# Robot navigation as hierarchical active inference

Ozan Çatal[a], Tim Verbelen[a], Toon Van de Maele[a], Bart Dhoedt[a], Adam Safron[b]

[a] *Ghent University - imec*
[b] *Indiana University*

**Abstract**

Localization and mapping has been a long standing area of research, both in neuroscience, to understand how mammals navigate their environment, as well as in robotics, to enable autonomous mobile robots. In this paper, we treat navigation as inferring actions that minimize (expected) variational free energy under a hierarchical generative model. We find that familiar concepts like perception, path integration, localisation and mapping naturally emerge from this active inference formulation. Moreover, we show that this model is consistent with models of hippocampal functions, and can be implemented in silico on a real-world robot. Our experiments illustrate that a robot equipped with our hierarchical model is able to generate topologically consistent maps, and correct navigation behaviour is inferred when a goal location is provided to the system.

*Keywords:* active inference, robot navigation, SLAM, RatSLAM, deep learning

## 1. Introduction

Being able to robustly explore and navigate an environment has been a long standing challenge in robotics. In past decades this has been mainly addressed by simultaneous localization and mapping (SLAM), in which a robot builds a metric (grid) map of the environment using sensory input of various sensors such as lidars and cameras [1]. Although the tremendous progress in this area, current approaches still fall short when operating in complex, dynamic environments for a longer period of time [2].

Being able to navigate successfully and consistently through an environment is a fundamental capability for almost all animals [3]. Humans easily explore and map their surroundings without much extra thought or considerations. Intuitively we build a logically consistent map without needing accurate distance measurements [4]. We find it natural to think about navigation in terms of a sequence of moves towards locations. When considering moving to a new location, humans typically do not consider lower level tasks such as opening doors or lifting feet in the mental process. Only when something unexpected occurs do we actively engage in planning in terms of individual movements of the joints. For example, when encountering a puddle on the side walk, we'll actively start planning in order to not get our feet wet. We can thus see the process of human navigation in terms of hierarchical model, the lower levels allows us to reason about joint states and immediate observations. The higher levels allows us to reason about different locations and how these are connected to each other.

Although the working of human mapping and localization is still not thoroughly understood, there are numerous models of rodent navigation in the hippocampus. Research suggests rodents build topological maps, in the sense that they recall one experience being correlated by some other set of experiences through some spatial relation. This relation, typically is not only expressed in terms of distance in meters, but also in closeness through spatio-temporal consistency and experiences [5, 6].

Besides building a map and localizing oneself therein, navigation also entails planning and following routes to goals, as well as environment exploration. Active inference [7] is a process theory of the brain that casts action as perception as two sides of the same coin. It rests upon the idea the free energy minimization underpins the mechanisms and motivations of organism agency. It builds upon the free energy principle, which states that every agent builds a generative model, of varying complexity, of the world. This model then is used to make predictions about possible outcomes of the world. Future actions and model beliefs are inferred in order to minimize the surprise this generative model induces. The complexity of this model typically depends on the capabilities of the

agent, i.e. a virus would have a very simple generative model of the world that is only capable of "reasoning" over short term homeostasis. A mammal on the other hand would need a complex generative model that can reason over longer timescales.

In this paper, we propose a hierarchical generative model casting navigation as minimizing (expected) variational free energy [8]. We show how perception, localization, mapping and navigation naturally emerge from optimizing this hierarchical generative model under active inference. Moreover, we implement this system in silico on a real-world robot platform, navigating a warehouse environment using camera sensor input only. Finally, we also relate our work to various findings and hypotheses in biology about navigation in the rodent and primate brain.

Recent work already illustrated how active inference can be applied for planning and navigation in simple, discrete maze tasks [9] [10]. Also, in our prior work we proposed how to learn generative state space models from pixel data to engage in active inference [11] [12]. However, these methods fall short when having to plan on sufficiently long timescales, as is ubiquitous in real-world navigation. Therefore, we extend these approaches with an hierarchical generative model, while also drawing inspiration from other bio-insipred SLAM approaches [13]. The remainder of this paper is structured as follows. In the next section, we first rehearse the literature on the neural correlates of navigation, both in rodent and primate brains. Section 3 then describes our hierarchical generative model, casting navigation and mapping as the minimization of (expected) variational free energy. In Section 4 we further discuss how the different factors of the generative model can be instantiated in silico, which we implement on a real-world mobile robot platform in Section 5. Finally we discuss our results in light of other recent research in Section 6.

## 2. Neural correlates of navigation: the hippocampus and beyond

The evolution of the hippocampal/entorhinal system (H/E) system represents a major transition in evolution [14, 15]. This advent is at least 300 million years old, as homologues are found in both birds and mammals [16], with large portions of this functionality likely being established over 500 million years ago in the Cambrian period [17, 18]. The H/E system provides a basis for both memory and control of behavior, solving multiple fundamental evolutionary challenges, including rapid associative learning, as well as the situating of an organism in space and time for the sake of navigation and searching/foraging for value.

We will ground the following discussion of the H/E system in terms of its role in (generalized) search and navigation [9]. A basic intuitive link of these functions can be found in the difficulty of finding something if an agent is unaware of its own location in space. Yet localization will be challenging if an agent lacks awareness of its motions - which when considered over time affords localizing via path integration - and of the likely observations accompanying its trajectory through space and time. In light of this, immense excitement was generated by the discovery of allocentric spatial representations in the hippocampus, where these "place cells" exhibited reliable responses when an organism frequented a given position in space [19]. Additional excitement was generated by the discovery of "grid cells" in entorhinal cortex [20] - and possibly elsewhere [21] - which represent a largely regular triangular or hexagonal tiling of space, organized according to egocentric reference frames. The regularity of these repeating grids provides a basis for path integration, via estimating distances and locations in terms of the directions and frequency with which this metric tiling is traversed. In physical domains, it has been observed that the spacing of grid representations varies as a function of the space being tiled, e.g. smaller and more fine-grained if an agent is situated within an enclosure, or larger and more coarse-grained if situated in an open space. The granularity of tilings appears to be influenced by brainstem locomotor nuclei [22] – potentially reflecting an

inductive bias for greater speed being more likely in larger spaces – as well as by exteroceptive stimulus cues such as optic flow [23].

Grid and place cells tend to be organized as 2D tilings in maps or graphs, as motion affordance tends to be constrained to planes for terrestrial animals. However, this is not necessarily true for aviation-capable beings [24]. The precise mechanistic and developmental relationships between place and grid cells remains a matter of some debate, with some viewing place cells as emerging from combinations of grid cells as their Fourier basis, and others viewing grid cells as a kind of compressed eigenvector description of likely place cell activity. The former perspective is emphasized in the Tolman-Eichenbaum machine [25], which has many parallels with the architecture described here. In this view, entorhinal grid cells encode particular structural details, and hippocampal cells link this basis set with sensory representations via an evolving map/graph of the world [5]. Alternatively, the latter view is associated with place cells representing a "predictive map" based on "successor representations"[1] of likely state transitions, from which grid cells are derived as higher-level, more invariant representations [26].

In addition to place and grid cells, a variety of additional specialized cell types have been observed in the H/E system. While it was previously assumed that these features represent innate inductive biases [27], increasing evidence suggests that these specialized cell types may arise from experience-dependent plasticity, including models with similar architectural principles to the ones described here. Support continues to accumulate for the H/E system as inherently predictive on multiple levels. Within the hippocampus, place fields appear to be tuned to the direction of motion, whether in 2 or 3 spatial dimensions [24]. Further, activation of place fields exhibit forward sweeps as organisms consider

[1]The notion of successor representations should not be confused with representations of successive states under different policies in the future. This is because the successor representation assumes that, following the subsequent action, agents pursue a fixed state-action policy. In other words, the successor representation does not call planning or inference.

alternative routes, and where the most robust sweeps predict and select the direction of subsequent locomotion [28, 29]. The precise sources of these predictive successor representations remain unclear [26, 30].

In rodent studies, H/E couples with multimodal (hexagonally organized) value representations of the ventromedial prefrontal cortex [31, 32], which may not only allow navigation to be organized according to a "common neural currency" of expected organismic value, but may also aid in updating those estimates based on histories of experience organized according to spatiotemporal trajectories. Similar bidirectional functional relationships may also be observed with the dorsomedial prefrontal cortex (and corresponding posterior structures), both allowing H/E successor representations to be informed based on activity in high level motor and gaze direction, but also potentially allowing visual and somatic spatial fields to be directed as a kind of navigation and foraging through informational and affordance spaces [33, 34], with saccades/samples and discrete actions orchestrated at theta frequencies [35, 36]. Somewhat less exotically, a substantial amount of predictive transitions between place fields may be explainable by relatively simple "bump attractor" models of CA3 [37, 38], or in terms of the ability of recurrent networks to encode predictive information - as a kind of spontaneous meta-learning - via their evolving attractor dynamics [39]. However, these may be compatible and complimentary possibilities, as predictive information could depend on both intrinsic properties of hippocampal circuits as well as the external systems with which they couple. Indeed, the H/E system and cortex dynamically interact in terms of flexible (near-critical) alternating of driving/responding [40, 41, 42]. Speculatively, if hippocampal pattern completion is inherently predictive, then this could help to scaffold the development of more sophisticated predictive abilities for the rest of the brain. Intra-hippocampal recurrent dynamics are notable for several reasons, one of which is the ability of such systems to serve as sequence memories, which is a highly influential interpretation of H/E functioning [43].

