Learning to Integrate Exploration Strategies for Reinforcement Learning via an Option Framework

(LESSON)

ICML 2023

Woojun Kim*, Jeonghye Kim*, Youngchul Sung
(* denotes equal contribution)

KAIST SISReL

Previous RL exploration method

Exploration refers to the method of efficiently exploring the environment within limited resources and time in order to gather crucial information for the agent's decision-making process.

RND

Temporally extended exploration

epsilon-greedy

Task specific

There is yet no single exploration method that is shown to be universally effective across all tasks.

Invariable during training period

The required exploration strategy can vary over time during the training period within a given task, but existing methods do not reflect these characteristics.

LESSON - A Unified Framework for Multiple Exploration Strategies

LESSON is a unified framework that utilizes the **option-critic architecture**[1] and an off-policy structure to **effectively integrate multiple exploration strategies** for adaptive exploration strategy selection during the learning phase.

Option-critic architecture

Options (Temporally-extended actions) : a form of action generalization that captures high-level behavior by combining multiple sub-actions.

Call-and-Return Option Execution Model : a hierarchical control framework that improves the efficiency and effectiveness of the agent's navigation by breaking down complex tasks into a series of options.

- **Call** : The agent **selects an option** that represents a sub-goal or desired behavior based on its current state.
- **Return** : The agent **executes the selected option until its termination condition is met**.

[1] Bacon, Pierre-Luc, Jean Harb, and Doina Precup. "The option-critic architecture." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. No. 1. 2017.

Method - Behavior Policy Construction via Option-Critic

Key ideas of LESSON are

(1) to **replace the behavior policy with an option-critic architecture** whose

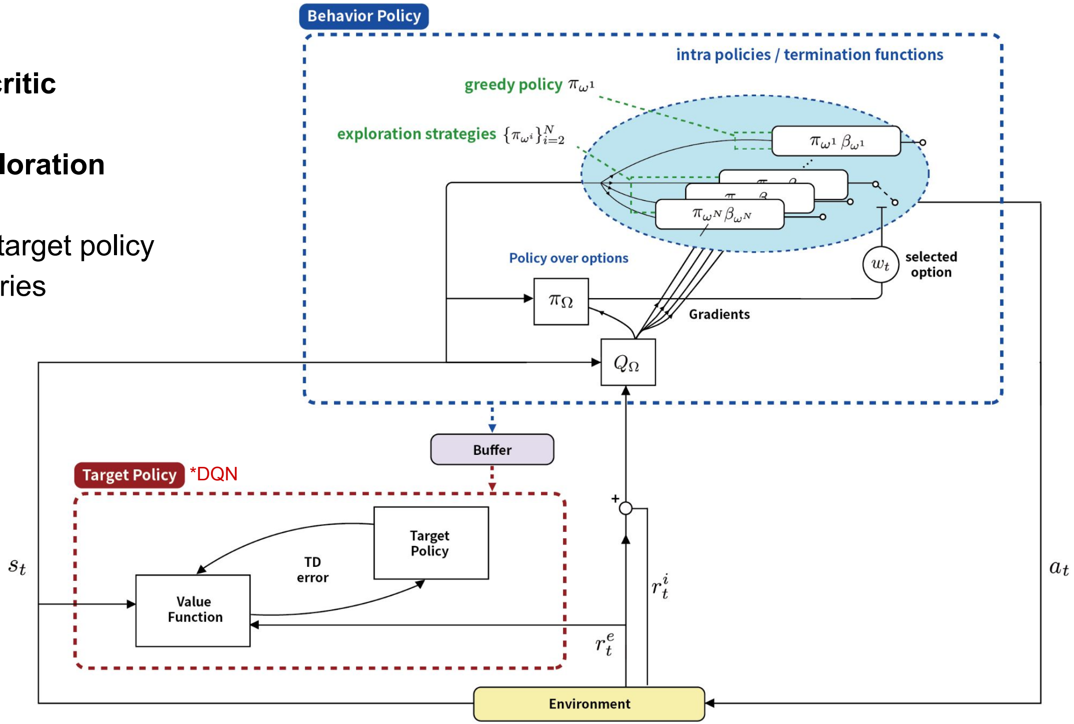
(N : number of option)

