Active Inference as an Alternative to Reinforcement Learning

Parth Mihir Patel
psxpp5@nottingham.ac.uk
School of Computer Science, The University of Nottingham

Introduction

Hierarchical Planning is crucial for problem solving in many real-world tasks. It requires the ability to infer future states of the world efficiently, and choose the actions accordingly. Current AI paradigms are not very good at this. Reinforcement Learning requires too many trials and errors to learn the optimal policy. It gets especially inefficient when the state space is only partially observable. Autoregressive Deep Learning models require massive amount of data to train, and still can't meta-learn. Moreover, they lack any notion of 'common sense', i.e. a coherent model of how the world works. Despite being a Universal Function Approximator, the neural network architecture is most likely not Turing complete.

Researchers in the Cognitive Science community have argued, and empirically demonstrated to a great extent [4][11], that Human Perception is fundamentally not representationalist in nature, and that cognition is more than just parsing of sensory data. Cognition is inherently Embodied, i.e. the result of the way an organism fits in its niche and the way it acts in the environment [10].

Philosophers ranging from Emmanuel Kant to Martin Heidegger advocated for this concept for centuries, but their arguments were (rightly) not accepted by the Scientific community on the basis of lack of empirical evidence and lack of falsifiability (which is a crucial Popperian criteria for Science). However, the post-war 3rd generation Cognitive Science, starting with the works of James Gibson, started hypothesizing and demonstrating Embodied Cognition. This culminated in development of the Active Inference theory, by Karl Friston in 2005 [12], which provides rigorous evidence for the brain being an autopoietic Bayesian inferencer. Cognition according to Active Inference is a mechanism used by self-organizing systems to preserve their homeostasis by avoiding surprising observations. Surprise can be minimized not just by updating the internal world model, but also by updating the world itself, i.e. by taking action. Hence the preferred sequence of actions would be the ones which minimize the integral of expected variational free energy (which is an approximation of surprise) over time. This closely mirrors Lagrangian mechanics, in which physical systems take the path which minimizes the Action (Kinetic Energy minus Potential Energy) integral.

In this study, I explore the theoretical underpinning of Active Inference, and implement it for a gird based scenario, in which Jerry, trapped in a Partially Observable Markov Decision Process, tries to figure out where the cheese is located, while trying to avoid Tom(s). Jerry doesn't directly observe where Cheese or Toms are located. The purpose of experiment is to demonstrate practical applicability of Active Inference, as an alternative to Reinforcement Learning.

Literature Review

Frame Problem, Relevance Realisation, and relevant Philosophies

As per Hoare triple, a Problem can be formalised as: Precondition -> Action -> Postcondition Newell and Simon (1972) formalized the concept of Problem Solving as searching through the state space [1]. Searching through all the states to bring the environment closer to postcondition, is combinatorically explosive [2], as the search space increases exponentially with respect to number of steps and number of operators. The same problem occurs for categorization as well. Forming categories is predicated upon finding common features or uncommon features. But there are technically infinite commonalities and discommanilities between objects. This is called the frame problem, and it was first formalized by Daneil Dannet (1984) [3]. Hence to solve problems or to form categories it is essential to zero-in on 'relevant' features. Vervaeke et al. coined the term 'relevance realisation' for this, and argued that it is the essence of human perception [4]. Ho et al. presented empirical evidence for this in their research paper "People construct simplified mental representations to plan" [5].

Over 250 years ago, Philosopher Emmauel Kant argued that our perception isn't based on sensory inputs, but rather on a-priori structure in our mind 'instigated' by sensory inputs [6]. German Philosopher Martin Heidegger's work on phenomenology [7], introduced the concept of "Dasein" to describe human existence, emphasizing that the existence is fundamentally "In-der-Welt-sein" ("beingin-the-world"). This concept challenged traditional Cartesian mind-body dualism by proposing that cognition emerges not from an isolated mind, but from the intertwined relationship between an individual and their environment.

