SUPPLEMENTARY MATERIAL

Anonymous authors
Paper under double-blind review

1 RELATED WORK

Recent works have extended MARL from small discrete state spaces (Yang & Gu, 2004; Busoniu et al., 2008) to high-dimensional, continuous state spaces (Lowe et al., 2017; Peng et al., 2017). The progresses of deep reinforcement learning give rise to an increasing effort in designing general-purpose deep MARL methods for complex multi-agent environments, including COMA (Foerster et al., 2018), MADDPG (Lowe et al., 2017), MAPPO (Yu et al., 2021) and etc. Currently, CTDE is considered to be the de facto mainstream paradigm in this field (Lowe et al., 2017; Iqbal & Sha, 2019). In terms of specific methods, the Value-Decomposition Network (VDN) (Sunehag et al., 2017) utilizes the factorization of joint-action Q-values as the sum of each agent's utility. QMIX (Rashid et al., 2018) is an extension of VDN which allows the joint action Q-value to be a monotonically increasing combination of each agent's utility, which can vary depending on the global state. There are also other variants proposed to extend the applicability of the value decomposition methods. For instance, QPLEX (Wang et al., 2020a) and QTRAN (Son et al., 2019) aim to learn value functions with complete expressiveness capacity. MAVEN (Mahajan et al., 2019) hybridises value and policy-based methods by introducing a latent space for hierarchical control. This allows MAVEN to achieve committed, temporally extended exploration. Weighted QMIX (Rashid et al., 2020) is based on QMIX and rectifies the suboptimality by introducing weights to place more importance on the better joint actions. UneVEn (Gupta et al., 2021) learns a set of related tasks simultaneously with a linear decomposition of universal successor features. Despite the effectiveness of these methods, they are commonly designed to facilitate the learning of similar policies, it can be detrimental to the acquisition of heterogeneous policies.

To solve the heterogeneous tasks, previous methods choose to add agent-specific information to the observation or assign different roles to learn the distinct policies. PSHA (Terry et al., 2020) proposes an agent indication to enable agents to represent heterogeneous policies. CDS (Li et al., 2021a) uses mutual information to learn an agent ID-specific policy to deal with the problem of learning diversity policies. ROMA (Wang et al., 2020c) proposes a role-oriented MARL framework to make agents specialized in certain tasks. However, these methods, which solely focus on learning distinct policies, often come at the cost of sacrificing the advantages associated with learning in homogeneous scenarios. Furthermore, these methods tend to learn fixed policies that lack the necessary flexibility.

Other methods use a sequential execution policy to represent distinct policies. AR (Fu et al., 2022) proposes a centralized sequential execution policy to solve permutation games. MAiF (Liu et al., 2021) uses a sequential execution policy to learn a path-finding and formation policy for a multi-agent navigation task. These methods can represent the optimal policy in both homogeneous and heterogeneous scenarios. However, a naive sequential execution policy is not guaranteed to converge to optimal policy and has the problem of credit assignment. Additionally, there are also methods such as HAPPO (Kuba et al., 2021) that use sequential policy updates to guarantee monotonic policy improvement of PPO (Schulman et al., 2017). MAT (Wen et al., 2022) adopts sequential policy updates within the structure of a transformer. This design is aimed at executing updates both monotonically and in parallel, thereby enhancing the time efficiency compared to previous methods like HAPPO. SeCA (Zang et al., 2023) constructs a new advantage value to improve upon PG-based methods. Different from focusing on an increment of PG-based methods, our work is proposed to extend the applicability of value decomposition methods to solve the mixing of homogeneous and heterogeneous tasks.

Our work is also related to the credit assignment. Previous methods usually use implicit credit assignment methods to learn the policy, such as VDN and QMIX. However, explicit credit assignment methods have also been proposed. For instance, COMA (Foerster et al., 2018) utilizes a counterfactual advantage to learn the value function. Other methods use Shapley Value (Shapley, 2016) as the

credit value of each agent. Shapley Value originates from cooperative game theory and is able to distribute benefits reasonably by estimating the contribution of participating agents. In these methods, SQDDPG (Wang et al., 2020b) and Shapley (Li et al., 2021b) use Shapley Value to estimate the complex interactions between agents. However, these methods can only get approximated Shapley value as calculating the Shapley value involves exponential time complexity (Wang et al., 2020b) and they are not designed to learn similar and distinct policies simultaneously. In this work, we introduce an explicit credit assignment method using marginal contribution in Shapley value to learn a sequential execution policy that can represent the optimal policy in scenarios with a mixing of homogeneous and heterogeneous tasks.

