ROMA: Multi-Agent Reinforcement Learning with Emergent Roles

Tonghan Wang [1] **Heng Dong** [1] **Victor Lesser** [2] **Chongjie Zhang** [1]

A. Mathematical Derivation

A.1. Identifiable Roles

For learning identifiable roles, we propose to maximize the conditional mutual information objective between roles and local observation-action histories given the current observations. In Sec. 3.1 of the paper, we introduce a posterior estimator and derive a tractable lower bound of the mutual information term:

$$
\begin{aligned}
I(\rho_i^t; \tau_i^{t-1} | o_i^t) &= \mathbb{E}_{\rho_i^t, \tau_i^{t-1}, o_i^t}\left[\log \frac{p(\rho_i^t | \tau_i^{t-1}, o_i^t)}{p(\rho_i^t | o_i^t)}\right] \\
&= \mathbb{E}_{\rho_i^t, \tau_i^{t-1}, o_i^t}\left[\log \frac{q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t)}{p(\rho_i^t | o_i^t)}\right] \\
&\quad + \mathbb{E}_{\tau_i^{t-1}, o_i^t}\left[D_{\mathrm{KL}}(p(\rho_i^t | \tau_i^{t-1}, o_i^t) \| q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t))\right] \\
&\geq \mathbb{E}_{\rho_i^t, \tau_i^{t-1}, o_i^t}\left[\log \frac{q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t)}{p(\rho_i^t | o_i^t)}\right],
\end{aligned} \tag{1}
$$

where the last inequality holds because of the non-negativity of the KL divergence. Then it follows that:

$$
\begin{aligned}
&\mathbb{E}_{\rho_i^t, \tau_i^{t-1}, o_i^t}\left[\log \frac{q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t)}{p(\rho_i^t | o_i^t)}\right] \\
=&\mathbb{E}_{\rho_i^t, \tau_i^{t-1}, o_i^t}\left[\log q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t)\right] - \mathbb{E}_{\rho_i^t, o_i^t}\left[\log p(\rho_i^t | o_i^t)\right] \\
=&\mathbb{E}_{\rho_i^t, \tau_i^{t-1}, o_i^t}\left[\log q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t)\right] + \mathbb{E}_{o_i^t}\left[H(\rho_i^t | o_i^t)\right] \\
=&\mathbb{E}_{\tau_i^{t-1}, o_i^t}\left[\int p(\rho_i^t | \tau_i^{t-1}, o_i^t) \log q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t) d\rho_i^t\right] + \mathbb{E}_{o_i^t}\left[H(\rho_i^t | o_i^t)\right]
\end{aligned} \tag{2}
$$

The role encoder is conditioned on the local observations, so given the observations, the distributions of roles, $p(\rho_i^t)$, are independent from the local histories. Thus, we have

$$
I(\rho_i^t; \tau_i^{t-1} | o_i^t) \geq -\mathbb{E}_{\tau_i^{t-1}, o_i^t}\big[\mathcal{CE}[p(\rho_i^t | o_i^t) \| q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t)\big] + \mathbb{E}_{o_i^t}\big[H(\rho_i^t | o_i^t)\big] \tag{3}
$$

In practice, we use a replay buffer $\mathcal{D}$ and minimize

$$
\mathcal{L}_I(\theta_\rho, \xi) = \mathbb{E}_{(\tau_i^{t-1}, o_i^t) \sim \mathcal{D}}\big[\mathcal{CE}[p(\rho_i^t | o_i^t) \| q_\xi(\rho_i^t | \tau_i^{t-1}, o_i^t) - H(\rho_i^t | o_i^t)\big]\,. \tag{4}
$$

A.2. Specialized Roles

Conditioning roles on local observations enables roles to be dynamic, and optimizing $\mathcal{L}_I$ enables roles to be identifiable by agents' long-term behaviors, but these formulations do not explicitly encourage specialized roles. To make up for this shortcoming, we propose a role differentiation objective in Sec. 3.2 of the paper, where a mutual information maximization

