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 $G = (S, N, A, R, O, \Omega)$, where:
- $S$ is a set of states, $s^i$ 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; $a^i$ is the joint action at time $t$;
- $T : S \times A \times S \to [0, 1]$ is the transition probability function;
- $O = Q_1 \times ... \times Q_n$ is a set of joint observations, where $Q_i$ is the agent $i$’s observation set. Joint observation at time $t$ is $o^t = [o^1, ..., o^n]$;
- $\Omega : S \times A \to O$ 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 for agent $i$.

The objective of meta learning can be described as follows:

$$\min_\theta \mathbb{E}_{T,T'} \left[ \sum_{t=1}^{T} L(s^t, a^t) \right]$$  
where $s^t \sim R(\cdot | s^t, a^t), a^t \sim \pi(\cdot | s^t, \theta)$.

**Framework**

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

**Algorithm: Countertualfactual 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:

$$A_{i,t} (s, \bar{a}_i) = v_i(s) - v_i(s) \mid \bar{a}_i$$

$$= \sum_{a_i \in A_i} \pi_i(a_i \mid s) \mathbb{E} \left[ R(s, a_i, \bar{a}_{-i}) \right] - \sum_{a_i \in A_i} \pi_i(a_i \mid s) \mathbb{E} \left[ R(s, \bar{a}_i, \bar{a}_{-i}) \right]$$

CRA basic policy gradient:

$$\nabla_\theta \mathbb{E}_{s^t \sim \mathcal{D}(s^t), a^t \sim \pi(\cdot | s^t, \theta)} \left[ \sum_{t=0}^{T} \nabla_\theta \log(\pi(a_t | s^t, \theta)) A_{i,t} (s^t, a^t) \right]$$

**Algorithm: Meta Imitation Learning**

The objective of MI is:

$$\min_{\theta_i} \mathbb{E}_{T} \left[ L^n_i (\theta_i) \right]$$

s.t. $\theta_i = \theta_i - \alpha_{\theta_i} \nabla_{\theta_i} L^n_i (\theta_i)$

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

**Evaluation**

- 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.

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