Off-Beat Multi-Agent Reinforcement Learning

Extended Abstract

Wei Qiu
Nanyang Technological University
Singapore
qiuw0008@e.ntu.edu.sg

Weixun Wang
Tianjin University
Tianjin, China
wxwang@tju.edu.cn

Rundong Wang
Nanyang Technological University
Singapore
rundong001@e.ntu.edu.sg

Bo An
Nanyang Technological University
Singapore
boan@ntu.edu.sg

Yujing Hu
NetEase Fuxi AI Lab
Hangzhou, China
huyujing@corp.netease.com

Svetlana Obraztsova
Nanyang Technological University
Singapore
lana@ntu.edu.sg

Zinovi Rabinovich
Nanyang Technological University
Singapore
zinovi@ntu.edu.sg

Jianye Hao
Tianjin University
Tianjin, China
jianye.hao@tju.edu.cn

Yingfeng Chen
NetEase Fuxi AI Lab
Hangzhou, China
chenyingfeng1@corp.netease.com

Changjie Fan
NetEase Fuxi AI Lab
Hangzhou, China
fanchangjie@corp.netease.com

ABSTRACT

We investigate cooperative multi-agent reinforcement learning in environments with off-beat actions, *i.e.*, all actions have execution durations. During execution durations, the environmental changes are not synchronised with action executions. To learn efficient multi-agent coordination in environments with off-beat actions, we propose a novel reward redistribution method built on our novel graph-based episodic memory. We name our solution method as LeGEM. Empirical results on stag-hunter game show that it significantly boosts multi-agent coordination.

KEYWORDS

multi-agent coordination; multi-agent reinforcement learning

ACM Reference Format:
Wei Qiu, Weixun Wang, Rundong Wang, Bo An, Yujing Hu, Svetlana Obraztsova, Zinovi Rabinovich, Jianye Hao, Yingfeng Chen, and Changjie Fan. 2023. Off-Beat Multi-Agent Reinforcement Learning: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023,* IFAAMAS, 3 pages.

1 INTRODUCTION

Despite the recent successes of multi-agent reinforcement learning (MARL) in autonomous systems [1, 12] and real-time strategy (RTS) video games [10], learning effective multi-agent coordination in environments with off-beat actions remains challenging for MARL. Many cooperative MARL methods [3, 4, 6, 7] fail to learn efficient multi-agent coordination in environments where action durations are caused by off-beat actions. The main reason is TD-learning [8] fails when displaced rewards caused by action durations are used in training. To this end, we propose a novel reward redistribution method built on our novel graph-based episodic memory. We name our method as LeGEM. Empirical results on stag-hunter game show that it significantly boosts multi-agent coordination in environments with off-beat actions and achieves leading performance.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

2 PRELIMINARIES

MARL aims to learn optimal policies for all the agents in the team. With TD-learning and a global Q value proxy Q^{tot} for the optimal Q^*, $\{Q_i\}_{i=1}^N$ are optimized via minimizing the loss [2, 11]: $\theta^* = \arg\min_{\theta^*} \mathcal{L}(\theta) := \mathbb{E}_{D' \sim \mathcal{D}}[(y_t^{\text{tot}} - Q_\theta^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{u}_t))^2]$, where $y_t^{\text{tot}} = r_t + \gamma \max_{\boldsymbol{u}'} Q_{\bar{\theta}}^{\text{tot}}(\boldsymbol{s}_{t+1}, \boldsymbol{u}')$ and θ is the parameters of the agents. $\bar{\theta}$ is the parameter of the target Q^{tot} and is periodically copied from θ. D' is a sample from the replay buffer $\mathcal{D}$.

3 METHODOLOGY

3.1 Temporal Recency Reward Redistribution

To learning efficient multi-agent coordination in environment with off-beat actions for MARL methods. We redistribute rewards to agents' pivot timesteps (we will introduce the method for searching agent's povit timesteps in the following subsection). The *pivot timestep* of each agent is the timestep when the off-beat action was executed and later triggered the reward.

The timestep to which the reward should be distributed is called the *final pivot timestep*. We denote the *final pivot timestep* at timestep t as e_t. For a shared reward at timestep t, each agent's pivot timestep