Mingyue Zhang[1], Zhi Jin[1], Yang Xu[2], Zehan Shen[3], Kun Liu[1], Keyu Pan[2]

[1] Key Lab of High Confidence Software Technologies Ministry of Education, Peking University
[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China
[3] Department of Computer Science and Technology, Nanjing University

Abstract

This paper focuses on the *multi-agent credit assignment* problem. We propose a novel multi-agent reinforcement learning algorithm called *meta imitation counterfactual regret advantage* (MICRA) and a three-phase framework for training, adaptation, and execution of MICRA. The key features are: (1) a *counterfactual regret advantage* is proposed to optimize the target agents' policy; (2) a meta-imitator is designed to infer the external agents' policies. Results show that MICRA outperforms state-of-the-art algorithms.

Background: Stochastic Game

A stochastic game is defined as a 7-tuple $\mathcal{G} = \langle S, N, A, T, R, O, \Omega \rangle$, where:

- S is a set of states. s^t is the state at time t;
- $N = \{1, ..., n\}$ is a set of n agents;
- $A = A_1 \times ... \times A_n$ is a set of joint actions, where A_i is the agent i's action set. $\vec{a}^t = [a_1^t, ..., a_n^t]$ is the joint action at time t;
- $T : S \times A \times S \to [0, 1]$ is the transition probability function;
- $O = O_1 \times ... \times O_n$ is a set of joint observations, where O_i is the agent i's observation set. Joint observation at time t is $\vec{o}^t = [o_1^t, ..., o_n^t]$;
- $\Omega : S \times A \to O$ is the observation function;
- $R = \{R_1, ..., R_n\}$ is the reward function set, where $R_i : S \times A \to \mathbb{R}$ is the reward function for agent i. ■

Background: Meta Learning

The objective of meta learning can be described as follows:

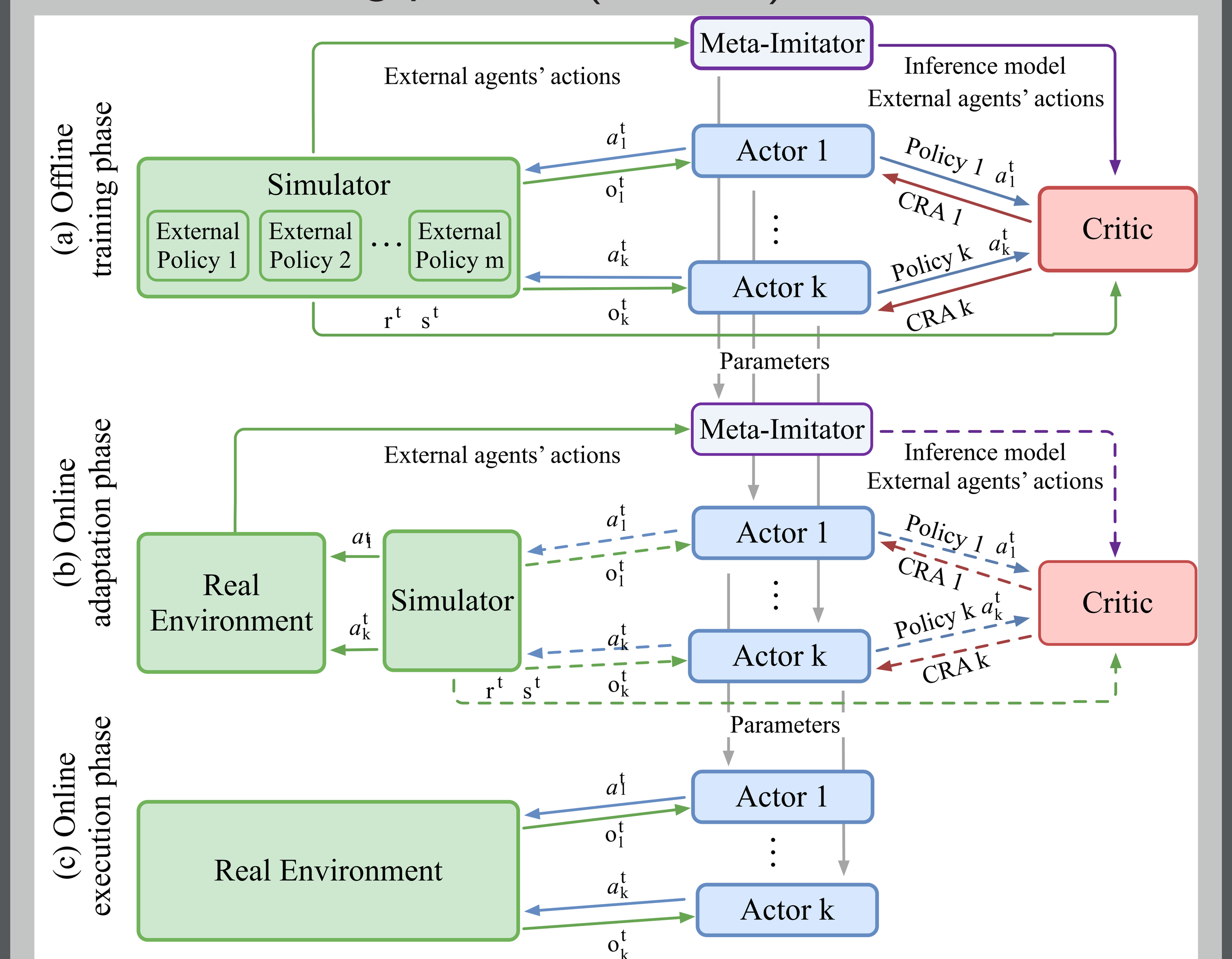
$$\min_{\theta} \mathbb{E}_{\mathcal{T}_i \sim \mathcal{T}} \Big[\sum_{t=1}^{H_i} \mathcal{L}_i(\mathrm{x}^t, \mathrm{a}^t)\Big] \tag{1}$$

