OptionRL: Estimating with Differential Equations (Draft ver.)

Dongsheng Hou*
Department of Computer Science and Engineering
Southern University of Science and Technology
12410421@mail.sustech.edu.cn

Yanqiao Chen*
Department of Computer Science and Engineering
Southern University of Science and Technology
12412115@mail.sustech.edu.cn

February 3, 2026

Contents

1 Introduction

2 Related Works

3 OptionRL

We proposed OptionRL, a novel framework that integrates the concept of options into reinforcement learning to enhance decision-making processes. OptionRL leverages differential equations to model the dynamics of options, allowing for more efficient learning and execution of complex tasks.

Usually in RL tasks, we encountered environments with sparse rewards, which makes it difficult for agents to learn optimal policies. To address this challenge, we introduce the concept of options pricing, which allow agents to refine their policies with both present values and future expected rewards. By incorporating options, agents can make more informed decisions, leading to improved performance in environments with sparse rewards.

3.1 Markov Decision Process Formulation

We formulate the multi-agent reinforcement learning problem as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where:

- $\mathcal{S}$ is the state space, representing the set of all possible configurations of the environment. At each time step t, the environment is in a state $s_t \in \mathcal{S}$.
- $\mathcal{A}$ is the action space, denoting the set of executable actions for the agents. Each agent i selects an action $a_{i,t} \in \mathcal{A}_i$, and the joint action is $a_t = \{a_{1,t}, \ldots, a_{N,t}\} \in \mathcal{A}$.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function, which defines the probability of transitioning to state s_{t+1} given state s_t and joint action a_t, denoted as $P(s_{t+1}|s_t, a_t)$.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. The team receives a global reward $r_t = \mathcal{R}(s_t, a_t)$ after executing the joint action a_t in state s_t.
- $\gamma \in [0, 1]$ is the discount factor, which determines the importance of future rewards.

The objective of the agents is to learn a joint policy $\pi(a_t|s_t)$ that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right]$$

In our OptionRL framework, we further enhance this formulation by attributing the global reward to individual agents and refining the value estimation using option pricing theory.

3.2 Levy Pricing

Like classical RL algorithms, OptionRL also relies on the Bellman equation to estimate the value functions. However, we extend the traditional Bellman equation by incorporating differential equations to model the evolution of options over time. This allows us to capture the dynamics of options more accurately, leading to better value function estimates.

In the OptionRL framework, we apply the two essential assumptions:

Assumption 3.1 (Levy Process Assumption)**.** *The noise term in the environment follows a Levy process.*

The another one is the classical assumption of Black-Scholes-Merton model:

Assumption 3.2 (Neutral Risk Assumption)**.** *The expected return of the option is the risk-free interest rate.*

Under the following assumptions, we can derive the differential equations that govern the evolution of options in the OptionRL framework. These equations allow us to estimate the value functions more accurately, leading to improved decision-making capabilities for agents.

Definition 3.1 (Levy Process)**.** *A stochastic process $X = \{X_t, t \geq 0\}$ is called a Levy process if it satisfies the following properties:*

1. *$X_0 = 0$ almost surely.*
2. *X_t has independent increments: for any $0 \leq t_0 < t_1 < \ldots < t_n$, the random variables $X_{t_1} - X_{t_0}, X_{t_2} - X_{t_1}, \ldots, X_{t_n} - X_{t_{n-1}}$ are independent.*
3. *X_t has stationary increments: for any $s, t \geq 0$, the distribution of $X_{t+s} - X_s$ depends only on t.*
4. *X_t is stochastically continuous: for any $t \geq 0$ and $\epsilon > 0$, $\lim_{h \to 0} P(|X_{t+h} - X_t| > \epsilon) = 0$.*

Such process can be rewritten as in differential form:

$$\mathrm{d}X_t = \mu(X_{t-})\mathrm{d}t + \sigma(X_{t-})\mathrm{d}W_t + \int_{\mathbb{R}\setminus\{0\}} \gamma(X_{t-}, z)\tilde{N}(\mathrm{d}t, \mathrm{d}z)$$

Applying the Ito formula for Levy processes, we can derive the following differential equation for the option price $V(t, S_t)$.

Theorem 3.3 (Levy Option Pricing Equation)**.**

$$\frac{\partial V}{\partial t} + rS_t\frac{\partial V}{\partial S_t} + \frac{1}{2}\sigma^2 S_t^2\frac{\partial^2 V}{\partial S_t^2} + \int_{\mathbb{R}\setminus\{0\}} [V(t, S_t + \gamma(S_t, z)) - V(t, S_t) - \gamma(S_t, z)\frac{\partial V}{\partial S_t}]\nu(\mathrm{d}z) - rV = 0$$

Where r is the risk-free interest rate, σ is the volatility of the underlying asset, and ν is the Levy measure associated with the jump component of the process.

