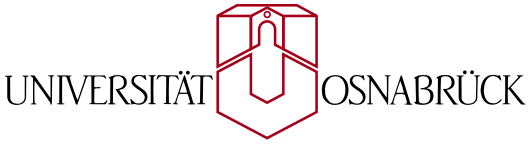

Fachbereich Humanwissenschaften
Institute of Cognitive Science

Bachelor's Program Cognitive Science

Bachelor Thesis

Towards a minimalistic free energy agent

Esther Chevalier
Matriculation number 972437
23/12/2020 - 13/04/2021

First supervisor: Georg Schröter, M.Sc.
Institute of Cognitive Science
Osnabrück

Second supervisor: Prof. Dr. Gordon Pipa
Institute of Cognitive Science
Osnabrück

Esther Chevalier
Springmannskamp 3
49090 Osnabrück

E-mail: echevalier@uos.de
Tel: 0157 35 72 03 46

Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

City, date

signature

Contents

1 Introduction **1**

2 Theoretical framework **3**

2.1 Motivation 3

2.2 Agent and environment 5

2.2.1 Environment 5

2.2.2 Artificial agent in a FEP framework 6

2.2.3 Generative model of agent 6

2.3 Free energy 7

2.4 Active inference 9

2.4.1 Planning as inference 9

2.4.2 Active inference 10

2.5 Learning 12

3 The rat in a maze - a practical example **14**

3.1 Problem statement 14

3.1.1 Environment 14

3.1.2 The rat - an active inference agent 17

3.2 Optimal behavior 19

3.3 Active inference 21

3.3.1 Agent's structure 21

3.3.2 Reducing ambiguity 22

3.4 Learning 23

3.5 Episode 23

3.5.1 Initialization 24

3.5.2 Run 24

4 Experiments and results **26**
4.1 Active inference 26
4.1.1 Agent's behavior 26
4.1.2 Comparison between optimal behavior and the agent's behavior 27
4.1.3 Comparison between agent's expected reward and true reward 28
4.2 Learning 29

5 Discussion of the implementation **33**
5.1 Design choices 33
5.1.1 Active Inference 33
5.1.2 Learning 37
5.2 Difficulties during the implementation process 38
5.3 Literature about FEP 39
5.4 Outlook 40

6 Conclusion **41**

7 Bibliography **45**

1 Introduction

Karl Friston proposes and defends a new formulation of habit-forming, learning, and behavior optimization in biological agents described in Friston et al. (2006). The principle he developed, namely the free energy principle (FEP), provides a behavior and behavior learning model. The framework has been applied to a number of research fields such as developmental biology (e. g. Levin et al., 2015 applied FEP for embryogenesis), psychology (e. g. for predicting saccadic eye movements as did Friston et al. (2012)) and predictive coding (e. g. Bogacz, 2017). Additionally, the theory was used as a mathematical unification of multiple theories like adaptive evolution, biological systems theory, and cognitive science in Ramstead et al. (2018). FEP is also used to specify the behavior of artificial agents, as did Ueltzhöffer (2018) for example. As for now, most of the literature about artificial agents based on FEP are used for demonstrating the theory (see Friston et al., 2016, McGregor et al., 2015 and Sajid et al., 2019); the design and implementation process often get sidelined. Furthermore, as explained in Andrews (2018, p.12), the mathematical formalism coming from information theory and statistical physics hinders non-experts to understand fully the methods used.

This paper aims to implement and design an artificial agent based on the free energy principle. It is designed to be as minimal as possible to demonstrate which elements are strictly necessary to define such an agent. This groundwork enables non-expert who have basic knowledge of probability theory, for example, undergraduate Cognitive Science students, to understand the fundamental workings of the free energy principle. Moreover, it should provide to the reader a basis to implement a FEP agent on their own.

To achieve this goal, the paper will first dive into the free energy principle's theoretical underpinnings. Then it will take a look at the agent implemented, in this case, a rat searching for a piece of cheese and evaluate its behavior when performing active inference and learning. Lastly, the paper will discuss the design choices made

during the implementation process, the difficulties that were faced, and quickly review the available literature.

2 Theoretical framework

2.1 Motivation

The FEP introduced by Karl Friston gained a lot of attention and interest in the past decade. The framework was applied to fields as different as clinical psychology (e. g. for explaining schizophrenia in Fletcher and Frith, 2009), machine learning (Ueltzhöffer, 2018), computational modelling of the brain (Friston, 2010), and predictive coding (Adams et al., 2015). Friston (2010) defended FEP as being a fundamental theory that unifies several theories such as neural Darwinism and information theory in the field of neuroscience. Friston also proposed to use FEP in the field of epidemiology to predict the evolution of the CoViD-19 pandemic in a technical report Friston et al., 2020. The free energy principle would not only be a unifying brain theory (Friston, 2010) but also sufficient to describe how a biological system maintains itself over time as explained by Friston, 2010, p. 128.

Arguing in this sense comes to answer the questions in Schrödinger, 1945, Chapter IV: How does living matter maintain the negative entropy necessary to avoid the decay to thermodynamical equilibrium? The free energy principle should provide an answer to how living organisms can maintain their boundaries to the environment over time. It aims to describe how they keep their internal states stable to ensure their survival. Though FEP is not falsifiable as elucidated by Andrews, 2020, p.3, it can be used to make models in a variety of contexts and fields and will probably continue to tract attention in scientific research.

The framework was also used to model and design artificial agents. This has several advantages. As stated in Sajid et al. (2019, p.6), the agent makes decision on how to act in the environment based on its beliefs about true environmental states. Therefore, the environmental states have not an inherent value that the agent would have to learn, but rather has a measure of how useful a state is to

attain its goal. The behavior arising is supposedly more realistic (Sajid et al., 2019, p.2) and does not rely on an externally defined reward function, which requires a complex design for an agent in a complex environment.

Furthermore, the goal of a free energy agent is defined as a probability distribution: each possible outcome is weighted by the agent's preferences. It will therefore tend to spend time in states that yield a certain outcome proportionately to the agent's prior preferences of this particular outcome. Sajid et al. (2019, p.6) explains it leads the agent to a more nuanced behavior than a simple reward-maximising process like reinforcement learning.

Into the bargain, there is also the possibility to take a non-stationary environment into account. Friston et al. (2015, p.7) defines how the agent can find itself in an environment that has certain features depending on context. The agent evaluates how likely a certain context is and can therefore change its behavior accordingly. This allows a much more flexible approach when dealing with non-stationary environments.

In addition, as explained in Sajid et al. (2019, p.3), the reward signal from the environment is not treated differently than any other type of outcomes. Conceptually, it is possible that an agent receives several types of signals from the environment and the agent has a preference over each of them. The agent's behavior is not uniquely driven by the environmental rewards, but rather by its features (i. e. prior preferences). This enables the agent to be capable of learning its prior preferences over outcomes by interacting with the environment as commentated by Sajid et al. (2019, p.4).

The properties summarized above arise by minimizing a free energy functional, further explained in Section 2.3. The process of minimizing the free energy in an artificial environment is called *active inference* (Friston et al., 2016), detailed in Section 2.4. Several papers discussed the formal framework and experimented with artificial agents using active inference. For example in Ueltzhöffer (2018), the car in the mountain-car problem has to reach the peak of the mountain by learning the appropriate acceleration. This agent is set in a continuous state-space agent and is using an artificial neural network to minimize free energy. Sajid et al. (2019) introduced an agent acting in a discrete state-space environment using OpenAI Gym's FrozenLake environment and comparing it to different reinforcement learning

agents. Here, the learning process is deterministic, but the implementation was not designed to be as minimal as possible. It relies on external packages automatically generating a Markov decision process (MDP). McGregor et al. (2015) introduced a minimal free energy agent to lay out the fundamental concepts necessary for such an agent to exist, but without providing the associated code.

This paper aims to define and program a simple artificial agent in a discrete-space environment. The implementation should be simple and programmed from scratch rather than relying on conceptually complex external packages. The goal is to enable the reader to understand the fundamental notions of the free energy principle. It provides a source code that can be used as a sandbox to experiment with hyper-parameters and functions. It should also provide first guidance to the reader for constructing a new free energy agent. The source code is provided at .

After establishing the theoretical basis, the implementation will be presented and discussed. To understand the practical part, it is helpful to first understand how artificial agents and their environment are defined and how the agent takes decisions on how to act in the environment using active inference.

2.2 Agent and environment

2.2.1 Environment

Only discrete state-space settings with discrete time steps will be discussed in this paper. The agent is bound to an environment and can perceive some of the environmental states. As described in Friston et al. (2006, p.73), the agent does not have full knowledge about the true environmental states though, due to the restrictions of its sensory receptors or because the environment is too complex. The true state of the environment that the agent cannot perceive directly are called *hidden states*.

Friston et al. (2006, p.73) describes the sensory input an agent can perceive in a given environment as a functional over actions. This means, when the agent performs an action, it changes the hidden state of the environment and therefore perceives different sensory signals. There are sensory signals the agent prefers, and that indicates it is able to endure over time, which is one characteristic of the biological agents described by Friston et al. (2006, p.74). To be able to perceive the

preferred sensory signals, the agent needs to understand how its actions influence its perception, i. e. the causal structure of the environment. The true causal structure of the environment is unknown to the agent, but the agent can design an approximate model of the environment's causal structure, which will be helpful to decide on how to behave in the environment.