2 SCENARIOS SETTINGS AND TRAINING DETAILS

In the Multi-XOR games, agents receive two types of rewards, as illustrated in Table 1 and 2. Table 1 displays the homogeneous reward, which exhibits a non-monotonic payoff. This poses a challenge of relative overgeneralization for the learning process. Table 2 presents the heterogeneous reward, where agents are required to take distinct actions. Specifically, if two agents choose the joint actions $C\&C$ to solve the task, the other two agents must choose $L\&L$; otherwise, a penalty will be imposed. However, if all agents select $L\&L$, the return will be zero.

In MAgent, each agent corresponds to one grid and has a local observation that contains a square view centered at the agent and a feature vector including coordinates, health point (HP) and ID of agents nearby, and the agent's last action. The discrete actions are moving, staying, and attacking. The global state of MAgent is a mini-map (6×6) of the global information. The opponent's policies used in experiments are randomly escaping policy in *pursuit*. We choose five different scenarios *lift*, *heterogeneous_lift*, *multi_target_lift*, *pursuit* and *bridge*. There are detailed settings of these scenarios, as shown in Table 3. We demonstrate the payoff matrix by showing the R as the reward returned when cooperation is achieved, P_{ho} as are penalty when taking cooperative action but failing to achieve cooperation in homogeneous scenarios, and P_{he} as the penalty for taking the same action in heterogeneous scenarios.

	C	L
C	+0.5	-0.3
L	-0.3	0

Table 1: Homogeneous payoff matrix of the Multi-XOR game.

	C&C	L&L
C&C	-10	+0.5
L&L	+0.5	0

Table 2: Heterogeneous payoff matrix of the Multi-XOR game.

	Lift	HeterogeneousLift	MultiTargetLift	Pursuit	Bridge
Agent number	3	4	4	4	4
Object number	3	1	2	1	0
Map size	6×6	15×15	12×12	10×10	11×11
Payoff	R=1,P_{ho}=0,-0.3,P_{he}=0	R=1,P_{ho}=0,P_{he}=-100	R=0.25,0.5,P_{ho}=-0.2,P_{he}=-20,-40	R=0.5,P_{ho}=-0.1,P_{he}=-1	R=0.5,P_{ho}=-0.03,P_{he}=0

Table 3: Settings of MAgent Scenarios. R is the reward, P_{ho} is the penalty of mis-coordination of homogeneous behavior, P_{he} is the penalty of mis-coordination of heterogeneous behavior.

In the Overcooked environment, the objective is to perform a series of tasks involving onions, dishes, and soups. The agents are required to place 2 onions in a pot, let them cook for 5 timesteps, transfer the resulting soup into a dish, and finally serve it, which rewards all players with a score of 20. There are six possible actions available to the agents: up, down, left, right, noop (no operation), and interact. Notably, the action of picking up onions requires two agents to simultaneously take the "interact" action, otherwise a penalty of -0.1 is incurred. On the other hand, actions such as putting onions into the pots can be performed by a single agent. To evaluate the difficulty level of different scenarios, we have designed three maps with varying levels of complexity. The easy map consists of more onions

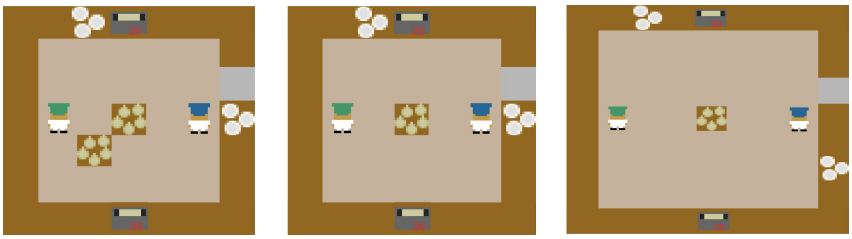

Figure 1: The images of the map of Overcooked tasks. From left to right is easy, medium and hard.

and a smaller map size (5×5), making it relatively easier to solve. The medium map, on the other hand, contains a single onion and a smaller map size (5×5). Finally, the hard map poses a greater challenge with its larger map size (7×7) and a single onion, making exploration more demanding for the agents.

