of surprisal. From here the role for variational inference in the Free Energy Principle starts to become more clear, as there is a necessity to 'indirectly' calculate or approximate a posterior over external states using information from blanket states. The following section explores this formal link between variational inference and Markov Blanket dynamics.

2.7.3 Inferring Blankets

As we have seen, the Markov Blanket acts as a mediating device between the flow of internal and external states. Given this premise, a natural step is to define them as a function of blanket states, with the most likely external and internal states given respectively by

$$\epsilon^*(b) = \arg\max_{\epsilon} p(\epsilon|b) \tag{45}$$

$$\mu^*(b) = \arg\max_{\mu} p(\mu|b) \tag{46}$$

furthermore, we can posit a function which serves as a mapping between most likely internal and external states

$$\epsilon'(b) = \mathcal{J}(\mu(b)) \tag{47}$$

However, in the spirit of variational inference, we are interested in a whole distribution over hidden states, rather than simply a most likely point. One way to define this could be to use the most likely point of hidden states to paramaterise a distribution over possible hidden states.

$$\epsilon \sim q(\mathcal{J}(\mu(b)) = q(\epsilon|b;\mu) \tag{48}$$

We can thus say that the flow of internal states explicitly parameterises a distribution over external

states and, implicitly, parameterises a distribution over internal states. Formally, this setup can be modeled via the field of information geometry (Amari, 1995) which, for exponential distributions, uses Fisher Information as a distance metric between distributions with varying parameters (Millidge, 2021).

The next fundamental step from here is to show that the dynamics of a NESS Markov Blanket can be interpreted as performing approximate variational Bayesian Inference (Millidge, 2021). To do this, we return to the decomposition of the NESS density (surprisal) into Inaccuracy and Complexity, with respect to the flow of particular states π.

$$
\begin{aligned}
-\ln p^*(\pi) &= \underbrace{\mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\pi|\epsilon)]}_{\text{Inaccuracy}} + \underbrace{D_{kl}[p^*(\epsilon|\pi)||p^*(\epsilon)]}_{\text{Complexity}} \\
\ln p^*(\pi) &= \mathbb{E}_{p^*(\epsilon|\pi)}[\ln p^*(\pi|\epsilon)] - D_{kl}[p^*(\epsilon|\pi)||p^*(\epsilon)] \\
&= \mathbb{E}_{p^*(\epsilon|\pi)}\left[\ln \frac{p(\pi,\epsilon)}{p(\epsilon)} - \ln p(\epsilon|\pi) + \ln p(\epsilon)\right] \\
&= \mathbb{E}_{p^*(\epsilon|\pi)}\left[\ln p(\pi,\epsilon) - \ln p(\epsilon|\pi)\right] \\
&= -D_{kl}[p(\epsilon|\pi)||p(\pi,\epsilon)] \\
&\approx -D_{kl}[q(\epsilon|b;\mu)||p(\epsilon,\pi) \\
\mathcal{F}_\pi &\approx \ln p^*(\pi) + D_{kl}[q(\epsilon|b;\mu)||p(\epsilon|\pi)]
\end{aligned}
\tag{49}
$$

In the last two lines, an approximating distribution, q replaces the true posterior, $p(\epsilon|\pi)$, while also noting that π can be expressed in terms of blanket states b parametrised by μ. With the assumption that the variational posterior is equal to the true posterior, the lower bound vanishes, and the particular variational free energy becomes equal to $\ln p^*(\pi)$. Given that now the particular variational free energy is equal to the log of the NESS density, the flow of autonomous states can be written in terms of this free energy

$$
f_a(x) = -(Q_{aa} - \Gamma_{aa})\nabla_a \mathcal{F}_\pi(\pi) \tag{50}
$$

When this assumption is not held, the NESS log density can be seen as approximating the particular variational free energy

$$
f_a(x) \approx -(Q_{aa} - \Gamma_{aa})\nabla_a \mathcal{F}_\pi(\pi) \tag{51}
$$

The core hypothesis here, one that is at the foundation of the Free Energy Principle, is that the dynamics of Markov Blankets at a NESS can be seen as performing approximate variational Bayesian inference over external hidden states, based on a paramterisation given by internal states (Millidge, 2021). If we model inference here as determining a MAP rather than the full posterior density, it can be shown that, for a Gaussian approximating density, under the Laplace assumption, the mode of external states can be represented as a function of the mode of internal state, the dynamics of which are achieved via a gradient descent on the variational free energy of internal states (Millidge, 2021; Friston, 2019).

Although this derivation does not offer a formal mathematical proof, it serves a conceptual purpose, allowing us to posit the core FEP premise that a self-organising system at NESS maintains its steady state via performing approximate modelling of external states in its environment and, in doing so,

minimises variational free energy.

2.7.4 Free Energy, Agents and the Brain

Perhaps the most attractive quality of the **FEP** is its capacity to describe, in broad terms, the functioning of brains. Its congruence with both predictive coding and the idea of the Bayesian Brain, arguably make it a viable candidate for a 'unifying' brain theory (Friston, 2010). In his seminal paper (2006), Friston applies the foundational mathematical ideas discussed above to describe the brain as a model of its external environment (albeit a constrained model). Here Friston introduces the so-called ensemble density over external states, the parameters of which are encoded internally by the brain. Naturally, this is exactly the same as the variational approximating density discussed above, however he expands upon the idea slightly by considering the density as a mean field approximation of many (time-varying) environmental states

$$q(\epsilon) = \prod_i q(\epsilon_i; \mu_i) \tag{52}$$

The fact that the partitioning of states and parameters is done to represent time-varying external parameters is not necessarily a core aspect of the **FEB**, however it is used here to conceptually show the interesting ways in which the internally-controlled approximating variational density can be used to encode external phenomena. This way of partitioning states is known as the mean-field approximation (Section 2.2), and serves to greatly simplify the process of approximation under a set of categorically separate latent variables. Despite the various forms described in the previous section, the explicit mention of a generative model was absent. Sticking with the most recent notation, this is simply the joint density of external and particular states $p(\epsilon, \pi) = p(\epsilon, b; \mu)$. For simplicity, in this context, we will view the active states, α as the force exerted by the system's effectors (Friston et al., 2006), which act to selectively sample sensory Observation states from the environment. The generative model can then be expressed as $p(s, o)$ where we define $s = \epsilon$ to adhere with earlier notation, and the sensory input the system 'receives', o, is implicitly a function of α, $o = o(\alpha)$ It is simple to show, via the theory of Variational Inference (Bishop, 2006), that the approximating density over each latent variable partition can be expressed as

$$q_i(s_i) = exp(\mathbb{E}_{q_{j \neq i}} [\ln p(s, o; \theta)]) \tag{53}$$

Where θ is the parameters of the generative model, as used in Section 2.6. This result offers a convenient way to calculate the variational posterior over each latent variable, however it must be noted that the assumption that the latent variables are statistically independent is often a simplification of the true dynamics of states.

Treating q as Gaussian (Laplace approximation), we can take inspiration from predictive coding, and frame perceptual inference as optimising the modes, encoded by (neuronal) internal states, of each time-varying state partition. Here perceptual inference can be cast as optimising the parameters encoding the faster temporal states, perhaps representing neuronal and electromagnetic activity of the brain that changes within a timescale of milliseconds. Perceptual learning, on the other hand, is represented by the optimisation of the parameters encoding the temporally slower external states, which could represent long-term changes in synaptic connections (Friston et al., 2006). In addition

to this, we can also consider the learning of the generative model as an interplay between a form of Bayesian model selection, and model optimisation (akin to the **M** step in the **EM** algorithm) (Friston et al., 2006). This interplay essentially yields a dual-process of optimising each potential model in the direction of Free Energy minimisation, with model selection based upon the results of these optimisations. In the aforementioned paper, Friston argues that model selection can be viewed as a form of Value-learning, whereby seemingly adaptive behavior can be described by the Free Energy Principle without using notions such as the maximisation of expected reward or value, as is the case in most machine learning applications. Rather, all that is required to describe such behavior is for the organism/brain to maximise the likelihood of its expectations, with evolution selecting for valuable **a priori** expectations, and lifetime experience modifying parameters of these models to best conform to both evidence and expectation. Although only subtly different from the frameworks of model-based Reinforcement Learning, this is an interesting notion that will be explored in more detail in Section 2.8. It essentially views the concept of agent behavior and learning through a slightly different lens, with the minimisation of variational Free Energy achieved via a *biased* form inference and model learning/selection, acting as the driving mechanism for optimal action, rather than an explicit value-based reward function.

2.7.5 Summary

In this section, we have attempted to present the Free Energy principle in a bottom-up manner, starting from its mathematical underpinnings. Its roots begin in the concepts of Non-equilibrium thermodynamics, where the Fokker-Planck equation is used to express the stochastic dynamics of a set of states, x

$$\frac{dp(x,t)}{dt} = -\nabla_s \left[f(x,t)p(x,t) - \nabla_x \Gamma \nabla_s p(x,t) \right] \tag{54}$$

The core idea of such a formulation is to try and represent the dynamics of living systems in terms of a stochastic trajectory of a set of states. The next step from here is to find a useful representation for $f(x)$, which is the original non-linear Langevin differential equation which describes $\frac{dx}{dt}$. Friston and Ao (2012; 2019) posit the use of the Helmholtz decomposition to represent this differential equation when it is at a Non-equilibrium steady state

$$\frac{dx}{dt} = (\Gamma(s) - Q(x))\nabla_s \ln p^*(x) \tag{55}$$

It is simple to show that this solves the differential equation for the case of a steady-state distribution, by substituting the Helmholtz decomposition into the Fokker-Planck equation and noting that this expression evaluates to 0 due to the solenoidal flow being anti-symmetric and orthogonal to the gradient of the **NESS** density.

Given this representation of state dynamics, we move on to describing the state-space in more detail, partitioning the states into a defined set which can cohesively interact in the form of a Markov Blanket. This set, $x = [\epsilon, o, \alpha, \mu]$ represents external, observation, active and internal states respectively. The

important point of this partition is to define the dependence relationships between these states.

$$p^*(x) = p^*(\mu, o, \alpha, \epsilon) = p^*(\epsilon|b)p^*(\mu|b)p^*(b) \tag{56}$$

This equation organises the inter-state relationships in a crucial way which limits their direct interactivity and by doing so, defines a set of 'blanket' states which act as mediating factors between the internal states of the system and the external states of the environment. Given this factorisation of states, we can define the differential equation of each partition, as a Helmholtz decomposition of the flows of the other state partitions it directly interacts with (including itself). For example, ignoring solenoidal coupling, the flow for active states is expressed as:

$$f_\alpha(\mu, b) = (Q_{\alpha\alpha} - \Gamma_{\alpha\alpha})\nabla_\alpha \ln p(\mu, b) \tag{57}$$

Our attention now turns toward the **NESS** probability density itself. Given that the particular states, are both the blanket (sensory and active) and internal states, $\pi = [b, \mu]$, we are able to represent $-\ln p^*(\pi)$ in terms of an expectation over external states conditioned on particular states. This yields the terms classically known as *inaccuracy* and *complexity*.

$$-\ln p^*(\pi) = \underbrace{\mathbb{E}_{p^*(\epsilon|\pi)}[-\ln p^*(\pi|\epsilon)]}_{\text{Inaccuracy}} + \underbrace{D_{kl}[p^*(\epsilon|\pi)||p^*(\epsilon)]}_{\text{Complexity}} \tag{58}$$

The purpose of expanding the **NESS** density in such a way is to yield a conceptual form that shows off the result of minimising surprisal. In doing so, we minimise both inaccuracy and complexity. In Bayesian statistics, inaccuracy/accuracy measures the precision of the model under posterior beliefs, and complexity the difference between prior and posterior beliefs. Importantly, this formulation can be rearranged to give the variational free energy for the particular states.

$$\begin{aligned} \ln p^*(\pi) &\approx -D_{kl}[q(\epsilon|b;\mu)||p(\epsilon)|\pi)] \\ \mathcal{F}_\pi &\approx \ln p^*(\pi) + D_{kl}[q(\epsilon|b;\mu)||p(\epsilon|\pi)] \end{aligned} \tag{59}$$

where we replace the true posterior, $p(e|\pi)$ with a variational posterior over external states, parameterised by internal states, $q(e|b;\mu)$. Naturally, this is the very same variational free energy discussed in previous sections. Given that the approximating density is equal to the true posterior, we have that the variational free energy for particular states is equal to the **NESS** log density. Without this assumption of equality, we can say that the variational free energy approximates the log density. It is this formulation which reveals the crucial hypothesis in question, that self-organising systems at **NESS** perform approximate Bayesian inference in order to maintain that steady state and resist dissipation. In his papers on Free Energy, the brain and life (as we know it), Friston (2006; 2013) presents conceptual yet concrete examples of how the **FEP** could explain the way in which living organisms, and brains in particular, could perform perceptual inference and learning. This brings us full circle, and ties the knot between the ideas of predictive coding and the **FEP**. An important point that the section ended on, was the commentary on the subtle difference between an agent employing the **FEP** to perform

adaptive action and the more classical machine learning techniques used to achieve such behavior. As we will see in more detail in the next section, an explicit value or reward function is absent from agents under the **FEP**. Rather, selective action is performed via a biased sampling of the environment as a function of the agent's expectations about the states of the world as well as its drive to seek out information.

2.8 Active Inference

We have mentioned *action* and *active* states many times up until this point, but have not yet explored how these fit into the bigger picture. While the flow of internal states, in relation to external states, can be seen as performing a type of perceptual inference, the role and functioning of active states has not been fully incorporated. Indeed, when discussing the concept of an *agent*, intentionality, as opposed to a purely passive existence, is an implicit assumption. In terms of organic systems, action can be seen in two slightly different ways. In one way, the organism uses energy in order to change itself or the external environment to satisfy some goal or requirement. The other way of viewing this is by considering an organism which acts to selectively sample the environment so as to better match its expectations as to what samples (sensory inputs) it will receive (Millidge et al., 2021). This might seem rather circular, but the circularity is almost the point in this view. It allows for a type of 'biased' inference (Millidge et al., 2021; Parr and Friston, 2019), whereby the organism is performing work to increase the likelihood that its samples of the external environment match its internal expectations. Most broadly, if this expectation is that it exists, then its actions will contribute towards its continued existence, inline with Hohwy's notion of the self-evidencing brain (Hohwy, 2016). Naturally, this broad metric of 'existence' is not the actual criteria by which actions are performed. Rather, it might be achieved via proxy of other, smaller-scale criteria, which all ultimately contribute towards the primary existential goal.

Action itself can perhaps further be broken down into two broad categories, one which is *reactionary* and the other which is in the service of planning and goal-directed behavior (we will call this sophisticated action). Reactionary action deals only with the present time and although it can be argued that it is in the service of future states of the organism, it does not rely on any form of belief-based internal simulation of those future states. Sophisticated action, on the other hand, requires prediction about the future states of the organism and its external environment. Thus, actions in the present are made based on expectations about future states (Pezzulo et al., 2018).

So far, in our formulations of variational free energy for the flow of states, we have grouped active states with sensory and internal states, or made the sensory states an implicit function of active states. For clarity, these are now separated out, and we represent the probability densities in a form more inline with classical Bayesian statistics and representations used in previous sections of this text. Going forward, the following notation is used

- $s \in S$ - The external state space upon which the agent/organism performs inference.
- $o \in O$ - The sensory input, or observation, received by the agent/organism. This is the same as the sensory states discussed in the previous section.

- π - The policy of the agent representing a set of actions over time. In machine learning, this is often a function of state. In more basic Active Inference literature, it is a function of discrete timesteps.
- ϕ - The parameters which define the approximating distribution of the true posterior. In the previous section, we saw that these are controlled by internal states, μ.
- θ - The parameters which define the generative model. Like the approximating distribution, these parameters are also controlled by the agent.

With this notation, we therefore express the variational free energy as we did in Section 2.7.2

$$\mathcal{F} = D_{kl}\left[q(s|o;\phi)||p(s|o;\theta)\right] - \ln p(o;\theta)$$

In this form, minimising free energy is equivalent to approximating the true posterior over external states, as well as maximising the marginal likelihood of the received observation. This ultimately causes the agent to both infer the true state of its environment (without having to compute complex integrals) as well as change the parameters of its generative model to be more congruent with the data it receives. Importantly, the environment here is partially observable, with the agent only receiving observations, which probabilistically map to external states via some known or unknown function. Thus, the agent can never deterministically know the true state of the external world, rather it formulates beliefs about possible state configurations. As we have seen, this sort of inference can be construed as a type of perception, but says nothing about actual action and goal-directed behavior. Furthermore, if actions are chosen to bring about preferred future outcomes, the agent must be equipped with the capability to predict such future states and observations. Crucially, in order to actually make decisions over actions, the agent must have *preferences* as to what those future observations or states should be.
Active Inference (Friston et al., 2011; Friston and Ao, 2012) is a **FEP** framework which describes how a system might achieve goal-directed action via the minimisation of variational free energy. Within this paradigm, the preferences are encoded as a prior distribution over observations, and make up part of the agent's generative model. These preferences are what gives the generative model an inherent biased quality - a feature that is unique to Active Inference (Millidge et al., 2021), and shapes the agent's expectations of sensory input in a way which is preferential to it. Action, then, works to sample the environment in order to increase the probability of receiving such observations. In this sense, the agent is performing a type of 'planning as inference' (Sajid et al., 2021), where action, planning and perception weave together to result in an agent which looks as if it is performing intentional, adaptive behavior within its environment.

2.8.1 Expected Free Energy (EFE)

As its name suggests, Expected Free Energy (**EFE**) is the Free Energy of the agent over future trajectories of action (Parr and Friston, 2019; Sajid et al., 2021). This prediction allows the agent to factor in future time-steps in order to make decisions over actions in the present.

$$
\begin{aligned}
G(\pi) &= \mathbb{E}_{q(o,s|\pi)}\left[\ln \frac{q(s|\pi)}{p(s|o)} - \ln p(o)\right] \\
&= \underbrace{\mathbb{E}_{q(o,s|\pi)}[\ln q(s|\pi) - \ln p(s|o,\pi)]]}_{\text{negative epistemic value}} - \mathbb{E}_{q(o|\pi)}[\ln p(o)] \\
&= \underbrace{-\mathbb{E}_{q(o,s|\pi)}[\ln p(s|o,\pi) - \ln q(s|\pi)]]}_{\text{epistemic value}} - \underbrace{\mathbb{E}_{q(o|\pi)}[\ln p(o)]}_{\text{pragmatic value}} \\
&= \underbrace{-\mathbb{E}_{q(o,s|\pi)}[\ln p(o|s,\pi) - \ln q(o|\pi)]]}_{\text{epistemic value}} - \underbrace{\mathbb{E}_{q(o|\pi)}[\ln p(o)]}_{\text{pragmatic value}} \\
&= \underbrace{D_{kl}(q(o|\pi)||p(o))}_{\text{risk}} - \underbrace{\mathbb{E}_{q(o,s|\pi)}[\ln p(o|s,\pi)]}_{\text{ambiguity}}
\end{aligned}
\tag{60}
$$

The first line of Equation 16 displays an equation very much like the one used for variational Free Energy. The only difference is that *observations* have been included into the expectation. This is one of the crucial elements that differentiates Expected Free Energy, both conceptually and mathematically, from Free Energy. Because **EFE** calculates the variational free energy the agent expects to receive in the future, it is conditioned upon observations that the agent has not yet actually received, but rather the observations it expects to receive. Thus, such observations are absorbed into the expectation of the variational free energy functional, resulting in *Expected* variational Free Energy. Another important element to notice is the explicit inclusion of the *policy*, π, into the equation. Naturally, states, and in turn observations, are conditioned upon the action trajectories the agent takes (which causes transitions between states). The inclusion of the policy as a term in the density functions of the **EFE** represents this dependence.