[1] IIIS, Tsinghua University, Beijing, China [2] University of Massachusetts, Amherst, USA. Correspondence to: Tonghan Wang <tonghanwang1996@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

objective is involved (maximizing $I(\rho_i^t; \tau_j^{t-1}|o_j^t)$). Here, we derive a variational lower bound of this mutual information objective to render it feasible to be optimized.

$$\begin{aligned} I(\rho_i^t; \tau_j^{t-1}|o_j^t) &= \mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \left[\log \frac{p(\rho_i^t, \tau_j^{t-1}|o_j^t)}{p(\rho_i^t|o_j^t) p(\tau_j^{t-1}|o_j^t)} \right] \\ &= \mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \left[\log \frac{p(\rho_i^t|\tau_j^{t-1}, o_j^t)}{p(\rho_i^t|o_j^t)} \right] \\ &= \mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \;\; \log p(\rho_i^t|\tau_j^{t-1}, o_j^t) \;\; + \mathbb{E}_{o_j^t} \;\; H(\rho_i^t|o_j^t) \\ &\geq \mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \;\; \log p(\rho_i^t|\tau_j^{t-1}, o_j^t) \;\; . \end{aligned} \tag{5}$$

We clip the variances of role distributions at a small value (0.1) to ensure that the entropies of role distributions are always non-negative so that the last inequality holds. Then, it follows that:

$$\begin{aligned} &\mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \;\; \log p(\rho_i^t|\tau_j^{t-1}, o_j^t) \\ =&\mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \;\; \log q_\xi(\rho_i^t|\tau_j^{t-1}, o_j^t) \;\; + \mathbb{E}_{\tau_j^{t-1}, o_j^t} \;\; D_{\mathrm{KL}} \;\; p(\rho_i^t|\tau_j^{t-1}, o_j^t) \;\; q_\xi(\rho_i^t|\tau_j^{t-1}, o_j^t) \\ \geq&\mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \;\; \log q_\xi(\rho_i^t|\tau_j^{t-1}, o_j^t) \;\; , \end{aligned} \tag{6}$$

where q_ξ is the trajectory encoder introduced in Sec. A.1, and the KL divergence term can be left out when deriving the lower bound because it is non-negative. Therefore, we have:

$$I(\rho_i^t; \tau_j^{t-1}|o_j^t) \geq \mathbb{E}_{\rho_i^t, \tau_j^{t-1}, o_j^t} \;\; \log q_\xi(\rho_i^t|\tau_j^{t-1}, o_j^t) \;\; . \tag{7}$$

Recall that, in order to learn specialized roles, we propose to minimize:

$$D_\phi^t \;\; {}_F - \sum_{i \neq j} \min\{I(\rho_i^t; \tau_j^{t-1}|o_j^t) + d_\phi(\tau_i^{t-1}, \tau_j^{t-1}), U\}, \tag{8}$$

where $D_\phi^t = (d_{ij}^t)$, and $d_{ij}^t = d_\phi(\tau_i^{t-1}, \tau_j^{t-1})$ is the estimated dissimilarity between trajectories of agent i and j. For the term $\min\{I(\rho_i^t; \tau_j^{t-1}|o_j^t) + d_\phi(\tau_i^{t-1}, \tau_j^{t-1}), U\}$, we have:

$$\begin{aligned} &\min\{I(\rho_i^t; \tau_j^{t-1}|o_j^t) + d_\phi(\tau_i^{t-1}, \tau_j^{t-1}), U\} \\ =&\min\{\mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} \left[\log \frac{p(\rho_i^t, \tau_j^{t-1}|o_j^t)}{p(\rho_i^t|o_j^t) p(\tau_j^{t-1}|o_j^t)} + d_\phi(\tau_i^{t-1}, \tau_j^{t-1}) \right], \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [U]\}, \end{aligned} \tag{9}$$

