MASER: Multi-Agent Reinforcement Learning with Subgoals Generated from Experience Replay Buffer

Jeewon Jeon

Advisor : Youngchul Sung

15 July, 2022

Smart Information Systems Research Lab
Dept. of Electrical Engineering, KAIST

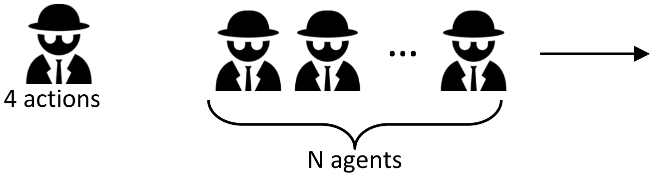
Content

1. Introduction & Contribution
2. Proposed Algorithm
3. Experiment and Result
4. Conclusion

Introduction

Multi agent reinforcement learning with sparse rewards

- Joint action space grows exponentially with the number of agents.
- Typically, agents receive a global reward.
- Reward comes only under certain circumstances, e.g., success/failure.
- We approach multi-agent sparse RL with sub-goals and
- propose a method to determine sub-goals by exploiting experience replay buffer.



- Total joint action space : 4^N
- 4^N contributes only 1 sparse & global reward
- Which agent and actions contributes more?

Contributions

Our method has three contributions :

- **Generating and assigning subgoals**: MASER finds subgoals for agents from the experience replay buffer. This eliminates the necessity of predesigning good subgoals based on domain knowledge.
- **Giving individual rewards**: MASER designs individual rewards for local agents to reach their subgoals while maximizing the joint return.
- **Actionable distance relevant to Q-learning** : To determine the intrinsic reward based on the Euclidean distance in the transformed domain, MASER uses representational transform based on actionable distance relevant to Q-learning derived from Amari 0-divergence.

Proposed Algorithm : Generating Subgoals

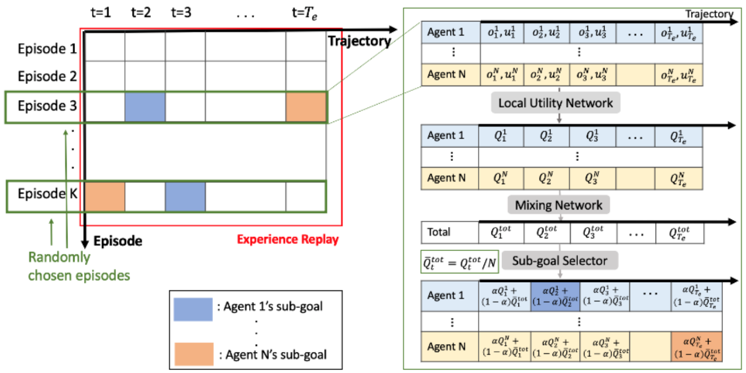


Figure1. Generating Subgoals

$$t_*^i = argmax_t[\alpha \underbrace{\boxed{Q^i\left(o_t^i, argmax_u Q^i(o_t^i, u)\right)}}_{\text{Local Q-value}} + (1-\alpha)\underbrace{\boxed{Q^{tot}(\boldsymbol{o}_t, \boldsymbol{u}_t)}}_{\text{Global Q-value}}]$$

Local Q-value → Generating different subgoals for each agent

Global Q-value → Consider other agents' status

Subgoal for agent i : $o_g^i = o_{t_*^i}^i$

Proposed Algorithm : Overall Reward Design

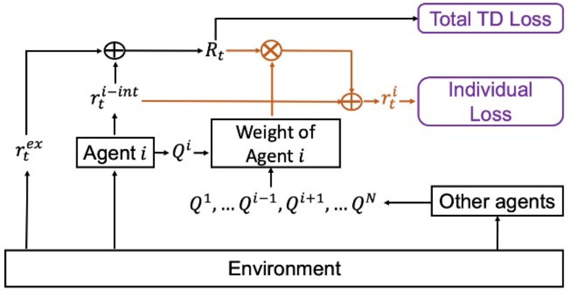


Figure2. Overall reward design diagram

Both mixing and local utility parameters are learned to achieve the subgoals as well as maximizing the overall extrinsic reward

Individual intrinsic reward

$$r_t^{i-int} = -\left\| \boxed{\phi^i}\left(o_t^i\right) - \phi^i\left(o_g^i\right) \right\|_2$$

Actionable representational transform

→ Agents to reach their subgoals

Proxy reward

$$R_t = \boxed{r_t^{ex}} + \lambda \frac{1}{N} \sum_{i=1}^{N} r_t^{i-int}$$

Extrinsic reward (sparse)

