Journal of Mathematical Psychology 96 (2020) 102348

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

Deep active inference as variational policy gradients

Beren Millidge

Department of Informatics, University of Edinburgh, United Kingdom

ARTICLE INFO

Article history:
Received 13 August 2019
Received in revised form 9 January 2020
Accepted 16 March 2020
Available online 13 April 2020

Keywords:
Active inference
Predictive processing
Neural networks
Policy gradients
Reinforcement learning

ABSTRACT

Active Inference is a theory arising from theoretical neuroscience which casts action and planning as Bayesian inference problems to be solved by minimizing a single quantity — the variational free energy. The theory promises a unifying account of action and perception coupled with a biologically plausible process theory. However, despite these potential advantages, current implementations of Active Inference can only handle small policy and state–spaces and typically require the environmental dynamics to be known. In this paper we propose a novel deep Active Inference algorithm that approximates key densities using deep neural networks as flexible function approximators, which enables our approach to scale to significantly larger and more complex tasks than any before attempted in the literature. We demonstrate our method on a suite of OpenAIGym benchmark tasks and obtain performance comparable with common reinforcement learning baselines. Moreover, our algorithm evokes similarities with maximum-entropy reinforcement learning and the policy gradients algorithm, which reveals interesting connections between the Active Inference framework and reinforcement learning.

Active Inference is a proposed unifying theory of action and perception which has emerged out of the Predictive Coding (Clark, 2012, 2013, 2015; Friston, 2003; Rao & Ballard, 1999) and Bayesian Brain (Doya, Ishii, Pouget, & Rao, 2007; Friston, 2012; Knill & Pouget, 2004) theories of brain function (Adams, Shipp, & Friston, 2013; Brown, Friston, & Bestmann, 2011; Friston, Daunizeau, & Kiebel, 2009). It has been applied in a variety of paradigms including modelling choice tasks (Friston et al., 2013, 2014), serving as a basis for exploration, artificial curiosity (Friston et al., 2017, 2015), the explore–exploit trade-off (Friston, Samothrakis, & Montague, 2012), and potentially for illuminating neuropsychiatric disorders (Adams, Perrinet, & Friston, 2012; Barrett, Quigley, & Hamilton, 2016; Mirza, Adams, Parr, & Friston, 2019). Moreover, a neuroscientifically grounded process theory has been developed (Friston, Lin et al., 2017), based on variational message passing (Friston, Parr and de Vries, 2017; van de Laar & de Vries, 2019; Parr & Friston, 2018; Parr, Markovic, Kiebel, & Friston, 2019), that can replicate several observed neuropsychological processes such as repetition suppression, mismatch negativity, and perhaps even place-cell activity (Friston, Lin et al., 2017).

Active Inference casts action and perception as Bayesian inference problems which can both be solved simultaneously through a variational approach that minimizes a single quantity — the variational free-energy. This is in line with the Free-Energy Principle: a deeper theory emerging from predictive processing (Friston, 2009, 2010, 2019; Friston, Kilner, & Harrison, 2006; Friston & Stephan, 2007) which posits that the brain, and perhaps all far-from-equilibrium self-organizing systems, must in some sense minimize their free-energy (Friston & Ao, 2012; Karl, 2012). The variational-free-energy is an information-theoretic quantity that is minimized during variational inference, and is also an upper bound on the negative log-likelihood (or surprisal) of a model. Under certain assumptions the free-energy principle can be translated into a biologically and neuroscientifically plausible process theory (Friston, 2003, 2005) that could theoretically be implemented in the brain.[1] The core idea is that the brain possesses a hierarchy of generative models capable of generating expected sense-data, which learn by minimizing the prediction error between the predicted and observed sense-data. Prediction error thus becomes a general unsupervised training signal, used to successively infer and improve our understanding of the state of the world. Due to the emphasis on prediction error, this theory is known as Predictive Processing. Active Inference extends this idea by applying it to action. There are two ways to minimize prediction error. The first is to update internal models to accurately account for incoming sense-data. This is perception. The second is to take actions in the world so as to bring the incoming sense-data into agreement with the prior predictions. This is action. This duality lets Active Inference treat action and perception under the same formalism, and enables both to be optimized simultaneously by minimizing the variational free-energy.

E-mail addresses: s1686853@sms.ed.ac.uk, beren@millidge.name.

https://doi.org/10.1016/j.jmp.2020.102348
0022-2496/

[1] Some tutorial introductions to the Free-Energy Principle and its applications to neuroscience are: Bogacz (2017), Buckley, Kim, McGregor, and Seth (2017) and Millidge (2019).

Active Inference was first applied to continuous time, state, and action spaces (Friston et al., 2009). A discrete version was later developed which is more in line with current trends in reinforcement learning (Friston et al., 2012), which we focus on here. Although there are significant differences from paper to paper, and we have elided much detail, the general setup of discrete-time Active Inference is as follows:

There is an agent which exists in a Partially Observed Markov Decision Process (POMDP). The agent receives observations o from an environment which has hidden states s which the agent tries to infer. The agent can also take actions which change the environment's state, and thus future observations. The agent's "goal" is to minimize its expected free energy into the future up to some time horizon T. The agent then infers its own actions at the current time to be consistent with this goal.

The agent is equipped with a generative model of its observations, states, and actions which can be factorized as follows:

$$p(o_{0:T}, s_{0:T}, a_{0:T}, \gamma) = p(s_0)p(o_0|s_0)p(\gamma)\prod_{t=1}^{T} p(o_t|s_t)p(s_t|s_{t-1}, a_{t-1}) \times p(a_{t-1}|s_{t-1}, \gamma) \quad (1)$$

where γ is a precision parameter which affects the distribution of actions. Each of the distributions in the factorized generative model is typically represented as a single matrix which is usually provided by the experimenter rather than being learned from experience.[2]

To obtain the posterior distribution of states and parameters given the observations, the agent then inverts this generative model through a process of approximate variational inference. This works by defining variational distributions $Q(s, a) = Q(s; \hat{s})Q(a; \hat{\pi})$ and then minimizing the KL-divergence between these distributions and the generative model (the $\hat{s}$ and $\hat{\pi}$ are learnable parameters of the variational distribution). The divergence between the variational distribution and the generative model is called the variational free energy.

To infer its policy, the agent needs to compute the expected-free-energy (EFE) of each policy, which is simply the sum of the free energies expected under the variational posterior up to the time horizon. What this means in practice is that, for every possible policy, the agent needs to run forward its generative model from the current time until the time horizon, generating fictitious future states and observations it projects it will be in if it follows that policy. It then must compute the free-energy of those states and observations and add them all up to get an estimate of the value of any particular policy. The agent can then sample actions from its action posterior using a Boltzmann distribution with the γ parameter acting as an inverse temperature.

Due to the need to enumerate over every possible policy and project them forwards in time up until the time horizon, this algorithm quickly becomes intractable for large policy spaces or time horizons. It also has trouble representing large state–spaces. We refer to this type of algorithm as tabular Active Inference, by analogy to tabular reinforcement learning, which represents every state in the state–space explicitly as entries in a giant table, and runs into similar scaling issues (Kaelbling, Littman, & Moore, 1996; Sutton, Barto, et al., 1998). Because of these scaling issues, tabular Active Inference has not been applied to any non-toy tasks with large state or action spaces.

[2] Several papers (Friston, FitzGerald, Rigoli, Schwartenbeck, Pezzulo, et al., 2016; Schwartenbeck et al., 2019) do learn at least the "A" matrix representing $p(o|s)$ which can be done by setting hyperparameters governing the distribution of the values in the A matrix and then deriving additional variational update rules for these hyperparameters, but this is not the norm and it has only been applied to learn the "A" matrix.

In this paper, inspired by recent advances in machine learning and variational inference (Goodfellow, Bengio, & Courville, 2016; Kingma & Welling, 2013), we propose a novel deep Active Inference algorithm which uses deep neural networks to approximate the key densities of the factorized generative model. This approach enables Active Inference to be scaled up to tasks significantly larger and more complex than any attempted before in the tabular Active Inference literature. We demonstrate performance comparable to common reinforcement learning algorithms for several baseline tasks in OpenAIGym (Brockman et al., 2016) — the CartPole, Acrobot, and Lunar-Lander task. Our algorithm does not need pre-specified transition or observation models hard-coded into the algorithm as it can learn flexible nonlinear functions to approximate these densities which can be optimized purely through a gradient descent on the variational free-energy functional without needing hand-crafted variational update rules. Moreover, we show how one can use a bootstrapping estimation technique to obtain amortized estimates of the expected-free-energy for a state–action pair without needing to explicitly project the policy forward through time, which potentially enables the algorithm to handle long or infinite time horizons.

We find that the mathematical form of the policy-selection part of our algorithm is somewhat similar to the policy gradients and actor-critic algorithms in reinforcement learning, despite having been derived from completely different frameworks and objectives. We compare and contrast these algorithms with our own algorithm, and highlight how Active Inference natively includes several adjustments that have been empirically found to improve policy gradients but which fall naturally out of our framework. We also investigate how Active Inference includes both information-seeking and reward-seeking terms and we compare them to related work in the reinforcement learning literature.

1. Deep Active Inference

Our deep Active Inference algorithm uses the same POMDP formulation as the tabular version. There is an environment which has internal states which then generate observations which the agent receives. The agent can then take actions in response to these observations that affect the internal state of the environment, and thus the observations that the agent receives in the future. In our case, the agent also receives rewards from the environment, which it uses to construct better policies. The states of the environment are assumed to be Markov, which means that the next state depends only on the current state and the agent's action. The observations generated from the current state are not necessarily Markov.

The agent maintains a generative model of the environment with the same high-level structure comprising states, observations, actions, and rewards. Unlike the observations, however, the states and actions are hidden or latent variables, which the agent must infer from the observations. In Bayesian terms, this means that the agent must compute the posterior probability $p(s_t, a_t|o_{<t})$, where $o_{<t}$ is the history of previous states and actions up to the current time t. However, due to the Markov assumption over states, the posterior is only affected by the current observation and the previous state and action. The effect of all the actions and observations prior to this is mediated through the past state and action. Thus the posterior to compute is really $p(s_t, a_t|o_t, s_{t-1}, a_{t-1})$.