Under the neutral risk assumption, we can solve the above differential equation to obtain the option price $V(t, S_t)$. This price can then be used to refine the value function estimates in the OptionRL framework, leading to improved decision-making capabilities for agents.

Theorem 3.4 (OptionRL Pricing Equation)**.** *The price of the value of a agent is given by the following differential equation:*

$$C_t = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[R_T|\mathcal{F}_t]$$

Where C_t is the option price at time t, R_T is the reward at terminal time T, and $\mathbb{Q}$ is the risk-neutral measure.

3.3 Shapley Value Attribution

In MARL, we often need to attribute the overall performance of a team to individual agents. To achieve this, we incorporate the concept of Shapley value from cooperative game theory into the OptionRL framework. The Shapley value provides a fair way to distribute the total reward among agents based on their contributions.

Definition 3.2 (Shapley Value)**.** *The Shapley value for an agent i in a cooperative game with a set of agents N and a characteristic function $v : 2^N \to \mathbb{R}$ is given by:*

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)]$$

Where the sum is over all subsets S of N that do not contain agent i.

We apply the Shapley value to attribute the overall reward in MARL.

3.4 OptionRL Framework

We propose the OptionRL framework, which integrates the concept of options into reinforcement learning to enhance decision-making processes. The framework consists of the following components:

- Running an episode in the environment to collect state, action, and reward data.
- Attribute the total reward to individual agents using the Shapley value.
- For each agent, estimate the option price using the Levy pricing equation to serve as the target value.
- Update the Critic network to approximate the option prices.
- Update the Actor policy using the advantage estimated by the Critic.
- Repeat the process until convergence.

Algorithm 1 OptionRL Framework (Actor-Critic Variant)

Initialize Actor π_θ and Critic V_ϕ with parameters θ, ϕ
Initialize replay buffer $\mathcal{D}$
for each episode **do**
    Reset environment and get initial state s_0
    Initialize trajectory $\tau = []$
    for $t = 0, \ldots, T$ **do**
        Select action $a_t \sim \pi_\theta(\cdot|s_t)$
        Execute action a_t, observe reward r_t and next state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in τ
    end for
    Calculate total reward $R = \sum_{t=0}^{T} r_t$
    Attribute reward to each agent i using Shapley Value: $R_i = \phi_i(v)$
    for each agent i **do**
        Estimate option price $C_{t,i}$ using Levy Pricing Equation (Theorem 3.2)
        Critic Update:
        Set target value $y_{t,i} = C_{t,i}$
        Update V_ϕ by minimizing Loss $\mathcal{L}_V(\phi)$:

$$\mathcal{L}_V(\phi) = \frac{1}{T} \sum_{t=0}^{T} \left(y_{t,i} - V_\phi(s_t)\right)^2$$

        Actor Update:
        Calculate Advantage $A_{t,i}$ (e.g., via TD error):

$$A_{t,i} = R_{i,t} + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$$

        Update π_θ using Policy Gradient:

$$\nabla_\theta J(\theta) = \frac{1}{T} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_{i,t}|s_t) A_{t,i}$$

        $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$
    end for
end for

4 Theoretical Analysis

4.1 MDP Formulation for OptionRL

Applying the MDP framework to OptionRL, we define the components not for a specific trading task, but for a general multi-agent coordination problem where agents are viewed as investment assets.

- **State Space $\mathcal{S}$**: The global state s_t encompasses the environmental context and the status of all agents. It captures the information necessary to compute the Shapley values and the parameters of the Levy process.

  $$s_t = [\mathbf{x}_t^{env}, \mathbf{h}_{1,t}, \ldots, \mathbf{h}_{N,t}]$$

  where $\mathbf{x}_t^{env}$ is the environment features and $\mathbf{h}_{i,t}$ is the hidden state or historical performance metrics of agent i.

- **Action Space $\mathcal{A}$**: $a_{i,t}$ represents the decision made by agent i to contribute to the cooperative task. The nature of actions depends on the specific domain (e.g., continuous control, discrete selection).

  $$a_t = (a_{1,t}, \ldots, a_{N,t})$$

- **Reward Assignment Mechanism**: Instead of directly optimizing the raw environmental reward, we adopt an investment perspective. We treat the sequence of an agent's marginal contributions (Shapley values $\phi_{i,t}$) as the returns of a stochastic asset. The effective reward assigned to agent i is derived from the option price $C_{i,t}$ of this asset:

  $$r_{i,t}^{\text{proxy}} = C_{i,t}(\phi_{i,t}, \tau, \sigma, \nu)$$

  Here, $C_{i,t}$ is estimated under the risk-neutral measure $\mathbb{Q}$ assuming the contribution process follows a Levy distribution. This mechanism rewards agents based on their potential future value and risk-adjusted expected returns, stabilizing learning in sparse-reward settings.

4.2 Convergence Analysis

4.3 Variance Analysis

A Proofs

A.1 Proof for theorem 3.4

B Experiment Details