Learning Implicit Credit Assignment for Multi-Agent Actor-Critic

Meng Zhou*, Ziyu Liu*, Pengwei Sui, Yixuan Li, Yuk Ying Chung

School of Computer Science
The University of Sydney
Agenda

- Multi-Agent Reinforcement Learning Setting
- LICA (Learning Implicit Credit Assignment)
- Adaptive Entropy Regularization
- Experiments
Multi-Agent Systems

- Many real-world problems can be solved with multi-agent systems
  - e.g. traffic management, network routing, cluster optimization
Dec-POMDP

- Multiple Agents: $A = \{1, \ldots, n\}$
- Shared Global State: $s_t \in S$
- Agent Observation: $z^a_t \in Z$ for agent $a$ drawn from $O(s_t, a): S \times A \rightarrow Z$
- Agent Action: $u^a_t \in U_a$ for agent $a$
- Joint Action: $u_t \in (U_1 \times \cdots \times U_n) \equiv U^n$
- State Transition Probability: $P(s_{t+1}|s_t, u_t): S \times U^n \times S \rightarrow [0, 1]$
- Shared Joint Action Reward: $r(s_t, u_t): S \times U^n \rightarrow \mathbb{R}$
- Goal: Maximize discounted accumulated reward: $R_t = \sum_{i=0}^{T} \gamma^i r_{t+i}$
Decentralised Control Approach

- Separate policies for every agent
- Only consider individual agent’s action space
- Agent select action following its policy:

\[ \pi^a(u_t^a | z_t^a) : Z \times U_\alpha \rightarrow [0, 1] \]

- Problem: Credit Assignment

Image Credit: Agarwal, 2019
Credit Assignment Problem

- All agents share the same global reward
- Difficult to attribute the reward to individual agent actions
- Training agent directly using global reward leads to poor performance and slow convergence

Image Credit: Liang and Liaw, 2018
Credit Assignment Problem / Limitations

Explicit Credit Assignment:

- Strategies to explicitly measure agent contributions against baselines
  - e.g. COMA (based on difference rewards)
- Hard to model complex cooperation behaviours
- Computationally expensive

Implicit Credit Assignment:

- Learning a mapping from individual Q-value to joint action Q-value
  - e.g. VDN, QMIX, QTRAN
- Limited to value-based algorithms with slow convergence
- Difficult to extend to continuous action space
Agenda

- Multi-Agent Reinforcement Learning Setting
- LICA (Learning Implicit Credit Assignment)
- Adaptive Entropy Regularization
- Experiments
LICA

● **Motivation:**

○ Credit assignment may not require an explicit formulation

○ Agent policies could be jointly learned to maximize the joint-action Q-value, given that:

  ■ the policy gradients of centralized critic carry sufficient information for learning optimal cooperation

  ■ the training procedure enforces a sustained level of agent exploration
LICA

- **Critic Training:**
  - Mean Square Error between sampled accumulated rewards and current estimation

\[
\min_{\theta_c} \mathbb{E}_{t,s_t,u^1_t,\ldots,u^n_t} \left[ R_t - Q^\pi_{\theta_c} \left( s_t, u^1_t, \ldots, u^n_t \right) \right]^2
\]

\[
R_t = \sum_{i=0}^{T} \gamma^i r_{t+i} \quad u^a_t \sim \pi^a_{\theta_a}
\]
LICA

- **Policy Training:**
  - Directly maximize the trained critic network output by optimizing policy networks

\[
\max_{\theta} \mathbb{E}_{t,s_t,z^1_t,\ldots,z^n_t} \left[ Q^\pi_{\theta_c} \left( s_t, \pi^1_{\theta_1} \left( \cdot \mid z^1_t \right), \ldots, \pi^n_{\theta_n} \left( \cdot \mid z^n_t \right) \right) + \mathbb{E}_a \left[ \mathcal{H} \left( \pi^a_{\theta_a} \left( \cdot \mid z^a_t \right) \right) \right] \right]
\]

\[
\theta = \{ \theta_1, \ldots, \theta_n \}
\]
LICA

- **Policy Training:**
  - Directly maximize the trained critic network output by optimizing policy networks

\[
\max_{\theta} \mathbb{E}_{t,s_t,z_1^t,\ldots,z_n^t} \left[ Q_{\theta_c}^\pi \left( s_t, \pi_{\theta_1}^1 \left( \cdot \mid z_t^1 \right), \ldots, \pi_{\theta_n}^n \left( \cdot \mid z_t^n \right) \right) + \mathbb{E}_a \left[ H \left( \pi_{\theta_a}^a \left( \cdot \mid z_t^a \right) \right) \right] \right]
\]

\[
\theta = \{\theta_1, \ldots, \theta_n\}
\]
Algorithm 1 Optimization procedure for LICA

1: Randomly initialize $\theta$ and $\theta_c$ for the policy networks and the mixing critic respectively.
2: for number of training iterations do
3: Sample $b$ trajectories $D_1, ..., D_b$ with $D_i = \{z_{0,i}, s_{0,i}, u_{0,i}, r_{0,i}, ..., z_{T,i}, s_{T,i}, u_{T,i}, r_{T,i}\}$.
4: Calculate accumulated rewards $\{R_{0,i}, ..., R_{T,i}\}$ for each trajectory $D_i$.
5: for $k$ iterations do
6: Update the mixing critic by descending its gradient according to Eq. (1):

$$\nabla_{\theta_c} \sum_{i=1}^{b} \sum_{t=1}^{T} \left[ R_{t,i} - Q_{\theta_c}^\pi \left( s_{t,i}, u_{t,i}^1, ..., u_{t,i}^n \right) \right]^2$$

