[IE801] Final Project Report

SC2ENV MARL

Team Members

Jaeseok Huh | ** | jshuh@kaist.ac.kr | *School of Computing*
**** Jang** | ** | **@kaist.ac.kr | *Department of Mathematical Sciences*
**** Lee** | ** | **@kaist.ac.kr | *Department of Industrial Engineering*

1. Main Algorithm

We trained three value-based CTDE (Centralized training with decentralized execution) method--VDN [1], QMIX [2], and QTRAN [3]--which train individual agents via a centralized value function (CVF)--and one adversarial method COMA (Counterfactual Multi-Agent policy gradients) [4].

CTDE learns CVF shared by all the agents during the training; while each agent's action is executed in a decentralized manner during the execution phase. The CVF works as a proxy for the environment of each agent. However, since it has no different schemes than in single-agent settings, it leads to larger estimation error in multi-agent environment. In order to reduce the difficulty of decomposing the centralized value function into individual value functions, many algorithms impose extra structural assumptions onto the hypothesis space of the CVF during training. VDN [1], QMIX [2], and QTRAN [3] assume that the optimal joint action is equivalent to the collection of each agent's optimal action. Also, there are inherent difficulties in estimating an accurate of CVF in MA environments: the curse of dimensionality, the challenges with non-Markovian property, partial observability, and the interaction among agents. COMA [4] is an on-policy actor-critic method that uses a carefully designed counterfactual baseline to perform credit assignment. We trained these models except for VDN [1], as known to perform worse than the others, for ten million iterations with a few minor engineering modifications. (Figure 1)

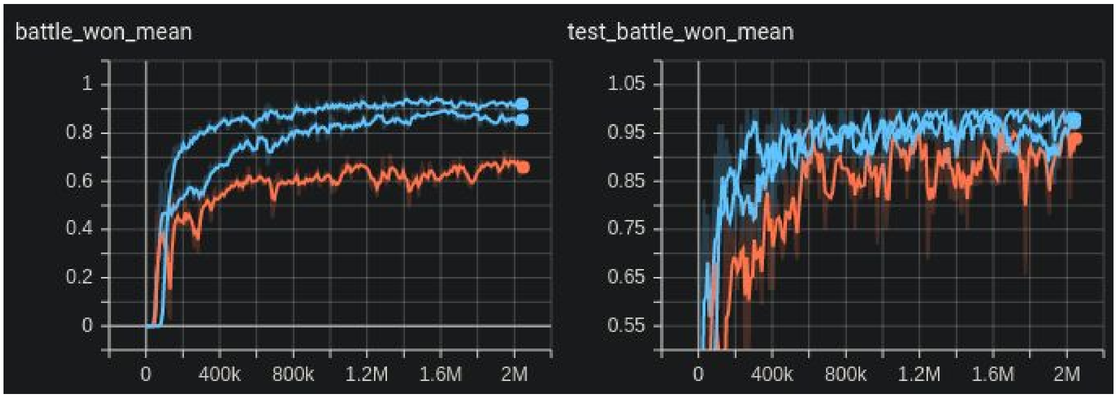

Figure 1. Winning ratio of QMIX (bottom), QTRAN (middle), COMA (top).
The left panel shows the training result and the right shows the test result.
The x-axis represents the number of iterations.

As QMIX showed continued uprise until the end of the training, we further trained QMIX for an additional eight million steps. (Figure 2) We did not train the others because of our limited time and the already plateaued result. As QMIX showed the highest winning rate while conspicuously more stable (compare it with QTRAN and COMA in Figure 1), thus, we searched for the variations of the QMIX model.

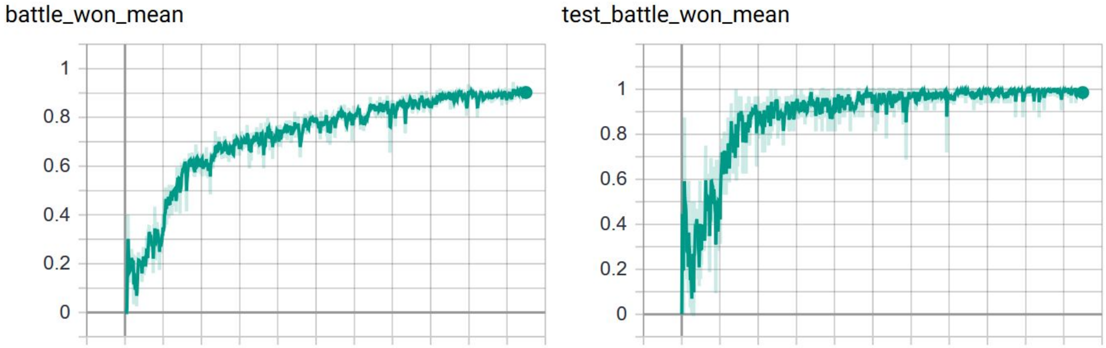

Figure 2. QMIX training result (left) and test (right) with ten million iterations. Each column of the grid denotes one million steps of training.