The third line of Equation 16 displays two terms that are integral to much of the work of this dissertation. The first is the *epistemic* term. This term measures the expected difference between posterior and prior belief over states, given a policy. Thus, the larger the difference between posterior and prior beliefs, the larger the epistemic value, and, by extension, the smaller the resulting **EFE**. As its name suggests, the epistemic term measures how much *information* the agent expects to receive, with a large posterior update meaning that its beliefs have shifted by a proportionally large amount. The concept of *information*, here is exactly the same as that of Shannon Information (Shannon et al., 1949), where maximum information gain is achieved when beliefs shift from maximally imprecise (uniform distribution) to maximally precise (deterministic distribution). Thus, all else being equal, the more information a policy is expected to yield, the more an agent employing an Expected Free Energy formulation of action trajectories will favour that policy. This is a rather unique feature of Active Inference (Friston et al., 2015), and one which essentially provides the agent with an 'in-built' mechanism for directed exploration - a heuristic often used in Reinforcement Learning methods (Mann et al., 2012). The difference here is that this term is naturally derived from the Free Energy principle, and is not a

'bootstrapped' heuristic as is the case in many reinforcement learning schemes (Pathak et al., 2017). To more precisely define the nature of this exploration, we note that this is a form of *state information* exploration. Thus, this is a feature which is useful in the context of a *POMDP*, where the agent has uncertainty as to what (hidden) state it is in. The utility of this term now becomes more clear: Such a mechanism encourages the agent to seek observations that it expects will provide it with more clarity as to what hidden state it is in. The first term in the fourth line represents the same things as in the third, with the conditionals switched. These are formally equivalent as they are both an expression of mutual information between states and observations.

The second term in the third and fourth lines of Equation 16 is known in the current literature as the *pragmatic* term (Smith et al., 2021), though it is also sometimes called the *extrinsic* value. Conceptually, this measures the extent to which the observations the agent expects to receive are inline with the observations it would prefer to receive. In this sense, the agent's preferences act as a scoring function for expected observations. It is important to reiterate that the agent's preferences for observations are constructed in the form of a *prior* over observations, and this term should not be confused with the marginal likelihood. This prior is the mechanism by which a bias is incorporated into the generative model, and is essential for including the concept of reward-seeking and goal-seeking into the Active Inference framework.

The fifth line presents a form of **EFE** that is commonly used in Active Inference literature. Here the first term measures the asymmetrical difference (divergence) between expected observations and preferred observations. Minimising the difference between expected and preferred observations thus minimises this term, and in turn minimises **EFE**. When standardising for preference over an observation modality (for example if the agent has a uniform preference distribution over all observations of the modality), this term drives the agent to seek out observations that have a more unique mapping to states, and so are more informative. This is a function of the risk term that has, for the most part, been overlooked in existing literature, which mainly focuses on it just for its mechanism of measuring the similarity between expected and preferred observations.

For example, given an agent that can receive four observations: [**ob1**, **ob2**, **ob3**, **ob4**], and its preferences for these observations are $p(o) = [0.25, 0.25, 0.25, 0.25]$ respectively. If it has a choice between transitioning to four states, S_1, S_2, S_3, S_4 then the likelihood is given by the matrix:

$$p(o|s) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Columns represent S_1, S_2, S_3, S_4 and rows represent **ob1**, **ob2**, **ob3**, **ob4**. Due to the risk term, the agent will favour policies which it believes will cause it to sample observation 1, as receiving this would inform it with maximum precision that it was in S_4. In contrast, if it received observation 4, it would have a less precise belief as to what state it was in, as three states are associated with this observation. If, for example the agent is considering two policies, with the prior over each state for the policies given as: $q(s|\pi_1) = [0.5, 0.5, 0, 0]$ and $q(s|\pi_2) = [0, 0, 0.5, 0.5]$, the risk term for each, remembering that

the preferences are uniform - $p(o) = [0.25, 0.25, 0.25, 0.25]$, is then calculated as:

$$\begin{aligned} risk(\pi_1) &= \mathbb{E}_{q(o|\pi_1)}[\ln q(o|\pi_1) - \ln p(o)] \\ &= 1.61 \\ risk(\pi_2) &= \mathbb{E}_{q(o|\pi_1)}[\ln q(o|\pi_1) - \ln p(o)] \\ &= 0.69 \end{aligned} \tag{61}$$

Where $q(o|\pi) = p(o|s)q(s|\pi)$.

The smaller this term, the less the Expected Free Energy and therefore this shows that, as well as measuring the difference between expected and preferred observations, the risk term drives the agent to seek observations which best serve to resolve uncertainty as to what state it is in. It is important to reiterate here that we have shown an example where the agent's preferences for an observation are uniform. In this case, it will favour posterior distributions over observations with high entropy. There are two ways to view this. One way is to think about this through the purely mathematical lens of KL-divergence minimisation, where the agent prefers posterior distributions over observations with high entropy because the preference distribution is also of high entropy (as it is uniform). The conceptual way to view this is that when the Active Inference agent has no relative preference over observations, policies which provide the posterior over observations which have the most potential for information gain with respect to hidden states (posterior distributions with highest entropy), will be favoured. When the preference distribution is not uniform, it is difficult to conceptually disentangle the elements of preference and information from the *risk* term, and more work might be required to fully analyse this, particularly, with respect to the effects of the ambiguity term.

The second term, *ambiguity*, serves a different purpose, and measures the precision of the likelihood mapping between states and observations. Thus, this term drives the agent to select actions it believes will transition it to states that have a more deterministic mapping to observations. This functionality goes hand-in-hand with that of the risk term described above. While the risk term causes the agent to favour policies it believes will result in more unique (and so more informative) and preferential observations, the ambiguity term causes the agent to favour policies which result in states for which the agent can have higher certainty about the observations it will receive. It is easy to see how these two terms balance each other: When considering the value of policies, a policy which potentially results in states which offer more unique or preferred observations is less useful if one has less certainty as to what observation those states will actually offer. Throughout this dissertation, we will mainly focus on the form of **EFE** given by lines three and four of Equation 16, as this form presents the most intuitive way to view both the preference-seeking and information-seeking elements of the Expected Free Energy.

In addition to the preference and state-information seeking elements, the Expected Free Energy equation incorporates the element of *parameter* uncertainty into its beliefs. For example, when including

the beliefs about the parameters of the likelihood function, θ, we have:

$$\begin{aligned} G(\pi) &= \mathbb{E}_{q(o,s,\theta|\pi)}\left[\ln \frac{q(s,\theta|\pi)}{p(s,\theta|o)} - \ln p(o)\right] \\ &= \underbrace{-\mathbb{E}_{q(o,s,\theta|\pi)}[\ln p(s,A|o,\pi) - \ln q(s,\theta|\pi)]}_{\text{epistemic value}} - \underbrace{\mathbb{E}_{q(o|\pi)}[\ln p(o)]}_{\text{pragmatic value}} \\ &= -\underbrace{\mathbb{E}_{q(o,s|\pi)}[\ln p(s|o,\pi) - \ln q(s|\pi)]}_{\text{epistemic value}} - \underbrace{\mathbb{E}_{q(o,s,\theta|\pi)}[\ln p(\theta|s,o,\pi) - \ln q(\theta)]}_{\text{novelty}} - \underbrace{\mathbb{E}_{q(o|\pi)}[\ln p(o)}_{\text{pragmatic value}} \\ &= \underbrace{-\mathbb{E}_{q(o,s|\pi)}[\ln p(s|o,\pi) - \ln q(s,|\pi)]}_{\text{epistemic value}} - \underbrace{\mathbb{E}_{q(o,s|\pi)}[D_{kl}(q(\theta|o,s)||q(\theta))}_{\text{novelty}} - \underbrace{\mathbb{E}_{q(o|\pi)}[\ln p(o)]}_{\text{pragmatic value}} \end{aligned} \tag{62}$$

Here, a new term has been derived, *novelty*. This term measures the expected difference between a prior and posterior belief over model parameters. A posterior belief, in this case, results from the agent updating its model parameters upon receiving some observation. Due to this term being negative, **EFE** becomes less upon this term increasing, with *novelty* increasing as the difference between posterior and prior beliefs over model parameters increases. Thus, the agent is driven to sample observations which will cause large changes to its model. In practice, this often equates to the agent preferring observations it has sampled relatively few times, as it is less sure about the parameters behind the statistics of said observations (or the hidden states that generated them). Therefore, the imperative of maximising novelty causes a type of *parameter exploration*, resulting in the agent seeking out novel parameter updates. This naturally lends itself to effective model learning, as it encourages a 'wide' approach to parameter sampling

The agent's beliefs about model parameters can be represented by a Dirichlet distribution, which essentially allows the agent to have a distribution over its model parameters. A Dirichlet distribution is defined by *concentration parameter counts*, α, which determine the relative probabilities of different parameters.

$$p(\theta|\alpha) = Dir(\theta|\alpha) = \frac{1}{\mathcal{B}(\alpha)} \prod_{k}^{i=1} \theta_i^{a_i - 1} \tag{63}$$

where $\mathcal{B}$ is a normalising gamma function, and α_i is the concentration parameter count for parameter θ_i .

The approach of using a Dirichlet distribution, similar to that described in Section 2.5.4, has the advantage of resulting in simple learning updates for parameters of categorical or multinomial distributions (which likelihood and state-transition functions can be constructed as). Each time an outcome of the categorical or multinomial distribution is observed (for example a specific state transition), the concentration parameter count of the associated parameter is increased.

In the context of an agent learning a discrete likelihood - $p(o|s)$ or transition function - $p(s_t|s_{t-1},\pi)$, this simply equates to using concentration parameters to represent the associated likelihood or transition matrices. For example, given an environment with two states, S_1 and S_2, an agent might start out with a uniform belief over the transition function for these states. This can be represented by a matrix

of Dirichlet concentration parameters, with the initial values of these perhaps given as:

$$B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Where columns 1 and 2 represent S_1 and S_2 and rows 1 and 2 represent each of their probabilities of transitioning to S_1 and S_2 respectively. When the agent observes a transition from S_1 to S_2, it increases the concentration parameter count for this transition:

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

A similar form of updating is done for the likelihood model, with an actual distribution resulting from the normalisation of these values. Although, in this example, we increased the concentration parameter count by 1, this increase can vary, and the amount by which it increases can be thought of as a type of *learning rate* (Smith et al., 2021).

In summary, the Expected Free Energy functional, which is an extension to the variational Free Energy discussed previously, allows us to think about a system which uses the Free Energy Principle to perform adaptive behavior in the service of sampling preferential observations. With this treatment, the system takes on the form of a decision-making *agent* which evaluates sets of action trajectories in order to decide which policy will best meet its epistemic, novelty and preference imperatives.

2.8.2 Active Inference Agents

Significant work has been done to investigate the comparative performance and behavioral characteristics of artificial Active Inference agents in classical machine learning benchmark environments. (Friston, 2009; Sajid et al., 2021; Fountas et al., 2020; Tschantz et al., 2020; Millidge, 2021). Although the minimisation of Expected Free Energy is a somewhat unique, first-principled formulation which drives agent behavior, each of its terms is analogous to devices used by many agent-based machine learning frameworks (Sajid et al., 2021). In its essence, Active Inference drives adaptive action selection in the service of maximising a reward signal (despite that reward signal being constructed as an expectation over observations), which is exactly the same imperative employed by Reinforcement Learning agents (Millidge, 2021). The unique element to Active Inference is that parameter and state information gain are naturally part of this reward signal. In this way, it is relatively easy to compare Active Inference agents to other agent-based frameworks.

In particular, Sajid et al. (2021) noted that it is plausible to view Bayesian Reinforcement Learning as essentially a special case of Active Inference, where state and parameter information seeking, in the form of the epistemic and novelty terms, are removed, and only a belief-based reward function is present. In terms of performance, Active Inference agents have shown comparable results to other such Bayesian machine learning methods in benchmark environments (Sajid et al., 2021; Tschantz et al., 2020; Millidge, 2021), however the scalability of Active Inference architectures remains a significant problem, particularly due to the computational cost of the inference calculations involved and the

mechanism by which it constructs and tests policies (policies are formulated as priors over pre-defined sets of actions). In response to these issues of scalability, there have been several successful attempts to utilise deep learning implementations of Active Inference (Çatal et al., 2020; Fountas et al., 2020; Millidge, 2021). Importantly, these works have shown that Active Inference is a flexible framework that can be integrated with other machine learning schemes, such as Deep Neural Network architecture, Monte-Carlo Tree search algorithms and policy gradient objectives.

2.9 Sophisticated Inference

In the face of the scalability issues of Active Inference, a seminal approach was presented by Friston et al. (2021). This method was named Sophisticated Inference, and improves several elements of the original Active Inference algorithm. Firstly, it alleviates the problem of 'hard-coded' priors over sets of actions, which act as policies that the agent scores using Free Energy and Expected Free Energy functional. It does this by formulating the Expected Free Energy as a recursive Bellman equation, where inference over states is now conditioned explicitly upon action (and observation) rather than a policy. Thus, defining the action at a given time-step as u_t and ignoring the inclusion of model parameter inference for simplicity, the Sophisticated Inference **EFE** takes the form:

$$\begin{aligned} G(u_t, s_t) =& \mathbb{E}_{q(o_{t+1}, s_{t+1}|u_t, s_t)}[\ln p(s_{t+1}|o_{t+1}, u_t, s_t) - \ln q(s_{t+1}|u_t, s_t) - \mathbb{E}_{q(o_{t+1})}[\ln p(o_{t+1})] \\ &+ \mathbb{E}_{q(s_{t+1}, u_{t+1})}[G(u_{t+1}, s_{t+1})] \end{aligned} \tag{64}$$

While the form that is shown here is slightly different to the one in the original literature, it is functionally equivalent, and we hope it provides a clear representation of the scheme. The Expected Free Energy is formulated here as a recursive function, and represents a process inline with the Bellman optimality principle (Bellman, 1958). This recursive element, where the variational Free Energy for a given action in a state is a function of the variational Free Energy of future actions, yields a tree-like structure of branching action/state/observation trajectories, each trajectory weighted by the probability over states state-action pairs that make it up. It is useful to think of these states as *belief* states, as when the agent simulates this tree-structured look-ahead, it is essentially traversing through beliefs about states rather than actual states. This Expected Free Energy look-ahead which the agent implements in order to find an optimal set of actions can be construed as the agent thinking:

Based on my current belief about what state I could be in, if I took action u and transitioned to state x, what possible observation could I receive, and how then would my beliefs about the hidden state I was now in change? Based on this new posterior belief about what hidden states I could be in, what observations would I expect to see if I then took action y and how does that match my preferences?

Hopefully this inner monologue of the agent is helpful in displaying the nested belief structure Sophisticated Inference induces. Within the context of the agent attempting to plan optimal actions (with respect to minimising **EFE**), it imagines how its beliefs about hidden states would evolve, were it to take certain actions and receive certain observations. This form of thinking is truly counterfactual, with the agent simulating trajectories of actions and (actual) states, and imagining how its *belief states*

would change were it to actually be on this trajectory. Thus the agent holds counterfactual beliefs about its own beliefs, giving rise to the term 'Sophisticated'.

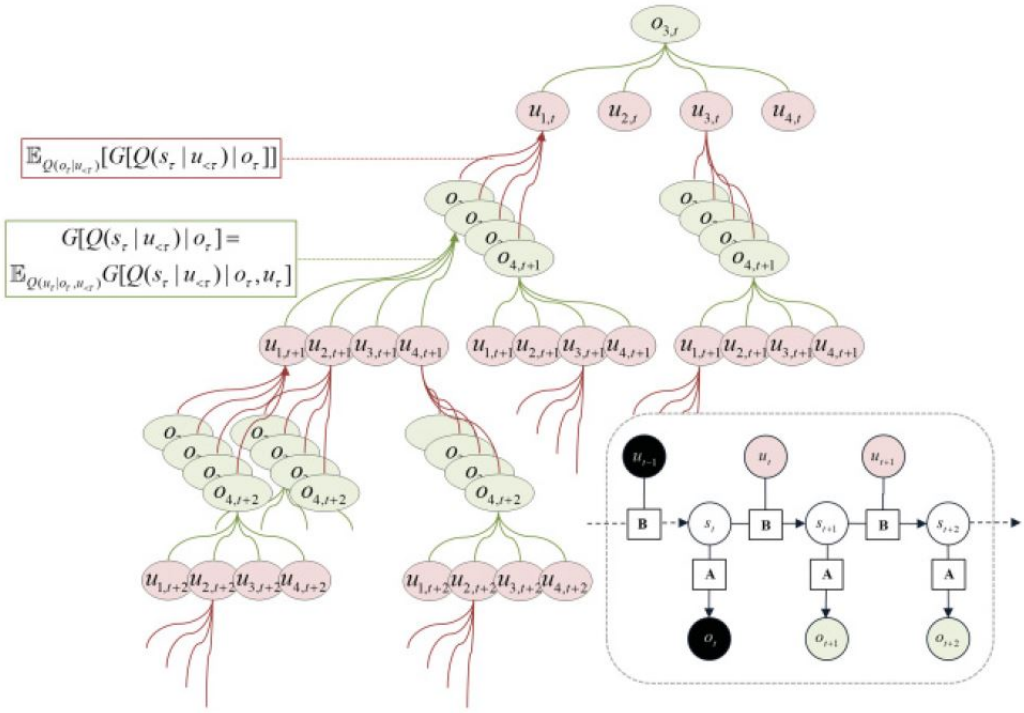

Figure 6: (Friston et al., 2021) A depiction of the branching recursive search used by the Sophisticated Inference agent. u indicates action, and O an observation. The structure in the lower right is a *factor graph*, a common device used in Active Inference literature to show the relationships between components of the agent's generative model

Figure 6 shows the tree-search which the agent implements in order to evaluate trajectories of actions. Here we see that the agent starts with a realised observation, from which, based on its beliefs about hidden states, it implements a simulated trajectory over possible actions, which in turn lead to potential observations and so on, up to some defined depth of the trajectory. While this process naturally represents an *exhaustive* search, the algorithm proposed by Friston et al. ((Friston et al., 2021)) incorporates a 'pruning' element, whereby a searches over trajectories of actions and observations are stopped given certain conditions. The first of these conditions is if the prior probability of a next hidden state, given the current belief state and an action, is below some threshold (set to $p = 0.16$ in the original paper). In this case, the occurrence of a transition to such a state is too unlikely, and so the agent disregards it. The second condition which elicits pruning is the case where the Expected Free Energy of an action is below some relative **EFE** threshold, when compared to the other possible actions. Similarly to the treatment of unlikely states, if this is the case, action (and the future trajectories it could lead to) is ignored.

While no formal comparison has been done between Sophisticated Inference and other non-Active Inference model-based Bayesian algorithms, Friston et al. compared its performance in simple environments to 'unsophisticated' inference. In these simulations, it was shown that the Sophisticated

Inference agent achieved, in many cases, better results when compared to its counterpart, particularly when exploration was required. However, it is also noted in this paper that more work is needed to evaluate Sophisticated Inference's capacity in larger, more complex environments, to fully determine its viability for improving the scalability of Active Inference.

2.9.1 Affective Active Inference

An on-going area of novel research centers on the ways in which we can represent subjective feeling, formally known as *affect* (Solms, 2019), in artificial agents. This is a notoriously difficult issue to investigate and ultimately leads back to the philosophical underpinnings of the *Hard problem of consciousness* (Chalmers, 1995). The issue of conscious feeling in artificial agents, despite being in its infancy, has recently gained a large amount of interest from a wide variety of disciplines (Tononi, 2004; Chella and Manzotti, 2007; Boyles, 2012; Koch et al., 2016; Seth, 2018; Solms and Friston, 2018; Solms, 2019; Meissner, 2019). In particular, effort has been made to distinguish between the neural correlates of consciousness and determining what consciousness, in its essence, actually is, and why it is present at all (Chalmers, 1995; Solms, 2019). Of specific relevance to this dissertation are works that have recently emerged which have the goal of viewing affective, 'feeling' agents through the lens of the Free Energy Principle (Solms and Friston, 2018; Solms, 2019). These arguments focus on articulately framing the notions of affect, attention and perception as mechanisms to achieve precision optimisation. Specifically, precision is used to weight prediction errors and so direct how we perceive and experience the world. It is argued that this precision modulation is itself synonymous with affect, due to it directing selective arousal and attention (Clark, 2013; Hohwy, 2013; Solms and Friston, 2018). Integral to the notion of affect, is the concept of homeostatic drives (Parvizi and Damasio, 2001; Solms, 2019) which need to be selectively prioritised with respect to both current and predicted future observations, in order to maintain the homeostatic 'settling-point' of a system. The congruence between this view, and that of an organism minimising free energy to resist dissipation, as described in Section 2.7, offers a strong case for the notion that affect, attention and perception are functions of an entity performing Active Inference.

Within the specific domain of Active Inference, affect has been used as an explanation for emotional valence (Hesp et al., 2021b), whereby deep hierarchical Active Inference agents infer the state of said valence values based on the expected precision of their action model. In addition to this, Sophisticated Inference has been used as a device to explore the concept of an agent whose affective states (defined by arousal, valence, and context sub-states) change based on its predictions about how its future beliefs might evolve (Hesp et al., 2021a).

