A. QMIX

A.1. Representational Complexity

The value function class representable with QMIX includes any value function that can be factored into a non-linear monotonic combination of the agents' individual value functions in the fully observable setting.

This follows since the mixing network is a universal function approximator of monotonic functions (Dugas et al., 2009), and hence can represent any value function that factors into a non-linear monotonic combination of the agent's individual value functions. Additionally, we require that the agent's individual value functions order the values of the actions appropriately. By this we mean that Q_a is such that $Q_a(s_t, u^a) > Q_a(s_t, u'^a) \iff Q_{tot}(s_t, (\mathbf{u}^{-a}, u^a)) > Q_{tot}(s_t, (\mathbf{u}^{-a}, u'^a))$, i.e., they can represent a function that respects the ordering of the agent's actions in the joint-action value function. Since the agents' networks are universal function approximators (Pinkus, 1999), they can represent such a Q_a. Hence QMIX is able to represent any value function that factors into a non-linear monotonic combination of the agent's individual value functions.

In a Dec-POMDP, QMIX cannot necessarily represent the value function. This is because each agent's observations are no longer the full state, and thus they might not be able to distinguish the true state given their local observations. If the agent's value function ordering is then wrong, i.e., $Q_a(\tau^a, u) > Q_a(\tau^a, u')$ when $Q_{tot}(s_t, (\mathbf{u}^{-a}, u)) < Q_{tot}(s_t, (\mathbf{u}^{-a}, u'))$, then the mixing network would be unable to correctly represent Q_{tot} given the monotonicity constraints.

QMIX expands upon the linear monotonic value functions that are representable by VDN. Table 3a gives an example of a monotonic value function for the simple case of a two-agent matrix game. Note that VDN is unable to represent this simple monotonic value function.

Agent 1 \ Agent 2	A	B
A	0	1
B	1	8

(a)

Agent 1 \ Agent 2	A	B
A	2	1
B	1	8

(b)

Table 3. (a) An example of a monotonic payoff matrix, (b) a non-monotonic payoff matrix.

However, the constraint in (5) prevents QMIX from representing value functions that do not factorise in such a manner. A simple example of such a value function for a two-agent matrix game is given in Table 3b. Intuitively, any value function for which an agent's best action depends on the actions of the other agents *at the same time step* will not factorise appropriately, and hence cannot be represented perfectly by QMIX.

B. Two Step Game

B.1. Architecture and Training

The architecture of all agent networks is a DQN with a single hidden layer comprised of 64 units with a ReLU nonlinearity. Each agent performs independent ϵ greedy action selection, with $\epsilon = 1$. We set $\gamma = 0.99$. The replay buffer consists of the last 500 episodes, from which we uniformly sample a batch of size 32 for training. The target network is updated every 100 episodes. The learning rate for RMSprop is set to 5×10^{-4}. We train for $10k$ timesteps. The size of the mixing network is 8 units. All agent networks share parameters, thus the agent id is concatenated onto each agent's observations. We do not pass the last action taken to the agent as input. Each agent receives the full state as input.

Each state is one-hot encoded. The starting state for the first timestep is State 1. If Agent 1 takes Action A, it transitions to State 2 (whose payoff matrix is all 7s). If agent 1 takes Action B in the first timestep, it transitions to State 3.

B.2. Learned Value Functions

The learned value functions for the different methods on the Two Step Game are shown in Tables 4 and 5.

(a)

	State 1		State 2A		State 2B	
	A	*B*	*A*	*B*	*A*	*B*
A	6.94	6.94	6.99	7.02	-1.87	2.31
B	6.35	6.36	6.99	7.02	2.33	6.51

(b)

	A	*B*	*A*	*B*	*A*	*B*
A	6.93	6.93	7.00	7.00	0.00	1.00
B	7.92	7.92	7.00	7.00	1.00	8.00

(c)

	A	*B*	*A*	*B*	*A*	*B*
A	6.94	6.93	7.03	7.02	0.00	1.01
B	7.93	7.92	7.02	7.01	1.01	8.02

(d)

	A	*B*	*A*	*B*	*A*	*B*
A	6.98	6.97	7.01	7.02	-1.39	2.57
B	6.37	6.36	7.02	7.04	2.67	6.58

(e)