As the top of the cortical hierarchy, the H/E system not only provides a basis for storing and synthesizing novel information with spatiotemporal organization,

but it also helps to create frames of sense-making for episodes [44]. Consistent with the predictive processing models described above, fMRI studies show how continuous audiovisual stimuli are well-modeled as a nested hierarchy of events, with the hippocampus representing the highest level of organization of these events into coherent individual scenes [45]. With respect to goal-oriented behavior, these could be thought of as chunks, or subgoals in the context of hierarchical reinforcement learning and imaginative planning (as inference) [46, 47, 48].

To summarize, there is ample evidence that spatial awareness is at the core of the mammal brain, with explicit representations for pose, heading and location. Also, the H/E system is closely involved with episodic memory suggesting the storage of information as individual experiences linked together. Finally, recent research suggests that these representations are formed through predictive processing. In the remainder of this paper, we will further build on these insights using a process theory of the brain deeply rooted in predictive processing: active inference.

## 3. Navigation as hierarchical active inference

We cast the SLAM problem in terms of a hierarchical Bayesian generative model. The agent reasons on two different levels: on a higher level for long term navigation and on a lower level for short term perception. On the higher level, the agent is capable of reasoning in terms of sequences of locations it wants to visit without having to worry about the intricacies of how to control its actuators to get there. On the lower level, the agent can reason and plan in terms of observation sequences without needing to "think" too far ahead. We will distinguish between the higher level and lower level actuation by calling them moves $m$ and actions $a$ respectively. Sensor readings at the lower level are called observations $o$.

At the lower level, the model builds upon earlier work [12] in order to learn and infer belief states $s$ from sequences of observations $o$ and actions $a$. This allows the agent to model short-term lower level dynamics. In addition to this

belief state $s$, in this work we also explicitly model the agent's pose $p$ as a separate variable, in order to allow explicit reasoning about agent positions in the environment.

At the higher level, the model takes the set of (pose, state) pairs as observations to the model and links them with locations. Locations can be traversed by means of higher level moves. The relation between locations $l$, moves $m$ and observations $(o, s)$ is treated in the same way as the corresponding variables at the lower level.

### *3.1. Active inference*

Active inference posits that intelligent agents entertain a generative model of the world they operate in, and act in order to minimize surprise, or equivalently, maximize their model evidence [49]. Before we dive into the details of the proposed hierarchical model, we'll introduce a prototypical generative model for rehearsing the core concepts of active inference. Suppose an agent entertains

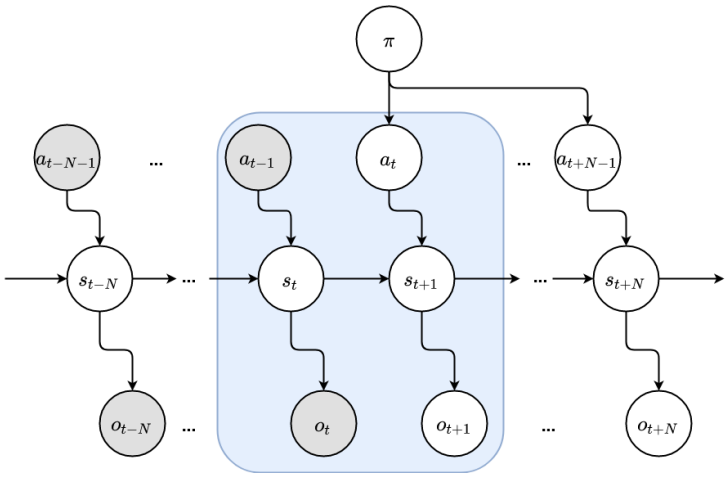



Figure 1: Prototypical generative model of an active inference agent. The nodes in the graphical model represent the random variables, and the grey color indicates the observed variables, while white nodes are assumed unobserved. An observation at the current timestep $o_t$ is generated by a hidden state $s_t$, which is in turn generated by the previous hidden state and action at timestep $t-1$. Future states and observations depend on future actions, which are determined by the policy $\pi$.

a generative model as shown graphically in Figure 1. The model $P(\tilde{o}, \tilde{s}, \tilde{a}, \pi)$ – note the usage of tildes to indicate sequences – specifies the way the agent expects the world to generate new observable outcomes. In particular, the agent believes that there is some hidden state $s$ in the world that connects observable actions $a$ with observable world outcomes, or observations, $o$. One could say that the agent is looking for hidden causes that explain the data it observes, given its actions. It is worth mentioning that the hidden state variables do not necessarily map in a one-to-one fashion to traditional physical quantities (such as locations, velocities in 3D space).

In active inference, the agent will infer beliefs over these hidden states based on experiences, i.e. looking back into the past observations, as well as infer future actions through a process of minimizing variational free energy. We can factorize the generative model up to a certain time horizon $H$ according to the relations specified in the graphical model as

$$P(\tilde{o}, \tilde{s}, \tilde{a}, \pi) = P(\pi)P(a_0|\pi)P(s_0)P(o_0|s_0)\prod_{t=1}^{H} P(s_t|s_{t-1}, a_{t-1})P(a_t|\pi)P(o_t|s_t). \tag{1}$$

The variational free energy up to the current timestep $t$ is then defined as [50]:

$$\begin{aligned} F &= \mathbb{E}_{Q(\tilde{s})}\big[\log Q(\tilde{s}) - \log P(\tilde{o}, \tilde{s}, \tilde{a})\big] \\ &= \underbrace{D_{KL}\big[Q(\tilde{s})||P(\tilde{s}|\tilde{o}, \tilde{a})\big]}_{\text{divergence}} - \underbrace{\log P(\tilde{o})}_{\text{log evidence}} \\ &= \underbrace{D_{KL}\big[Q(\tilde{s})||P(\tilde{s}, \tilde{a})\big]}_{\text{complexity}} - \underbrace{\mathbb{E}_{Q(\tilde{s})}\big[\log P(\tilde{o}|\tilde{s})\big]}_{\text{accuracy}}, \end{aligned} \tag{2}$$

in which the approximate posterior distribution $Q$ is introduced due to the variational approximation [51], which is necessary as full Bayesian inference is intractable for this kind of models. Note that wessume no uncertainty about the actions we executed in the past, which allows us to omit $\pi$ and set $\log P(\tilde{a}|\tilde{o}) = 0$. When unpacking the free energy equation, we see that when minimizing free energy, the agent is actually maximizing a lower bound on the log evidence.

The difference between the free energy and the log evidence is given by the KL divergence between the approximate posterior $Q(\tilde{s})$ and the true posterior. In machine learning this is known as the evidence lower bound (ELBO) [52, 53]. Similarly, we can also write the free energy as a complexity and accuracy term. When minimizing the free energy, the agent is actively attempting to find the least complex set of state values that still explain the observations the agent receives.

Crucially, in active inference the agent will not only optimize its generative model by minimizing $F$ for past observations. In addition, the agent will also select actions that it believes will minimize its future surprise. However, as future observations are not yet available, the free energy $F$ cannot be computed, and the expected free energy $G$ is used instead to compare the effect of various policies or actions in relation to the goal of reaching the preferred state. In essence $G$ is the amount of free energy the agent expects to achieve under a certain action sequence, or policy:

$$G(\pi) = \sum_{\tau} G(\pi, \tau) \tag{3}$$

The expected free energy $G(\pi, \tau)$ for a certain policy $\pi$ and timestep $\tau$ in the future for the generative model is defined as:

$$\begin{aligned} G(\pi, \tau) &= \mathbb{E}_{Q(s_\tau, o_\tau|\pi)}\big[\log Q(s_\tau|\pi) - \log P(o_\tau, s_\tau|\pi)\big] \\ &= \mathbb{E}_{Q(s_\tau, o_\tau|\pi)}\big[\log Q(s_\tau|\pi) - \log P(o_\tau|s_\tau, \pi) - \log P(s_\tau|\pi)\big] \\ &= \underbrace{D_{KL}\big[Q(s_\tau|\pi)||P(s_\tau)\big]}_{\text{risk}} + \underbrace{\mathbb{E}_{Q(s_\tau)}\big[H(P(o_\tau|s_\tau))\big]}_{\text{ambiguity}}. \end{aligned} \tag{4}$$

A key aspect here is that the agent has prior beliefs about future states, which can be regarded as its goals. Mathematically, this means that the dependency on $\pi$ in the prior over states $P(s_\tau)$ in the KL term can be left out. This reflects that the agent wants to realize its preferred states of the world, independent of which policies it takes. The expected free energy inherently balances the

agent's drive towards its preferences, i.e. minimizing risk, with the expected uncertainty of the path towards the goal, i.e. reducing ambiguity.

The expected free energy is calculated for each future timestep the agent wants to consider and is then aggregated, after which the most likely sequence of actions is inferred through

$$P(\pi) = \sigma(-\gamma G(\pi)), \tag{5}$$

where $\sigma$ denotes the softmax function with temperature parameter $\gamma$, which transforms the expected free energy of policies into a categorical distribution over policies. This effectively casts planning as an inference problem, and beliefs over policies are proportional to the expected free energy. The softmax temperature $\gamma$ then reflects the confidence the agent has in its current beliefs over policies.

### *3.2. A hierarchical active inference model for navigation*

We now turn to the setting of navigation and mapping, and extend the generative model to explicitly incorporate pose and location information. In formal terms, we can visualize the proposed generative navigation model as a two level hierarchical model, as shown in Figure 2. At the lower level, the model reasons about the relation between actuator actions and sensory observations. At the higher level, it reasons about the relations between possible moves between locations and the possible corresponding states of the world and poses.

