Autonomous Agents and Multi-Agent Systems (2025) 39:3
https://doi.org/10.1007/s10458-024-09683-4

Information gathering in POMDPs using active inference

Erwin Walraven[1] · **Joris Sijs**[2] · **Gertjan J. Burghouts**[1]

Accepted: 31 October 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
Gathering information about the environment state is the main goal in several planning tasks for autonomous agents, such as surveillance, inspection and tracking of objects. Such planning tasks are typically modeled using a Partially Observable Markov Decision Process (POMDP), and in the literature several approaches have emerged to consider information gathering during planning and execution. Similar developments can be seen in the field of active inference, which focuses on active information collection in order to be able to reach a goal. Both fields use POMDPs to model the environment, but the underlying principles for action selection are different. In this paper we create a bridge between both research fields by discussing how they relate to each other and how they can be used for information gathering. Our contribution is a tailored approach to model information gathering tasks directly in the active inference framework. A series of experiments demonstrates that our approach enables agents to gather information about the environment state. As a result, active inference becomes an alternative to common POMDP approaches for information gathering, which opens the door towards more cross cutting research at the intersection of both fields. This is advantageous, because recent advancements in POMDP solvers may be used to accelerate active inference, and the principled active inference framework may be used to model POMDP agents that operate in a neurobiologically plausible fashion.

Keywords Planning under uncertainty · POMDP · Information gathering · Active inference

1 Introduction

In many planning domains autonomous agents take actions to eventually reach a predefined goal. For example, in robotics agents need to navigate towards a specific location [1], and in a congested power grid electric vehicles need to decide when to charge in

✉ Erwin Walraven
erwin.walraven@tno.nl

Joris Sijs
j.sijs@tudelft.nl

Gertjan J. Burghouts
gertjan.burghouts@tno.nl

[1] Netherlands Organisation for Applied Scientific Research, The Hague, The Netherlands

[2] Delft University of Technology, Delft, The Netherlands

Published online: 07 November 2024

order to reach a full battery before departure [2]. In the definition of the planning problem such goals are typically expressed by defining a reward function which the agent aims to maximize by taking actions [3]. In order to be able to maximize reward, it may be necessary for the agent to gather information in the environment, as a means to reach the goal. For example, a robot can first execute sensory actions to find out which exit it can use in a room, before actually navigating towards the exit. In the aforementioned domains planning problems are typically formalized as a Partially Observable Markov Decision Process [3], also known as POMDP, which are solved using e.g. value iteration algorithms [4] or Monte Carlo Tree Search [5].

In other types of domains information gathering is not just a means to reach a goal. Instead, gathering information about the environment state can be the primary goal of the task. Examples include surveillance tasks [6], inspections in industrial areas [7], and tracking of moving objects [8]. We refer to such tasks as information gathering tasks. In the literature there are several approaches which address such problems, such as information gain planning, active sensing, entropy minimization and active perception [9]. Even though multiple different names have appeared in the literature, all these approaches address similar planning problems in which agents actively plan and execute actions to collect knowledge about the environment state.

The ability to plan ahead for the purpose of reaching goals has also been studied in the field of neuroscience, in which the active inference formalism integrates acting, perception and planning in one integrated framework [10]. In active inference the agents also actively collect information by taking actions, in order to be able to reach a goal. For example, active inference has been used to navigate to a target location in a maze task [10]. The framework also relies on the POMDP formalism, but for action selection it uses other principles compared to researchers that work in planning. In the literature there are relatively few connections between the fields, and this raises the question of how active inference relates to concepts used by POMDP planning researchers, and whether it may be used for tasks in which information about the environment state needs to be gathered.

In this paper we bring planning research on POMDPs and active inference closer together, specifically for such information gathering tasks. In particular, we extend an important measurement from active inference called expected free energy by adding an additional term, which encourages information gathering behavior in active inference, and we show how it can be integrated in commonly used online planning algorithms for POMDPs. Our main contribution is a new method to express information gathering tasks directly in active inference, which means that active inference becomes an alternative to existing information gathering approaches for POMDPs. Another important result is that our work creates a more tangible connection between POMDP planning research on information gathering and active inference, potentially opening the door towards more crosscutting research between both fields.

The structure of our paper is as follows. In Sect. 2 we start with background information on POMDP planning and information gathering tasks. Section 3 introduces active inference, and makes the connection to the POMDP concepts that we introduced earlier. Next, we introduce an online planning algorithm for active inference in Sect. 4, which we extend to adopt our tailored expected free energy term for information gathering with POMDPs. This algorithm is evaluated and compared with existing POMDP approaches in Sect. 5. We describe related work in Sect. 6, and we conclude by discussing our main findings and future work in Sect. 7.

2 Planning for information gathering tasks

In this section we introduce planning under uncertainty using Partially Observable Markov Decision Processes (POMDP), and we recapitulate existing approaches for information gathering in POMDPs. It is important to mention that the field of active inference and literature on planning both use the POMDP formalism to model an environment. In this section we specifically focus on planning literature which focuses on algorithms to perform information gathering in environments modeled as POMDP.

2.1 Partially observable markov decision processes

Planning problems involving uncertainty can be modeled using a POMDP [3], which consists of states $s \in S$, actions $a \in A$ and observations $o \in O$. When executing action $a \in A$ in state $s \in S$, the state of the environment transitions to $s' \in S$ based on the distribution $P(s' \mid s, a)$. The agent receives reward $R(s, a)$ for executing this action. The state s' of the environment is partially observable, which means that the agent cannot observe it directly. Instead, it receives an observation $o \in O$ based on distribution $P(o \mid a, s')$. The agent maintains a belief over states, denoted by b, which can be updated by using Bayes' rule. For a given belief b we let $b(s)$ denote the belief that the true state is s. POMDP states can be factored, such that a state s is defined by multiple state variables. We let s^j denote state variable j of state s. We consider a finite-horizon setting involving T timesteps, and the agent wants to maximize the total expected reward: $E[\sum_{t=1}^{T} R_t]$, in which R_t denotes the reward received at time t. POMDPs can be solved optimally using incremental pruning [11, 12]. However, in practice approximate and online approaches are used such as approximate value iteration [4, 13] and Monte Carlo Tree Search [5].

2.2 Information gathering in POMDPs

We focus on information gathering tasks, which is the execution of actions to gather information about the environment state. Intuitively, agents start with an initial belief b_0 in which there is uncertainty regarding the actual environment state, and the agent aims to execute actions in such a way that the uncertainty is reduced. If factored state representations are used, it is also possible to express that uncertainty with respect to a specific state variable should be reduced. For example, in an inspection task the agent may be interested in collecting information about a specific gas meter in an environment, and in that case it is not necessary to collect information about other parts of the environment.

Multiple different names are used in the literature to refer to the type of information gathering tasks that we consider, such as active perception [9] and active sensing [14]. A common characteristic is that agents have the incentive to actively execute actions to influence the uncertainty regarding the true environment state. In a surveillance task this can consist of turning cameras in such a way that uncertainty remains low, and an inspection robot may decide to bring a flashlight if it enables the robot to effectively perform uncertainty-reducing inspections later during the inspection mission.

Information gathering can be modeled by rewarding the agent directly for collecting information [15]. This requires a belief-based reward function $\rho(b, a)$ which quantifies the amount uncertainty in belief b. As a metric for uncertainty the (negative) entropy can be used:

$$\rho(b, a) = \sum_{s \in S} b(s) \cdot \log(b(s)), \tag{1}$$

such that maximizing the belief-based reward function corresponds to choosing actions which reduce uncertainty. In a similar fashion a belief-based reward function $\rho(b, a, b')$ can be constructed, which expresses the reduction of uncertainty when transitioning from b to b' through action a. In contrast to Eq. 1 such a function rewards the uncertainty reduction based on two consecutive beliefs b and b', rather than using the uncertainty in a belief b directly:

$$\rho(b, a, b') = \left(- \sum_{s \in S} b(s) \cdot \log(b(s)) \right) - \left(- \sum_{s \in S} b'(s) \cdot \log(b'(s)) \right). \tag{2}$$

It is also an option to assign a large scalar reward to beliefs with low entropy, rather than using the entropy directly.