	A	*B*	*A*	*B*	*A*	*B*
A	6.95	6.99	6.99	7.06	-1.21	2.73
B	6.18	6.22	7.01	7.09	2.46	6.40

Table 4. Q_{tot} on the 2 step game for (a) VDN, (b) QMIX, (c) QMIX-NS, (d) QMIX-Lin and (e) VDN-S

	A	*B*	*A*	*B*	*A*	*B*
Agent 1	6.96	4.47	6.98	7.00	0.50	4.50

	A	*B*	*A*	*B*	*A*	*B*
Agent 2	5.70	5.78	7.00	7.02	0.50	4.47

Table 5. Q_a for IQL on the 2 step game

B.3. Results

Figure 5 shows the loss for the different methods. Table 6 shows the final testing reward for each method.

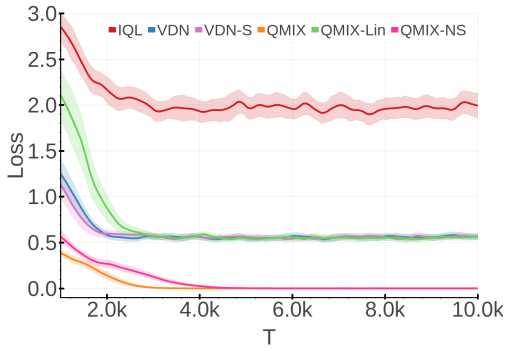


Figure 5. Loss for all six methods on the Two Step Game. The mean and 95% confidence interval is shown across 30 independent runs.

IQL	VDN	VDN-S	QMIX	QMIX-Lin	QMIX-NS
7	7	7	8	7	8

Table 6. The final Test Reward acheived.

C. StarCraft II Setup

C.1. Environment Features

The local observations of individual agents are drawn within their field of view, which encompasses the circular area of the map surrounding units and has a radius equal to the sight range. Each agent receives as input a vector consisting of the following features for all units in its field of view (both allied and enemy): `distance`, `relative x`, `relative y` and `unit_type`.[2]

The global state, which is hidden from agents, is a vector comprised of features of units from the entire map. It does not contain the absolute distances between agents and stores only the coordinates of units relative to the centre of the map. In addition, the global state includes the `health`, `shield` and `cooldown` of all units.[3] In addition, the global state contains the last actions taken by all allied agents. Marines, Stalkers, Zealots, and Colossi have 45, 80, 100, and 200 hit points, respectively. In addition, Stalkers, Zealots, and Colossi have 80, 50, and 150 shield points, respectively. All features, whether in local observations or global state, are normalised by their maximum values. For all unit types, the agent sight range and shooting ranges are set to 9 and 6, respectively.

C.2. Architecture and Training

The architecture of all agent networks is a DRQN with a recurrent layer comprised of a GRU with a 64-dimensional hidden state, with a fully-connected layer before and after. Exploration is performed during training using independent ϵ-greedy action selection, where each agent a performs ϵ-greedy action selection over its own Q_a. Throughout the training, we anneal ϵ linearly from 1.0 to 0.05 over $50k$ time steps and keep it constant for the rest of the learning. We set $\gamma = 0.99$ for all experiments. The replay buffer contains the most recent 5000 episodes. We sample batches of 32 episodes uniformly from the replay buffer and train on fully unrolled episodes. The target networks are updated after every 200 training episodes.

To speed up the learning, we share the parameters of the agent networks across all agents. Because of this, a one-hot encoding of the `agent_id` is concatenated onto each agent's observations. All neural networks are trained using RMSprop[4] with learning rate 5×10^{-4}.

During training and testing, we restrict each episode to have a length of 60 time steps for 3m and 5m maps, 120 time steps for 8m and 2s_3z maps, 150 for 3s_5z and 200 for 1c_3s_5z. If both armies are alive at the end of the episode, we count it as a loss. The episode terminates after one army has been defeated, or the time limit has been reached.

The mixing network consists of a single hidden layer of 32 units, utilising an ELU non-linearity. The hypernetworks are then sized to produce weights of appropriate size. The hypernetwork producing the final bias of the mixing network consists of a single hidden layer of 32 units with a ReLU non-linearity.