Directly computing this posterior through Bayesian inference is intractable for any but the simplest cases. Instead variational methods are used. These posit additional variational densities Q that the agent controls, which are then optimized by minimizing

the KL divergence between them and the true posterior so that ultimately the Q densities approximate the true posterior densities. The KL divergence to minimize is thus[3]:

$$KL[Q(s_t, a_t) \parallel p(s_t, a_t, |o_t, s_{t-1}, a_{t-1})] = \int Q(s_t, a_t) log(\frac{Q(s_t, a_t)}{p(s_t, a_t, |o_t, s_{t-1}, a_{t-1})}) \quad (2)$$

Using Bayes' rule, the properties of logarithms, and the linearity of the integral, we can split this expression up as follows:

$$\begin{aligned} &KL[Q(s_t, a_t) \parallel p(s_t, a_t, |o_t, s_{t-1}, a_{t-1})] \\ &= E_{Q(s_t,a_t)}[logQ(s_t, a_t)] \\ &\quad - E_{Q(s_t,a_t)}[logp(s_t, a_t, o_t, s_{t-1}, a_{t-1})] \\ &\quad + \int Q(s_t, a_t) logp(o, s_{t-1}, a_{t-1}) \\ &= \int Q(s_t, a_t) logQ(s_t, a_t) \\ &\quad - \int Q(s_t, a_t) logp(s_t, a_t, o_t, s_{t-1}, a_{t-1}) \\ &\quad + logp(o_t, s_{t-1}, a_{t-1}) \end{aligned}$$

This last integral vanishes since $logp(o_t, s_{t-1}, a_{t-1})$ has no dependence on any of the variables in Q – s_t and a_t (we assume here that the future cannot affect the past) – so these can be taken out of the integral, and as the remainder $Q(s_t, a_t)$ is just a distribution it integrates to 1. The final result is:

$$\begin{aligned} &KL[Q(s_t, a_t) \parallel p(s_t, a_t, |o_t, s_{t-1}, a_{t-1})] \\ &= KL[Q(s_t, a_t) \parallel p(s_t, a_t, o_t, s_{t-1}, a_{t-1})] \\ &\quad + logp(o_t, s_{t-1}, a_{t-1}) \end{aligned}$$

Since the *logp* term does not depend on the parameters of Q (and since KL divergence is non-negative), minimizing the KL divergence on the right-hand side is the same as minimizing that on the left. This replaces the difficulties of minimizing the KL between the variational distribution and the true posterior (which is unknown), with that of the KL divergence between the variational distribution and the joint distribution, which is given by the generative model. The variational free energy is simply the KL divergence between the variational and joint distributions:

$$KL[Q(s, a) \parallel p(s, a, o, s_{<t}, a_{<t})] = -F \quad (3)$$

This quantity is the negative of the evidence-based-lower-bound ELBO used in machine learning and variational inference (Blei, Kucukelbir, & McAuliffe, 2017; Hoffman, Blei, Wang, & Paisley, 2013). This is because the variational free-energy is an upper bound on the negative log-likelihood (i.e. surprisal Friston et al., 2006) which is minimized, while the ELBO is a lower-bound on the negative log-likelihood which is maximized.

To evaluate the free-energy at a particular time, we have to deal with the joint distribution $p(s_t, a_t, o_t, s_{t-1}, a_{t-1})$. This term can be factorized according to the generative model of the agent. The agent's generative model assumes that the agent exists inside a POMDP such that the generative process is that observations depend on states, states depend on the previous state and the previous action, and the current action is inferred from the current state only. These assumptions result in the following factorization for the joint density:

$$p(s_t, a_t, o_t, s_{t-1}, a_{t-1}) = p(o_t|s_t)p(a_t|s_t) \times p(s_t|s_{t-1}, a_{t-1})Q(s_{t-1}, a_{t-1}). \quad (4)$$

Here the variational distribution takes the place of the prior from the previous time-step since it is assumed variational inference has already been done to obtain estimates for the previous state and action. The free-energy to be minimized is thus:

$$F = -KL[Q(a_t|s_t)Q(s_t) \parallel p(o_t|s_t)p(a_t|s_t)p(s|s_{t-1}, a_{t-1})] \quad (5)$$

The previous timestep's variational posterior $Q(s_{t-1}, a_{t-1})$ is removed from the equation above since it has no variation with respect to the current variational parameters of $Q(s_t, a_t)$ and so does not affect the optimization. The current variational distribution is taken to factorize as $Q(s_t, a_t) = Q(a_t|s_t)Q(s_t)$. This factorization is not an assumption since it follows directly from the laws of probability. Using standard properties of logs and the definition of the KL divergence, the expression for the free energy splits apart into three terms:

$$-F = -E_{Q(s_t)}[logp(o_t|s_t)] - KL[Q(s_t) \parallel p(s_t|s_{t-1}, a_{t-1})] - E_{Q(s_t)}KL[Q(a_t|s_t) \parallel p(a_t|s_t)] \quad (6)$$

The key element of our method is using deep neural networks to approximate each of the densities in this expression. Looking at the first term, we see we need to approximate the densities $Q(s_t)$ and $logp(o_t|s_t)$. These two densities, one mapping from observations to states and the other mapping back from states to observations are highly reminiscent of the variational-autoencoder (VAE) objective, Kingma and Welling (2013) and can be modelled directly as one.[4] We call this the observation model.

The second term is the divergence between the posterior expected given the observation and the prior expected value of the state given the previous state and action. The variational posterior is given by the encoder model of the observation model, while the prior $p(s_t|s_{t-1}, a_{t-1})$ can be modelled directly by a neural network which outputs the mean and variance of a Gaussian given the previous state and action. We call this the transition model. In this paper we represent the transition model as a simple feedforward neural network, but more complex stateful models such as LSTMs could also be used. Assuming both the posterior and prior densities are Gaussian, the KL divergence is computable analytically. These two terms and their instantiations as neural networks take care of the perception aspect of the Active Inference agent.

The key term for Active Inference is the third term. Let us examine it in more detail. First we note that the KL divergence is under an expectation over the possible states the agent believes it could be in. In this paper we approximate this expectation by a single sample. However, given that we possess the approximate density $Q(s_t)$, a better approximation can simply be computed by sampling as many potential states as desired from this distribution and then rerunning the action selection step. This would increase the accuracy of the approximation at the cost of increased computational expense.

We next see that we can decompose the KL divergence into an energy and an entropy term:

$$E_{Q(s_t)}[KL[Q(a_t|s_t) \parallel p(a_t|s_t)]] = E_{Q(s_t)}[\int Q(a_t|s_t)logp(a_t|s_t) + H(Q(a_t|s_t))] \quad (7)$$

The variational density $Q(a_t|s_t)$ is under complete control of the agent, and we parametrize it by a deep neural network. This

[3] On a notational note, integrals will be used throughout for integrations/sums over states s, actions a, and observations o. This is because the maths is effectively the same whether s,a, or o are discrete or continuous variables. In our experiments, s was continuous and a was discrete. Sums are always used for the time variable t, since the POMDP framework only applies in discrete time.

[4] For a tutorial on VAEs see Doersch (2016).

makes the entropy term $H(Q(a_t|s_t))$ simple to compute as it is simply the entropy of the action distribution output of the neural network. This can be computed simply as a sum for discrete actions. The energy term is more tricky. This is because it involves the true action posterior $p(a_t|s_t)$, which we do not know precisely. First, we will make some assumptions about the form of this density. Specifically, we will assume, following Friston et al. (2015) and Schwartenbeck et al. (2019), and as in the tabular case, that the agent expects that it will act to minimize its expected-free-energy into the future up to a time horizon T, and that the distribution over actions is a precision-weighted Boltzmann distribution over the expected free energies. That is:

$$p(a_t|s_t) = \sigma(-\gamma G(s_{t:T}, o_{t:T})) \tag{8}$$

where σ is a softmax function. This just tells us that the "optimal" free-energy agent would first compute the expected free-energy of all paths into the future for each action it could take, and then choose an action probabilistically by sampling from a Boltzmann distribution over the expected free energies for each action. This method has some empirical support. Boltzmann or softmax choice rules have been regularly used to model decision-making in humans and other animals (Daw, O'doherty, Dayan, Seymour, & Dolan, 2006; Gershman, 2018a, 2018b) and similar rules are applied in reinforcement learning under the term Boltzmann Exploration (Cesa-Bianchi, Gentile, Lugosi, & Neu, 2017). The key term in this equation is the expected free energy $G(s_{t:T}, o_{t:T})$. This is a path integral (or sum) of the free-energy of the expected trajectories into the future given the current state and action. This means that in the case of discrete time-steps and a time horizon T, we can write this as a simple sum:

$$G(s_{t:T}, o_{t:T}) = \sum_{t}^{T} G(s_t, o_t)$$

We first take out the first term of the sum to obtain an expression of the expected free energy for a single time-step:

$$G(s_{t:T}, o_{t:T}) = G(s_t, o_t) + E_{Q(s_{t+1}, a_{t+1})}[\sum_{t+1}^{T} G(s_{t+1}, o_{t+1})] \tag{9}$$

The expectation over states and actions over the second term is due to the stochasticity of the environment. This stochasticity means that there is not a single next state–action pair, so we take an expectation over all possibilities instead.

We can then expand the expected free energy for a single time-step using the definition of the free energy from before as the KL divergence between the variational distribution and the joint distribution to obtain[5]:

$$G(s_t, o_t) = KL[Q(s_t) \parallel p(s_t, o_t)] \tag{10}$$

$$= \int Q(s_t)[logQ(s_t) - logp(s_t, o_t)] \tag{11}$$

$$= \int Q(s_t)[logQ(s_t) - logp(s_t|o_t) - logp(o_t)] \tag{12}$$

$$\approx \int Q(s_t)[logQ(s_t) - logQ(s_t|o_t) - logp(o_t)] \tag{13}$$

$$\approx -logp(o_t) + \int Q(s_t)[logQ(s_t) - logQ(s_t|o_t)] \tag{14}$$

$$\approx -r(o_t) + \int Q(s_t)[logQ(s_t) - logQ(s_t|o_t)] \tag{15}$$

In the third line, since we do not know the true posterior distribution of the states given the observations in the future, we approximate it with our variational density. In the penultimate line, $logp(o_t)$ term can be taken out of expectation since it has no dependence on s_t. This term is crucial to Active Inference as it is the prior expectation over outcomes, which encodes the agents preferences about the world. In the final line we replace the prior probability of an observation $-logp(o_t)$ directly with the reward $-r(o_t)$. This is because, in Active Inference, the agent is simply driven to minimize surprise, and therefore all goals must be encoded as priors built into the agent's generative model, so that the least surprising thing for it to do would be to achieve its goals. Due to the complete class theorem (Friston et al., 2012) any scalar reward signal can be encoded directly as a prior using $p(o_t) \propto exp(r(o_t))$. In this paper,to enable effective comparisons with reinforcement learning methods, the agent's priors simply are to maximize rewards. However, it is important to note that the Active Inference framework is actually more general than reinforcement learning. It can represent reward functions directly as priors using the complete class theorem, but it can also encode other more flexible functions. Additionally, the action posterior $p(a_t|s_t)$ does not necessarily have to be computed using the expected-free-energy. For instance, the action probabilities could be provided directly by observing another agent, which would allow the Active Inference agent to switch seamlessly between imitation and reinforcement learning styles.

The prior term represents the external reward the agent receives, but there is a second term $\int Q(s_t)[logQ(s_t) - logQ(s_t|o_t)]$ which represents something quite different. It requires the maximization of the difference between the prior over future states, and the future "posterior" generated after "observing" the fictive predicted future outcome. This incentivizes visiting states where the transition model is poor, thus endowing the agent with what can be thought of as an intrinsic curiosity. In the Active Inference literature, this term is often called the epistemic or intrinsic value (Friston et al., 2015). Our epistemic value term differs from the standard epistemic action in active inference in an important way. The standard term applies to active inference agents in a POMDP, and drives agents to select observations for which states can be more unambiguously inferred. Our agent is currently situated in an MDP without partial observability, so since there is no uncertainty about state "observations" this term does not apply. Instead our epistemic action term reflects model uncertainty in the transition model, and drives the agent to explore regions where there is a large difference between prior and posterior distributions over the state. This is similar to the epistemic action derived in Schwartenbeck et al. (2019), however we apply it to computing the epistemic action in continuous state spaces rather than discrete MDPs. Since our agent does not have the hidden-state epistemic action term associated with active-inference, only the model-parameter epistemic action term, then it will not behave so as to minimize ambiguity in hidden state mappings. Extending the deep active inference framework to POMDPs and investigating the effect of this extra epistemic action term would be an interesting avenue for future work.

