# Recurrent networks, hidden states and beliefs in partially observable environments

Gaspard Lambrechts, Adrien Bolland and Damien Ernst

# Background

## Partially observable Markov decision process

$P = (\mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, O, p_0, \gamma)$

- State $\mathbf{s}_t \in \mathcal{S}$,
- Action $\mathbf{a_t} \in \mathcal{A}$,
- Observation $\mathbf{o}_t \in \mathcal{O}$,
- Transition distribution $T(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$,
- Reward function $r_t = R(\mathbf{s}_t, \mathbf{a}_t)$,
- Observation distribution $O(\mathbf{o}_t \mid \mathbf{s}_t)$,
- Initialisation distribution $p_0(\mathbf{s}_0)$,
- Discount factor $\gamma \in [0, 1[$.

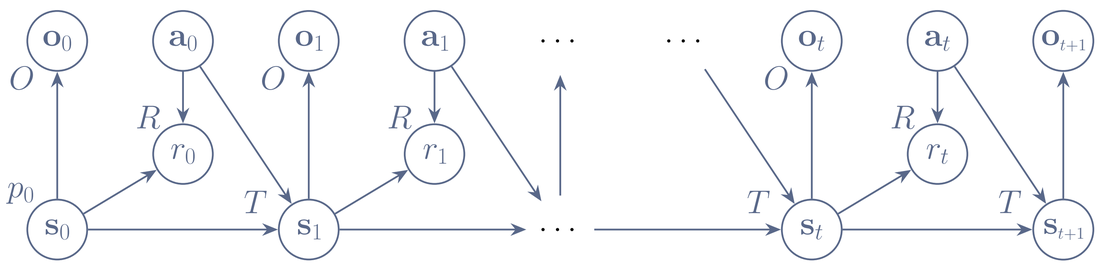


**Fig 1.** Partially observable Markov decision process.

The **history** $\eta_{0:t}$ until time step $t$ is defined as

$$\eta_{0:t} = (\mathbf{o}_0, \mathbf{a}_0, \dots, \mathbf{a}_{t-1}, \mathbf{o}_t) \in \mathcal{H}_{0:t}.$$

A history of arbitrary length is denoted by $\eta \in \mathcal{H} = \bigcup_{t=0}^{\infty} \mathcal{H}_{0:t}$.

## Belief sufficiency

The **belief** $b = f^*(\eta)$ of a history $\eta$ is defined as

$$b(\mathbf{s}) = p(\mathbf{s} \mid \eta), \ \forall \mathbf{s} \in \mathcal{S}.$$

**The belief is a sufficient statistic from the history for the $\mathcal{Q}$-function**

There exists a function $Q$ such that, for all $\eta \in \mathcal{H}$ and $\mathbf{a} \in \mathcal{A}$, with $b = f^*(\eta)$ the belief resulting from history $\eta$,

$$\mathcal{Q}(\eta, \mathbf{a}) = Q(b, \mathbf{a}). \tag{1}$$

# Motivation

## Belief filter

The belief filter $f^*$ is defined as the function that maps a history to its corresponding belief.

**Recurrent belief update**

For a history $\eta' = \eta \cup (\mathbf{a}, \mathbf{o}')$, the belief $b = f^*(\eta)$ can be updated to $b' = f^*(\eta')$ based on $(\mathbf{a}, \mathbf{o}')$ only

