Policy Agents with MADDPG in SCML2020World

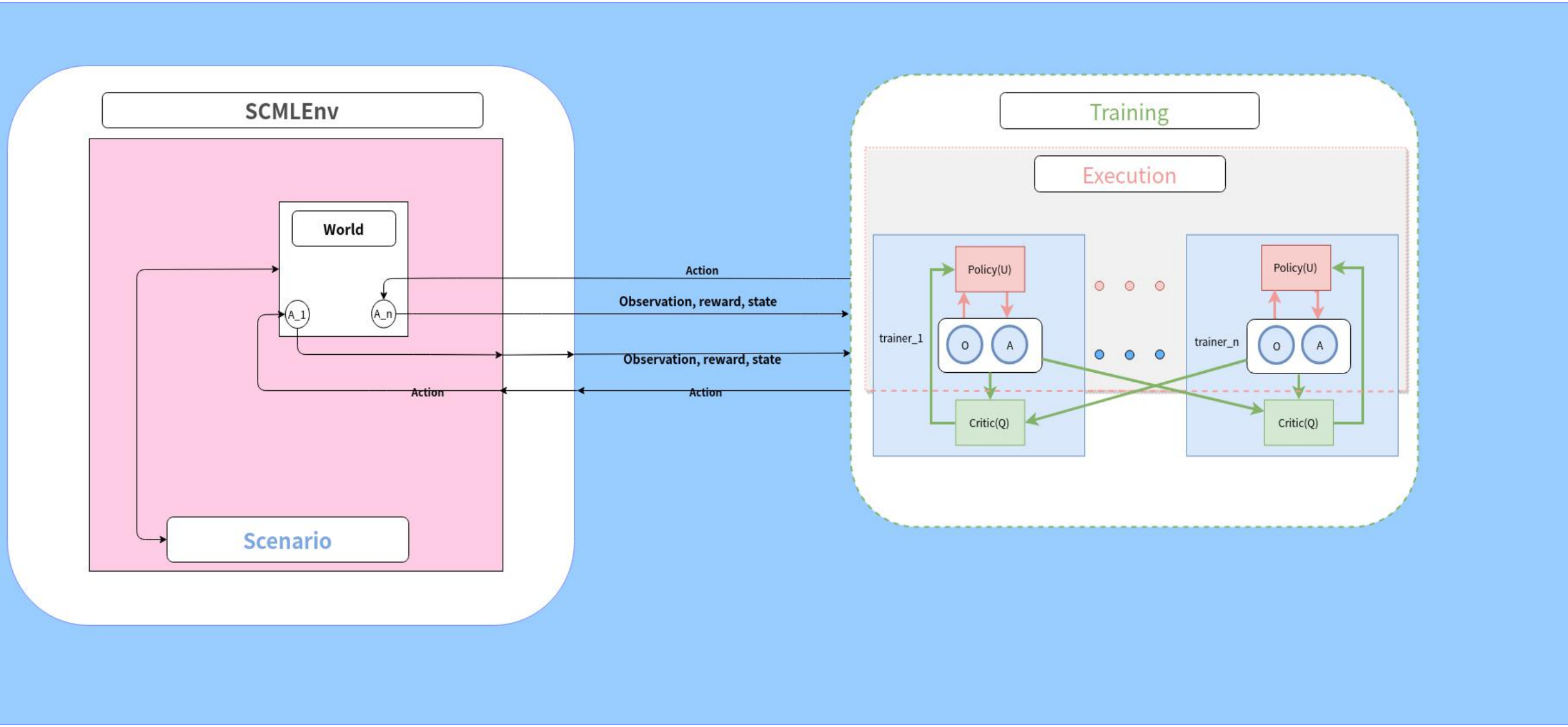
YUE NING, 19.12.2020,
Karlsruhe, Germany

Problems, Goal

- Problem
  - Dynamically decide the ranges allowed for negotiation issues based on the market conditions and the status of agents, heuristic policy agent could only consider part of the situation. Many important factors will be ignored or wrongly considered. ***How to let agents learn to decide range by themselves?***
  - Many problems can arise in multi-agent environments, ***One problem is that each agent's policy is changing as training progresses, and the environment becomes non-stationary from the perspective of any individual agent.*** Traditional RL approaches such as Q-learning or policy gradient are poorly suited to multi-agent environments, and can not solve this problem.