2.2.2 Artificial agent in a FEP framework

Artificial agents are entities bound to a specific environment and capable to change it by performing actions. They intend to achieve a specific goal or to observe specific outcomes. They employ a specific, pre-defined set of actions to do so. In a FEP framework, the agent typically only has limited time to achieve the goal, as stated in Sajid et al. (2019, p.9). The agent plans its actions up to this time limit. To do so, it compares multiple action sequences called *policies*. Sajid et al. (2019, p.10) defines the policy π as a sequence of choices the agent can take at each time step t. Each action a_t is only dependent on t and not on the state the agent finds itself in contrarily to traditional reinforcement learning (RL). The agent evaluates a number of policies, rates the outcomes of each action sequence, and compares them. To generate the outcomes in order to compare future outcomes, the agent needs a model of the environment and its rules, called the *generative model*, defined in Schwöbel et al. (2018, p. 2534)

2.2.3 Generative model of agent

In a FEP framework, the agent is inferring over hidden environmental states and their outcomes by using probabilities distribution. An FEP agent considers the state s_t as a probability distribution over all possible hidden states and uses its belief about (current and future) states to make decisions. It needs a model of the environment and its rules to anticipate the outcome of an action. Schwöbel et al. (2018, p. 2534) explains how the agent infers in which state it would be after completing this action through a generative process. It is called generative process because the agent generates outcomes and rates them before actually performing any action and thus perceiving any outcome. The agent repeats the process until it arrives at a state from which no action is possible or until the time limit is reached.

To keep generating the outcomes of each action in the inferred states, the agent relies on a model of the environment, called *generative model*.

Friston, 2010, p. 129 defines the generative model as a joint probability distribution over outcomes o, states s and policies π: $p(o, s, \pi)$. The outcomes and states are inferred at each time step t. Additionally, the policies π are evaluated separately and can be marginalized into the distribution after evaluating them individually. Thus, the generative model can be rewritten as $p(o_t, s_t) = p(o_t|s_t)p(s_t)$. o_t is defined as the outcome at time step t and s_t is the state at time step t.

From there, the agent's generative model is initialized with three prior distributions:

- $A_t = p(o_t|s_t)$, the likelihood of outcomes given states decomposed in the previous equation.
- $B_t = p(s_t|s_{t-1}, a_{t-1})$, where a_t is defined as the action a taken at time step t. This distribution encodes the transition probability to be at state s given the state the agent was previously in and the action it took.
- $p(o_t)$, the prior preferences over observations i.e. outcomes at time step t. It encodes what the agent will prefer and therefore will influence the choice of actions taken.

Additionally, the agent's behavior is regulated by a temperature term T, which specifies the agent's exploration/exploitation trade-off. To keep the agent minimal this paper considers A, B and $p(o)$ as independent from time t.

2.3 Free energy

Friston (2010, p.128) describes free energy as a concept coming from statistical thermodynamics. It is characterized as being used to convert difficult integration problems over probability distributions into optimization problems that are easier to solve. Variational free energy on the other hand is a quantity used in information theory. The core aspect of FEP is minimizing free energy to optimize an agent's

behavior; the free energy used here is the variational free energy (VFE). In the remainder of this paper, free energy will always refer to VFE.

To further explain how the concept of free energy is used in active inference, it is useful to introduce a central concept of information theory: information content. The information content $I(x)$ (also called Shannon information or self-information) is a basic quantity derived from a probability distribution p. The information content of an event x is defined as $I(x) = -log_b(p(x))$. The base b of the logarithm corresponds to a multiplicative scaling factor of self-information. Exclusively relative information content is considered in a FEP context, therefore, the logarithm was arbitrarily set to the natural logarithm for the remainder of the paper.

The agent models environment's outcomes with its generative model $p(o, s, \pi)$ (see 2.2.3). Additionally, it needs to approximate the environmental structure and rules with a separate distribution defined as $q(s, o, \pi)$ which is called recognition density by Friston (2010, p.2) and variational density by Sajid et al. (2019, p.8).

In Sajid et al. (2019, Eq. (6)), the variational free energy of q is defined as:

$$F = \sum_{s} q(s) \frac{q(s, o)}{p(o, s)}$$

where q is a distribution approximating the environment's causal structure. F is then rewritten as $F = D_{KL}[q(s)||p(s|o)] - \log p(o)$ in Sajid et al. (2019, Eq. (10)). Here $D_{KL}[q(s)||p(s|o)]$ refers to the Kullback-Leibler divergence (KL) of $q(s)$ and $p(s|o)$. The Kullback-Leibler divergence measures how close the first probability distribution is to the latter; KL is also called relative entropy. The definition of F shows variational free energy can be expressed as inequality letting the implicit information content of the prior preference $p(o)$ appear: $F \geq -\log p(o)$. This is possible due to the strict positivity of the KL.

The agent seeks to maintain itself in states with its preferred outcomes, which minimize the self-information (i. e. surprise) of $p(o)$. Based on the inequality $F \geq -\log p(o)$ discussed above, variational free energy turns into an upper-bound on surprise. Minimizing VFE implicitly minimizes the agent's surprise about outcomes and leads it to stay in states whose outcomes it favors.

The KL term can also be translated in agent's behavior. $q(s)$ describes the approximate posterior belief about hidden states after performing an action sequence and $p(s|o)$ models the hidden states given preferred outcomes. The posterior distribution $q(s)$ should be as close as possible to the prior distribution of states $p(o|s)$ observed when the preferred outcomes are met. Reducing the KL term comes back to take the agent in states with outcomes it favors over others. To achieve this, it needs to perform actions leading it to those states.

The optimizaton (i. e. minimization) of VFE is realised during the planning phase, before the agent performs the next action. The optimization process is called active inference.

2.4 Active inference

2.4.1 Planning as inference

The generative model treats outcomes, states and policies (i.e. action sequences) as hidden variables. This allows treating both phenomena, the perception of the environment (i.e. observations) used to infer the current hidden state and the actions of the agent within the same mathematical framework. The agent evaluates the individual policies separately to make a decision about its future actions using Bayesian inference. This process is called *planning as inference*. Schwöbel et al., 2018

The generative model plays two roles at once: it predicts possible observations and contains the prior beliefs about actions which encode beliefs about optimal policies. At each time step, the agent possesses a prior belief about the hidden state of the environment $p(s_t)$. Using the generative process, it infers its next hidden state if it were to take action a_t, with:

$$p(s_{t+1}) = p(s_t) * p(s_{t+1}|s_t, a_t) = p(s_t) * B$$

This posterior belief about the hidden state at time $t + 1$ is later used as prior to infer the belief about the hidden state s_{t+2} and so forth. It then forms an expectation

about outcomes o_{t+1} using the likelihood A:

$$p(o_{t+1}) = p(o_{t+1}|s_{t+1})p(s_{t+1}) = A * p(s_{t+1})$$

The expected outcome $p(o_{t+1})$ is distinct from the preferred observations $p(o)$ mentioned in 2.2.3. It rather represents what the agent expects to see at time step $t+1$, while $p(o)$ describes what the agent *wants* to see at the next time step. The agent ultimately chooses an action based on the future states and outcomes predicted through Bayesian.

Since the environment is typically complex, the hidden states s are difficult to infer over. Therefore, the posterior beliefs $p(s_t)$ about hidden states have to be approximated. In a free energy framework, this happens through *active inference*.

2.4.2 Active inference

As first intuition, active inference is the process of an agent acting to see what it prefers. To do so, the agent relies upon the knowledge of the posterior beliefs about the environmental hidden states. Since the agent does not have direct access to present or future hidden states, a new approximate distribution is introduced: $q(s, o, \pi)$, further elucidated by Schwöbel et al., 2018, p.2538. This distribution is a tool that will be used to control the planning process. Its parameters are entirely known and predefined. It provides two advantages: it is possible to use it to during the planning process and it allow using a posterior belief about hidden states before the observations are obtained.

The true environmental posterior about hidden states is therefore replaced by the approximate posterior about hidden states $q(s|\pi)$. As with the true posterior distribution, the policies π are marginalized in a second step and can be disregarded for now. The approximate expected outcome $q(o_{t+1})$ is computed similarly as the true expected outcome $p(o_{t+1})$ after performing action a_t, as following:

$$q(o_{t+1}) = p(o_{t+1}|s_{t+1})p(s_{t+1}|s_t, a_t)q(s_t) = A \cdot B \cdot q(s_t)$$

It was previously discussed, why VFE is an upper-bound on surprise (see Section 2.3. Since the agent needs to approximate the VFE to decide which action to perform next, it needs to approximate the free energy of each policy. According to Sajid et al. (2019, Def.5), the expected free energy (EFE) is used to find support for plausible policies based on outcomes that have not been observed yet. As described in Sajid et al. (2019, Eq. (24)), the expected free energy is the sum of the following two terms:

- *Expected cost.* This term measures how similar the expected outcome $q(o_{t+1})$ is to the prior preferences $p(o)$. The similarity is measured using the Kullback-Leibler divergence. It is equal to the relative entropy of both distributions.

- *Expected ambiguity.* This terms measures how certain the agent can be to find itself in the next state. It is equal to the sum of the approximate probabilities $q(s_{t+1})$ weighted by the entropy of likelihood $p(o|s)$. The entropy of $p(o|s)$ is equivalent to the expected value of surprise of $p(o|s)$; in Sajid et al. (2019, p.11), it is described as the ambiguity over outcomes for each hidden state and is denoted by the function H.

Formally, the expected free energy G is defined as:

$$G = D_{Kl}[q(o_{t+1})||p(o)] + \sum_{s} q(s_{t+1})H[p(o|s)]$$

.

The EFE of an entire policy is simply the sum of the EFE of each of its actions. Finally, the agent needs to chooses its next action. It first computes the expected free energy of all possible policies. The agent then computes the softmax of the negative EFEs - this posterior probability distribution $q(\pi)$ describes the probability of following policy π as explained by Schwöbel et al. (2018, Table 1). The softmax function here is regulated by a temperature term inherent to the agent. Sajid et al. (2019, p.15) explains that in a practical setting, the approximate posterior probability of each action $q(a_t)$ is the sum of the probabilities of all policies π which action at time step t is equal to A. The action is then sampled from the distributions over actions. If no exploration is required, the action with the highest probability is performed.