We can formalize this model mathematically, similar to the prototypical model of the previous section, in a Bayesian framework as follows. Considering the joint distribution over observations $o$, states $s$, actions $a$, locations $l$, poses $p$, moves $m$ and policy $\pi$ to be $P(\tilde{o}, \tilde{s}, \tilde{a}, \tilde{l}, \tilde{p}, \tilde{m}, \pi)$, where tildes again indicate that we are considering sequences of the corresponding variable. We can factorize the model according to:

$$P(\tilde{o}, \tilde{s}, \tilde{a}, \tilde{l}, \tilde{p}, \tilde{m}, \pi) = \underbrace{P(\tilde{o}, \tilde{s}_{i>0}, \tilde{a}, \tilde{p}_{i>0}, \pi | \tilde{l}, \tilde{s}_0, \tilde{p}_0)}_{\text{lower level}} \underbrace{P(\tilde{l}, \tilde{m}, \tilde{s}_0, \tilde{p}_0)}_{\text{higher level}}, \tag{6}$$

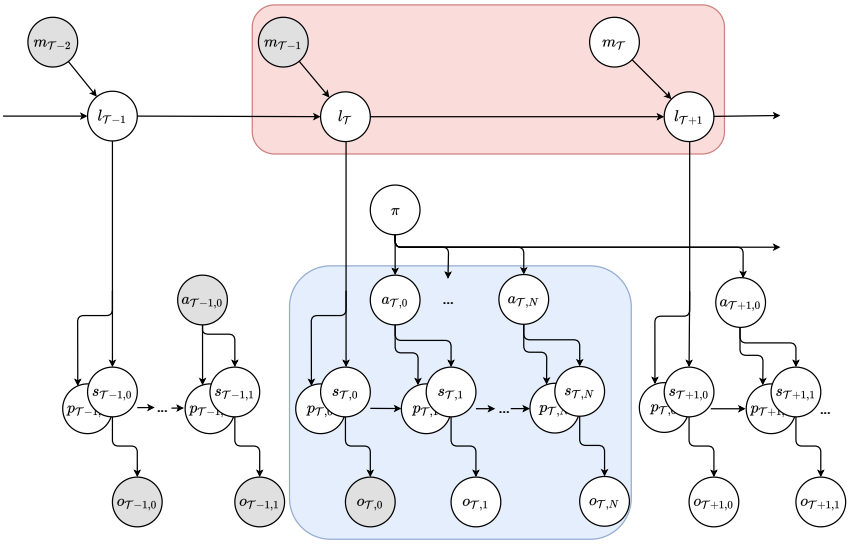



Figure 2: Navigation as a hierarchical generative model for active inference. At the lower level, highlighted in blue, the model entertains beliefs about hidden state $s_{\mathcal{T},t}$ and the agents pose $p_{\mathcal{T},t}$ at the current timestep $t$. Again, the hidden states $s$ give rise to an observations $o$, whereas the hidden states $s$ and pose $p$ are influenced by the previous state, pose and action or the higher level model incase of the initial states. At the higher level, highlighted in red, the agent reasons about locations $l$. The next location $l_{\mathcal{T}+1}$ will be determined by executing a move $m_{\mathcal{T}}$. Note that the higher level operates on a coarser timescale. The initial state $s_{\mathcal{T},0}$ and pose $p_{\mathcal{T},0}$ have a dependency on the current location $l_{\mathcal{T}}$, i.e. the initial state and pose are "observations" of the higher level model.

.

which contains a lower level and higher level part of the hierarchy. It is important to note that these two levels operate on a different time scale. Note that we use the notation $\tilde{s}_0$, $\tilde{p}_0$, $\tilde{s}_{i>0}$and $\tilde{p}_{i>0}$ to indicate the sequence of lower level initial states and lower level follow-up states respectively. The lower level operates on a fine-grained time scale $t$, and is responsible for lower level visual perception and path integration. The higher level operates on a coarser time scale $\mathcal{T}$, where a single step from $\mathcal{T}$ to $\mathcal{T}+1$ actually involves $N+1$ steps $0, 1, ..., N$ on the lower level. The current time step is denoted $t$ for the lower level or $\mathcal{T}$ for the higher level depending on the level we are currently considering. We will omit the $\mathcal{T}$ when we are only considering a single slice of the lower level. Although the entire sequences of variables cover the same time in the environment, the lower- and higher level models operate on different levels of detail. The higher level operates at a coarser timescale, indicating that multiple lower level time steps come to pass in one step of the higher level. Finally, in the higher level the global policy is omitted, every move is treated independent of each other. This hierarchical arrangement allows the agent to reason about its environment further ahead, both temporal and spatial.

We can further decompose these terms.

$$
\begin{aligned}
&P(\tilde{o},\tilde{s},\tilde{a},\tilde{l},\tilde{p},\tilde{m},\pi) = \\
&\qquad P(\pi)\prod_{\mathcal{T}>0} P(m_{\mathcal{T}})P(l_{\mathcal{T}}|l_{\mathcal{T}-1},m_{\mathcal{T}-1})P(s_{\mathcal{T},0}|l_{\mathcal{T}})P(p_{\mathcal{T},0}|l_{\mathcal{T}}) \\
&\qquad P(o_{\mathcal{T},0}|s_{\mathcal{T},0})\prod_{t\geq 1} P(s_{\mathcal{T},t}|s_{\mathcal{T},t-1},a_{\mathcal{T},t-1})P(p_{\mathcal{T},t}|p_{\mathcal{T},t-1},a_{\mathcal{T},t-1},s_{\mathcal{T},t}) \\
&\qquad\qquad P(o_{\mathcal{T},t}|s_{\mathcal{T},t})P(a_{\mathcal{T},t}|\pi)
\end{aligned}
\tag{7}
$$

From which we can see that the generative model decomposes into terms for both the lower level dynamics, i.e. how actions influence the next state and pose. Furthermore we see that in the lower level action selection is modelled through policy inference. Moreover, an observation likelihood is modelled. From the higher level part the low level initial states $\tilde{s}_0 = \{s_{0,0}, s_{1,0}, ...\}$ and $\tilde{p}_0 = \{p_{0,0}, p_{1,0}, ...\}$ emerge as likelihood terms, while the higher level dynamics over

$\tilde{l}$ remain similar to the prototypical active inference case.

### *3.3. Variational (expected) free energy of the hierarchical model*

Now that we have fully specified our generative model, we again turn the the variational (expected) free energy that will be minimized in active inference. Again, the agent infers posterior beliefs $Q(\tilde{s}, \tilde{p}, \tilde{l})$ over hidden states, poses and locations through free energy minimization. Applying the free energy functional to the hierarchical generative model, and using a mean-field approximation for $Q(\tilde{s}, \tilde{p}, \tilde{l}) = Q(\tilde{s})Q(\tilde{p})Q(\tilde{l})$ we get

$$\begin{aligned} F_{hier} &= \mathbb{E}_Q\big[\log Q(\tilde{s}, \tilde{p}, \tilde{l}) - \log P(\tilde{o}, \tilde{s}, \tilde{a}, \tilde{l}, \tilde{p}, \tilde{m})\big] \\ &= \mathbb{E}_Q\big[\log Q(\tilde{s}, \tilde{p}) + \log Q(\tilde{l}) - \log P(\tilde{o}, \tilde{s}, \tilde{a}, \tilde{l}, \tilde{p}, \tilde{m})\big] \\ &= \mathbb{E}_Q\big[\log Q(\tilde{s}, \tilde{p}) - \log P(\tilde{o}, \tilde{s}_{i>0}, \tilde{a}, \tilde{p}_{i>0}|\tilde{l}, \tilde{s}_0, \tilde{p}_0) + \log Q(\tilde{l}) - \log P(\tilde{l}, \tilde{m}, \tilde{s}_0, \tilde{p}_0)\big] \\ &= F_{low} + F_{high}. \end{aligned} \tag{8}$$

This falls apart into a term for the lower level and the higher level of the hierarchy, which we can further unpack. For the higher level we get

$$\begin{aligned} F_{high} &= \mathbb{E}_Q\big[\log Q(\tilde{l}) - \log P(\tilde{l}, \tilde{m}, \tilde{s}_0, \tilde{p}_0)\big] \\ &= \sum_{\mathcal{T}} \underbrace{D_{KL}\big[Q(l_{\mathcal{T}})||P(l_T|l_{\mathcal{T}-1}, m_{\mathcal{T}-1})\big]}_{\text{localization complexity}} \\ &\qquad + \underbrace{\mathbb{E}_Q\big[-\log P(s_{\mathcal{T},0}|l_{\mathcal{T}}) - \log P(p_{\mathcal{T},0}|l_{\mathcal{T}})\big]}_{\text{localization accuracy}}. \end{aligned} \tag{9}$$

Here we plugged in the higher level part of Equation 7 and assume that our moves from the past are observed without uncertainty. Again we find a complexity and accuracy term, meaning that the agent searches the least complex location description that explains the matching hidden states and poses. In practice this boils down to the agent building a mental map of its environment, and localizing itself therein. Hence, localization and mapping naturally follow from the free energy minimization.

Similarly, plugging the lower level part of Equation 7 over all possible lower level time values, into $F_{low}$ we get

$$\begin{aligned} F_{low} &= \mathbb{E}_Q\big[\log Q(\tilde{s},\tilde{p}) - \log P(\tilde{o},\tilde{s},\tilde{a},\tilde{p})\big] \\ &= \sum_t \underbrace{\mathbb{E}_Q\big[\log Q(p_t) - \log P(p_t|p_{t-1},a_{t-1},s_t)\big]}_{\text{pose estimation}} \\ &\quad + \underbrace{D_{KL}\big[Q(s_t)||P(s_t|s_{t-1},a_{t-1})\big]}_{\text{hidden state complexity}} \\ &\quad + \underbrace{\mathbb{E}_Q\big[-\log P(o_t|s_t)\big]}_{\text{observation accuracy}}. \end{aligned} \tag{10}$$

Now we recover the same complexity and accuracy terms as Equation 2, but complemented with a pose estimation term. This can be interpreted as the agent engaging in visual perception on the one hand, and path integration and odometry estimation on the other hand. We omitted the first time index from the variables as we only consider one tick of the higher level model. Note that we considered the sum over all $t$ for which $\tilde{s}_{i>0}$ is applicable.

