Hierarchical Polyphonic Active Inference in a Gridworld

Alexander D. Shaw
Computational Psychiatry & Neuropharmacological Systems (CPNS) Lab
Department of Psychology, Faculty of Health & Life Sciences
University of Exeter, UK
https://cpnslab.com

1 Introduction

1.1 Overview

We implemented a hierarchical active inference agent in a partially observed gridworld containing goals, hazards, walls, and charging stations. The model extends a simpler polyphonic control scheme into a more explicitly active-inference-like architecture with the following features:

1. multiple "voices" or control units, each evaluating candidate policies according to a distinct set of preferences and costs;
2. a higher-level latent mode variable that encodes behavioural context, such as exploration, goal pursuit, recharge, threat avoidance, or verification;
3. partial and noisy observations of the environment, forcing the agent to maintain probabilistic beliefs over local hazard structure;
4. explicit short-horizon policy evaluation using an approximate expected free energy (EFE) decomposition into risk, ambiguity, and epistemic value;
5. subgoal selection conditioned on the inferred behavioural mode;
6. adaptive arbitration over voices, producing a mixed posterior over policies and hence over actions.

The key idea is that action does not emerge from a single utility signal; instead, different control units express partially competing imperatives, such as "move toward the goal", "avoid danger", "reduce uncertainty", or "preserve energy". These are integrated by a higher-level arbitration mechanism, producing behaviour that is context-sensitive, *non-monolithic*, and closer in spirit to active inference than a conventional reward-maximising controller.

1.2 Environment

The world is a discrete two-dimensional grid with dimensions $N_x \times N_y$. At any time t, the agent occupies a position

$$s_t^{\text{pos}} = (x_t, y_t),$$

with battery state

$$s_t^{\text{batt}} \in [0, 1].$$

The environment contains:

- impassable walls;
- one current goal location;
- a set of charging stations;
- a set of true hazards, hidden from the agent except through noisy local observations.

The agent can choose from five actions:

$$a_t \in \{\text{Up}, \text{Right}, \text{Down}, \text{Left}, \text{Stay}\}.$$

Movement is battery-gated; that is, even if the agent selects a movement action, execution can fail with probability increasing as battery falls. Thus, low energy affects not only preferences but also the state transition dynamics themselves.

1.3 Partial observation and hazard beliefs

The environment is only partially observed. At each time step, the agent observes a square neighbourhood of radius r_{obs} around its current position. For each observed cell i, it receives a noisy binary cue

$$z_i \in \{0, 1\},$$

where the likelihood depends on whether that cell truly contains a hazard:

$$p(z_i = 1 \mid h_i = 1) = p_{\text{detect}}, \qquad p(z_i = 1 \mid h_i = 0) = p_{\text{false}}.$$

Thus, if a hazard is truly present, the agent usually but not always receives a positive cue. If a hazard is absent, a false positive can still occur.

The agent maintains a belief map

$$B^{\text{haz}}(x, y) \in [0, 1],$$

representing the posterior probability that a given location is dangerous. Beliefs are updated approximately via a cellwise Bayesian rule. If the prior hazard belief at a cell is p and the observation is z, then the posterior is

$$p' = \begin{cases} \dfrac{p_{\text{detect}}\, p}{p_{\text{detect}}\, p + p_{\text{false}}(1 - p)}, & z = 1, \\ \dfrac{(1 - p_{\text{detect}})\, p}{(1 - p_{\text{detect}})\, p + (1 - p_{\text{false}})(1 - p)}, & z = 0. \end{cases}$$

After updating observed cells, beliefs are weakly regularised back toward a hazard prior:

$$B^{\text{haz}} \leftarrow \lambda B^{\text{haz}} + (1 - \lambda) p_0,$$

with λ close to 1. This prevents unstable saturation everywhere while still allowing strongly learned threat locations to remain highly aversive.

1.4 Reset memory and learned danger

A distinctive feature of the model is the use of a short-term reset memory

$$B^{\text{reset}}(x, y),$$

which records locations associated with recent catastrophic outcomes. If the agent steps onto a true hazard, it is reset to the start location and receives a penalty. At that point:

1. the hazard belief at the exact hit cell is forced close to 1;
2. neighbouring cells also receive elevated hazard beliefs;
3. the reset memory map is boosted locally, creating a temporary *no-go* region;
4. the current behavioural mode is forced to update on the next time step.