Active inference is the process of choosing actions at every time step. It does make the agent act over time, but it is possible to integrate a learning process into the active inference framework.

2.5 Learning

The agent's behavior relies heavily on its generative model, specifically on the likelihood distribution A, the transition distribution B and the prior preferences $p(o)$. The temperature also influences the action selection process and specifies the trade-off between exploration and exploitation.

One possibility to update and optimize the generative model and the temperature is through gradient descent as did Sajid et al. (2019, p.14). The gradient descent is computed over the negative free energy by using a mean-field approximation. Schwöbel et al. (2018, p.2532) argues the Bethe approximation is a better choice: While the mean-field approximation assumes that the belief over states are time independent, the Bethe approximation takes these statistical dependencies into account. It is also possible to minimize free energy by optimizing the agent's parameters by using a neural network, as did Ueltzhöffer (2018).

In this paper, we will focus on learning using gradient descent computed with the mean-field approximation as it is still a very common method in the literature. It was also used in Friston et al. (2016) which serves as foundation of the practical part of this paper. Sajid et al. (2019, p.14) defines the gradient ε of the negative variational free energy with respect to states as:

$$\varepsilon_t = (\log A \cdot o_t + \log B_{t-1} s_{t-1} + \log B_t s_{t+1}) - \log s_t$$

This gradient is used to update the likelihood and transition distributions (i. e. $p(o|s)$ and $p(s_{t+1}|s_t, a_t)$).

Furthermore, it is also possible for the agent to learn the temperature, that means the exploration exploitation trade-off. This differs from e. g. traditional RL where the exploration is maximized at the beginning of the learning and gradually minimized. The FEP agent takes into account that a further exploration might useful in

some environments, therefore integrating the concept of perpetual learning into the framework.

An additional possibility is to let the agent learn which prior preferences it has. Sajid et al. (2019, p.26) illustrates this by letting both the likelihood distribution and the prior preferences be learned, using a prior over prior distribution. They define a conjugate prior distribution. The agent then explores the environment entirely without getting any rewards. The learning is then incrementally enabled; the agent then observes the outcomes of its actions and as one outcome becomes more familiar, the agent will change its behavior to seek those outcomes.

Until here, this paper dived into the theoretical underpinnings of the free energy principle. The following pages will focus on the practical implementation of a simple active inference agent, which will be discussed in the next chapter.

3 The rat in a maze - a practical example

This section considers a toy example of a free energy agent acting in a simple environment. It allows visualizing and making experiments with a free energy agent in its minimal form. In order to keep the agent's implementation minimal, the following decisions were made: the agent acts in a discrete state-space and in a time-restricted manner. The agent was based on Karl Friston's example in Friston et al., 2016.

3.1 Problem statement

3.1.1 Environment

The environment used here has four distinct locations arranged like a T-shape maze: location 0 is the center, location 1 the upper left arm, location 2 the upper right arm, and location 3 the bottom of the maze, as shown below. The agent starts each episode at location 0 i.e. the center. From there, it needs to take a sequence of actions that leads it to the reward, which in this setting is a piece of cheese. The cheese is either in the upper left arm i. e. location 1 or the upper right arm i. e. location 2.

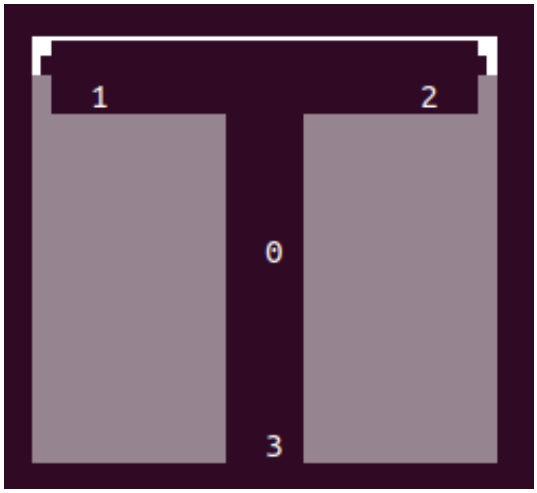

Figure 3.1: Empty environment with numbered positions. The rat always starts at location 0, the cheese is either at location 1 or location 2 and the cue at location 3 either hints at the upper left-hand corner or the upper right-hand corner based on parameter r on the true location of the cheese.

At the bottom of the maze (i.e. position 3), a discriminative cue (or hint) indicates whether the cheese is in the right or left arm (i. e. location 1 or location 2). The reliability of the cue is a free variable r that is fixed at the initialization of the environment. If the cue's reliability r is 1, it always indicates the cheese's location accurately, if $r = 0$, the cue shows the correct location 50% of the time. The cue is only visible if the agent goes to location 3. The agent find itself facing two possibilities: either the reward is on the left side or on the right side. These possibilities are introduced in Friston et al. (2016) as *contexts*. The agent has to infer which environmental *context* is currently the case and takes decisions based on the context to get the reward.

Each episode ends automatically after two time-steps. It is possible to reach each location in one time-step from any other location. The agent may revisit all locations and can stay at its current position. However, if it goes in either arm, that is either location 1 or 2, the rat is trapped and cannot leave anymore. These locations are called *absorbing states* (Sajid et al., 2019, p.18).

The environmental states are defined by a probability distribution over all possible hidden states. It encodes the position of the agent and does not contain any

uncertainty. In the present implementation, the true hidden state is encoded as an array of shape 4×1. The index of each entry specifying the position and the entry yields the probability that the agent is positioned at this location. Since there is no uncertainty, the probability is either 0 or 1. The position of the agent is updated after each time-step.

To determine what happens after the rat made a move, one needs to define the environment's causal structure. It is encoded by a function, which in principle could be any that maps the agent's state s_t and an action a_t to a new state s_t and a reward. To be comparable to the agent's generative model, the environment causal structure has been defined by two matrices A and B.

- A: Encodes the position of the cheese. When the rat gets to the cheese, it receives a positive reward, set to 1. The rat immediately eats the cheese when it finds it, henceforth the reward is set to 0 for all locations after the rat got to the cheese. A corresponds to a function mapping state s_t to a reward. A is a vector of shape 1×4

- B: Describes the transition from one state s_t to the next state s_{t+1}. The transition always depends on the action taken. The shape of matrix B corresponds to 4 possible actions times 4 possible positions at t times 4 possible positions at $t+1$. The slice index corresponds to the action taken, the rows to the position at t and the columns to the position at $t+1$. For example, the transition matrix for action 0 (i. e. go to location 0) is:

$$B_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

  It describes the agent is able to get to the center of the maze (i. e. location 0) if it was at location 0 or 3 beforehand. If the rat was inside one of the upper arms, it cannot escape them anymore.

At the initialization of the environment, it is possible to set two parameter: the cue reliability r indicating how often the cue is correct and the probability of the cheese appearing on the right side p. The cheese location is computed by sampling

from the prior distribution over reward location, defined by using p. The cue is then set to either hint at location 1 or location 2, depending on the true location of the cheese and r. The rat's task is to find the cheese; its actions will depend on its prior belief about the environment.

3.1.2 The rat - an active inference agent

For this toy example, the agent needs to perform some actions based on its beliefs about the environment to get to the cheese. It is able to choose from four different actions i. e. go to location 0, 1, 2, or 3. The rat does not have knowledge about the current environmental state. Therefore, it is considering eight states: four possible locations factorized by two contexts. The position does not yield any uncertainty in this example: the agent always has perfect knowledge of its current location.

After each time-step, the agent perceives both its position in the maze and the reward of its actions from the environment. There are two types of potential reward, neutral and positive. The agent receives a positive reward if it gets to the cheese and a neutral reward otherwise.

Generative model

At the beginning, the three prior distributions $p(o|s)$, $p(s_{t+1}|s_t, a_t)$ and $p(o)$ (see Section 2.2.3) are initialized. The agent's prior belief about the cheese's being on the right p, its prior on cue reliability r and the temperature T can be set manually in the present implementation. The prior preferences $p(o)$ are encoded as a vector of probabilities over two possible outcomes: positive reward and neutral reward. The rat wants to get to the cheese and dislikes not getting it: therefore, the probability of getting a positive reward is set to 1. The distributions $p(o|s)$ and $p(s_{t+1}|s_t, a_t)$ are defined as matrices A and B, where $A = p(o|s)$ and $B = p(s_{t+1}|s_t, a_t)$.

- A: The likelihood of getting a positive reward at certain position. It is encoded as vector of shape 4×1 Since they are two outcomes, positive and neutral, this matrix formally needs two columns, one for each outcome to accurately describe the distribution. Since the second column can easily be generated by

subtracting the first column for one, it has been left out in the implementation. It is initialized using the parameter p set before-hand, such as $A = \begin{pmatrix} 0 \\ 1-p \\ p \\ 0 \end{pmatrix}$

- B: For convenience, B is equal to the true transition probability defined in the environment. It is possible to define a transition probability such as the agent is not sure about its position in the maze, as in Schwöbel et al. (2018, p.2550). The goal of this paper is too keep the implementation simple; introducing uncertainty into the transition distribution is an interesting lead for future work. Accordingly, the agent always has full knowledge about its position,

The agent needs an additional temperature parameter to be able to perform active inference. The temperature indicates how much the agent will prefer policies with a low free energy over policies with a higher free energy. In other words, it describes the trade-off between exploration and exploitation; the higher T, the more explorative the agent will act. The temperature is used in the softmax function computing the approximate posterior distribution over policies $q(\pi)$.