Given this expansion of the expected free energy, we can then represent the total as the sum:

$$G(s_{t:T}, o_{t:T}) = -r(o_t) + \int Q(s_t)[logQ(s_t) - logQ(s_t|o_t)] + E_{Q(s_{t+1}, a_{t+1})}[G(s_{t+1:T}, a_{t+1:T})] \tag{16}$$

Trying to compute this quantity exactly is intractable due to the need to explicitly compute many future trajectories and the expected-free-energy associated with each one. However, it is possible to learn a bootstrapped estimate of this function

[5] Action is not included as a free parameter in the variational or generative densities since the expected-free-energy is a function of both states and observations — i.e. the EFE is evaluated for every action so the action is implicitly conditioned on.

from samples using a neural network to learn an amortized inference distribution. We define an approximate expected-free-energy (EFE)-value network $G_\phi(s_t, o_t)$, with parameters ϕ, which maps a state action pair to an estimated EFE. This estimated EFE is then compared to a second estimate of the EFE $\widehat{G(s_t, o_t)}$ which uses the free energy calculated at the current time-step but approximates the rest of the trajectory with another estimate by the EFE-value net estimate for the next time-step. That is:

$$\widehat{G(s_t, o_t)} = -r(o_t) + \int Q(s_t)[logQ(s_t) - logQ(s_t|o_t)] + G_\phi(s_{t+1}, o_{t+1}) \tag{17}$$

The difference between the two estimates can then be minimized by a gradient descent procedure with respect to the parameters of the EFE-valuenet ϕ. In this paper the L2 norm is used as a loss function for the gradient descent:

$$L = \|G_\phi(s_t, o_t) - \widehat{G(s_t, o_t)}\|^2 \tag{18}$$

This procedure is analogous to bootstrapped value or Q function estimation procedures in reinforcement learning, for which guarantees of convergence exist in tabular cases. It also empirically has been found to work for deep neural networks in practice, albeit with various techniques needed to boost the stability of the optimization procedure, despite the lack of any theoretical convergence guarantees.

Given that we now possess a means to estimate $G(s_t, o_t)$ and thus the action posterior $p(a_t|s_t)$, the action model $Q(a_t|s_t)$ can be trained to directly minimize the loss function $\int Q(a_t|s_t)logp(a_t|s_t) + H(Q(a_t|s_t))$. Gradients of this expression with respect to the parameters of $Q(a_t|s_t)$ can be computed analytically or by using automatic differentiation software.

To recap, the deep Active Inference agent possesses four internal neural networks. A perception model which maps observations to states and back again and models the distributions $Q(s_t|o_t)$ and $p(o_t|s_t)$, and is trained with a VAE-like log-probability-of-observations loss. A transition model which models the distributions $p(s_t|s_{t-1}, a_{t-1})$ is trained to minimize differences between the predicted transition and the actually occurring state at the next time-step obtained through $Q(s_t|o_t)$. An action model, which models the distribution $Q(a_t|s_t)$, and can be trained directly through gradient descent on the loss function $\int Q(a_t|s_t)logp(a_t|s_t) + H(Q(a_t|s_t))$, and a value network $G_\phi(s, a)$ that is trained through a bootstrapped estimate of the expected-free-energy, as explained above.

We present our deep-active-inference algorithm in full in Algorithm 1:

Unlike the tabular Active Inference algorithms proposed by Friston and colleagues, this algorithm approximates important densities with neural networks, which can all be optimized through a simple gradient descent procedure on the expression for the variational free energy. The computation graph is fully differentiable so that derivatives of the parameters of the networks can be computed automatically using automatic differentiation software without the need for hand-derived complex variational update rules or black-box optimization techniques. Moreover, although in this paper the densities were approximated using simple multi-layer perception networks, in principle each density can be approximated by a neural network, or other differentiable function, of any size or complexity, thus enabling this algorithm to scale indefinitely.

Algorithm 1 Deep Active Inference

Initialization:
Initialize Observation Networks $Q_\theta(s|o), p_\theta(o|s)$ with parameters θ.
Initialize State Transition Network $p_\phi(s_t|s_{t-1}, a_{t-1})$ with parameters ϕ
Initialize policy network $Q_\xi(a|s)$ with parameters ξ
Initialize bootstrapped EFE-network $G_\psi(s, o)$ with parameters ψ
Receive prior state s_0
Take prior action a_0
Receive initial observation o_1
Receive initial reward r_1
function ACTION-PERCEPTION-LOOP
  while $t < T$ **do**
    $\hat{s}_t \leftarrow Q_\theta(s_t|o)(o_t)$ ▷ Infer the expected state from the observation
    $\hat{s}_{t+1} \leftarrow p_\phi(s|s_{t-1}, a_{t-1})(\hat{s}_t)$ ▷ Predict the state distribution for the next time-step
    $a_t \sim Q_\xi(a_t|s_t)$ ▷ Sample an action from the policy and take it
    Receive observation o_{t+1}
    Receive reward r_{t+1}
    $\tilde{s}_{t+1} \leftarrow Q_\theta(s|o)(o_{t+1})$ ▷ Infer expected state from next observation
    Compute the bootstrapped EFE estimate of from the current state and action:
    $\widehat{G(s_t, o_t)} \leftarrow r_{t+1} + E_{Q(s_{t+1})}[log\hat{s} - log\tilde{s}] + E_{Q(s_{t+1}, a_{t+1})}[G_\psi(s_{t+2}, o_{t+2})]$
    Compute the Variational Free Energy F:
    $F_t \leftarrow E_{Q(s_t)}[logp(o_t|s_t)] + KL[\hat{s}_{t+1}||\tilde{s}_{t+1}] - E_{Q(s_t)}[\int da Q_\xi(a_t|s_t)\sigma(-\gamma G_\psi(s_t, o_t)(s_{t+1})) + H(Q_\xi(a_t|s_t))]$
    $\theta \leftarrow \theta + \alpha \frac{dF}{d\theta}$ ▷ Update the θ parameters
    $\phi \leftarrow \phi + \alpha \frac{dF}{d\phi}$ ▷ Update the ϕ parameters
    $\xi \leftarrow \xi + \alpha \frac{dF}{d\xi}$ ▷ Update the ξ parameters
    $L \leftarrow ||\widehat{G(s_t, o_t)} - G_\psi(s_t, o_t)||^2$ ▷ Compute the boostrapping loss
    $\psi \leftarrow \psi + \alpha \frac{dL}{d\psi}$ ▷ Update the ψ parameters
  end while
end function

2. Relation to policy gradients

Reinforcement learning is perhaps the dominant paradigm used to train agents to solve complex tasks with high dimensional state–spaces (Lillicrap et al., 2015; Mnih et al., 2015; Silver et al., 2017). Reinforcement learning also formulates the action-problem as an MDP with states, actions and rewards. In reinforcement learning, however, instead of acting to minimize expected surprise, we simply maximize expected rewards. The agent's goal at every time-step is simply to maximize the sum of discounted rewards over its trajectories into the future. This can be written as:

$$G_{t:T} = \sum_i^T \gamma^{i-1} r(s_t, a_t) \tag{19}$$

Where γ is a discount factor that reduces the impact of future rewards. We can also define state and state–action reward functions which map states and actions to the reward expected from that state or state–action pair under a particular policy.

$$Q^\pi(s_t, a_t) = E_\pi[G_{t:T}|s = s, a = a] \tag{20}$$

$$V^\pi(s_t) = E_\pi[E_{a_{t:T}}[G_{t:T}|s = s]] \tag{21}$$

where π is shorthand for the controlled state trajectory under a particular policy: $\pi = p(s_{1:T}|a_{1:T})$.

The goal of an agent is to find a policy which can maximize the expected sum of discounted rewards. This goal can be written as the objective function:

$$J(\theta) = E[G_{t:T}] = \sum_{t=0}^{t=\infty} \int p(s_t|s_{t-1}, a_{t-1}) p_\theta(a_{t-1}|s_{t-1}) G(s_t, a_t) dsda \tag{22}$$

where θ are the parameters of the policy which outputs the distribution $p_\theta(a|s)$. There are two ways to optimize this objective. The first is to obtain it directly by maximizing over the Q function, and thus not explicitly represent the policy at all. This leads to the Q and TD learning family of algorithms. The second way is to explicitly represent the policy, for instance using a deep neural network, and train the policy directly to maximize the sum of discounted rewards. We focus on this second approach since it bears the greatest similarities with our deep Active Inference algorithm.

We can compute the gradients of J with respect to θ directly using the following log-gradient trick (Sutton, McAllester, Singh, & Mansour, 2000):

$$\nabla_\theta J(\theta) = \sum_t \int \nabla_\theta p(s_t|s_{t-1}, a_{t-1}) p_\theta(a_{t-1}|s_{t-1}) G(s_t, a_t) \tag{23}$$

$$= p(s_t|s_{t-1}, a_{t-1}) p_\theta(a_{t-1}|s_{t-1}) \times \frac{\nabla_\theta p(s_t|s_{t-1}, a_{t-1}) p_\theta(a_{t-1}|s_{t-1})}{p(s_t|s_{t-1}, a_{t-1}) p_\theta(a_{t-1}|s_{t-1})} G(s_t, a_t) \tag{24}$$

$$= \sum_t \int p(s_t|s_{t-1}, a_{t-1}) p(a_{t-1}|s_{t-1}) \nabla_\theta log p_\theta(a_{t-1}|s_{t-1}) \times G(s_t, a_t) \tag{25}$$

$$= \sum_t E_{p(s_t|s_{t-1}, a_{t-1}) p(a_{t-1}|s_{t-1})} [\nabla_\theta log p_\theta(a_{t-1}|s_{t-1}) G(s_t, a_t)] \tag{26}$$

These algorithms directly maximize the reinforcement objective and so, unlike Q-learning methods, have some guarantees of formal convergence. However, the gradients often suffer from high variance. The return $G(s_t, a_t)$ can be estimated using Monte-Carlo methods or can be approximated using a function approximation method such as Q-learning. Algorithms that do the latter are called actor-critic algorithms since they contain both an "actor" network — the policy, and a "critic" network which learns to approximate the value function.