While work on exploring *affective* Active Inference agents is in its infancy, it appears to present a promising framework through which to investigate the representation of subjective feeling - combining the first-principled basis of the Free Energy Principle with the theory of homeostatic optimisation in living systems to create an articulate and novel approach to the infamous problem of consciousness.

2.10 Summary

In this section, we presented a formal framework for modelling an agent which uses the Free Energy Principle to achieve goal-directed behavior. The conceptual mechanism by which this type of behavior is achieved is subtly different from other machine learning paradigms such as Reinforcement Learning. Instead of an explicit reward signal which the Reinforcement Learning agent optimises behavior for, the Active Inference agent deliberately acts to sample *expected observations* from the environment. Therefore, this creates a scheme where the agent works to resolve errors between what it expects to observe and what it actually observes, naturally creating a scenario analogous to the process of predictive coding.

To achieve the ability to compute how variational Free Energy might change in future time-steps, the concept of Expected Free Energy is introduced. This mathematical object differs from variational Free Energy in two ways. Firstly, observations become random variables, included into the expectation of the functional. Secondly, a prior over observations is included into the agent's generative model. This prior essentially implements a biased expectation over observations, and allows the Active Inference agent to favour certain observations over others.

The Expected Free Energy functional can be decomposed in various ways, with each of these decompositions representing different conceptual imperatives for the agent. One decomposition results in the *epistemic* and pragmatic terms, with the epistemic term driving the agent to maximally update its beliefs - and so sample observations it expects will cause large posterior belief updates - and the pragmatic/extrinsic term which drives the agent to sample observations which best match its preferences. The other common decomposition is into *risk* and *ambiguity*. The risk term drives the agent to both seek observations which are congruent with its preferences, as well as seek observations which provide state information. The ambiguity term causes the agent to seek out states which have a precise mapping to observations, allowing the agent to have greater confidence about what observations it would actually receive were it to visit said states.

In recent years, there has been growing interest in exploring the scalability and comparative properties of Active Inference agents. This has resulted in cutting-edge experimentation into the ways Active Inference can be algorithmically represented and integrated with other machine learning paradigms. However, the comparative potential of Active Inference agents remains a topic of ongoing investigation. In response to some of the issues around scalability and algorithmic design that is present in Active Inference agents, Sophisticated Inference emerged to frame the Active Inference algorithm as a recursive policy search. Here the agent 'imagines' counterfactual trajectories of actions, observations and belief states, and formulates a Bellman-optimal policy based on the Expected Free Energy evaluation of these trajectories. In this sense, the agent forms beliefs about how its own beliefs might change conditioned on action. This nested belief structure allows for a type of sophisticated thinking when compared to the original Active Inference agent. While Sophisticated Inference has shown promise in its comparative capacity, it is a recent addition to the Active Inference framework, creating opportunity for additional investigation into its implications for the field.

One of the more interesting applications of Active Inference, and the Free Energy Principle in general, is to the field of the Science of Consciousness, specifically in constructing mechanisms by which to

represent and interpret *affect*. Although in its early stages, this has shown promise in providing conceptual and computational progress on understanding the difficult scientific and philosophical ideas circling the field. Specifically, Active Inference agents have the ability to modulate precision in order to achieve what appears to be arousal and attention. This modulation of precision, itself, could be seen as being synonymous with affect, in that it represents a divergence from the agent's predicted expectations.

The next section presents the methodology used to implement the practical work of this dissertation.

3 Methodology

The ideas and concepts discussed in the previous sections have led in the direction of describing an actual entity which implements an Active Inference algorithm to navigate an environment. To solidify this, the practical work accompanying this dissertation was achieved via the use of an artificial agent, which was used as a mechanism to simulate decision-making, learning and inference. Based on this agent, the resulting experimentation revolves around four core investigative aims:

1. A comparison between Active Inference and Bayesian Reinforcement Learning algorithms in a multi-objective dynamic environment.

2. An attempt to combine both model-free Reinforcement Learning and Active Inference in a hierarchical set-up, in a way that acts as a proof of concept for the viability of the integration of both domains.

3. An investigation into the effect of omitting and including the 'epistemic' term of the Expected Free Energy equation in the Active Inference algorithms, as well as the effect of this in combination with varying preference distribution precisions.

4. Investigating the effect of using a 'sophisticated' belief propagation of model parameters, where the agent holds beliefs about how its future beliefs about model parameters might change.

3.1 Environment Details and Agent Model

As outlined in section 2.9, although a variety of environments have been used to test Active Inference's performance and compare it to other machine learning algorithms (Sajid et al., 2021; Millidge, 2021), these environments mostly focus on a specific elements of classical machine learning behavior, such as exploration, model learning, or reward optimisation. The testing iterations used for this dissertation, in contrast, attempt to create scenarios where multiple combinations of such elements are in-play, with the general aim being the investigation and comparison of Active Inference agents' behavior and performance in complex belief-based dynamic environments.

Two such environments were constructed to conduct these investigations, one essentially a smaller version of the other. While a smaller environment might seem redundant, it acts as a useful device to test concepts which are independent of environment scale. An ever-present problem when solving POMDPs is the 'curse of dimensionality' (Duff and Barto, 2002) where computation time increases exponentially with the planning horizon. Therefore, for the most part and where possible, smaller-scale, detailed concepts were tested and compared in the smaller environment, while broader patterns of agent behavior were explored in the larger one. Both smaller and larger environments are discrete, symmetrical 'grid worlds', comprising of 5-by-5 and 10-by-10 blocks respectively.

The agent, within these environments, has five different possible actions: move up, move down, move left, move right or remain in the same position. The transition function for the positional state (first state factor), conditioned on these actions, is known by the agent and is deterministic. If the agent is at a border of the grid world, any action that would otherwise move it out over the border causes it to remain in the same position.

Of particular interest in much experimental Active Inference literature is the idea of a hidden 'context' state which the agent must attempt to infer (or sometimes guess) in order to make decisions which are optimal with respect to its internal preference distribution, and so reduce Expected Free Energy (Smith et al., 2021). This context state of the environment is fully or partially revealed upon the agent receiving (sampling) some observation - often given by the agent transitioning to a particular 'clue-giving' state within the environment. This setup is useful due to it posing an explicit explore-exploit dilemma in the context of a POMDP. The behavioral patterns of the agent, then, take on two general forms, with it either attempting to sacrifice immediate potential reward and visit the 'context-revealing' state, in order to be more sure about how best to achieve reward in the future, or, more greedily, to try to exploit resources earlier on with a lower precision of belief over the context it might be in.

The environments used for this dissertation take inspiration from this paradigm. Each includes four contexts (which can be thought of as *versions* of the environment) which define the position of a group of three categorically different 'resources', which act as sources of three different rewards for the agent, and so define a multi-objective environment. For thematic purposes, these resources are labeled food, water and shelter.

To add some complexity, at each timestep, each context state probabilistically either remains as it is, or transitions to another context. For example, such a transition function takes the form of a matrix which describes the transition probabilities between each context. Here each column represents a specific context, and each row represents the probability that it will transition to another context (or itself).

	Context 1	Context 2	Context 3	Context
Context 1	0.8	0	0	0.2
Context 2	0.2	0.8	0	0
Context 3	0	0.2	0.8	0
Context 4	0	0	0.2	0.2

Along with the two state factors described here (position and context), which are explicitly included as part of the agent's generative model, the agent also registers the number of time-steps it has been without each of the three resources. These measurements function as 'internal' states of the agent and are not included in the agent's generative model, as they are not associated with an accompanying observation modality. Therefore, the agent, at all time-points, has direct access to these states and does not have to infer them via observations. Although it would have been possible to absorb these states into the agent's generative model, we decided against it due to the concept of an agent attempting

to infer its own internal states (as opposed to external environment states) representing a separate research initiative which deserves more attention than what could be provided in this dissertation. Despite these internal states not representing formal state factors with modeled transition functions, they do have an implicit transition function which is known by the agent in all trials. This is simply expressed as:

$$
\begin{aligned}
&T_{resources} = T_{resources} + 1 \\
&\textbf{if } S_t \text{ is in } S_{resources} \textbf{ then} \\
&\quad T_{resources}(S_t) \leftarrow \text{ResetResourceTime}(S_t) \\
&\textbf{end if}
\end{aligned}
$$

The external positional state which 'reveals' the current context state is always positioned in the middle of the grid world, and can be thought of as a hill that the agent climbs in order to survey the land around it. This specific implementation of the context state is somewhat novel when compared to the practical implementations of existing literature, which mostly make use of environments where the agent, at any point and any state, can transition to the 'context-revealing' state in one time-step (for examples see(Friston et al., 2021) and (Smith et al., 2021)). This is in contrast to the environments used for this dissertation, in which the agent must plan optimal trajectories of multiple actions to reach the context-revealing state, and so must incorporate this planning into the greater goal of optimally navigating the environment for resources. Ultimately, this yields a slightly more complex setup than what has previously been used.

Depending on the testing iteration, as will be described in more detail below, the transition function of the context state is either known by the agent, or must be learned. At the beginning of each trial, the agent starts at the positional state given by the left-most column and middle row. The initial context state is randomised with the agent having an initial uniform belief over said context for all trials

There are two observation modalities which the agent can receive from the environment. The first consists of four possible observations: **Empty**, **Food**, **Water** and **Shelter**. The second observation modality is structured around the context state, with the agent receiving 5 possible observations: **In Context 1**, **In Context 2**, **In Context 3**, **In Context 4** and a **No Context** observation, which, as the name suggests, gives the agent no information as to what context the environment is in. Observations 1-4 of this modality are provided by the hill state (depending on the context the environment is in), while all other states give observation 5 (**no context**). As is often the case with variational Bayesian inference frameworks, the different state factors are factorised, with the likelihood of a context observation given by

$$p(\textbf{context observation} \mid \textbf{positional state}, \textbf{context state})$$

In all testing iterations except for the last, this *likelihood* component of the agent's generative model is known, deterministic and accurate. Practically, this means that the agent knows, with maximum certainty, what observations it will receive conditioned on the states it is in. In the last testing iteration, as will be presented, the agent is unaware of the resource locations, conditioned on the context, and so must learn which locations of resources map to which context.

Mathematically, for the 5x5 environment, modalities 1 and 2 (resource observation and context observation) take the form of 4x25x4 and 5x25x4 tensors respectively. With this structure, the first dimension represents the possible observations for the modality, the second dimension the positional state (state factor 1) and the third the context state (state factor 2). For example, for the second observation modality and for the third dimension set to context-state 1, the likelihood matrix is given as:

$$A\{2\}(:,:,1) = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

Here the columns represent the positional state (factor 1) and the rows represent the 5 different possible context-related observations of observation modality 2. From this, we can see that when the agent is in position 13 (the hill state), given that the context-state is 1, it will receive a 'context 1' observation with 100% probability, while all other positional states yield a 'no context' observation with 100% probability (100% probability here being indicated by the number 1).
Figures 6 and 7 show graphical depictions of the environments 1 and 2 in their entirety.

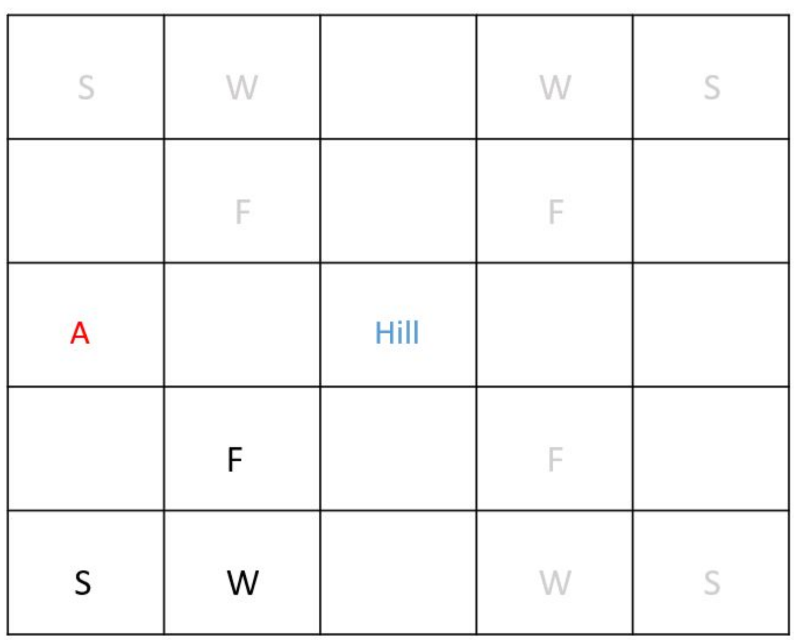

Figure 7: **Environment 1 -** A 5-by-5 gridworld with structured resource placements for each context. The bold letters indicate the resources of the actual randomised context at the first time-step, whereas the transparent letters indicate the resource locations for the other possible contexts

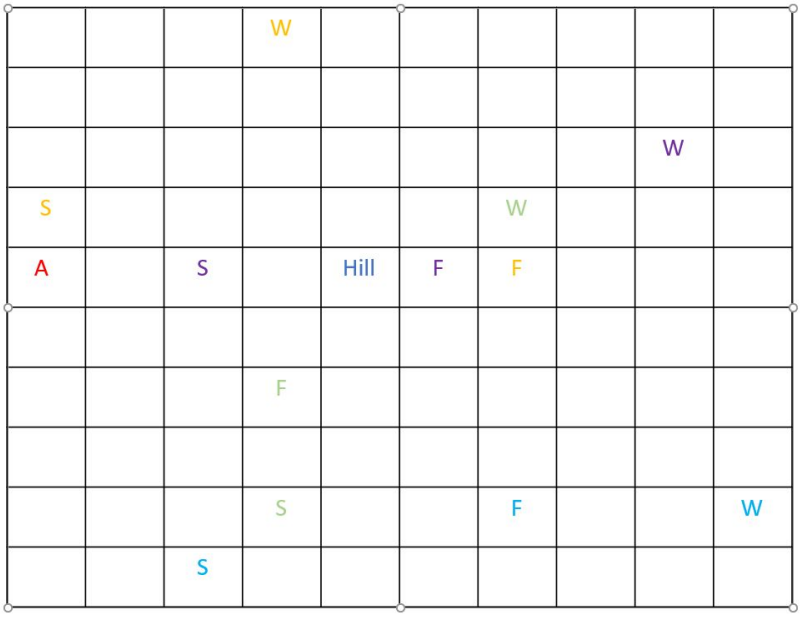

Figure 8: **Environment 2** - A 10-by-10 grid world with randomised resource placements for each context, with each one of the four contexts represented by a different colour.

In Figure 7, the resource locations for each context are grouped closely together, with each group symmetrically positioned around the environment. Thus, when an agent moves to a 'pocket' of resources, based on the current context, there is little complex planning needed to navigate said pocket. Figure 8 shows a more complex set-up, where the locations of the resources for each context are randomised. Therefore, based on the belief of the agent about what context the environment might be in, more complex planning must take place to determine how it should visit each of the resources, taking into account what next context the current context might transition to, as well as the agent's current and predicted resource-related needs. These 'needs' are ultimately determined by the reward function, which is based on the number of discrete time-steps the agent has gone without visiting each resource category. This reward function is defined by Algorithm 1.

Algorithm 1 Multi-objective reward function

function PREFERENCES($T_{resources}$, $L_{resources}$, penalty)
  $P_{empty} \leftarrow$ -1
  for each resource in [water, food, shelter] **do**
    $P_{resource} \leftarrow T_{resource}$
    if $P_{resource} \geq L_{resource}$ **then**
      $P_{empty} \leftarrow$ penalty
      $P_{resources \neq resource} \leftarrow$ penalty
    end if
  end for
return P
end function

Here $T_{resources}$ is the count of time-steps the agent has gone without resources and $L_{resources}$ is set of

time-limits for each resource. Essentially, the agent's individual preference for each resource increases the longer it has been without said resource. Additionally, each resource has a distinct time-step limit. If the agent goes over one of these limits, it incurs a large negative reward (penalty). In the context of this simulation, this penalty represents the death of the agent and the trial ends. All transitions to empty squares incur a pre-normalised value of -1. This multi-objective reward function was structured to be as simple as possible, and follows the forms classically used in Reinforcement Learning grid world environments (Sutton and Barto, 2018).

The form of this reward function ultimately results in a *dynamic* preference distribution. This is a feature which has not yet been investigated in previous works of Active Inference literature, in which the preference distribution is either entirely static, or only a function of time (Sajid et al., 2021; Friston et al., 2021; Smith et al., 2021; Tschantz et al., 2020). In contrast, for all simulations run in this work, the preference distribution of the agent at any given time-step is a function of the agent's current internal states (time since each resource). Seen through the lens of Active Inference, this creates the interesting conceptual scenario where the agent's future (predicted) preferences are implicitly defined by its own policy. Based on the actions it takes, its preferences at each time-step will differ. This creates an almost circular structure, where the agent must determine a policy which best satisfies the preferences, at each time-step, that the policy itself induces.

Table 1 summarises the elements of the agent's model and displays what features/parameters are known/unknown.

Table 1: A display of the components which make up the agent's generative model

State Factor	Partially Observable	Initial State	Transition Function	Part of Generative Model	Observation Modality
Position	No	Known	Known	Yes	None
Time since food	No	Known	Known	No	None
Time since water	No	Known	Known	No	None
Time since shelter	No	Known	Known	No	None
Resource positions	No	Context-dependent	Known	Yes	1
Context	Yes	Unknown	Known/Unknown	Yes	2

3.1.1 Hierarchical State-Space

As part of the setup for the different environments used in this experimental work, there was a general aim to integrate Reinforcement Learning with Active Inference in a way which both preserved the concept of biological plausibility and offered computational efficiency.

The specific implementational goal for this work was to have Active Inference operate in a higher-level state-space, and, by extension, construct higher-level policies within this state-space. Each step of these higher-level policies would then act as a directive, which would dictate how a policy for the lower-level state-space, determined by an RL algorithm, should be formulated.

This setup is appealing for two main reasons. Firstly, it allows for a factorisation of the state-space into hierarchical levels, which has shown to significantly improve performance in many tasks (Pateria et al., 2021). Secondly, it aligns with an interesting way of viewing how organisms might operate in terms of

higher-level goals and lower-level actions in order to achieve those goals (Morris and Cushman, 2019). For example, it is plausible to view the event of a person walking two steps forward as being divided into two broad yet connected modes: One involving some higher-level belief-based planning, which serves to cover the larger scale contexts and beliefs about *walking forward*, and the other consisting of lower-level reflexive muscular movements, which employ directives from the higher level, and which would ultimately cause the individual to lift their foot up and forward. Of course the division of this view into only two hierarchical domains is arbitrary (in reality it would be many more), and while this dissertation makes no strong claim about the validity of such a view, the setup described above acts as a potential proof of concept of such a scheme existing - at least for an artificial agent.

To implement such a proof of concept, for all environments used, each block of the grid world, representing the positional state, was further divided into a 5x5 state-space, as is shown in Figure 9. With this setup, the higher level state-space, represented by the larger squares of the grid world, is solved via various model-based inference algorithms (as presented in the next section), while the lower-level state-space is solved via model-free Q-Learning. At each time-step of the trial, the agent's higher-level action, determined by the higher-level policy, is used to direct the 'goal' for the lower level state-space. For example, if the agent wishes to move one state to the right of its current state, the 'move right' directive is sent to the lower-level scheme. In this scenario, a reward of $+1$ is received by the agent if it takes a right (lower-level) action in one of the right-most states of the smaller environment, as this will cause the agent to transition into the next higher-level state. For all other state-action pairs, a reward of -1 is received.

Importantly, the agent must learn the correct directive-state-action values. This is achieved via repeated episodes of the standard Q-Learning algorithm at each higher-level time-step. 100 episodes of Q-learning were conducted at each of these time-steps for all trials, until convergence of all directive-state-action values was achieved. The number of Q-learning episodes (100) is a hyper-parameter that was chosen following heuristic experimentation. 100 was found to be the number of episodes which ensured the agent's convergence on an optimal Q-learning policy for a given directive. A discount factor of $\gamma = 0.7$ was used for all episodes.