The parameter r indicates how correct the cue is on average. Accordingly, the probability of the hint being correct is 0.5 if $r = 0$. To translate r into the actual probability of the hint pointing to the correct needs to be scaled. For example, if $r = 0.5$, then the probability the hint is correct becomes 0.75.

The rat is able to learn new information about the environment by checking which way the cue is hinting at. It needs to update its belief on the cheese's location when visiting the hint, based on its prior beliefs over location and cue reliability.

Belief update after visiting the cue

Let L be the event "the food is in the left arm of the maze" and R the event "the food is in the right arm of the maze". Likewise, let o_l be the observation "The hint is showing the left arm." and o_r "The hint is showing the right arm.". Performing Bayesian updating returns:

$$P(L|l) = \frac{P(l|L)P(L)}{P(l)} = \frac{P(l|L)P(L)}{P(l|L)P(L) + P(l|R)P(R)}$$

$P(R|l)$, $P(L|r)$ and $P(R|r)$ are computed likewise using Bayesian updating. As an example, let the prior probability that the food is on the left side be $P(L) = 0.8$. $P(R)$, the prior probability that the food is on the right side, therefore is $P(R) = 1 - P(L) = 0.2$. $P(r|L)$ is the probability of the hint showing right when the food is on the right side. It is therefore regulated by the reliability of the cue r. Let $r = 0.8$. Accordingly, the cue hints at the correct location with a probability of $P(r|L) = .1$. Then $P(r|R) = 1 - P(r|L) = 0.9$. That mean $P(L|r) \approx 0.17$. The figure below shows how the prior beliefs over the cheese's location, respectively on the left and right side of each image varies over time. The image on the left shows the environment and the agent at $t = 0$, the middle image shows the agent finding the hint and updating its belief at $t = 1$ and the last image shows how the agent behaves afterwards at $t = 2$.

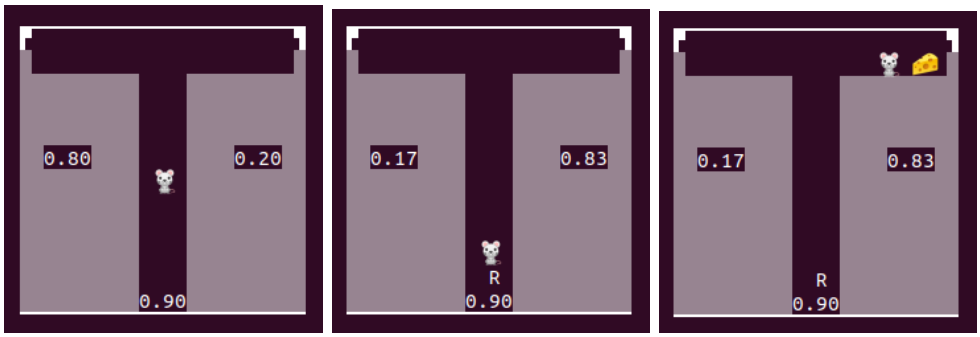

Figure 3.2: Belief updating after visiting the cue at $t = 1$. The agent begins with some prior beliefs about the cheese's location at $t = 0$ (left). The number on the left side of the image specify the prior belief to find the cheese on the left side. The number at the bottom is equal to r. The rat's prior belief on cheese location gets updated at $t = 1$ (middle). It acts accordingly at $t = 2$ (right) and finds the cheese.

3.2 Optimal behavior

Ideally, the rat should choose the optimal policy in order to get the cheese. The optimal behaviour is depending on the true environmental parameters p and r. p is the probability the cheese appears on the right side of the maze; consequently, the cheese appears on the left side of the maze is $1 - p$. r regulates the reliability of the

cue at position. As mentioned in Section 3.1.1, the cue hints at the correct location with a probability of 0.5 if $r = 0$.

Policy with the highest reward on average depending on the environment parameters

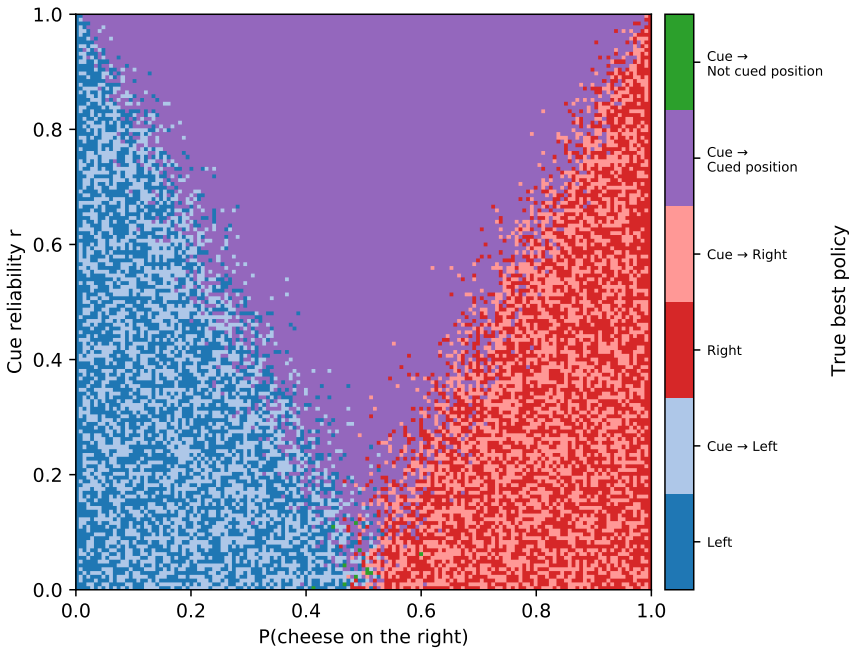

Figure 3.3: Policy with the highest reward on average when performed a 150 times. This figure shows which policy is the most adequate depending the environmental parameters: p is the probability that the cheese appears on the right side of the maze and r is the cue's true reliability.

The figure shows the policy with the highest average reward when performed a 100 times. One of six policies could be the best policy depending on the environmental parameters. Either the rat goes directly to the left or right side without checking the cue (respectively dark blue and dark red), or the rat goes first to the cue but chooses the left or right side without taking the hint into account (respectively light blue and light red) or the rat decides to check out the cue and to follow hint or not (respectively in violet and green). In the following, the individual policies will be elucidated:

- Going directly to the left (or right) arm. This policy is preferred when the cue provides little information because r is close or equal to 0. In this case, if the

cheese is almost certainly on the left (or the right) arm, meaning if p is close to 0 (or to 1), it is best to ignore the cue.

- Checking the cue first and choose the left (or the right) side anyway. This policy is, reward-wise, identical to the policy discussed above. The rat is not incentivize to get the cheese as soon as possible and the reward it gets is in both cases identical. This explains why both policies (the one above and this one) are overlapping in Figure 3.3.

- Following the cue. The agent first checks the cue and goes to the arm indicated by the hint. This policy is the best policy if there is a high uncertainty about the cheese's location, that means when p is close to 0.5. The policy is preferred the higher the hint reliability r, the optimal policy then depends on p. If $r = 1$, that means when the cue is always hinting at the correct location, it is always optimal to first check the hint.

- Checking the cue and then do the opposite. This policy is only seen as optimal when $r \approx 0$ and $p \approx 0.5$. The uncertainty on both the cheese's location and the cue reliability makes the expected reward of all policies discussed here very similar. This policy should not be an optimal policy in any case and is in an artifact due to averaging.

3.3 Active inference

To solve the problem, the rat (i. e. the agent) implemented infer which policy it should follow using *active inference*. The next sections elaborates on how active inference was realized in the program available at . (Explain attributes)

3.3.1 Agent's structure

The rat was divided into three components, having different roles in the planning process: the agent itself, responsible for the interaction with the environment, the long-term memory, where the planning and learning process takes place and the short-term memory which updates the prior beliefs after visiting the cue.

class Agent

The class Agent acts as a link between the long-term memory and the short-term memory. It also the entity which interacts with the environment. It selects and executes the action based on the approximate posterior belief about action $q(a_t)$ provided by the long-term memory. The agent is also the instance which gives the short-term memory the signal to update the prior beliefs on location when it perceives the hint.

class LongTerm

The long-term memory encodes the generative model and the generative process which enables the agent to select the next action. First, it samples the outcomes and the states from the generative model $p(o, s, \pi)$. Based on the results, it computes the expected free energy for every policy. The posterior probability over policies $q(\pi)$ is then computed by the taking the softmax of the negative EFEs. Then, the posterior probability over action $q(a_t)$ is computed as explained in Section 2.4.2.

class ShortTerm

The short-term memory updates the beliefs over the cheese's location using Bayes' rule as explained in Section 3.1.2.

3.3.2 Reducing ambiguity

During the planning process, the agent evaluates the expected free energy of each policy. Since the free energy is computed using an ambiguity term (see Section 2.3), the rat has an incentive to reduce the ambiguity of its future states. The agent does always its position in the maze; the uncertainty of its states only comes from the uncertainty about environmental context. There are two possible contexts in the discussed example; either the cheese is on the left or right side. When the rat goes to location 3 and sees the cue, it updates its belief about the context and the uncertainty about its true state is reduced. Hence, when the rats *plans* to get to the cue, but has not seen it at the current time-step, it must factor in that it will

learn something that it does not know yet. Despite introducing the rat example, Friston et al. (2016) did not elaborate on how the uncertainty is reduced during the planning phase.