Solving POMDPs for information gathering tasks is not straightforward if the reward signal is dependent on the belief rather than being dependent on the environment state. Many traditional planning algorithms for POMDPs have been designed for a reward function $R(s, a)$. For example, point-based value iteration algorithms [4, 13] use a set of vectors to construct a piecewise linear and convex (PWLC) value function, which requires a state-based reward function. Similarly, online algorithms such as Partially Observable Monte Carlo Planning (POMCP) [5] do not support a belief-based reward function.

In the literature two lines of work can be distinguished to deal with belief-based rewards during planning. The first line of work consists of approaches which aim to formulate the planning problem in such a way that belief-based rewards can be integrated in standard POMDP algorithms which require a PWLC value function. The ρ-POMDP framework [15] takes belief-based rewards, and shows how it can be expressed as a set of vectors while the error introduced by this approximation remains bounded. The POMDP-IR framework [16] also aims to stay within the classic POMDP framework by introducing additional actions which reward the agent for predicting the true state correctly. Intuitively, this forces the agent to reduce uncertainty, and it comes with the advantage that standard POMDP algorithms can be used. The second line of work directly plans with belief-based rewards, rather than integrating it in POMDP algorithms for state-based rewards. For example, ρ-POMCP [17] extends the POMCP algorithm in such a way that belief-based rewards can be used in the search tree that is constructed.

Until now we have considered information gathering from the viewpoint of POMDP planning. Acting, perception and information gathering have also been studied from a neuroscientific point of view, known as active inference [10]. Although several connections between both lines of work can be identified, work at the intersection of both fields seems rare in the AI planning literature. The next section introduces active inference, and it describes how it relates to information gathering.

3 Active inference

Active inference is a computational formalism from the field of neuroscience which models acting, perception and planning in an integrated framework [10]. This section introduces active inference in such a way that it connects to known concepts from the planning literature, bridging the gap between both fields. We start with an intuitive introduction to the mathematical concepts, followed by a translation to common POMDP notation. Finally, we provide a discussion which intuitively explains how active inference agents navigate towards their goal and how this can be interpreted from the viewpoint of POMDP planning.

We consider a model-based planning task in which the environment is modeled as a POMDP. In active inference the goal of an agent is expressed by defining a preference for receiving specific observation signals, which is modeled as a prior over POMDP observations. Policies in planning research are typically dependent on the state or on the belief, defining actions to be executed or probability distributions over actions. In active inference it is common to define policies as a deterministic action sequence, and during execution a posterior over policies is computed, which also defines a probability distribution over actions. The theory behind active inference states that agents naturally want to minimize the information-theoretic surprise while acting in an environment [10]. Agents can plan their actions by minimizing a single quantity that is known as the expected free energy. Before providing a mathematical introduction to expected free energy, we first provide an intuitive explanation. The expected free energy effectively balances exploration and exploitation during action execution. It is a single quantity that can be used to evaluate a policy π for a given POMDP, and it can be divided into a goal-seeking (pragmatic) component and an uncertainty-resolving (epistemic) component. The goal-seeking component represents the value of the policy based on preferred observations. Specifically, it measures to what extent the observations that are expected when executing π correspond to observations that are preferred. The uncertainty-resolving component, also known as ambiguity-minimizing component, represents the uncertainty in states and observations that are expected under the execution of π. Minimizing expected free energy naturally steers the agent towards its goal, while taking uncertainty-reducing actions in case this is relevant for reaching the goal. The remainder of this section provides more details on expected free energy and its interpretation. However, it is important to note that our paper does not aim to provide an extensive treatment of the underlying theories from neuroscience, for which we refer to seminal work by Friston [18]. Additional background on the role of active inference in robotics can be found in work by Pezzato et al. [19] and Da Costa et al. [20]. Throughout this paper we use active inference only for action selection. For belief updates and estimation we use a particle representation, which we further discuss in Sect. 4.

3.1 Computing expected free energy for POMDPs

The expected free energy of a policy π consists of the total free energy that can be expected while executing π until the planning horizon. In order to find policies which minimize expected free energy, it is convenient to use an expression which defines the instantaneous expected free energy at a specific moment in time. For a given belief b the instantaneous expected free energy $G(b)$ can be computed as follows [21–23]:

$$G(b) = (A \cdot b) \cdot (\ln(A \cdot b) - \ln(\bar{o})) + H \cdot b, \tag{3}$$

in which A is a matrix with $|O|$ rows and $|S|$ columns. In this equation, it is assumed that b is a vector containing an entry for each state. The matrix element $A_{i,j}$ denotes $P(o_i \mid s_j)$.

Note that the conditional probability is not dependent on the executed action a before reaching s_j, which is common in POMDP notation. However, it is actually equivalent to $P(o_i \mid a, s_j)$ if the last executed action is part of the state description, which can always be done. H is a vector containing the entropy of the expected observations in each state: $H = -\operatorname{diag}(A^T \cdot \ln(A))$, in which $\operatorname{diag}(\cdot)$ takes the elements from the diagonal of the matrix. The vector $\bar{o}$ contains $|O|$ elements and defines a prior preference for each observation. This vector can be seen as a set of weights assigned to the observations to define a goal. For example, if the goal of the agent is to see a specific observation after executing an action, then the corresponding entry in $\bar{o}$ can be set to a value close to one. It is important to note that the natural logarithm is used element-wise when applying it to a vector or matrix. To prevent numerical issues it is important to make sure that the prior values are not zero, because that would make the terms with the logarithm diverge.

Before we discuss the interpretation of expected free energy from the viewpoint of POMDP planning in the next section, it is convenient to rewrite Eq. 3 to conventional POMDP notation. Based on the definition of A we can define element i of H as follows:

$$H_i = \sum_{o \in O} -1 \cdot P(o \mid s_i) \cdot \ln(P(o \mid s_i)). \tag{4}$$

The expression $A \cdot b$ can be written as:

$$\begin{aligned} A \cdot b &= \left[\sum_{s \in S} P(o_1 \mid s) b(s), \ldots, \sum_{s \in S} P(o_{|O|} \mid s) b(s) \right]^T \\ &= \left[P(o_1 \mid b), \ldots, P(o_{|O|} \mid b) \right]^T. \end{aligned} \tag{5}$$

We use this expression to rewrite $\ln(A \cdot b) - \ln(\bar{o})$:

$$\ln(A \cdot b) - \ln(\bar{o}) = \begin{bmatrix} \ln(P(o_1 \mid b)) - \ln(\bar{o}_1) \\ \vdots \\ \ln(P(o_{|O|} \mid b)) - \ln(\bar{o}_{|O|}) \end{bmatrix}, \tag{6}$$

in which $\bar{o}_i$ denotes element i of $\bar{o}$, representing the prior preference for an observation. Now we can put Eqs. 4 until 6 together to rewrite Eq. 3:

$$\begin{aligned} G(b) &= (A \cdot b) \cdot (\ln(A \cdot b) - \ln(\bar{o})) + H \cdot b \\ &= \sum_{o \in O} [P(o \mid b)(\ln(P(o \mid b)) - \ln(\bar{o}_o))] \\ &\quad + \sum_{s \in S} b(s) \sum_{o \in O} -1 \cdot P(o \mid s) \cdot \ln(P(o \mid s)). \end{aligned} \tag{7}$$

In the first term, we take the sum over observations, which follows from the fact that Eq. 5 and 6 contain $|O|$ vector elements. For this reason we let $\bar{o}_o$ denote the prior preference for observation $o \in O$.

In the remainder of this paper, we use the rewritten equation to help us interpret the expected free energy, and to integrate the term in a planning algorithm. This is easier to understand than using Eq. 3 directly.

