Additional Experiments for CAVDN

1: *The benfits introduced by intention module.*

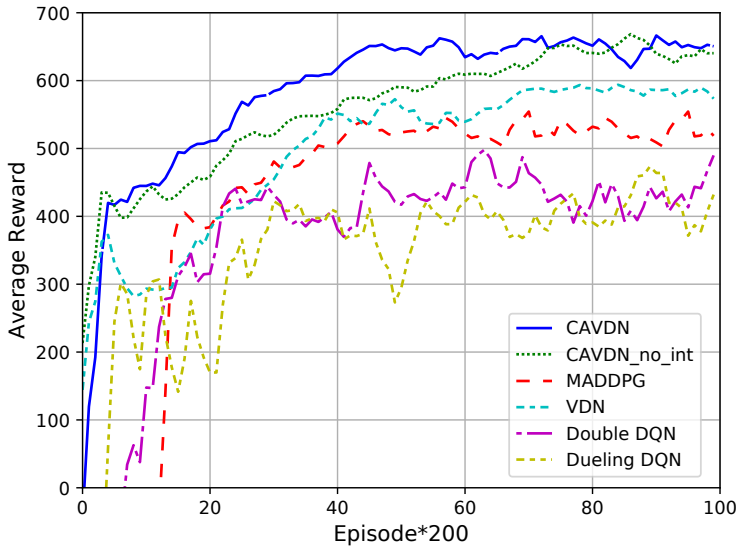


Fig. 1. The performance comparison between the proposed framework and one without the intention module. "CAVDN_no_int" refers to the approach without the intention module.

In our manuscript, the *intention module* shares the same RNN with the *encode module* , the only difference between them is that the intention module uses a different FNN. Intuitively, the intention h_n is trained to learn how to behave with only local observation, while the encoded message m_n is trained to learn how to share messages for cooperation among all agents. The intention h_n is also introduced to accelerate the training process of CAVDN. The combining network of agent n receives many messages from other agent. However, initially, the parameters of neural networks are randomly generated, the messages from others may disturb the local decision. To avoid such effect, we take intention h_n as extra information including encoded messages m_n. Therefore, the combine network will have enough information from local observation initially, including both m_n and h_n.

We have added the results to compare the difference between CAVDN and CAVDN without intention modules (denoted as CAVDN_no_int). In Fig.1, compared to CAVDN_no_int, we can see that the reward of CAVDN increases more rapidly and converges more early. This validates the benefits by introducing intention module.

2: *Comparison with Baselines.*

We have added experiments results for baseline CommNet [1] and attention mechanism used in [2] approaches. COMMNET directly process the received messages from other agents by arithmetic mean. "att_enc_only" applies attention mechanism-based message encoding module based on raw observation to replace our encoding module in CAVDN. "att_enc_+CAVDN_enc" combine our encoding module and attention mechanism-based encoding module, by letting m_n to be the input of the "att_enc". "att_pro_+CAVDN_enc" applies attention mechanism to replace the combining module in CAVDN to process the receiving messages.

[1] Sainbayar Sukhbaatar, Arthur Szlam, Rob Fergus. Learning Multiagent Communication with Backpropagation. NIPS 2016: 2244-2252

[2] Abhishek Das, Thophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, Joelle Pineau. TarMAC: Targeted Multi-Agent Communication. ICML 2019: 1538-1546

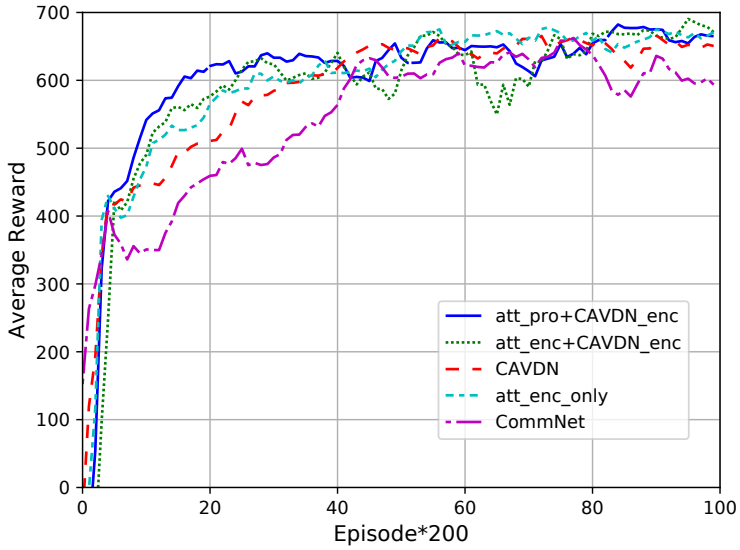


Fig. 2. The performance comparison of different approaches.