(2) intra-policies are defined by **N-1 component exploration strategies and the greedy policy**, and

(3) to train both the option-critic architecture and the target policy with two different objectives based on the trajectories generated by the option-critic architecture

Important factor for exploration-exploitation trade-off in sampling

- Inclusion of the greedy policy
- Design of the objective function



Method - Behavior Policy Construction via Option-Critic

Design of intra-policies

π_{ω^1}

The greedy intra-policy

select greedy action.
This policy aims to generate trajectories for exploitation by the target policy

π_{ω^2}

Random intra-policy

select a random action independently at each time step until the option is terminated

π_{ω^3}

TE-random intra-policy

select a random action when this option is chosen and then repeats the selected action until the option is terminated

π_{ω^4}

Prediction-error maximizing (PEM)

select the action that maximizes the sum of prediction-error intrinsic rewards

π_{ω^1} + π_{ω^2} = ϵ-greedy

π_{ω^1} + π_{ω^3} = ϵz-greedy

π_{ω^1} + π_{ω^4} = RND

Method - Learning Option-Critic

With the predefined intra-policies, we need to learn the **option selection policy** π_Ω and the **termination functions** $\{\beta_\omega\}$

Objective function for behavior policy :

$$J(\pi_\Omega, \{\beta_w\}) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \big(\underbrace{r^e_{t+1}}_{exploitation} + \alpha \underbrace{r^i_{t+1}}_{exploration} \big)\Big]$$

behavior policy should not only sample for exploration but also for exploitation for a trade-off between these two

Option-value function :

$$Q_\Omega(s_t, \omega_t) = \mathbb{E}\Big[\sum_{l=t}^{\infty} \gamma^{l-t} (r^e_{l+1} + \alpha r^i_{t+1}) \big| s_t, \omega_t\Big]$$

Learning Option-Value Function

$$\mathcal{L}(\theta_\Omega) = \mathbb{E}_{(s_t, w_t, r^e_t + \alpha r^i_t, s_{t+1}) \sim \mathcal{D}} \Big[(y_t - Q_\Omega(s_t, \omega_t; \theta_\Omega))^2 \Big]$$, where

$$y_t = r^e_t + \alpha r^i_t + \gamma \Bigg(\Big(1 - \beta_{\omega_t}(s_{t+1})\Big) Q_\Omega(s_{t+1}, \omega_t; \theta^-_\Omega) + \beta_{\omega_t}(s_{t+1}) \max_{w'} Q_\Omega(s_{t+1}, w'; \theta^-_\Omega) \Bigg)$$

Learning Termination Functions

$$\frac{\partial Q_\Omega}{\partial \theta_{\beta_\omega}} = -\mathbb{E}\Big[\nabla_{\theta_{\beta_\omega}} \beta_\omega(s_{t+1}; \theta_{\beta_\omega}) (Q_\Omega(s_{t+1}, \omega_t) - \max_\omega Q_\Omega(s_{t+1}, \omega)) \Big]$$