→ To update mixing network θ

Individual reward

$$r_t^i = \boxed{softmax(\max_u Q^i(o_t^i, u))} \cdot R_t + \lambda r_t^{i-int}$$

Contribution of Agent i to overall extrinsic reward

→ To update utility parameter θ_i

Proposed Algorithm : Q-Function-Based Representation Learning

Actionable distance

$$D_Q(o_t^i, o_g^i) = 1 - \frac{< Q^i(o_t^i,\cdot), Q^i(o_g^i,\cdot) >}{\|Q^i(o_t^i,\cdot)\| \times \|Q^i(o_g^i,\cdot)\|}$$

→ 1 – cosine similarity between $Q^i(o_t^i,\cdot), Q^i(o_g^i,\cdot)$

Loss function

$$L_D(\phi^i) = E_{o_t^i}\left[\left\|\phi^i(o_t^i) - \phi^i(o_g^i)\right\|_2 - D_Q(o_t^i, o_g^i)\right]^2$$

→ By minimizing loss function, ϕ^i is learned to represent actionable distance

Proposed Algorithm : Overall Flow

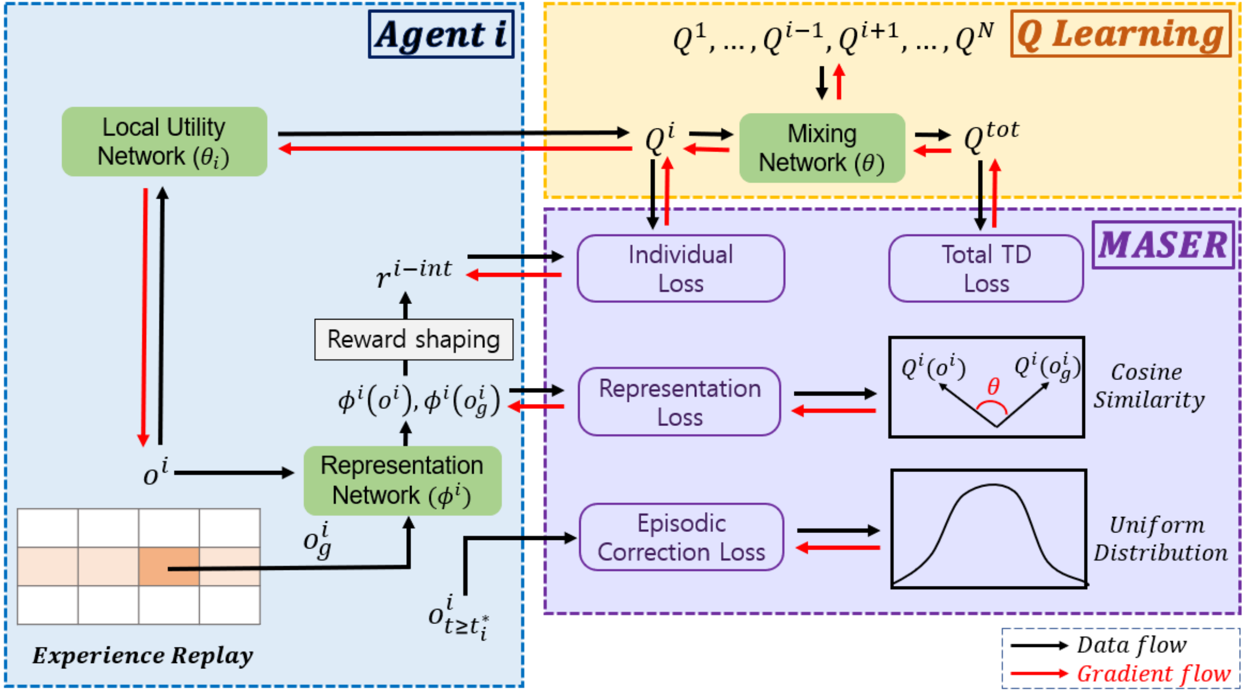


Figure3. Overflow

- Reach subgoals by giving intrinsic reward
- Novel distance function with cosine similarity
- Episodic correction after reaching subgoals.