D. StarCraft II Results

The results for all six methods and the heuristic-based algorithm on the six maps.

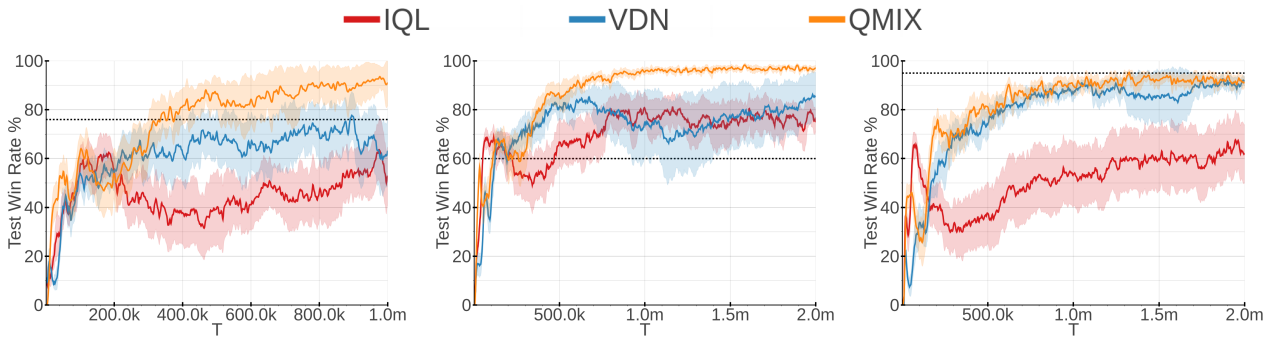
3m	5m	8m	2s_3z	3s_5z	1c_3s_5z
76	60	95	82	45	70

Table 7. The Test Win Rate % of the heuristic-based algorithm on the six maps.

[2] `unit_type` is only included in the 2s_3z, 3s_5z and 1c_3s_5z maps.

[3] A unit's `cooldown` is the time it must wait before firing again. Shields act as additional forms of hit points and are lost first. In contrast to health, shields regenerate over time after absorbing damage.

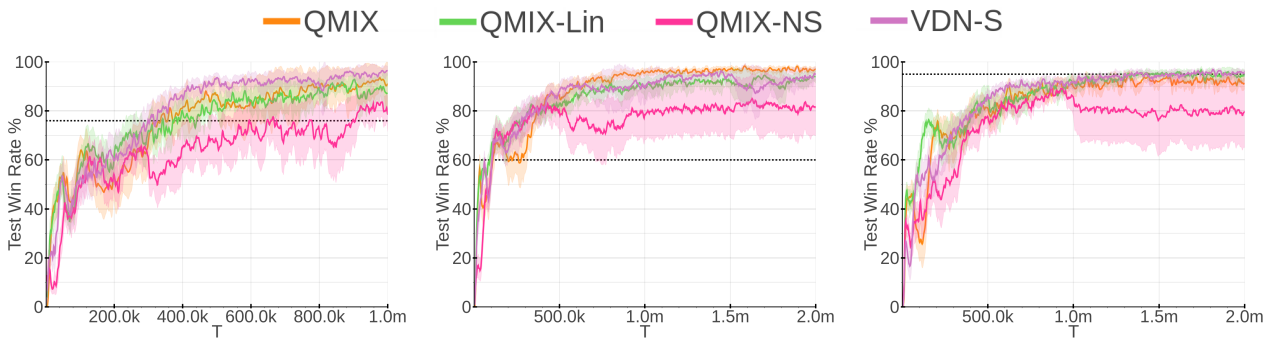
[4] We set $\alpha = 0.99$ and do not use weight decay or momentum.



(a) 3m (b) 5m (c) 8m

(d) 2s_3z (e) 3s_5z (f) 1c_3s_5z

Figure 6. Win rates for IQL, VDN, and QMIX on six different combat maps. The performance of the heuristic-based algorithm is shown as a dashed line.



(a) 3m (b) 5m (c) 8m

(d) 2s_3z (e) 3s_5z (f) 1c_3s_5z

Figure 7. Win rates for QMIX and ablations on six different combat maps. The performance of the heuristic-based algorithm is shown as a dashed line.