where $\boldsymbol{\tau}^{t-1}$ is the joint trajectory, $\boldsymbol{o}^t$ is the joint observation, and $\boldsymbol{\rho}^t = \langle \rho_1^t, \rho_2^t, \cdots, \rho_n^t \rangle$. We denote

$$\begin{aligned} T_1 &\equiv \log \frac{p(\rho_i^t, \tau_j^{t-1}|o_j^t)}{p(\rho_i^t|o_j^t) p(\tau_j^{t-1}|o_j^t)}, \\ T_2 &\equiv \log q_\xi(\rho_i^t|\tau_j^{t-1}, o_j^t). \end{aligned} \tag{10}$$

Because

$$\begin{aligned} T_2 &\geq \min\{T_2, U\}, \\ U &\geq \min\{T_2, U\}, \end{aligned} \tag{11}$$

it follows that:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [T_2] &\geq \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [\min\{T_2, U\}], \\ \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [U] &\geq \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [\min\{T_2, U\}]. \end{aligned} \tag{12}$$

So that

$$\begin{aligned} &\min\{\mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [T_1], \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [U]\} \\ \geq&\min\{\mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [T_2], \mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [U]\} \quad \{\text{Eq. 7}\} \\ \geq&\mathbb{E}_{\boldsymbol{\tau}^{t-1}, \boldsymbol{o}^t, \boldsymbol{\rho}^t} [\min\{T_2, U\}] \quad \{\text{Eq. 12}\}, \end{aligned} \tag{13}$$

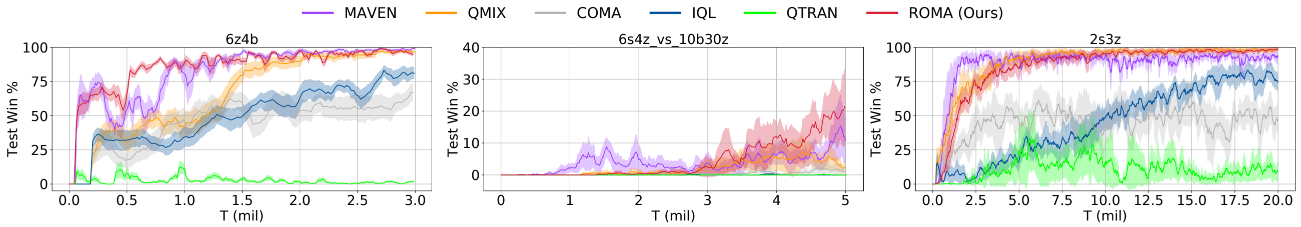


Figure 1. Additional results on the SMAC benchmark.

which means that Eq. 8 satisfies:

$$
\begin{aligned}
& D^t_{\phi} \quad _F - \underset{i\neq j}{\quad} \min\{I(\rho_i^t;\tau_j^{t-1}|o_j^t) + d_\phi(\tau_i^{t-1},\tau_j^{t-1}), U\} \\
=& \mathbb{E}_{\boldsymbol{\tau}^{t-1},\boldsymbol{o}^t,\boldsymbol{\rho}^t} \quad D^t_{\phi} \quad _F - \underset{i\neq j}{\quad} \min\{\mathbb{E}_{\boldsymbol{\tau}^{t-1},\boldsymbol{o}^t,\boldsymbol{\rho}^t} \quad T_1 + d_\phi(\tau_i^{t-1},\tau_j^{t-1}) \quad , U\} \\
\leq& \mathbb{E}_{\boldsymbol{\tau}^{t-1},\boldsymbol{o}^t,\boldsymbol{\rho}^t} \quad D^t_{\phi} \quad _F - \underset{i\neq j}{\quad} \mathbb{E}_{\boldsymbol{\tau}^{t-1},\boldsymbol{o}^t,\boldsymbol{\rho}^t} \quad \min\{T_2 + d_\phi(\tau_i^{t-1},\tau_j^{t-1}), U\} \qquad \{\text{Eq. } 13\} \\
=& \mathbb{E}_{\boldsymbol{\tau}^{t-1},\boldsymbol{o}^t,\boldsymbol{\rho}^t} \left[\quad D^t_{\phi} \quad _F - \underset{i\neq j}{\quad} \min\{T_2 + d_\phi(\tau_i^{t-1},\tau_j^{t-1}), U\} \right].
\end{aligned} \tag{14}
$$