$$b' = f(b; \mathbf{a}, \mathbf{o}'). \tag{2}$$

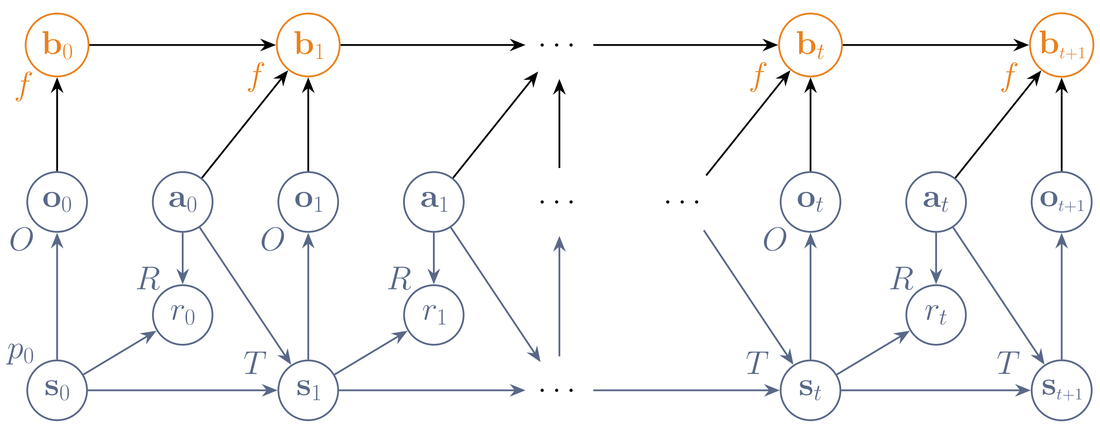


**Fig 2.** Belief filter.

$\Rightarrow$ The belief is a **sufficient statistic** from the history for the $\mathcal{Q}$-function **at all later time steps**.

## Recurrent Q-learning

Recurrent Q-learning aims at learning an approximation $\mathcal{Q}_\theta$ of the $\mathcal{Q}$-function with an RNN. The latter produces a **hidden state** $\mathbf{h} = u_\theta^*(\eta)$ from the history $\eta$ and outputs $\mathcal{Q}_\theta(\eta, \mathbf{a}) = o_\theta(\mathbf{h}, a)$.

**Recurrent hidden state update**

For a history $\eta' = \eta \cup (\mathbf{a}, \mathbf{o}')$, the hidden state $\mathbf{h} = u_\theta^*(\eta)$ can be updated to $\mathbf{h}' = u_\theta^*(\eta')$ based on $(\mathbf{a}, \mathbf{o}')$ only

$$\mathbf{h}' = u_\theta(\mathbf{h}; \mathbf{a}, \mathbf{o}'). \tag{3}$$

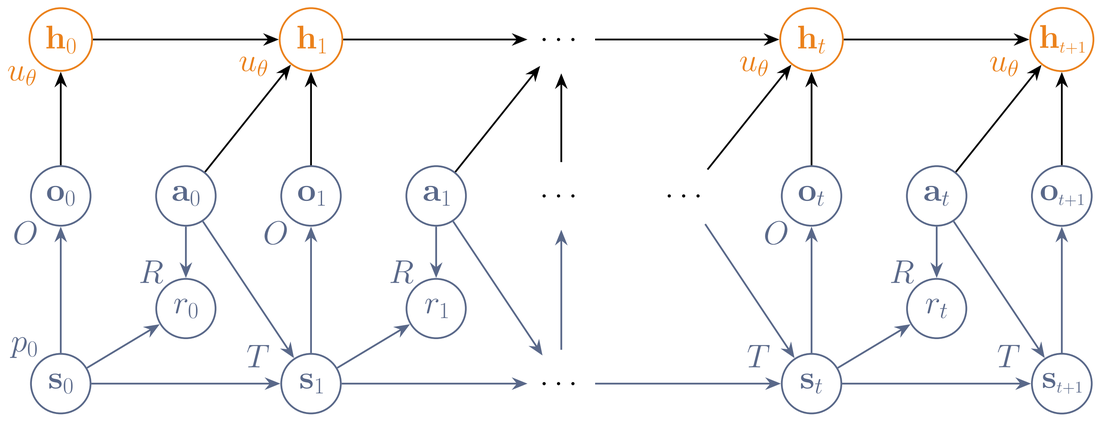


**Fig 3.** Recurrent neural network.

**Research question**

Given the **belief sufficiency** (1), **belief update** (2) and **hidden state update** (3), **should recurrent Q-learning encode information about the belief in the hidden state?**

# Experiments

During recurrent Q-learning, we study the evolution of:

- The **empirical return** $\hat{J}(\theta_e) \approx \mathop{\mathbb{E}}_{P,\pi_{\theta_e}}[\sum_{t=0}^{\infty} \gamma^t r_t]$,
- The **estimated mutual information** $\hat{I}(\theta_e) \approx I(\mathbf{h}, b)$ under distribution $p_{\pi_{\theta_e}}(\mathbf{h}, b)$.