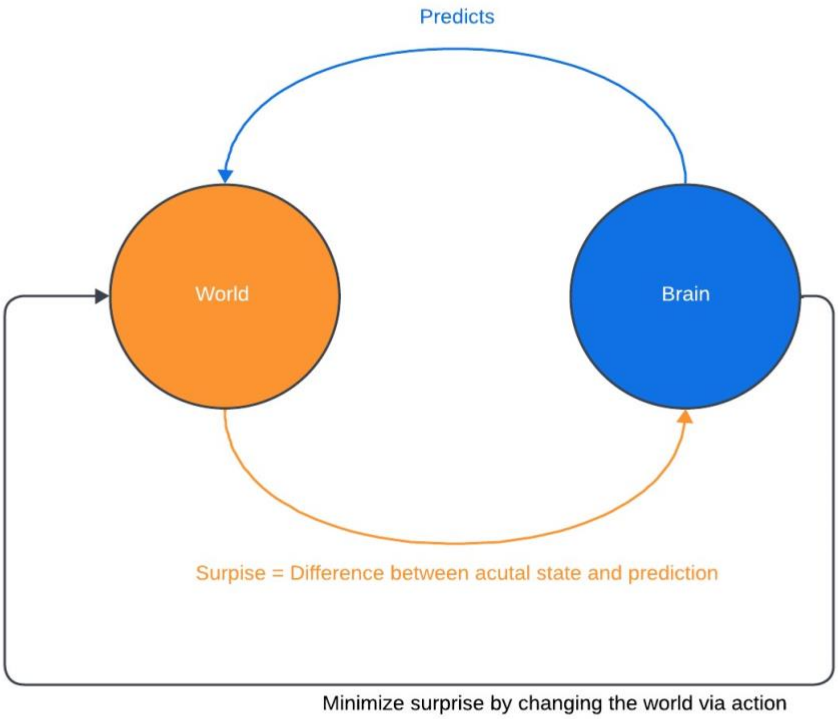
Embodied Cognition, Autopoiesis, and Dynamic Systems

Dynamic Systems Theory, developed by Thelen and Smith (1994) [8], explains how complex interactions between the components of a system can give rise to emergent properties, i.e. new attributes or behaviours that can't arise solely from the properties of the individual components. Closely related to this is the concept of autopoiesis, introduced by Maturana and Varela (1980) [9], which characterizes living systems as self-producing and self-sustaining entities maintaining their structure through ongoing self-renewal and environmental interaction. Autopoiesis has significantly influenced 3rd generation cognitive science, a paradigm shift that shows cognition arises not just from computational processes within the brain but through the dynamic and reciprocal interactions between the brain, the body, and the environment. It emphasizes 4 E's - the embodied, embedded, enactive, and extended nature of cognition, suggesting that cognitive processes are deeply integrated with the organism's physical form and its situational context. Cognition emerges through continuous loops of action and perception, where organisms act upon their environment and perceive the outcomes of their actions, leading to adjustments in future behaviors. James Gibson (1979) [10] laid foundation for theory of affordances. It suggests that our mental models for objects are predicated upon how we interact with them. Perception is closely linked to action. For example, the way we perceive a chair isn't as a set of some structures and designs, but as something we can sit on. Rodney Brooks (1991) [11] highlighted the limitations of purely representationalist AI models, and advocated for more embodied approach to intelligence.

Active Inference

Karl Friston (2005) [12] demonstrated that traditional representationalist model of brain is empirically invalid. Instead of predicting percepts by feed-forwarding sensory inputs, the brain instead does the reverse. It tries to predict sensory inputs from its world model. This breakthrough observation led to the formulation of the theory of Active Inference [10]. It provides a rigorous neuro-scientific basis for Embodied Cognition. It also aligns with the Bayesian method of Inference.

The universe can be conceptualized of as a continuous state space (a continuous Partially Observable Markov Decision Process to be precise). Self-organizing systems form a subset of this state space. They possess internal states separated from external states (environment) by the Markov Blanket [13]. Entropy is the natural condition of the world. Self-organizing systems resist dissipation (entropy-increase), by constantly adapting and updating their internal states. Cognition arises as a means to fulfill this self-organizing process. Organisms minimize prediction errors (Bayesian Surprise) between expected and actual sensory inputs. This minimization could occur both by updating the internal states, as well as by changing the external states (action).