In this implementation, the uncertainty about states was reduced by assuming that the cue always hints at the left side of the maze. Therefore, at the beginning of an episode, the agent represents its belief about states by a matrix of shape 2×4. This corresponds to the two possible contexts times the four locations in the maze. The two columns are always identical since the position is independent from the context. When the rat sees the cue, it changes its state representation to a vector of shape 1×4 i. e. either the first column (if the cue shows to the left) or the second column (if the cue shows to the right). During the planning phase, the rat assumes it will see a hint to the left side. Accordingly, when considering future time-steps, it will continue with a 1×4 state representation, thus reducing ambiguity.

3.4 Learning

Additionally, the agent is capable to perform gradient descent to optimize the vector A. The gradient was approximated using mean-field approximation and was taken from Sajid et al. (2019, p.14), see Section 2.5. The long-term memory computes the variational free energy at every time-step for every policy depending on the agent's true reward. Using the definition from Sajid et al. (2019), the variational free energy is solved for the likelihood A. The likelihoods of each policy considered are then averaged using the approximate posterior $q(\pi)$.

3.5 Episode

Finally, it is necessary to combine all steps in a meaningful manner in order to let the agent perform active inference and, if intended, let it optimize the likelihood A. An individual run was based on the pseudo-code provided by Sajid et al. (2019, p.34).

3.5.1 Initialization

First, some key distributions needs to be initialized:

- The agent's temperature T
- The agent's prior belief over the cue's reliability, called r
- The likelihood A
- The state transition matrix B
- The prior preferences $p(o)$
- The agent's prior belief over state at $t = 0$, s_0.

s_0 is set to $\left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{smallmatrix}\right)$ at the beginning of every run, this means the agent believes it starts at location 0, independently from the environmental context. The state transition matrix B corresponds to the true transition distribution and is identical across runs. The prior preferences is a distribution over two possible outcomes: the prior preference to get to the cheese is set to 1, the probability of getting a reward equal to 0 is therefore set to 0.

The likelihood over state outcomes A is defined with the parameter `food_is_right_prior` (called p in the remainder of this Section, an argument passed to the class Agent at its initialization. The likelihood A is equal to $\begin{pmatrix} 0 \\ 1-p \\ p \\ 0 \end{pmatrix}$. Likewise, the agent's prior belief on the cue's reliability is defined by the argument `reliability_prior`. This argument is then scaled linearly such as $r = 0.5$ if `reliability_prior` is equal to 0. Finally, the agent's temperature is a fixed value initialized with the argument `T`.

3.5.2 Run

After the initialization, the agent performs the inference procedure at each time step. The agent first uses its generative process, described in Section 2.4 to sample the expected future states $q(s)$ and the future outcomes from the generative model. The states and outcomes are computed for every policy the agent considers. Based on the expected states and outcomes, the agent then computes the expected free

energy of every policy and infers the probability distribution over the next action, regulated by the temperature parameter. The agent then sample an action from the latter and executes the chosen action. This cycle repeats until the time limit is reached.

The agent performs an additional learning step if needed, after the end of the run. A gradient descent step is performed over the negative variational free energy of each policy. Using the definition of variational free energy of each policy the agent then solves for A to obtain a new likelihood for each policy. Those A vectors are then averaged using the posterior probability on policies as weights.

4 Experiments and results

4.1 Active inference

4.1.1 Agent's behavior

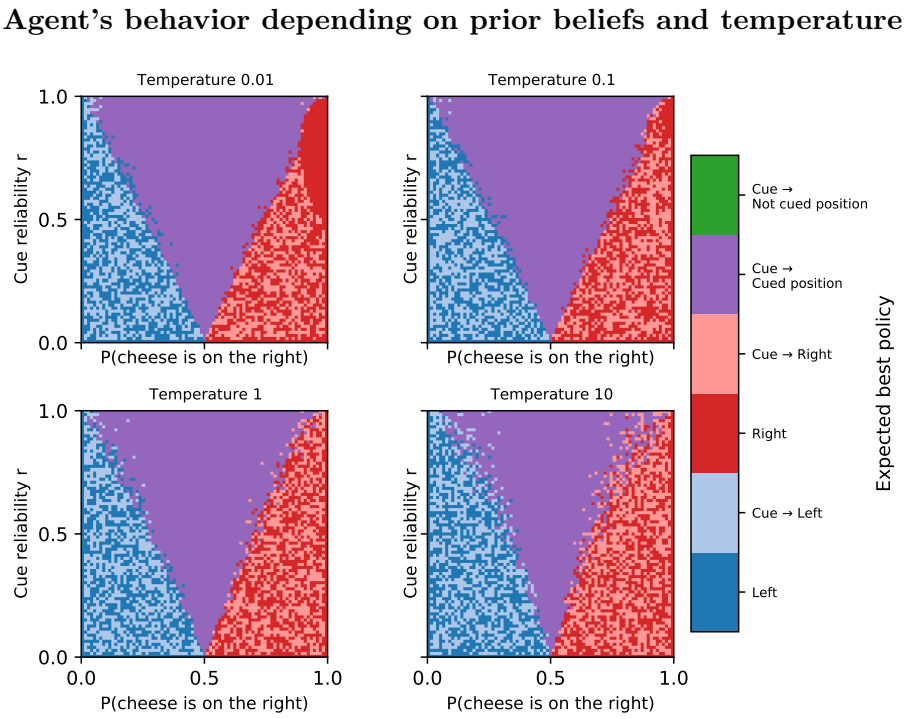

Figure 4.1: Policy chosen by the rat averaged over 80 runs depending on the prior belief on the cheese's location and the cue's reliability and the temperature. The agent's prior beliefs matches with the true environmental parameters in each case.

The policies the agent chooses are closely resembling the optimal policies for the corresponding environmental parameters (see 3.3). The agent's behavior resembles the optimal behavior even more, when the temperature term is higher than one. This means the rat's behavior is better when some exploration is allowed, despite being initialized with the true environmental parameters. It indicates that if the cheese's location is uncertain due to the environmental parameters, some kind of exploration can be beneficial: the policy with lowest free energy yields an expected reward which is similar to the expected reward of the other policies.

4.1.2 Comparison between optimal behavior and the agent's behavior

Average difference between optimal policy's reward and agent's reward

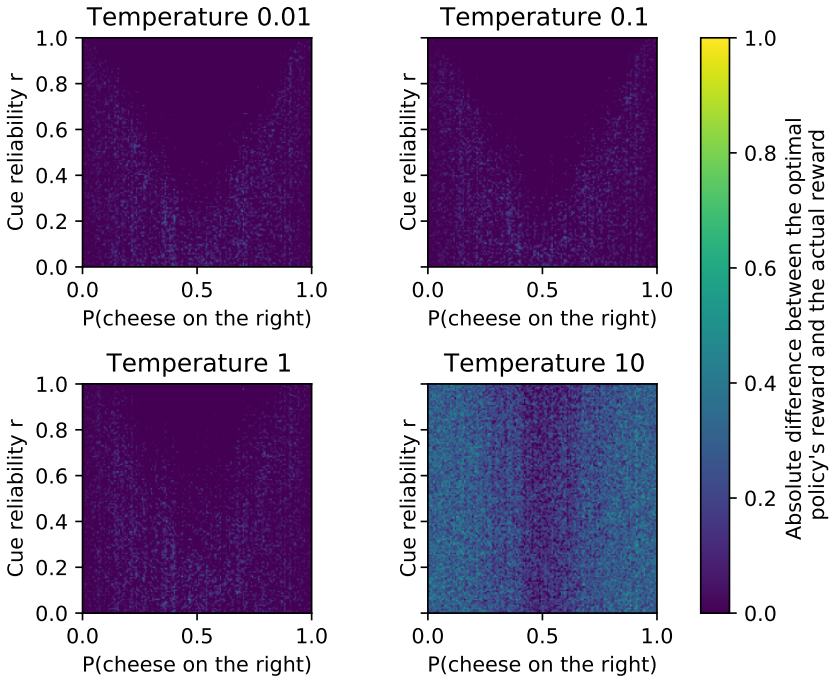

Figure 4.2: Average difference between the optimal policy's reward and the reward the agent actually gets, depending on temperature, the prior belief over the cue's reliability and the prior belief over the cheese's location.

4.1.3 Comparison between agent's expected reward and true reward

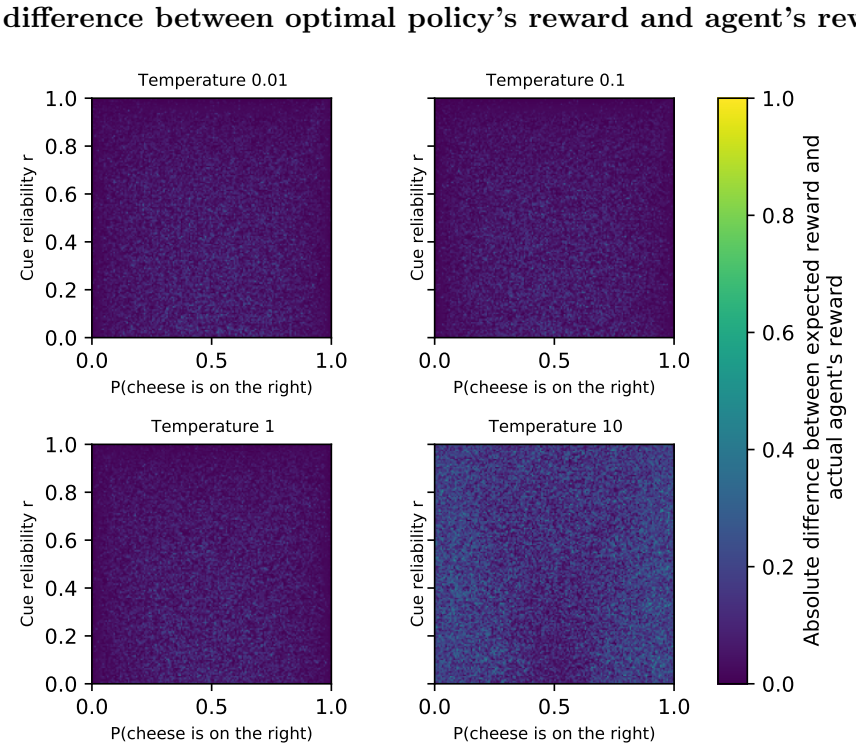

Figure 4.3: Average difference between the optimal policy's reward and the reward the agent actually gets, depending on temperature, the prior belief over the cue's reliability and the prior belief over the cheese's location.