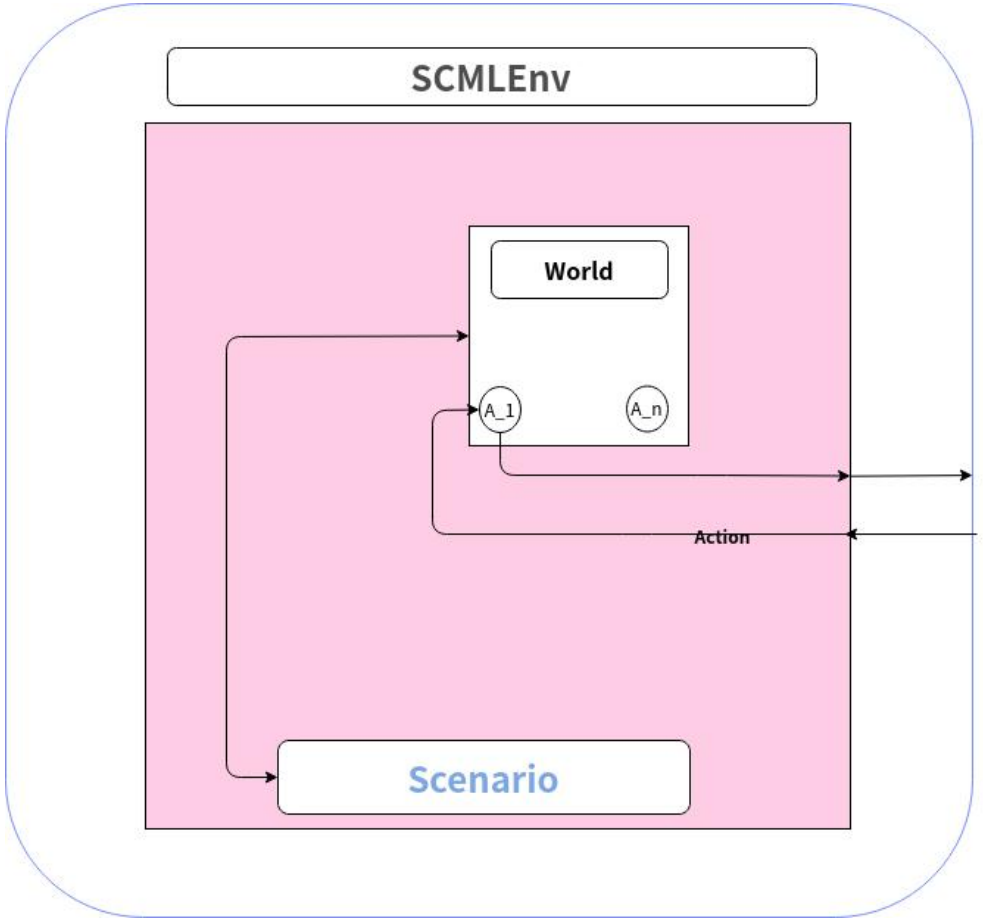
- Goal
  - let agents learn how to decide the range of negotiation issues by observing the market conditions and agent conditions.

Overview of Model (SCMLEnv + MADDPG)