Predicts
World
Brain
Surpise = Difference between acutal state and prediction
Minimize surprise by changing the world via action

(Figure 1: Action Perception Loop)

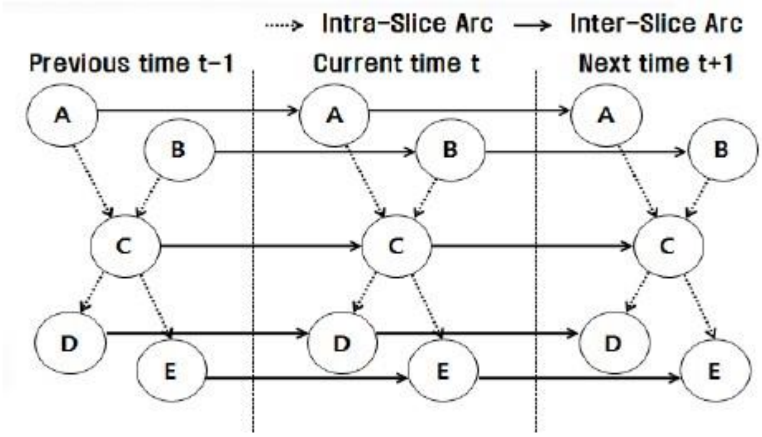
Theoretical Underpinning of Active Inference

To understand and implement Active Inference, we first need to go through some concepts in Graphical Models and Bayesian theory. There are various ways to model the state space. If the state transitions are deterministically determined by just the actions of agent, state space can be represented as a directed Graph. Problem solving on directed graph involves using search algorithms like A* or Dijsktra. If the state transitions are not just dependent on actions but also have a stochastic nature, then Markov Decision Process is used for modelling the state space. Each action has a corresponding probability distribution of outcome: $P(s' | s,a)$, i.e. the probability of reaching state s′, given current state s and action a. Simple search algorithms can't be used for problem solving in MDP, as we need to find policies (i.e. a series of stochastic actions that would bring about the desired state over time), rather than path. Optimal policy is the one which maximizes reward. For this, we need to solve Bellman equations using techniques such as Value iteration or Policy iteration.

In MDP, all states are fully observable. But that's usually not the case in real life. For modelling partial observability, Partially Observable MDP is used. In POMDP, agent doesn't fully know which state it currently exists in. Instead, only a probability distribution over multiple states is known. This is called the belief state. Even if the state space is finite, the belief state space is infinite. Hence finding optimal policies in POMDP is very challenging.

POMDP as a Dynamic Bayesian Network

When we think of state space as a grid, we lose the essence of what a state actually is. A state is a collection of random variables, or a random vector. This collection of random variables forms a Bayesian network. Under partially observability, we have a latent vector which is unknown, and an observation vector which is known. For modelling spaces over time, Dynamic Bayesian Networks could be used.