This mechanism is important because purely probabilistic hazard beliefs may still be too soft. Reset memory adds a second, more aversive signal: not only is a cell believed to be dangerous, but it is associated with a recent realised failure.

1.5 High-level behavioural mode inference

Above the voice layer sits a high-level latent mode variable

$$m_t \in \{\text{Explore}, \text{PursueGoal}, \text{Recharge}, \text{AvoidThreat}, \text{Verify}\}.$$

Rather than fixing the agent's behavioural style, the model infers a posterior over modes

$$q(m_t),$$

based on the current context. This context includes:

- local threat level;
- global uncertainty in the hazard map;
- battery urgency;
- distance to the goal;
- local revisitation and oscillation, used as a proxy for being stuck;
- local uncertainty near the agent, relevant for verification.

For each mode j, a heuristic logit z_j is computed, for example:

$$z_{\text{Explore}} = \alpha_1 \,\text{uncGlobal} + \alpha_2 \,\text{stuck} + \alpha_3 \, d_{\text{goal}},$$

$$z_{\text{PursueGoal}} = \beta_1 \, d_{\text{goal}} + \beta_2(1 - \text{threat}) + \beta_3(1 - \text{battNeed}),$$

$$z_{\text{Recharge}} = \gamma_1 \,\text{battNeed} + \gamma_2(1 - \text{threat}),$$

and similarly for AvoidThreat and Verify.

These are converted to a posterior using a regular softmax:

$$q(m_t = j) = \frac{\exp(z_j)}{\sum_k \exp(z_k)}.$$

1.6 Subgoal selection

The model distinguishes between a *final goal* and the *current subgoal*; the final goal is the blue target in the world and yields reward upon being reached. The subgoal is an internal waypoint chosen according to the current behavioural mode.

Formally, the subgoal is

$$g_t^{\text{sub}} \in \mathcal{S},$$

where $\mathcal{S}$ is the set of reachable grid cells.

The mode determines how the subgoal is chosen:

- Pursue Goal: the subgoal is the true goal.
- Recharge: the subgoal is a waypoint along the shortest path to the nearest charger.
- Avoid Threat: the subgoal is a locally safe refuge.
- Verify: the subgoal is a high-uncertainty location.
- Explore: the subgoal is a frontier-like location chosen to balance uncertainty, distance, threat clearance, reset memory, and revisitation.

To reduce erratic behaviour, the subgoal is not changed every time beliefs change slightly. Instead, subgoal persistence is introduced: the current subgoal is retained for a minimum duration unless there is a strong reason to refresh it, such as mode change, reset, completion, or extreme stuckness. This is important because without persistence, the subgoal behaves like a noisy instantaneous *argmax*, producing a jumping cyan marker and unstable plans. With persistence, it becomes more like an internal intention.

1.7 Distance maps and path-aware planning

A key step toward more realistic behaviour is the replacement of Euclidean distance with path-aware grid distance. Given a target location g, we compute a shortest-path distance map

$$D_g(x, y),$$

using breadth-first search over the free grid cells. This means that the effective distance to a goal or subgoal respects walls and maze structure.

Similarly, a multi-source distance map is computed for chargers:

$$D_{\text{charger}}(x, y),$$

and (optionally) for threats:

$$D_{\text{threat}}(x, y).$$

These maps are then used inside planning; for example, progress toward a subgoal is defined by

$$\Delta_{\text{prog}} = D_{g^{\text{sub}}}(s_t) - D_{g^{\text{sub}}}(s_{t+1}),$$

rather than by raw Euclidean closeness. This is critical because without it, the agent prefers cells that are close in straight-line distance even if they are separated by walls or dangerous bottlenecks.

1.8 Polyphonic voice architecture

The model contains $K = 5$ voices:

$$k \in \{\text{Safety}, \text{Goal}, \text{Epistemic}, \text{Energy}, \text{Habit}\}.$$

Each voice evaluates candidate policies according to a distinct weighting over components of approximate expected free energy. Thus each voice corresponds not to a different action system, but to a different *interpretation* of the same candidate futures.