Results (Experiments on StarCraft 2 with Sparse Rewards)

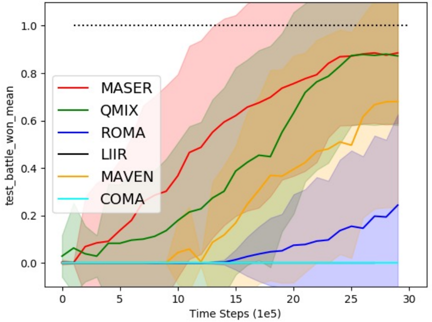


Figure4(a). 3m

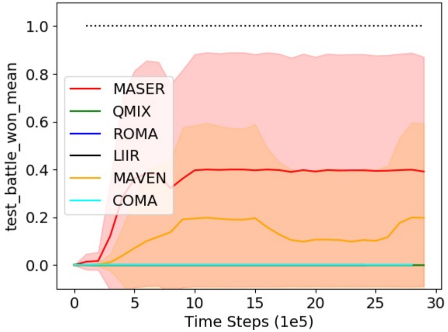


Figure4(b). 2m_vs_1z

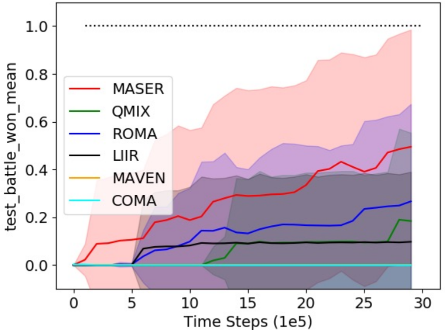


Figure4(c). 8m

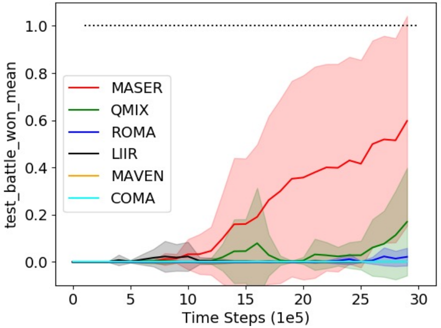


Figure4(d). 2s3z

Example of Generated Subgoals

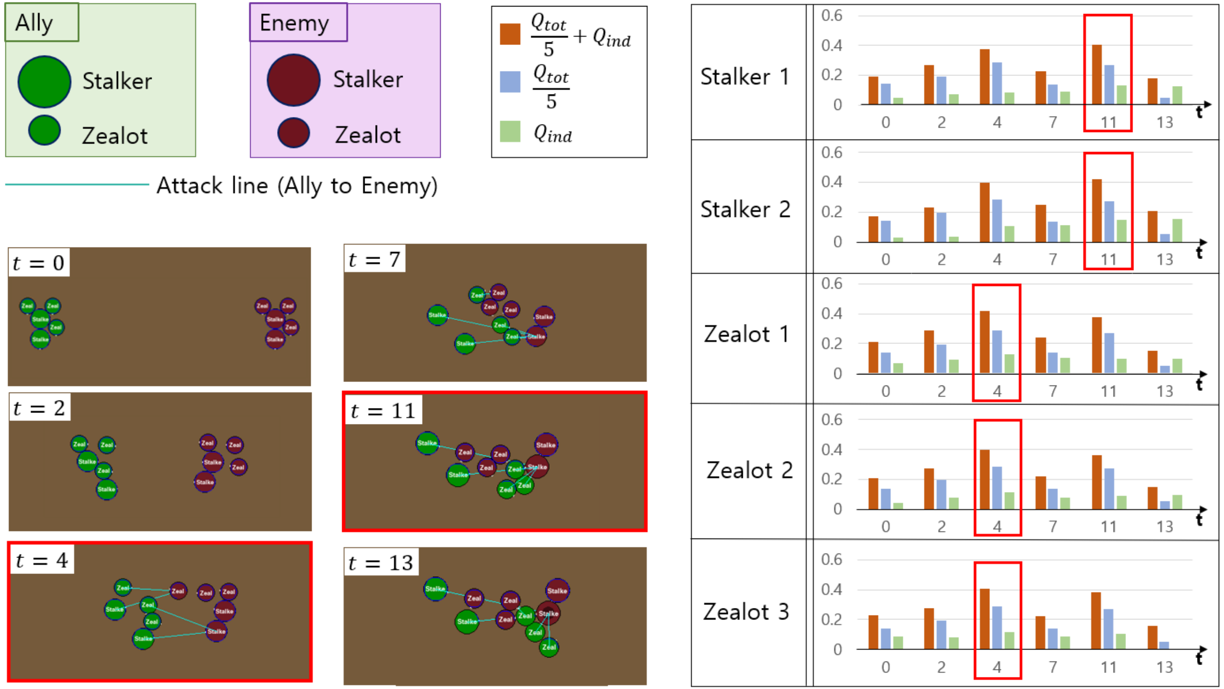


Figure5. Visualization of suboglas on 2s3z

Thank you!