Additional Experiments for CAVDN

1: *The benfits introduced by intention module.*

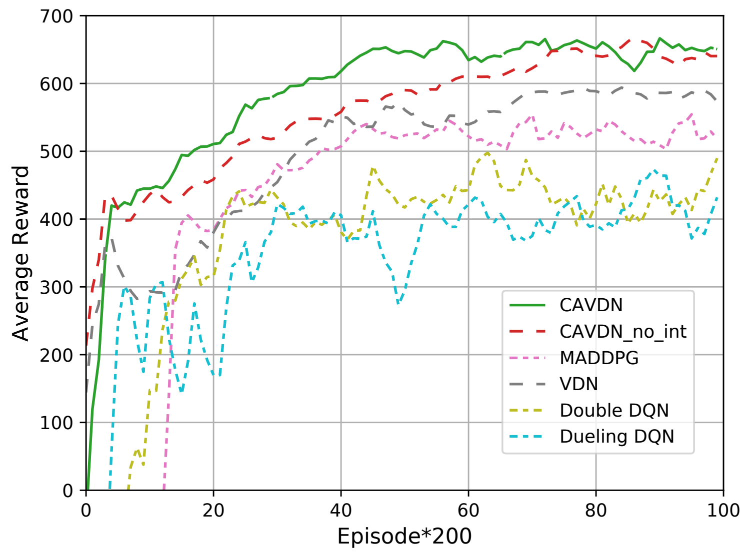


Fig. 1. The performance comparison between the proposed framework and one without the intention module. “CAVDN_no_int” refers to the approach without the intention module.

the role of the intention module is to improve the convergence efficiency and we explain this more clearly in the following. At the beginning of the training phase of CAVDN, since the parameters of neural networks are randomly initialized, the messages gleaned from others (which are generated by randomly initialized neural networks) may not be useful for cooperation, but can be noisy and meaningless, and hence have negative impact on the decision making of each agent. Therefore, in this stage, each agent should make its decisions by relying more on its local observation than on the messages from others. This thereby motivates us to introduce the intention module designed for extracting the information termed as intention $\boldsymbol{h}_n$ solely from the agent's local observation $\boldsymbol{o}_n$ to reflect the agent's own intention without any influence from others. The intention is then incorporated into the combining module and helps the agent estimate the action value more accurately. As seen in Fig. 1, the intention module can improve the convergence efficiency, i.e., the reward of CAVDN increases more rapidly and converges earlier than that of the CAVDN without using the intention module (with legend “CAVDN_no_int”).

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, 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.

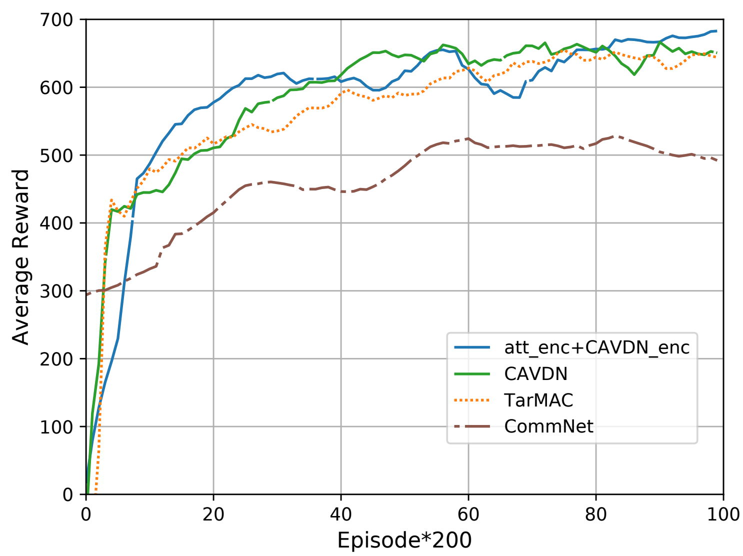


Fig. 2. Performance comparison with existing frameworks including message encoding.

[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.*

Theoretically, the centralized MADRL (CMADRL) architecture takes the observations of all agents as its input, and potentially serves as the upper-bound for the proposed CAVDN.