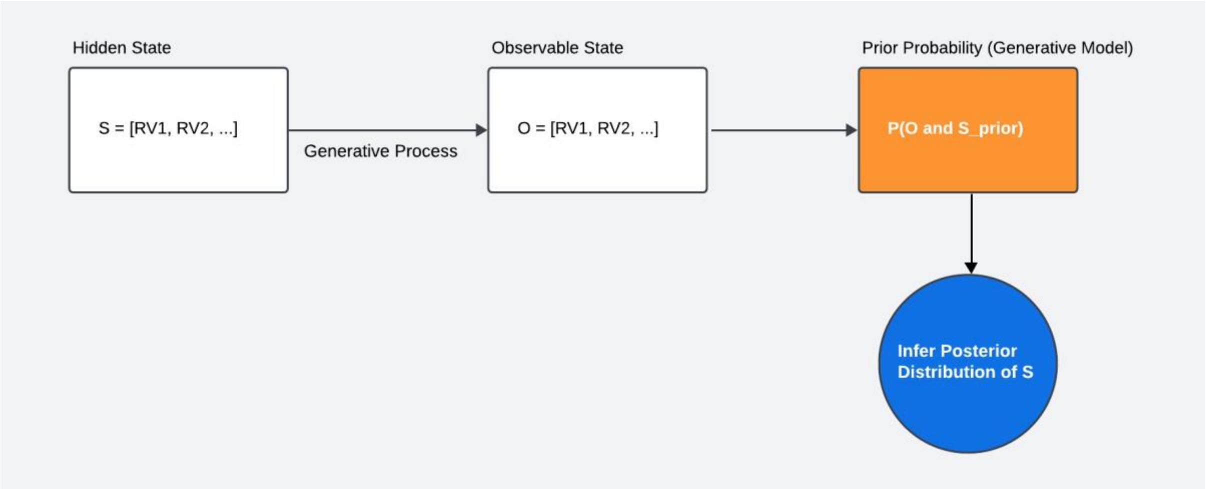


(Figure 2, DBNs, source: [14])

Bayesian Inference

Latent vector can be estimated from observable vector, using Bayesian inference. We assume a prior distribution P(S) of latent vector. The goal is to find out a posterior distribution of hidden state given the observations O. We can use Bayes Rule for this.

$P(S_{posterior} \mid O) = P(O \mid S_{prior}) * P(S_{prior}) / P(O)$
i.e. (Likelihood * Prior Probability) / Marginal Probability, or
$P(O, S_{prior}) / P(O)$
i.e. (Joint Probability of Observation and State / Marginal).
$P(O, S_{prior})$ is also often referred to as the "Generative Model"



(Figure 3 : Bayesian Inference)

The problem with Bayesian inference is that it requires us to know the marginal probability i.e. P(O), which is usually not possible to calculate in real-life scenarios due to complex integrals over all random variables. This is where Variational Inference comes in.

Variational Inference

As the marginal P(O) is unknown, we can't directly infer the posterior distribution of hidden state. Hence, we use a surrogate distribution q(S) which is the best possible approximation of the actual posterior. We find a surrogate which minimizes KL divergence from actual posterior.

$$\mathrm{KL}(q(S) \| p(S|O)) = \int (q(S) \,.\, \log(q(S) \;\; / \, p(S|O)))\, dS$$

The problem is we don't know posterior p(S|O), so we have just reframed the problem. How could we find KL divergence when we don't know the posterior? It turns out that log of marginal: *log(P(O)),* i.e. the Evidence, provides an upper bound to a certain functional (function of function) of q(s): L(q(s)). Hence we call this L functional the Evidence Lower Bound (ELBO). This implies that minimizing KL divergence is equivalent to finding a surrogate q from the family of functions Q (such as Gaussian) which maximizes the ELBO.

$$best\_q(S) = \mathrm{argmax}_{q(S) \in Q}\ \mathrm{ELBO}(q(S))$$

Active Inference

Active Inference extends Variational Inference. Whereas ELBO maximization in Variation Inference is for the goal of finding the best surrogate, in Active Inference it is also tied to agent's Action and Perception. In this framework, Bayesian Surprise can be thought of as the negative of Evidence, i.e. *-log(P(O)).* Surprise would be infinite when P(O) is 0.

Free Energy can be thought of as the negative of ELBO. Hence Free Energy provides an upper bound to Surprise. By minimizing Free Energy, we can consequently minimize surprise. To do this, we tweak the surrogate function q, and we also tweak the generative model $P(O, S_{prior})$. This is for just one time step. To do this for each time step, we repeat the same process for states and observations at multiple timesteps. State at time step t+1 will depend on state at timestep t. But here's the thing: State at time step t+1 will also depend on action at time step t.

Hence, to minimize free energy across multiple time steps, we need to consider different sets of actions, i.e. the policies, and pick the one which minimizes sum of free energy over time. This ties action selection to free energy minimization, and hence consequently to surprise minimizal. This way of action selection is a complete shift of paradigm from Reinforcement Learning based methods. Unlike traditional RL methods that maximize a cumulative reward by trial and error, active inference focuses on minimizing the expected free energy. And unlike RL, this is a single shot process.