We minimize this upper bound to optimize Eq. 8. In practice, we use a replay buffer, and minimize:

$$
\mathcal{L}_D(\theta_\rho,\phi,\xi) = \mathbb{E}_{(\boldsymbol{\tau}^{t-1},\boldsymbol{o}^t)\sim\mathcal{D},\boldsymbol{\rho}^t\sim p(\boldsymbol{\rho}^t|\boldsymbol{o}^t)} \quad D^t_{\phi} \quad _F - \underset{i\neq j}{\quad} \min\{q_\xi(\rho_i^t|\tau_j^{t-1},o_j^t) + d_\phi(\tau_i^{t-1},\tau_j^{t-1}), U\} \quad , \tag{15}
$$

where $\mathcal{D}$ is the replay buffer, $\boldsymbol{\tau}^{t-1}$ is the joint trajectory, $\boldsymbol{o}^t$ is the joint observation, and $\boldsymbol{\rho}^t = \langle \rho_1^t, \rho_2^t, \cdots, \rho_n^t \rangle$.

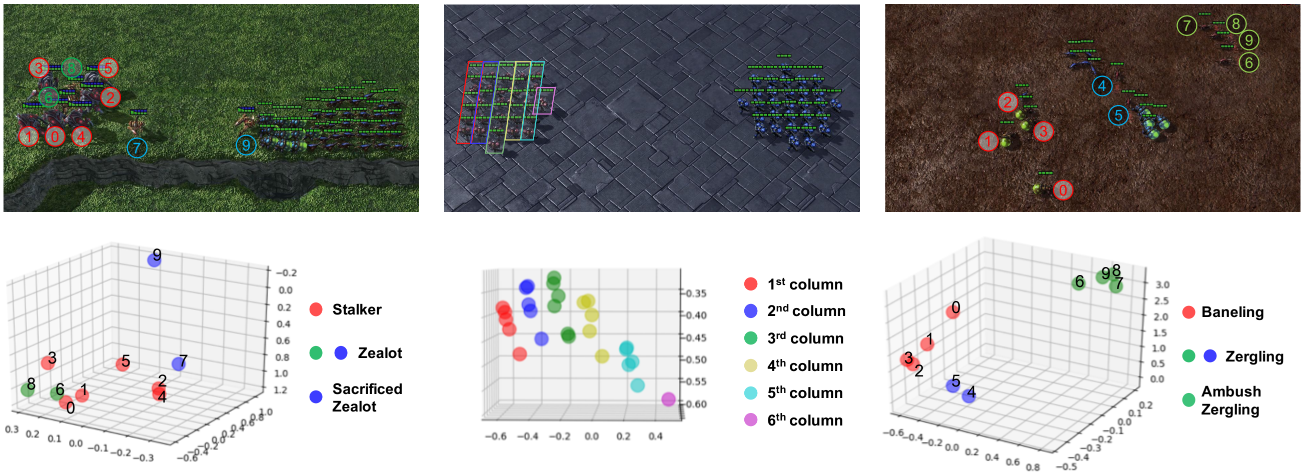
B. Architecture, Hyperparameters, and Infrastructure

B.1. ROMA

In this paper, we base our algorithm on QMIX (Rashid et al., 2018), whose framework is shown in Fig. 5 and described in Appendix D. In ROMA, each agent has a neural network to approximate its local utility. The local utility network consists of three layers, a fully-connected layer, followed by a 64 bit GRU, and followed by another fully-connected layer that outputs an estimated value for each action. The local utilities are fed into a mixing network estimating the global action value. The mixing network has a 32-dimensional hidden layer with ReLU activation. Parameters of the mixing network are generated by a hyper-net conditioning on the global state. This hyper-net has a fully-connected hidden layer of 32 dimensions. These settings are the same as QMIX.