Of interest is the close similarity between our algorithm which was derived directly from a variational approximate inference approach minimizing the variational free energy, and the policy gradient algorithm derived from maximizing the discounted sum of expected returns. We write the out the loss functions side by side, while simplifying some of the extraneous notation to make the comparison more clear. J_{AI} is Active Inference and J_{PG} represents policy gradients.

$$J_{AI}(\theta) = E_{Q(s)}[\int Q_\theta(a|s) log p(a|s) + H(Q(a|s))] \tag{27}$$

$$J_{PG}(\theta) = E_{p(s,a)}[log p_\theta(a|s) G(s, a)] \tag{28}$$

There are several interesting similarities and differences. The first is the additional entropy term $H(Q(a|s))$ in the Active Inference objective. This means that the Active Inference agent does not merely try to maximize the expected-free-energy into the future, it does so while also trying to maximize the entropy of the distribution over its actions. Intuitively, this maximum entropy objective makes the agent try to act as randomly as possible while still achieving high reward, which significantly aids exploration. Interestingly, a recent strand of reinforcement learning literature also focuses on adding an entropy regularization term to the loss (Haarnoja, 2018; Haarnoja, Tang, Abbeel, & Levine, 2017; Haarnoja, Zhou, Abbeel, & Levine, 2018; Rawlik, Toussaint, & Vijayakumar, 2010, 2013; Ziebart, Maas, Bagnell, & Dey, 2008) and have shown this to empirically aid performance and exploration on many benchmark tasks. The maximum-entropy framework has also inspired work on relating reinforcement learning to probabilistic and variational approaches (Fellows, Mahajan, Rudner, & Whiteson, 2018; Levine, 2018).

The second difference between the algorithms lies in the value function. Policy gradients uses the state–action value functions directly, while Active Inference replaces this with a log probability which is derived from a precision-weighted softmax over the value-function. It is unclear which approach is to be preferred, although the log-probabilities may help reduce the variance of the gradient by reducing the magnitude of the multiplier since probabilities are inherently normalized. This difference is central to Active Inference, which at a high level aims to minimize surprise in the future, as opposed to reinforcement learning which aims to maximize reward. As stated in the introduction, reinforcement learning can be subsumed within Active Inference since rewards can simply be defined to be highly expected states. The precision-weighting of action-policy selection is another interesting mechanism unique to active inference, and introduces another parameter – precision – into the algorithm which effectively controls the randomness of action selection. The effect of this precision parameter is mathematically equivalent to Boltzmann exploration (Sutton et al., 1998), which is another widely used exploration method in reinforcement learning. In our experiments to follow, the precision was always set to 1. Exploring the effects of precision, and perhaps learning it adaptively by treating it as a free parameter in the free-energy would be a fruitful avenue for future work.

Another difference lies in the representation of the policy. Policy gradient methods optimize the log probability of the policy, while Active Inference directly optimizes the raw probability values. It is still an open question how this changes the dynamics of learning under Active Inference compared to policy gradients, although there is some reason to expect the log-probabilities to be better conditioned. Additionally, Active Inference explicitly computes the integral over the action (at least in discrete action spaces) using the counterfactual predicted results from the action, while the policy gradient method only samples from this integral by using the actions the agent actually took during the episode. This should theoretically reduce variance since it is computing the true expectation rather than the sample advantage, and an analogous scheme has also been empirically found to improve performance in actor-critic algorithms (Asadi et al., 2017; Ciosek & Whiteson, 2018).

The final difference relates to the outer expectation. The policy gradient expression is taken under an expectation over the trajectory distribution which includes the true environmental dynamics, which are generally unknown. This means that the only way to correctly make updates is to sample the expectation using states that are derived from the current policy. This means that policy gradients are naturally on-policy algorithms which can only validly use the data obtained under the current policy. However, during training the policy changes often, rendering past data unusable. Active Inference, however, requires an expectation taken under the transition model, which is known. This means that Active Inference algorithms can be applied "off-policy", which is a large advantage. Any data can be used to train an Active Inference agent — even that which was collected under a completely different and perhaps even random policy, provided the transition model is known and accurate. While off-policy

variants of policy gradient and actor-critic algorithms have been proposed (Degris, White, & Sutton, 2012; Haarnoja et al., 2018; Song, Lewis, Wei, & Zhang, 2015) the native off-policy status of Active Inference is a large advantage.

Interestingly, many of the differences between Active Inference and policy gradients, such as the entropy term and the explicit computation of the expectation over the action, have been theoretically and empirically shown to improve performance of policy gradients. These improvements to policy gradients naturally fall out of the Active Inference framework. The similarities between the two algorithms also highlight a potentially close connection between reinforcement learning and Active Inference, especially the link between maximum entropy reinforcement learning and variational inference. The exact relationships between the paradigms of Active Inference, maximum entropy reinforcement learning, and stochastic optimal control (Botvinick & Toussaint, 2012; Friston, 2011; Rawlik et al., 2010) still remain unclear, however.

3. Related work

A significant amount of work has gone into exploring tabular Active Inference algorithms and questions such as how the framework encompasses epistemic value and exploration (FitzGerald, Schwartenbeck, Moutoussis, Dolan, & Friston, 2015; Friston et al., 2015; Moulin & Souchay, 2015; Pezzulo, Cartoni, Rigoli, Pio-Lopez, & Friston, 2016), models of active vision (Friston, Rosch, Parr, Price, & Bowman, 2018; Mirza, Adams, Mathys, & Friston, 2016; Parr & Friston, 2017, 2018a), biologically plausible neural process theories (Friston, Lin et al., 2017; Parr & Friston, 2018c), the connections to the motor system (Adams et al., 2013), implementations based on variational message passing (Friston, Parr et al., 2017; van de Laar & de Vries, 2019; Parr et al., 2019), and even insight, curiosity, and concept-learning (Friston, Lin et al., 2017; Schwartenbeck et al., 2019; Smith, Schwartenbeck, Parr, & Friston, 2019). These methods though, while providing insight into the qualitative dynamics of Active Inference and the importance of various parameters, are inherently non-scalable due to their exponential complexity, and they have not been applied to anything beyond simple toy-tasks.

Ueltzhöffer (2018), to our knowledge, is the first paper to propose approximating the observation and transition models with deep neural networks. They use single layer tanh networks with sixteen neurons which outputs the mean and variance of a diagonal conditional gaussian. They used this model to solve the Mountain-Car problem from OpenAI gym. A key difference of this work is how they represented action. They computed continuous actions in a manner that required them to know the partial derivatives of the sensations given the action, which meant propagating through the environmental dynamics, which are unknown. Due to this they had to use a black box evolutionary optimizer to optimize their models, which is substantially more sample-inefficient.[6] In our model we do not use this approach, but instead use a learned amortized inference distribution $Q(a|s)$ and minimize this using a variational approach on the divergence with the approximated true posterior of the value function $p(a|s)$, which is learned through a bootstrapping estimation procedure. Due to this our method is end-to-end differentiable and all networks can be trained through gradient descent on the variational free-energy.

While this paper was in preparation, a paper by Catal, Nauta, Verbelen, Simoens, and Dhoedt (2019) came out along similar lines. They also parametrized the observation and transition models with deep neural networks, and they used a "habit" policy to approximate the expected free energy, analogously to Q learning in reinforcement learning. However, they only applied their model to the Mountain-Car task and also did not utilize the full variational derivation of the KL divergence of the action model and the action posterior, but instead used their habit policy or EFE-approximating network to select actions directly through a softmax choice rule. Instead we maintain a separate policy network which adheres more closely to the full free-energy derivation and also solve significantly more complex tasks than the Mountain-Car.

A related approach was taken by Cullen, Davey, Friston, and Moran (2018) who present an Active Inference model that can play a simplified model of Vizdoom taken from the OpenAI gym environment. Their model is presented as an example of computational psychiatry in that they attempt to relate ageing and anhedonia to differences in model or model parameters (6-state vs 8 state model for ageing, and the C vector for anhedonia). They then compare the behaviours of the model to data generated from human subjects. Their model, however, is an example of the tabular case and not especially scalable due to this reason. Their agent plays an extremely simplified version of DOOM where the goal is simply to move the player in front of the monster and shoot. Moreover, their model does not handle the raw sensory input of DOOM, but rather they first parse the visual scene into 8 or 6 discrete states using the Harris corner detection algorithm. Our algorithm, on the other hand, uses deep neural network to directly learn from raw continuous sensory variables, plays over a long temporal horizon, and deals with significantly more complex goals.

Furthermore, although the B matrix is learned, their model does not utilize neural networks or any functional approximation method, instead following the tabular method of directly representing all possible states of the game, which works in this case because there are only 8. Similarly, policies are computed through an exhaustive combinatorial search through all possible combinations of actions up to a time horizon three actions deep. Once again this is not scalable to large state or action spaces or long time horizons. Their model also deals entirely with discrete states in an MDP whereas our neural network model handles multidimensional continuous states, only requiring discrete actions. This is due to the fact that the focuses of the works are different. In Cullen et al. (2018), they are primarily interested in altering aspects of the model to fit to human decision-making under various psychiatric conditions. Our model is concerned primarily with demonstrating the scalability of active inference even if this comes at the cost of the biologically plausible process theory.

Lastly, Active Inference also brings together several contemporary strands of deep reinforcement learning. There has been much work on model-based reinforcement learning which uses models for planning and state estimation, including using separate observation and transition models and unsupervised objectives similar to Active Inference (Deisenroth & Rasmussen, 2011; Ha & Schmidhuber, 2018; Wayne et al., 2018). Deep Active Inference is model-based from the start, and the three separate models effectively fall out of the probabilistic formalism. There has also been work posing the reinforcement learning problem as a variational inference problem. One thread of this work derives maximum entropy reinforcement learning and has been shown to improve benchmark results on many tasks (Botvinick & Toussaint, 2012; Fellows et al., 2018; Levine, 2018; Rawlik et al., 2010, 2013). The formalism of Active Inference differs slightly in the way it handles rewards and sets up the general MDP formalism. The maximum entropy reinforcement learning inference methods typically set up the inference problem by assuming binary optimality variables, and then conditions on those variables being true, where

[6] This is not to say that this is the wrong approach. Some parts of the generative model of living organisms are effectively fixed by evolution, and so using an evolutionary optimizer is also a valid and principled approach.

the probability of them being true is proportional to the exponentiated reward. Active Inference by contrast does not introduce any auxiliary variables but instead encodes the reward directly in the priors. Beyond this, the detailed connection between Active Inference and the maximum entropy formulation of reinforcement learning remain obscure, despite the fact that they may be equivalent given the close similarities of many of the resulting equations.

Moreover, there has also been much work focusing on intrinsic motivations for reinforcement learning agents. For a theoretical review see Oudeyer and Kaplan (2009). There has been work which uses prediction error directly, Stadie, Levine, and Abbeel (2015), and also information gain (Houthooft et al., 2016a, 2016b; Mohamed & Rezende, 2015) as epistemic rewards in a manner similar to our approach. In Active Inference, however, the form of the epistemic rewards naturally falls out of the framework rather than being postulated on an ad-hoc basis. However, it is still unclear whether the type of epistemic reward proscribed by the expected-free-energy is optimal for exploration, and indeed other forms of epistemic reward may be better. Moreover, it is worth noting, as done in Biehl, Guckelsberger, Salge, Smith, and Polani (2018), that despite the common use of the expected free energy as the prior, this is in fact arbitrary, and other intrinsic motivations can be substituted.

4. Model

While the derivation above has been couched in the language of POMDPs, and all the models presented below can be straightforwardly extended to handle them, the environments we used in this paper were not partially observed, but rather only MDPs. This means that the mapping from observations to hidden states was unnecessary and thus dispensed with for greater simplicity. The key contribution of this paper is fundamentally the action selection mechanism, and not training a neural network to learn $p(o|s)$. However the transition model was still required and used to compute the epistemic reward.