Implementation

Description of The Environment and the Agent

There's an agent – FristonianJerry in a grid world of size 6*6. It has to reach the reward of cheese, and avoid two negative reward locations i.e. Toms at all cost. It is aware of the 3 spots where the negative rewards and the positive reward are located, but doesn't know which spot contains what, i.e., doesn't know which of the 3 spots contains positive reward. There's a spot in grid where a cheat sheet is located, which reveals the location of where the positive reward is located. But the agent doesn't know the location of Cheat Sheet either. It could be located in one of the 4 spots. But thankfully there's an observable spot in the grid which contains hint about where the cheat sheet could be found. Hence to reach the cheese, FristonianJerry needs to perform hierarchical planning, by first finding the spot which contains hint to location of cheat sheet, then finding the location of cheat sheet, and then finding the location of cheese.

Generative Process

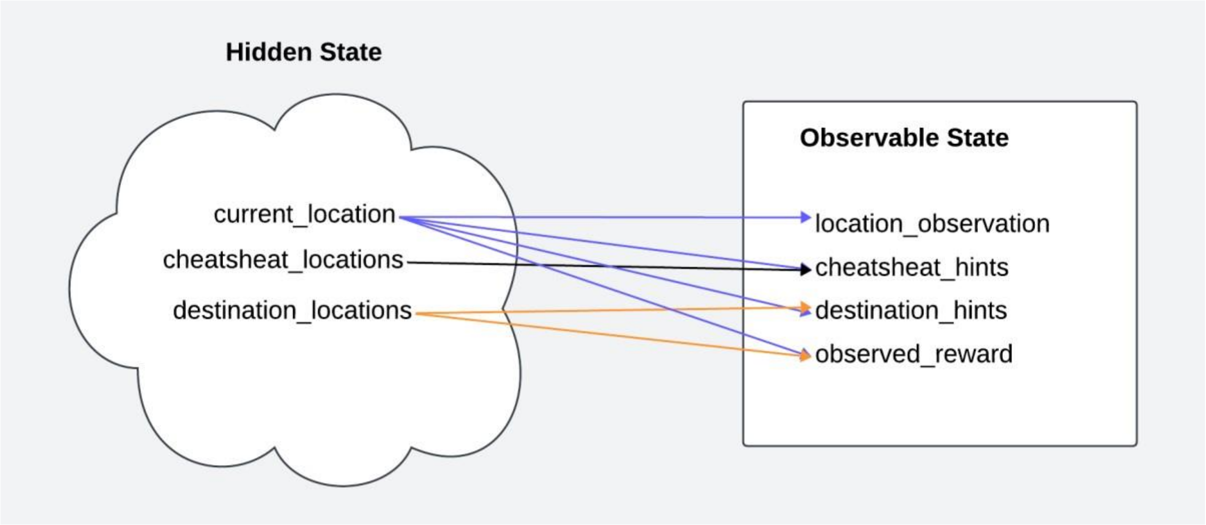
To formulate the environment in terms of Generative Process, I define hidden states, observable states, and likelihoods of observable states given hidden states.

There are 3 hidden random variables: **current_location :** Probability distribution over 36 grid locations in which agent is currently present **cheatsheat_locations:** Probability Distribution over 4 locations where the cheat sheet is possibly located.

destination_locations : Probability Distribution over 3 locations where the cheese reward is possibly located.

Please note that the agent doesn't observe distributions either.

The random variables which are observable to the agent, are: **location_observation** : an indicator of where the agent is currently located **cheatsheat_hints** : an indicator of where the cheat sheat is located. This random variable gives
"Nothing" signals until the agent reaches location which contains hint as to where the cheat sheet is located.
destination_hints : an indicator of where the reward destination is located. This random variable remains gives "Nothing" signals until the agent reaches location which contains cheat sheet.
reward_hints : an indicator of reward at agent's current location



(Figure 5: Generative Process)

Now, I encode the likelihoods of each observable random variable given the hidden state. For this I construct a matrix.

