Learning to Cooperate with Unseen Agents Through Meta-Reinforcement Learning

**Introduction**

In this work, we posit that a cooperative agent needs the ability to adapt its policy to the dynamics of unseen agents and this can be achieved with meta-reinforcement learning. Meta-learning paradigm considers learning an adaptive behaviour using data.

**Method**

We consider two-player cooperative tasks, where our agent will have to cooperate with a set of agents from a pool, $P$, of partner agents. Similar to previous work our meta-RL agent is implemented with an RNN and trained with a distribution of partners $P$. We use two cooperative environments with different cooperative circumstances: level game and speaker-listener. The objective of the meta-RL agent is to maximise the expected return $E_{p \sim P, r \sim \pi} \left[ \sum_{t=0}^{H_1} \gamma^t r_t \right]$. When an agent is deployed into the real world, test-time scenarios might differ from the ones that are used during the training. In this section, we examine the ability of a meta-RL agent to extrapolate under unexpected situations including working under longer horizon and partner switching. We find that the meta-RL agent is robust when it performs to longer horizon length in both environments. The performance is stable throughout the entire trajectory.

**Neural network models**

<table>
<thead>
<tr>
<th>$\pi$</th>
<th>$Y$</th>
<th>$\pi$</th>
<th>$Y$</th>
<th>$\pi$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRU</td>
<td>RNN</td>
<td>GRU</td>
<td>a-RNN</td>
<td>GRU</td>
<td>r-RNN</td>
</tr>
</tbody>
</table>

**Continual adaptation**

![Figure 1: Architecture variations. Four variations of architectures considered in this work. Each architecture has different input features. These architectures are tested in the experiments to study the impact of each component of meta-RL.](image)

![Figure 2: Chance of getting a reward within a trajectory. The graph shows the chance of getting a reward at each timestep in a trajectory from level game (left) and speaker-listener (right).](image)

![Figure 3: Continual adaptation. We test meta-RL agents in episodes with longer horizon and partner switching. The agents have stable performance when put into extremely long episodes. Also, they can adapt to multiple partners when the partner is changed periodically within an episode.](image)

**Limitation**

We also study the impact of the number of training partners. We consider the number of training partners from the set of (5,10,15,20). We observe that the ad hoc teamwork performance and cooperation get better as we increase the number of training partners. Next, we study the impact of diversity of the training partners. Instead of randomly selecting training partners from the pool of all possible agents, we select the training partners such that they only come from a specific part of a behaviour space. This is similar to out-of-distribution testing in supervised learning. The training score of meta-RL agent does not deteriorate when trained under this skewed distribution. However, we notice that the test-time performance reduced significantly in both environments.