When considering future time steps $\tau$, the variational free energy becomes an expected free energy, which again unfolds to a term for each level in the hierarchy. For the higher level this yields:

$$\begin{aligned} G_{high}(\tilde{m},\tau) &= \mathbb{E}_Q\big[\log Q(l_\tau|\tilde{m}) - P(s_{\tau,0},p_{\tau,0},l_\tau|\tilde{m})\big] \\ &= \underbrace{D_{KL}\big[Q(l_\tau|\tilde{m})||P(l_\tau)\big]}_{\text{reach goal location}} + \underbrace{\mathbb{E}_{Q(l_\tau)}\big[H(P(p_{\tau,0}|l_\tau)) + H(P(s_{\tau,0}|l_\tau))\big]}_{\text{route ambiguity}}. \end{aligned} \tag{11}$$

Similar to Equation 4, this unpacks in a risk term to reach a prior goal location $P(l_\tau)$, and an ambiguity term. Intuitively this means the agent selects a route to its goal location with the lowest ambiguity. Hence, if the agent operates in a completely static environment without uncertainty, this basically becomes equivalent to shortest path planning.

For the lower level we get:

$$
\begin{aligned}
G_{low}(\pi,\tau) &= \mathbb{E}_Q\big[\log Q(s_\tau,p_\tau|\pi) - \log P(o_\tau,s_\tau,p_\tau|\pi)\big] \\
&= D_{KL}\big[Q(s_\tau,p_\tau|\pi)||P(s_\tau,p_\tau)\big] + \mathbb{E}_{Q(s_\tau)}\big[H(P(o_\tau|s_\tau))\big] \\
&\approx \underbrace{D_{KL}\big[Q(s_\tau,p_\tau|\pi)||Q(s_{T+1},p_{T+1}|l_T,m_T)\big]}_{\text{short term goal from higher level}} + \underbrace{\mathbb{E}_{Q(s_\tau)}\big[H(P(o_\tau|s_\tau))\big]}_{\text{observation ambiguity}}.
\end{aligned}
\tag{12}
$$

Again this unfolds in a risk and ambiguity term, now encouraging the agent to reach a short term preferred state and pose, while minimizing the entropy of the expected observations. Crucially, here the prior preference $P(s_\tau, p_\tau)$ is provided by the higher level plan. Concretely, we use $Q(s_{\mathcal{T}+1,0}, p_{\mathcal{T}+1,0}|l_\mathcal{T}, m_\mathcal{T})$, which depicts the expected initial state and pose when reaching the next location $l_{\mathcal{T}+1}$, assuming we are following the first move $m_T$ of $\tilde{m}$ that minimizes $G_{high}$. Note that both the lower level policy and the higher level moves are modelled as independent in the generative model, even though that in planning they affect each other through the process of inferring actions according to the expected free energy $G$. The preferred higher level states guide the selection of higher level moves, which in turn set the preferred lower-level states, leading the to the inference of the optimal lower level policy $\pi$.

So basically we recover a two stage navigation process, where at the higher level the agent plans a route of moves visiting a sequence of locations up to the goal location, and at the lower level the agent infers the actions that will bring it to the next location on that route.

## 4. Active inference SLAM

Now that we have established our generative model and the variational free energy optimization objectives, we present how to instantiate such a model in silico for navigation on a real robot with camera input. In this case, the observations are the raw pixels from the camera sensor, and actions consist of the twist commands specifying the linear and angular velocity $v_l$ and $v_a$. Concretely,

the agent requires installation of the various dynamics and likelihood models, as well as the approximate posterior models over states, poses and locations. For this, we draw inspiration from previous work on generative models [12], as well as a biologically-inspired SLAM algorithm [13].

### *4.1. Visual perception*

For the lower level perception system, the agent needs to infer beliefs over a hidden state space $s$, given pixel observations $o$. As it is hard to specify such a mapping by hand, we turn to deep learning methods to learn such a representation from data. Following the example of earlier related techniques [11, 12, 54, 55], we model the likelihood $P(o_t|s_t)$, state dynamics $P(s_t|s_{t-1}, a_{t-1})$ and approximate posterior $Q(s_t)$ respectively by three neural networks $p_\xi(o_t|s_t)$, $p_\theta(s_t|s_{t-1}, a_{t-1})$ and $q_\phi(s_t|s_{t-1}, a_{t-1}, o_t)$.

The likelihood model $p_\xi$ reconstructs the current observation from the current state estimate. The state dynamics model $p_\theta$ takes the previous state estimate and action as input in order to get a new prior state estimate. Finally, the posterior model $q_\phi$ takes the same inputs but also utilizes the current observation as an extra input to get the posterior estimate on the state. Note that this recurrent nature of both the prior and posterior models allow the model to capture temporal relations in the observation space. All three of these networks output a mean and variance of a multivariate Gaussian distribution with diagonal covariance matrix. Finally, it is worth mentioning that only the prior and posterior models are strictly necessary at runtime for planning and inference. The likelihood model aids the training process by relating the agents state space to real world observations. This model can of-course also be used to visualize the agents imaginations at runtime.

Given a dataset of sequences of actions and observations, we can then train these neural networks end-to-end by minimizing the corresponding terms of the variational free energy of Equation 10. This boils down the (negative) evidence lower bound (ELBO) that is well known from the Variational Auto Encoder

(VAE) [52, 53].

$$\mathcal{L} = \sum_t D_{KL}\big[q_\phi(s_t|s_{t-1}, a_{t-1}, o_t)||p_\theta(s_t|s_{t-1}, a_{t-1})\big] - \log p_\xi(o_t|s_t) \tag{13}$$

Hence, we can train our neural networks accordingly, using the re-parameterisation trick to backpropagate gradients from $p_\xi$ to $q_\phi$.

### *4.2. Path integration*

The second part of the lower level generative model involves inferring beliefs about the agents pose. We specify the pose as $(x, y, \theta)$, with $x$ and $y$ a position in Cartesian space, and $\theta$ the heading or rotation around the z-axis. For the pose dynamics model $P(p_t|p_{t-1}, a_{t-1}, s_t)$ we simply estimate the odometry by integrating the action velocities over the time interval $\Delta t$:

$$\begin{aligned} \theta_t &= \theta_{t-1} + v_a \Delta t \\ x_t &= x_{t-1} + v_l \Delta t \cos\theta_t \\ y_t &= y_{t-1} + v_l \Delta t \sin\theta_t \end{aligned} \tag{14}$$

The posterior distribution $Q(p_t)$ is represented by a Continuous Attractor Network (CAN) [13]. The CAN is implemented as a 3D cube that wraps around the edges with dimensions $x$, $y$ and $\theta$. Activity is injected based on the estimated odometry and exhibits locally excitatory, globally inhibitory connectivity yielding the following energy distribution [56]:

$$\epsilon_{\Delta x, \Delta y, \Delta\theta} = \exp\frac{-\Delta x^2 - \Delta y^2}{k_p^{exc}} \exp\frac{-\Delta\theta^2}{k_d^{exc}} - \exp\frac{-\Delta x^2 - \Delta y^2}{k_p^{inh}} \exp\frac{-\Delta\theta^2}{k_d^{inh}}. \tag{15}$$

Here $k_d$ and $k_p$ are the variance constants for place and direction. Pose estimates based solely on proprioceptive information will eventually lead to estimation errors and drift. Therefore, there is an extra excitatory link based on the current belief state $s_t$. We keep an episodic memory of encountered state-pose pairs $(s, p)$, and if the current state $s_t$ matches with one of the stored pairs, we inject

energy on the corresponding pose in the CAN. Hence the CAN closely resembles grid cells in the rodent medial entorhinal cortex, with a similar torus layout [57]. Moreover, the state-pose pairs $(s, p)$ will form the basis for building a topological map for the higher level generative model.

### *4.3. Localization and mapping*

At the higher level, the agent needs to infer a posterior belief $Q(l_{\mathcal{T}})$ about its current location, and be equipped with a dynamics model $P(l_{\mathcal{T}}|l_{\mathcal{T}-1}, m_{\mathcal{T}-1})$, i.e. which moves lead to which locations. To this end, we build an experience map as a graph that keeps track of the location of the agent at the higher abstraction level [13]. Each node in the graph defines a unique location $l$, together with corresponding state-pose pairs $(s, p)$. Links between nodes indicate that the agent can traverse between those two nodes and are weighted with the Euclidean distance between the two nodes in the map. The dynamics model $P(l_{\mathcal{T}}|l_{\mathcal{T}-1}, m_{\mathcal{T}-1})$ can then be deduced from the adjacency matrix of the graph, where possible moves are the outbound links of the experience node. For posterior belief distribution $Q(l_{\mathcal{T}}|s_t, p_t)$ on the other hand, we assign probability inversely proportional to some distance function $D$ between the current state $s_t$ and pose $p_t$ and the $(s, p)$ pairs belonging to the different experience map nodes. In particular, we use the cosine similarity to compare states and Euclidian distance to compare poses.

When none of the experience map nodes matches the current state and pose, i.e. $D[(s_t, p_t), (s^i, p^i)] > \delta \quad \forall i$, where $s^i$ and $p^i$ are the state and pose associated with the $i^{\text{th}}$ node in the map, a new experience is added to the map[2], linked to

[2]Although not the focus of this work, adding a new experience to the generative model addresses an important issue in active inference and modelling in general; namely, the issue of structure learning. Here, we implement structure learning in a straightforward way by adding model nodes or components if they improve accuracy in relation to complexity cost. Formal procedures for this kind of structure learning would be seen in the light of Bayesian model comparison using variational free energy as a bound on model evidence. This aspect of model optimisation is closely related to nonparametric Bayes and provides yet another hierarchical

the previously visited node. This way, the map is gradually expanded as new area is explored. The threshold value $\delta$ will then influence the granularity of the map, and is determined empirically. A higher $\delta$ will result in a coarser map, but physically different locations might be erroneously mapped to the same experience node.

