

Chapter 1

Active Inference(AIF)

1.1 Active Inference

Recall the definition of POMDP, we will define a 7-tuple for the whole decision making process: $\{\mathbb{S}, \mathbb{A}, \mathbb{O}, \mathbf{T}, \mathbf{R}, \mathbf{O}, \gamma\}$, where at time step t , the state $s_t \in \mathbb{S}$ are hidden in the real world, its transition mechanism is illustrated by the transition model $T(s_{t+1}|s_t, a_t)$, action will be taken according to a policy $\pi(a_t|s_t)$, the observation comes from the observation model $o_t = O(o_t|s_t)$. To optimize the policy, we define a reward model $r_t = R(s_t, a_t)$ and the weighted accumulated total return $\mathbb{E}[U_t^\pi] = \mathbb{E}[\sum_{i=t}^{\infty} \gamma^{i-t} r_i]$.

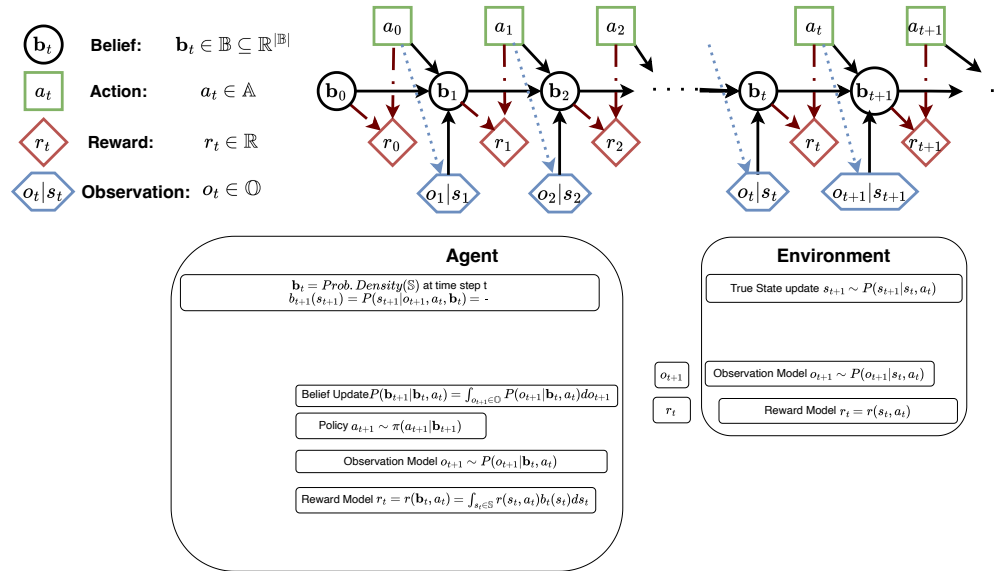


Figure 1.1: Original POMDP Model

There are only one model for the real world

After understanding the basics of Partially Observable Markov Decision Process (POMDP), we can To have an basic understanding of Active Inference shown in the Figure 1.2, we should clarify that there are two models defined, i.e. the real world model described by the generative process $P^*(s^*, o)$ and the generative model $P(s, o) = P(s|o)P(o) = P(o|s)P(s)$.

The true process is always unknown, the agent has a generative model about how the outside world works. The only true connection between the interior and external world is the outcome (or observation).

The internal generative model would be updated every time observe some outcome. The prior belief is defined as the probability density of the state $P(s)$, the posterior is defined as

$$P(s|o) = \frac{P(s, o)}{P(o)} = \frac{P(o|s)P(s)}{P(o)} \quad (1.1)$$

Due to the difficulty of calculating the evidence $P(o) = \int_{s \in \mathbb{S}} P(o|s)P(s)$, we may use $Q(s)$ to approximate the posterior distribution $P(s|o)$. The information loss caused when using $Q(s)$ to approximate $P(s|o)$ is given by the relative Kullback-Divergence:

$$D_{KL}[Q(s)||P(s|o)] = \mathbb{E}_{Q(s)} [\ln(Q(s)) - \ln(P(s|o))] \quad (1.2)$$

