Additional Experiments for CAVDN

1: *The benfits introduced by intention module.*

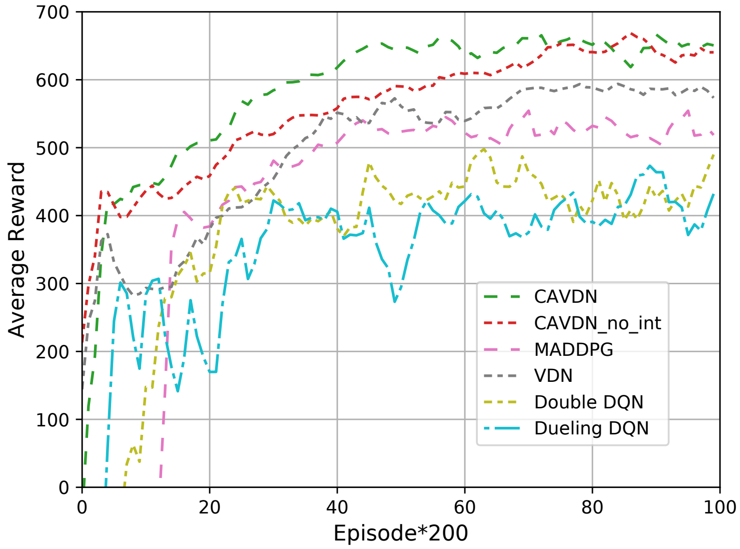


Fig. 1. The performance comparison between the proposed framework and one without the intention module. “CAVDN_no_int” refers to the approach without the intention module.

In our manuscript, the encoding module is used for conveying the local observations for other agents. Yet, it is worth noting that the intention module is not used for conveying any information to other agents, but it is used for extracting the intention of local agents in order to estimate the action value, as shown in Fig. 1 of our manuscript. Specifically, the intention indicates the agent’s behavior without the influence of other agents’ observation, which is then combined with the messages gleaned from other agents to finally obtain the estimated action value. The reason for extracting the intention without the influence of other agents is that the parameters of neural networks are randomly generated at the beginning, while the messages arriving from others may perturb the local decision. This structure helps the agent estimate the action value more accurately and hence accelerate learning.

We added the learning curves of our proposed CAVDN without the intention module (with legend “CAVDN_no_int”). We can observe that compared to CAVDN_no_int, the reward of CAVDN increases more rapidly and converges earlier, which validates the benefit of the intention module.

2: *Comparison with Baselines.*

In Fig. 2, we compare our proposed CAVDN to two more baselines with message encoder, i.e. CommNet proposed in [A] and TarMAC proposed in [B]. Specifically, both CommNet and TarMAC use a fullyconnected neural network (FNN) to encoding the message. The difference lies in the receiver where CommNet directly processes the messages received from other agents by computing their arithmetic mean, while TarMAC exploits the benefit of multi-head attention at the receiver to obtain the attention weight of each message, and then computes their weighted mean. Moreover, inspired by your suggestion, we also evaluate the performance of our proposed CAVDN combined with an attention mechanism (with legend “att_enc+CAVDN_enc”), where the original message encoder is cascaded with an attention encoder. As shown in Fig. 2 of this Response, the performance of CommNet is lower than that of CAVDN and that of TarMAC. This is because CommNet simply uses the arithmetic mean for processing the messages received, which results in information loss. Moreover, our proposed CAVDN performs slightly better than TarMAC because we also use recurrent neural network (RNN) for encoding the messages. By further integrating the attention mechanism into CAVDN, “att_enc+CAVDN_enc” initially improves slightly faster than CAVDN, which implies that an attention mechanism-based encoding may indeed help. Nonetheless, the convergence performance of all approaches except for CommNet is close to each other.

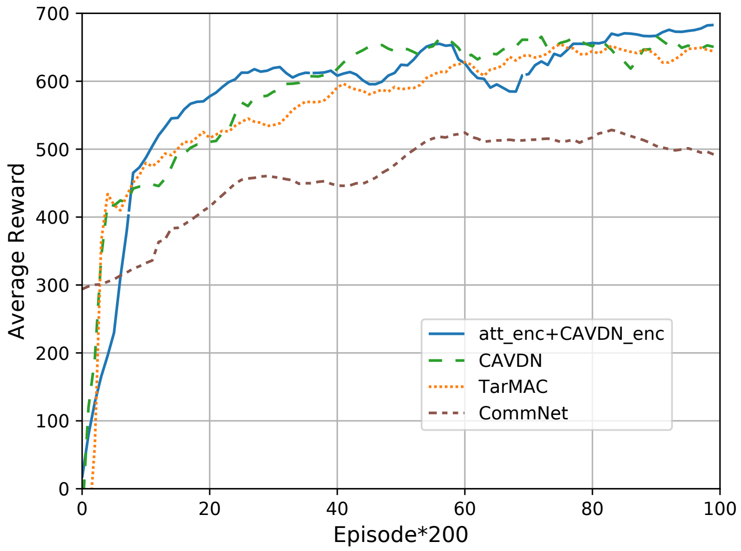


Fig. 2. Performance comparison with existing frameworks including message encoding.

[A] Sainbayar Sukhbaatar, Arthur Szlam, Rob Fergus, “Learning multiagent communication with backpropagation,” in *Proc. NIPS*, 2016.

[B] Abhishek Das, Thophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, Joelle Pineau, TarMAC: Targeted multi-agent communication, in *Proc. ICML*, 2019.