As shown in Fig. 2, the performance of COMMNET is lower than CAVDN and TARMAC. This is because COMMNET use too simple model, the arithmetic mean, for the received messages, which introduce much information loss. We can also see that "att_pro_+CAVDN_enc", "att_enc_+CAVDN_enc" and "att_enc_only" firstly increase rapidly than CAVDN. This shows that by introducing attention mechanism in both encoding of observation or the process of received messages, the convergence speed can be increased. Nonetheless, the convergent performance of all approaches except for CommNet are close to each other.

3: *Centralized training and the decentralized execution (CTCE) performance.*

In reinforcement learning, CTCE with a centralized controller does not always perform better than CTDE, even without considering the communication overhead in CTCE. For instance, in the reference

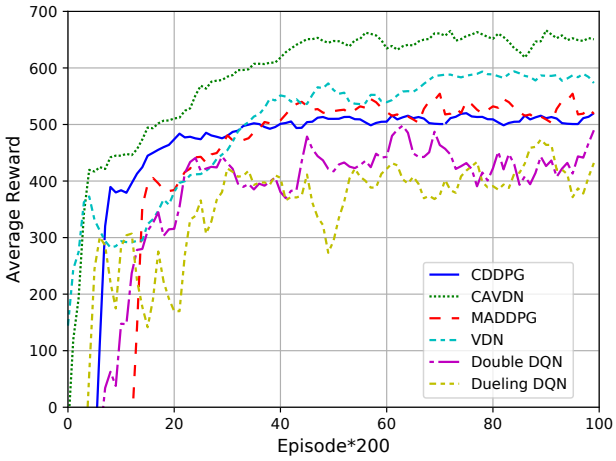


Fig. 3. The performance of the CTCE framework.

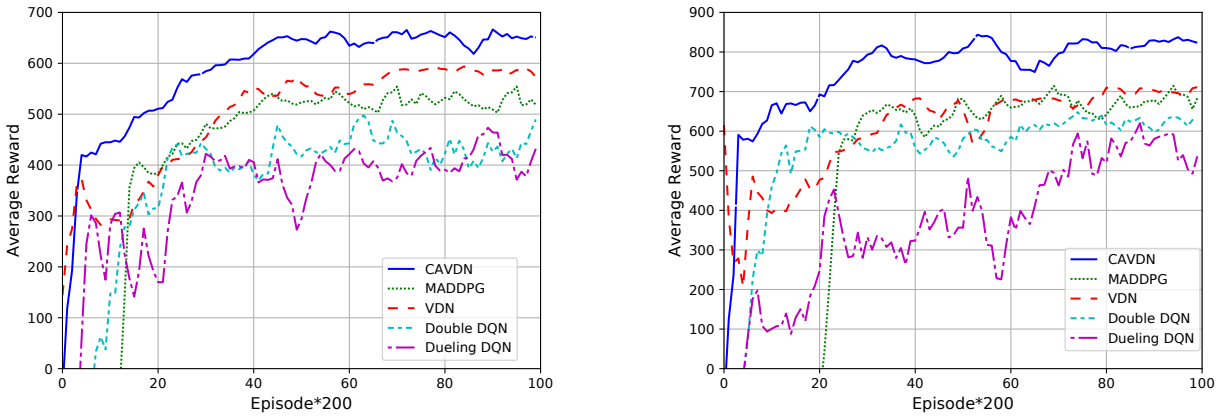
[1], the centralised approach fails by learning inefficient policies with only one agent active and the other being lazy. This happens when one agent learns a useful policy, but a second agent is discouraged from learning because its exploration would hinder the first agent and lead to worse team reward.

Nonetheless, we have performance the experiment with centralized DDPG (CDDPG) methods, which follow the CTCE framework. In CDDPG, the actor network takes input the observations of all agents, and output the actions of all agents as in [1].