where $\mathrm{x}^{t+1} \sim P_i(\cdot|\mathrm{x}^t, \mathrm{a}^t), \mathrm{a}^t \sim f(\cdot|\mathrm{x}^0, \mathrm{x}^1, ..., \mathrm{x}^t; \theta)$

Meta-learning has been widely used in supervised learning, and single-agent reinforcement learning.

Framework

The proposed three-phase framework integrates the CTDE (Lowe,17) paradigm with the meta-learning process (Finn,17).



Algorithm: Counterfactual Regret Advantage

(1) A centralized critic evaluates a *regret* value for an agent with the assumption that other agents follow the current policies; (2) Multiple actors independently update their individual policies minimizing the regret value.

Immediate counterfactual regret advantage:

$$\begin{aligned} \mathcal{A}_{T,i,\pi^T}(s,\vec{a}) &= v_{\pi^T|s\mapsto a_i}(s) - v_{\pi^T}(s) \\ &= \sum_{\vec{a}_{\tau-i},\vec{a}_\epsilon} \pi^T_{\tau-i}(\vec{a}_{\tau-i}|s)\pi^T_\epsilon(\vec{a}_\epsilon|s)Q(s,[a_i,\vec{a}_{\tau-i},\vec{a}_\epsilon]) \\ &\quad - \sum_{\vec{a}_\tau,\vec{a}_\epsilon} \pi^T_\tau(\vec{a}_\tau|s)\pi^T_\epsilon(\vec{a}_\epsilon|s)Q(s,[\vec{a}_\tau,\vec{a}_\epsilon]) \end{aligned} \tag{2}$$

CRA based policy gradient:

$$g_{\mathrm{cr},i} = \mathbb{E}_{s^t\sim D,\vec{a}^t\sim\pi}\Big[\sum_{t=0}^{H} \nabla_{\theta_i^a}\log(\pi_i(a_i^t|o_i^t;\theta_i^a))\mathcal{A}^\gamma_{i,\pi}(s^t,\vec{a}^t)\Big] \tag{3}$$