The policy network, transition network, and value-network were each a two-layered perceptron with 100 hidden units and a relu activation function. All networks were trained through minimizing the free-energy objective using the ADAM optimizer. A learning rate of 0.001 was used throughout. The value network was trained using the bootstrapped estimator. All hyperparameters were held constant for all tasks. No complex hyperparameter tuning was necessary for reasonable performance on our benchmarks. No preprocessing was done on the states, rewards, or actions [7].

The stability of bootstrapped value estimation is a large topic in reinforcement learning. Convergence is not guaranteed for nonlinear function approximators, and stability has been shown to be an issue empirically (Fujimoto, van Hoof, & Meger, 2018). Numerous methods have been discovered in the literature to aid the stability and learning. In this paper we implemented only two of the most basic techniques which are now used universally in deep-Q learning: a memory buffer (Mnih et al., 2013) and a target network (Van Hasselt, Guez, & Silver, 2016). A memory buffer stores a history of previous states visited, and each gradient descent step is taken on a batch of state–action-reward-next-state tuples taken from the buffer. This prevents overfitting to the immediate history, which contains many states that are highly correlated with one another and reduces gradient variance. Secondly, we used a target network, which "freezes" the weights of the value network used in the $G(\hat{s}, a)$ estimator for a number of epochs — in our experiments we updated after fifty epochs. This enables the value-network to make gradient steps without constantly chasing a moving target, and so aids stability.

Since the bootstrapping estimator for the value function estimator had a significant effect on the behaviour of the model, we believe that large performance gains could be had by implementing many of these techniques and fine-tuning hyperparameters. However, this sort of optimization was not the goal of this paper, which is intended to be more of an introduction and a demonstration of the potential of deep-active-inference rather than a performance contest. Examples of such implementation details can be found in Fujimoto et al. (2018).

For comparison, we implemented two reinforcement baseline methods: Q-learning and Actor-critic. Q-learning simply learns the state–action value function through a bootstrapping procedure similar to the one we used to approximate the EFE. It does not maintain a separate representation of the policy, but simply chooses actions directly based on the maximum Q-value that it computes. For a fair comparison, the Q-learning agent we implemented used a Boltzmann exploration rule in its action selection, similar to the softmax over policies implemented in our deep Active Inference algorithm. This also gave the Q-learning agent sufficient stochasticity to explore enough to converge to a good policy for all of the environments.

The Q-learning agent learned a value-network, which was a multi-layer perceptron with a single hidden layer of 100 neurons and a relu activation function. This was identical in the numbers of neurons and activation function to the neural networks used in the deep Active Inference agent. All hyperparameters were kept the same as in the deep Active Inference agent. Like the Active Inference agent, the Q-learning agent used a memory replay buffer and a target network to help stabilize training.

Actor critic algorithms are variants of policy gradient algorithms that use a separate "critic" neural network to estimate the value function instead of directly estimating it through Monte-Carlo returns. We instantiated the actor critic algorithm with a policy network and a value network identical to the deep-active-inference agent. All hyperparameters were the same as for the deep Active Inference agent. The value-network was trained using Q-learning and also possessed a memory buffer and a target network.

5. Results

The performance of the Active Inference and baseline reinforcement learning agents was measured on three tasks from the OpenAI Gym. The tasks were Cartpole Environment, the Acrobot Environment, and the Lunar-Lander environment. While not extremely high dimensional tasks, they are significantly more challenging than any before attempted in the Active Inference literature. Example frames from the three games are shown in Fig. 1.

The goal of the CartPole environment is to keep the pole balanced upright atop the cart. The state space comprises of four values (the cart position and cart velocity, and the angle θ of the pole and the angle velocity). The reward schedule is $+1$ for every time-step the episode does not end. The episode ends whenever the angle of the pole is more than 15 degrees from vertical, or the base of the cart moves more than 2.4 units from the centre.

In the Acrobot environment, the agent controls a two-jointed pendulum, and the aim is to swing it from a downward position to being balanced vertically at 180 degrees. Reward is 0 if the arm of the Acrobot is above the horizontal and -1 otherwise. This poses a challenging initial exploration problem since getting any reward other than -1 is very unlikely with random actions. The optimal solution would net slightly less than 0 reward (given

[7] The code to reproduce the model and all experiments can be found at: https://github.com/BerenMillidge/DeepActiveInference

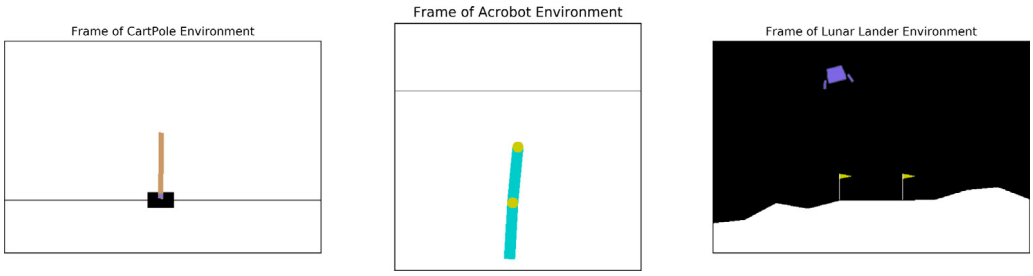


Fig. 1. Frames from the three environments. From left to right: CartPole-v1, Acrobot-v1, LunarLander-v2. For more information about these environments beyond what is in this paper, please consult the OpenAIGym documentation.

the time needed to swing up the Acrobot when it would be accruing negative reward). The state–space of the environment is a 6 dimensional vector, which represents the various angles of the joints. The action space is a 3 dimensional vector representing the force on each joint.

The goal of the Lunar-Lander environment is to land the rocket on a landing pad that is always at coordinates (0,0). The state–space is 8-dimensional and the action space is 4-dimensional with the actions being fire left engine, right engine, upwards engine, and do nothing. The agent receives a reward of $+100$ for landing on the pad, $+10$ for each of the rocket-legs are standing, and -0.3 for every time-step the rocket's engines are firing. The maximum possible reward for an episode is $+200$.

The performance of the Active Inference agent was compared to two baseline reinforcement learning algorithms (Q-learning and actor-critic). Each agent began with randomly initialized neural networks and had to learn how to play from scratch, using only the state and reward data provided by the environment. We ran 20 trials of 15 000 episodes each, and the mean reward the agent accumulated on each episode of the CartPole environment is plotted in Fig. 2:

Here we can see that the Active Inference agent actually outperforms both of the reinforcement learning baselines by a significant margin in the end, and the mean reward reaches the maximum score of $+500$. The actor-critic algorithm does slightly better but does not manage to reach a mean of $+500$ reward per episode, and the Q learning algorithm performs even worse. This demonstrates that the Active Inference agent can be competitive with, and can even beat conventional reinforcement learning algorithms on some benchmarks.

We now perform an ablation experiment on the Active Inference network to test how the various terms in the algorithm affect performance. We compare the full Active Inference network with two ablated versions. One model ("No-Tmodel" on the graph) lacks the epistemic value component of the value function (see Eq. (15)), and instead estimates the reward only, as in Q learning. The second model ("No-Entropy" on the graph) lacks the entropy term of the KL loss, and so only optimizes the policy by minimizing $\int Q(a|s)logp(a|s)$ without the entropy term. The results are plotted in Fig. 3:

An interesting result here is that the main contribution to the success of Active Inference is the entropy term in the loss function. Without the entropy term the Active Inference agent converges to a lower mean reward which is comparable to the performance of the actor-critic and slightly better than the Q-learning algorithm (see Fig. 4).

However, the mean rewards are somewhat misleading since the actual distribution of rewards over the runs appears to be bimodal. In most cases all algorithms successfully converged to the maximum of 500 rewards per episode. However, in several cases, the algorithm fails to converge at all, and usually collapses to a very low reward per episode. The mean reward obtained therefore effectively measures the proportion of successful runs rather than the mean of an average run. To see this, the graphs below show a superposition of every single run for the Active Inference agent, the actor-critic, the q-learning agent, and the active-inference-with-entropy agent. We see that the rewards per episode in each run typically bifurcate and either end up being nearly optimal or near zero. This is especially obvious in the ablated Active Inference agent and also in the Q-learning agent, albeit with much more variance (see Fig. 5).

We call this bifurcation into either near-optimal or completely failed runs "policy collapse" since, at some point during training, the distribution over actions given by the policy will abruptly collapse to put all probability mass on a single action, resulting in the agent rapidly tipping over the pole and obtaining a very low score. The entropy-regularized Active Inference algorithm does better because the entropy term encourages the optimizer to spread the probability between all the possible actions as much as possible while also maximizing reward. It is unclear, however, why exactly policy collapse occurs. Moreover, it appears to be a phenomenon the baseline reinforcement learning agents suffer from as well. Interestingly, the entropy term in the Active Inference agent, while it appears to prevent policy collapse, does not simply cause the agents reward to converge cleanly to the maximum. Instead, the reward obtained per episode fluctuates wildly from optimal to near 0, which may be the policy constantly attempting to collapse but being prevented by the entropy term repeatedly.

We found little to no difference when the epistemic value was not computed and the expected-free-energy was simply reduced to the reward. This also held true in the other tasks and runs counter to some of the proposed benefits in the literature for explicit epistemic foraging. We may have observed little effect of the epistemic value for several reasons. Firstly, the tasks used were fairly simple in terms of goals: all they required was simple motor control without any particular need for long-term goal-directed exploration. It is thus possible that random exploration alone, ensured by the entropy-maximizing component of the policy provided sufficient exploration. Secondly, the epistemic value only enters the Active Inference equations as a term in the expected-free-energy, which was estimated through bootstrapping methods. If these estimates were inaccurate or unstable in the first place, then adding an epistemic value term could have little effect. Thirdly, the epistemic value term is defined as the difference between the expected and the observed state posteriors from the transition model. While these were large initially, the magnitude of these prediction errors rapidly declined as the transition model improved, reaching a steady state far smaller than the average extrinsic reward. Thus, the contribution to the expected free-energy from the epistemic rewards would be small, and so would have little impact on behaviour. We demonstrate this by showing the mean time-course of the epistemic reward over the course of the episodes (see Fig. 6).

The transition-model loss declines extremely rapidly to near 0, and the agent thus moves from an exploratory to an exploitatory

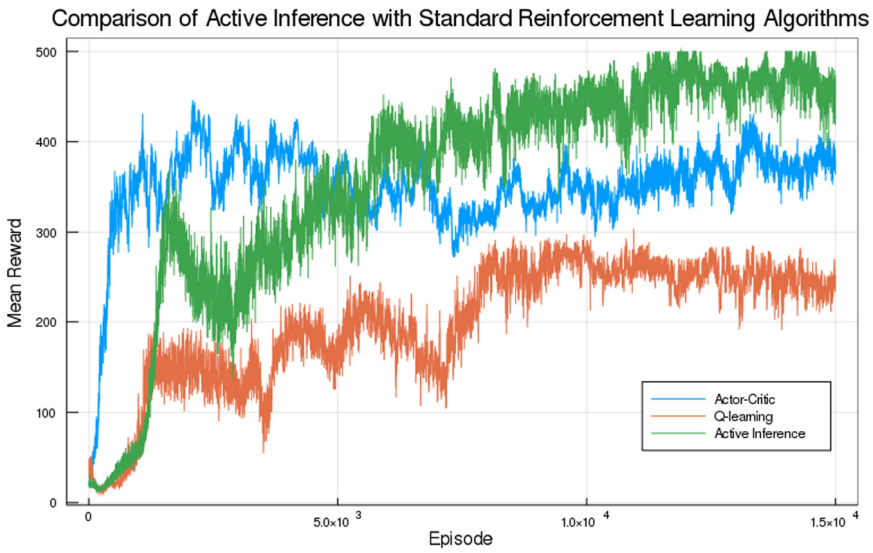


Fig. 2. Comparison of the mean reward obtained by the Active Inference agent compared to two reinforcement learning baseline algorithms — actor-critic and Q learning.