We use very simple network structures for the components related to role embedding learning, i.e., the role encoder, the role decoder, and the trajectory encoder. The multi-variate Gaussian distributions from which the individual roles are drawn have their means and variances generated by the role encoder, which is a fully-connected network with a 12-dimensional hidden layer with ReLU activation. The parameters in the second fully-connected layers of the local utility approximators are generated by the role decoder whose inputs are the individual roles, which are 3-dimensional in all experiments. The role decoder is also a fully-connected network with a 12-dimensional hidden layer and ReLU activation. For the trajectory encoder, we again use a fully-connected network with a 12-dimensional hidden layer and ReLU activation. The inputs of the trajectory encoder are the hidden states of the GRUs in the local utility functions after the last time step.

For all experiments, we set $\lambda_I = 10^{-4}$, $\lambda_D = 10^{-2}$, and the discount factor $\gamma = 0.99$. The optimization is conducted using RMSprop with a learning rate of 5×10^{-4}, α of 0.99, and with no momentum or weight decay. For exploration, we use ϵ-greedy with ϵ annealed linearly from 1.0 to 0.05 over $50k$ time steps and kept constant for the rest of the training. We run 8 parallel environments to collect samples. Batches of 32 episodes are sampled from the replay buffer, and the whole



(a) Strategy: sacrificing Zealots 9 and 7 to minimize Banelings' splash damage.

(b) Strategy: forming an offensive concave arc quickly

(c) Strategy: green Zerglings hide away and Banelings kill most enemies by explosion.

Figure 2. (Reproduced from Fig. 6 in the paper, for quick reference.) Learned roles for `6s4z_vs_10b30z`, `27m_vs_30m`, and `6z4b` (means of the role distributions, $\boldsymbol{\mu}_{\rho_i}$, are shown, without using any dimensionality reduction techniques), and the related, automatically discovered responsibilities.

framework is trained end-to-end on fully unrolled episodes. All experiments on StarCraft II use the default reward and observation settings of the SMAC benchmark.

Experiments are carried out on NVIDIA GTX 2080 Ti GPU.

B.2. Baselines and Ablations

We compare ROMA with various baselines and ablations, which are listed in Table. 1 of the paper. For COMA (Foerster et al., 2018), QMIX (Rashid et al., 2018), and MAVEN (Mahajan et al., 2019), we use the codes provided by the authors where the hyper-parameters have been fin-tuned on the SMAC benchmark. QMIX-NPS uses the identical architecture as QMIX, and the only difference lies in that QMIX-NPS does not share parameters among agents. Compared to QMIX, for the local utility function of agents, QMIX-LAR adds two more fully-connected layers of 80 and 25 dimensions after the GRU layer so that it approximately has the same number of parameters as ROMA.

C. Additional Experimental Results

We benchmark our method on the StarCraft II unit micromanagement tasks. To test the scalability of the proposed approach, we introduce three maps. The `6z4b` map features symmetry teams consisting of 4 Banelings and 6 Zerglings. In the map of `6s4z_vs_10b30z`, 6 Stalkers and 4 Zealots learn to defeat 10 Banelings and 30 Zerglings. And `10z5b_vs_2z3s` characterizes asymmetry teams consisting of 10 Zerglings & 5 Banelings and 2 Zealots & 3 Stalkers, respectively.

C.1. Performance Comparison against Baselines

Fig. 1 presents performance of ROMA against various baselines on three maps. Performance comparison on the other maps is shown in Fig. 4 of the paper. We can see that the advantage of ROMA is more significant on maps with more agents, such as `10z5b_vs_2z3s`, `MMM2`, `27_vs_30m`, and `10m_vs_11m`.