3.2 Interpretation of expected free energy

Given the derivation of the expected free energy equation from the previous section, we can now rewrite it in such a way that we can explain its interpretation from a POMDP planning point of view, strengthening the connection between POMDP planning research on information gathering and active inference. In the field of neuroscience the expected free energy has been proposed to effectively balance exploration and exploitation. However, based on Eq. 7 it is not immediately apparent how minimizing expected free energy induces this behavior. It becomes more intuitive when rewriting the equation as follows:

$$\begin{aligned} G(b) = &\sum_{o \in O} P(o \mid b) \ln(P(o \mid b)) \\ &- \sum_{o \in O} P(o \mid b) \ln(\bar{o}_o) \\ &+ \sum_{s \in S} b(s) \sum_{o \in O} -1 \cdot P(o \mid s) \cdot \ln(P(o \mid s)). \end{aligned} \tag{8}$$

The first term can be interpreted as the negative entropy of the marginal belief over observations o. Since the expected free energy is minimized, it actually maximizes this entropy, which encourages exploration. The second term can be interpreted as the expected value, multiplied by -1. If the expected free energy is minimized, it means that it plans towards beliefs b in which the expected observations o are also preferred based on the prior $\bar{o}_o$. The third term can be seen as the expected entropy of the observations, which steers towards policies that result in unambiguous observations during execution. Minimizing this term may also implicitly lead to behavior where the agent seeks for informative observations to diminish uncertainty regarding the state s, because in order to minimize ambiguity it may be necessary to have low uncertainty with respect to the actual state of the environment.

Our discussion shows that expected free energy consists of several terms with an intuitive interpretation, but it can also be seen that it is mainly an expression that is dependent on the observation signals o that the agent perceives while taking actions in an environment. For information gathering tasks we would like to express that uncertainty about the environment is reduced. Although some terms in the expected free energy are correlated with state uncertainty reduction, we cannot express information gathering directly when defining an active inference problem, because it only allows us to define preferred observations. In the next sections we present techniques which can be used to overcome this limitation, and we make the comparison with existing information gathering approaches for POMDPs.

4 Information gathering in active inference

In this section, we present an active inference approach which can be used for information gathering tasks that are modeled as a POMDP. First we introduce an online planning algorithm for active inference in Sect. 4.1, which is based on Monte Carlo Tree Search. The use of such an online algorithm is an important step, because common approaches for active inference rely on full policy enumeration, which quickly becomes intractable. Our main contribution is a derivation of an adjusted equation for

the expected free energy in Sect. 4.2, which enables us to solve information gathering tasks based on active inference, and the adjusted equation easily integrates in the online planning algorithm.

4.1 Online planning algorithm for active inference

In the active inference literature, planning typically consists of iterating over all possible policies, and selecting the policy with the smallest expected free energy. In this paper, we do not rely on full policy enumeration, because it quickly becomes intractable. Instead, we present a tailored variant of Monte Carlo Tree Search that integrates well with the active inference concepts, and thereby it eliminates the need to do full policy enumeration. We present this algorithm in detail because it is an important prerequisite for understanding our derivation in the next section.

For planning based on expected free energy, it is crucial that the planning algorithm supports belief-based rewards. Existing tree search algorithms such as POMCP [5] cannot be applied because its rewards are based on states. Our variant of Monte Carlo Tree Search plans actions based on POMDP belief states rather than MDP states, in such a way that rewards based on beliefs can be used during the search. The tree structure that we use is shown in Fig. 1. The root of the tree corresponds to a belief b, which is represented by a particle set containing k states. In the figure four particles are visualized. The transitions from the root node to the square nodes correspond to actions a that can be taken, and the transitions from a square node to the circle nodes correspond to observations o that can be received after taking an action. After executing an action a and receiving an observation o, the resulting belief b' is also represented by a particle set. The transition from b to b' after taking action a can be computed using a sampling procedure. First, a state s, successor state s' and observation o are sampled based on b. Next, we use a weighted particle filter to estimate the belief $b_a^o = b'$. These steps are a common procedure in POMDP algorithms which require transitions from beliefs to successor beliefs [17, 24, 25].

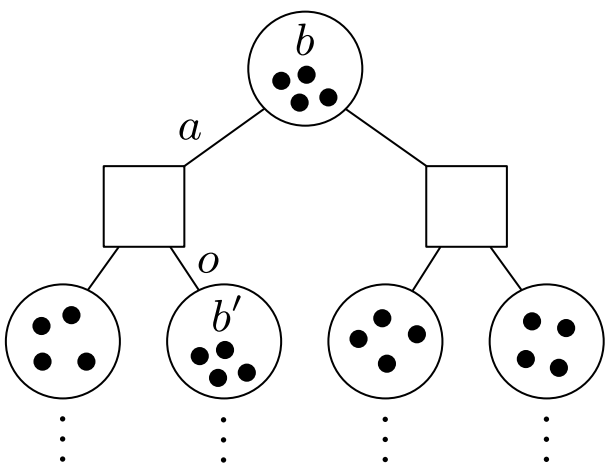


Fig. 1 Tree structure for MCTS

Algorithm 1: MCTS for active inference

```
1  Procedure plan(b)
2      for i = 1, ..., n do
3          simulate(b, T)
4      end
5      return arg max_{a∈A} Q(ba)
1  Procedure particleTransition(b, a)
2      s ← sample state from b
3      s' ← sample state based on P(s' | s, a)
4      o ← sample observation based on P(o | s')
5      i ← 0
6      for s ∈ b do
7          s' ← sample state based on P(s' | s, a)
8          w_i ← P(o | s'), i ← i + 1
9      end
10     normalize weights w_i such that Σ_i w_i = 1
11     b' ← sample |b| states based on weights w_i
12     return b', o
1  Procedure simulate(b, d)
2      if d = 0 then
3          return 0
4      end
5      if C(b) = 0 then
6          create child node for each a ∈ A
7          C(b) ← |A|
8          return rollout(b, d)
9      end
10     a ← getBestAction(b)
11     b', o ← particleTransition(b, a)
12     Y ← −1 · G(b) + simulate(b', d − 1)
13     N(b) ← N(b) + 1, N(ba) ← N(ba) + 1
14     Q(ba) ← Q(ba) + ((Y − Q(ba)) / N(ba))
15     return Y
```

A full description of our algorithm is shown in Algorithm 1. In the literature, Monte Carlo Tree Search is commonly used to maximize expected reward, while we aim to minimize expected free energy. In order to keep notation consistent with existing literature, we describe our algorithm in such a way that it maximizes the reward function $-1 \cdot G(b)$ (i.e., minimize the expected free energy term $G(b)$). Our algorithm is similar to several other Monte Carlo planning algorithms for POMDPs [5, 17, 24, 26], and therefore we only provide a high-level discussion. The `simulate` procedure is executed n times starting from the root of the search tree, and it recursively executes actions until it reaches the planning horizon. The free energy term $G(b)$ is integrated on line 12. $N(b)$ denotes how many times the node corresponding to belief b has been visited, and $Q(ba)$ denotes the current estimate of expected reward (i.e., free energy multiplied by -1) when executing a in belief b. $C(b)$ denotes the number of children of belief node b, which is 0 by default. The procedure `particleTransition` implements a transition from belief b to b' using a

weighted particle filter. Since beliefs are represented by particles, the belief b can be used as a set based on which states $s \in b$ are enumerated. The procedure `getBestAction` uses an exploration strategy such as Upper Confidence Bounds (UCB) [27] to choose an action for a given belief, and the `rollout` procedure uses a random rollout policy from b until the planning horizon and returns the total reward (i.e., free energy) obtained.

During the search process, the belief transition implemented by the `particleTransition` procedure is the most expensive operation from a computational point of view. However, for a given belief b, action a and observation o, the sampling process on lines 6-9 needs to be performed only once. Therefore, a caching mechanism is implemented to prevent redundant computations.

It is important to note that the use of Monte Carlo Tree Search for active inference has also been proposed in the literature [23, 28]. An important difference is that we consider explicit belief transitions, and we branch based on actions and observations. For a more elaborate discussion we refer to the related work section.