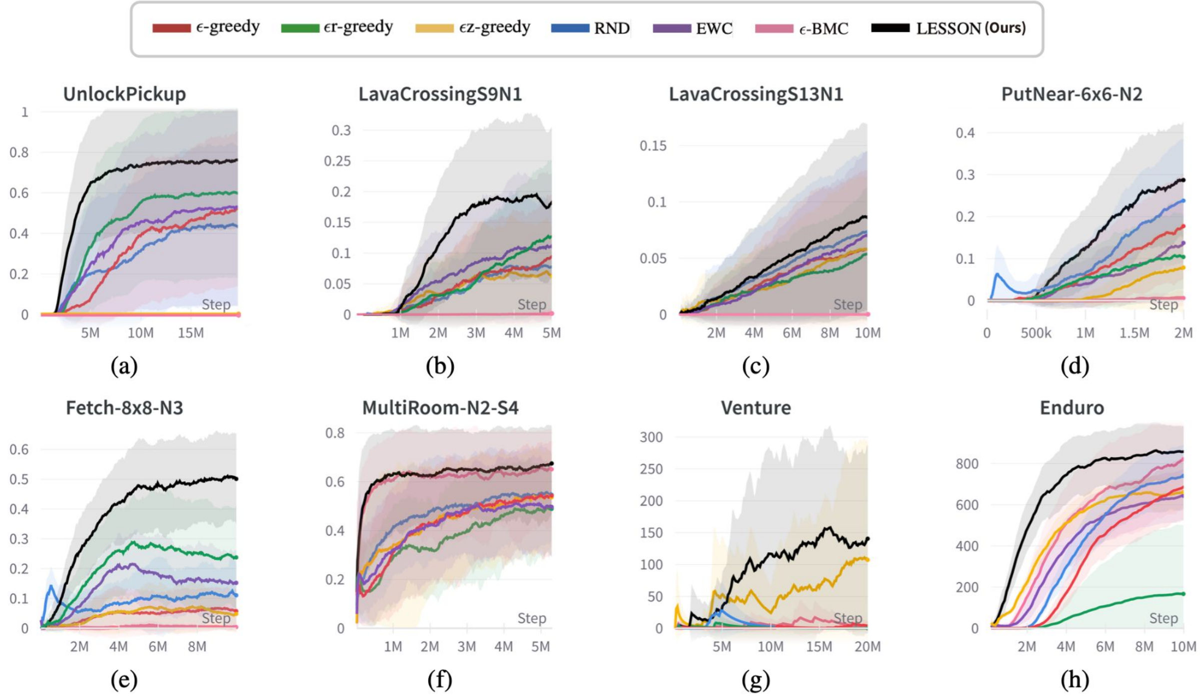
Experiments - Performance Comparison

Baselines

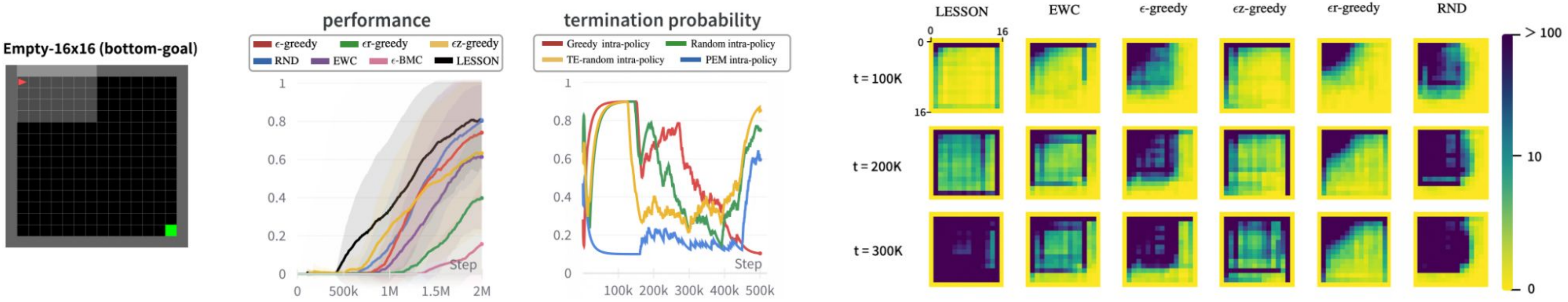
- ϵ-greedy (vanilla DQN)
- two simple DQN variants
  - ϵz-greedy, ϵr-greedy
- RND-based DQN
- Equal weight combining(EWC)
- ϵ-BMC, which learns ϵ

Environments

- 14 MiniGrid environments
- 4 Atari environments



Experiments - Exploration Behavior Analysis



(a) (b) (c) (d)

- Adaptive exploration-exploitation trade-off achieved through adaptive selection of intra-policies over time. (c)
- **LESSON combines visitation patterns of RND and ϵz-greedy to cover entire state space. (d)**

The LESSON We Learned

- It is crucial to **adaptively select exploration strategies** according to environmental characteristics and the learning phase to achieve an **efficient exploration-exploitation balance**.

- In order to create a unified framework, we propose to incorporate the option-critic architecture with intra-policies consisting of a **greedy policy and a set of exploration strategies**, and meticulously **designing the objective function**.

Thank you for listening