Also, due to odometry integration drift, matching experiences will not have exactly the same associated pose. These displacement errors are distributed throughout the graph by use of graph relaxation, shifting the stored pose according to

$$\Delta p^i = \frac{1}{2}\Big[ \sum_{j=1}^{inbound} (p^j - p^i - \Delta p^{ij}) + \sum_{k=1}^{outbound} (p^k - p^i - \Delta p^{ki})\Big]. \qquad (16)$$

This enforces the map to be topologically consistent, even after loop closures, which is often challenging in metric SLAM systems [1].

### *4.4. Navigation*

To navigate the environment, the agent now has to infer future moves and actions, which minimize $G_{high}$ (Eq. 11) and $G_{low}$ (Eq. 12) respectively. In our hierarchical model this is a two stage process. First, given a goal location $l_\tau$, the higher level will infer a sequence of moves $\tilde{m}$ that minimize $G_{high}$. Next, we use the higher level dynamics and likelihood model to get $Q(s_{\mathcal{T}+1,0}, p_{\mathcal{T}+1,0}|l_\mathcal{T}, m_\mathcal{T})$, which is basically the belief over states and poses that we want to attain at $\mathcal{T}+1$. This is then used as preferred prior probability for the lower level model. Finally, we estimate $G_{low}$ and infer the lower level policy $\pi$ to follow. Actions are then chosen from $P(a|\pi)$, which is in our case a deterministic mapping. As the agent executes the action, new observations are obtained, after which the agent updates its posterior beliefs and the process repeats.

In the higher level model, a sequence of moves $\tilde{m}$ basically corresponds to a path in the experience map graph. The goal-directed term in $G_{high}$ reduces

level that may be an important metaphor for hierarchical structure learning when exposed to a new environment - or, indeed, in developmental neurorobotics.

to the Euclidian distance between the visited and the goal location in the map. The ambiguity terms can be estimated by looking at the variance of the $(s, p)$ pairs associated with the visited experience nodes. The optimal route can then easily be found using a shortest path algorithm with $G_{high}$ as node weights.

In the lower level model, we adopt the approach proposed in [12] for estimating $G_{low}$ through Monte Carlo sampling. For a discrete number of policies (i.e. turn left, turn right, move forward), we sample $N$ trajectories to approximate $Q(s_\tau, p_\tau|\pi)$. From these samples we can then estimate both the KL divergence term and the ambiguity term.

## 5. Experiments

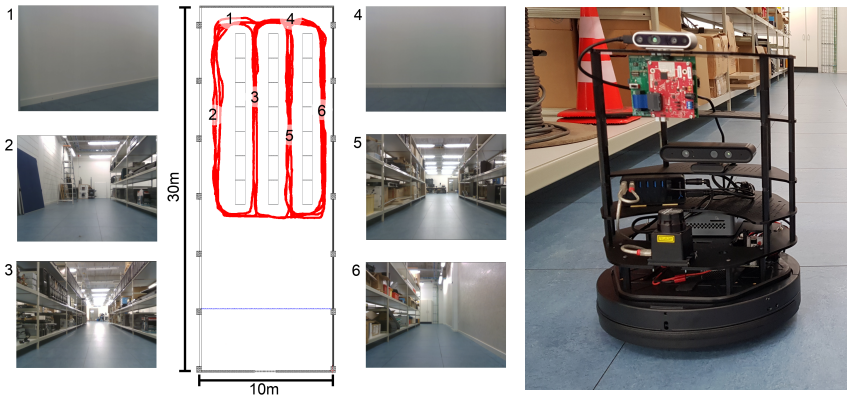
In order to be able to validate the approach laid out in Section 4, we conducted experiments on a real world robot navigating a warehouse-like environment. We will first lay out the experimental setup and implementation details, before we present the results of the hierarchical model.

### *5.1. Setup*

We collected data by teleoperating a mobile robot through the IIoT lab which has a warehouse-style layout. The lab consists of 4 aisles with racks as shown in Figure 3(a). This is a very challenging environment for visual SLAM, as the visual inpu ts can be very similar on various locations. The robot, shown in Figure 3(b), is a Turtlebot 2i equipped with various sensors mounted, such as an Intel Realsense D435 camera, a Hokuyo lidar, TI mmWave radar and an Astra 3D camera. In this experiment, we only use the RGB data of the Realsense in the hierarchical model, and the Astra RGBD data was used to establish a baseline with a metric visual SLAM system. The entire dataset consists of 65400 datapoints sampled at 100 ms intervals. For a more detailed description of the dataset, we refer the reader to [58].

| | Layer | Neurons/Filters | stride |
|---|---|---|---|
| $q_\phi$ | Convolutional | 32 | 2 |
| | Convolutional | 32 | 1 |
| | Convolutional | 64 | 2 |
| | Convolutional | 64 | 1 |
| | Convolutional | 128 | 2 |
| | Convolutional | 128 | 1 |
| | Convolutional | 256 | 2 |
| | Convolutional | 256 | 1 |
| | Concat | | |
| | Linear | 512 | |
| | Linear | 2 x 32 | |
| $p_\xi$ | Linear | 256 x 15 x 20 | |
| | Convolutional | 256 | 1 |
| | Upsample | | |
| | Convolutional | 256 | 1 |
| | Convolutional | 128 | 1 |
| | Upsample | | |
| | Convolutional | 128 | 1 |
| | Convolutional | 64 | 1 |
| | Upsample | | |
| | Convolutional | 64 | 1 |
| | Convolutional | 32 | 1 |
| | Upsample | | |
| | Convolutional | 32 | 1 |
| $p_\theta$ | LSTM cell | 512 | |
| | Linear | 2 x 32 | |

Table 1: Neural network parameterization. The posterior model $q_\phi$ processes images through a convolutional pipeline, after which state and action vectors are concatenated for fully connected layers. The likelihood model $p_\xi$ reconstructs image data from a state vector. The dynamics model $p_\theta$ consists of an LSTM cell followed by a fully connected layer. All layers (except the last one) have LeakyReLU activations. It's worth noting that these models implement a form of amortization, i.e. they learn to infer.




(a) Lab

(b) Turtlebot

Figure 3: The dataset consists of trajectories recorded around racks in a warehouse lab. The red lines depict the robot trajectories, with visual impressions on six different locations. All data is captured with a Turtlebot 2i equipped with an Intel Realsense D435 camera.

### *5.2. Implementation*

For the visual perception pipeline we instantiate the three neural networks as follows. The posterior neural network $q_\phi$ is implemented as a convolutional pipeline at the end of which the previous state and action values are concatenated before being processed in a final fully connected layer with variational output (i.e. $\mu$ and $\sigma$). The likelihood model $p_\xi$ performs the inverse operation of the posterior model, generating an image from a state vector. In the likelihood model we utilize shape preserving convolutions by settings stride and padding to 1, while using and nearest-neighbor interpolation between every other convolutional layer to upsample to the required image resolution. Finally the dynamics model $p_\theta$ consists of an LSTM cell [59] and a fully connected layer with variational output. Table 1 shows the exact parameterization for each neural network layer in the architecture. The input resolution of the images is 320x240, whereas our state space has 32 dimensions. As action input we use the raw twist command containing linear and angular velocities.

We split our dataset into a train set and test set, and train for 50 epochs using the ADAM optimizer [60] with initial learning rate 1e-4. During training,

we randomly sample minibatches of size 8 of subsequences of length 8, mainly due to memory restrictions on our GPU. We skip every other timestep in the dataset, such that the neural network learns on 200ms $\times$ 8 = 1.6s time horizons.

The path integration and experience mapping are implemented based on the Milford's RatSLAM implementation [56]. For the pose CAN we instantiate a $[61 \times 61 \times 36]$ cube for tracking the $(x, y, \theta)$ pose. For matching experiences we use $1 - \frac{s_1 s_2}{\|s_1\|\|s_2\|}$ to find matching state vectors if the distance is below a threshold of $\delta = 0.04$. For poses we declare a match if the Euclidian distance is below the threshold of $\delta = 6$. If no match is found, a new experience is added to the map.

### *5.3. Results*

We evaluate the resulting hierarchical model to resolve the following research questions:

- Does the lower level model learn accurate representations for inference and prediction by minimizing free energy?
- Can these representations be used in the hierarchical model for generating topological maps of the environment?
- Does the system infer sensible moves and actions by minimizing expected free energy to engage in navigation?