4.2 Modeling information gathering tasks

In information gathering tasks, the agent needs to perform actions to reduce uncertainty with respect to one or more state variables s^j. For example, if s^j represents whether a pipe is leaking or not, then the agent needs to execute actions leading to beliefs in which there is low uncertainty about the existence of the leak. As discussed in Sect. 3.2, in its standard formulation expected free energy does not capture state uncertainty, which means that we cannot use Algorithm 1 directly for information gathering tasks.

In the remainder of this section, we derive a modification of Eq. 7 which enables us to include state uncertainty within the active inference framework. Our derivation consists of two parts. First, we provide a visual illustration which shows how state uncertainty can be reduced by using additional actions and preferred observations. The sole purpose of this illustration and discussion is to describe the reasoning and intuition behind a change in the expected free energy, which eventually provides an incentive for the agent to reduce uncertainty. Second, we discuss and prove that there is an equivalent expected free energy term that we can add to Eq. 7 without expanding the POMDP model with additional actions and observations.

We start with a specific example to explain and visualize our idea, which we later generalize. We consider a binary state variable s^j for which the agent wants to reduce uncertainty. In our example it is assumed that the agent is currently in a belief b in which $P(s^j = 1 \mid b) = 0.4$. Our line of reasoning holds for any value of $P(s^j = 1 \mid b)$, but for clarity we take 0.4 to illustrate the approach. It is also important to note that the approach does not necessarily require a state variable that is binary.

The belief node corresponding to b is visualized in Fig. 2. In our approach, we use auxiliary 'conclude' actions which correspond to the values s^j can take, inspired by the commit actions that are used in POMDP-IR [16]. Intuitively, the agent can use these actions to guess the true value of s^j. When executing a conclude action, the corresponding value is stored in a fully-observable state variable v, which is also depicted in the belief nodes in the figure. After executing a conclude action, the agent receives observation o_c if the guess was correct, and o_i if the guess was incorrect. This observation signal is dependent on the true state variable s^j and on the guess that was just made, stored in state variable v. In Sect. 3 the subscript of o was used as an index, but please note that o_c and o_i refer to

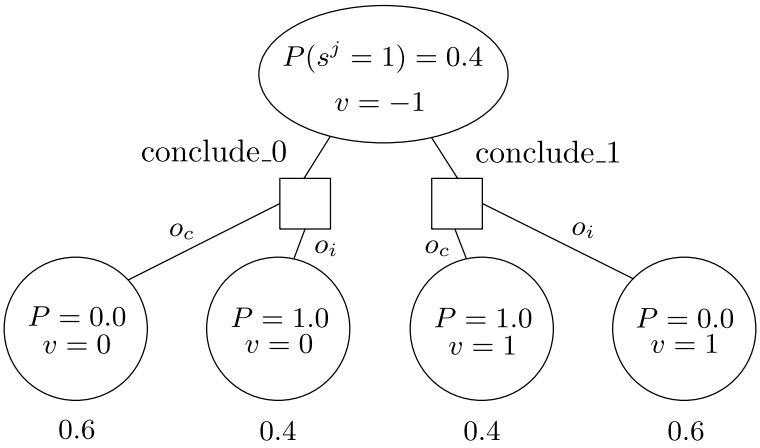


Fig. 2 Tree expansion using conclude actions

specific observations in the remainder of this paper, with a label c and i for being correct or incorrect.

We consider an example to illustrate the use of the conclude actions and observations. Given the initial belief in the root we execute action 'conclude_0', which stores value 0 in state variable v. By executing this action we guess that the true state of s^j is 0. This guess can be either correct or incorrect. In the root it holds that $P(s^j = 0 \mid b) = 1 - P(s^j = 1 \mid b) = 0.6$, and therefore observation o_c will be seen with probability 0.6 after executing the conclude action. Similarly, the observation o_i will be seen with probability 0.4, indicating that the guess was incorrect. For convenience both probabilities are shown in the figure below the belief nodes. After executing 'conclude_0' and observing o_c it must be the case that the true value of s^j is 0, and therefore $P(s^j = 1 \mid b) = 0$, which is depicted as $P = 0.0$ in the belief node. Similarly, after executing 'conclude_0' and observing o_i it must be the case that the true value of s^j is 1, which means that $P(s^j = 1 \mid b) = 1$. For action 'conclude_1' the line of reasoning is identical and therefore omitted.

Given the tree structure in Fig. 1, we use active inference to create an incentive for the agent to guess the true value of s^j correctly. This is modeled by defining the observation o_c as a preferred observation in the prior $\bar{o}$. Intuitively, this ensures that the agents wants to make a correct guess. Recall from Sect. 2 that this is modeled by assigning a positive weight to observation o_c, which we denote by $\bar{o}_c$. Since o_c is triggered if the correct conclude action was executed, defining this observation as preferred observation means that receiving preferred observations corresponds to state uncertainty reduction. Intuitively, the probability to receive preferred observation o_c becomes high if the agent sufficiently reduces the uncertainty with respect to s^j before executing a conclude action. When integrating the conclude actions and observations o_c and o_i in the planner, the agent will first execute actions to reduce uncertainty about s^j. Eventually, it will execute one of the conclude actions in order to receive the preferred observation o_c.

4.3 Adjusting the expected free energy equation

The approach that we presented creates an incentive for the agent to execute actions which reduce uncertainty, such that the probability to receive the preferred observation o_c

increases. However, we cannot integrate this directly in Monte Carlo Tree Search, because the observations o_c and o_i reveal the true state to the planner. This can be seen in the leaf nodes in Fig. 2, in which the probability $P(s^j = 1)$ is either 0 or 1. In reality the agent is uncertain about the actual value of s^j, and therefore we cannot proceed with the planning process starting from these nodes. Additionally, it is inconvenient to use auxiliary conclude actions and observations, because this extends the size of the original POMDP.

Rather than explicitly integrating conclude actions and additional observations, we show that an expected free energy term can be added to Eq. 7. This term is equivalent to the expected free energy defined by the construction illustrated in Fig. 2, but it comes with the additional advantage that it is not required to expand the POMDP model with conclude actions during the planning process. We derive and formalize this term in Theorem 1 below.

Theorem 1 *For a given belief b, the expected free energy induced by conclude actions is equal to*

$$\min_{l \in V_j} \left[-P(s^j = l \mid b) \cdot \ln(\bar{o}_c) - (1 - P(s^j = l \mid b)) \cdot \ln(\bar{o}_i) \right], \tag{9}$$

in which V_j is a set containing all possible values of s^j. The terms $0 < \bar{o}_c \leq 1$ and $0 < \bar{o}_i \leq 1$ denote the prior preference for observation o_c and o_i, respectively.

Proof We first consider the expected free energy of action 'conclude_l', as defined by the tree expansion illustrated in Fig. 2. We first consider two cases. When receiving observation o_c, the free energy is equal to $(\ln(1) - \ln(\bar{o}_c)) + (-1 \cdot 1 \cdot \ln(1)) = -\ln(\bar{o}_c)$ according to Eq. 7. When receiving observation o_i, the free energy is equal to $(\ln(1) - \ln(\bar{o}_i)) + (-1 \cdot 1 \cdot \ln(1)) = -\ln(\bar{o}_i)$. Observation o_c is received with probability $P(s^j = l)$, and observation o_i is received with probability $1 - P(s^j = l)$. Now it follows that the total expected free energy when taking action 'conclude_l' equals $-P(s^j = l \mid b) \cdot \ln(\bar{o}_c) - (1 - P(s^j = l \mid b)) \cdot \ln(\bar{o}_i)$. The agent wants to minimize expected free energy, which means that the total expected free energy induced by conclude actions can be obtained by taking the minimum over all possible values l that can be taken by s^j: $\min_{l \in V_j} \left[-P(s^j = l \mid b) \cdot \ln(\bar{o}_c) - (1 - P(s^j = l \mid b)) \cdot \ln(\bar{o}_i) \right]$, in which the variable $l \in V_j$ enumerates all values l of s^j. □

The additional expected free energy term can be easily added to the original expected free energy as defined by Eq. 7. This means that it is not required to actually introduce conclude actions and additional observations in the POMDP model. The prior preferences $\bar{o}_c$ and $\bar{o}_i$ can be used to define to what extent the agent wants to reduce uncertainty about the actual value of s^j. Intuitively, if $\bar{o}_c$ is much higher than $\bar{o}_i$, then the agent aims to reduce uncertainty about s^j. If both prior preferences are equal, then the agent has no incentive to reduce uncertainty. If $\bar{o}_c = \bar{o}_i$, then the expected free energy of each conclude action equals $-\ln(\bar{o}_c)$, which is a constant that is not dependent on the uncertainty with respect to s^j. It is important to note that the additional expected free energy term is added for one specific belief, but in the planning search tree the future beliefs and their expected free energy are also considered because their expected free energy is propagated upwards in the tree.