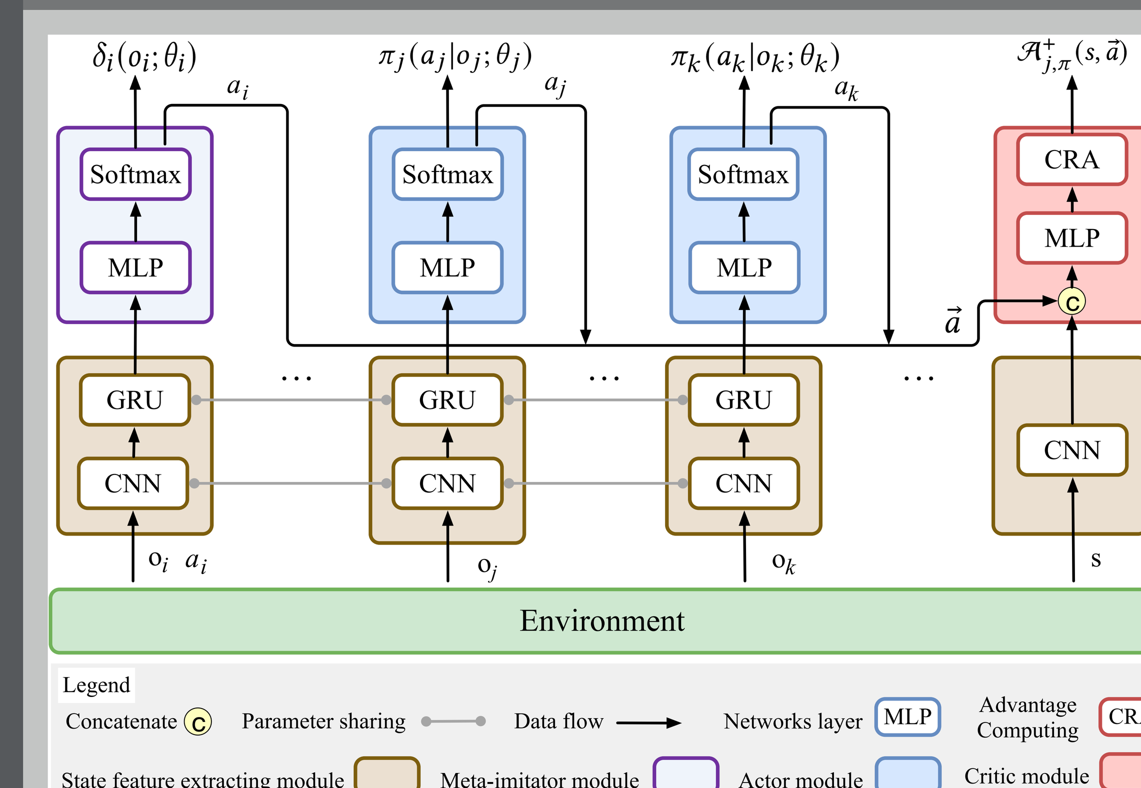
Algorithm: Meta Imitation Learning

The objective of MI is:

$$\begin{aligned} &\min_{\theta_i} \sum_{\mathcal{T}_i\sim p(\mathcal{T})} L^{im}_{\mathcal{H}_i}(\delta(\cdot;\theta_i')) \\ &\text{s.t. } \theta_i' = \theta_i - \alpha_{\mathrm{adp}}\nabla_{\theta_i}L^{im}_{\mathcal{H}_i}(\delta(\cdot;\theta_i)) \end{aligned} \tag{4}$$

where $p(\mathcal{T})$ is the distribution of all external agents' policies. θ_i is the meta parameters which will be used as initial parameters in online adaptation phase.

Algorithm: Network Structures



- *State feature extractor*, which extracts the high-level feature from the raw data.
- *Meta-imitator*, which monitors the external agents' observation-action pairs, and learns an inference model to predict their behaviors with meta-imitation learning. The module's output layer is softmax, which generates the probability of all available actions to the external agents.
- *Actor*, which trains the individual policy for each targeted agent using the CRA policy gradient.
- *Critic*, which trains a joint Q-function using temporal difference learning and computes CRA for instructing each actor to update its policy correctly.

