2023 IEEE 12th International Conference on Cloud Networking (CloudNet)

RoMA: Resilient Multi-Agent Reinforcement Learning with Dynamic Participating Agents

1st Xuting Tang
Computer Science Department
Stevens Institute of Technology
Hoboken, USA
xtang18@stevens.edu

2nd Jia Xu
Computer Science Department
Stevens Institute of Technology
Hoboken, USA
jxu70@stevens.edu

3rd Shusen Wang
Xiaohongshu Inc
Shanghai, China
shusenwang@xiaohongshu.com

2023 IEEE 12th International Conference on Cloud Networking (CloudNet) | 979-8-3503-1306-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CloudNet59005.2023.10490082

***Abstract*—This paper presents RoMA, a novel resilient Multi-Agent Reinforcement Learning (MARL) framework designed to handle dynamic participating agents during centralized training, addressing the limitations of standard MARL frameworks in accommodating agent variability and enabling efficient adaptation and training of agents, thus providing a scalable and flexible solution for model training and execution in cloud computing environments. For standard MARL frameworks, if new agents need to join or existing agents leave unexpectedly due to unreliable communication channels, standard MARL models need to be rebuilt and trained from scratch because of their structural limitations, which is very time-consuming. RoMA addresses this issue with a novel neural network architecture and a few-shot learning algorithm to enable the number of agents to vary during centralized training. When new agents join, RoMA can adapt all agents to the change in a few shots, and when agents leave the training process unexpectedly, RoMA can continue training the remaining agents without disruption.**

Our experiments demonstrate that RoMA is at least 70 times faster at adapting to new agents compared to baseline methods, and it can handle the leaving of agents without affecting the training of other agents. RoMA is applicable to a wide range of MARL settings, including cooperative, competitive, independent, and mixed environments.

***Index Terms*—Multi-agent Reinforcement Learning, Resilient Model, Cloud Computing**

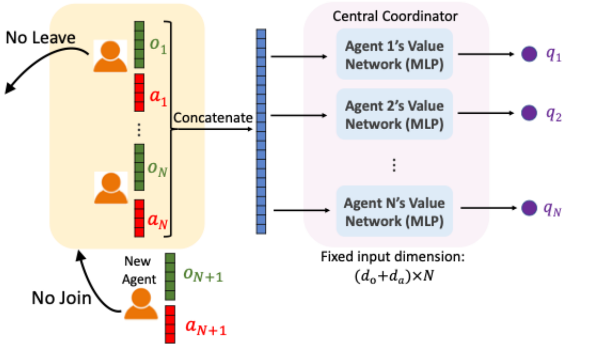
No Leave
o_1
a_1
o_N
a_N
Concatenate
New Agent
o_{N+1}
a_{N+1}
No Join
Central Coordinator
Agent 1's Value Network (MLP)
Agent 2's Value Network (MLP)
Agent N's Value Network (MLP)
q_1
q_2
q_N
Fixed input dimension: $(d_o+d_a)\times N$

Fig. 1: Architecture of the value networks (critics) of MADDPG. The observations and actions are concatenated into a single input vector. The value network of each agent expects input with fixed dimension: $(d_o + d_a)N$, where d_o and d_a are the dimensions of observation and action, respectively, and N is the number of existing agents. Agents can not join or leave the training because of the dimension restriction. Our work aims to enable dynamic participating agents.

I. Introduction

In recent years, reinforcement learning (RL) has been applied to a variety of tasks that exceed human ability, such as the games of Go and Poker, robotics, and autonomous driving. Many of these applications involve multiple agents and fall under the umbrella of MARL [1]. While there exists many algorithm paradigms, Centralized Training with Decentralized Execution (CTDE) has become the most popular MARL paradigm [2] due to its advantages in terms of scalability and stationary environment. For actor-critic methods using this paradigm, each agent's critic is trained in a centralized way using global context, while its policy is executed in a decentralized manner based on local information.

Existing actor-critics-based CTDE algorithms like MADDPG [3] and COMA [4] have demonstrated strong empirical performance. However, as shown in Figure 1, these methods concatenate the observations and actions of all agents into a single input vector for each critic. Centralized training in this manner assumes that the number of agents is fixed during training, meaning that new agents cannot join and existing agents cannot leave midway through the training process. However, this approach can pose challenges when applied to dynamic participating agents, particularly in cloud computing environments where the number of agents may vary. If the number of agents changes, the input dimension of the model also changes, requiring the network structure to be adjusted and a new model to be trained from scratch. Even if a model could handle dynamic input dimensions, the inclusion of a new agent with unlearned value and policy networks can slow down centralized training.

The above mentioned issues significantly limit the applicability of these approaches, because in real-life scenarios, it is common for the number of agents to change during centralized training. For example, during training, the communication channel between each agent and the central coordinator may not be available and stable. An agent may go offline unexpectedly during training, which will cause the termination of the training process; or some agents may face communication issues upon the training starts, and the existing CTDE