The high-level mode posterior is converted into a prior mixture over voices using a template matrix

$$T \in \mathbb{R}^{K \times M},$$

where M is the number of modes. If $q(m_t)$ is the current mode posterior, then the unnormalised voice mixture is

$$\tilde{\pi}_t = T\, q(m_t).$$

This is normalised and smoothed over time:

$$\pi_t = (1 - \alpha)\pi_{t-1} + \alpha\, \text{norm}(\tilde{\pi}_t),$$

with floor constraints to prevent any voice from disappearing entirely.

Thus, the voice weights

$$\pi_t(k)$$

represent the current influence of each control unit.

1.9 Policy space

At each time step, the agent evaluates a finite set of short-horizon policies:

$$\pi = (a_t, a_{t+1}, \ldots, a_{t+H-1}),$$

where H is the planning horizon. Since each action can take one of $A = 5$ values, the total number of candidate policies is

$$A^H.$$

For each voice k, the model evaluates all candidate policies and constructs a posterior

$$q_k(\pi).$$

1.10 Approximate expected free energy decomposition

For each voice and policy, the agent simulates future trajectories under approximate deterministic transition dynamics and accumulates three terms:

1. Risk: expected mismatch with preferences, including hazard exposure, distance from subgoal, low battery, charger viability failure, recent reset memory, and control penalties.

2. Ambiguity: uncertainty in outcomes, approximated here by the entropy of the hazard belief map at predicted locations.

3. Epistemic value: expected information gain proxy, approximated by preferring locations that are both uncertain and relatively unvisited.

For a given voice k and policy π, the approximate expected free energy is

$$G_k(\pi) = w_{\text{risk}} R_k(\pi) + w_{\text{amb}} A_k(\pi) - w_{\text{epi}} E_k(\pi).$$

The sign convention follows the usual active inference intuition: ambiguity and risk are undesirable, while epistemic value is beneficial, hence its subtraction.

Although this is not a full exact discrete-state active inference derivation, it captures the central structure:

$$G \approx \text{risk} + \text{ambiguity} - \text{epistemic value}.$$

1.11 Detailed policy rollout terms

Let s_h denote the predicted state at rollout step h under policy π. Then the planner computes terms such as:

Hazard risk

$$r_{\text{haz}}(s_h) = B^{\text{haz}}(s_h).$$

Subgoal distance cost

$$r_{\text{goal}}(s_h) = \frac{D_{g^{\text{sub}}}(s_h)}{D_{\max}},$$

where $D_{\max}$ is a normalising constant based on grid size.

Battery risk

$$r_{\text{batt}}(s_h) = \max(0, b_{\text{set}} - b_h).$$

Viability risk If the shortest path distance to a charger is $D_{\text{charger}}(s_h)$ and the remaining number of executable steps is estimated as

$$n_h = \left\lfloor \frac{b_h}{\delta_b} \right\rfloor,$$

then

$$r_{\text{viab}}(s_h) = \max(0, D_{\text{charger}}(s_h) - n_h).$$

Reset memory cost

$$r_{\text{reset}}(s_h) = B^{\text{reset}}(s_h).$$

Ambiguity If $p_h = B^{\text{haz}}(s_h)$, then ambiguity is measured by Bernoulli entropy:

$$H(p_h) = -p_h \log p_h - (1 - p_h) \log(1 - p_h).$$

Epistemic value A simple proxy is

$$e_h = H(p_h)\,(0.8 + 0.6\,\nu_h)\,,$$

where ν_h is a novelty term inversely related to visit count.

These stepwise components are weighted differently for each voice. For example, the safety voice gives high weight to hazard and reset-related terms, while the energy voice emphasises battery and charger viability.

1.12 Known-threat exclusion

In later versions, we found that treating known lethal cells as merely high-cost states was sometimes insufficient, especially when goal and energy pressures were strong. As a result, a practical refinement is to treat strongly learned threat cells as prohibited states. If a predicted position satisfies

$$B^{\text{haz}}(s_h) > \theta_{\text{haz}} \quad \text{or} \quad B^{\text{reset}}(s_h) > \theta_{\text{reset}},$$

then that branch of the policy rollout receives a very large penalty or is terminated early. This is not a pure active inference derivation, but it is a useful stabilising approximation in discrete environments with hard catastrophic states.

1.13 Voice-wise policy posteriors

Once $G_k(\pi)$ has been computed for every candidate policy under voice k, a posterior is formed by softmax:

$$q_k(\pi) = \frac{\exp\left(-\beta_\pi [G_k(\pi) - \min_{\pi'} G_k(\pi')]\right)}{\sum_{\pi'} \exp\left(-\beta_\pi [G_k(\pi') - \min_{\pi''} G_k(\pi'')]\right)}.$$

Subtracting the minimum improves numerical stability without changing relative probabilities.