However, it is notable that the performance of neural networks depends not only on the network structure, but also on the training algorithm. Thus, the upper bound performance can be only achieved when the globally optimal policy is obtained. However, the commonly adopted gradient descent training algorithm of deep learning cannot guarantee to find the globally optimal policy in practice.

Therefore, the performance of the CMADRL is not guaranteed to be better than that of CAVDN. In reference [A], the experiments show that VDN performs better than the CMADRL architecture. As shown in Fig.3, we can also see that both CAVDN and VDN perform better than centralized DDPG.

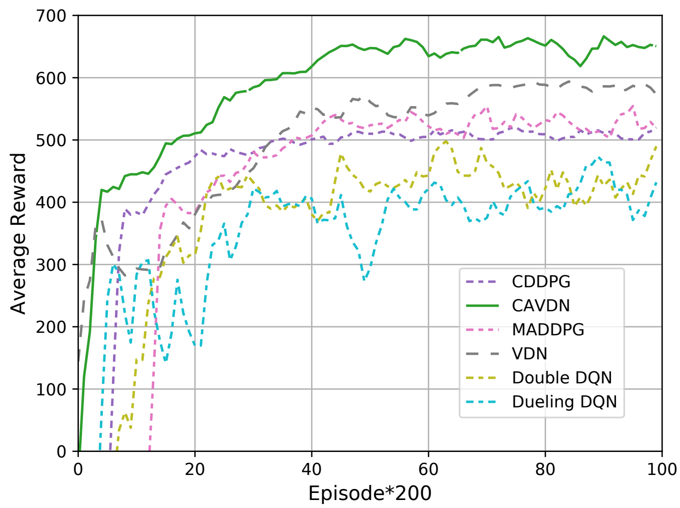


Fig. 3. Performance comparison with the CTCE framework.

To explain these results, we compare the difference between CMADRL and CAVDN in Fig. 4. In the left figure, CMADRL utilizes “one” model that takes the observations of all agents as its input and then outputs the actions of all agents, establishing dense connections among all observations and all neural nodes. By contrast, CAVDN seen at the right of Fig. 4 can be regarded as a special case of CMADRL, which has sparse connections by pruning the connections between observations and neural nodes. This pruning mitigates the redundancy in the structure of neural networks and reduces the number of parameters in their computation graphs. This can improve the performance compared to the original unpruned networks. In reference [B], the authors corroborate the findings that “sparse networks perform better than dense networks for the same parameter count” in the deep reinforcement learning domain.

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

[B] Laura Graesser, Utku Evci, Erich Elsen, Pablo Samuel Castro, “The State of Sparse Training in Deep Reinforcement Learning,” in *Proc. ICML*, 2022.

4: *Experiments with more users.*

In Fig. 5(b), we have doubled the number of users, considering 24 users. We can see from Fig. 5(b) that our proposed CAVDN still performs better than the baseline approaches. Moreover, as you kindly