Recurrent architectures: **LSTM**, **GRU**, **BRC**, **nBRC**, **MGU**.

## T-Maze

The position of the treasure is observed in $(0, 0)$ and the position of the agent is never observed in the corridor. The agent has to reach the treasure by taking the correct direction at the crossroads.

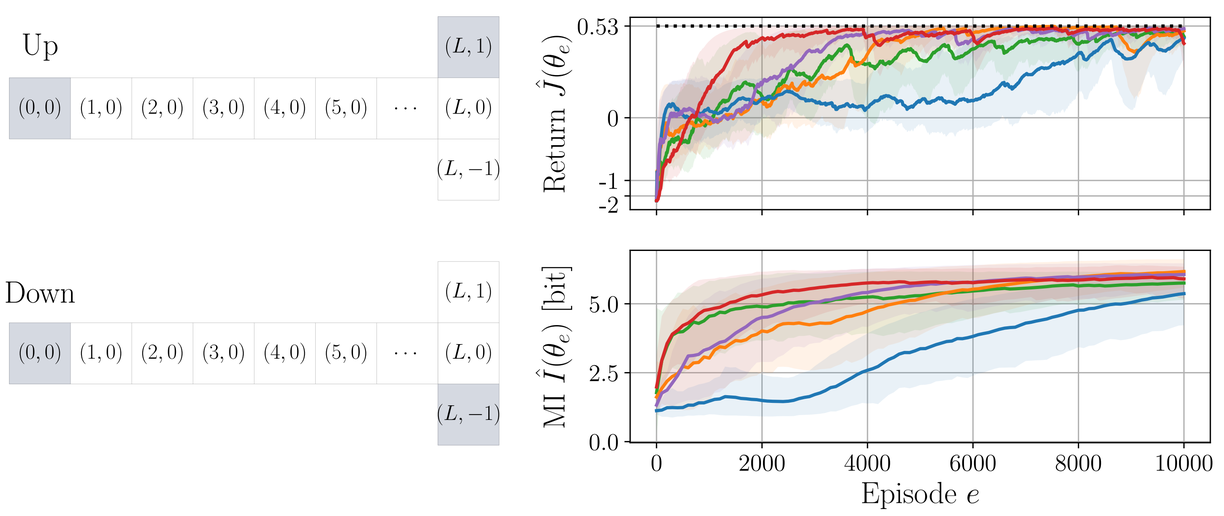


**Fig 4.** T-Maze state space (left). Empirical return $\hat{J}$ and estimated MI $\hat{I}$ (right).

## Varying Mountain Hike

The agent only gets a noisy measure of its altitude and does not know its initial orientation. The agent has to maximise its altitude.

