# What Can Learned Intrinsic Reward Capture?

**Gao Chenxiao**

LAMDA, Nanjing University

November 17, 2020

## Table of Contents

## Table of Contents

## Contributions

- Use intrinsic reward to capture long-term knowledge about **both exploration and expoitation**
- Distinguish the roles of policy and reward function in RL problems:
  - Policy decribes "How should the agent behave", while the reward function decribes "What the agent should strive to do"
  - Knowledge about "what" is indirect and slower to take effect on agent's behavior (through palnning and learning), but it can generalize to difference algorithms better.

## Related Work

- Reward Shaping: aims to handcraft reward towards known optimal rewards
  - Two main mathods: task dependent / task independent
- Reward learned from experience
  - Optimal Reward Framework: introduced by Singh et al.,2009
  - Compared to LIRPG and AGILE, this agent is able to generalize to new agent-environment interfaces and algorithms.
- Cognitive study
  - Humans use both a random exploration strategy and an information seeking strategy when facing uncertainty, however the latter one hasn't been fully discussed in prior articles and essays.

## Table of Contents

## Terminology: MDP Part

- MDP $=$ System Dynamic $+$ Extrinsic Reward
  - Agent: A learning system interacting with an environment. Each time step the agent selects an action $a_t$, receives an extrinsic reward $r_t$ defined by a task $\mathcal{T}$, and transits from state $s_t$ to $s_{t+1}$.
  - Policy: A mapping $(\pi_\theta(a|s))$ from the environment state to the agent's behavior, which is parameterized by $\theta$.
- Episode: A finite sequence of agent-environment interactions until the end.
  - **episodic return:** $G^{\text{ep}} = \sum_{t=0}^{T_{\text{ep}}-1} \gamma^t r_{t+1}$

## Terminology: Intrinsic Part

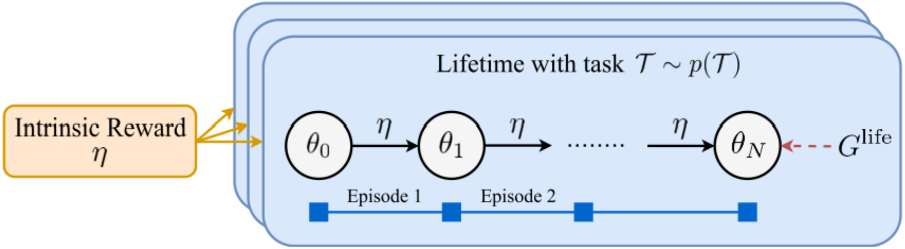
- **Lifetime:** A finite sequence of agent-environment interactions until the end of the training defined by an agent designer. In this paper, *lifetime* consists of a fixed number of episodes.
    - **Lifetime return:** $G^{\text{life}} = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$
- **Intrinsic Reward:** A reward function $r_\eta(\tau_t)$ parameterised by $\eta$, where $\tau_t = (s_0, a_0, r_1, d_1, s_1 \ldots, r_t, d_t, s_t)$ is a **lifetime** history experienced by the agent.
- **Lifetime Value Function:** A value function $V_\phi(s)$ which is used to approximate the accumulate lifetime return of the states.

## Terminology: The Optimal Reward Problem[1]

- Objective

$$\textbf{maximize}\;\; J(\eta) = \mathbb{E}_{\theta_0 \sim \Theta, \mathcal{T} \sim p(\mathcal{T})}\left[\mathbb{E}_{\tau \sim p_\eta(\tau|\theta_0)}[G^{\text{life}}]\right]$$

  - $\Theta$ and $p(\mathcal{T})$ are an initial policy distribution and a distribution over tasks, $\tau$ is agent's history and $p_\eta(\tau|\theta_0) = p(s_0)\prod_{t=0}^{T-1}\pi_{\theta_t}(a_t|s_t)p(d_{t+1}, r_{t+1}, s_{t+1}|s_t, a_t)$.