To elaborate our intuition about the influence of the prior preferences on expected free energy, we provide a visual example in Fig. 3. In this example we set $\bar{o}_i = 1 - \bar{o}_c$, and

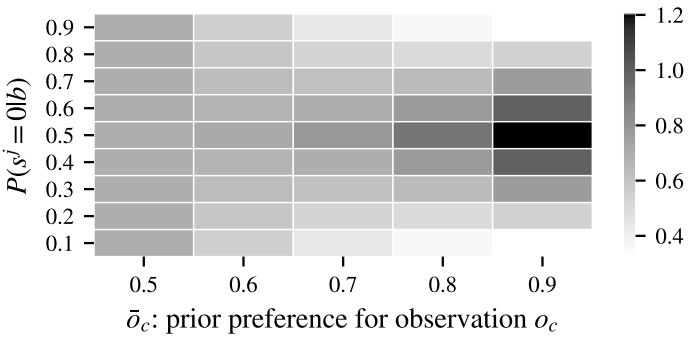


Fig. 3 Expected free energy induced by conclude actions

since s^j is binary it holds that $P(s^j = 1 \mid b) = 1 - P(s^j = 0 \mid b)$. As expected, if the preference for both observations is equal, then the expected free energy is not affected by the uncertainty with respect to s^j. This can be seen in the first column in the figure. When increasing $\bar{o}_c$, the expected free energy becomes low if $P(s^j = 0 \mid b)$ is either low or high, which is the case if uncertainty with respect to s^j is low. In the rightmost column it can also be seen that the expected free energy is high if uncertainty with respect to s^j is high. The figure confirms visually that minimizing our expected free energy term corresponds to minimizing state uncertainty.

The planning algorithm and the expected free energy term that we have presented in this section enables modeling of information gathering tasks directly in the active inference framework. This is an important step, because in current active inference approaches with the standard definition of the expected free energy it is not possible to express this directly in the problem formulation. In our approach, this can be modeled directly by including additional terms in the expected free energy for the state state variables for which information gathering is relevant. As a result, it becomes possible to use active inference for information gathering tasks, and thereby we have created a new alternative for existing information gathering approaches for POMDPs. An evaluation of the approach will be provided in the next section.

5 Experiments

In this section, we present the results of our experimental evaluation, in which we compare our algorithm based on active inference with a baseline planning algorithm in three different domains. Our active inference planner uses expected free energy to gather information about the environment state while navigating towards a goal, which is defined using preferred observations. Our planner corresponds to Algorithm 1, and we use UCB with scalar parameter 10 to balance exploration and exploitation during tree search. Since it is required to use a planning algorithm which supports belief-based rewards, we only use belief-based MCTS in our experiments. More details and a description of the domains are provided in the corresponding sections below.[1]

[1] Source code of the algorithm is available online: https://github.com/erwinwalraven/active_inference.

5.1 Inspecting pipes in industrial area

In our first experiment, we show that our planner is able to gather information about a potential leak in a pipe. Below we first discuss the domain itself. Next, we discuss how the experiment is executed, followed by our results and conclusions.

The domain that we consider is an inspection task in which the agent needs to walk a path from a start location to an end location, and along the way it needs to check whether there is a leak in one of the pipes. The domain is depicted in Fig. 4, in which the agent walks from cell 0 to cell 4. The destination in cell 4 needs to be reached before a deadline, which means that the agent has a limited amount of time to execute its actions. In cell 1, the agent is able to perform inspections in order to determine whether pipe A is leaking. However, if the agent detects a leak from the inspection point, it can be caused by either a leak in pipe A or a leak in pipe B. In order to determine whether pipe A leaks, the agent first needs to go to cell 2 to switch off the pump. Intuitively, by switching off the pump it eliminates the possibility that the leak is caused by pipe B. Furthermore, when switching off the pump, the agent observes whether cell 3 is accessible or not. If it is accessible, then it can walk from cell 2 to cell 4 via cell 3. Otherwise it must take the longer path via cell 6 to reach the end location. There are two types of inspections that can be performed in cell 1: a long inspection that is accurate and a short inspection that is more noisy. The agent keeps track of time in the state, and long and short refers to the amount of time the action takes. While choosing one of the inspection actions, the agent needs to reason whether it is still able to reach the end location on time. For example, if cell 3 is not accessible, then the agent has less time for inspection and therefore a long inspection in cell 1 may not be feasible. Similarly, if cell 3 is accessible, then the agent has plenty of time for a longer inspection, which provides accurate information about the existence of a leak.

The time and location of the agent are fully observable in this planning task. The original state of the pump, and the existence of a leak in A and B are partially observable. Prior to planning, we initialize a uniform initial belief for the state variables that are partially observable, consisting of 1000 state particles. In the initial belief the location of the

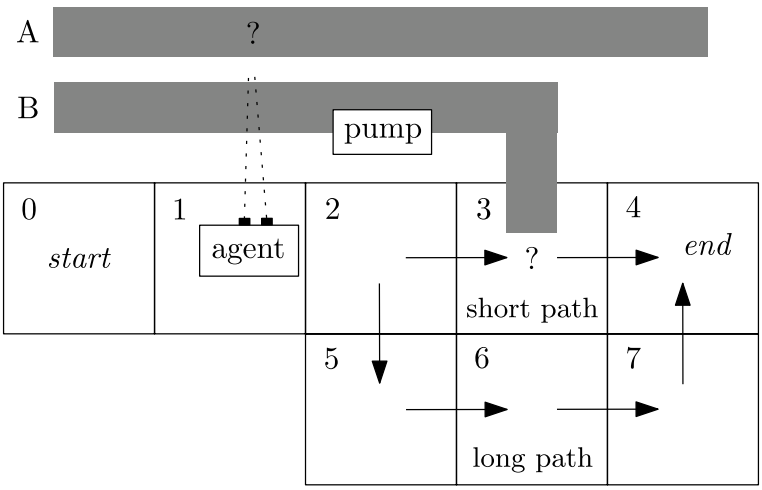


Fig. 4 Pipe inspection domain

Table 1 Observation probabilities in case of leak in A

State	Inspection action	Probabilities
leak	short	$P(o_1) = 0.9, P(o_0) = 0.1$
leak	long	$P(o_1) = 1.0$
leak	long_2	$P(o_1) = 0.5, P(o_1') = 0.5$
no leak	short	$P(o_1) = 0.1, P(o_0) = 0.9$
no leak	long	$P(o_0) = 1.0$
no leak	long_2	$P(o_0) = 1.0$

Table 2 Scenario with leak in A and short path

	Active inference	Baseline
Destination reached	1.00 ± 0.00	1.00 ± 0.00
Entropy leak A	0.00 ± 0.00	0.00 ± 0.00
Total runtime (s)	65.88 ± 1.99	60.09 ± 2.98
Entropy obs long inspect	0.00 ± 0.00	0.29 ± 0.30

agent is always cell 0, and it always starts at time 0. When performing an inspection, the observation signal of the agent depends on state of pipe A and B. If at least one of these pipes leaks, a long inspection informs the agent that there is a leak. A short inspection only reveals this information in 90 percent of the cases. We apply active inference to choose the actions to be executed. When reaching cell 4, a preferred observation is triggered, defining the goal of the agent. Furthermore, for the binary state variable corresponding to the existence of a leak in pipe A, we insert additional terms in the expected free energy as defined by Eq. 9. By minimizing the expected free energy, we expect that the agent chooses actions which reveal information about a potential leak in pipe A, while navigating towards cell 4.