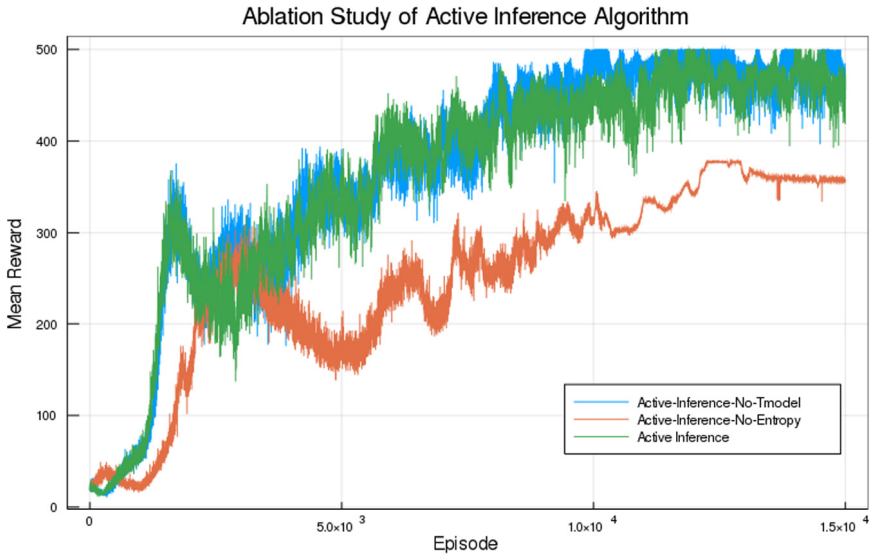


Fig. 3. We compare the full Active Inference agent (entropy regularization + transition model) with an Active Inference agent without the transition model, and without both the entropy term and the transition model.

mode. This happens long before the policy or value-networks have converged, meaning that the epistemic value ends up driving very little exploration and having very little effect overall. As a comparison, the reward magnitude of the CartPole task was +1.

We also compared the Active Inference agent to the two baseline reinforcement learning agents on two more complex tasks than the CartPole — the Acrobot and the Lunar-Lander environments from OpenAIGym. The graphs of the performance of the agents are shown in Figs. 7 and 8:

As can be seen, the Active Inference agent is competitive with baseline reinforcement leaning agents on both of these tasks. In terms of computational cost, Active Inference without the transition model has roughly the same cost as the deep-actor-critic algorithm from reinforcement learning. Adding the transition model has a greater computational cost and, for the current suite of tasks, appears to add little benefit — the random exploration ensured by the stochastic action selection and the entropy regularization appears to be sufficient exploration for solving these tasks without more complex epistemic rewards. We believe, however, that in more complex tasks with a compositional structure and longer temporal gaps between rewards and actions, then this sort of goal-directed exploration will become increasingly necessary. The Active Inference agent outperforms the two standard reinforcement learning approaches on the acrobot task. This is likely due to the entropy regularization term of the Active Inference agent driving more exploration, which is a key difficulty of this task since no rewards are obtained until the agent happens to swing up the arm to above the horizontal. It is unclear why the performance of policy gradients declines over time in this task, but it could be due to policy collapse. Active Inference underperforms policy gradients on the lunar lander tasks, but it is comparable with Q-learning and actor-critic. We believe this is due to inaccuracies and bias in using neural networks to estimate the value function, as done in actor-critic, Q-learning and our Active Inference algorithm as opposed to simply using unbiased Monte-Carlo returns, as is done by policy gradient.

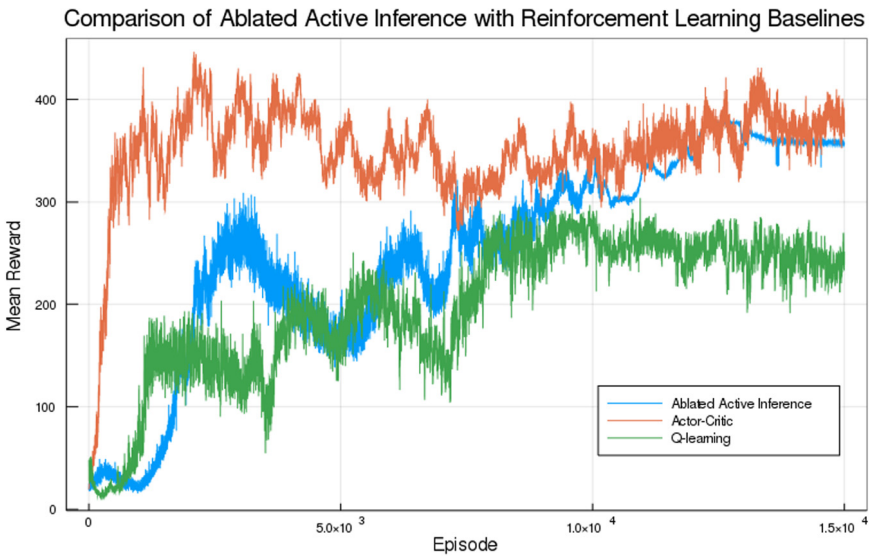
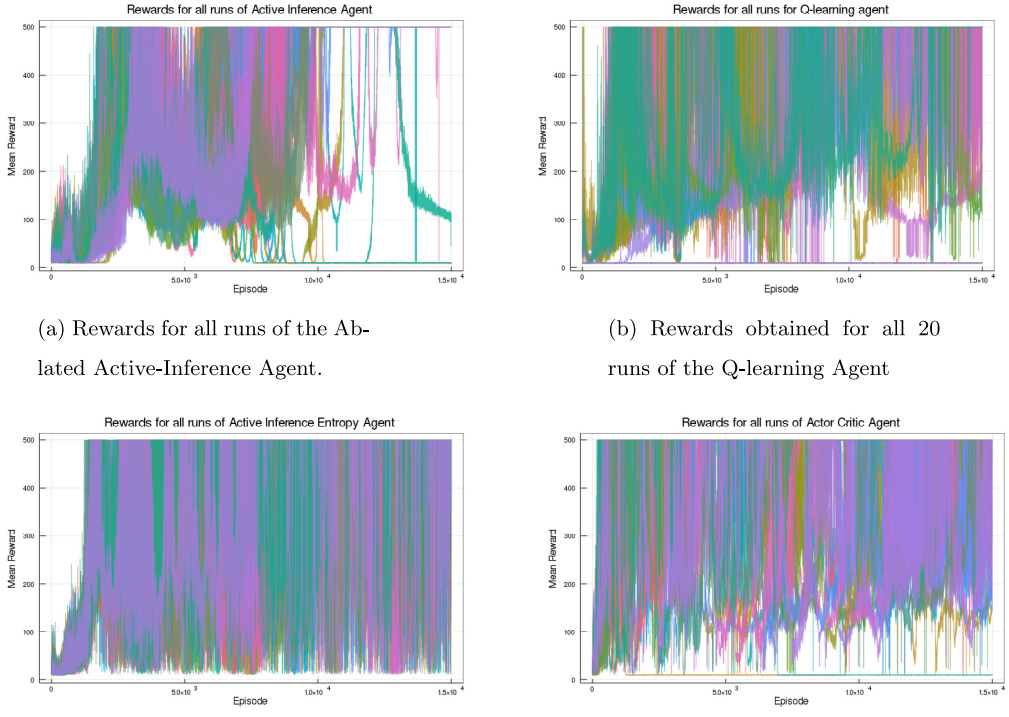


Fig. 4. Comparison of the rewards obtained by the fully ablated Active Inference agent with standard reinforcement-learning baselines of Q-learning and actor-critic.



(a) Rewards for all runs of the Ablated Active-Inference Agent.

(b) Rewards obtained for all 20 runs of the Q-learning Agent

(c) Rewards for all 20 runs of the Active-Inference Agent

(d) Rewards for all 20 runs of the Actor-Critic Agent

Fig. 5. Rewards obtained for each episode for all 20 runs of the different agents. Observe the policy collapse and bifurcations, especially of the Active Inference agent and Q-learning agent for which rewards in an episode will tend towards the optimal +200 or near 0. The entropy term in the Active Inference formulation appears to prevent policy collapse not by causing convergence to a perfect policy, but instead by preventing policy collapses becoming permanent.

6. Discussion

In this paper, we have derived a novel deep Active Inference algorithm which uses deep neural networks to approximate the key densities of the variational free energy. We have contrasted this approach with tabular Active Inference and shown that deep Active Inference is significantly more scalable to larger tasks and state–spaces. Moreover, we have shown that our algorithm is competitive, and in some cases superior, to standard baseline reinforcement learning agents on a suite of reinforcement learning benchmark tasks from OpenAIGym. While Active Inference performed worse than direct policy gradients on the Lunar-Lander task, we believe this is due to the inaccuracy of the expected-free-energy-value-function estimation network, since the policy gradient method used direct and unbiased Monte-Carlo samples of the reward rather than a bootstrapping estimator. Since the performance of Active Inference, at least in the current incarnation, is sensitive to the successful training of the EFE-network, we believe that improvements here could substantially

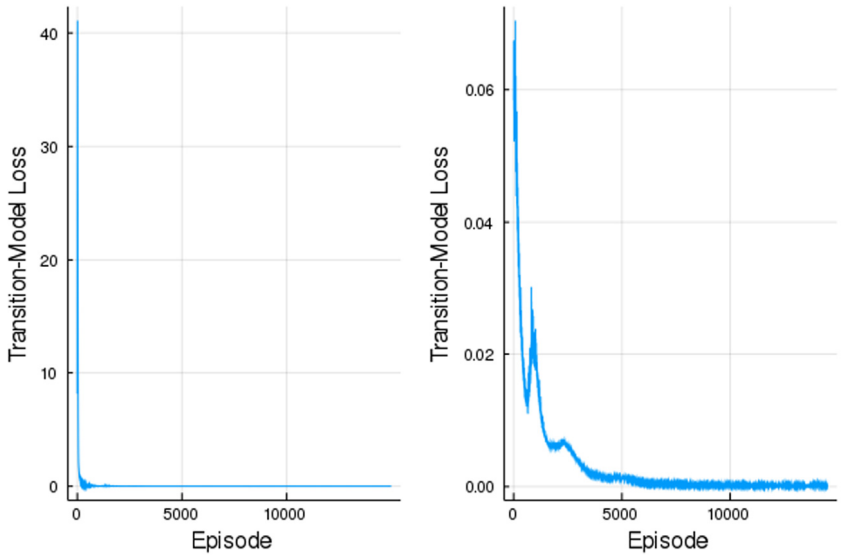


Fig. 6. Mean Transition model loss over 15 000 episodes. The right graph starts from 500 episodes into a run to show the convergence better since the initial losses are extremely high.