multi-agent decentralized actor, centralized critic

environment



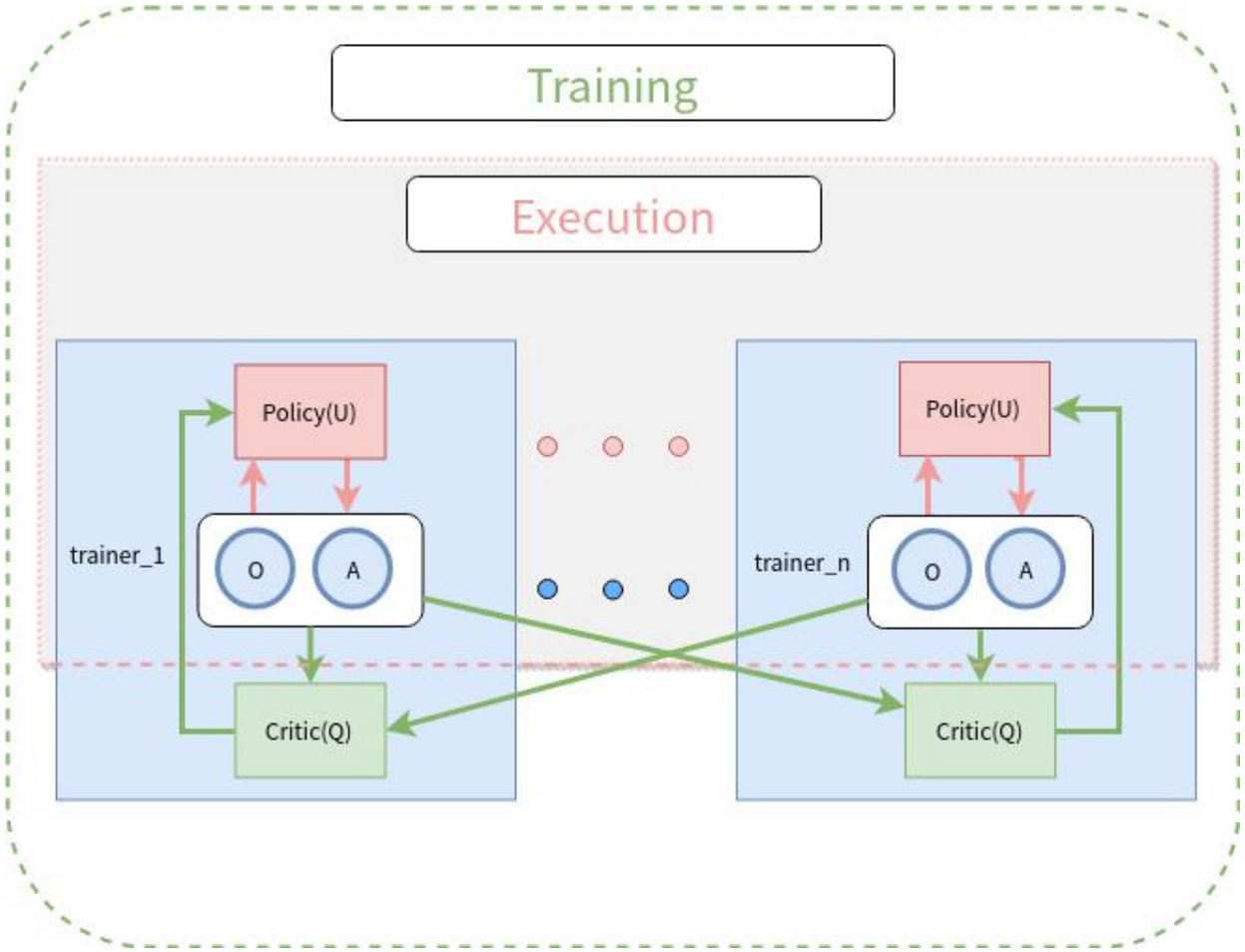
Environement: SCMLEnv <= Gym.env,
transfer informations of reward, observation to trainer

World: TrainWorld <= SCML2020World
world running logic

Agent: MyComponentsBasedAgent <= SCML2020Agent
agents with learning capabilities run in the SCML2020World

Scenario:
make environement, make world, reward callback, observation callback, done callback

Model of Training, Execution(MADDPG)



Training:
allowing the policies to use extra information to ease training, so long as this information is not used at test time.

Execution:
observe the local information, decide the action of next step (increase or decrease ranges allowed negotiation issues.)

$$\nabla_{\theta_i} J(\boldsymbol{\mu}_i) = \mathbb{E}_{\mathbf{x},a\sim\mathcal{D}}[\nabla_{\theta_i}\boldsymbol{\mu}_i(a_i|o_i)\nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, ..., a_N)|_{a_i=\boldsymbol{\mu}_i(o_i)}]$$

$$\mathcal{L}(\theta_i) = \mathbb{E}_{\mathbf{x},a,r,\mathbf{x}'}[(Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_N) - y)^2], \quad y = r_i + \gamma\, Q_i^{\boldsymbol{\mu}'}(\mathbf{x}', a'_1, \ldots, a'_N)\big|_{a'_j=\boldsymbol{\mu}'_j(o_j)}$$

Summary

Solved Problems

1. learn to decide the ranges allowed negotiation issues
2. non-stationary environment

Remaining Problems

1. SCML2020World is not fixed. **Does the trained strategy performs well in other world configurations?**
2. At the beginning of every step in the world, new ranges allowed negotiation issues are created. But this ranges will not change within one world step.

Possible next step

1. Train agents' ability: Wisely decide sell negotiation issues.
2. Implement agents' ability: Wisely decide buy negotiation issues.
3. Expand to different worlds, learning different strategies in different worlds.
4. Implements a predictor to predict the type of world, when running the agent in tournaments.