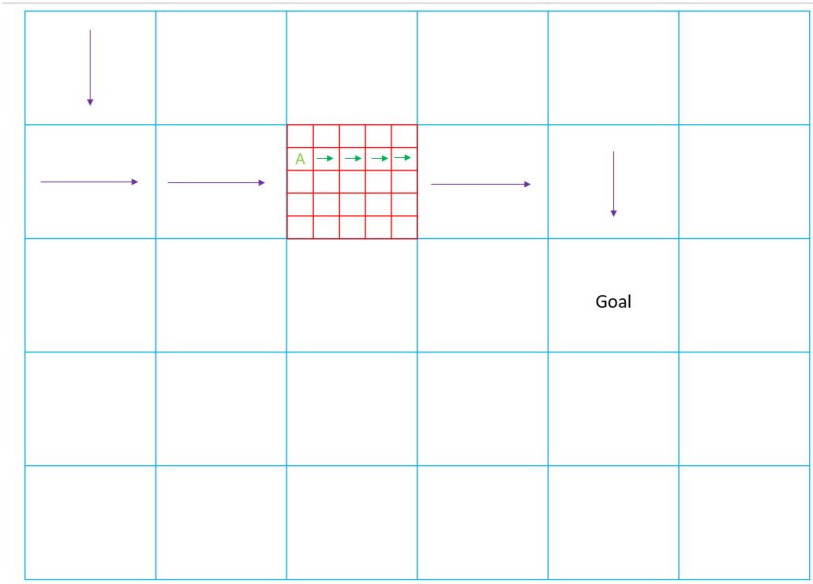

Figure 9: A graphical depiction of how each state of the higher-level state-space is further divided into a 5x5 smaller level state-space, which the agent learns and navigates via model-free Q-Learning

It is important to note that although the time taken by the agent to traverse the smaller state-space did not act as time-steps which incremented the agent's time since each resource count, the time-steps of the lower-level state-space were implicitly included in the overall reward evaluation of the algorithm, as each step that did not transition the agent to the correct higher-level state incurred a cost of -1, and so encouraged the agent to reach the next higher-level state as quickly as possible.

3.2 Agent Algorithms

Four different core agent-based algorithms were tested and compared within the two specified environments. Although each of these algorithms represent different solutions to solving POMDPs, as we will see, there exist many structural and functional overlaps between them. A fundamental algorithmic feature used in three of the four algorithms - Sophisticated Inference, Bayesian Reinforcement Learning and Bayes-adaptive Reinforcement Learning - is the use of a recursive tree search over possible future belief states. This is the same type of search mentioned in Section 2.9, and is predicated on an initial (belief) state. From this initial state, the agent simulates all possible trajectories of actions, and the transitions between states as a result of these actions, up to some planning horizon The general structure of such a search tree used by these algorithms is shown in Figure 9. Importantly, the nodes of the tree are *belief* states, ultimately stemming from the agent's prior beliefs upon initiating the search function. The observations too, are observations that the agent believes it could receive given the trajectory of belief states and actions. This simulated planning done by the agent is different from every 'real' time-step of the trial where the agent actually takes an action.

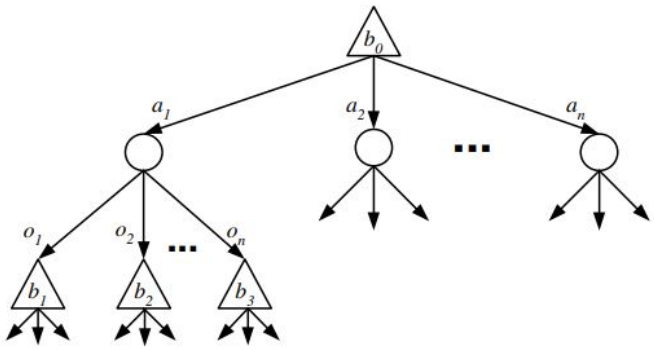

Figure 10: (Paquet et al., 2005). A recursive search tree, where nodes are belief states which, conditioned on action, generate observations which in turn map to subsequent belief states and so on. Circles represent hidden states that the agent transitions to, while triangles represent beliefs the agent forms based on observations generated by those hidden states.

Featured in all algorithms using the recursive tree search, is the technique of memoization (Michie, 1968; Norvig, 1991). Due to the exhaustive nature of this type of search, such algorithms suffer from exploding dimensionality, resulting in only very small planning horizons being computationally feasible. Memoization is a mechanism which drastically reduces the number of trajectories explored. Within a tree search such as the one we just described, there are many repetitions of nodes. Memoization ensures that, if a node has already been evaluated, any subsequent search which encounters such a node uses the result of the previous evaluation. Practically, this works by storing the evaluation of state-value information in memory. Subsequently, if this state (belief) node is encountered along the path of another search trajectory, the value of the node can be returned to the parent node without needing to traverse the tree further, as this has already been done in a previous simulated trajectory. Given that the state-space for the larger 10x10 environment is formally defined as

$$position \times t\_food \times t\_water \times t\_shelter \times context$$

with time since food, water and shelter capped at 30, the number of recursive function calls made with and without memoization from t = 1 and for different search depths is shown below.

Depth	Function calls with memoization	Function calls without memoization
0	1	1
1	17	21
2	53	421
3	117	8421
4	257	160421

Table 2: A comparison of the number of recursive function calls with and without memoization

As is evident, memoization is extremely effective in reducing the number of recursive function calls required. A drawback to memoization is the significant memory requirement. However, if needed, this can often be solved via various approximation techniques, such as Coarse Coding (Sutton and Barto, 2018), a form of linear function approximation, or Artificial Neural Networks (Abiodun et al., 2018).

In-line with the theme of this dissertation, the Active Inference algorithms are the central focus of this practical work. As discussed in Section 2.9, the two broad types of Active Inference algorithms used in the testing iterations (Sophisticated Inference and 'Vanilla' Active Inference) differ in their fundamental construction and use of policies. While vanilla Active Inference offers the strength of biological plausibility and a computational account of neural process theory (Friston et al., 2006, 2017), it is not well-suited for actual agent-based decision making in the types of environments common in standard machine learning benchmarks, and scales poorly in larger environments due to its reliance on pre-trial manual policy construction and the need for separate posterior beliefs over states for each of these policies (Friston et al., 2021; Smith et al., 2021). On the other hand, the Sophisticated Inference algorithm takes on a form in-line with the *Bellman-optimality principle* - using recursive tree searches to construct online belief-optimal policies up to some time-step horizon. This technique avoids some of the mentioned problems of vanilla Active Inference and is more closely related to other Bayesian machine learning methods.

Two such Bayesian methods were implemented here in order to act as comparative devices to the Active Inference algorithms. These are a simple model-based Bayesian Reinforcement Learning method (for trials where the model is known a priori) and a form of Bayes-adaptive Reinforcement Learning (for trials where the transition function of the context state factor or the likelihood is unknown). These two methods were specifically chosen due to their current extensive use in solving POMDP problems (Ghavamzadeh et al., 2015), as discussed in section 2.6. Both Bayesian Reinforcement Learning methods use a real-time search scheme, over a subset of belief-states . This subset populated by belief states that are reachable from the start state up to some trajectory horizon, avoiding the often intractable computational requirement of a full dynamic programming approach (Paquet et al., 2005). 'Reachable' here is defined by some level of proximity to the initial belief, in this case a defined search horizon. With all heuristics stripped away, this form of belief-state search is exactly the same as the search used in sophisticated inference, and so further standardises the two approaches to allow for better behavioral and performance comparison. Finally, we implement a new algorithm which takes inspiration from elements of Bayes-adaptive methods and integrates these into the sophisticated inference procedure. We have named this Propagated Parameter Belief Search, due to its feature of simulating parameter updates during the agent's recursive look-ahead procedure. Each of the four recursive search algorithms mentioned use a version of Algorithm 2:

Algorithm 2 Core trial loop

Input: True Likelihood function A – True transition function B – Agent likelihood model a – Agent transition model b – Initial-state distributions D – Agent initial-state model D – resource limits l – Resource locations $S_{resources}$ – Number of trial time-steps T

$T_{resources}$ = [0, 0, 0]
resources = [food, water, shelter]
$\pi \leftarrow$ InitialiseRandomPolicies(T)
$T_{resources} \leftarrow$ [0, 0, 0]
resources $\leftarrow$ [food, water, shelter]
while t $< T$ & $T_{resources} < l$ **do**
    if t ≥ 1 **then**
        states{factor} $\leftarrow$ B(states(t-1), action)
        posteriors $\leftarrow$ b(posterior(t-1), action)
        UpdateConcentrationParameters(*posterior*, a, b, *observation*)
    else
        true_states(t) $\leftarrow$ Sample(D)
    end if

    $T_{resources} = T_{resources} + 1$
    if S_t is in $S_{resources}$ **then**
        $T_{resources}(S_t) \leftarrow$ ResetResourceTime(S_t)
    end if
    $h \leftarrow$ CalculateHorizon(t_resources)
    observation $\leftarrow$ SampleObservation(true_states, A) true_t $\leftarrow$ t
    [G, *posterior*]$\leftarrow$ ForwardTreeSearch(*posterior*, a, b, *observation*, t_resources, t T)
    action $\leftarrow$ min(G)
end while

The *UpdateConcentrationParameters* function increments the Dirichlet counts of the agent's model as described in Section 2.8. The *ForwardTreeSearch* function is given as:

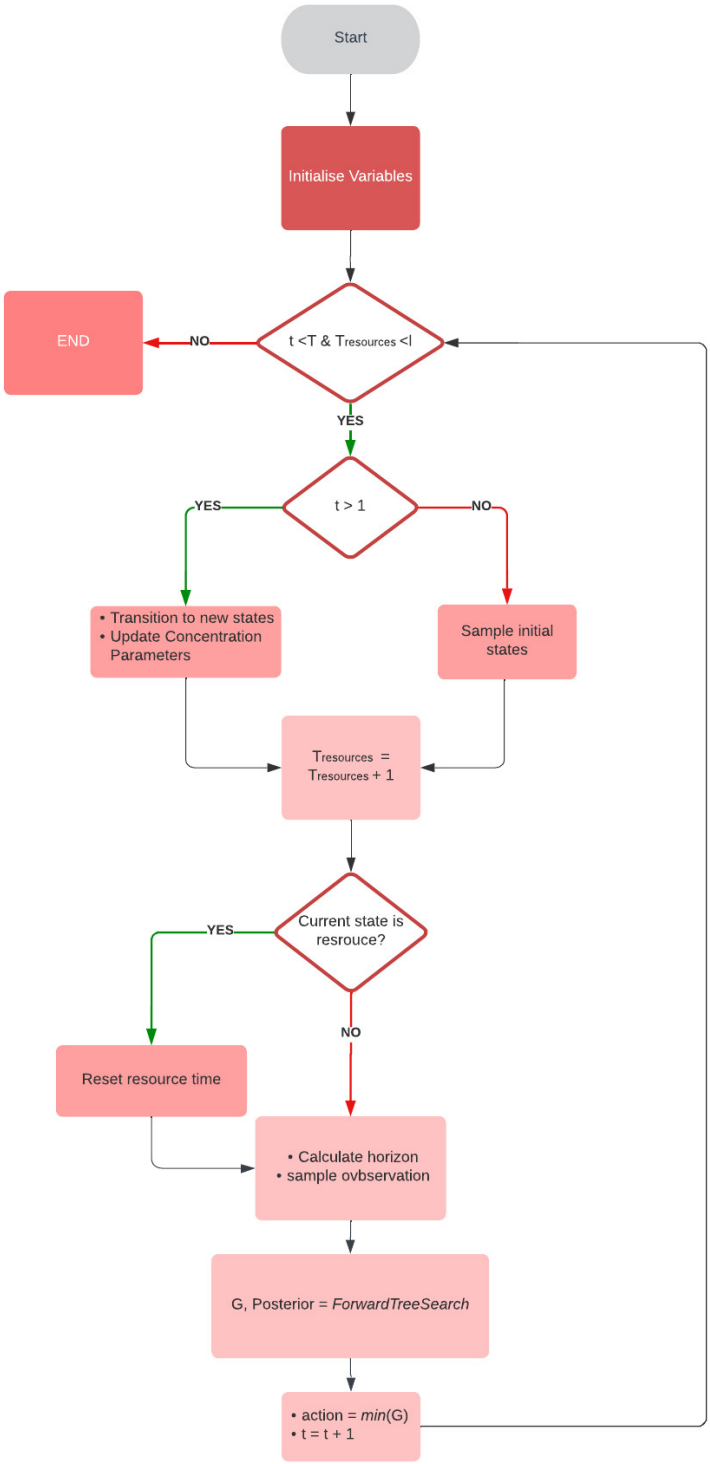

Figure 11: (**Flow Diagram depicting the Core Trial Loop**)

```
function ForwardTreeSearch(posterior, a, b, observation, t_resources, t, true_t, T, W, h)
    if t > true_t then
        posterior = CalculatePosterior(posterior, b, a, observations)
        if posterior(t) is in resource_locations then
            t_resources ← ResetResourceTime(posterior(t), resource_locations)
        end if
    end if
    for each action do
        Q(action) = b(posterior(t),action)
        predicted_observations ← a(Q(action)
        G(action) ← ExpectedFreeEnergy(predicted_observations, t_resources, W)
    end for
    if t < h then
        actions ← ViableActions(G)
        for action in actions do
            posterior(t + 1) ← Q(action)
            states ← Q(action)
            states ← LikelyStates(states)
            for state in states do
                observation ← SampleObservation(state, A)
                [G, posterior ← ForwardTreeSearch(posterior, a, b,
                observation, t_resources, t + 1, true_t, T, W, h)
                S ← softmax(G) * G
                K(state) ← S
            end for
            G(action) ← K(states)*states
        end for
    end ifreturn [G, posterior]
end function
```

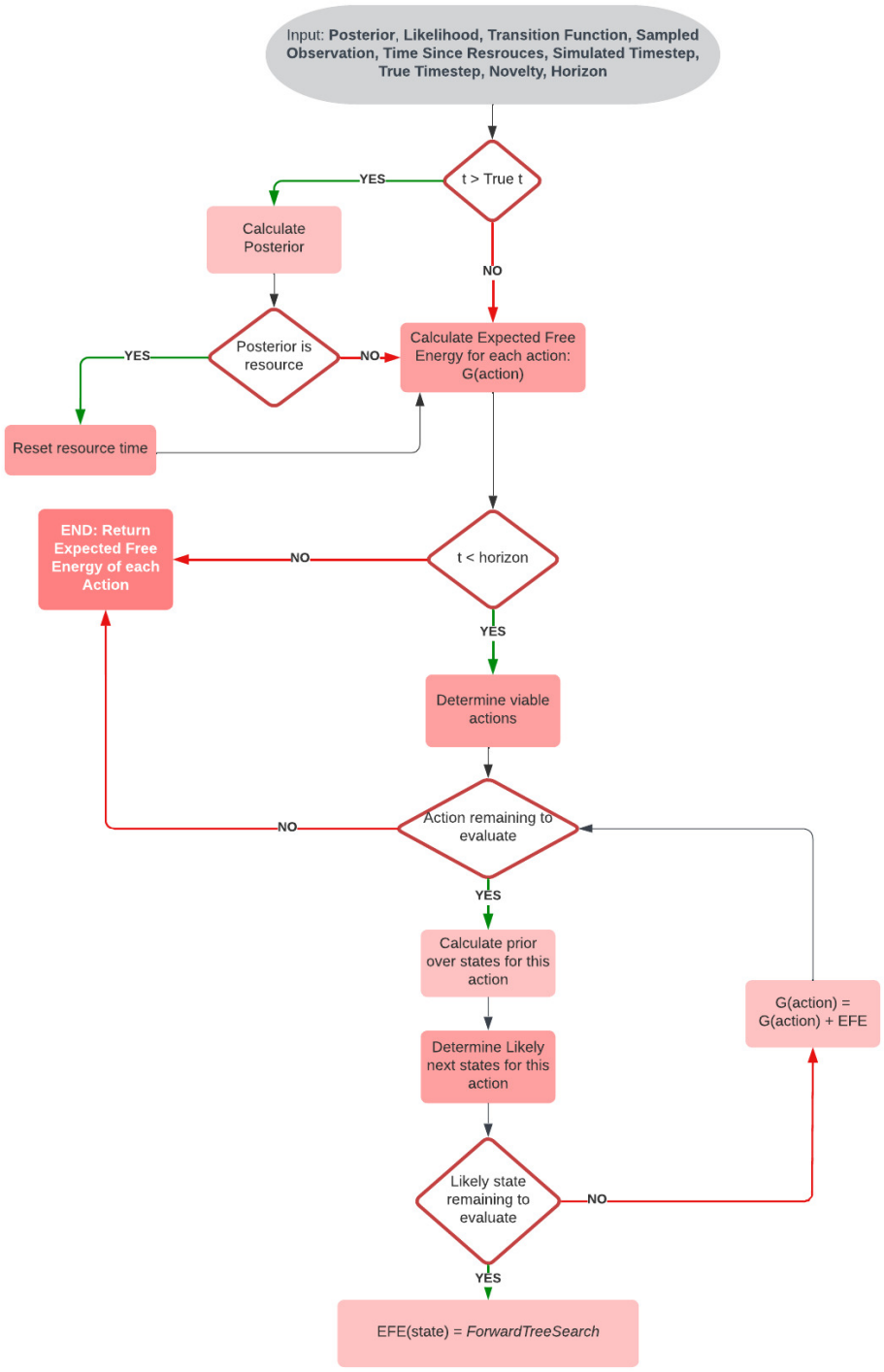

Figure 12: (**Flow Diagram depicting the Forward Tree Search Function**)

A small element to take note of here, and in all algorithms, is the *CalculateHorizon* function. This allows for varying planning horizons and is based on the time since visiting each resource. The idea here is that if the agent is x time-steps away from dying due to a resource limit, there is no point in it looking further than x time-steps ahead. Conversely, if the agent is many time-steps away from dying,

it also does not need to look very far ahead, due to it being in a 'comfortable' state. The number of look-ahead steps thus follows a parabola shape, where the furthest look-ahead happens when the agent is in an intermediate state between complete 'satiation' and immediate death. Thus the equation for the look-ahead is parabolic and formulated as follows:

$$H = -a * min(l - T_{resources})^2 + b * min(l - T_{resources})$$

With l being the resource time-limits, $T_{resources}$ the number of time-steps the agent has been without each resource, and a and b being constants, the value of which are hyper-parameters and depend on the resource time limits.

The difference between the recursive Active Inference algorithms (Sophisticated Inference and PPBS), and the Bayesian Reinforcement Learning algorithms (Bayesian RL and Bayes-adaptive) mainly appears in the reward function that is used. Active Inference methods make use of the Expected Free Energy functional, which includes directed state and parameter exploration via the *epistemic* and *novelty* terms respectively. The Bayesian methods use a reward function based only on the multi-objective preference function shown in **Algorithm 1**, with no extra exploration heuristics added. The choice to not include such heuristics for these algorithms is based on the idea that the exploration functionality of Active Inference is 'baked' into the Expected Free Energy functional, and is naturally derived from the mathematical formulation of the Free Energy Principle. Therefore, these exploration factors are not added-on heuristics, as would be the case had we used some form of exploration procedure for the Bayesian RL methods. The idea was to compare these algorithms in as much of a 'base' state as possible without added procedures which would affect behavior. As we will see, a core difference between the PPBS or Bayes-adaptive methods and the Sophisticated Inference algorithm comes in the form of how the different algorithms update their Dirichlet concentration parameter counts when attempting to learn the model of environment dynamics. We now go through, in detail, each algorithm to describe their key elements and comparative differences.