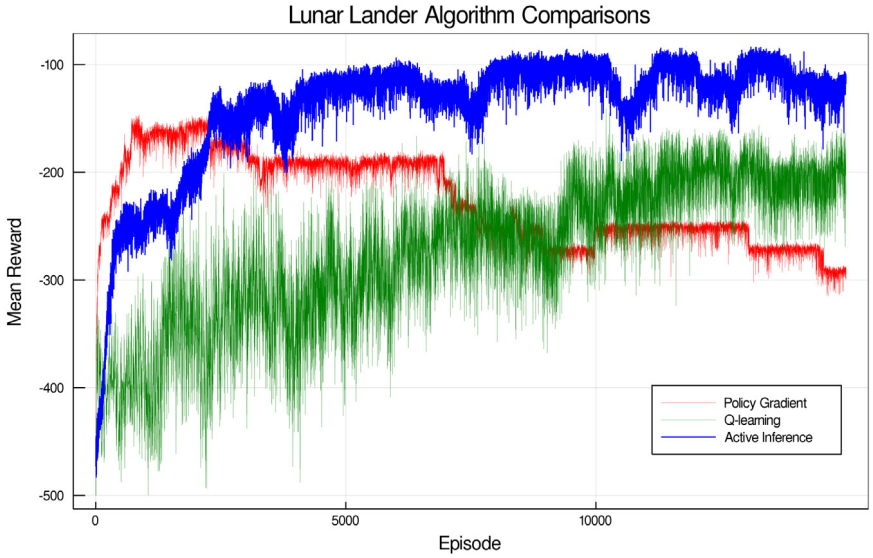


Fig. 7. Comparison of Active Inference with standard reinforcement learning algorithms on the Acrobot environment.

aid performance. Moreover, it is also possible to forego or curtail the use of the bootstrapping estimator and use the generative model to directly estimate future states and the expected-free-energy thereof, at the expense of greater computational cost.

An additional advantage of Active Inference is that due to having the transition model, it is possible to predict future trajectories and rewards N steps into the future instead of just the next time-step. These trajectories can then be sampled from and used to reduce the variance of the bootstrapping estimator, which should work as long as the transition model is accurate. The number N could perhaps even be adaptively updated given the current accuracy of the transition model and the variance of the gradient updates. This is a way of controlling the bias–variance trade-off in the estimator, since the future samples should reduce bias while increasing the variance of the estimate, and also the computational cost for each update.

Another important parameter in Active Inference and predictive processing is the precision (Feldman & Friston, 2010; Kanai, Komura, Shipp, & Friston, 2015), which in Active Inference corresponds to the inverse temperature parameter in the softmax and so controls the stochasticity of action selection. In all simulations reported above we used a fixed precision of 1. However, in tabular Active Inference the precision is often explicitly optimized against the variational free energy, and the same can be done in our deep Active Inference algorithm. In fact, the derivatives of the precision parameter can be computed automatically using automatic differentiation. Determining the impact of precision optimization on the performance of these algorithms is another worthwhile avenue for future work.

While we did not find that using the epistemic reward helped improve performance on our benchmarks, this could be due to the simplicity of the tasks we were trying to solve, for which random exploration is sufficient. It would be interesting to see if the epistemic value terms of Active Inference become much more important on more complex tasks with a hierarchical and compositional structure, and with long temporal dependencies which are exactly the sort of tasks that current random-exploration

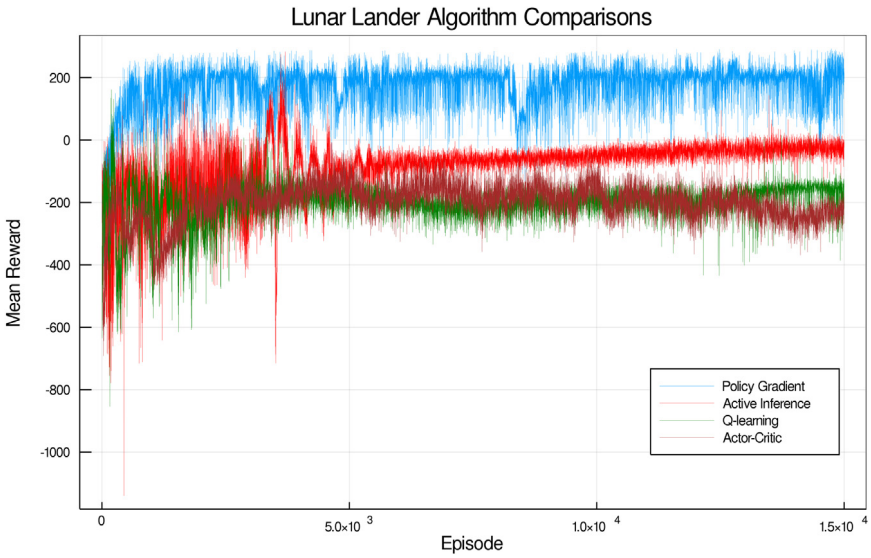


Fig. 8. Comparison of Active Inference with reinforcement learning algorithms on the Lunar-Lander environment.

reinforcement agents struggle to solve. Moreover, as suggested by Biehl et al. (2018), Active Inference can also be extended to use other intrinsic motivations, and their effects on the behaviour of Active Inference agents are still unknown.

The entropy regularization term in Active Inference proved to be extremely important, and was often the factor causing the superior performance of Active Inference to the reinforcement learning baselines. This entropy term is interesting, since it parallels similar developments in reinforcement learning, which have also found that adding an entropy term to the standard sum of discounted returns objective improves performance, policy stability and generalizability (Haarnoja, 2018; Haarnoja et al., 2017). This is of even more interest given that these algorithms can be derived from a similar variational framework which also casts control as inference (Levine, 2018). How these variational frameworks of action relate to one another is an important avenue for future work, particularly since Active Inference possesses a biologically plausible process theory which casts neuronal signalling as variational message passing. Additionally, many of the differences between Active Inference and the standard policy gradients algorithm – such as the expectation over the action, and the entropy regularization term – have been independently proposed to improve policy gradient and actor critic methods. The fact that these improvements fall naturally out of the Active Inference framework could suggest that there is deeper significance both to them and to Active Inference approaches in general. The other key differences between policy gradients and Active Inference are the optimization of the policy probabilities as opposed to the log policy probabilities, and multiplying by the log of the probabilities of the estimated values, rather than the estimated values directly. It is currently unclear precisely how important these differences are to the performance of the algorithm, and their effect on the numerical stability or conditioning of the respective algorithms, and this is also an important avenue for future research. However, the comparable performance of Active Inference to actor-critic and policy gradient approaches in our results suggest that the effect of these differences may be minor.

Our model uses deep neural networks trained using backpropagation, which is generally not thought to be biologically plausible — although there are proposed ways to implement backpropagation or approximations thereof in a biologically plausible manner (Betti, Gori, & Marra, 2018; Liao, Leibo, & Poggio, 2016; Luo, Fu, & Glass, 2017; Scellier & Bengio, 2016; Whittington & Bogacz, 2019). This means that our model is not, nor is it intended to be, a direct model of how Active Inference is implemented in the brain. Instead, this work aims towards implementing Active Inference in artificial systems, and providing a proof-of-concept that Active Inference can solve more complex tasks than those currently tackled in the literature and have the potential, ultimately, to be scaled up to the kind of complex problems the brain regularly solves. Our model shows that Active Inference can be applied in a machine learning context using deep neural networks, and can be scaled up to achieve performance comparable with reinforcement learning benchmarks on more complex tasks than any before attempted in the literature. We believe our work takes a step towards answering the question of whether Active Inference approaches can be actually used to solve the kinds of real-world problems that the brain must ultimately solve.

Although our model is not designed to be biologically plausible, and thus eschews the neuroscientifically grounded process-theory which comes with tabular active inference, it is an interesting question whether they could be combined in some manner. There is intriguing work by Whittington and Bogacz (2017) who show that the limit case of predictive coding (upon convergence) is the standard backpropagation error, and they use this to train a small artificial neural network on MNIST through predictive coding. This implies that in theory, at least, it is possible that all the work in this paper could be implemented entirely with predictive coding networks as described in Whittington et al.'s work, and would thus possess a biologically plausible process theory of predictive coding. Moreover, recent work in active inference has shown how it can be described as a variational message passing scheme on factor graphs (Friston, Parr et al., 2017; van de Laar & de Vries, 2019; Parr et al., 2019). This is important because it provides a different optimization scheme, based on variational message passing, with a biologically plausible process theory, which could be used to fit the neural networks used in this paper. There has been some early work on fitting neural networks with probabilistic and message passing models such as with EM (Lázaro, Santamaría, & Pantaleón, 2003; Ma & Ji, 1998; Ng & McLachlan, 2004) and Kalman filters (which can be derived as inference on factor graphs) (Haykin, 2004; Sum, Leung, Young, & Kan, 1999). Although these approaches have been largely superseded by optimization methods based on stochastic gradient descent, they show that in principle it is possible to train neural networks using Bayesian inference

algorithms. Thus, it may be possible to adapt and to train the neural networks in this paper with variational message passing schemes such as Parr et al. (2019) which come with a biologically plausible process theory. This would be a very interesting avenue for future work.

7. Conclusion

In sum, we have derived a novel deep Active Inference algorithm directly from the variational free energy. The full model consists of four separate neural networks approximating the terms of the variational free energy functional. We demonstrate that our approach can handle significantly more complex tasks than any previous Active Inference algorithm, and is comparable to common reinforcement learning baselines on a suite of tasks from OpenAIGym. We also highlight interesting connections between our method and policy gradient algorithms and maximum-entropy-reinforcement-learning. Finally, albeit not in a biologically plausible manner, we have shown that Active Inference algorithms can be scaled up to meet large-scale tasks, and that they provide a useful foil to the standard paradigm of reinforcement learning.

Acknowledgments

I would like to thank Mycah Banks for her invaluable comments and work proofing this manuscript. I would also like to thank the two anonymous reviewers who reviewed an earlier draft of this manuscript, and whose feedback has improved it considerably. This research was made possible by an EPSRC-funded Studentship (grant number EP/N509644/1) and the University of Edinburgh.

References

Adams, R. A., Perrinet, L. U., & Friston, K. (2012). Smooth pursuit and visual occlusion: active inference and oculomotor control in schizophrenia. *PloS one*, *7*(10), e47502.

Adams, R. A., Shipp, S., & Friston, K. J. (2013). Predictions not commands: active inference in the motor system. *Brain Structure and Function*, *218*(3), 611–643.

Asadi, K., Allen, C., Roderick, M., Mohamed, A.-r., Konidaris, G., Littman, M., et al. (2017). Mean actor critic. *stat*, *1050*, 1.

Barrett, L. F., Quigley, K. S., & Hamilton, P. (2016). An active inference theory of allostasis and interoception in depression. *Philosophical Transactions of the Royal Society, Series B (Biological Sciences)*, *371*(1708), 20160011.

Betti, A., Gori, M., & Marra, G. (2018). Backpropagation and biological plausibility. arXiv preprint arXiv:1808.06934.