1.14 Polyphonic integration

The voice-specific policy posteriors are then combined according to the current voice weights:

$$q(\pi) = \sum_{k=1}^{K} \pi_t(k)\, q_k(\pi).$$

This is the core "polyphonic" step; each voice contributes a different posterior over policies; arbitration forms a mixture over these voices; the result is a context-sensitive overall posterior over policies.

Finally, the policy posterior is marginalised to obtain an action posterior over the first action:

$$q(a_t = a) = \sum_{\pi : \pi_1 = a} q(\pi).$$

A final softmax with action precision β_a may be applied:

$$q(a_t) \propto \exp\big(\beta_a \log q(a_t)\big).$$

The executed action is then taken as

$$a_t^* = \arg\max_a q(a_t = a).$$

1.15 Battery-gated transition dynamics

The chosen action is not always executed perfectly. Instead, battery affects the probability of successful motion:

$$p_{\text{move}} = \sigma(\kappa(b_t - b_{\text{mid}}))\,,$$

where σ is the logistic sigmoid. At very low battery, movement may fail or the agent may be forced to stay.

This creates an important distinction between:

- the *intended* transition under policy rollout;
- the *realised* transition in the environment.

Thus the agent may plan well but still suffer occasional failures late in the run, especially under low battery.

1.16 Algorithm

A single time step can be summarised as follows.

1. Decay the reset-memory map.
2. Observe a local patch around the current position.
3. Update hazard beliefs using noisy observations.
4. Compute battery viability and contextual features.
5. Infer the high-level mode posterior $q(m_t)$.
6. If appropriate, update or retain the current subgoal.
7. Convert mode posterior into smoothed voice weights π_t.
8. For each voice k:
   (a) enumerate candidate policies;
   (b) roll each policy forward under approximate dynamics;
   (c) accumulate risk, ambiguity, and epistemic value;
   (d) compute $G_k(\pi)$;
   (e) form $q_k(\pi)$.
9. Combine voice-specific posteriors into a global policy posterior.
10. Marginalise to a posterior over actions.
11. Execute the selected action under battery-gated dynamics.
12. If a hazard is hit, apply reset penalty and update threat memory.
13. If the goal is reached, award reward and respawn the goal.
14. Log state, beliefs, mode, voice weights, and diagnostics.

1.17 Interpretation

The model is useful because it sits between a simple reinforcement-learning controller and a full exact active inference implementation. It is more structured than a monolithic reward agent because:

- it explicitly separates multiple control imperatives into voices;
- it includes hierarchical latent behavioural context;

- it acts under partial observability using probabilistic beliefs;
- it incorporates epistemic terms, not just exploitative reward;
- it supports internal subgoals and route restructuring under conflict.

At the same time, it remains computationally tractable and visually interpretable. In particular, failures such as oscillations between goal pursuit and threat avoidance, local deadlock, subgoal instability, and catastrophic revisits to known hazards are informative rather than merely undesirable, because they expose specific missing ingredients in the model / agent architecture.

1.18 Limitations and future directions

Several simplifications remain.

1. The expected free energy is approximate rather than exact.
2. Hazard beliefs are represented by a single map rather than a full posterior with separate mean and uncertainty parameters.
3. Policy evaluation uses finite-horizon brute force over a discrete policy set.
4. Subgoals are represented as single points rather than soft attractor fields or path distributions.
5. Known threats are partly handled by engineered prohibitive penalties rather than fully emergent inference alone.

Future improvements include:

- a smooth probabilistic threat field over the grid;
- path-level rather than point-level subgoal inference;
- explicit belief updates over route structure;
- policy pruning or tree search to extend the planning horizon;
- a fuller discrete-state active inference formalism with explicit likelihood and transition matrices.

1.19 Summary

In summary, the model implements a hierarchical, polyphonic active inference agent in which behaviour emerges from the interaction of multiple policy-evaluating voices, a higher-level inferred behavioural mode, probabilistic hazard beliefs, and subgoal-based route planning. This produces flexible but interpretable behaviour in a partially observed environment, while also exposing concrete failure modes that motivate further theoretical and algorithmic refinement.