We set the discount factor as 0.99 and use the RMSprop optimizer with a learning rate of 5e-4 for policy and 1e-3 for the critic. The ϵ-greedy is used for exploration with ϵ annealed linearly from 1.0 to 0.05 in 700k steps. The batch size is 4 and updating the target network every 200 episodes. The length of each episode in MAgent is limited to 100 steps in bridge and 50 for others, except for Multi-XOR which is a single-step game. The sample number M of our method is 5 in all scenarios. We run all the experiments five times with different random seeds and plot the mean/std in all the figures. All experiments are carried out on the same computer, equipped with Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz, 64GB RAM and an NVIDIA RTX3090. The system is Ubuntu 18.04 and the framework is PyTorch.

3 Details of Model Implementation and Hyperparameters

The network of all compared methods uses the same LSTM network, consisting of a recurrent layer comprised of a GRU with a 64-dimensional hidden state, with one fully-connected layer before and two after. All mixing networks use a fully-connected layer with 32-dimensional hidden state. The network of our critic and policy uses two fully-connected layers with 64-dimensional hidden state and one fully-connected layers with 32-dimensional hidden state after.

4 Proof of the Value Decomposition of Critic

First of all, according to the decentralized execution setting, there exists a reward decomposition,

$$r_{tot}(s,u) = \sum_{i=1}^{n} r_c^i(o_i,u) = \sum_{i=1}^{n} r_c^i(o_i,u_i^-,a_i). \tag{1}$$

This is because if the task can be solved by decentralized execution, the observation of each agent must contain all the necessary information to identify the goals. Otherwise, agents will require density communication to receive information about others' observations to identify the goals, which is not the setting that we discussed in our works. Then, we define the value decomposition Q_c^i which models each agent's individual utility. From Eq. (1), we have

$$Q_{tot}(s,u) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_{tot}\left(s,u\right) \mid \pi\right] = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \sum_{i=1}^{n} r_{team}^i(o_i,u_i^-,a_i) \mid \pi\right] = \sum_{i=1}^{n} Q_c^i(s,u). \tag{2}$$

In addition, we have

$$\arg\max_{a_i}(Q_{tot}(s,u)) = \arg\max_{a_i}(Q_c^i(s,u)) = \arg\max_{a_i}(Q_c^i(\tau_i,u_i^-,a_i)). \tag{3}$$

The first part is because the value of a_i is represented by item Q_c^i and the reason for the second part is that Q_c^i is only related to agent i as well as the actions of potential cooperative agents and all the necessary information is contained in (τ_i,u_i^-,a_i), so we can get the unbiased estimated value of Q_c^i given (τ_i,u_i^-,a_i). Therefore, from Eq. (2) and Eq. (3) we have

$$\arg\max_{u}(Q_{tot}(s,u)) = \{\arg\max_{a_1}(Q_c^1(\tau_1,u_1^-,a_1)),...,\arg\max_{a_n}(Q_c^n(\tau_n,u_n^-,a_n))\}. \tag{4}$$

	A	B
A	$+r_1$	$-r_2$
B	$-r_2$	0

Table 4: Example of a homogeneous payoff matrix.

An intuitive understanding of Eq. (4) is that each agent takes action based on the perception of other potential cooperative agents' actions, so they can take the corresponding cooperative action and the joint action is the optimal cooperative joint action.