3.2.1 Vanilla Active Inference (VAI)

As referenced earlier, **VAI** simulations require one to define sets of priors over actions which are essentially used as hard-coded policies which the agent scores using the metrics of both Free Energy and Expected Free Energy. The algorithm is as follows, with the blue rectangle demarcating the elements of the algorithm unique to **VAI**, particularly when compared to the other Active Inference algorithms:

Algorithm 3 Vanilla Active Inference

Input: Likelihood function A – True transition function B – Agent transition model b – Initial-state distributions D – Agent initial-state model d resource limits l – Resource locations $S_{resources}$ – Number of time-steps T – Number of policies p – Number of Marginal Message passing iterations N

```
t_resources = [0, 0, 0]
resources = [food, water, shelter]
π ← InitialiseRandomPolicies(p, T)
posteriors ← InitialiseStatePosteriors(T,π)
while t < T & T_resources < l do
    if t ≥ 1 then
        true_states{factor} ← B(true_states(t-1)), action)
        UpdateConcentrationParameters(posteriors, b)
    else
        true_states(t) ← Sample(D)
    end if
    if true_states(t) is in S_resources then
        ResetResourceTime(true_states(t), S_resources)
    end if
    h ← CalculateHorizon(t_resources)
    observation ← SampleObservation(true_states)
    for each policy π do
        for tau = 1 to T do
            [F, posteriors] ← MarginalMessagePassing(posteriors, observation, T_resources, b, A)
        end for
    end for
    G ← ExpectedFreeEnergy(posteriors, π)
    posteriors ← PolicyPosterior(G, F)
    action ← max(policy_posteriors(t))
end while
```

The *ExpectedFreeEnergy* function seen here is simply the calculation of the terms of the **EFE** equation. A notable difference in this algorithm, when compared to the recursive tree-search algorithm previously described, is the mechanism by which state inference is performed. Here, marginal message passing is used, which, as described in section 2.9, uses a combination of forward and backward transition probabilities, as well as the likelihood model, to iteratively update the posterior over states for each policy. While marginal message passing is not used in the other algorithms, it was incorporated here to keep it as close as possible to the original algorithms used in earlier Active Inference literature. The *InitialiseRandomPolicies* function creates a list of random actions, the length of this list being equal to the number of time-steps in the trial, T. Another difference to this algorithm is the structure of the posterior over the state-space. The algorithm stores, in memory, the posterior for each policy at each time-step. Due to the nature of the posterior updates over the course of the simulation, these posteriors, on average, converge - driven by the data (observations) received at each time-step of the trial (Smith et al., 2021)

When compared to what is normally found in the literature for **VAI**, this implementation differs in how it updates model parameters - updating them after every time step, rather than at the end of an episode. The attempt in this practical work has been to have all the algorithms operate in a dynamic, 'online' manner. In this context, this means applying learning after every step so that the agent updates its model over the course of an episode rather than only between episodes. As it is presented

here, the updating of Dirichlet concentration parameters (essentially a model over the various possible parameter models) is only applied to the transition model. Learning of the likelihood model was not attempted in VAI, and was reserved as a final set of tests between Sophisticated Inference and PPBS. Sophisticated Inference and VAI use the exact same mechanism and algorithmic structure to update these concentration parameters, therefore a comparison between VAI and PPBS in this regard (a comparison which specifically focuses on how and when these updates happen) would be redundant.

3.2.2 Sophisticated Inference (SI)

Much of the Sophisticated Inference algorithm has already been discussed, and its general algorithmic structure is covered by **Algorithm 2**. Like the **VAI** algorithm, this implementation of SI updates concentration parameter counts after every time-step.
Included in the *ForwardTreeSearch* function that this algorithm uses are two pruning techniques which, along with memoization, reduce the number of sub-trees the algorithm has to evaluate. The first of these is discarding actions which, at the current time-step of the look-ahead, are under a relative value threshold, when compared to the other actions. In all simulations, this is set to 1/16, following the value proposed in the original literature on sophisticated inference. In addition to this, there is also a form of 'state pruning' whereby, after an action has been selected in the current state, potential (belief) states in the next time-step, which are relatively unlikely (also a threshold of 1/16), are not explored, again reducing the number of sub-trees which are iterated over in the planning horizon.

3.2.3 Bayesian Reinforcement Learning Methods

Both the standard Bayesian Reinforcement Learning and Bayes-adaptive methods are similar to each other, differing only in that the Bayes-adaptive method is used when elements of the model are unknown. Importantly, these are both online algorithms, based on that used by (Paquet et al., 2005). As mentioned, the planning structure is identical to that used in the Sophisticated Inference algorithm, with the differences appearing only in the way the reward function is constructed. In general for these recursive search algorithms, it is important to note that this kind of search exactly equates to a directed value iteration approach, over a subset of reachable states from the initial belief state.
While the preferences in Sophisticated Inference are treated as a distribution, the preferences in the Bayesian Reinforcement Learning methods are treated as scalar quantities, as is the case in most Reinforcement Learning approaches (Sutton and Barto, 2018). The difference here is that Reinforcement Learning views reward as an explicit function that must be optimised, whereas Active Inference methods view reward as the similarity between expected observational preferences and actual received observations. Although this difference might seem inconsequential, it can sometimes be important, as the nature of the Sophisticated Inference preference function being a *distribution* shapes the resulting scalar values differently (always logarithmically due to the terms of the Expected Free Energy equation), and allows them to proportionally combine with the values resulting from the evaluation of the

epistemic and novelty terms, which are also distribution-based. The Bayesian methods, on the other hand, using only scalar numerical values for the reward function, potentially allow for more freedom in how the output of the reward function is shaped, as this shape is not confined to be logarithmic. An example of where this could be particularly useful, over a preference *distribution*, is in the case where the reward function must be learned - though no experimentation or formal analysis of this is conducted in this dissertation.

Bayes-adaptive methods for POMDPs (BAPOMDPs), as described in Section 2.6.2, are used for model learning, and incorporate the Dirichlet parameter counts into the belief state (or hyper-state as it is sometimes called). In this sense, its belief search is over possible parameter models, as well as possible states and observations. Algorithmically, this is implemented by simulating searches over these hyper-states which implicitly contain the agent's uncertainty over model parameters. Within the context of the simulations for this work, this essentially means that the *UpdateConcentrationParameters* is then performed at every recursive step of the *ForwardTreeSearch* function, rather than only after every real time-step. Importantly, these updates to the concentration parameters which happen during the forward search are not carried over to the next real time-step, but only exist within the context of the recursive planning. Like Sophisticated Inference, both the Bayesian Reinforcement Learning and Bayes-adaptive method implement action and state pruning

3.2.4 Propagated Parameter Belief Search (PPBS)

PPBS is a novel algorithm we implemented to use in the testing iterations where model learning was required. This algorithm essentially combines the Sophisticated Inference and Bayes-adaptive methods in a way which harnesses the strengths of both. As will be discussed in later sections, during the course of testing the algorithms in different environments, it quickly became clear that Sophisticated Inference and Bayes-adaptive methods showed poor convergence properties when complex learning was required in dynamic environments. Indeed, while little previous investigation has been done to measure Sophisticated Inference's capability in complex environments (Friston et al., 2020), it is well established that Bayes-adaptive methods, specifically the **BAPOMDP** algorithm, are heavily reliant on good initial prior beliefs to implement effective learning (Ross et al., 2007; Katt et al., 2018). The PPBS algorithm is an attempt to improve the ability for an agent to perform a more sophisticated counterfactual reasoning about how its beliefs might evolve, and, in doing so, make decisions which improve its capacity to learn model parameters.

To create the PPBS algorithm, the Sophisticated Inference scheme is modified to include an update to concentration parameter counts after every time-step, as is the case with the Bayes-adaptive method. As the name suggests, this allows for Sophisticated Inference to propagate beliefs about how parameters would change along its forward tree search. This is important, as it more adequately represents a 'simulation' of how an actual real-time trajectory would be if the agent were to really take a particular set of actions and, in doing so, update its parameter model after every step. This simulation done by the agent is, of course, based on the agent's prior belief over states and parameter models. However such a technique has shown good convergence properties (Ross et al., 2007).

In addition to this method of parameter belief propagation, PPBS additionally implements a 'backwards-smoothing' function, a feature suggested, in a more limited scope, in the original text on Sophisticated Inference (Friston et al., 2021). This backwards-smoothing function backtracks from the current time-step to adjust its posterior beliefs over states at previous time-steps. This is particularly useful in the case of learning, as it allows for 'matching' of observation and state pairs so as to update the Dirichlet concentration parameter counts. In the spirit of parameter belief propagation, the difference in PPBS, when compared to normal Sophisticated schemes, is the addition of simulated backward smoothing at every step in the forward search. One can view this as the agent pondering:

If I were to take action x, receive observation y and transition to belief state z, how would I then update my posterior over states for previous time-steps, and based on these posterior updates, change my model?

As we will see, this method of multi-leveled counterfactual thinking proves very useful in the case where the likelihood model (in this particular experiment the positions of the resources conditioned on the context) is unknown.

This backward-smoothing function is given as:

---

function BACKWARDSSMOOTHING(*posterior*, a, b, *observation*, action_history, t, τ)
    L ← posterior
    p ← 1
    for timestep = t + 1 to τ **do**
        p ← b(action_history(timestep-1))×p
        for state in L **do**
            L(state) ← L(state)×(observation(timestep))×a×p(state)
        end for
    end for
return normalise(L)
end function

---

and its integration into the Forward search function:

Algorithm 3 Propagated Parameter Belief Search

```
function ForwardTreeSearch(posterior, a, b, observation, t_resources, t, true_t, T, W, h)
    if t > true_t then
        if posterior(t) is in resource_locations then
            t_resources ← ResetResourceTime(posterior(t), resource_locations)
        end if
        start_time ← t - backwards_horizon
        if start_time < 1 then
            start_time ← 1
        end if
        novelty ← 0
        a_prior, b_prior ← a,b
        for τ = start_time to t do
            L ← BackwardsSmoothing(posterior, a, obvservations, action_history, t, τ)
            a, b ← UpdateConcentrationParameters(L, b, observations, a)
        end for
        W ← CalculateNovelty(a_prior, b_prior, a, b)
    end if
    for each action do
        Q(action) = b(posterior(t),action)
        predicted_observations ← a(Q(action))
        G(action) ← ExpectedFreeEnergy(predicted_observations, T_resources, W)
    end for
    if t < h then
        actions ← ViableActions(G)
        for action in actions do
            posterior(t + 1) ← Q(action)
            states ← Q(action)
            states ← LikelyStates(states)
            for state in states do
                observation ← SampleObservation(state, A)
                [G, posterior] ← ForwardTreeSearch(posterior, a b, observation, t_resources, t + 1,
true_t, T, W, h)
                S ← softmax(G) × G
                state_efe(state) ← S
            end for
            G(action) ← state_efe(states)*states
        end for
    end if
return [G, posterior]
end function
```

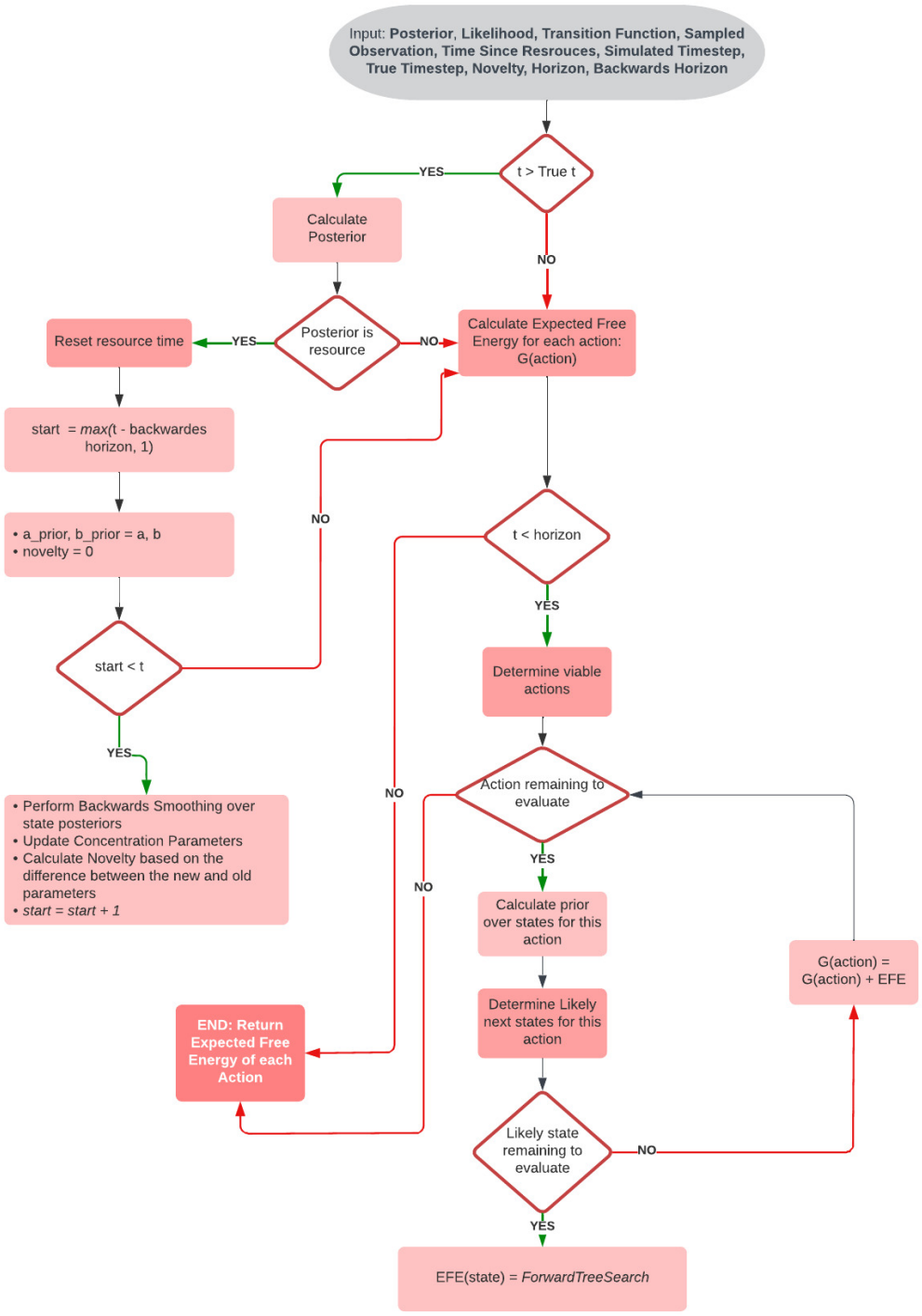

Figure 13: (**Flow Diagram depicting the Forward Tree Search Function with parameter updating included in the forward search)**

Where the cyan rectangle displays the elements of the algorithm that are unique to PPBS. It is important to note here that the updated concentration parameters are not only passed on to the next recursive call, but are also used to construct (via normalisation) the transition/likelihood functions

that are used in subsequent function calls of the recursive search.

3.3 Testing and Comparisons

The first iteration of testing focused on attempting to analyse the effect that the *epistemic* term had on agent behavior and survival. Survival here is measured by the number of time-steps the agent stays alive. The effect of including or excluding the *epistemic* term was a focus due to it being one of the defining features of Active Inference, and an element which results in directed exploration (Friston et al., 2015). In other machine learning techniques, such a feature is usually included as an extra heuristic, such as an $\epsilon - greedy$ search policy or a Boltzmann temperature parameter (Sutton and Barto, 2018). Of particular behavioral focus was the number of time-steps the agent spends on the hill state. This is the defining feature of all the environments and acts as the main device to show off the agent's decisions between exploration and exploitation.

These initial tests were conducted in **Environment 1**, as only small-scale behavioral comparisons were of interest. Additionally the first set of testing equipped the agent with a fully known model (i.e accurate model of the transition and likelihood functions). The combinations of algorithms tested with varying parameters are as follows.

Algorithm	Epistemic Term	Model
Active Inference	Yes	Known
Active Inference	No	Known
Sophisticated Inference	Yes	Known
Sophisticated Inference	No	Known
Bayesian RL	N/A	Known

Table 3: (**Transition function known, Likelihood known**). A display of different algorithms compared. Each trial was a maximum of 50 time-steps long. The resource limits for this testing iteration were 5, 7 and 8 for food, water and shelter respectively.

Additionally, the resource limits for this round of testing were:

Time since food limit	Time since water limit	Time since shelter limit
5	7	8

Table 4: The time-limits for each of the respective resources. After some heuristic testing in the smaller environment, these numbers were chosen based on an optimal range which drove the agent to quickly find resources, while also allowing for some margin of error. For example with these specific values, the agent could initially guess the position of the resources incorrectly at the start of the trial, and still have a chance to survive even after moving to this incorrect location.

As shown here, each resource was given different time-limits in order to highlight the categorical separation between them. In doing so, the conceptual view that the agent views each resource as a source of a specific *drive*, each with qualitative differences was accentuated. More practically, the

difference in time-limits sets the stage for the emergence of more interesting behavior, as the agent will often forgo resources with the more lenient time-limits in search of those with stricter limits.

The two metrics used for *performance* comparison are **Time-steps on hill** and **Time-steps alive**. In such experiments it is not always clear how one should measure the 'success' of an agent. We chose the number of time-steps survived for such a metric, as this correlated highly with the agent's ability to determine optimal planning trajectories, as well as its ability to learn a model of the environment dynamics. **Time-steps on hill**, as will be later discussed, was chosen as another metric due to its importance in defining key aspects of agent behavior, as well as correlating with the number of time-steps survived by the agent.

Each episode of testing consisted of 100 trials, with each trial lasting a maximum of 50 time-steps. For all testing iterations, there was high variance in *time-steps alive* between separate trials, and after some experimentation, 100 trials was found to be an adequate number to ensure a relatively consistent average. In addition to this, 50 time-steps was used as the maximum trial length for all small environments. This specific number was chosen as it appeared to represent a realistic upper-bound for the number of time-steps survived by the agent in the smaller environment (the agent would, on average, rarely remain alive for 50 time-steps). In all testing iterations, a trial ends when either the agent dies, or the time-step reaches the maximum time-step limit (in this case 50). Results were taken as an average over these 100 trials. Note that in this first iteration of testing, the Bayes-adaptive and PPBS algorithm were not used, as it did not include scenarios where parts of the model are unknown.

The second modality of testing introduced scenarios where the transition function was initially unknown. Unknown here essentially means that the agent starts with a uniform prior over transition probabilities between contexts, and so must attempt to learn the true transition probabilities. In addition to the inclusion/exclusion of the *epistemic* term, another parameter tested here was whether the precision of the preference distribution was high or low. Precision, in this context, is raised or lowered by simply multiplying the preference distribution by some constant $0 < c \leq 1$. This effectively flattens the distribution, and so reduces the difference in preference which the agent has for the various observations. Additionally, this implicitly increases the relative weighting of the epistemic and novelty terms. The testing of this element was included in order to show how risk-seeking behavior in the agent can easily be reduced, and, in turn, to see how this reduction affected the time spent on the hill, and the time survived. In all trials the constant was $c = 0.1$ and $c = 1$ for low and high precision respectively. The choice of these numbers was a result of heuristic experimentation with different values. The value of $c = 1$ simply represents no precision adjustment of the preference distribution, thus the distribution is maximally precise with respect to the values returned by the reward function shown in **Algorithm 1**. The value of $c = 0.1$ was chosen due to it acting, for these environments at least, as an upper bound at which the agent would transition to displaying noticeable epistemic behavior (due to the lowering of the preference distribution precision increasing the relative weighting of the epistemic term).

Algorithm	Epistemic Term	Preference Precision	Transition Model	Likelihood Model
Active Inference	Yes	High	Unknown	Known
Active Inference	Yes	Low	Unknown	Known
Active Inference	No	High	Unknown	Known
Active Inference	No	Low	Unknown	Known
Sophisticated Inference	Yes	High	Unknown	Known
Sophisticated Inference	Yes	Low	Unknown	Known
Sophisticated Inference	No	High	Unknown	Known
Sophisticated Inference	No	Low	Unknown	Known
Bayes-adaptive RL	N/A	N/A	Unknown	Known
PPBS	Yes	High	Unknown	Known
PPBS	Yes	Low	Unknown	Known
PPBS	No	High	Unknown	Known
PPBS	No	Low	Unknown	Known

Table 5: (**Transition function unknown, Likelihood known**). A list of different algorithm and experimental parameter combinations for trials in the 5x5 environment. Each trial was a maximum of 50 time-steps long, with 100 trials run to ensure an accurate average.

Time since food limit	Time since water limit	Time since shelter limit
8	9	10

Table 6: The time-limits for each of the respective resources for the second test iteration. These were slightly increased, in comparison to those used for the first iteration (where the model was known), to compensate for the agent's lack of model knowledge, and to allow for more time spent learning.

As shown, this set of tests includes the PPBS and Bayes-adaptive algorithms, as the transition model is unknown. As with the first iteration of testing, this was conducted in **Environment 1**. Additionally, the time limits the agent could go without resources were increased for this set of tests. Due to the agent not knowing the transition function, surviving in such an environment was much harder than in the first set of tests, and so the resource time limits were adjusted accordingly to give the agent more leniency with respect to resource time-limits, and so allow it more time to effectively learn without quickly dying.

The third round of testing used **Environment 2**. This set of tests focused less on smaller-scale behavioral differences, based on the inclusion/exclusion of the *epistemic* term and high or low preference precision, and more on the broader metric of survivability under known and unknown transition models. For this reason the epistemic term was included for all Active Inference algorithms, and the preference precision constant was standardised at $c = 0.1$ to relatively encourage exploratory behavior. This round of testing did not include VAI, as for an environment of this size, constructing hard-coded policies, implementing marginal message passing and holding posterior beliefs over each state for each policy, was computationally infeasible. The time-step limit of each trial was increased to $T = 100$, with the results of an episode taken as an average of 100 trials. The increase to the trial length for this testing iteration was done to proportionally match the increase in resource time-limits.