4.2 Learning

Accuracy of pure active inference and a learning active inference agent in two environments

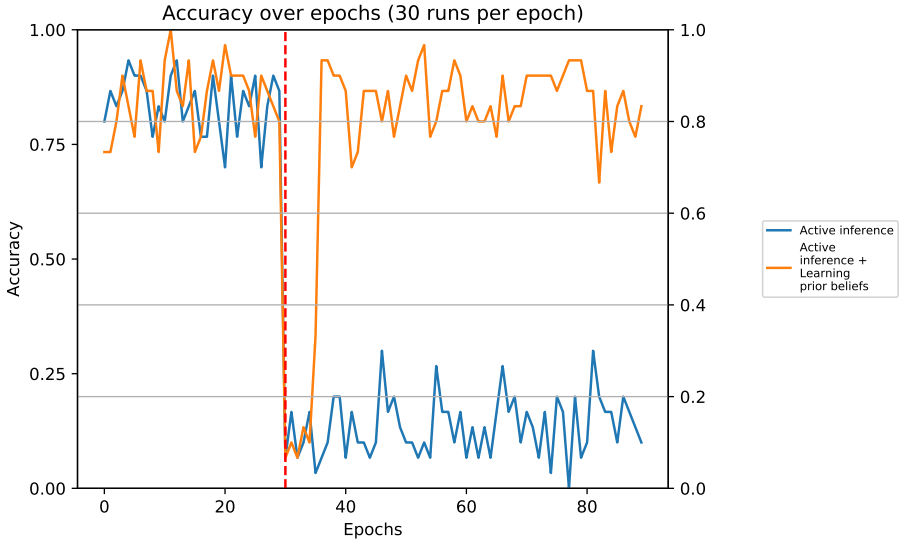

Figure 4.4: Average accuracy over time. The first environment was initialised with a cue reliability of 0.1 and the cheese as a probability of 0.1 to appear on the right side of the maze. The second environment is initialized with a cue reliability of 1 and the cheese has a probability of 0.9 to appear to the left. Both agents were initialized with a prior over cue reliability $r = 0.9$ and a prior belief that the cheese is right $p = 0.1$. The temperature of both agents is set to $T = 4$.

In this example, the agents were positioned in two environments, one after the other. The first environment matches approximately with the prior beliefs: both agents have a high accuracy and stable behavior. When the agents are now put into the new environment, they display two different behaviors. The agent performing only active inference has a low accuracy, since the environment does not corresponds to its prior beliefs anymore. The agent which additionally learns its behavior first performs badly at the beginning and then performs to a level comparable to before the environmental shift.

Accuracy of pure active inference and a learning active inference agent

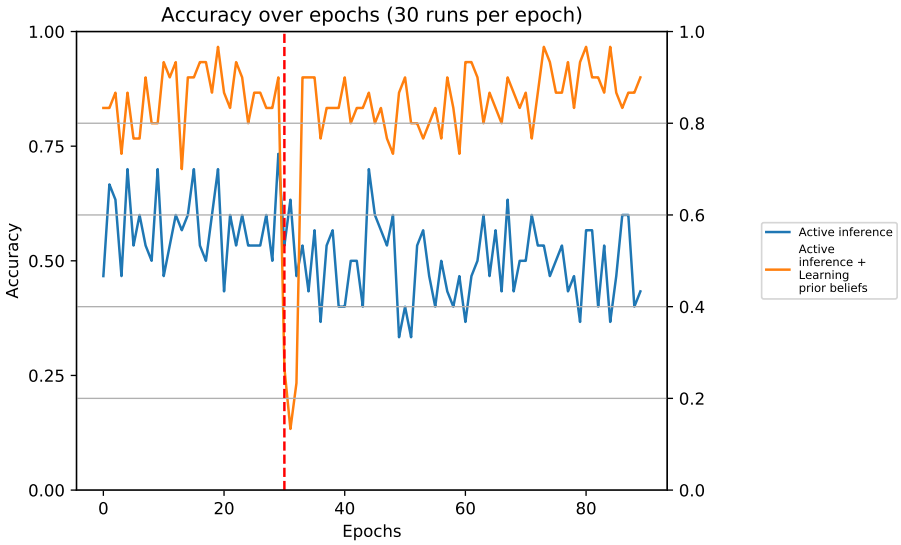

Figure 4.5: Average accuracy over time in two distinct environments. The first environment was initialised with a cue reliability of 0.1 and the cheese as a probability of 0.1 to appear on the right side of the maze. The second environment is initialized with a cue reliability of 1 and the cheese has a probability of 0.9 to appear to the left. Both agents were initialized with a prior over cue reliability $r = 0.9$ and a prior belief that the cheese is right $p = 0.6$. The temperature of both agents is set to $T = 4$.

The agents are now both initialized with a prior belief over cheese location that is close to 0.5. This means the agent have a high uncertainty about the cheese's location. In turn, the agent performing pure active inference shows an approximately identical accuracy in the two consecutive environments. The learning agent on the other hand, recovers quickly from the environmental shifts and attains overall a high accuracy.

Accuracy of pure active inference and a learning active inference agent

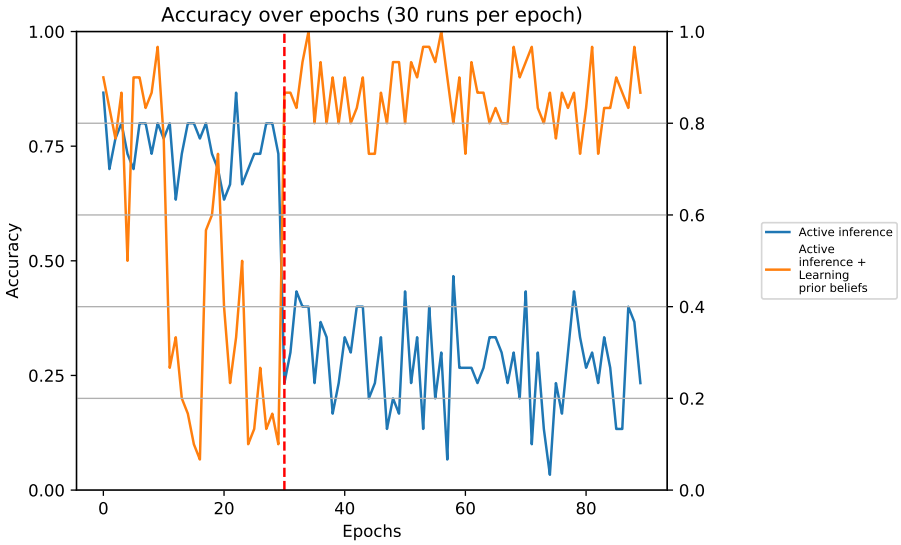

Figure 4.6: Average accuracy over time in two distinct environments. The first environment was initialised with a cue reliability of 0.1 and the cheese as a probability of 0.1 to appear on the right side of the maze. The second environment is initialized with a cue reliability of 1 and the cheese has a probability of 0.9 to appear to the left. Both agents were initialized with a prior over cue reliability $r = 0.6$ and a prior belief that the cheese is right $p = 0.75$. The temperature of both agents is set to $T = 4$.

Accuracy of pure active inference and a learning active inference agent

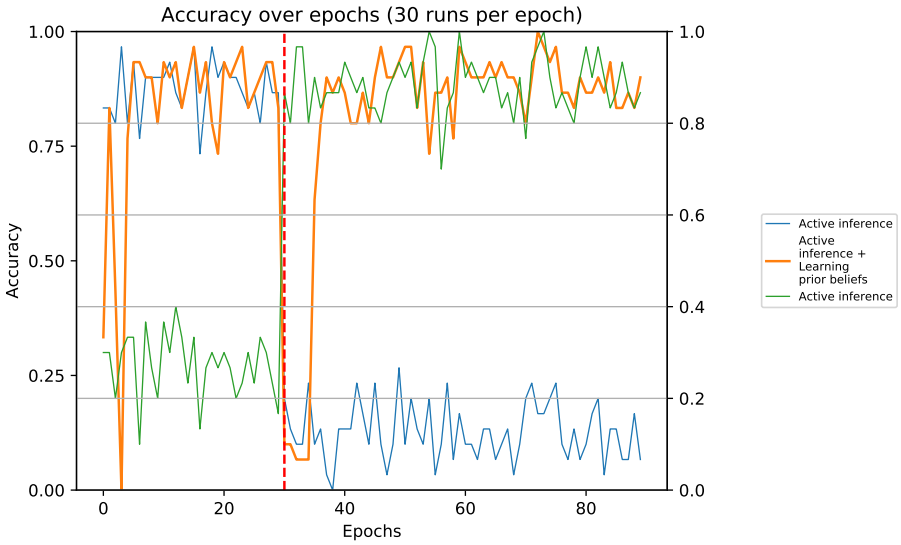

Figure 4.7: Average accuracy over time in two distinct environments. The first environment was initialised with a cue reliability of 0.1 and the cheese as a probability of 0.1 to appear on the right side of the maze. The second environment is initialized with a cue reliability of 1 and the cheese has a probability of 0.9 to appear to the left. In this case, the first agent performing active inference (blue) was initialized with prior beliefs over cue reliability $r = 0.3$ and that the the cheese is right $p = 0.9$. The second agent performing active inference (green) was initialized with prior beliefs over cue reliability $r = 0.9$ and that the the cheese is right $p = 0.2$. The learning agent was set to same parameters as the second agent (green). The temperature of the three agents is set to $T = 4$.