[1] Intrinsically motivated reinforcement learning: An evolutionary perspective

## Algorithm: Overview

Parameters:

- $\theta \to \pi_\theta$
- $\eta \to r_\eta$
- $\phi \to V_\phi$

**Algorithm 1** Learning intrinsic rewards

```
Input: $p(\mathcal{T})$: Task distribution
Input: $\Theta$: Randomly-initialised policy distribution
Initialise intrinsic reward $\eta$ and lifetime value $\phi$
repeat
    Initialise task $\mathcal{T} \sim p(\mathcal{T})$ and policy $\theta \sim \Theta$
    while lifetime not ended do
        $\theta_0 \leftarrow \theta$
        for $k = 1, 2, \dots, N$ do
            Generate a trajectory using $\pi_{\theta_{k-1}}$
            Update policy $\theta_k \leftarrow \theta_{k-1} + \alpha \nabla_{\theta_{k-1}} J_\eta(\theta_{k-1})$
            using intrinsic rewards $r_\eta$ (Eq. 3)
        end for
        Update intrinsic reward function $\eta$ using Eq. 4
        Update lifetime value function $\phi$ using Eq. 6
        $\theta \leftarrow \theta_N$
    end while
until $\eta$ converges
```

## Algorithm: Policy Update

- maximizing the **episodic** cumulated **intrinsic** reward

$$J_\eta(\theta) = \mathbb{E}_\theta\left[\sum_{t=0}^{T_{\text{ep}}-1} \bar{\gamma}^t r_\eta(\tau_{t+1})\right]$$

$$\nabla_\theta J_\eta(\theta) = \mathbb{E}_\theta\left[G^{\text{ep}}_{\eta,t} \nabla_\theta \log \pi_\theta(a|s)\right] \quad (\text{for each } t)$$

  where $G^{\text{ep}}_{\eta,t} = \sum_{k=t}^{T_{\text{ep}}-1} \bar{\gamma}^{k-t} r_\eta(\tau_{k+1})$.
- similar to REINFORCE

## Algorithm: Intrinsic Reward Update

- maximizing the lifetime reward

$$J(\eta) = \mathbb{E}_{\theta_0 \sim \Theta, \mathcal{T} \sim p(\mathcal{T})} \left[ \mathbb{E}_{\tau \sim p_\eta(\tau|\theta_0)} [G^{\text{life}}] \right]$$

$$\nabla_\eta J(\eta) = \mathbb{E}_{\theta_t, \mathcal{T}} \left[ G_t^{\text{life}} \nabla_{\theta_t} \log \pi_{\theta_t}(a_t|s_t) \nabla_\eta \theta_t \right]$$

Computing the meta-gradient requires backpropagation through the entire lifetime. In practice we truncate the meta-gradient after $N$ steps and use $G_t^{\text{life},\phi}$ to approximate $G_t^{\text{life}}$.

$$G_t^{\text{life},\phi} = \sum_{k=0}^{N-1} \gamma^k r_{t+k+1} + \gamma^n V_\phi(\tau_{t+n})$$

## Algorithm: Intrinsic Reward Update (cont'd)

similar to the drivation of policy gradient theorem.

$$\begin{aligned}
&\nabla_\eta J(\eta)\\
&= \nabla_\eta v(\tau_0|\eta)\\
&= \nabla_\eta \left[\sum_{a_0} \pi_{\theta_0}(a_0|\tau_0) q(\tau_0, a_0|\eta)\right]\\
&= \sum_{a_0} \left[\nabla_\eta \pi_{\theta_0}(a_0|\tau_0) q(\tau_0, a_0|\eta) + \pi_{\theta_0}(a_0|\tau_0) \nabla_\eta q(\tau_0, a_0|\eta)\right]\\
&= \sum_{a_0} \left[\nabla_\eta \pi_{\theta_0}(a_0|\tau_0) q(\tau_0, a_0|\eta) + \pi_{\theta_0}(a_0|\tau_0) \nabla_\eta \sum_{\tau_1, r_0} p(\tau_1, r_0|\tau_0, a_0)\Big(r_0 + v(\tau_1|\eta)\Big)\right]\\
&= \sum_{a_0} \left[\nabla_\eta \pi_{\theta_0}(a_0|\tau_0) q(\tau_0, a_0|\eta) + \pi_{\theta_0}(a_0|\tau_0) \sum_{\tau_1} p(\tau_1|\tau_0, a_0) \nabla_\eta v(\tau_1|\eta)\right]\\
&= \mathbb{E}_{\tau_t}\left[\sum_{a_t} \nabla_\eta \pi_\eta(a_t|\tau_t) q(\tau_t, a_t|\eta)\right]\\
&= \mathbb{E}_{\tau_t}\left[\nabla_\eta \log \pi_\eta(a_t|\tau_t) q(\tau_t, a_t|\eta)\right]\\
&= \mathbb{E}_{\tau_t}\left[G_t \nabla_\eta \log \pi_\eta(a_t|\tau_t)\right]\\
&= \mathbb{E}_{\tau_t}\left[G_t \nabla_{\theta_t} \log \pi_{\theta_t}(a_t|s_t) \nabla_\eta \theta_t\right],
\end{aligned}$$

