Common Concerns. For the performance of IC3Net and TarMAC on cooperative navigation and predator-prey, we were also surprised that they perform poorly, so we changed to optimizing individual rewards (easier settings) to check whether we operated these algorithms wrongly. The results showed they could converge and learned some elementary cooperative strategies. However, they cannot on team reward. Based on these observations, we hypothesize that they are hard to deal with team reward that cannot be decomposed. Both IC3Net and TarMAC are the official implementations. In addition, similar conclusions can be found in Ref.[24], "learning nearly decomposable value functions via communication minimization, ICLR 2020." They found that TarMAC struggles in all the scenarios of StarCraft II (cooperative game) and believed that this is because it cannot deal with the issue of reward assignment.

Reviewer 1. Thanks for your comments. As for difference in predator-prey between ours and the IC3Net paper, our setting has three moving preys, while IC3Net has only one prey and it is *stationary*, which means the task is extremely easy. Predators do not need to learn sophisticated strategy to encircle multiple moving preys. In traffic junction, our main purpose is to investigate the effectiveness of I2C on communication reduction, and thus we built I2C directly on TarMAC. We will benchmark I2C alone and add it up for a more thorough comparison in the final version.

If agents do not have any visibility radius, it means the environment is fully observable including all other agents. In this case, MADDPG can accomplish missions very well and it even converges faster than FC. Communication does not make much difference for no limited visibility because communication with other agents mainly gets their observations. And in fully observable environments, communication is not necessary since observation has already contained most of information from communication. In addition, too much redundant information could impair the learning.

For request-reply mechanism, it sends out a request (scalar) to the agent via a communication channel. Message encoder has two LSTM layers. The size of hidden layers is 128. In this paper, ID vector d is set to its location info that an agent can access from its observation.

Reviewer 2. Thanks for your comments. We have tuned the key hyperparameters of I3CNet and TarMAC for better convergence, but it turns out their poor performance should have nothing to do with the hyperparameters. For the typos, we would correct them in the next version.

Reviewer 3. Thanks for your comments. In traffic junction, our main purpose is to investigate the effectiveness of I2C on communication reduction, and thus we built I2C directly on TarMAC, a communication method.

As for two-phase manner, phase one is to train the prior network with data generated from a pre-trained CTDE algorithm, and phase two is to train the rest of the architecture from scratch, with the prior network fixed. The two-phase manner learns slightly faster than end-to-end manner, but they converge to the same performance which is better than other methods. Another two-phase manner which shows similar results is that phase one is to train all the networks without prior network and use its action-value function to train prior network, and then phase two is to fine-tune the networks with the prior network fixed.

For the parameter sharing, we mean that all agents share their network weights.

Reviewer 4. Thanks for your comments. The literature has shown that excessive communication among agents could impede performance as in Ref [8, 21]. Our experiments also empirically verify this, comparing I2C with FC, IC3Net, and TarMAC. Our thought about the worse performance of excessive communication is that the agent can hardly learn which messages are useful, given excessive received messages. Even with attention like TarMAC, it is still hard.

Multiple rounds of communication per timestep may be required for explicit action coordination. However, as our work considers sharing encoding of observation between agents, it is not necessary to use multiple rounds of communication.

If "broadcast" means broadcast within the field of view, I2C can also achieve this if necessary (the agent can send the request to each of the agents in the field of view). If "broadcast" means broadcast to all other agents, this contradicts to our motivation since it leads to information redundancy that could even impair the performance, even with a budget of broadcast messages.

It is an excellent view for an agent to send the message to whoever it thinks the message is important for that agent. However, an agent cannot access the complete state information of other agents, therefore it is hard for the agent to infer whether it is really important for others. On the other hand, the agent knows exactly what situation it is in, so it is more reasonable to evaluate the others‘ influence on itself.

TarMAC receives messages no matter whether they are useful and uses attention to differentiate messages. I2C employs request-reply mechanism and learns to directly cut off messages they are not helpful. Intuitively, it is easy to learn whether a message is useful in terms of optimizing the RL target (I2C), while it is hard to weight all the received messages together in terms of optimizing the RL target (TarMAC).

We will have another thorough round of proofreading.