When substitute the Eq. 1.1 to the Eq. 1.2 we could get:

$$D_{KL}[Q(s)||P(s|o)] = \mathbb{E}_{Q(s)} [\ln(Q(s)) - \ln(P(s|o))] \quad (1.3a)$$

$$= \mathbb{E}_{Q(s)} \left[\ln(Q(s)) - \ln \left(\frac{P(s, o)}{P(o)} \right) \right] \quad (1.3b)$$

$$= \mathbb{E}_{Q(s)} [\ln(Q(s)) - \ln(P(s, o))] + \mathbb{E}_{Q(s)} [\ln(P(o))] \quad (1.3c)$$

The first term is defined as the variational free energy $\mathcal{F} \stackrel{\text{def}}{=} \mathbb{E}_{Q(s)} [\ln(Q(s)) - \ln(P(s, o))] = D_{KL}[Q(s)||P(s, o)]$ Then we can write that

$$\mathcal{F} = D_{KL}[Q(s)||P(s, o)] \quad (1.4a)$$

$$= D_{KL}[Q(s)||P(s|o)] - \mathbb{E}_{Q(s)} [\ln(P(o))] \quad (1.4b)$$

$$= D_{KL}[Q(s)||P(s)] - \mathbb{E}_{Q(s)} [\ln(P(o|s))] \quad (1.4c)$$

The physical meaning of Eq. 1.4b

1.1.1 Data Collection

The research employs multiple datasets to ensure comprehensive evaluation:

Table 1.1: The physical meaning of variational Free Energy \mathcal{F}

Variational	=	$D_{\text{KL}} [Q(s) P(s o)] - \mathbb{E}_{Q(s)}[\ln(P(o))]$	+	$-\mathbb{E}_{Q(s)}[\ln(P(o))]$
Free	=	Divergence	+	Surprisal
Energy	=	$D_{\text{KL}} [Q(s) P(s)]$	-	$\mathbb{E}_{Q(s)}[\ln(P(o s))]$
\mathcal{F}	=	Complexity	-	Accuracy (Goodness of fit)

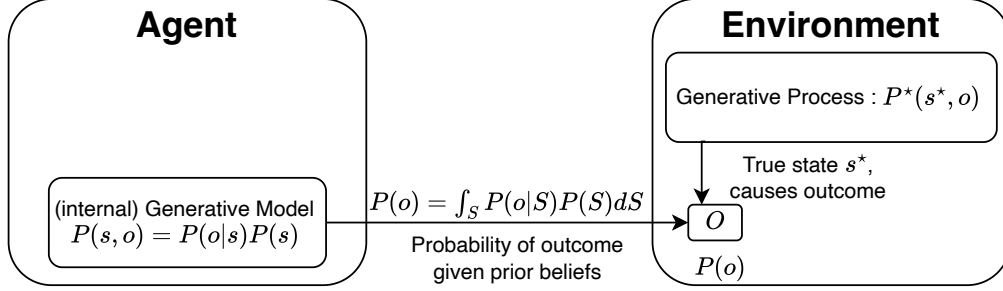


Figure 1.2: Graphical Representation of the Generative Process(based on the true states s^*) in the world and the corresponding (internal) generative model (based on probabilistic beliefs random variables s) that best explain the outcomes o . The outcomes are shared between the generative process and model. [sajid2021active]

- **Dataset A:** Contains 10,000 samples with 50 features...
- **Dataset B:** Comprises time-series data spanning 5 years...
- **Dataset C:** Includes multimodal data from various sources...

1.2 Algorithm Design

The proposed algorithm builds upon the foundation established by [lee2019optimization]. The pseudocode is presented in Algorithm 1.

1.3 Experimental Setup

All experiments were conducted on a computing cluster with the following specifications:

- CPU: Intel Xeon Gold 6248R (3.0 GHz)
- GPU: NVIDIA A100 (40GB)

Input: X, θ
Output: Best solution in P

```

1 Initialize population  $P \leftarrow \emptyset$ ;
2 for  $i \leftarrow 1$  to  $N$  do
3    $x_i \leftarrow \text{InitializeSolution}()$ ;
4    $P \leftarrow P \cup \{x_i\}$ ;
5 end
6 while not converged do
7   Evaluate fitness for all  $x \in P$ ;
8   Select parents for reproduction;
9   Apply crossover and mutation;
10  Update population  $P$ ;
11 end
12 return best solution in  $P$ ;

```

Algorithm 1: Proposed Optimization Algorithm

- Memory: 256 GB DDR4
- Storage: 2 TB NVMe SSD