5 Limitations of Individual Utility

5.1 Homogeneous Scenarios

First, for the homogeneous task, we have the payoff matrix in Table 4. Since, we indicate that the individual utility $Q_i(\tau_i, a_i)$, should be viewed as a variable sampled from distribution $Q_c^i(\tau_i, u_i^-, a_i)$. Following this conclusion, we have the loss of $Q_i(\tau_i, a_i)$ should be

$$\mathcal{L}_i = \sum_{k=1}^{K_i} p_k \cdot (\hat{Q}_c^i(\tau_i, u_i^{k-}, a_i) - Q_i(\tau_i, a_i))^2. \tag{5}$$

where $\hat{Q}_c^i$ means the ground true value function, u_i^{k-} means one of the combination of u_i^- and p_k is the possibility of u_i^{k-} occurred. Therefore, $Q_i(\tau_i, a_i)$ learns to the converged value by optimizing L_i, we have the converged $\hat{Q}_i(\tau_i, a_i)$ when L_i is minimized,

$$\hat{Q}_i(\tau_i, a_i) = \sum_{k=1}^{K_i} p_k \cdot \hat{Q}_c^i(\tau_i, u_i^{k-}, a_i). \tag{6}$$

For a simple demonstration, we take the example payoff into Eq. (6). The value of cooperative action a_i^* is

$$\hat{Q}_i(\tau_i, a_i^*) = p_a \cdot \hat{Q}_c^i(\tau_i, u_i^{-*}, a_i^*) + p_b \cdot \hat{Q}_c^i(\tau_i, u_i^-, a_i^*). \tag{7}$$

where p_a means the possibility of other agent taking cooperative actions u_i^{a-*} ($u_i^{a-*} = A$) and p_b means the possibility of other agents taking the other actions u_i^{b-} ($u_i^{b-} = B$). Additionally, we have

$$p_a + p_b = 1 \tag{8}$$

Similarly, we have the value of lazy action a_i^- as

$$\hat{Q}_i(\tau_i, a_i^-) = p_a \cdot \hat{Q}_c^i(\tau_i, u_i^{-*}, a_i^-) + p_b \cdot \hat{Q}_c^i(\tau_i, u_i^-, a_i^-). \tag{9}$$

We know the policy represented by $Q_i(\tau_i, a_i)$ fails when $\hat{Q}_i(\tau_i, a_i^-)$ is larger than $\hat{Q}_i(\tau_i, a_i^*)$, which is

$$\begin{aligned} \hat{Q}_i(\tau_i, a_i^-) - \hat{Q}_i(\tau_i, a_i^*) = p_a \cdot (\hat{Q}_c^i(\tau_i, u_i^{-*}, a_i^-) - Q_c^i(\tau_i, u_i^{-*}, a_i^*)) \\ + p_b \cdot (\hat{Q}_c^i(\tau_i, u_i^-, a_i^-) - Q_c^i(\tau_i, u_i^-, a_i^*)) > 0 \end{aligned} \tag{10}$$

We take the $+r_1$ and $-r_2$ into Eq. (10),

$$\hat{Q}_i(\tau_i, a_i^-) - \hat{Q}_i(\tau_i, a_i^*) = p_a \cdot (-r_2 - r_1) + p_b \cdot (0 - (-r_2)) = (p_b - 1) \cdot (r_2 + r_1) + p_b \cdot r_2 > 0 \tag{11}$$

This means the policy represented by $Q_i(\tau_i, a_i)$ will fail when

$$r_1 \cdot (1 - p_b) < (2p_b - 1) \cdot r_2. \tag{12}$$

which equals to

$$\frac{r_1}{r_2} < \frac{2p_b - 1}{1 - p_b}. \tag{13}$$

5.2 Heterogeneous Scenarios

First, for the heterogeneous task, we have the payoff matrix in Table 5. Similarly, agents with the policy represented by $Q_i(\tau_i, a_i)$ fails when $\hat{Q}_i(\tau_i, a_i^-)$ is larger than $\hat{Q}_i(\tau_i, a_i^*)$ in the heterogeneous

	A	B
A	$-r_2$	$+r_1$
B	$+r_1$	$-r_2$

Table 5: Example of a heterogeneous payoff matrix.

scenario. However, there are multiple optimal joint actions (1=A,2=B), (1=B,2=A), which are different from the homogeneous scenarios. We first consider the (1=A,2=B) situation which is

$$\begin{aligned}\hat{Q}_i(\tau_i, a_i^-) - \hat{Q}_i(\tau_i, a_i^*) = p_b \cdot (\hat{Q}_c^i(\tau_i, u_i^{-*}, a_i^-) - Q_c^i(\tau_i, u_i^{-*}, a_i^*)) \\ +p_a \cdot (\hat{Q}_c^i(\tau_i, u_i^-, a_i^-) - Q_c^i(\tau_i, u_i^-, a_i^*)) > 0\end{aligned} \tag{14}$$