#### *5.3.1. Lower level perception*

Figure 4 shows the results when inferring states and poses for a sequence of the test set using our lower level model. The top row shows the ground-truth observation as captured by the camera. The second and third row show the model reconstructions and corresponding state space values. One can see how the likelihood model is fairly good at reconstructing the ground truth from the 32 dimensional state vector. Also note that at the start in the reconstruction a tripod appears, which was there in some of the train sequences, but not in

term.png

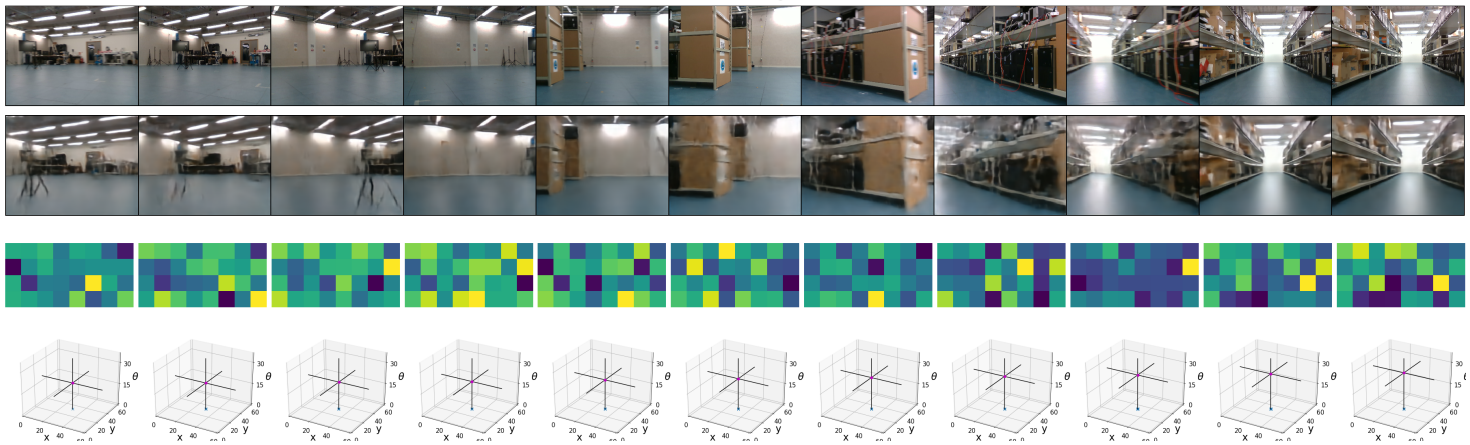


Figure 4: Sequences of the ground-truth observations (first row), reconstructed observations (second row), corresponding latent space samples (third row) and pose cube activation (bottom row). The robot is turning towards the left in the sequence, as can be seen from both the sequence of observations as well as the shift towards higher $\theta$ values in the pose cube. The latent space samples are normalized for visualization purposes, with yellow indicating a high activation.

the test sequence. The final row shows the corresponding pose CAN activation. One notices how the activity packet shifts upwards in $\theta$ direction as the robot is turning left. Note that we subsampled images over a half minute period, to clearly show the dynamics, whereas the model is actually processing an observation every 200ms.

Because we train a separate dynamics model $p_\theta$ to predict next states given actions, the model can also predict the consequences of its actions. By sampling trajectories from the dynamics model, and visualizing the corresponding likelihoods we can get insights of the dynamics it learnt. Figure 6 (d) shows imaginary rollouts for either turning left, turning right or moving straight ahead. One can clearly see that the model successfully captures short term dynamics. It is clear that the robot is turning in the aisle when turning left, or moving closer towards the rack when going straight. On the other hand, we also find that the reconstructions becomes more and more blurry the further we predict in time, especially for the turning left or right scenarios. This can be attributed to the fact that on the one hand the model is only trained on subsequences of a limited length of 8, and on the other hand that in the dataset the robot is much more often moving straight ahead compared to turning. Also, when moving straight ahead all information about the next frame is basically in view, whereas when turning you have a harder time imagining what comes next. This shows some of the challenges of using deep learned state space models for long term prediction, in that these typically become more blurry and temporally inconsistent as the time horizon grows bigger. Our hierarchical model addresses this by operating at a coarser time scale, as well as by using what is effectively an episodic memory in a graphical model to accommodate long-range dependencies.

*5.3.2. Localization and mapping*

To evaluate the mapping, we create a map with our system by feeding it a long, 30 minute sequence of the robot navigating various aisles. The resulting map is shown on Figure 5. We can see that, despite the visual ambiguity of different locations, our system is able to produce a consistent topological map.

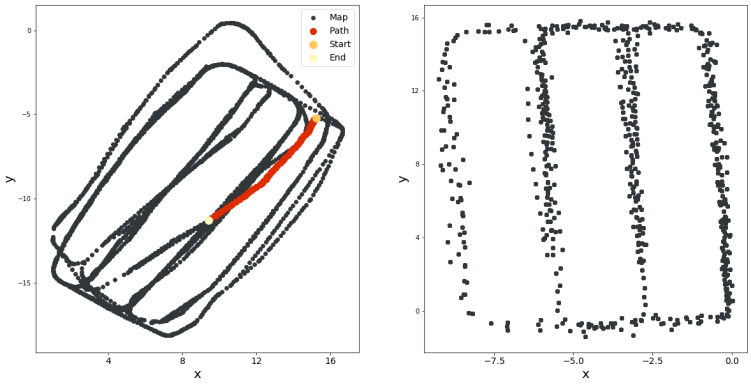



Figure 5: The generated experience map (left) and a comparison map extracted from localization with RTAB-Map [61]. The experience map is topological, while the RTAB-Map map is metric. One clearly observes how the different aisles are mapped consistently and the topological map matches the metric map up to an arbitrary rotation and scale factor.

It correctly maps the four different aisles, and flawlessly closes all loops in the map.

For reference, we added the localization over time of the robot using RTAB-map [61], a real-time appearance based SLAM algorithm using RGB-D data. Note that with RTAB-map we had to map the different aisles separately in 3 runs, and then combine the maps offline. Otherwise the map would collapse due to the visual aliasing mapping different aisles onto each other. Also, as we compress the images to latent vectors of size 32, our map only takes up about 2MB in memory, whereas the resulting RTAB-map point cloud requires over 850MB.

### *5.3.3. Navigation and planning*

To illustrate navigation as inference, we we equip the robot with a goal location and infer the actions as described in Section 4. Figure 6 shows visualizations of the current state (a) and the goal location (c).

First, we use the experience map to find a higher level path from the destination to the goal, as shown in Figure 5. The first location in this path is then

(a) Current state (b) Short term goal (c) Long term goal

(d) Imaginary policies: *go-left (top), go-right (middle) and go-straight (bottom)*

Figure 6: Imaginary policies (d) to achieve the short term goal (b) from the current state (a) as part of the trajectory 5 that leads to the long term goal (c). When calculating $G_{low}$ and evaluating $P(\pi)$, this results in a probability close to 1 to take the policy *go-left*.

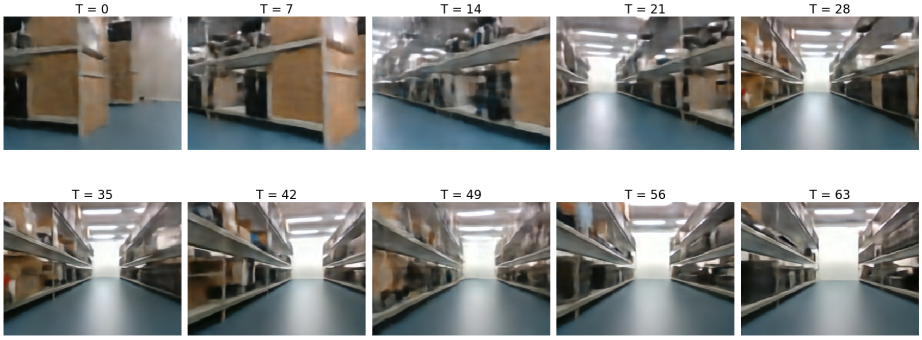



Figure 7: Different goals for the long term path of Figure 5. The lower level dynamics model will take the corresponding states as preferred state goals sequentially.

used as preferred state for the lower level model, as visualized in Figure 6 (b).

To evaluate $G_{low}$ we sample N=10 trajectories for each considered policy (i.e. turn left, turn right, go straight ahead) for a time horizon of 10 timesteps in the future. We have visualized one imaginary trajectory for each policy in Figure 6 (d). The agent correctly decides that turning left will bring it towards its current preferred state. In this particular case the ambiguity term plays little role, as the expected free energy is dominated by the goal-directed term.

Finally we visualize the complete planned trajectory by interpolating in state space between the experience nodes on the path in Figure 7. Note that in comparison with the imaginary rollouts in Figure 6 (d), the resulting observations from the trajectory are much more crisp and temporally consistent even for these long time horizons. This illustrates the benefit of using the hierarchical model, and building a coarse grained map as a graph structure for this navigation task.

## 6. Discussion

Active inference is a process theory of the brain that tries to explain autonomous behavior [7]. In Section 2, we unpacked the active inference formulation focused on navigation. We introduced a hierarchical generative model, which models visual inputs, poses and locations similar to the neural correlates

that contain the visual cortex, grid cells and place cells. From our active inference treatment, it followed that subtasks often identified in robotics such as path integration, localization and mapping naturally emerge from minimizing free energy.

We implemented the lower level visual perception part using deep neural networks, learning state representations purely from data by minimizing the free energy objective. Although we found that this yields compact latent representations that are suited for indoor navigation, we also saw that the temporal consistency is rather short-term. It is to be further investigated whether this is due to the limited time horizon at train time, or a fundamental limitation of the architecture.

For the higher level we opted for building an experience map, which can be seen as a graph structured episodic memory. This is also biologically plausible as the hippocampus plays an important role in episodic memory [62]. However, the free energy minimizing process is less outspoken in this case. To some extent, the process of adding new nodes to the map can be interpreted as improving accuracy, whereas the graph relaxation algorithm acts a complexity reducing regularizer.

A major shortcoming of our current approach, however, is the reliance on pre-training the lower level dynamics model. It is necessary to create a sufficient amount of data samples by manually driving the robot around in order to collect enough examples to train the dynamics model for the environment. Also, navigation is limited to the few trajectories that were tele-operated to build the map. Ideally, one would be able to build a system that constantly navigates, explores, maps and learns its environment. To integrate exploration, future work is needed to include novelty seeking terms to the formulation by also incorporating uncertainties on the (hyper)parameters and inferring beliefs thereof [50].