5 Discussion of the implementation

In this chapter, we will discuss how the implementation was designed based on the available literature, especially Friston et al. (2016) and Sajid et al. (2019). It will focus on design choices different from the previously mentioned papers and which consequences these choices had. To begin with, this paper will first dive into the difficulties the author had while implementing the toy example described in Chapter 3, followed by a review of possible developments of the attached implementation. Finally, some remarks concerning the literature about the free energy principle will be made.

5.1 Design choices

5.1.1 Active Inference

Number of different outcomes

Friston et al. (2016, p.871) designed an environment with three possible outcomes: a positive and neutral outcome as explained in Section 3.1.1 and an additional negative outcome when the rat gets trapped into the arm without getting to the piece of cheese. The choice of the numbers of outcomes can easily be tweaked by varying the shape of $p(o)$ and the likelihood A (i. e. $p(o|s)$). $p(o)$ would be replaced by a distribution over three outcomes and A would be replaced by a matrix of shape 4×3. The author chose to keep two outcomes for two reasons.

First, for better readability; having only two outcomes made it possible to define A as the probability distribution over positive outcomes. The probability distribution over neutral outcomes is then simply $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - A$ and is implicitly encoded in A.

Secondly, Sajid et al. (2019, Table 2) shows that the average score of the active inference agent (FrozenLake environment) is similar when the environment only returns a positive reward or both positive and negative rewards. It could be an interesting development to experiment on the presented toy example with both negative and positive rewards. Additionally, one could compare the rat's behavior if the rewards are weighted differently. As an example, the rat could receive electroshock as a punishment for entering the wrong arm: the primary incentive hypothetically should shift from finding the cheese (e. g. reward is equal to 1) to avoiding the electroshock (e. g. the reward is equal to -10), which might lead the rat to stay at the center or the bottom of the maze more often.

Incentivization of speed

It could have been possible to implement a mechanism motivating the rat to get the cheese as fast as possible. This would correspond to a more realistic behavior especially if the agent has a prior belief about the cheese's location with very low uncertainty. The policy going directly to the left or the right would always yield a higher reward than first checking the cue and entering the left or right side anyway; therefore, the latter would not appear on Figure 3.3.

This mechanism was not implemented again for better readability. The environmental A and could have been designed to be time-dependent. It would suffice to add a further dimension, the dimension of time, to A. The true reward of getting to the cheese could then easily be decreasing over time. Since the likelihood $p(o|s)$ is a probability distribution over getting *any* positive reward, it would remain the same over time even if the true reward the agent can achieve decreases over time. It would be necessary to introduce different kinds of rewards into the generative model, as described in the paragraph above. $p(o)$ would then encode preference over positive rewards of different magnitudes and a neutral outcome. This was not done here for the reasons mentioned in the preceding subsection.

Furthermore, as described in the following section, the author had difficulties implementing the reduction of uncertainty into action 3 (i. e. "Go to cue."). Thus, increasing the uncertainty of A by introducing different kinds of outcomes would have been a counter-productive step during the implementation process.

Reduction of uncertainty

During the evaluation of policies, it has to be taken into account that the rat gains information when checking the cue. According to Friston et al. (2016, p.872), the cue at the bottom of the maze provides information about the context. This ensures the rat represents its future states with less ambiguity than before getting to the cue. But since the direction hinted at by the cue is unknown and the states s_t are defined over both position and context, it is not possible to infer in which state the rat will be after getting to the cue. How the expected states are represented after getting to the cue is not described in Friston et al. (2016) to the knowledge of the author of this paper.

Therefore, a workaround had to be implemented. In this case, the agent always assumes the cue is hinting at the upper left-hand arm of the maze. This works because of two reasons. First, by virtue of the agent always knowing its position accurately. This would not work if the maze was filled with artificial fog in such a way the rat does not know exactly if it is at location 0 or 1, but sees clearly if it is at location 2, for example. Secondly, reducing uncertainty this way works on behalf of the symmetry of the cue's information. In this environment, the cue is equally reliable if it shows to the left or the right, however one could design an environment where the cue is biased towards a certain side. In both cases, the ambiguity of future states would be dependent on the actual information about the side displayed by the cue. How context information should be integrated into the planning phase in a more general manner requires further research. This difficulty also derives from the definition of context used in Friston et al. (2016), elucidated in the next subsection.

Definition of context

In the presented implementation, the context is defined by the location of the cheese. The cue, therefore, gives information about the context but is not a perfect indicator. The cue is not always reliable enough to give perfect information about context; additionally, the posterior belief on context (i. e. on the cheese position) after visiting the cue depends on the prior belief on context and is not perfect. The example presented by Friston et al. (2016, p.872) conflates two definitions of the possible contexts in the rat example. The context is both defined by the cheese's location and the information given by the cue, e. g. context 1 is defined as both "cheese is

left" and "cue shows to the left". This conflation results from the cue's reliability being fixed to 1; the cue is always showing the correct side. In turn, being in context 1 provides a reward at location 1 with probability $p = 98\%$.

The author of this paper did not fully comprehend the function and usage of the environmental contexts in the Friston et al. (2016, p.871) paper up until far into the implementation; it would have been necessary to make substantial adjustments to match Friston's definition of context. Therefore, the cue was not used as a definite indicator of the context and but instead used the cue to update the prior belief on context. The belief over context is encoded in the agent's likelihood A as the probability of getting a positive reward in context 1. The likelihood of getting a positive reward is encoded implicitly as stated in Section 5.1.1. This choice ensured better readability of likelihood A and was very useful during the research process leading to the understanding of the active inference framework. Since the outcomes and the contexts are easily mapped onto each other, it was possible to integrate the prior belief over context in likelihood A. It would be interesting to decouple the prior belief over context and reward for more complex environments in future work. By merging those two probability distributions, the generative model presented in Friston et al. (2016, p.871) had to be modified.

Generative model adjustments

In the original paper, Friston et al. (2016, p.871) incorporated the belief about the context into the generative model of the agent independently of the prior belief of the cheese location. The shape of the matrices and vectors encoding the agent's beliefs over hidden states s, the likelihood A, the transition distribution B, and the prior preferences are therefore varying from the ones used in the attached implementation. The following changes have been performed:

- *State representation.* The beliefs about hidden states s_t are encoded in an one-dimensional vector with 8 entries in Friston et al. (2016, p.872). This corresponds to four locations in the maze factorized by two contexts. Every two entries next to each other correspond to a single location within the maze; the context is then alternating between indices. All even indices correspond to context 1, all odd numbers correspond to context 2. In the attached implementation, the states are represented by a matrix of shape 2×4. When

the context becomes known through the cue, the representation shrinks to a vector of shape 1×4.

- *Prior preferences* $p(o)$. There are 3 types of possible outcomes in the original paper. The prior distribution over outcome preferences is encoded as a vector with 7 elements. This comes from the fact they are 8 possible states but the outcome at location 0 is independent of context. Location 1, 2, and 3 are dependent on context because they yield contextual information in this environment, meaning there are 7 different possible outcomes. In the attached implementation, the prior distributions are independent of location and context; it simply encodes which of two outcomes, either positive or neutral, is preferred. Therefore, it is encoded by a one-dimensional vector of size 2.

- *Likelihood* A. As previously explained, A is a vector with four elements in the attached implementation. In Friston et al. (2016, p.872) A is matrix of shape 7×8. A maps a state to a posterior distribution over expected outcomes, which explains its shape.

- *State transition distribution* B. The transition distribution is defined as a $4 \times 4 \times 4$ matrix which is nearly identical to the the one presented in Section 2.2.3. In the attached implementation, each row in corresponds to the next state s_{t+1}. In the original paper,Friston et al. (2016) defined each column as state s_{t+1}. However, the transition matrix for a single action is multiplied with an identity matrix of shape 2×2 by using a Kronecker tensor product. This is necessary to obtain a state with the characteristics described above, encoding both context and position.

5.1.2 Learning

The decision was made to only learn and optimize the likelihood A and to leave B, the temperature T and $p(o)$ out of the learning process. This decision was made due to different reasons for each of the mentioned distributions.

First, the learning over $p(o)$, as described in Sajid et al. (2019, p.26) is vastly different from the other three distributions. It needs an additional set of priors (a prior distribution over prior preferences) learned via the accumulation of Dirichlet parameters. Secondly, The temperature T used in the calculation of the approximate

posterior distributions over policies is as well optimized through gradient descent, except the gradient is different. Thirdly, in this example, the agent has perfect knowledge about the transition distribution B and thus does not need to learn it. It is possible to implement an agent which does not have perfect knowledge about its position and in that case, it would be appropriate to optimize B through gradient descent. The gradient used is the same as the gradient used to optimize A.

The focus of this paper is to provide a simple agent which performs both active inference and learning; for a proof of concept, it sufficient that the agent optimizes only one aspect of the generative model, namely, the likelihood $p(o|s)$.

The main problem in the current implementation is the incapacity of the agent to optimize its prior belief over hint reliability. If the prior belief and the true reliability differ vastly from one to the other, the agent will not be able to follow the true optimal policy. Optimizing the reliability parameter is an interesting lead for future work on this agent.