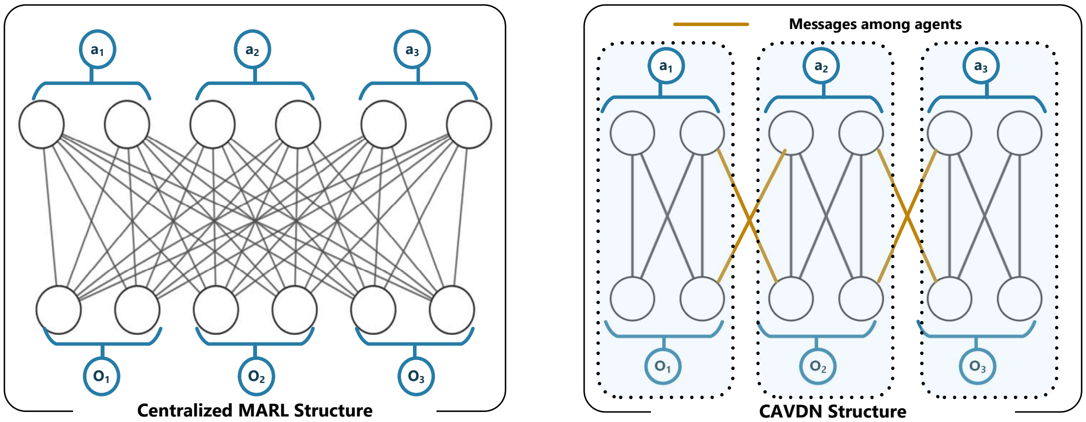
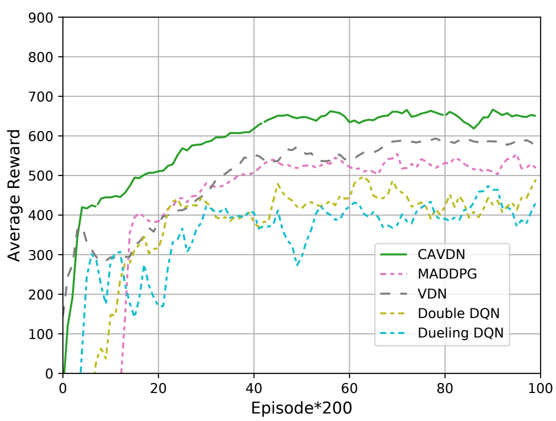
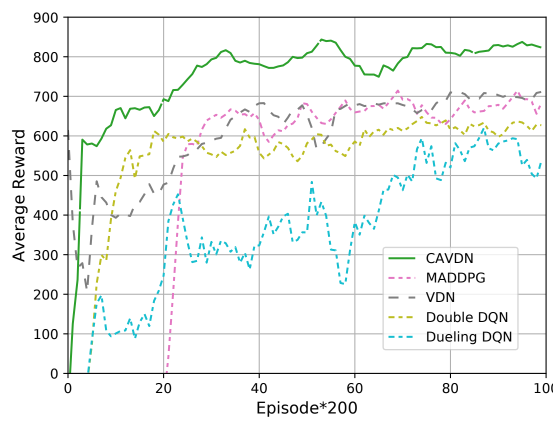


Fig. 4. The comparison between the centralized MARL and CAVDN.



(a) The performance comparison with 12 users.



(b) The performance comparison with 24 users.

Fig. 5. Performance comparison with different numbers of users.

suggest, when the number of UAVs decreases, it can be observed by comparing Fig. 5(b) to Fig. 5(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.

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

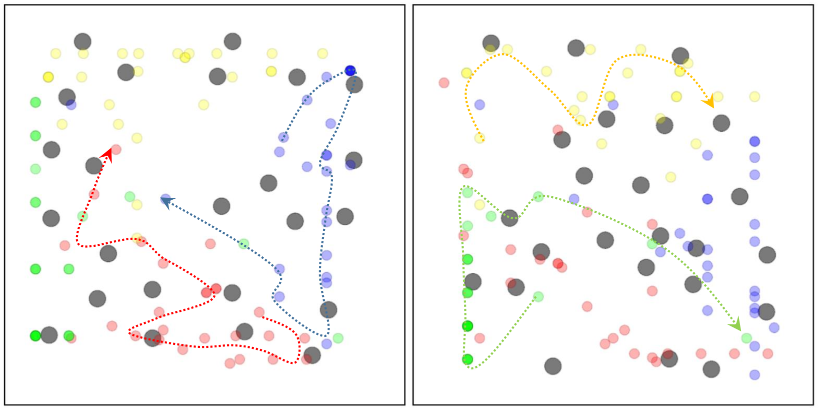


Fig. 6. The trajectory of the UAVs. The black nodes represent the ground users, while the colored nodes represent the UAVs.

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 $\sum_{l=2}^{L^{enc}-1}(U_{l-1}^{enc}U_l^{enc} + U_l^{enc}U_{l+1}^{enc}) + \sum_{l=2}^{L^{int}-1}(U_{l-1}^{int}U_l^{int} + U_l^{int}U_{l+1}^{int}) + \sum_{l=2}^{L^{comb}-1}(U_{l-1}^{comb}U_l^{comb} + U_l^{comb}U_{l+1}^{comb})$.