Furthermore this reliance on a pre-trained lower level dynamics model currently precludes any form of lifelong learning, an important aspect of true autonomy. The way the lower level and higher level model cooperate forces the higher

level model to reconsider and reevaluate all stored state templates. A possible way to achieve some form of life-long learning is to introduce a sleep-wake cycle, where the agent retrains and updates stored experiences during charging cycles, i.e. “sleep” and collects novel data during operation, i.e. ”wake”. However, if the visual appearance and the internal dynamics of the world remain fairly consistent, the model is capable of coping with geometric changes in the world such as changes to room layouts and objects, by virtue of being related to the RatSLAM algorithm [63, 64].

In recent work [65], a recurrent neural network was trained to predict sequences of visual inputs from (the latent space) of variational autoencoders. A natural mapping from egocentric information to an allocentric spatial reference frame was observed, including the induction of specialized units with response properties similar to head direction, place, band, landmark, boundary vector, and egocentric boundary cells. These results are in line with the Tolman-Eichenbaum machine [25], including reliable cell remapping, thus enabling transfer learning across episodes. Suggesting that a learnt latent hidden state variable could be sufficient for the lower level generative model. However, explicitly encoding pose information in a separate random variable as in this work allows for the resulting topological maps to be more human interpretable, with an intuitive notion of spatial space. Nonetheless it is an interesting research direction to see whether navigable maps can be obtained without explicit pose encoding, but using only implicit state representations.

Compared to traditional metric SLAM systems, our learnt state representations offers the benefit of reducing the dimensionality, and better cope with perceptual aliasing. Our results show that indeed our system was able to correctly perform loop closures for a long running sequence in a challenging environment with visual ambiguity. The representation was also able to cope with small changes in the environment, as certain tripods and ladders were changed location in between the different recordings. The generated maps, however should not be considered as exact representations of the real – Euclidean – world, as they represent a topological view of the environment.

Finally the expected free energy is shown to be also an effective method for inferring both the higher level moves and lower level actions, selecting a trajectory through the experience map and being able to execute it at the lower level. Navigation becomes a two-stage inference process, which closely resembles manually engineered navigation routines using global and local cost maps. This allows for a holistic approach to navigation, the same mechanism drives the decision making process at both levels. This two-stage process complies with the findings of [66], which indicates that human planning relies on a model based prediction mechanism and that the limiting factor in human planning might be the planning horizon. Furthermore in [67, 68] they make the same distinction between lower-level motor commands and human path planning.

### *6.1. Related Work*

Active inference has been applied to a multitude of problem domains ranging from the classical thermostat [69] and animal foraging [70] to perception [71], robotic control [72] and reinforcement learning [73] among others. Many of these applications of active inference, however, are applied only to a simplified simulated version of the real problem, allowing for hand crafted prior, posterior and likelihood models as well as sate space definitions. More recent approaches to active inference have focused on integrating deep learning in various ways in order to scale active inference [11, 12, 73, 74, 75].

Ours is not the first approach to casting navigation as an active inference process, earlier work [9, 10] however, only considered small grid worlds and maps thereof. In Kaplan and Friston [9], the authors consider a hand-crafted active inference model with a shallow policy of two moves. On the other hand Friston et al. [10] casted the navigation problem to be tackled with a sophisticated active inference model. Sophistication in this context indicates the usage of search trees in order to infer the optimal action according to the expected free energy $G$. Our approach however differs in the utilization of a hierarchical model in conjunction with a learnable dynamics model, allowing our model to scale beyond grid-worlds to the real world.

The field of robotics has considered the SLAM problem for almost half a century [76]. Over this period many different approaches to mapping and navigating evolved, ranging from engineered approaches [77, 78] to bio-inspired approaches [13, 79, 80]. These bio-inspired SLAM algorithms also rely on a pose-CAN and experience map, however they traditionally use classical feature extractors to achieve the view-template matching. Traditional SLAM approaches often rely on different metrics to achieve the map building and localization and separately the navigation. In contrast active inference and the free energy principle allows for a holistic treatment of these concepts in a biologically plausible way. There has been some research recently in the application of deep learning on (visual)SLAM [81, 82, 83], however these approaches have typically focused on improving the perception aspect of SLAM, while leaving the mapping and navigation aspects untouched.

## 7. Conclusion

In this paper, we have shown how navigation can be cast as active inference using a hierarchical generative model. By unpacking the variational free energy and expected free energy terms, we presented how concepts like visual perception, path integration, localization, mapping and navigation naturally emerge. Moreover, we have shown how such a hierarchical model can be instantiated using recent advances in deep learning as well as other bio-inspired SLAM approaches. We validated our approach on real-world data captured by a mobile robot in a warehouse setting. We think that this research direction might offer new insights both on how navigation works in the mammal brain, as well as how to scale active inference to real-world applications.

## Acknowledgements

Ozan Çatal is funded by a Ph.D. grant of the Flanders Research Foundation (FWO). This research received funding from the "AI Flanders" program of the Flemish government.

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, Trans. Rob. 32 (6) (2016) 13091332.

[2] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. M. Chan, Q. She, Are we ready for service robots? the openloris-scene datasets for lifelong slam, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3139–3145.

[3] K. Knight, Navigation: from animal behaviour to guiding principles, Journal of Experimental Biology 222.

[4] I. van der Ham, M. Claessens, A. Evers, M. van der Kuil, Large-scale assessment of human navigation ability across the lifespan, Scientific Reports 10 (3299).

[5] M. Peer, I. K. Brunec, N. S. Newcombe, R. A. Epstein, Structuring knowledge with cognitive maps and cognitive graphs, Trends in Cognitive Sciences 25 (1) (2021) 37 – 54.

[6] R. V. Rikhye, N. Gothoskar, J. S. Guntupalli, A. Dedieu, M. Lázaro-Gredilla, D. George, Learning cognitive maps as structured graphs for vicarious evaluation, bioRxiv.

[7] K. J. Friston, Life as we know it, Journal of the Royal Society Interface.

[8] K. Friston, The free-energy principle: a unified brain theory?, Nat. Rev. Neurosci. 11 (2) (2010) 127–138.

[9] R. Kaplan, K. Friston, Planning and navigation as active inference, Biological Cybernetics 112 (7).

[10] K. Friston, L. Da Costa, D. Hafner, C. Hesp, T. Parr, Sophisticated inference.

[11] O. Çatal, S. Wauthier, T. Verbelen, C. D. Boom, B. Dhoedt, Deep active inference for autonomous robot navigation, in: The Bridging AI and Cognitive Science (BAICS) Workshop, ICLR, 2020.

[12] O. Çatal, S. Wauthier, C. De Boom, T. Verbelen, B. Dhoedt, Learning generative state space models for active inference, Frontiers in Computational Neuroscience 14 (2020) 103.

[13] M. J. Milford, G. F. Wyeth, D. Prasser, Ratslam: a hippocampal model for simultaneous localization and mapping, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, Vol. 1, 2004, pp. 403–408 Vol.1.

[14] G. Striedter, Principles of Brain Evolution, 1st Edition, Sinauer Associates is an imprint of Oxford University Press, Sunderland, Mass.

[15] J. A. Gray, The neuropsychology of anxiety : An inquiry into the functions of the septo-hippocampal system, 2nd Edition, Oxford psychology series ; no. 33, Oxford University Press, Oxford [England], 2000.

[16] S. M. Suryanarayana, J. Prez-Fernndez, B. Robertson, S. Grillner, The evolutionary origin of visual and somatosensory representation in the vertebrate pallium, Nature Ecology and Evolution 4 (2020) 639651.

[17] T. Feinberg, J. Mallatt, The evolutionary and genetic origins of consciousness in the cambrian period over 500 million years ago, Frontiers in Psychology 4 (2013) 667. doi:10.3389/fpsyg.2013.00667.
URL https://www.frontiersin.org/article/10.3389/fpsyg.2013.00667

[18] A. Honkanen, A. Adden, J. da S. Freitas, S. Heinze, the insect central complex and the neural basis of navigational strategies, j. exp. biol., vol, Suppl 1. doi:doi:10.1242/jeb.188854.

[19] J. OKeefe, L. Nadel, The Hippocampus as a Cognitive Map, Clarendon Press, Oxford, 1978.

[20] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, E. I. Moser, Microstructure of a spatial map in the entorhinal cortex, Nature 436 (2005) 801806.

[21] J. Hawkins, S. Blakeslee, On Intelligence, Times Books, USA, 2004.

[22] A. Lee, J. Hoy, A. Bonci, L. Wilbrecht, M. Stryker, C. M, Niell, identification of a brainstem circuit regulating visual cortical state in parallel with locomotion, neuron, vol 2 455466. doi:doi:10.1016/j.neuron.2014.06.031.

[23] H. Dannenberg, H. Lazaro, P. Nambiar, A. Hoyland, M. E, Hasselmo, effects of visual inputs on neural dynamics for coding of location and running speed in medial entorhinal cortex, elife, vol, 9 p. 62500. doi:doi:10.7554/eLife.62500.

[24] A. Sarel, A. Finkelstein, L. Las, N., vectorial representation of spatial goals in the hippocampus of bats, science, vol 6321 176180. doi:doi:10.1126/science.aak9589.

[25] J. C. Whittington, T. H. Muller, S. Mark, G. Chen, C. Barry, N. Burgess, T. E. Behrens, The tolman-eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation, Cell 183 (5) (2020) 1249 – 1263.e23.

[26] K. L. Stachenfeld, M. M. Botvinick, S. J. Gershman, The hippocampus as a predictive map, Nature Neuroscience 20 (2017) 16431653(2017).

[27] A. M. Zador, A critique of pure learning and what artificial neural networks can learn from animal brains, Nature Communications 10 (2019) 3770.