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

Actually, in reinforcement learning, CTCE does not always perform better than CTDE, even without considering the communication overhead in CTCE. For instance, in reference [E], the centralized approach learns an inefficient policy with only a single 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 shown the performance of centralized DDPG (CDDPG) method, which follows the CTCE framework. In CDDPG, the actor network takes the observations of all agents as the input, and outputs the actions of all agents as in [1]. As shown in Fig. 3 of this Response, the performance of CDDPG is close to that of MADDPG, but it is worse than that of CAVDN.

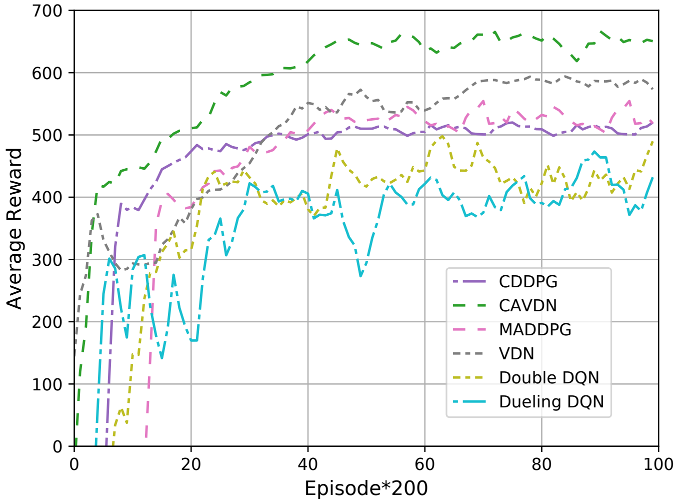
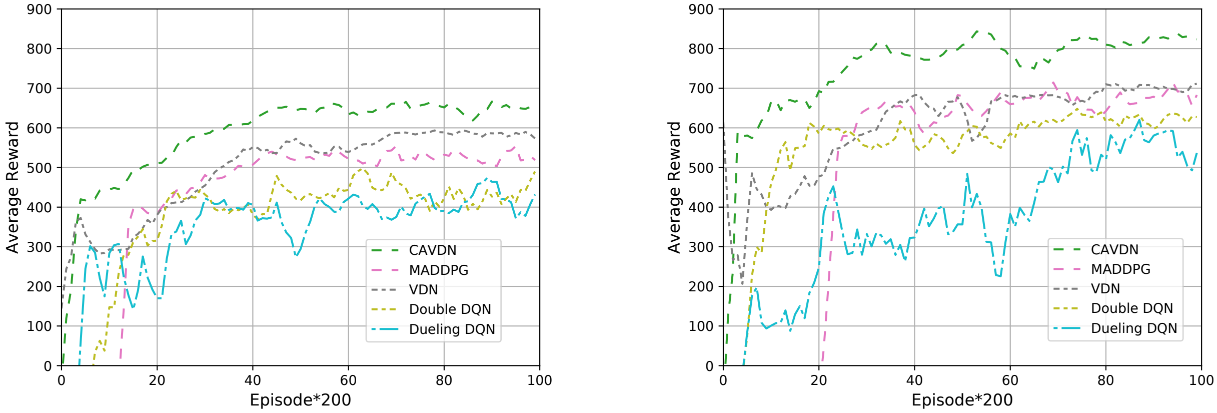


Fig. 3. Performance comparison with the CTCE framework.

[1] P. Sunehag, G. Lever, et. al, “Value-decomposition networks for cooperative multi-agent learning based on team reward,” in *Proc. AAMAS*, 2018.

4: *Experiments with more users.*



(a) The performance comparison with 12 users.

(b) The performance comparison with 24 users.

Fig. 4. Performance comparison with different numbers of users.

In Fig. 4(b), we have doubled the number of users, considering 24 users. We can see from Fig. 4(b) that our proposed CAVDN still performs better than the baseline approaches. Moreover, as you kindly suggest, when the number of UAVs decreases, it can be observed by comparing Fig. 4(b) to Fig. 4(a) that the gain of our proposed CAVDN increases from 8% (with 12 users) to 12% (with 24 users).

5: Trajectory of UAVs.

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

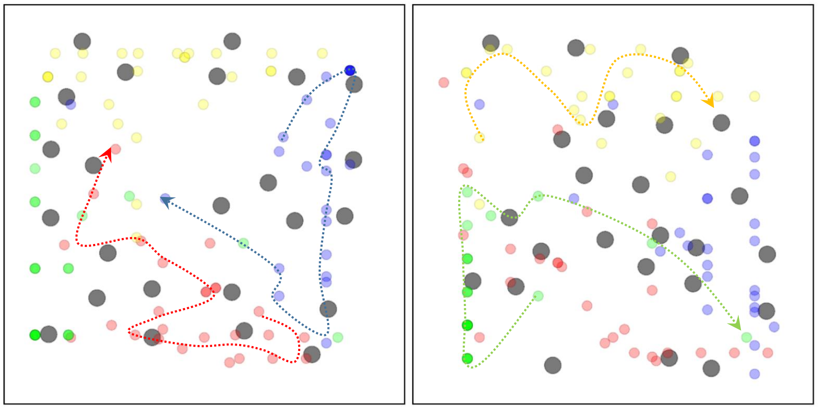


Fig. 5. The trajectory of the UAVs. The black nodes represent the ground users, while the colored nodes represent the UAVs.

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 can be reflected by the number of parameters of the agent network. Therefore, we analyze the number of parameters 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 $\sum_{l=2}^{L^{enc}-1}(U_{l-1}^{enc}U_l^{enc}+U_l^{enc}U_{l+1}^{enc})+\sum_{l=2}^{L^{int}-1}(U_{l-1}^{int}U_l^{int}+U_l^{int}U_{l+1}^{int})+\sum_{l=2}^{L^{comb}-1}(U_{l-1}^{comb}U_l^{comb}+U_l^{comb}U_{l+1}^{comb})$.