5.2 Difficulties during the implementation process

The main difficulty of implementing the agent described in Chapter 3 came from the misconceptions about the environmental contexts. The implementation process was therefore hindered, since the contexts affect the generative model itself. Although it influenced the implementation process negatively, it drove the author to find an alternate solution to reduce the ambiguity during the planning process.

Furthermore, the resulting generative model is constituted from relatively small matrices: $p(o)$ is a vector with two elements, A is a vector with four elements and B is a matrix of shape $4 \times 4 \times 4$. This supposedly could help future readers to understand the function and usage of each distribution more easily.

Moreover, during the implementation of the optimization of A, the author first relied on the pseudo-code provided by Sajid et al. (2019, p.34). In the provided pseudo-code, the gradient descent over variational free energy was performed during the agent was still performing actions in the environment. The learning updates through accumulation of parameters happens at the end of the pseudo-code, outside the training-loop. The author of this paper assumed the accumulation of parameters

was referring to the learning process of $p(o)$, since the accumulation of Dirichlet parameters is largely described in Sajid et al. (2019, p.27).

The present implementation could therefore be improved by calculate the gradient inside the training loop and compute the updated likelihood A after the end of the run.

5.3 Literature about FEP

Despite having this framework presenting itself as fundamental, the scholarly reception remains rather slow. Since the readership is naturally composed of scientists and students who enjoyed very diverse education, it is necessary to lay out the fundamental concepts before introducing the free energy framework. Concepts like *belief*, *surprise* which a very precise meaning in information theory obscure comprehension to non-experts as described by Andrews (2018, p.12).

It is rather arduous to understand the literature coming from the field due to a lack of description of the ideal or intended audience. The necessary knowledge to understand papers is not previously settled. Furthermore, the software written to make experiments and design active inference agents is sometimes not available, for example, as in Friston et al. (2016). The lack of software in scientific publications has lapsed since the awareness about the reproducibility crisis described by Hutson (2018) has risen.

An additional problem in the literature is the usage of technical terms coming from the common physics literature. There are some confusions and misuses in the vocabulary which obscures the meaning of the presented concepts. For example, Sajid et al. (2019, p.14) defines β as the inverse precision, equal to the temperature when the classical thermodynamics literature calls β the inverse temperature, which means the precision (e. g. in Bertola and Cafaro, 2015). This is especially the case in the earlier literature about the free energy principle: e. g. Friston et al. (2006, p.72) describes biological agents as avoidant of "phase-transitions" to endure over time. This term is usually not appropriate for organisms, and one can argue that preventing phase-transitions does not explain why biological agents endure over time, contrary to what Friston implies in said paper.

5.4 Outlook

The FEP can be used to implement a simple minimal agent in a small environment as shown in this paper. It is possible to make more complex agents also relying on free energy minimization to behave optimally in their environment. There are multiple ways to make more complex agents to evaluate, which are all promising approaches for future work on free energy agents.

For example, the present agent could be implemented using a partially observable Markov decision process (POMDP) as did Sajid et al. (2019) using the Matlab SPM toolbox designed by Karl Friston in 1994. Additionally, the usage of messages as defined by Schwöbel et al. (2018, p.2542) as unification of action and perception could lead to an elegant description of the agent's decision processes.

Free energy agents need perceptions from the environment and have the possibility to act on it. The environment of the agent can be very flexible: it is possible to design hierarchical artificial models by defining one agent as the environment of the other. This is further discussed by Friston (2008). It is also possible to define an environment with multiple agents.

Further complexity can be achieved by using continuous state-spaces and learning methods that are different from a simple gradient descent using a mean-field approximation. Overall, the applications domains of artificial agents using FEP still need further groundwork. The results from future research about the free energy principle will decide over the utility and use cases of the framework in agent-based modeling.

6 Conclusion

This paper first introduced how active inference and learning works in a free energy principle framework. It first establishes the FEP agent and its environment by defining its generative model and then clarifies how the agent performs active inference by minimizing the expected free energy of its next action. The agent learns through gradient descent performed over the negative variational free energy, thus optimizing the generative model and permitting the agent to behave more efficiently. Secondly, it spells out the design of the toy example, the rat in the maze: the environmental set-up and the agent's generative model are elucidated. It also specifies the functioning of the agent during a run through the maze. Then the paper compares the agent's behavior to the optimal behavior and analyses the learning process of the agent. Finally, the paper goes deeper into the design choices made during the agent's implementation, the difficulties faced, and a short overview on the FEP literature. It ends with an outlook on potential improvements and extensions of the presented agent.

In summary, it was possible to narrow down the essential elements needed for designing a free energy agent: the generative model decomposed in likelihood, transition probability and prior preferences. It is also necessary to set a temperature and the prior belief over the initial state to be able to model an artificial agent capable of performing active inference and optimizing its behavior. The behavior of the rat resembles the optimal behavior depending on its temperature.

The agent can be extended in a numbers of ways: introduce uncertainty about position, introduce several types of rewards, define a multi-agent environment or a hierarchical model of the agent, etc. The free energy principle showed that is flexible enough to encompass a diverse set of artificial agents and will probably gain more traction in the machine-learning community in the next years if it shows its strengths even in more complex environments.

List of Figures

3.1 Empty environment with numbered positions. 15
3.2 Belief updating after visiting the cue at $t = 1$. 19
3.3 Policy with the highest reward on average when performed a 150 times. 20

4.1 Policy chosen by the rat averaged over 80 runs depending on the prior belief on the cheese's location and the cue's reliability and the temperature. 26
4.2 Average difference between the optimal policy's reward and the reward the agent actually gets, depending on temperature, the prior belief over the cue's reliability and the prior belief over the cheese's location. 27
4.3 Average difference between the optimal policy's reward and the reward the agent actually gets, depending on temperature, the prior belief over the cue's reliability and the prior belief over the cheese's location. 28
4.4 Average accuracy over time. 29
4.5 Average accuracy over time in two distinct environments. 30
4.6 Average accuracy over time in two distinct environments. 31
4.7 Average accuracy over time in two distinct environments. 32

Glossary

EFE expected free energy

FEP free energy principle

KL Kullback-Leibler divergence

MDP Markov decision process

POMDP partially observable Markov decision process

RL reinforcement learning

VFE variational free energy

7 Bibliography

Adams, R. A., Friston, K. J., & Bastos, A. M. (2015). Active inference, predictive coding and cortical architecture. *Recent advances on the modular organization of the cortex*. https://doi.org/10.1007/978-94-017-9900-3_7

Andrews, M. (2018). The Free Energy Principle: An Accessible Introduction to its Derivations, Applications, & Implications. (May), 1–16.

Andrews, M. (2020). The Math is not the Territory: Navigating the Free Energy Principle.

Bertola, V., & Cafaro, E. (2015). Response of a thermodynamic system subject to stochastic thermal perturbations. *Physics Letters, Section A: General, Atomic and Solid State Physics*, *379*(47-48), 3035–3036. https://doi.org/10.1016/j.physleta.2015.10.026

Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, *76*, 198–211. https://doi.org/10.1016/j.jmp.2015.11.003

Fletcher, P. C., & Frith, C. D. (2009). Perceiving is believing: A Bayesian approach to explaining the positive symptoms of schizophrenia. *Nature Reviews Neuroscience*. https://doi.org/10.1038/nrn2536

Friston, K. (2008). Hierarchical models in the brain. *PLoS Computational Biology*, *4*(11). https://doi.org/10.1371/journal.pcbi.1000211

Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, *11*(2), 127–138. https://doi.org/10.1038/nrn2787

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., O'Doherty, J., & Pezzulo, G. (2016). Active inference and learning. *Neuroscience and Biobehavioral Reviews*, *68*, 862–879. https://doi.org/10.1016/j.neubiorev.2016.06.022

Friston, K., Kilner, J., & Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology Paris*, *100*(1-3), 70–87. https://doi.org/10.1016/j.jphysparis.2006.10.001

Friston, K., Parr, T., Zeidman, P., Razi, A., Flandin, G., Daunizeau, J., Hulme, J., Billig, A. J., Litvak, V., Moran, R. J., & Price, C. J. (2020). *COVID-19* (tech. rep.).

Friston, K., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., & Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, *6*(4), 187–214. https://doi.org/10.1080/17588928.2015.1020053

Friston, K., Thornton, C., & Clark, A. (2012). Free-energy minimization and the dark-room problem. *Frontiers in Psychology*, *3*(MAY), 1–7. https://doi.org/10.3389/fpsyg.2012.00130

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*. https://doi.org/10.1126/science.359.6377.725

Levin, M., Friston, K., Sengupta, B., & Pezzulo, G. (2015). Knowing one's place: a free-energy approach to pattern regulation. https://doi.org/10.1098/rsif.2014.1383

McGregor, S., Baltieri, M., & Buckley, C. L. (2015). A Minimal Active Inference Agent, 1–19.

Ramstead, M. J. D., Badcock, P. B., & Friston, K. J. (2018). Answering Schrödinger's question: A free-energy formulation. *Physics of Life Reviews*, *24*, 1–16. https://doi.org/10.1016/j.plrev.2017.09.001

Sajid, N., Ball, P. J., Parr, T., & Friston, K. J. (2019). Active inference: demystified and compared. *44*(0), 1–69.

Schrödinger, E. (1945). What is Life? The Physical Aspect of the Living Cell. *The American Naturalist*. https://doi.org/10.1086/281292

Schwöbel, S., Kiebel, S., & Marković, D. (2018). Active inference, belief propagation, and the Bethe approximation. https://doi.org/10.1162/neco_a_01108

Ueltzhöffer, K. (2018). Deep active inference. *Biological Cybernetics*. https://doi.org/10.1007/s00422-018-0785-7