Additionally, in the test cases where the transition model was unknown, 100 iterative episodes (each

consisting of an average of 100 trials) were conducted to capture how the success metric of the agent changed as the learning of one episode carried over to the next. Due to the environment being larger and the resources more spread out, the resource time-limits were proportionally increased.

Algorithm	Transition Model	Likelihood Model
Sophisticated Inference	Known	Known
Sophisticated Inference	Unknown	Known
Bayesian RL	Known	Known
Bayes-adaptive RL	Unknown	Known
PPBS	Unknown	Known

Table 7: (**Transition function known/unknown, Likelihood known**). A display of the different algorithms in the cases of both known and unknown transition models. Each trial was a maximum of 100 time-steps long. The resource limits for this test iteration in the case of a known transition model were 15, 17 and 20 for food, water and shelter respectively, and 20, 22 and 25 for the case of an unknown transition model.

For the trials where the transition model was known, the resource limits were:

Time since food limit	Time since water limit	Time since shelter limit
15	17	20

Table 8: Resource time-limits used for testing iteration 3, in the larger 10x10 environment in trials where the model was known

These values were increased, in proportion to the values used for testing iterations 1 and 2, to match the increase in environment size, and the increase in resource spread that accompanied this.
For the trials where the transition model was unknown, the time-limits similarly increased slightly, so as to compensate for model uncertainty and allow for time spent learning, as described previously.

Time since food limit	Time since water limit	Time since shelter limit
20	22	25

Table 9: Resource time-limits used for testing iteration 3, in the larger 10x10 environment in trials where the transition function was initially unknown.

As was the case in the smaller environment, the relative increase in resource time-limits, in the tests where the transition model was unknown, was implemented to allow the agents to spend more time learning the model.

Finally, the fourth testing modality, using **Environment 2**, conducted trials only comparing Sophisticated Inference and **PPBS**. **PPBS** is an extension to the Sophisticated Inference algorithm, and so

the aim here was to focus entirely on the difference between these two algorithms in a scenario where propagated parameter beliefs were hypothesised to noticeably affect the agent's behavior. This specific scenario is a setup where, although the context transition function is known, the resource likelihood function (the position of the resources given the context) is unknown. This is a particularly difficult environment in which to survive and learn, as context observations given by the hill initially mean nothing with respect to knowing where resources are. Additionally, the agent must attempt to connect resources it discovers with the context the environment is in, in order to learn the connection between context and resource positions. However, when the context is stochastically shifting, learning this way is notoriously difficult and requires behavior that most algorithms struggle to achieve (Costa et al., 2017) There are then only two variants of tests done for this testing modality:

Algorithm	Transition Model	Likelihood Model
Sophisticated Inference	Known	Unknown
PPBS	Known	Unknown

Table 10: The model configuration used for the Sophisticated Inference and PPBS algorithms in the fourth testing iteration. In all trials the transition model was known, but the likelihood was initially unknown (maximally uniform belief over all likelihood probabilities.

With the resource limits given as:

Time since food limit	Time since water limit	Time since shelter limit
20	22	25

Table 11: The resource time-limits used for the fourth testing iteration, comparing PPBS and Sophisticated Inference in trials where the likelihood was initially unknown.

Due to this testing iteration also presenting a case where an element of the model is unknown (likelihood), the values for these trials were chosen to be the same as those used for the third testing iteration (in the trials where the transition model was unknown). The reasoning behind this choice was the same as above: Due to the agent having inaccurate model knowledge as to how it can visit resource state, more leniency was given to the resource time limits to compensate for this inaccuracy and to allow for more time for the agent to learn the model.

4 Results

We now present the results of the four testing iterations described above.

4.1 First Iteration

The first iteration tested different algorithms in the smaller 5x5 environment, with the transition and likelihood being known for each algorithm. The results for the first iteration are displayed below:

Algorithm	Epistemic Term	Time-steps on Hill	Time-steps Alive
Active Inference	Yes	2.3/50	36.2/50 ($\sigma = 17$)
Active Inference	No	1.4/50	22.2/50 ($\sigma = 13.3$)
Sophisticated Inference	Yes	0.9/50	40.8/50 ($\sigma = 18.1$)
Sophisticated Inference	No	0.8/50	39.4/50 ($\sigma = 17.6$)
Bayesian RL	No	0.9/50	42.2/50 ($\sigma = 16.9$)

Table 12: (**Transition function known, Likelihood known**). Results, averaged over 100 trials in the 5x5 environment, of different algorithms. Each trial was a maximum of 50 time-steps long, with the last column indicating how long the agent survived within those 50 time-steps. The resource limits for this iteration were 5, 7 and 8 for food, water and shelter respectively. Results are rounded to 1 decimal place.

An important detail here is that Bayesian Reinforcement Learning methods naturally do not include an epistemic term in any of their formulations as they are not Active Inference algorithms. Nonetheless, Bayesian RL was compared here due to its similarity with Sophisticated Inference (Sajid et al., 2021). The numbers in the third column indicate the average number of time-steps in a trial which the agent visited the hill state, while the fourth column shows the average number of time-steps the agent survived, with a trial automatically ending after 50 time-steps.

4.2 Second Iteration

The second iteration of tests tested the different algorithms in the smaller 5x5 environment. In these tests the likelihood model was known, but the transition model was not. Incorporated in this iteration is the variable of preference precision, due to it largely affecting agent behavior in the presence of model uncertainty. The results of the second iteration of tests follow a similar format to the first.

Algorithm	Epistemic Term	Preference Precision	Time-steps on Hill	Time-steps Alive
Active Inference	Yes	High	2.3/50	13/50 ($\sigma = 7.3$)
Active Inference	Yes	Low	7.8/50	15/50 ($\sigma = 7.9$)
Active Inference	No	High	1.6/50	12/50 ($\sigma = 7.6$)
Active Inference	No	Low	1.9/50	11/50 ($\sigma = 8.1$)
Sophisticated Inference	Yes	High	0.7/50	17/50 ($\sigma = 8.6$)
Sophisticated Inference	Yes	Low	11.7/50	23/50 ($\sigma = 14.3$)
Sophisticated Inference	No	High	0.1/50	15/50 ($\sigma = 8.47$)
Sophisticated Inference	No	Low	0.8/50	16/50 ($\sigma = 9.2$)
Bayes-adaptive RL	No	N/A	0.1/50	17/50 ($\sigma = 7.6$)
PPBS	Yes	High	1.7/50	17/50 ($\sigma = 8.3$)
PPBS	Yes	Low	14.1/50	22/50 ($\sigma = 13.7$)
PPBS	No	High	0.1/50	14/50 ($\sigma = 7.9$)
PPBS	No	Low	2.9/50	19/50 ($\sigma = 6.1$)

Table 13: (**Transition function unknown, Likelihood known**). Results, averaged over 100 trials in the 5x5 environment, of different algorithm and experimental parameter combinations. Each trial was a maximum of 50 time-steps long, with the last column indicating how long the agent survived within those 50 time-steps. The resource limits for this iteration were 8, 9 and 10 for food, water and shelter respectively. Results are rounded to 1 decimal place

In this iteration the distinguishing factor is that the transition function of the context is now unknown. Therefore, we include the PPBS algorithm (as this was specifically designed for cases where the model needs to be learned) and switch the Bayesian Reinforcement Learning algorithm with a Bayes-adaptive method, which is a variant of Bayesian Reinforcement learning used when model parameters are unknown and learning is required. As is the case with Bayesian RL, the Bayes-adaptive method does not include an epistemic term in any of its variations. Additionally, because the reward function is not represented as a distribution, the variable of preference precision is not applicable.

4.3 Third Iteration

The third iteration tested the different algorithms in the larger 10x10 environment. In these trials the epistemic term was always included for the Active Inference agents, and the preference precision was standardised at ($c = 0.1$)

Algorithm	Transition Model	Likelihood Model	Time-steps on Hill	Time-steps Alive
Sophisticated Inference	Known	Known	3.3/100	79.4/100 ($\sigma = 23.1$)
Sophisticated Inference	Unknown	Known	9.6/100	24.5/100 ($\sigma = 6.3$)
Bayesian RL	Known	Known	5.1/100	82.9/100 ($\sigma = 25.7$)
Bayes-adaptive RL	Unknown	Known	6.7/100	23.5/100 ($\sigma = 9.9$)
PPBS	Unknown	Known	14.3/100	24.1/100 ($\sigma = 4.9$)

Table 14: (**Transition function unknown, Likelihood known**). Results, averaged over 100 trials in the 10x10 environment, of different algorithm and known-model combinations in. Each trial was a maximum of 100 time-steps long, with the last column indicating how long the agent survived within those 100 time-steps. The resource limits for this iteration in the case of a known transition function were 15, 17 and 20 for food, water and shelter respectively, and 20, 22 and 25 for the case of an unknown transition function. Results are rounded to 1 decimal place

The key factors to note here are the use of the larger **Environment 2**, the standardisation of the preference precision at a low precision ($c = 0.1$), and the inclusion of the epistemic term for both Active Inference algorithms (Sophisticated Inference and PPBS). Naturally, the PPBS and Bayes-adaptive algorithms are only used for the cases where the transition function is unknown and must be learned, as is the case in the previous testing iteration.

Figure 14 displays the **multi-episode** performance of the three different algorithms in the case of an unknown transition function. Importantly, the learning of model parameters (parameters of the transition function in this case) undertaken in a given episode is passed on to the next episode (and so on). Thus the progression of episodes is iterative, with the resulting learned model parameters of a given episode used as the starting model parameters of the next episode. This is in contrast to the tables presented above, which show the results for a single (initial) episode, and thus involve no iterative transferred learning. The smaller figures, below the main comparison plot, show the individual plots of the algorithms, additionally displaying the standard deviation for each episode which was the result of an average of 100 trials.

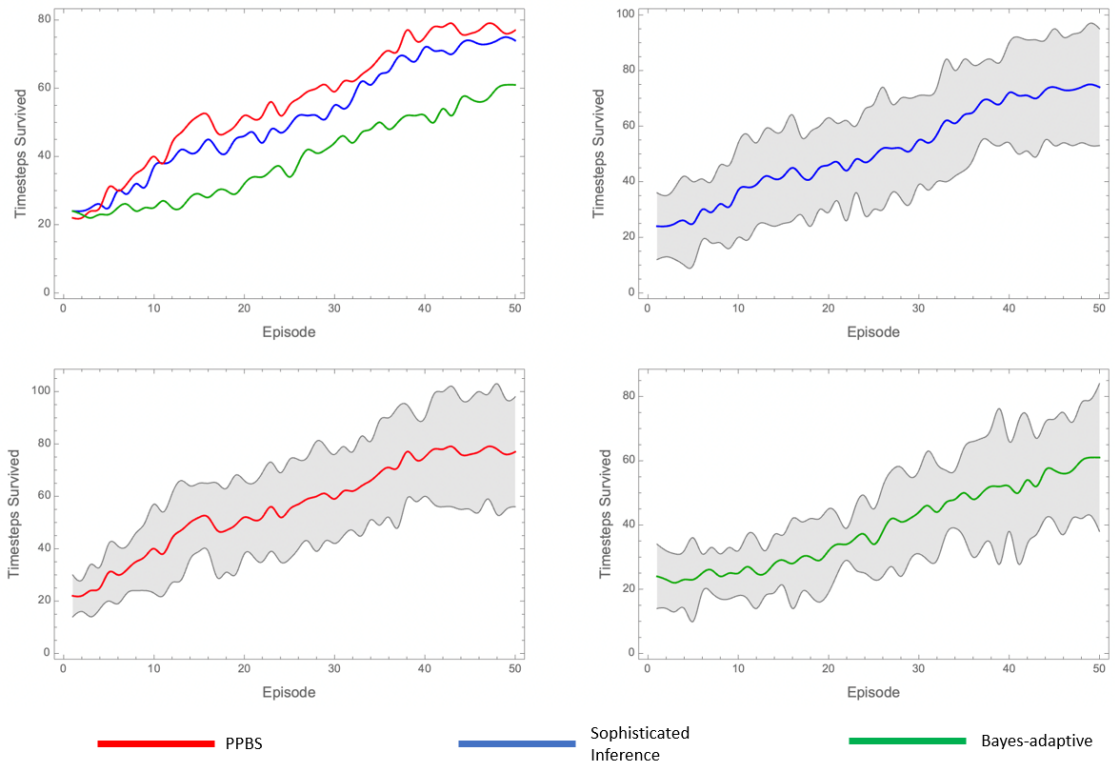

Figure 14: (**Transition function unknown, Likelihood known**). A performance comparison between PPBS, Sophisticated Inference and the Bayes-adaptive method over 100 iterative episodes, with the results of each episode being the average over 50 trials. Learning is transferred between each episode, resulting in an accumulation of learning across episodes. The individual plots are presented to show the standard deviation across the trials for each episode..

4.4 Fourth Iteration

The fourth iteration, also in the 10x10 environment, compares Sophisticated Inference and PPBS in trials where the transition model is known, but the likelihood is unknown.

In similar style to the third test iteration, **Figure 15** shows the multi-episodic comparison between PPBS and Sophisticated Inference in the case where the transition function is known, but the likelihood is unknown. In this iteration, because we were only interested in the multi-episodic learning, we forgo displaying the results for just a single initial episode. Like the previous iteration, the epistemic term is included for both algorithms and the preference precision is set to low for all episodes.

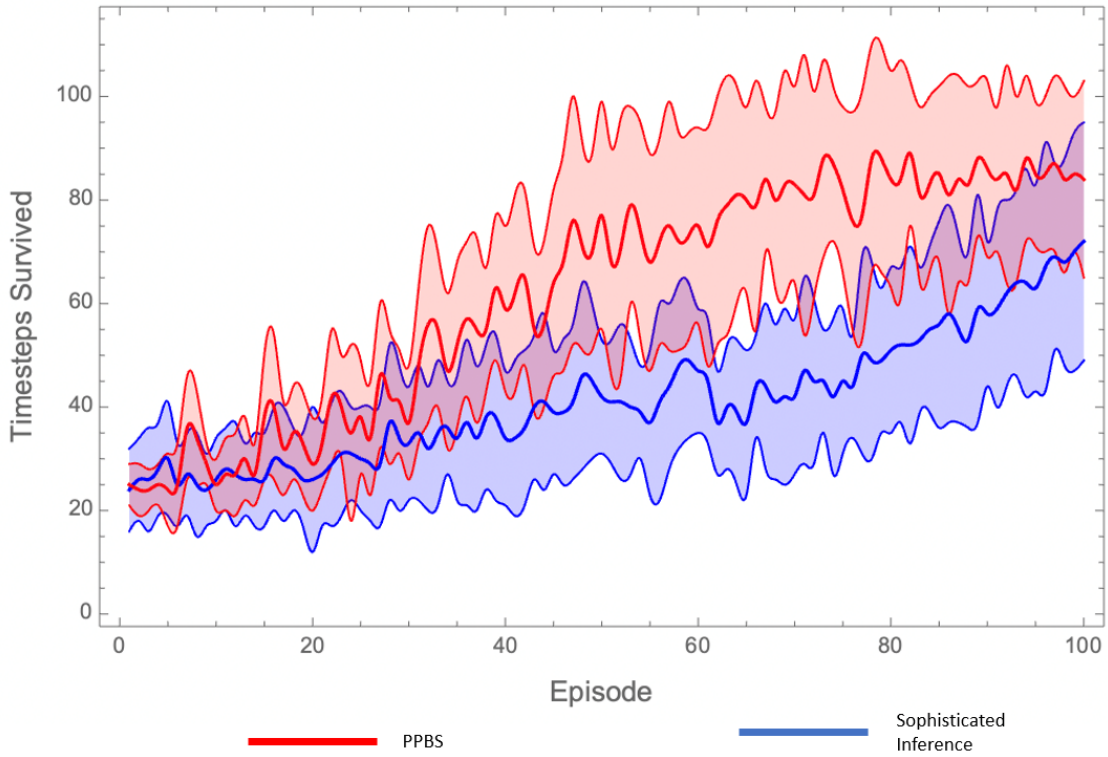

Figure 15: (**Transition function known, Likelihood unknown**): A performance comparison between PPBS and Sophisticated Inference over 100 iterative episodes, with the results of each episode being the average over 50 trials. Learning is transferred between each episode resulting in an accumulation of learning across episodes.

5 Discussion

The overarching aim of this dissertation has been to investigate the relative capacity of different formulations of Active Inference algorithms to navigate a multi-objective dynamic environment - in cases where elements of the model are either known or unknown. Additionally, we presented a novel extension to the Sophisticated Inference algorithm - (PPBS) - which aimed to improve the agent's ability to learn in the presence of model uncertainty. The setting for such investigation was a multi-objective *context-driven* environment which tested the agent's ability to optimise both exploration and exploitation, as well as effectively learn model parameters in the presence of state-uncertainty. We begin this section by discussing some interesting aspects of general agent behavior that were noted over the course of the testing iterations. This behavior is useful in conceptually demarcating the effects of the different testing variables, particularly the inclusion/exclusion of the epistemic term, the preference precision and use of propagated parameter beliefs. Subsequently, we will analyse the results presented in the previous section, and attempt to provide conceptual and mathematical explanations as to their form. Finally, we focus on the PPBS algorithm and discuss implications it has for the Active Inference framework.

As an aside, we here note that due to much of the methodology for this dissertation representing relatively novel approaches, the comparison of results, hereby attained, with results and conclusions presented in previous Active Inference literature works was difficult. We thus mainly focus on drawing comparisons and conceptual conclusions based on the differences between the different results presented *within* this dissertation. The main and central point of comparison which is included between the work presented in this dissertation and other literature, comes in the form of comparing the new Algorithm, PPBS, with the other most closely-related established approach, Sophisticated Inference.

5.1 Agent Behavioral Patterns

The agent's general forms of behavior were, on the whole, very sensitive to the various different hyper-parameters used for the trials. Specifically, these were: The inclusion/exclusion of the epistemic term, the preference distribution precision, the transition function precision and the resource time limits. The main consequence of altering these was the varying of the number of time-steps the agent spent at the hill-state.

Figure 13 shows a case of an example behavior when the epistemic term is included, but the preference precision is high ($c = 1$). In this case, both Sophisticated Inference and PPBS agents, despite having a maximally imprecise (unknown) transition function and not knowing the initial context, initially ignore the hill, and attempt to guess at the current context by visiting resource positions they know (due to having a precise likelihood model) are associated with a particular context. This is due to the epistemic term having proportionally low impact when compared to the agent's preferences, and so the agent's behavior is driven by its imperative to maximally meet its multi-objective preferences, rather than to seek information in the form of large posterior updates to its beliefs about hidden states. This is inline with the classic *risk-seeking* behavior previously described in Active Inference literature (Smith et.al

2021). For the Sophisticated Inference algorithm, similar behavior is seen when the epistemic term is omitted, regardless of the preference precision. However, a more interesting case emerges in the scenario where the PPBS algorithm has no epistemic term and a low preference precision. In this case, the agent does, initially, visit the hill, and proceeds to stay there for a several time-steps depending on the resource limits. This is due to the PPBS algorithm including the concept of how the agent's beliefs about its model might evolve when it implements planning in the form of its belief tree-search look-ahead. In this way, the agent is able to 'simulate' how the hill state might provide it with large *novelty* updates, due to the hill state indirectly providing more precise updates to model parameters when compared to another state. Specifically, due to the backward-smoothing approach incorporated into the PPBS algorithm, the agent understands that if the hill provides it with consecutive context transitions, it will be able to look back and at previous time-steps, have a more precise belief about what context it was in at those time-steps, and so more precisely update its beliefs about the context transition function. For this reason, even without the epistemic term included, the PPBS agent will go to the hill initially, and stay there for a certain number of consecutive time-steps. Throughout the rest of this section, we will expand upon this description and analysis of PPBS behavior.