Taking the the $+r_1$ and $-r_2$ into Eq. (14),

$$\begin{aligned}\hat{Q}_i(\tau_i, a_i^-) - \hat{Q}_i(\tau_i, a_i^*) = p_b \cdot (-r_2 - r_1) + p_a \cdot (r_1 - (-r_2)) \\ = -p_b \cdot (r_2 + r_1) + (1 - p_b) \cdot (r_2 + r_1) > 0\end{aligned} \tag{15}$$

which equals to

$$1 - 2p_b > 0 \tag{16}$$

$$p_b < \tfrac{1}{2}. \tag{17}$$

For situation (1=B,2=A), we have a similar conclusion,

$$p_a < \tfrac{1}{2}. \tag{18}$$

We notice that the overall possibility of failure is

$$P_f = P(p_a < \tfrac{1}{2}) + P(p_b < \tfrac{1}{2}) = P((1 - p_b) < \tfrac{1}{2}) + P(p_b < \tfrac{1}{2}) = P(\tfrac{1}{2} < p_b) + P(p_b < \tfrac{1}{2}) = 1. \tag{19}$$

Therefore, the policy represented by $Q_i(\tau_i, a_i)$ can never promise to solve the heterogeneous task. Furthermore, we can calculate the possibility of reaching cooperation,

$$P_c = P((1 = B, 2 = A)) + P((1 = A, 2 = B)) = p_b \cdot p_a + p_a \cdot p_b = 2 \cdot p_b \cdot (1 - p_b). \tag{20}$$

The maximization of Eq. (20) is 0.5 when $p_b = p_a = 0.5$. The result demonstrated that decreasing p_b when $p_b < 0.5$ causes cooperation more difficult to be reached.

6 Analysis of Individual Utility of Sequential Execution Policy

We have the IGM principal as

$$\arg\max_u(Q_{tot}(s, u)) = \{\arg\max_{a_1}(Q_1(\tau_1, a_1)), ..., \arg\max_{a_n}(Q_n(\tau_n, a_n))\}. \tag{21}$$

For sequential execution method, the policy is the form of

$$u = \{\arg\max_{a_1}(Q_s^i(\tau_1, a_1)), ..., \arg\max_{a_n}(Q_s^i(\tau_i, a_{1:n-1}, a_n))\}. \tag{22}$$

We take the payoff matrix in Table 5 as an example, there are multiple optimal joint actions (1=A,2=B), (1=B,2=A), we take the optimal actions into Eq. (22),

$$\begin{aligned}(1 = A, 2 = B) = \{\arg\max_{a_1}(Q_s^i(\tau_1, a_1)), \arg\max_{a_n}(Q_s^i(\tau_i, A, a_n))\} \\ (1 = B, 2 = A) = \{\arg\max_{a_1}(Q_s^i(\tau_1, a_1)), \arg\max_{a_n}(Q_s^i(\tau_i, B, a_n))\}.\end{aligned} \tag{23}$$

We notice that although the latter utility has the correct maximization of the utility, the former one has a conflict maximum result as it lacks the necessary information about other agents' actions.

REFERENCES

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. Revisiting some common practices in cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 6863–6877. PMLR, 2022.

Tarun Gupta, Anuj Mahajan, Bei Peng, Wendelin Böhmer, and Shimon Whiteson. Uneven: Universal value exploration for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 3930–3941. PMLR, 2021.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 2961–2970. PMLR, 2019.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021a.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, 2021b.

Shanqi Liu, Licheng Wen, Jinhao Cui, Xuemeng Yang, Junjie Cao, and Yong Liu. Moving forward in formation: A decentralized hierarchical learning approach to multi-agent moving together. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4777–4784. IEEE, 2021.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32, 2019.

Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Lloyd S Shapley. *17. A value for n-person games*. Princeton University Press, 2016.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos, and Benjamin Black. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2020a.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: a local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7285–7292, 2020b.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning*, 2020c.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

Erfu Yang and Dongbing Gu. Multiagent reinforcement learning for multi-robot systems: A survey. Technical report, tech. rep, 2004.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, and Junliang Xing. Sequential cooperative multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 485–493, 2023.