We conduct an experiment to show that our planner based on active inference is able to gather information about a potential leak in pipe A. We also use a baseline POMDP planner in which we define a large reward that the agent receives when reaching cell 4 while uncertainty with respect to a leak in pipe A is low. We define the true environment state in such a way that pipe A leaks, the pump is running and the short path is accessible. The running pump ensures that the agent must switch off the pump first in order to see whether pipe A leaks, and the accessibility of the short path ensures that the agent has the flexibility to choose either a short or long inspection in cell 1. We use the observation probabilities shown in Table 1. The observation o_1 indicates that there is a leak, and o_0 indicates that there is no leak. It can be seen that a long inspection provides more accurate information than a short inspection. Therefore, we expect that the agent chooses long inspection actions. We use two variants of the long inspection. The regular long inspection always triggers observation o_1 in case of a leak, and the auxiliary action long_2 triggers either observation o_1 or o_1' in case of a leak. The latter introduces ambiguous observations in case of a leak, and we expect that the active inference planner tends to avoid such observations, whereas it does not matter for the baseline POMDP planner since they provide the same information about the leak.

The results of our evaluation are shown in Table 2, measured based on 100 runs. Destination reached indicates whether the agent reached cell 4, which is ideally 1. It can be seen that both planners ensure that the agent actually reaches the end location in time. Entropy

leak A represents the uncertainty in the state variable that indicates whether pipe A leaks. Both planners are able to plan actions in such a way that the agent collects information about the leak. The total runtime indicates how long one run takes. For all long inspection actions chosen by the planner during 100 runs, we evaluate the entropy of the expected observations (entropy obs long). As discussed above, we expect this entropy to be low for active inference, and this is confirmed by the results in the table. It can be seen that active inference always chooses long inspections with low observation uncertainty. The baseline POMDP planner does not capture this type of uncertainty in its reward signal, and it can be seen that it sometimes executes long inspections which lead to more uncertainty in observations (the mean entropy is 0.29 rather than 0.00).

We repeated the experiment for the scenario in which the short path is not accessible. This means that the agent must execute a short inspection in cell 1 after switching off the pump, because otherwise there is insufficient time to reach cell 4. The results are shown in Table 3. As expected, it can now be seen that active inference also chooses short inspection actions with uncertainty in the observation signal (both mean entropy values in the bottom row are positive). Furthermore, compared to the baseline planner active inference inspects in such a way that state uncertainty becomes lower. On average the entropy of the belief with respect to the leak in A is 0.15, whereas this entropy value is 0.49 on average when using the baseline planner.

To summarize, our experiment has shown that our tailored active inference approach for information gathering is able to collect information about the environment state. This confirms that our approach introduced in Sect. 4.2 is able to effectively reduce uncertainty with respect to states.

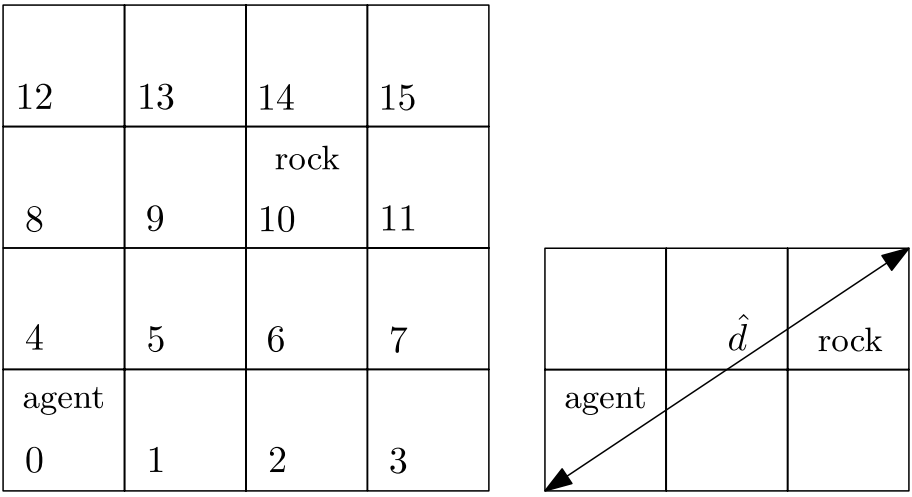
5.2 Comparison with entropy-based rewards

In our next experiment, we make the comparison between our active inference planner and a planner that uses an entropy-based reward signal. We perform this experiment in multiple configurations of a rock inspection domain, inspired by the RockSample domain [29], with various settings for the prior preference for $\bar{o}_c$. Furthermore, we also compare with a simple baseline that always executes a random feasible action. In the remainder of this section we first provide an introduction to the domain, after which we present and discuss the results of the comparison.

We consider a rock inspection task in a 4×4 grid, as illustrated in Fig. 5a. The agent starts in cell 0, and is tasked to infer the state of a rock which can be positioned in the cells $8 - 15$. The true state of the rock can take value 0, 1 or 2, and initially the belief over these values is uniform. The state variables for the position of the agent, the position of the rock and the time are fully observable. The agent can perform four deterministic move actions, one for each direction, and there is an inspect action which can be used to reveal

Table 3 Scenario with leak in A and long path

	Active inference	Baseline
Destination reached	1.00 ± 0.00	1.00 ± 0.00
Entropy leak A	0.15 ± 0.23	0.49 ± 0.29
Total runtime (s)	66.55 ± 3.10	61.99 ± 3.58
Entropy obs short inspect	0.43 ± 0.10	0.44 ± 0.11



(a) Grid with agent and rock

(b) Example of distance $\hat{d}$

Fig. 5 Rock inspection domain

information about the true state of the rock. These inspections can only be executed in cells $8-15$, and therefore the agent first needs to move to the upper half of the grid before it can execute inspections.

The inspect action can trigger three observations, each of which corresponds to the true state of the rock, and the correctness of these observations is dependent on the distance $\hat{d}$ between the agent and the rock. An example of such a distance is shown in Fig. 5b. Based on this distance we define the probability that inspect returns the correct observation signal as follows:

$$1 - \frac{(\hat{d} - \sqrt{2^2 + 1^2})}{(\sqrt{4^2 + 4^2} - \sqrt{2^2 + 1^2})}, \tag{10}$$

in which $\sqrt{2^2 + 1^2}$ and $\sqrt{4^2 + 4^2}$ denote the minimum and maximum distance, respectively, such that the entire term evaluates to a number in the interval [0, 1]. For example, if the distance $\hat{d}$ is maximum, then the probability becomes 0, and if the robot stands next to the rock then the probability becomes 1.

We perform an experiment in which the agent can perform actions to infer the true state of the rock, and we assess this based on the entropy over rock states based on the final belief after action execution. The agent is able to execute at most 7 actions, such that the agent is potentially able to reach cell 15 and perform an inspection there. This deadline also creates the incentive to be efficient, which means that it does not have the time to walk around a large amount of time before starting inspections. We use a random planner which takes random actions, for which we expect that the entropy remains high. We also include a planner which uses an entropy-based reward signal, which rewards the agent for reducing entropy. In every step the agent gets reward $E_m - E$, in which E denotes the entropy over rock state values according to the final belief and $E_m = -1 \cdot 1/3 \cdot \log(1/3) \cdot 3$ is the maximum entropy for a state variable with three possible values. We also use three variants of our active inference planner, abbreviated AI, with various values for the prior

preference $\hat{o}_c$. We expect that the active inference planner with $\hat{o}_c = 0.5$ does not always reveal the true rock state, whereas the planner with $\hat{o}_c = 1.0$ does reveal the true state, confirming the intuition that we visualized in Fig. 3. Our experimental setup is the same as in the previous experiment, and each run is repeated 100 times.