As shown in Fig. 3, the performance of CDDPG is close to MADDPG, which is worse than CAVDN.

[1] P. Sunehag, G. Lever, et. al, Value-decomposition networks for cooperative multi-agent learning based on team reward, in AAMAS 2018, Stockholm, Sweden, July 10-15, 2018, 2018, pp. 20852087.

4: *Experiments with more users.*



(a) The performance comparison with "12" users.

(b) The performance comparison with "24" users.

Fig. 4. Performance comparison for different number of users.

We have added the experiment results of performance comparison with the number of users $K = 24$. As shown in Fig. 4, CAVDN (our proposed) is also better than other baseline approaches. When $K = 12$, as shown in Fig. 4(a), the performance gain of CAVDN compared to VDN is 8%. When $K = 24$, as shown in Fig. 4(b), the performance gain of CAVDN compared to VDN increases to 12%. This validates that the importance of agent message is more noticeable, when there are insufficient UAV servers that serves many ground users.

***5*: Trajactory of UAVs.**

The numerical results in Fig. 4 of our original manuscript is the average performance over 500 test samples.

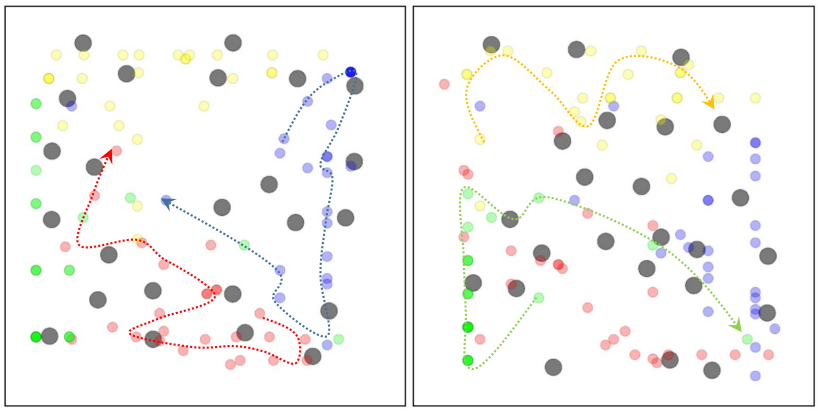


Fig. 5. The trajectory of the UAVs. The black nodes represent the ground users, while the colored nodes represent the UAVs.

The trajectory for different random seed are showed as follows. We can see the UAVs will firstly find the ground users closely, and then they move to serve remotely, and can move around all the region to improve the fairness.

5*: *Complexity Analysis.

The complexity of CAVDN is has positive correlation with the number of parameters of the agent network. Thus, we analyze the complexity of the agent network to discuss the complexity of the algorithm. Let U_l^{enc} denote the number of neurons in the lth layer of the encoding network with L^{enc} layers, where $1 \leq l \leq L^{enc}$. Let U_l^{int} denote the number of neurons in the lth layer of the intention network with L^{int} layers, where $1 \leq l \leq L^{int}$. Let U_l^{comb} denote the number of neurons in the lth layer of the combining network with L^{mix} layers, where $1 \leq l \leq L^{comb}$. Then the total number of parameters of the agent network of our CAVDN is given by $\sum_{l=2}^{L^{enc}-1}(U_{l-1}^{enc}U_l^{enc} + U_l^{enc}U_{l+1}^{enc}) + \sum_{l=2}^{L^{int}-1}(U_{l-1}^{int}U_l^{int} + U_l^{int}U_{l+1}^{int}) + \sum_{l=2}^{L^{comb}-1}(U_{l-1}^{comb}U_l^{comb} + U_l^{comb}U_{l+1}^{comb})$. Thus, the complexity of the algorithm can be expressed as $\mathcal{O}(\sum_{l=2}^{L^{enc}-1}(U_{l-1}^{enc}U_l^{enc} + U_l^{enc}U_{l+1}^{enc}) + \sum_{l=2}^{L^{int}-1}(U_{l-1}^{int}U_l^{int} + U_l^{int}U_{l+1}^{int}) + \sum_{l=2}^{L^{comb}-1}(U_{l-1}^{comb}U_l^{comb} + U_l^{comb}U_{l+1}^{comb}))$.