Evaluation

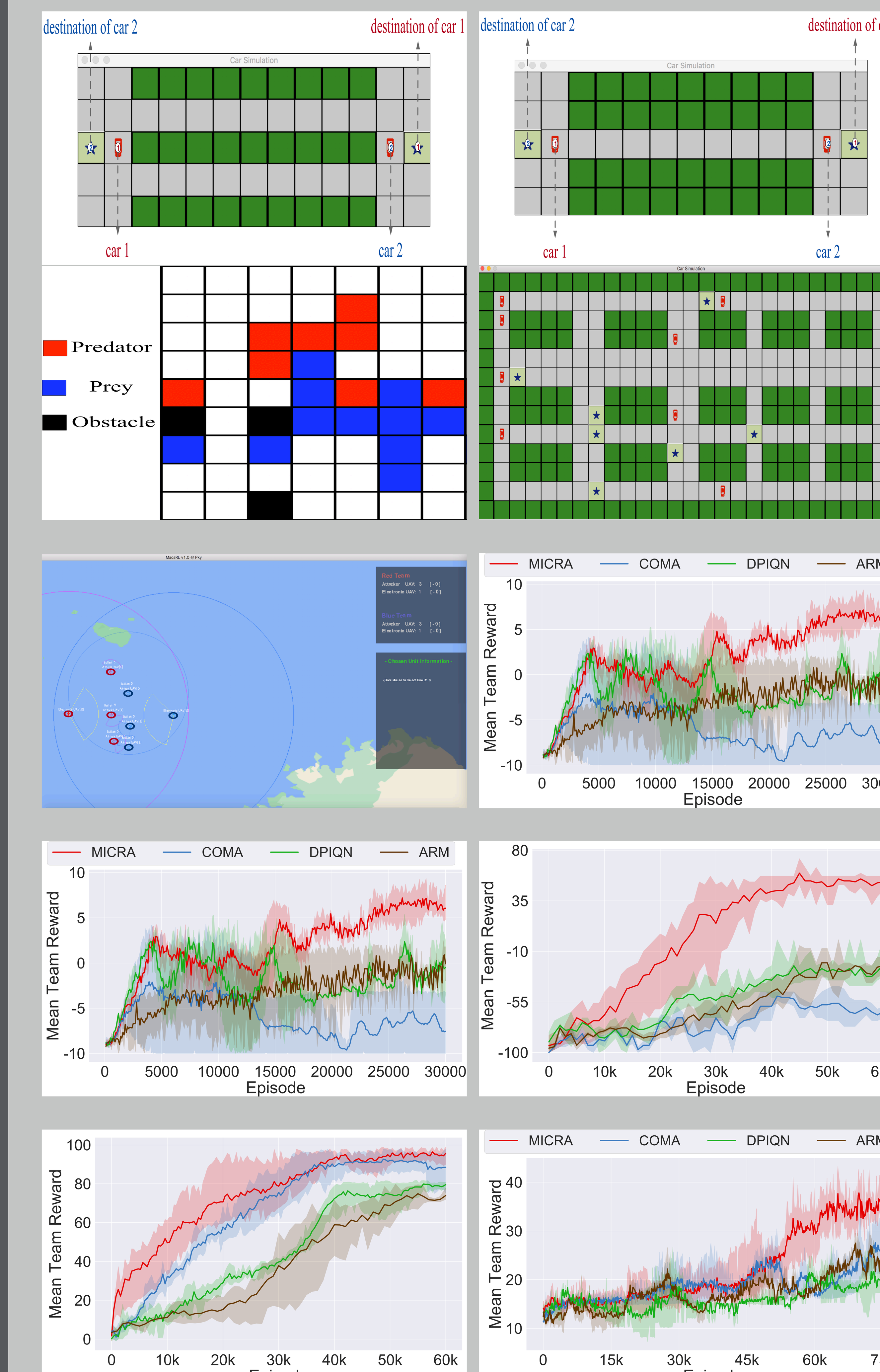


Figure: Offline training: the learning curves on different tasks (red line is ours).