where $G_t = \sum_{k=t}^{T-1} r_k$ is the lifetime return given the history $\tau_t$, and we assume the discount factor $\gamma = 1$ for brevity. Thus, the derivative of the overall objective is:

$$\nabla_\eta J(\eta) = \mathbb{E}_{\theta_0 \sim \Theta, \mathcal{T} \sim p(\mathcal{T})}\left[\mathbb{E}_{\tau_t \sim p(\tau_t|\eta, \theta_0)}\left[G_t \nabla_{\theta_t} \log \pi_{\theta_t}(a_t|s_t) \nabla_\eta \theta_t\right]\right]. \tag{9}$$

## Algorithm: Lifetime Value Update

- using a temporal difference from n-step trajectory

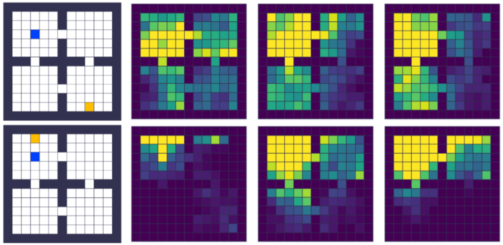
$$J(\phi) = \frac{1}{2}(G_t^{\text{life},\phi} - V_\phi(\tau_t))^2$$

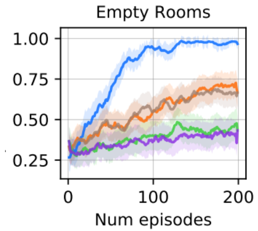
$$\nabla_\phi J(\phi) = (G_t^{\text{life},\phi} - V_\phi(\tau_t))\nabla_\phi V_\phi(\tau_t)$$

## Table of Contents

## Empty Room: Exploring Uncertainty

- blue squares: the hidden goal — yellow squares: the agent

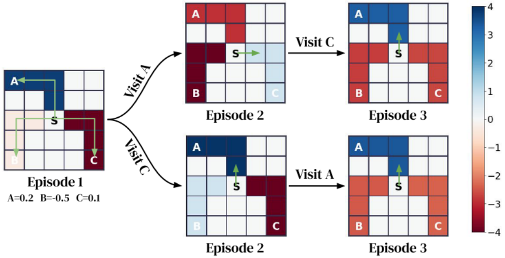

(a) Room (b) Intrinsic (c) Count (d) ICM

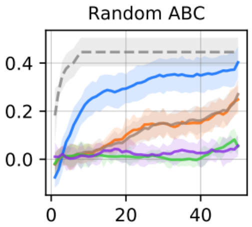


- When the goal is not found, the intrinsic reward encourages the agent to visit unknown locations; after the goal is found, it makes the agent to exploit the knowledge.

## ABC: Exploring Uncertain Objects

- $r(A) \sim U[-1, 1], r(B) \sim U[-0.5, 0], r(c) \sim U[0, 0.5]$

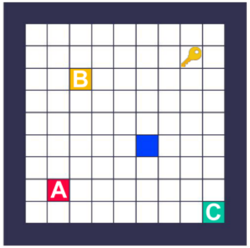




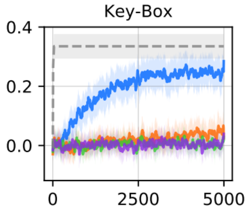
- These results show that avoidance and curiosity about uncertain objects can emerge in the intrinsic reward.