1) Let's start with **location_observation**. *P(location_observation = o | S).*
It can take on 36 realisations (as there are 36 grid spots). For each realisation, there would be a likelihood w.r.t. each hidden state. Hidden State can take on 36 * 4 * 3 combinations. Hence the likelihood matrix would be of dimension 36 * (36 * 4 * 3). But the likelihood of location_observation doesn't really depend on cheatsheat_locations and destination_locations, hence for every combination of cheatsheat_locations and destination_locations, the likelihood would remain constant for a given current_position.
2) Similarly, **cheatsheat_hints** can take on 5 realisations ('Nothing' realisation plus realisations corresponding to 4 possible cheatsheet locations). Hence the likelihood matrix for it would be of dimension 5 * (36 * 4 * 3). Likelihood for 'Nothing' realisation would be the highest everywhere, except for the one corresponding to hidden state in which current_position takes a realization of cheat sheet hint's position.

3) **destination_hints** can take on 4 realisations ('Nothing' realisation plus realisations corresponding to 3 possible destination locations). Hence the likelihood matrix for it would be of dimension 4 * (36 * 4 * 3). Likelihood for 'Nothing' realisation would be the highest everywhere, except for the one corresponding to hidden state in which current_position takes a realization of cheat sheet's position.

Likelihood matrices of all 4 observable random variables are nested inside a single list called likelihood_matrices.

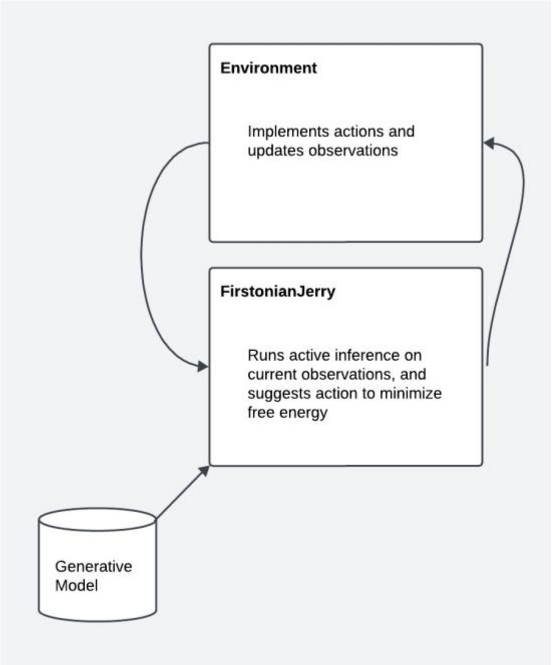
Now, I define the prior distribution of Hidden State, that is, a prior distribution of realisations for each hidden random variable. I just set it to uniform distribution for all variables except current_position. For current_position, the first gridspot would have probability of 1, and rest would be 0.

We're done with the setup of generative model.