2. Details of Tweaks

Before explaining the tweaks, we identify three challenges in multi-agent (MA) environment. First, the dimension of the joint action space grows exponentially with the number of agents. Secondly, the interaction among individual agents causes non-stationarity, therefore, unstable environments. Last, it is non-trivial to credit each agent properly given a single global reward.

A sample efficient and effective value-based method, named SMIX(λ) [5], was proposed in November 2019. It belongs to the CTDE framework as well with some key features differ from QMIX. First, at the beginning of updating Q values with the expected errors of target policy, there was large variance due to the product of the ratio and the "curse of dimensionality" issue of the joint action space. Second, off-policy training could improve the centralized value function estimation with an off-policy based CVF learning method, which removes the need to explicitly rely on the centralized greedy behavior (CGB) assumption during training. It is implemented by setting the importance sampling ratio as 1.0 and using instead an experience replay memory to store the most recent off-policy execution. Last, it adopts a new loss function objective, so-called λ-return, as the TD target estimator. Using λ as a coefficient of n-step return enabled a better balance in the bias and variance trade-off and to account better for the environment's non-Markovian property.

Although SMIX(λ) has the same deep network structure as QMIX in aggregating total Q value, it abandons the Q-learning updating rule when updating its CVF. It made SMIX have a larger hypothesis space than QMIX. Moreover, by restricting weights of mixing networks to be non-negative--sufficient condition for CGB assumption--SMIX has smaller hypothesis space than QTRAN. (Figure 3) We trained SMIX with a few minor engineering modifications for ten million iterations. (Figure 4)

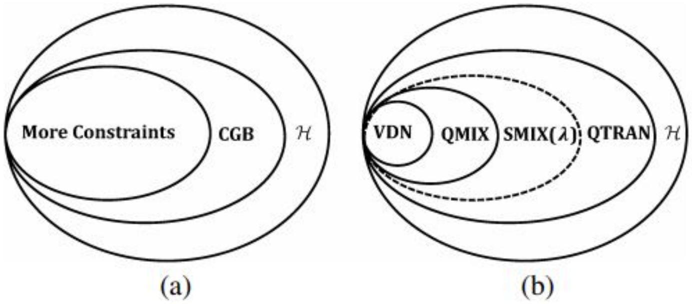

Figure 3. (a) The size of the hypothesis space corresponding to different constraints. (b) the relationship of hypothesis spaces of different algorithms.

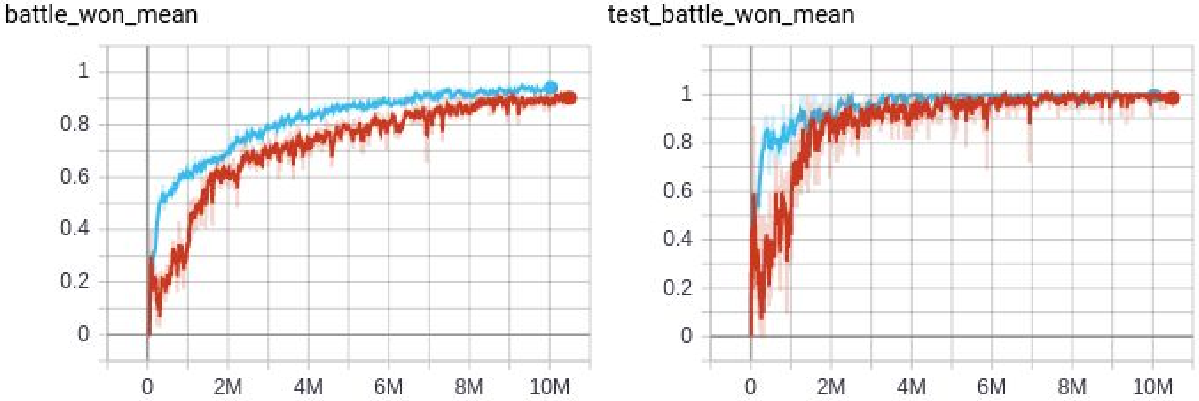

Figure 4. Training (left) and test (right) result of QMIX (red) and SMIX (blue). The x-axis represents the number of iterations.

3. Test Performance

With `num_runs=1000`, the winning rate was 0.998.
For setting up a running environment, please see README.md.

References

[1] Sunehag, Peter, et al. "Value-decomposition networks for cooperative multi-agent learning." arXiv preprint arXiv:1706.05296 (2017).
[2] Rashid, Tabish, et al. "QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning." arXiv preprint arXiv:1803.11485 (2018).
[3] Son, Kyunghwan, et al. "QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning." arXiv preprint arXiv:1905.05408 (2019).
[4] Foerster, Jakob N., et al. "Counterfactual multi-agent policy gradients." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
[5] Yao, Xinghu, et al. "SMIX ($\lambda $): Enhancing Centralized Value Functions for Cooperative Multi-Agent Reinforcement Learning." arXiv preprint arXiv:1911.04094 (2019).