(3)

7: Update the decentralized policy networks by ascending their gradients according to Eq. (2):

$$\nabla_{\theta} \sum_{i=1}^{b} \sum_{t=1}^{T} \left[ \frac{1}{n} \sum_{a=1}^{n} \mathcal{H} \left( \pi_{\theta_a}^\pi \left( \cdot | z_{t,i}^a \right) \right) \right]$$

(4)
Naive Critic Network Structure
Naive Critic Network Structure

- Joint action value gradient does not contain state information as \( s_t \) is part of the concatenation.
- Gradient expects a redundant update to \( z_t \), giving sub-optimal/inaccurate estimates.

\[ \nabla_{\theta} Q \]
LICA Critic Network Structure

(a) Policy Network

(b) Mixing Critic
LICA Critic Network Structure

(a) Policy Network

(b) Mixing Critic

\[ Q \]
Agenda

- Multi-Agent Reinforcement Learning Setting
- LICA (Learning Implicit Credit Assignment)
- Adaptive Entropy Regularization
- Experiments
Adaptive Entropy Regularization

- **Early Convergence to local sub-optimality:**

  Policy-based RL algorithms could easily converge to local optimum due to insufficient exploration and over-confidence in the current policy
Adaptive Entropy Regularization

- Vanilla Entropy Regularization

$$\max_{\theta} \mathbb{E}_{t,s_t,z_t^1,...,z_t^n} \left[ Q_{\theta_c} \left( s_t, \pi_{\theta_1}^1 (\cdot | z_t^1), \ldots, \pi_{\theta_n}^n (\cdot | z_t^n) \right) \right] + \mathbb{E}_a \left[ \mathcal{H} \left( \pi_{\theta_a}^a (\cdot | z_t^a) \right) \right]$$

$$\mathcal{H} = \beta H \left( \pi^a (\cdot | z^a) \right) = \beta \mathbb{E}_{u^a \sim \pi^a} \left[ - \log \pi^a (u^a | z^a) \right]$$

![High entropy vs Low entropy diagram](chart.png)
Adaptive Entropy Regularization

- Vanilla Entropy Regularization

\[
\max_\theta \mathbb{E}_{t, s_t, z_{t,1}, \ldots, z_{t,n}} \left[ Q_{\theta_c} (s_t, \pi_{\theta_1}^1 (\cdot \mid z_{t,1}^1), \ldots, \pi_{\theta_n}^n (\cdot \mid z_{t,n}^n)) + \mathbb{E}_a \left[ H (\pi_{\theta_a}^a (\cdot \mid z_{t}^a)) \right] \right]
\]

\[
H = \beta H (\pi^a (\cdot \mid z^a)) = \beta \mathbb{E}_{u^a \sim \pi^a} [- \log \pi^a (u^a \mid z^a)]
\]

Drawbacks:

- High sensitivity to initial regularization strength \( \beta \)
- Once policy starts to converge, the same regularization term hardly encourages further exploration. However, in complex environments, consistent exploration is often required to obtain good performance.
Adaptive Entropy Regularization

- Vanilla Entropy Regularization

\[ \mathcal{H} = \beta H\left(\pi^a(\cdot | z^a)\right) = \beta \mathbb{E}_{u^a \sim \pi^a} \left[ -\log \pi^a(u^a | z^a) \right] \]

\[ d\mathcal{H} = \left[ \frac{\partial \mathcal{H}}{\partial p^a_1}, \ldots, \frac{\partial \mathcal{H}}{\partial p^a_k} \right] = \left[ -\beta \left( \log p^a_1 + 1 \right), \ldots, -\beta \left( \log p^a_k + 1 \right) \right] \]

\[ d\mathcal{H}_i := -\beta (\log p^a_i + 1) \]
Adaptive Entropy Regularization

- Adaptive Entropy Regularization
  - Increase regularization strength for low entropy policy (divide a smaller term)
  - Decrease regularization strength for high entropy policy (divide a larger term)

\[ d\mathcal{H}_i := -\beta (\log p_i^a + 1) \]

\[ d\mathcal{H}_i := -\lambda \cdot \frac{\log p_i^a + 1}{H(p^a)} \]
Agenda

- Multi-Agent Reinforcement Learning Setting
- LICA (Learning Implicit Credit Assignment)
- Adaptive Entropy Regularization
- Experiments
Experiments: LICA Critic Component Study

1-Step Traffic Junction:

- Two cars controlled by two agents trying to pass the junction in sequential order
- Multiple optimal joint actions exist, which makes explicit credit assignment methods work poorly

- (pass, pass) → 0
- (wait, wait) → 0
- (pass, wait) → 1
- (wait, pass) → 1
Figure 5: 1-Step Traffic Junction agent move probabilities throughout training for the two optimal outcomes. LICA agents learn to pass in order much faster than COMA agents.
Experiments

- **StarCraft II Micromanagement Task**
  - Agents control a group of units to defeat another group of units controlled by heuristics
  - Rich set of actions and higher environment complexity
Experiments

- StarCraft II Micromanagement Task
  - Compare with state-of-the-art baselines released in NeurIPS 2019

*Figure 2. Performance comparison on StarCraft II 1c3s5z (Easy), 5m_vs_6m (Hard), MMM2 (Super Hard).*
Ablation Experiments

- StarCraft II Micromanagement Task
  - Compare with Naive Critic Network Structure (MLP)
Ablation Experiments

- **StarCraft II Micromanagement Task**
  - Compare with vanilla entropy regularization ($\beta$)
  - Compare with Naive Critic Network Structure (MLP)
Thanks