Active Inference Loop

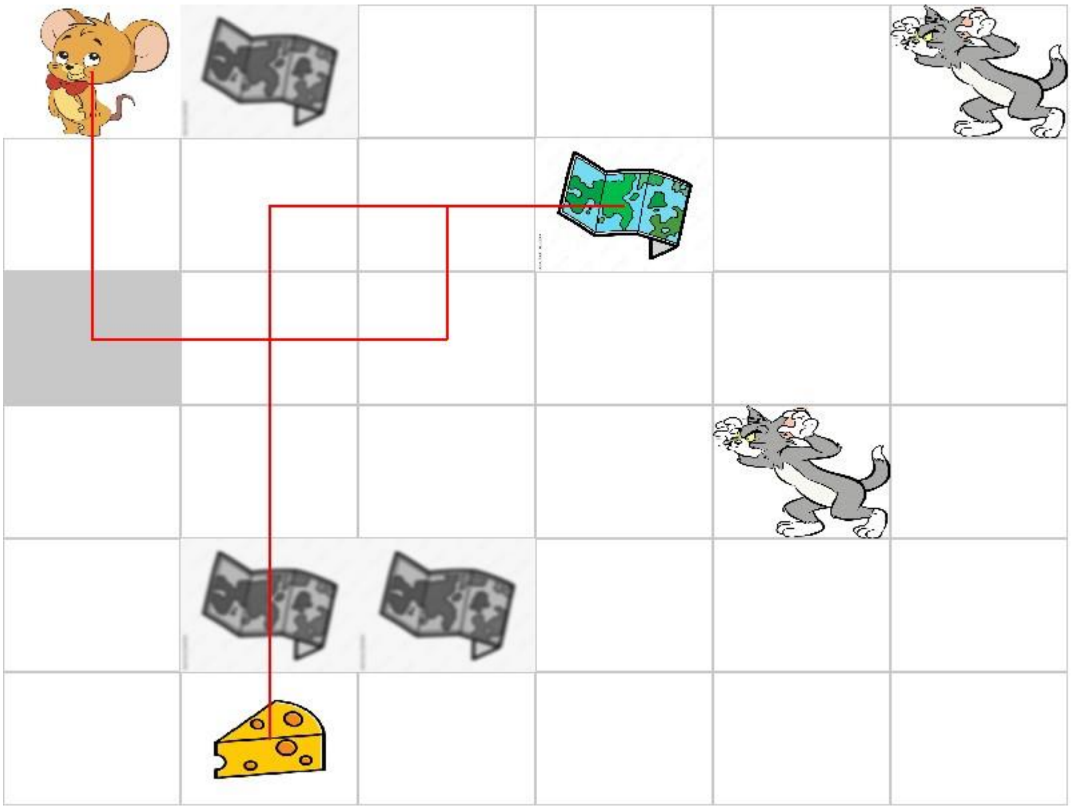
I am using inferactively-pymdp library for free energy minimization.

The Environment would generate observations for the agent. Agent will suggest actions. Environment will implement action and change the observations.


Environment
Implements actions and updates observations
FirstonianJerry
Runs active inference on current observations, and suggests action to minimize free energy
Generative Model

(Figure 7: Inference Loop)

Finally, I run the active inference loop. Here's the result:



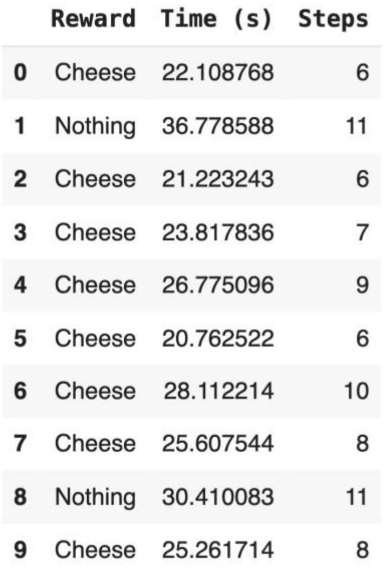
(Figure 8: Output)

The visualization was done using PyGame. Grey spot indicates cheat sheet hint location, world map indicates cheat sheet location, and greyed world maps indicate fake cheat sheet locations.

Evaluation

For evaluating how well Active Inference is actually doing, I run the inference loop 10 times. In each iteration, cheat sheet locations and destination locations are randomly chosen.

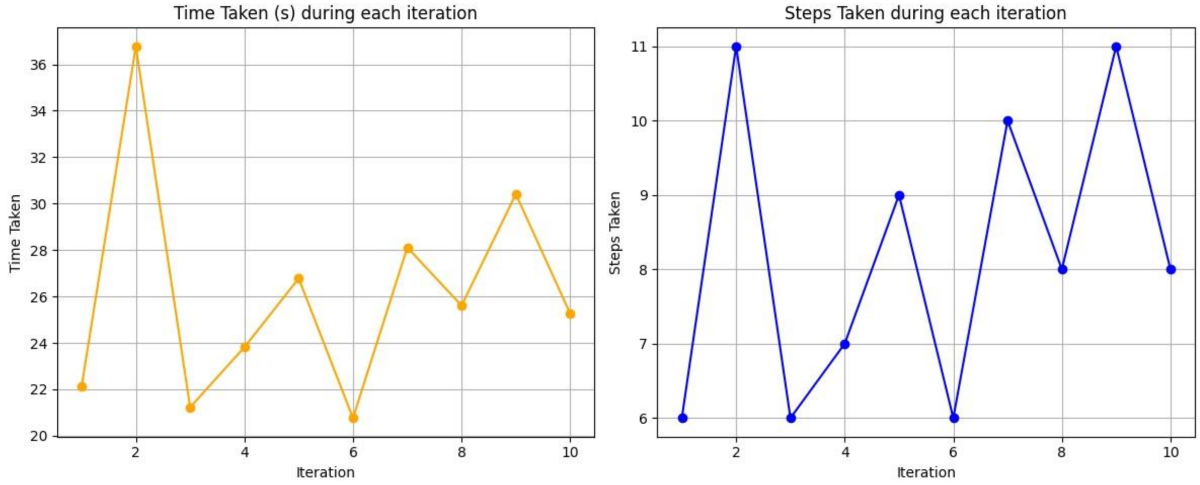
Here's the table of results:



	Reward	Time (s)	Steps
0	Cheese	22.108768	6
1	Nothing	36.778588	11
2	Cheese	21.223243	6
3	Cheese	23.817836	7
4	Cheese	26.775096	9
5	Cheese	20.762522	6
6	Cheese	28.112214	10
7	Cheese	25.607544	8
8	Nothing	30.410083	11
9	Cheese	25.261714	8

(Figure 9)

Encountered Tom: 0 times
Success rate (reaching cheese): 80%
Mean steps during iterations in which completion was successful: 7.5
Mean time taken during each iteration: 26.08 seconds


Time Taken (s) during each iteration

Steps Taken during each iteration

(Figure 10)

Iteration takes more time to complete when it isn’t able to find cheese.

Conclusion and Future Work

This study explored the theoretical background behind Active Inference, and implemented it for a grid-based scenario in Python. The results of experiment demonstrate its ability to plan hierarchically, achieve desired goal states, and completely avoid negative reward states. The important thing to note about Active Inference is that it is **single shot**. There is no explorationexploitation involved. That’s what makes it considerably better than Reinforcement Learning. While this study uses a basic generative model based on gaussian distribution, it could be considerably improved even further by generating custom prior distributions using LSTM or Joint Embedding Predictive Architecture (JEPA).

References

[1] Allen Newell and Herbert Simon, 1972. Human Problem Solving. Prentice-Hall, Englewood Cliffs, NJ.

[2] Richard Bellman, 1961. Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton.

[3] Daniel Dennett, 1984. Cognitive Wheels: The Frame Problem of AI.

[4] John Vervaeke, Timothy Lillicrap, and Blake Richards, 2012. Relevance Realization and the Emerging Framework in Cognitive Science. J. Log. Comput. 22 (2012), 79–99.

[5] Mark Ho, David Abel, Carlos Correa, Michael Littman, Jonathan Cohen, and Thomas Griffiths, 2022. People construct simplified mental representations to plan. Nature, 606(7912), 129-136.

[6] Immanuel Kant, 1787. Critique of Pure Reason (2nd edition)

[7] Martin Heidegger, 1927. Sein und Zeit. Max Niemeyer Verlag.

[8] Esther Thelen and Linda Smith, 1994. A Dynamic Systems Approach to the Development of Cognition and Action. MIT Press.

[9] Humberto Maturana and Francisco Varela, 1980. Autopoiesis and Cognition: The Realization of the Living. D. Reidel Publishing Company.

[10] James Gibson, 1979. The Ecological Approach to Visual Perception. Boston: Houghton Mifflin.

[11] Rodney Brooks, 1991. Intelligence Without Representation. Artificial Intelligence 47, 139–159.

[12] Karl Friston, 2005. "A theory of cortical responses." Philosophical Transactions of the Royal Society B: Biological Sciences.

[13] Karl Friston, James Kilner, and Laurence Harrison, 2006. A free energy principle for the brain. Journal of Physiology- Paris, 100(1-3), 70-87.

[14] Hwang, Ju-Won & Lee, Young-Seol & Cho, Sung-Bae. (2011). Structure evolution of dynamic Bayesian network for traffic accident detection. 2011 IEEE Congress of Evolutionary Computation, CEC 2011. 1655 - 1671. 10.1109/CEC.2011.5949815.