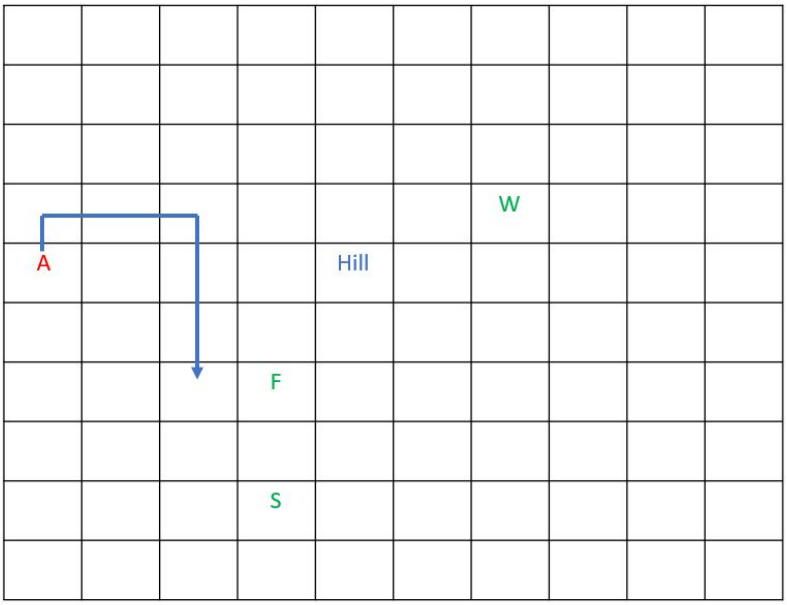

Figure 16: (**Unknown transition model, known Likelihood, high preference precision**): If the precision of the preference distribution is high, both Sophisticated Inference and PPBS agents often ignore the hill and immediately attempt to guess where resources might be.

Figure 14 shows the simple example case where the agent proceeds straight to the hill at the beginning of the trial. As mentioned above, this is the behavior of the PPBS agent when the preference precision is low, even without the epistemic term. When the epistemic term is included and the preference precision is low, Sophisticated inference also shows such initial behavior. Additionally, in the case where the transition function is precisely known by the Sophisticated Inference algorithm, it will also often exhibit this behavior, whether the epistemic term is included or not. The reason for this is interesting, and at

the core of the similarity between Sophisticated Inference and Bayesian Reinforcement Learning, which like-wise will often initially visit the hill for one time-step. This behavior is due to both algorithms being Bellman-optimal with respect to their prior beliefs, meaning that, given an initial belief state and a mechanism to calculate the value of some subset of additional belief states (for example all belief states reachable from the initial belief state up to some horizon, as is the case in these implementations), it will optimally calculate the value of each of these belief states. Given a deterministic and greedy policy construction procedure, it will subsequently be able to pick an optimal policy which maximises expected value (reward/preference).

Importantly, the accuracy with which it calculates the value of these belief states is entirely predicated on the initial belief state. Thus, if the initial belief state is inaccurate, its calculation and evaluation of subsequent belief states will be inaccurate. Therefore, in the trials where the transition model is accurate (known) but the initial context unknown, the agent knows that transitions are relatively static (80% chance to remain in the same context and 20% chance to transition to one other context) and so often views visiting the hill as optimal, as it is the state which will most precisely update its belief about what context the environment is in. Due to the nature of the counterfactual trajectory planning the agent implements, it searches through all possible belief trajectories up to the planning horizon and thus calculates, ahead of time, the optimal set of subsequent actions for whatever observation the hill state gives it. Put more simply, the agent calculates that the hill will provide it with *some* precise context observation. For each of these observations the hill could potentially provide the agent, it calculates the optimal trajectory following on from that time-point. These belief trajectories have high precision compared to other belief trajectories which do not include the hill, thus the agent views them, in the expectation, as being valuable.

In contrast to this, if the agent does not know the transition function and starts with an initial belief that the transitions between contexts are entirely uniform, it does not view the hill as providing more precise belief trajectories, as whatever the hill shows it is meaningless with regards to providing any kind of precision for future time-steps, due to the maximal stochasticity of the context transitions.

It is important to note at this point that these types of behavioral patterns are also influenced by the resource time limits, and ultimately shaped by the reward function of the problem. For example, given a very lenient time-limit for each resource. All agents will initially ignore the hill and engage in behavior similar to that shown in Figure 13, where attempts at guessing the context are made. This is due to the agent not believing that it will incur the penalty of reaching a resource time-limit and so loses little by guessing at contexts, even if its guesses are wrong. In these scenarios of lenient resource time-limits, the agent will often initially move around guessing the context and only move to the hill if it believes that subsequent guessing would have a high chance of incurring death. Although we describe this conceptually here, mathematically, this is due to the agent precisely following the actions it believes will yield the largest return in the expectation, as is the case with all Bellman-optimal algorithms.

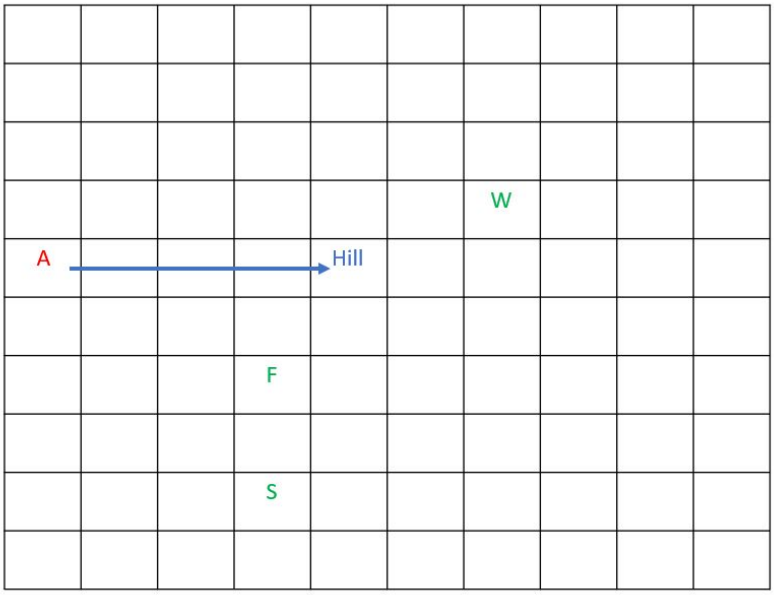

Figure 17: (**Unknown transition model, known Likelihood, low preference precision**): If the precision of the preference distribution is low, both Sophisticated Inference and PPBS agents tend to immediately go to the hill and stay there for as long as they can before their time since a resources near a limit

The behavioral patterns described here set the stage for an analysis of the results obtained over the four testing iterations, and act as a general conceptual explanatory backdrop to the nature of the results presented henceforth.

5.2 Results Analysis

The results of the first test iteration are in-line with the Bellman-optimality explanation for belief-based algorithms described in the previous section. When the transition model is relatively precise and known, The agent will value the hill state with or without the inclusion of the epistemic term. Thus, Bayesian RL, and Sophisticated Inference with/without the epistemic term both perform comparably and both, on average, visited the hill once initially in order to disambiguate context, after which the agent uses its precise knowledge of the transition function to predict the transition of contexts and so has little need for the hill. Although the vanilla Active Inference algorithm is not of focus in much of this discussion, it is interesting to note that in the 5x5 environment, with an included epistemic term, it performs similarly to the other algorithms, without being a Bellman-optimal method. This is likely due to the simplicity of the environment, and, as we will see, its performance degrades significantly when the transition function is unknown.

A repeating theme throughout all these results is that the agent often was simply unlucky with respect to how the contexts changed, despite acting optimally. For this reason, in many trials, despite the agent having an accurate model of the context transition function and acting in a probabilistically optimal way, it would die long before the maximum number of time-steps of the trial was reached.

An immediate fact which emerges when looking at the results of the second testing iteration, in which the transition function is unknown, is the correlation between the number of time-steps spent on the hill and the number of time-steps the agent survived. Due to the agent having an imprecise model of the transition function it is unable to simply predict how the contexts will transition as it could in the previous test iteration. Therefore, it seems critical for the agent to visit the hill in order to both improve its model of the transition function (as it can directly view how contexts transition when at the hill state and so accumulate Dirichlet concentration parameter counts for these transitions) and to disambiguate the current context when its posterior belief about such becomes imprecise.

The variable of preference distribution precision becomes particularly important in these tests. Due to the agents not viewing the hill as intrinsically valuable, as a consequence of their initial imprecise belief about context transitions (as discussed above), it becomes important to 'encourage' visits to the hill via the epistemic term. The term becomes proportionally more weighted as the preference distribution becomes less precise. Such experimentation with 'encouraging' the agent to act in favour of epistemic value is seen in many previous works on Active Inference agents (Sajid et al., 2021; Smith et al., 2020; Friston et al., 2021; Smith et al., 2021). Pairing this with the correlation between time-steps on the hill and time-steps survived, the benefit of the epistemic term becomes evident: In scenarios where the agent has an imprecise model, and so does not implement accurate Bellman-optimal searches, it is important to encourage exploration via some other device. The epistemic term, as part of the Expected Free Energy equation, is such a device, and, unlike other exploration heuristics often used in machine learning implementations, is naturally derived from the Free Energy formulation.

All four variations of the vanilla Active Inference algorithms do relatively poorly when the transition model is initially imprecise (uniform at the beginning of the trial). Although when the epistemic term is included and preference precision is low, the vanilla Active Inference agent spends a relatively high number of time-steps on the hill, on average, it struggled with alternating between 'exploring' (i.e visiting the hill) and 'exploiting' (attempting to move to resource positions). This is likely due to its inability to generate counterfactual beliefs about how its beliefs might change along action trajectories, as is done in Sophisticated Inference. The hill was only attractive to the agent due to its epistemic value, with the **VAI** agent having no concept of how its beliefs about context becoming more precise, upon visiting the hill, would subsequently aid it in future time-steps. Specifically, along its trajectories, the vanilla agent would not explicitly use the hill as a 'stepping-stone' device, from which it could devise counterfactual trajectories for whatever context observation it was shown. Rather the hill was only attractive to it when its beliefs about what context it was in became imprecise. Another important observation was the fact that the vanilla agent would only visit the hill upon reaching some actual imprecise belief about what context it was in. This is in contrast to the other Active Inference algorithms (Sophisticated Inference and PPBS) which were able to predict at what future time-points their context beliefs might become imprecise, and so factored visiting the hill into trajectories where this was the case. Certainly, there is some functional overlap between the **VAI** and Sophisticated agents, regardless of what the agents' 'reasons' are for visiting the hill - and the random nature of Vanilla Active Inference policy generation could also have played a role in its poorer performance, however it is plausible that the 'unsophisticated' belief scheme of the vanilla Active Inference agent

was inadequate for such a dynamic environment.

PPBS and Sophisticated Inference algorithms compare similarly across all variable combinations. Despite this, it is important to note that PPBS spends more time, on average, at the hill state. Of specific importance is the fact that the PPBS agent visits the hill even when the epistemic term is omitted. This is in contrast to Bayes-adaptive and Sophisticated Inference methods (with similar variables) which, on average, visit the hill 0 times, spending the entire trial attempting to 'exploit' possible resource locations. The reason for PPBS's different behavior, as somewhat discussed in the previous section, is due to the novelty term of the Expected Free Energy equation becoming relatively highly weighted for trajectories where the agent spends consecutive time-steps at the hill. Thus, when the preference precision is relatively low, these propagated novelty beliefs play a significant role in driving agent behavior. Although, in the case of an isolated individual episode, this does not significantly affect the agent's performance, as we will see, the difference this creates emerges when multiple consecutive episodes with transferred learning are considered.

Not unexpectedly, the Bayes-adaptive algorithm performs relatively poorly. Although this algorithm absorbs model parameters into its belief state-space, and so implicitly propagates parameter beliefs in a similar fashion to PPBS, these model parameter beliefs are not explicitly incorporated into the reward function. Therefore, the fact that the Bayes-adaptive agent simulates how its model parameters might change, like the PPBS agent, plays no explicit role in determining how it values state-action trajectories. Rather, in Bayes-adaptive methods, this mechanism is used to more accurately represent the possible states of both the environment, and the agent's model, and so provide the agent with a more accurate evaluation and counterfactual search over states. As previously mentioned, Bayes-adaptive methods are sensitive to the accuracy of prior beliefs. Thus, in the case where its prior belief over context is uniform, the Bayes-adaptive method struggles to generate accurate belief trajectories. This, in combination with no explicit incentive to visit the hill, such as epistemic or novelty gain, results in the agent's relatively poor performance.

Like the Bayes-adaptive algorithm, when the Sophisticated Inference agent is not equipped with the epistemic term, or has high preference precision, it completely ignores the hill on average, and thus survives for fewer time-steps than an agent which makes use of the hill.

It is important to reiterate, when viewing the results of the third testing iteration, that the epistemic term was included and the preference precision set to low ($c = 0.1$) for all Active Inference algorithms. Although the results for the cases where the transition model is known (accurate) are proportional to the results of the first testing iteration, the results of the cases where the transition model is unknown differ somewhat. The most obvious point which stands out is the fact that, for a single isolated episode, there no longer appears to be a correlation between the number of time-steps spent on the hill and the number of time-steps survived, with the Sophisticated Inference, Bayes-adaptive and PPBS algorithms all performing similarly on average. While the reason for this is not clear, it could be due to the feature of greatly increased resource time-limits for this set of trials (food = 20, water = 22, shelter = 25), allowing for context guessing to be an equally viable strategy. The amounts by which resource time-limits were changed for each environment was relatively arbitrary, and perhaps, in this case, served to create relatively greater resource time-limit leniency, despite the environment

being larger. Further investigation, with respect to resource time-limit settings, would be required to determine if this factor is indeed the cause of the results. A second observation, which might at first seem surprising, is the number of time-steps spent on the hill by the Bayes-apative agent. This is in stark contrast to the previous testing iteration in the smaller environment, where the agent almost always completely ignored the hill. This, however, is most likely due to the actual layout of the larger environment. As seen in Figure 8, showing the 10x10 environment, many resource positions are close to the hill, therefore, in this case it is likely that the visits to the hill by the Bayes-adaptive agent were simply transitional steps in order to travel to these proximal locations. This is in contrast to the smaller environment, where the general optimal behavior to visit possible resource locations was for the agent to circle the hill.

We next turn to analysing the multi-episodic comparison between the Bayes-adaptive method, Sophisticated Inference and PPBS, and it is with these results that more interesting factors emerge. Although, as discussed above, there appears to be very little difference in average performance between these three algorithms (in the scenario of an unknown transition model), this is for the case of a single, isolated episode, where no transfer of learning takes place between trials. In Figure 11, we see that although the algorithms perform similarly in early episodes, the Sophisticated Inference and PPBS algorithms start to rapidly out-perform the Bayes-adaptive algorithm as the episodes progress. The reason behind this is intuitive: The more time-steps the agent spends at the hill in earlier episodes, the more quickly it learns the transition function. In this way, despite the agent perhaps not living comparatively longer in earlier episodes, it 'invests' more into the success of later episodes, as in those later episodes, it will subsequently have a more accurate model of the transition function. It is here that a potential usefulness of the PPBS algorithm emerges. For reasons previously discussed, the PPBS algorithm encourages the agent to visit the hill to a greater degree than any other algorithm investigated in these tests. A natural result of this, then, is that the PPBS agent learns the transition function more quickly than both the Sophisticated Inference and Bayes-adaptive agent. This is clearly shown in figure 11, with the PPBS algorithm seeming to converge long before the other two algorithms.

Following from this, the results of the fourth testing iteration, as shown in Figure 12, solidify the usefulness of the PPBS algorithm when taking into account the case of multiple consecutive episodes with transferred learning. As previously mentioned in section 3.2.1, the setup where the context transition function is known, but the likelihood function for the first observation modality (position of resources given context) is unknown presents a difficult problem for the agent. In such a setup, the agent must essentially randomly discover resource positions, and subsequently attempt to associate them with a context in order to effectively learn the likelihood function. However this association is hard, as for many of the time-steps in a given trial, the agent has a relatively imprecise belief over what context it is in.

The simulated novelty gain over a backward-smoothing of previous posterior states is key here in explaining why the PPBS agent out-performs the Sophisticated Inference agent over these 100 consecutive episodes. Figures 15 and 16 show two example behavior scenarios, representing the PPBS and Sophisticated Inference algorithms respectively. As shown in Figure 15, after randomly discovering a resource location, the PPBS agent immediately returns to the hill state. This is due to the agent,

when at the Food state in the present time-step, 'imagining' how, in subsequent time-steps, it would look back and update its posterior of the context at previous time-steps. Therefore, in its look-ahead tree-search, when considering the hill state, it realises that, for whatever context observation the hill generates, it will be able to most precisely update its beliefs about what context it was in at previous time-steps. Included in this backward view is the time-step when it was at the Food state. Thus it understands that by visiting the hill it will be able to most precisely retrospectively 'assign' a context to the Food state when compared to visiting any other non-hill state. This might be rather difficult to parse, so we here give a conceptual first-person thought process of the agent:

I have now discovered a state where a Food resource is located. I am unsure of what context I am in at this point, but if I were to move from here and visit the hill state, it would tell me what context I was in. Then, given my transition model, I would be able to work backwards and retrospectively figure out what context I might have been in when I was at the Food state. Although not maximally precise, visiting the hill would allow me to do this with more precision than moving to some other state which would not improve my knowledge of what context I am in.

This anthropomorphising of the agent's reasoning, hopefully provides more clarity on the type of nested belief structures the PPBS algorithm encourages. The agent here essentially simulates how future action would affect the process of it thinking back about previous time-steps. Wrapped up in this process is its capacity to propagate its beliefs about how its model parameters would change, were it to receive certain future observations, based on how those observations would affect its beliefs about the current and previous time-steps.

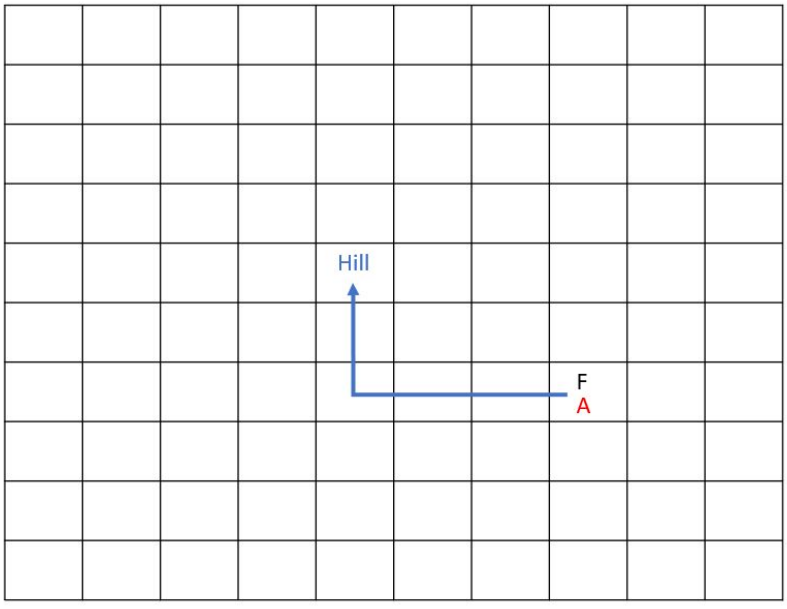

Figure 18: (**PPBS behavior**): The PPBS agent will frequently visit the hill within a trial to contextualise the information it has learned with greater accuracy.

In contrast to this, Figure 16 shows the general behavior pattern of the Sophisticated Inference agent. Upon discovering a food source, the agent does not imagine how visiting the hill might update its model parameters with maximum precision (relative to any other state). It therefore continues to explore states it has not visited before (driven by the normal Active Inference novelty mechanism) and visits the hill much less frequently (only driven to do so by the epistemic term) than the PPBS agent. Learning in this way is relatively slow, as the Sophisticated Inference agent, on average, has less precise beliefs about what context it is/was in when at a resource state.

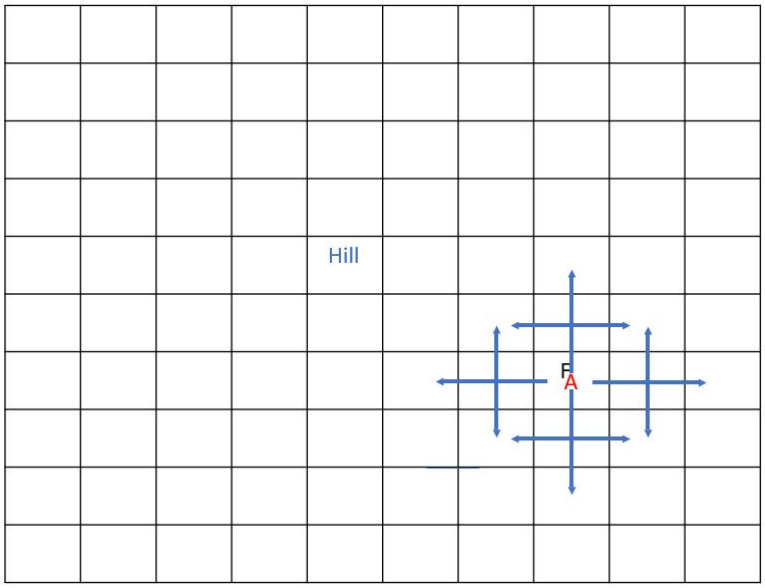

Figure 19: (**Sophisticated Inference behavior**): The Sophisticated Inference agent sees no novelty-based value in the hill, and so does not frequently revisit it to contextualise information, but rather continues to explore the state-space, seeking more novel observations.