Biehl, M., Guckelsberger, C., Salge, C., Smith, S. C., & Polani, D. (2018). Expanding the active inference landscape: More intrinsic motivations in the perception-action loop. *Frontiers in Neurorobotics*, *12*.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877.

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

Botvinick, M., & Toussaint, M. (2012). Planning as inference. *Trends in Cognitive Sciences*, *16*(10), 485–488.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., et al. (2016). Openai gym. arXiv preprint arXiv:1606.01540.

Brown, H., Friston, K. J., & Bestmann, S. (2011). Active inference, attention, and motor preparation. *Frontiers in Psychology*, *2*, 218.

Buckley, C. L., Kim, C. S., McGregor, S., & Seth, A. K. (2017). The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, *81*, 55–79.

Catal, O., Nauta, J., Verbelen, T., Simoens, P., & Dhoedt, B. (2019). Bayesian policy selection using active inference. arXiv preprint arXiv:1904.08149.

Cesa-Bianchi, N., Gentile, C., Lugosi, G., & Neu, G. (2017). Boltzmann exploration done right. In *Advances in neural information processing systems* (pp. 6284–6293).

Ciosek, K., & Whiteson, S. (2018). Expected policy gradients. In *Thirty-second AAAI conference on artificial intelligence*.

Clark, A. (2012). Dreaming the whole cat: Generative models, predictive processing, and the enactivist conception of perceptual experience. *Mind*, *121*(483), 753–771.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(3), 181–204.

Clark, A. (2015). *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press.

Cullen, M., Davey, B., Friston, K. J., & Moran, R. J. (2018). Active inference in openai gym: A paradigm for computational investigations into psychiatric illness. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, *3*(9), 809–818.

Daw, N. D., O'doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, *441*(7095), 876.

Degris, T., White, M., & Sutton, R. S. (2012). Off-policy actor-critic. arXiv preprint arXiv:1205.4839.

Deisenroth, M., & Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 465–472).

Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.

Doya, K., Ishii, S., Pouget, A., & Rao, R. P. (2007). *Bayesian brain: Probabilistic approaches to neural coding*. MIT press.

Feldman, H., & Friston, K. (2010). Attention, uncertainty, and free-energy. *Frontiers in Human Neuroscience*, *4*, 215.

Fellows, M., Mahajan, A., Rudner, T. G., & Whiteson, S. (2018). VIREL: A variational inference framework for reinforcement learning. arXiv preprint arXiv:1811.01132.

FitzGerald, T. H., Schwartenbeck, P., Moutoussis, M., Dolan, R. J., & Friston, K. (2015). Active inference, evidence accumulation, and the urn task. *Neural Computation*, *27*(2), 306–328.

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

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

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

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

Friston, K. (2011). What is optimal about motor control? *Neuron*, *72*(3), 488–498.

Friston, K. (2012). The history of the future of the Bayesian brain. *NeuroImage*, *62*(2), 1230–1233.

Friston, K. (2019). A free energy principle for a particular physics. arXiv preprint arXiv:1906.10184.

Friston, K., & Ao, P. (2012). Free energy, value, and attractors. *Computational and Mathematical Methods in Medicine*, *2012*.

Friston, K. J., Daunizeau, J., & Kiebel, S. J. (2009). Reinforcement learning or active inference? *PloS one*, *4*(7), e6421.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G., et al. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, *68*, 862–879.

Friston, K., Kilner, J., & Harrison, L. (2006). A free energy principle for the brain. *Journal de Physiologie (Paris)*, *100*(1–3), 70–87.

Friston, K. J., Lin, M., Frith, C. D., Pezzulo, G., Hobson, J. A., & Ondobaka, S. (2017). Active inference, curiosity and insight. *Neural Computation*, *29*(10), 2633–2683.

Friston, K. J., Parr, T., & de Vries, B. (2017). The graphical brain: belief propagation and active inference. *Network Neuroscience*, *1*(4), 381–414.

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

Friston, K. J., Rosch, R., Parr, T., Price, C., & Bowman, H. (2018). Deep temporal models and active inference. *Neuroscience & Biobehavioral Reviews*, *90*, 486–501.

Friston, K., Samothrakis, S., & Montague, R. (2012). Active inference and agency: optimal control without cost functions. *Biological Cybernetics*, *106*(8–9), 523–541.

Friston, K., Schwartenbeck, P., FitzGerald, T., Moutoussis, M., Behrens, T., & Dolan, R. J. (2013). The anatomy of choice: active inference and agency. *Frontiers in Human Neuroscience*, *7*, 598.

Friston, K., Schwartenbeck, P., FitzGerald, T., Moutoussis, M., Behrens, T., & Dolan, R. J. (2014). The anatomy of choice: dopamine and decision-making. *Philosophical Transactions of the Royal Society, Series B (Biological Sciences)*, *369*(1655), 20130481.

Friston, K. J., & Stephan, K. E. (2007). Free-energy and the brain. *Synthese*, *159*(3), 417–458.

Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. arXiv preprint arXiv:1802.09477.

Gershman, S. J. (2018a). Deconstructing the human algorithms for exploration. *Cognition*, *173*, 34–42.

Gershman, S. J. (2018b). Uncertainty and exploration. (p. 265504). bioRxiv.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Ha, D., & Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. In *Advances in neural information processing systems* (pp. 2450–2462).

Haarnoja, T. (2018). *Acquiring diverse robot skills via maximum entropy deep reinforcement learning* (Ph.D. thesis), UC Berkeley.

Haarnoja, T., Tang, H., Abbeel, P., & Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th international conference on machine learning-Volume 70* (pp. 1352–1361). JMLR. org.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290.

Haykin, S. (2004). *Kalman filtering and neural networks, Vol. 47*. John Wiley & Sons.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research (JMLR)*, *14*(1), 1303–1347.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., & Abbeel, P. (2016a). Curiosity-driven exploration in deep reinforcement learning via bayesian neural networks. arXiv preprint arXiv:1605.09674.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., & Abbeel, P. (2016b). Vime: Variational information maximizing exploration. In *Advances in neural information processing systems* (pp. 1109–1117).

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kanai, R., Komura, Y., Shipp, S., & Friston, K. (2015). Cerebral hierarchies: predictive processing, precision and the pulvinar. *Philosophical Transactions of the Royal Society, Series B (Biological Sciences)*, *370*(1668), 20140169.

Karl, F. (2012). A free energy principle for biological systems. *Entropy*, *14*(11), 2100–2121.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Knill, D. C., & Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, *27*(12), 712–719.

van de Laar, T. W., & de Vries, B. (2019). Simulating active inference processes by message passing. *Frontiers in Robotics and AI*, *6*(20).

Lázaro, M., Santamarıa, I., & Pantaleón, C. (2003). A new EM-based training algorithm for RBF networks. *Neural Networks*, *16*(1), 69–77.

Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review. arXiv preprint arXiv:1805.00909.

Liao, Q., Leibo, J. Z., & Poggio, T. (2016). How important is weight symmetry in backpropagation? In *Thirtieth AAAI conference on artificial intelligence*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Luo, H., Fu, J., & Glass, J. (2017). Adaptive bidirectional backpropagation: Towards biologically plausible error signal transmission in neural networks. arXiv preprint arXiv:1702.07097.

Ma, S., & Ji, C. (1998). Fast training of recurrent networks based on the EM algorithm. *IEEE Transactions on Neural Networks*, *9*(1), 11–26.

Millidge, B. (2019). Combining active inference and hierarchical predictive coding: A tutorial introduction and case study. PsyArXiv.

Mirza, M. B., Adams, R. A., Mathys, C. D., & Friston, K. J. (2016). Scene construction, visual foraging, and active inference. *Frontiers in Computational Neuroscience*, *10*, 56.

Mirza, M. B., Adams, R. A., Parr, T., & Friston, K. (2019). Impulsivity and active inference. *Journal of Cognitive Neuroscience*, *31*(2), 202–220.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

Mohamed, S., & Rezende, D. J. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems* (pp. 2125–2133).

Moulin, C., & Souchay, C. (2015). An active inference and epistemic value view of metacognition. *Cognitive Neuroscience*, *6*(4), 221–222.

Ng, S.-K., & McLachlan, G. J. (2004). Using the EM algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification. *IEEE Transactions on Neural Networks*, *15*(3), 738–749.

Oudeyer, P.-Y., & Kaplan, F. (2009). What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, *1*, 6.

Parr, T., & Friston, K. J. (2017). Uncertainty, epistemics and active inference. *Journal of the Royal Society Interface*, *14*(136), 20170376.

Parr, T., & Friston, K. J. (2018). Generalised free energy and active inference: can the future cause the past?. (p. 304782). BioRxiv.

Parr, T., & Friston, K. J. (2018a). Active inference and the anatomy of oculomotion. *Neuropsychologia*, *111*, 334–343.

Parr, T., & Friston, K. J. (2018c). The anatomy of inference: Generative models and brain structure. *Frontiers in Computational Neuroscience*, *12*.

Parr, T., Markovic, D., Kiebel, S. J., & Friston, K. J. (2019). Neuronal message passing using Mean-field, Bethe, and Marginal approximations. *Scientific Reports*, *9*(1), 1889.

Pezzulo, G., Cartoni, E., Rigoli, F., Pio-Lopez, L., & Friston, K. (2016). Active inference, epistemic value, and vicarious trial and error. *Learning & Memory*, *23*(7), 322–338.

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

Rawlik, K., Toussaint, M., & Vijayakumar, S. (2010). Approximate inference and stochastic optimal control. arXiv preprint arXiv:1009.3958.

Rawlik, K., Toussaint, M., & Vijayakumar, S. (2013). On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-third international joint conference on artificial intelligence*.

Scellier, B., & Bengio, Y. (2016). Towards a biologically plausible backprop. arXiv preprint arXiv:1602.05179.

Schwartenbeck, P., Passecker, J., Hauser, T. U., FitzGerald, T. H., Kronbichler, M., & Friston, K. J. (2019). Computational mechanisms of curiosity and goal-directed exploration. *eLife*, *8*, e41703.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, *550*(7676), 354.

Smith, R., Schwartenbeck, P., Parr, T., & Friston, K. J. (2019). An active inference approach to modeling concept learning. (p. 633677). bioRxiv.

Song, R., Lewis, F. L., Wei, Q., & Zhang, H. (2015). Off-policy actor-critic structure for optimal control of unknown systems with disturbances. *IEEE Transactions on Cybernetics*, *46*(5), 1041–1050.

Stadie, B. C., Levine, S., & Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. arXiv preprint arXiv:1507.00814.

Sum, J., Leung, C.-S., Young, G. H., & Kan, W.-K. (1999). On the Kalman filtering method in neural network training and pruning. *IEEE Transactions on Neural Networks*, *10*(1), 161–166.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning, Vol. 135*. Cambridge: MIT press.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).

Ueltzhöffer, K. (2018). Deep active inference. *Biological Cybernetics*, *112*(6), 547–573.

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*.

Wayne, G., Hung, C.-C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., et al. (2018). Unsupervised predictive memory in a goal-directed agent. arXiv preprint arXiv:1803.10760.

Whittington, J. C., & Bogacz, R. (2017). An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural Computation*, *29*(5), 1229–1262.

Whittington, J. C., & Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in Cognitive Sciences*.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Aaai, Vol. 8*, Chicago, IL, USA (pp. 1433–1438).