## Key-Box: Exploring and Exploiting Casual Relationship

- Based Random ABC, but the agent need first to collect the key.



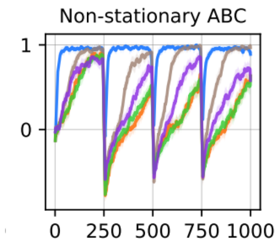
(c) Key-Box

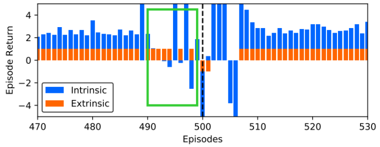


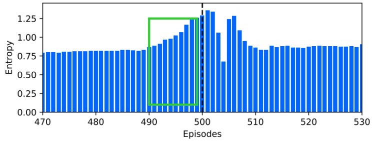
- Algorithms except Intrinsic Reward all failed to capture that the key is necessary to open any box,which demonstrates that intrinsic reward can learn the relationships between objects when the domain has this kind of invariant dynamics.

# Non-stationary ABC: Dealing with Non-stationary

- based on Random ABC, but the rewards of A and C are swapped every 250 episodes.





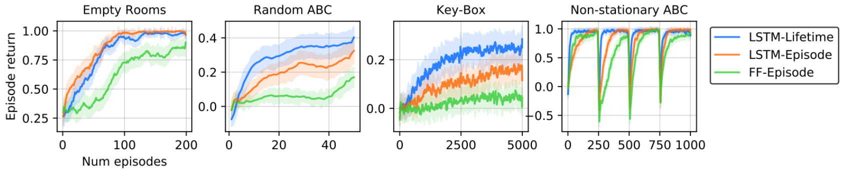


- This experiment shows that the intrinsic reward can capture the regularly repeated non-stationary pattern of the tasks.

## Ablation

Two ablation studies

- replace the lifetime return objective $G^{\text{life}}$ with episodic return $G^{\text{EP}}$
- restrict the input of the reward network to current state instead of the lifetime history



## Back to The Title

What can intrinsic reward capture?

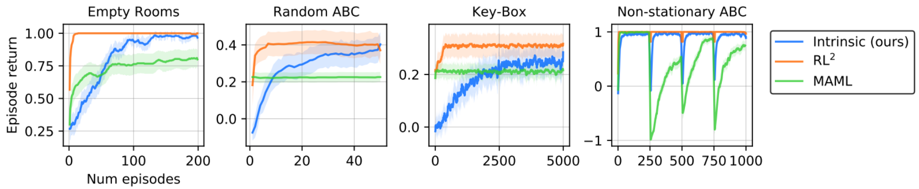
- useful exploring strategy
- characteristics of the task: stochastic reward, casual relationship and non-stationary patterns

and Why?

- it tells agent "what to do", rather than "do what"
- lifetime return utilizes cross-episode knowledge in a more explicit way, such as the comparison between the goal states(Random ABC)

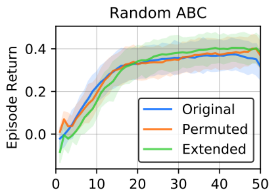
## Intrinsic Reward for Meta RL

- Knowledge captured by intrinsic is useful for training randomly-initialised policies, while other Meta-RL algorithms like MAML and RL2 are designed for fast adaptation to new tasks.

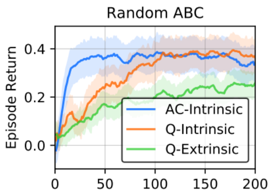


# Intrinsic Reward for Meta RL

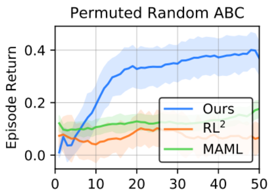
- ...but it can provide more robustness because it is model-agnostic and agent-agnostic.



(a) Action space



(b) Algorithm



(c) Comparison to baselines

Thanks for listening!