Thus, due to the PPBS agent learning the model parameters more rapidly as a result of this behavior, it naturally outperforms the Sophisticated Inference agent when measured across multiple consecutive episodes with transferred learning.

Further Remarks on Propagated Parameter Belief Search

As discussed throughout this section, PPBS accelerates learning - a factor that becomes particularly useful in the scenario of multi-episodic implementations. The exact mathematical mechanism for this is based on the nature of the KL-divergence measure. In the context of Active Inference, novelty is the expected difference between prior and posterior model parameters.

$$Novelty = \mathbb{E}\left[D_{kl}(q(A)||q(A|o))\right] \tag{65}$$

In-line with information theory, for any given quantitative difference between two distributions (for example a difference due to different Dirichlet concentration parameter counts), the measure of diver-

gence between the two distributions is greater when that quantitative difference is more concentrated (precise) as opposed to more evenly distributed. Thus, more precise updates made to the agent's model results in a larger KL-divergence between prior and posterior models, in turn resulting in a larger novelty gain.

In this way, the PPBS agent harnesses the ability to calculate counterfactual novelty updates in a more sophisticated and nested way. This results in an agent which more accurately calculates the intrinsic value of certain action trajectories.

Overall, based on these results, it appears that the PPBS algorithm is a useful mechanism by which to drive more complex and intelligent agent behaviour. However, the algorithm has only been tested in the environments mentioned in this work. While we here hypothesise that PPBS offers a general approach that would be valid in most belief-based environments, further work is required to validate the truth of this claim.

5.3 Summary

The results discussed herein present three core findings: Firstly, they clarify the utility of the epistemic term of the **EFE** functional in the context of Sophisticated Inference schemes - which are Bayes-optimal with respect to prior beliefs. In the cases presented in this dissertation, the epistemic term was useful in encouraging exploratory behavior when the transition or likelihood functions were unknown. This is important, as when the agent starts out with inaccurate prior beliefs, despite it acting in a Bayes-optimal manner, this behavior will not be congruent with the true dynamics of the environment and will therefore be objectively non-optimal in most cases. The epistemic term, in these cases, encourages the agent to initially resolve uncertainty before attempting to determine optimal policies based on its current and expected preferences.

The second finding was in relation to Active Inference's comparative performance to Bayesian RL methods. It is clearly evident that, in the context of the environments presented in this work, Bayesian methods performed at best on equal par with Sophisticated Inference. In the case where elements of the model were unknown/unlearned it is clear that Bayes-adaptive methods (BAPOMDP) perform comparatively worse than both Sophisticated Inference and PPBS. This is most likely entirely due to the lack of added exploration heuristics to these algorithms. Bayes-adaptive methods rarely visited the hill, and only considered it a valuable state at later episodes, at a point where its model had already become relatively precise. While depriving the Bayesian RL algorithms of exploration heuristics might appear to be biasing Active Inference, it is important to remember that the Active Inference algorithms come equipped *a priori* with such mechanisms, which form part of the Expected Free Energy functional, and it was important to compare the different algorithms in as much of a 'normalised' state as possible.

The third finding pertains to the proposal of the Propagated Parameter Belief Search algorithm. This was shown to outperform both Sophisticated Inference and BAPOMDP in the environments used for this work. This is due to it incorporating *novelty* into the belief state. Its ability to compute how said novelty might evolve over counterfactual trajectories, allows it to more effectively leverage trajectories which have the potential to resolve model uncertainty. This effect is especially useful in scenarios

where mappings between states and observations are hard to make, requiring intelligent behavior in order to 'consolidate' gathered information.

6 Conclusion

For the most part, the practical work of this dissertation has provided clarity on the four core research questions presented in Section 1.1. We now iterate these questions and, with respect to each, analyse the implications of the findings this dissertation has made. We repeat the core research questions here for clarity:

1. How do Active Inference and Sophisticated Inference compare to Bayesian Reinforcement Learning in a partially observed, dynamic, context-dependent environment?

2. To what extent, and in what manner, do the *epistemic* and *novelty* terms affect agent behavior in a complex dynamic environment?

3. Can model-free Reinforcement Learning be integrated with Active Inference in ways that offer both biological plausibility and computational efficiency?

4. Can the belief propagation mechanism of Sophisticated Inference be applied to novelty to create a complex nested belief structure which encourages more intelligent behavior?

Although the comparison between the Active Inference and Bayesian RL algorithms was confined to a specific environment, it is plausible that this sort of environment is representative of a general setup - one where it is integral that the agent views trajectories not just in the context of reward maximisation, but with respect to information gathering, so as to form more precise beliefs over hidden states, and thus have higher accuracy in maximising reward in the future. It is evident that the Sophisticated methods (Sophisticated Inference and PPBS) out-perform typical Bayesian methods in this type of environment, due to their incorporation of the *epistemic* and *novelty* terms. In order to match the behaviour of the Sophisticated Active Inference algorithms, one would have to manually equip the Bayesian methods with such mechanisms. Although we have presented a set of findings with respect to a comparison between these methods, naturally these findings are still somewhat restricted in scope, and more work would need to be done to continue analysing how these different algorithms relate and compare. Specifically, testing a greater range of diverse environments would aid in providing more definite conclusions on this subject. Additionally, it would be interesting to compare Active Inference and Bayesian RL methods in scenarios where the Bayesian methods are equipped with exploration heuristics.

The second research question was thoroughly investigated via the methodologies used in the practical work of this thesis. As discussed, although these terms provide no benefit to the Sophisticated/PPBS agent when the model is completely known/accurate, it is evident that the epistemic and novelty values provide a large benefit to the agent when it has an imprecise model of the environment dynamics. While the general utility of these terms is not surprising, and has been analysed to some extent in other works (Friston et al., 2015; Friston and Herreros, 2016) the results of this dissertation provide an articulate explanation as to why, and and in what types of environments, these terms are useful.

An interesting area of future work, with respect to these two terms, could be to investigate the effect of explicitly weighting each of them individually. Although this is implicitly achieved by adjusting the precision of the preference distribution, this does not account for creating proportionally different weightings between the epistemic and novelty terms themselves (when the preference precision is adjusted, these terms will always still be equally weighted with respect to each other). Additionally, achieving such weighting via the modulation of precision is, ironically, an imprecise approach, as it is plausible to imagine that, in some scenarios, we would want to define precise preferences for the agent, as well as a large information-seeking imperative. Simply reducing preference precision to achieve information-seeking behavior is a 'zero-sum game' approach, and in some cases, we might desire the agent's behavior to be dictated by a mixture of both precise preferences, and weighted epistemic/novelty terms.

Although we did not include any explicit testing scheme for the implementation of the lower-level model-free state-space, this part of the dissertation was simply meant to act as a proof of concept approach in response to the third research question. Ultimately, however, this approach was not enough to provide adequate clarity with respect to this question, and more work is required to properly investigate this general topic. A short-coming of the implementation used in this work is the fact that the two state-spaces are entirely symmetrical. For example, moving right in the higher-level state-space is simply achieved by moving right in the lower-level state-space. This is rather redundant, and a more interesting approach would be to have a significant difference between the nature and functioning of the action and state modalities of the two levels, resulting in the construction of connections between these two sets of modalities which would otherwise be unrelated. This would more effectively capture the essence of this research question, which was to determine whether we could use model-based and model-free paradigms to solve different yet connected tasks within the context of a multi-layered environment.

To answer the fourth research question, we developed the Propagated Parameter Belief search algorithm. While this is certainly only a single way (amongst many) that one could extend the functionality of Sophisticated Inference's belief propagation scheme, this algorithm appropriately captures the functionality of nested counterfactual parameter belief propagation. The scope of testing PPBS, however, has been narrow, with only a specific type of environment being used to analyse its performance. It is entirely plausible that it would do no better than other approaches in other specific environments. However, we here hypothesise that PPBS will potentially offer improved performance in any dynamic environment where the counterfactual dynamic belief searches that PPBS encourages are required to achieve accurate beliefs over models of environment dynamics that would otherwise be difficult to learn. There is much room for future investigation here, particularly to formally analyse and test the bounds of the utility that PPBS can offer. Nevertheless, we hope that the introduction of this algorithm proves useful to the field, and encourages the development of more intelligent Active Inference agents.

The field of Active Inference is currently experiencing an influx of interest due to its ability to represent biologically plausible models of behavior, and is seeing experimentation and integration into a wide variety of fields, from psychoanalytic theory (Connolly, 2018) to deep learning (Millidge, 2021; Fountas

et al., 2020). As the field expands and evolves upon its application and experimentation within various disciplines, it is important to analyse, and develop upon, its core theoretical and algorithmic framework, in order to fully understand its capacity for integration with other such fields.

This thesis has aimed to present a thorough articulation of Active Inference's inner algorithmic and behavioural workings, specifically when compared to other machine learning methods. In addition, this work has proposed a variation of the Sophisticated Inference algorithm, where the agent incorporates beliefs about novelty into its planning - propagating beliefs about how hidden states might change in the future, how those beliefs would affect its beliefs about parameters, and, in turn, how these counterfactual posterior beliefs over parameters and hidden states would affect its posterior beliefs about hidden states and parameters at previous time-steps. This complex nested belief structure potentially has the effect of creating intricate and intelligent behaviour, which cannot be achieved by other comparative machine learning algorithms. .

Throughout the theory and practical implementation of this thesis, the theme of *affect* has been present. Although we have offered no formal analysis on it, in an attempt to contain the scope of this work, it will not surprise the reader that the types of environment (multi-objective drives), and agent (Active Inference), were heavily inspired by the themes of biological plausibility, homeostasis and, most broadly, the Science of Consciousness. Moving forward, we hope to use this work as a foundation upon which to build future research ideas involving these themes. Particularly, such future work would involve the incorporation of inference into the agent's internal measurement of its time without resources (deviation from categorically separate homeostatic states), different reward imperatives that are less 'symmetrical' in their action domains (and so require a more accentuated demarcation of policy prioritisation), and the incorporation of additional machine learning devices such as Recurrent Neural Networks and Monte Carlo tree-search techniques to improve computational efficiency and expand functionality.

The possibilities for Active Inference agents are vast, and while our understanding of the mysteries surrounding minds, intelligence and subjective feeling is, as of yet, limited, it remains to be seen just how far the field can take us in accelerating our knowledge of the conscious brain - that object which is unfathomable to we who are it.

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.
— T. S. Eliot

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938.

Amari, S.-I. (1995). Information geometry of the em and em algorithms for neural networks. *Neural Networks*, 8(9):1379–1408.

Baum, L. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563.

Bellman, R. (1958). Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228–239.

Berger-Tal, O., Nathan, J., Meron, E., and Saltz, D. (2014). The exploration-exploitation dilemma: A multidisciplinary framework. *PLoS ONE*, 9(4):e95693.

Bishop, C. (2006). Pattern recognition and machine learning.

Blei, D., Kucukelbir, A., and Mcauliffe, J. (2016). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877.

Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211.

Boyles, R. (2012). Artificial qualia, intentional systems and machine consciousness.

Chalmers, D. (1995). *Facing Up to the Problem of Consciousness*, pages 3–34. Oxford University Press.

Chella, A. and Manzotti, R. (2007).

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. the behavioral and brain sciences.

Clark, A. (2015). Radical predictive processing. *The Southern Journal of Philosophy*, 53:3–27.

Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., and Abbeel, P. (2018). Model-based reinforcement learning via meta-policy optimization. *ArXiv*.

Colombo, M. and Wright, C. (2021). First principles in the life sciences: the free-energy principle, organicism, and mechanism. *Synthese*, 198(14):3463–3488.

Conant, R. C. and Ashby, W. R. (1970). Every good regulator of a system must be a model of that system. *International Journal of Systems Science*, 1:511–519.

Connolly, P. (2018). Expected free energy formalizes conflict underlying defense in freudian psychoanalysis. *Frontiers in Psychology*, 9.

Costa, J., Silva, C., Antunes, M., and Ribeiro, B. (2017). Adaptive learning for dynamic environments: A comparative approach. *Engineering Applications of Artificial Intelligence*, 65:336–345.

Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The helmholtz machine. *Neural Computation*, 7(5):889–904.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Drake, A. (1962). Observation of a markov process through a noisy channel.

Duff, M. and Barto, A. (2002). Optimal learning: computational procedures for bayes-adaptive markov decision processes.

Erik, P., Randy, V. ., and Cogill (2015). Expectation-maximization for bayes-adaptive pomdps. *Journal of the Operational Research Society*, pages 1605–1623.

Fountas, Z., Sajid, N., Mediano, P., and Friston, K. (2020). Deep active inference agents using monte-carlo methods. *ArXiv*.

Friston, K. (2003). Learning and inference in the brain. *Neural Networks*, 16(9):1325–1352.

Friston, K. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):815–836.

Friston, K. (2008). Hierarchical models in the brain. *PLoS Computational Biology*, page 4.

Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301.

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

Friston, K. (2013). Life as we know it. *Journal of The Royal Society Interface*, 10(86):20130475–20130475.

Friston, K. (2019).

Friston, K. and Ao, P. (2012). Free energy, value, and attractors. computational and mathematical methods in medicine.

Friston, K., Costa, L., Hafner, D., Hesp, C., and Parr, T. (2021). Sophisticated inference. *Neural Computation*, 33:713–763.

Friston, K. and Herreros, I. (2016). Active inference and learning in the cerebellum. *Neural Computation*, 28(9):1812–1839.

Friston, K., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3):70–87.

Friston, K., Mattout, J., and Kilner, J. (2011). Action understanding and active inference. *Biological Cybernetics*, 104(1-2):137–160.

Friston, K., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., and Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, 6(4):187–214.

Friston, K. J., FitzGerald, T. H. B., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: A process theory. *Neural Computation*, 29:1–49.

Ghavamzadeh, M., Mannor, S., Pineau, J., and Tamar, A. (2015). Bayesian reinforcement learning: A survey. *Found. Trends Mach. Learn*, 8:359–483.

Guez, A., Silver, D., and Dayan, P. (2012). Scalable and efficient bayes-adaptive reinforcement learning based on monte-carlo tree search. *Journal of Artificial Intelligence Research*, 48:841–883.

Ha, D. and Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. *NeurIPS*.

Harshvardhan, G., Gourisaria, M., Pandey, M., and Rautaray, S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Comput. Sci. Rev*, 38:100285.

Hesp, C., Smith, R., Parr, T., Allen, M., Friston, K., and Ramstead, M. (2021a). Deeply felt affect: The emergence of valence in deep active inference. *Neural Comput*, 33(2):398–446.

Hesp, C., Smith, R., Parr, T., Allen, M., Friston, K., and Ramstead, M. (2021b). Deeply felt affect: The emergence of valence in deep active inference. *Neural Computation*, 33:398–446.

Hinton, G. and Zemel, R. (1993). Autoencoders, minimum description length and helmholtz free energy.

Hohwy, J. (2013).

Hohwy, J. (2016). The self-evidencing brain. *Noûs*, 50:259–285.

Katt, S., Oliehoek, F., and Amato, C. (2018). Bayesian reinforcement learning in factored pomdps.

Kirchhoff, M., Parr, T., Palacios, E., Friston, K., and Kiverstein, J. (2018). The markov blankets of life: autonomy, active inference and the free energy principle. *Journal of The Royal Society Interface*, 15(138):20170792.

Koch, C., Massimini, M., Boly, M., and Tononi, G. (2016). Neural correlates of consciousness: progress and problems. *Nature Reviews Neuroscience*, 17(5):307–321.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. (2018). Model-ensemble trust-region policy optimization. *ArXiv*.

Mann, T., Choe, Yoonsuck, ., Marc, Editor, ., Deisenroth, Peter, ., Szepesvári, and Peters, J. (2012). Directed exploration in reinforcement learning with transferred knowledge. *Journal of Machine Learning Research*, 24:59–75.

Meissner, G. (2019). Artificial intelligence: consciousness and conscience. *AI SOCIETY*, 35:225–235.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

Michie, D. (1968). "memo" functions and machine learning. *Nature*, 218(5138):306–306.

Millidge, B. (2021). Applications of the free energy principle to machine learning and neuroscience. *ArXiv*.

Millidge, B., Tschantz, A., and Buckley, C. (2021). Whence the expected free energy. *Neural Computation*, 33:447–482.

Morris, A. and Cushman, F. (2019). Model-free rl or action sequences? *Frontiers in psychology*, 10:2892.

Mumford, D. (1992). On the computational architecture of the neocortex. *Biological Cybernetics*, 65(2):135–145.

Norvig, P. (1991). Techniques for automatic memoization with applications to context-free parsing. *CL*.

Paquet, S., Tobin, L., and Chaib-Draa, B. (2005). An online pomdp algorithm for complex multiagent environments. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05*, page 5. ACM Press.

Parisi, G. (1988). Addison-Wesley, Redwood City.

Parr, T. and Friston, K. (2019). Generalised free energy and active inference. *Biological Cybernetics*, 113(5-6):495–513.

Parvizi, J. and Damasio, A. (2001). Consciousness and the brainstem. *Cognition*, 79(1-2):135–160.

Pateria, S., Subagdja, B., Tan, A.-H., and Quek, C. (2021). Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv*, 54.

Pathak, D., Agrawal, P., Efros, A., and Darrell, T. (2017). *Curiosity-Driven Exploration by Self-Supervised Prediction*, pages 488–489. CVPRW.

Pearl, J. (1989). Probabilistic reasoning in intelligent systems - networks of plausible inference. morgan kaufmann series in representation and reasoning.

Pezzulo, G., Rigoli, F., and Friston, K. (2018). An active inference view of cognitive control. *Frontiers in Psychology*, 3:294–306.

Poupart, P. and Vlassis, N. (2008). *Model-based Bayesian Reinforcement Learning in Partially Observable Domains*.

Rao, R. and Ballard, D. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87.

Ross, S., Chaib-Draa, B., and Pineau, J. (2007).

Sajid, N., Ball, P., Parr, T., and Friston, K. (2021). Active inference: Demystified and compared. *Neural Comput*, 33(3):674–712.

Seth, A. (2018). Consciousness: The last 50 years (and the next). *Brain and Neuroscience Advances*, 2:239821281881601.

Seth, A., Gouveia, M., Curado, D., Mendonça, and Bloomsbury (2020). *The brain as a prediction machine*.

Shannon, C., Weaver, W., and Wiener, N. (1949). The mathematical theory of communication. *Physics Today*, 3(9):31–32.

Sikl, R. (2001). Hermann von helmholtz (1821-1894) on perception. ceskoslovenská psychologie.

Smith, R., Friston, K., and Whyte, C. (2021). A step-by-step tutorial on active inference and its application to empirical data.

Smith, R., Schwartenbeck, P., Parr, T., and Friston, K. (2020). An active inference approach to modeling concept learning. *bioRxiv*.

Solms, M. (2019). The hard problem of consciousness and the free energy principle. *Frontiers in Psychology*, 9.

Solms, M. and Friston, K. (2018). How and why consciousness arises: Some considerations from physics and physiology.

Sondik, E. (1971). The optimal control of partially observable decision processes.

Srinivasan, M., Laughlin, S., and Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. In *Proceedings of the Royal Society of London. Series B. Biological Sciences*, volume 216, pages 427–459.

Sutton, R. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163.

Sutton, R. and Barto, A. (2018). Reinforcement learning: An introduction.

Tononi, G. (2004). An information integration theory of consciousness. *BMC neuroscience*, 5:42.

Tschantz, A., Millidge, B., Seth, A., and Buckley, C. (2020). Reinforcement learning through active inference. *ArXiv*.

Çatal, O., Wauthier, S., Verbelen, T., Boom, C., and Dhoedt, B. (2020). Deep active inference for autonomous robot navigation. arxiv, abs.

7 Supplementary Material

The code for the practical work of this thesis can be found at https://github.com/pianopwner/active-inference. Therein are instructions on how to run each of the different algorithms and environments.