The results of our experiment are shown in Table 4. In the first column we can see that, as expected, the random planner does not manage to reduce the uncertainty regarding the true state of the rock. The planner with an entropy-based reward function does always reduce the uncertainty, which is also aligned with our expectations. As we have seen in Fig. 3, in an active inference planner with $\bar{o}_c = 0.5$ the additional free energy induced by conclude actions is constant, which means that it does not create an additional incentive to reduce uncertainty. This is also confirmed by the results in the table. When increasing the prior to $\bar{o}_c = 1.0$ we see that the planner is effectively able to reduce uncertainty. This result also confirms that active inference is an attractive alternative approach for information gathering tasks in POMDPs, providing an alternative for existing reward signals for such tasks.

5.3 Avoiding specific observation signals during inspections

In our final experiment, we consider a task in which the agent needs to check in which room a person is located, as illustrated in Fig. 6. The purpose of this experiment is to intuitively confirm that active inference can be used to pick actions which lead to observation signals that are preferred, and to pick actions which avoid observation signals that are not preferred, while still being able to reduce uncertainty about the environment state. The agent starts in cell 0, where it can decide to travel to cell 3 via either cell 1 or cell 2. After arriving in cell 3 the agent performs one inspection action, which reveals in which room the person is located. This sensory action provides the correct answer in 95 percent of the cases. While traveling through cell 1 it is silent, whereas the agent hears music while it travels through cell 2. For the purpose of this experiment we define that the agent prefers to travel through a silent area, by defining the corresponding observation as preferred observation. As we will show next, active inference automatically plans its actions in such a way that the area with music is avoided. We use the same experimental setup as in Sect. 5.1, and the results are shown in Table 5. Both planners ensure that the inspection point is reached and they execute an inspection to reduce uncertainty about the location of the person. Please note that the agent executes only one inspection action which reveals the location correctly in 95 percent of the cases, and therefore it is not possible to reduce state

Table 4 Final entropy in rock inspection experiment

Rock position	Random	Entropy-based	AI ($\bar{o}_c = 0.5$)	AI ($\bar{o}_c = 0.75$)	AI ($\bar{o}_c = 1.0$)
8	0.98 ± 0.30	0.00 ± 0.00	0.21 ± 0.26	0.00 ± 0.00	0.00 ± 0.00
9	0.89 ± 0.41	0.00 ± 0.00	0.11 ± 0.20	0.00 ± 0.00	0.00 ± 0.00
10	0.90 ± 0.36	0.00 ± 0.00	0.25 ± 0.30	0.20 ± 0.25	0.00 ± 0.00
11	1.02 ± 0.20	0.00 ± 0.00	0.25 ± 0.30	0.24 ± 0.27	0.00 ± 0.00
12	0.93 ± 0.36	0.00 ± 0.00	0.15 ± 0.23	0.00 ± 0.00	0.00 ± 0.00
13	0.93 ± 0.34	0.00 ± 0.00	0.07 ± 0.13	0.05 ± 0.15	0.00 ± 0.00
14	0.97 ± 0.28	0.00 ± 0.00	0.54 ± 0.34	0.20 ± 0.27	0.00 ± 0.00
15	1.02 ± 0.21	0.00 ± 0.00	0.59 ± 0.29	0.24 ± 0.28	0.00 ± 0.00

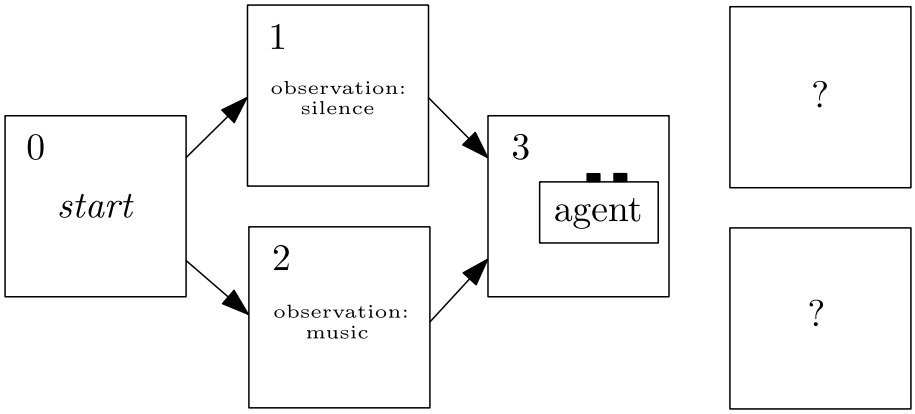


Fig. 6 Agent inspecting two rooms to find a person

Table 5 Finding a person using an inspection

	Active inference	Baseline
Inspection point reached	1.00 ± 0.00	1.00 ± 0.00
Entropy person location	0.19 ± 0.04	0.17 ± 0.03
Total runtime (s)	16.13 ± 0.51	15.88 ± 0.48
Music observed	0.00 ± 0.00	0.47 ± 0.50

uncertainty completely in the second row of the table. The baseline planner does not model preferred observations, and therefore both paths to cell 3 are equivalent from the viewpoint of the planner. In the table, it can be seen that the baseline planner sometimes sends the agent via cell 2, where it perceives music, whereas the active inference planner always chooses cell 1 as desired. As expected, our experiment shows that active inference may be used to express that specific types of observations are preferred or not preferred to be perceived. Compared to the use of a regular POMDP algorithm this is advantageous, because as a modeler it is not required to first reason about the states in which this signal may be perceived. Instead, the modeler can express this directly by setting the prior. Furthermore, the results also show that the agent still manages to reduce uncertainty about the environment state.

6 Related work

In active perception, agents consider the effects of their actions on the performance of sensors, in such a way that these sensors can be used to get information about the true environment states [9]. The POMDP-IR formalism models this using information-gain rewards, which reward the agent for reducing uncertainty in the state belief [16]. POMDP-IR uses commit actions which reward the agent for guessing the state correctly, which follows a similar intuition as the conclude actions that we use in Fig. 2. Active inference is more generic, because in addition to planning for state uncertainty reduction it also enables agents to plan towards goals, which are defined by preferred observations. Planning

towards goals is also considered by Goal-Directed POMDPs [30, 31], but this framework only models goal states that should be reached, and it is not directly suitable for perception.

Active inference researchers also studied the connection to planning, specifically for navigation tasks. It has been shown how it can be used to navigate towards goals in maze domains [10], but scalability is limited because it requires full policy enumeration and evaluation of the expected free energy. Active tree search has been proposed to address this issue [23]. In particular, the Active Inference Tree Search algorithm (AcT) also constructs a search tree during simulation runs. However, the AcT algorithm builds a tree with branches for actions only, rather than constructing action and observation branches. A tree based on actions only is not suitable for our planning task because successor beliefs are dependent on both actions and observations. Furthermore, the tree in the AcT algorithm can be used to plan the first action to take, but during execution of multiple subsequent actions it is not possible to follow branches of the tree based on actions and observations. Our planning tree does support this, in such a way that the planning tree only needs to be computed once prior to action execution. Monte Carlo Tree Search has also been proposed in the context of deep active inference [28], with the goal to avoid policy enumeration when constructing scalable active inference agents. Deep active inference uses tree search combined with deep neural networks for planning and for policy approximation, but there is no focus on information gathering tasks specifically. Related methodologies are Sophisticated Inference [32] and Branching Time Active Inference [33], which also rely on tree search. However, these similar approaches do not use a particle representation and information gathering is not the main focus.

Planning as inference interprets planning as maximum likelihood estimation of a policy, conditional on the future reward that is expected according to a cognitive generative model [34]. However, its generative model is purely used to condition on reward, and it is not used to reason about uncertainty-resolving behavior. Inference frameworks for planning and decision making are able to consider observation ambiguity. Compared to active inference they encode the concept of value differently in its generative model. We refer to work by Millidge et al. for more details [35].

Besides planning based on a given POMDP model, active inference has also been studied in the context of reinforcement learning. Deep active inference is able to learn policies directly from sensory inputs in a partially observable setting [28, 36, 37]. We expect that our tailored expected free energy term can also be used in these settings, because deep active inference is also based on the expected free energy, in which our adjustments integrate naturally.