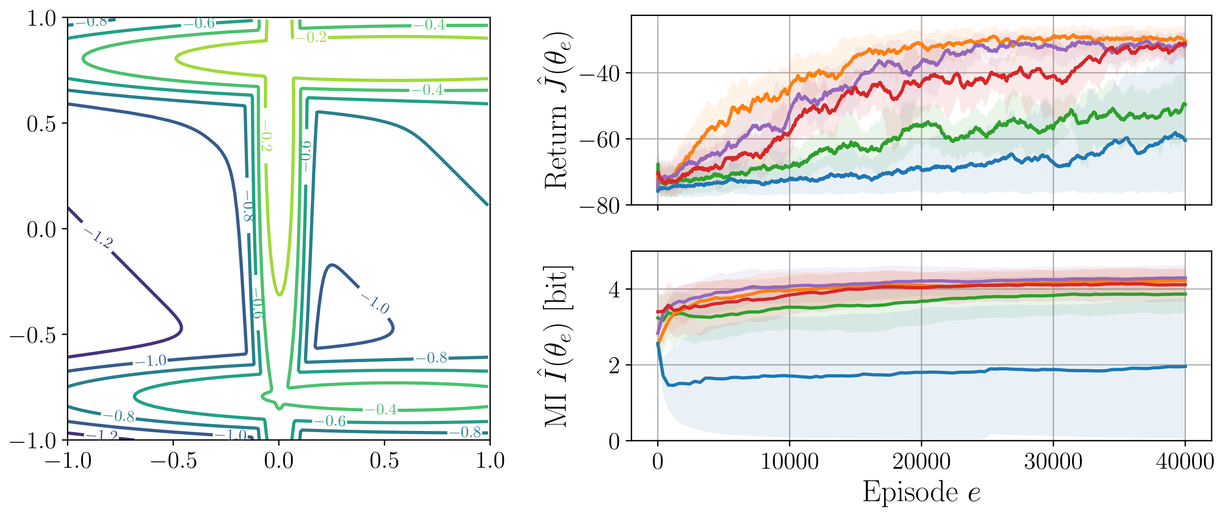


**Fig 5.** Mountain Hike state space (left). Empirical return $\hat{J}$ and estimated MI $\hat{I}$ (right).

## Irrelevant Variables

Noisy observations $\mathbf{o}_t^I$ of a random Gaussian walk $\mathbf{s}_t^I$ are added. We separately report the MI for the belief of the original state variables $\mathbf{s}$ and for the belief of the irrelevant state variables $\mathbf{s}^I$.

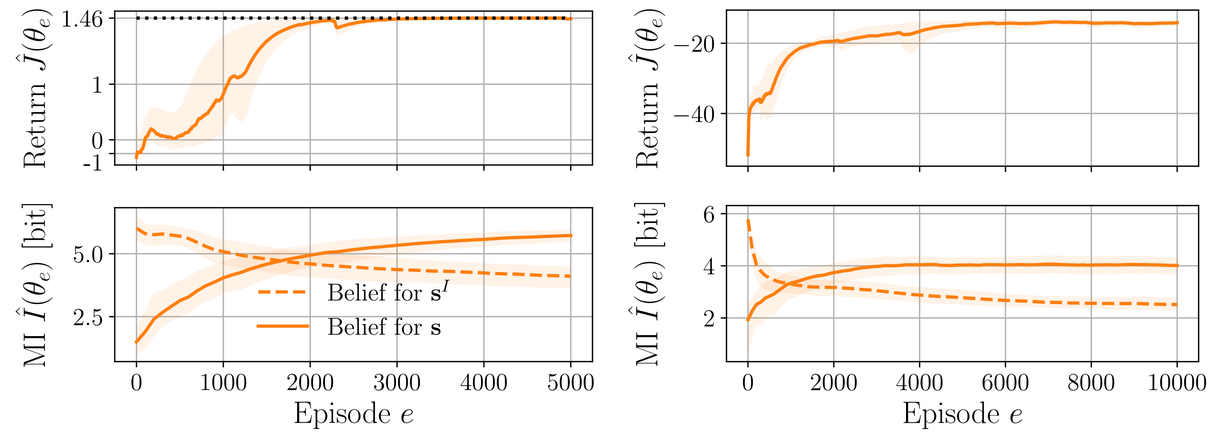


**Fig 6.** T-Maze (left) and Mountain Hike (right) with irrelevant state variables and observations.

# Conclusion

**Contribution**

High return policies obtained by recurrent Q-learning produce **hidden states that encode a high amount of information about beliefs** of relevant state variables.

**Future works**

- How do these observations **generalise to other recurrent RL algorithms**?
- Can we speed up learning by **biasing the hidden state to represent the belief**?
- Can we bias the hidden state to **implicitly represent the belief** by maximising MI?