[28] A. Johnson, A. Redish, Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point, The Journal of neuroscience : the official journal of the Society for Neuroscience 27 (2007) 12176–89.

[29] K. Kay, J. Chung, M. Sosa, J. Schor, M. Karlsson, M. Larkin, D. Liu, L. Frank, Constant sub-second cycling between representations of possible futures in the hippocampus, Cell 180.

[30] H. Igata, Y. Ikegaya, T. Sasaki, Prioritized experience replays on a hippocampal predictive map for learning, Proceedings of the National Academy of Sciences 118.

[31] M. Mack, A. Preston, B. Love, Ventromedial prefrontal cortex compression during concept learning, Nat. Commun.

[32] M. Laubach, L. Amarante, M. Caetano, N. Horst, Reward signaling by the rodent medial frontal cortex, 2020. doi:10.1016/bs.irn.2020.11.012.

[33] K. Hoffman, M. Dragan, T. Leonard, C. Micheli, R. Montefusco-Siegmund, T. Valiante, Saccades during visual exploration align hippocampal 38 hz rhythms in human and non-human primates, Frontiers in Systems Neuroscience 7 (2013) 43.

[34] A. Johnson, Z. Varberg, J. Benhardus, A. Maahs, P. Schrater, The hippocampus and exploration: Dynamically evolving behavior and neural representations, Frontiers in human neuroscience 6 (2012) 216.

[35] A. Kikumoto, U. Mayr, Decoding hierarchical control of sequential behavior in oscillatory eeg activity, eLife 7.

[36] T. Parr, K. J. Friston, The discrete and continuous brain: From decisions to movement-and back again, Neural Comput. 30 (9) (2018) 23192347.

[37] D. S. Corneil, W. Gerstner, Attractor network dynamics enable preplay and rapid path planning in maze–like environments, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 28, 2015, pp. 1684–1692.

[38] J. Knierim, K. Zhang, Attractor dynamics of spatially correlated neural activity in the limbic system, Annual review of neuroscience 35 (2012) 267–85.

[39] J. X. Wang, Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Z. Leibo, D. Hassabis, M. Botvinick, Prefrontal cortex as a meta-reinforcement learning system, Nature Neuroscience 21 (2018) 860868.

[40] W. Clawson, A. F. Vicente, M. Ferraris, C. Bernard, D. Battaglia, P. Quilichini, Computing hubs in the hippocampus and cortex, Science Advances 5.

[41] B. Griffiths, G. Parish, F. Roux, S. Michelmann, M. Plas, L. Kolibius, R. Chelvarajah, D. Rollings, V. Sawlani, H. Hamer, S. Gollwitzer, G. Kreiselmeyer, B. Staresina, M. Wimber, S. Hanslmayr, Directional coupling of slow and fast hippocampal gamma with neocortical alpha/beta oscillations in human episodic memory`doi:10.1101/305698`.

[42] J. Karimi Abadchi, M. Nazari-Ahangarkolaee, S. Gattas, E. Bermudez Contreras, A. Luczak, B. Mcnaughton, M. Mohajerani, Spatiotemporal patterns of neocortical activity around hippocampal sharp-wave ripples, eLife 9.

[43] G. Buzski, D. Tingley, Space and time: The hippocampus as a sequence generator, Trends in Cognitive Sciences 22 (2018) 853–869.

[44] M. Shanahan, B. Baars, Applying global workspace theory to the frame problem, Cognition 98 (2006) 157–76.

[45] C. Baldassano, U. Hasson, K. A. Norman, Representation of real-world event schemas during narrative perception, Journal of Neuroscience 38 (45) (2018) 9689–9699.

[46] W. Shang, A. Trott, S. Zheng, C. Xiong, R. Socher, Learning world graphs to accelerate hierarchical reinforcement learning (2019). `arXiv:1907.00664`.

[47] M. Eppe, C. Gumbsch, M. Kerzel, P. D. H. Nguyen, M. V. Butz, S. Wermter, Hierarchical principles of embodied reinforcement learning: A review (2020). `arXiv:2012.10147`.

[48] G. Parascandolo, L. Buesing, J. Merel, L. Hasenclever, J. Aslanides, J. B. Hamrick, N. Heess, A. Neitz, T. Weber, Divide-and-conquer monte carlo tree search for goal-directed planning (2020). `arXiv:2004.11410`.

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

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

[51] J. M. Beal, Variational algorithms for approximate bayesian inference.

[52] D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: Proceedings of the 31st International Conference on Machine Learning (ICML), Vol. 32, 2014, pp. 1278–1286.

[53] D. P. Kingma, M. Welling, Auto-encoding variational bayes., CoRR abs/1312.6114.

[54] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, J. Davidson, Learning latent dynamics for planning from pixels, arXiv preprint arXiv:1811.04551.

[55] D. Ha, J. Schmidhuber, Recurrent world models facilitate policy evolution, in: Advances in Neural Information Processing Systems 31, Curran Associates, Inc., 2018, pp. 2451–2463, https://worldmodels.github.io. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolutio

[56] D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, M. Milford, Openratslam: an open source brain-based slam system, Autonomous Robots 34 (2013) 1–28.

[57] A. Guanella, D. Kiper, P. Verschure, A model of grid cells based on a twisted torus topology, International journal of neural systems 17 (2007) 231–40.

[58] O. Çatal, W. Jansen, T. Verbelen, B. Dhoedt, J. Steckel, Latentslam: unsupervised multi-sensor representation learning for localization and mapping,

in: 2021 International Conference on Robotics and Automation (ICRA), 2021 (In Press).

[59] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[60] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[61] M. Labb, F. Michaud, Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation, Journal of Field Robotics 36 (2) (2019) 416–446.

[62] N. J. Fortin, K. L. Agster, H. B. Eichenbaum, Critical role of the hippocampus in memory for sequences of events, Nature Neuroscience 5 (2002) 458–462.

[63] M. Milford, D. Prasser, G. Wyeth, Experience mapping: Producing spatially continuous environment representations using ratslam, Proceedings of the 2005 Australasian Conference on Robotics and Automation, ACRA 2005.

[64] M. Milford, G. Wyeth, Persistent navigation and mapping using a biologically inspired slam system, The International Journal of Robotics Research 29 (9) (2010) 1131–1153. arXiv:https://doi.org/10.1177/0278364909340592, doi:10.1177/0278364909340592.
URL https://doi.org/10.1177/0278364909340592

[65] B. Uria, B. Ibarz, A. Banino, V. Zambaldi, D. Kumaran, D. Hassabis, C. Barry, C. Blundell, The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains, bioRxiv.

[66] Y. Marghi, F. Towhidhkhah, S. Gharibzadeh, Human brain function in path planning: A task study, Cognitive Computing.

[67] O. Trullier, S. I. Wiener, A. Berthoz, J. A. Meyer, Biologically based artificial navigation systems review and prospects, Progress in Neurobiology 51 (5) (1997) 483–544. doi:https://doi.org/10.1016/S0301-0082(96)00060-3.
URL https://www.sciencedirect.com/science/article/pii/S0301008296000603

[68] J. Wiener, N. Ehbauer, H. Mallot, Planning paths to multiple targets: memory involvement and planning heuristics in spatial problem solving, Psychological Research 73 (644). doi:https://doi.org/10.1007/s00426-008-0181-3.

[69] K. Friston, S. Samothrakis, R. Montague, Active inference and agency: optimal control without cost functions, Biol. Cybern. 106 (8) (2012) 523–541.

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

[71] S. J. Kiebel, J. Daunizeau, K. J. Friston, Perception and hierarchical dynamics, Front. Neuroinf. 3.

[72] L. Pio-Lopez, A. Nizard, K. Friston, G. Pezzulo, L. Pio-Lopez, A. Nizard, K. Friston, G. Pezzulo, Active Inference and Robot Control: A Case Study, Journal of the Royal Society, Interface 13 (122).

[73] A. Tschantz, M. Baltieri, Anil. K. Seth, C. L. Buckley, Scaling active inference, ArXiv e-printsarXiv:1911.10601.

[74] A. Tschantz, B. Millidge, A. K. Seth, C. L. Buckley, Reinforcement learning through active inference, in: The Bridging AI and Cognitive Science (BAICS) Workshop, ICLR, 2020.

[75] B. Millidge, Deep active inference as variational policy gradients, Journal of Mathematical Psychology 96 (2020) 102348.

[76] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics, Vol. 1, 1986, pp. 435–461.

[77] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, Intelligent robotics and autonomous agents, MIT Press, 2005.

[78] F. Yang, B. Jin, An overview of SLAM, in: Global Intelligence Industry Conference (GIIC 2018), Vol. 10835, 2018, pp. 214 – 219.

[79] J. Steckel, H. Peremans, Batslam: Simultaneous localization and mapping using biomimetic sonar, PLOS ONE 8 (1) (2013) 1–11.

[80] L. Silveira, F. Guth, P. Drews-Jr, P. Ballester, M. Machado, F. Codevilla, N. Duarte-Filho, S. Botelho, An open-source bio-inspired solution to underwater slam4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV 2015.

[81] R. Li, S. Wang, Z. Long, D. Gu, Undeepvo: Monocular visual odometry through unsupervised deep learning, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 7286–7291.

[82] I. Melekhov, J. Ylioinas, J. Kannala, E. Rahtu, Relative camera pose estimation using convolutional neural networks, in: J. Blanc-Talon, R. Penne, W. Philips, D. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems, 2017, pp. 675–687.

[83] C. Duan, S. Junginger, J. Huang, K. Jin, K. Thurow, Deep Learning for Visual SLAM in Transportation Robotics: A review, Transportation Safety and Environment 1 (3) (2020) 177–184.