7 Conclusion

In this paper, we have considered planning tasks in which gathering information about the environment state is the main goal, rather than being a means to reach a goal. In the POMDP literature several approaches have emerged which can be used to enable agents to gather information about the environment state. Similar types of problems were studied in the field of active inference, which integrates planning and perception in a single framework. These developments raise the question how both fields relate to each other, and to what extent both lines of work can be use for information gathering tasks. In this paper we made a step to bring both fields closer to each other by discussing how active inference relates to information gathering in POMDPs, and how active inference can be

extended in such a way that it can be used to plan in information gathering tasks. In particular, we derived an approach to introduce an additional expected free energy term in the active inference framework, and by minimizing this quantity we induce information gathering behavior in active inference based on reducing state uncertainty. A series of experiments confirmed that our tailored active inference approach can be used in information gathering tasks, providing an alternative to common POMDP approaches for information gathering. Furthermore, we hope that our work opens the door towards more research at the intersection of both research areas in the future. For example, for information gathering tasks it may be relevant to consider additional rules and constraints, and developments on constrained planning [38] may also be applicable to active inference planning algorithms.

Acknowledgements The research described in this article has been funded by the SNOW project of the TNO Appl.AI program.

Author contributions E.W has created the proposed method, performed the experiments and wrote the manuscript text. J.S. was involved in creating the method, refining the experimental setup, writing parts of the manuscript text, and acquired funding. G.B. discussed with the other authors about active inference and about setting up the experiments. All authors reviewed the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Veldman, E., & Verzijlbergh, R. A. (2014). Distribution grid impacts of smart electric vehicle charging from different perspectives. *IEEE Transactions on Smart Grid, 6*(1), 333–342.
2. Walraven, E., Spaan, M. T. J. (2016). Planning under uncertainty for aggregated electric vehicle charging with renewable energy supply. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence, 904–912
3. Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence, 101*(1–2), 99–134.
4. Pineau, J., Gordon, G., Thrun, S. (2003). Point-based value iteration: An anytime algorithm for POMDPs. In: Proceedings of the International Joint Conference on Artificial Intelligence, 1025–1032
5. Silver, D., Veness, J. (2010). Monte-Carlo Planning in Large POMDPs. In: Advances in Neural Information Processing Systems, 2164–2172
6. Di Paola, D., Milella, A., Cicirelli, G., & Distante, A. (2010). An autonomous mobile robotic system for surveillance of indoor environments. *International Journal of Advanced Robotic Systems, 7*(1), 8.
7. Almadhoun, R., Taha, T., Seneviratne, L., Dias, J., & Cai, G. (2016). A survey on inspecting structures using robotic systems. *International Journal of Advanced Robotic Systems, 13*(6), 1729881416663664.
8. Almeida, J., Almeida, A., Araújo, R. (2005). Tracking multiple moving objects for mobile robotics navigation. In: 2005 IEEE Conference on Emerging Technologies and Factory Automation.
9. Spaan, M.T.J. (2008). Cooperative Active Perception using POMDPs. In: Proceedings of the AAAI 2008 Workshop on Advancements in POMDP Solvers, 49–54
10. Kaplan, R., & Friston, K. J. (2018). Planning and navigation as active inference. *Biological Cybernetics, 112*(4), 323–343.
11. Cassandra, A., Littman, M. L., Zhang, N. L. (1997). Incremental Pruning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Processes. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, 54–61
12. Walraven, E., Spaan, M. T. J. (2017). Accelerated Vector Pruning for Optimal POMDP Solvers. In: Proceedings of the AAAI Conference on Artificial Intelligence, 3672–3678
13. Kurniawati, H., Hsu, D., Lee, W. S. (2008). SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. In: Proceedings of Robotics: Science and Systems
14. Veiga, T., & Renoux, J. (2023). From reactive to active sensing: a survey on information gathering in decision-theoretic planning. *ACM Computing Surveys, 55*(13s), 1–22. https://doi.org/10.1145/3583068

15. Araya-Lopez, M., Buffet, O., Thomas, V., Charpillet, F. (2010). A pomdp extension with belief-dependent rewards. Advances in Neural Information Processing Systems.
16. Spaan, M. T. J., Veiga, T. S., & Lima, P. U. (2015). Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems, 29*(6), 1157–1185.
17. Thomas, V., Hutin, G., Buffet, O. (2020). Monte carlo information-oriented planning. In: Proceedings of the European Conference on Artificial Intelligence 2020, 2378–2385
18. Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience, 11*(2), 127–138.
19. Pezzato, C., Corbato, C. H., Bonhof, S., & Wisse, M. (2023). Active inference and behavior trees for reactive action planning and execution in robotics. *IEEE Transactions on Robotics, 39*(2), 1050–1069.
20. Da Costa, L., Lanillos, P., Sajid, N., Friston, K., & Khan, S. (2022). How active inference could help revolutionise robotics. *Entropy, 24*(3), 361.
21. Sajid, N., Ball, P. J., Parr, T., & Friston, K. J. (2021). Active inference: demystified and compared. *Neural Computation, 33*(3), 674–712.
22. Da Costa, L., Parr, T., Sajid, N., Veselic, S., Neacsu, V., & Friston, K. (2020). Active inference on discrete state-spaces: a synthesis. *Journal of Mathematical Psychology, 99*, 102447.
23. Maisto, D., Gregoretti, F., Friston, K., Pezzulo, G. (2021). Active Inference Tree Search in Large POMDPs. arXiv preprint arXiv:2103.13860
24. Sunberg, Z. N., Kochenderfer, M. J. (2018). Online algorithms for pomdps with continuous state, action, and observation spaces. In: Proceedings of the International Conference on Automated Planning and Scheduling, 259–263
25. Doshi, P., & Gmytrasiewicz, P. J. (2009). Monte carlo sampling methods for approximating interactive POMDPs. *Journal of Artificial Intelligence Research, 34*, 297–337.
26. Coulom, R. (2006). Efficient selectivity and backup operators in monte-carlo tree search. In: Proceedings of the International Conference on Computers and Games, 72–83
27. Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning, 47*(2), 235–256.
28. Fountas, Z., Sajid, N., Mediano, P., Friston, K. (2020). Deep active inference agents using monte-carlo methods. In: Advances in Neural Information Processing Systems, 11662–11675
29. Smith, T., Simmons, R. (2004). Heuristic Search Value Iteration for POMDPs. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, 520–527
30. Geffner, H., Bonet, B. (2013) A Concise Introduction to Models and Methods for Automated Planning. Synthesis Lectures on Artificial Intelligence and Machine Learning
31. Hou, P., Yeoh, W., Varakantham, P. (2016). Solving Risk-Sensitive POMDPs with and without Cost Observations. In: Proceedings of the AAAI Conference on Artificial Intelligence, 3138–3144
32. Friston, K., Da Costa, L., Hafner, D., Hesp, C., & Parr, T. (2021). Sophisticated inference. *Neural Computation, 33*(3), 713–763.
33. Champion, T., Da Costa, L., Bowman, H., & Grześ, M. (2022). Branching time active inference: the theory and its generality. *Neural Networks, 151*, 295–316.
34. Botvinick, M., & Toussaint, M. (2012). Planning as inference. *Trends in Cognitive Sciences, 16*(10), 485–488.
35. Millidge, B., Tschantz, A., Seth, A. K., Buckley, C. L. (2020). On the relationship between active inference and control as inference. In: Proceedings of the International Workshop on Active Inference, 3–11
36. Himst, O.v.d., Lanillos, P. (2020). Deep Active Inference for Partially Observable MDPs. In: Proceedings of the International Workshop on Active Inference, 61–71
37. Millidge, B. (2020). Deep active inference as variational policy gradients. *Journal of Mathematical Psychology, 96*, 102348.
38. De Nijs, F., Walraven, E., De Weerdt, M. M., & Spaan, M. T. J. (2021). Constrained multiagent Markov decision processes: a taxonomy of problems and algorithms. *Journal of Artificial Intelligence Research, 70*, 955–1001.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.