C.2. Role Embedding Representations

Fig. 2 shows various roles learned by ROMA. Roles are closely related to the sub-tasks in the learned winning strategy.

For the map `27m_vs_30m`, the winning strategy is to form an offensive concave arc before engaging in the battle. Fig. 2(b) illustrates the role embedding representations at the first time step when the agents are going to set up the attack formation. We can see the roles aggregate according to the relative positions of the agents. Such role differentiation leads to different

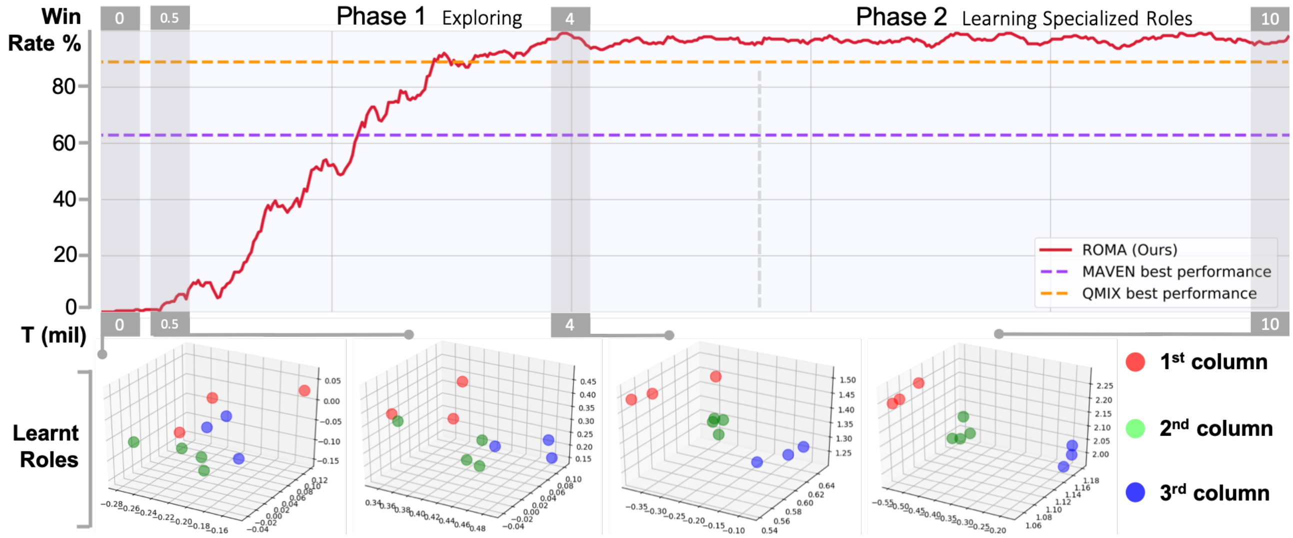


Figure 3. The process of role emergence and evolution on the map `10m_vs_11m`.

moving strategies so that agents can quickly form the arc without collisions.

Similar role-behavior relationships can be seen in all tasks. We present another example on the task of `6z4b`. In the winning strategy learned by ROMA, Zerglings 4 & 5 and Banelings kill most of the enemies, taking advantage of the splash damage of the Banelings, while Zerglings 6-9 hideaway, wait until the explosion is over, and then kill the remaining enemies. Fig. 2(c) shows the role embedding representations before the explosion. We can see clear clusters closely corresponding to the automatically detected sub-tasks at this time step.

Supported by these results, we can conclude that ROMA can automatically decompose the task and learn versatile roles, each of which is specialized in a certain sub-task.

C.3. Additional Results for Role Evolution

In Fig. 7 of the paper, we show how roles emerge and evolve on the map `MMM2`, where the involved agents are heterogeneous. In this section, we discuss the case of homogeneous agent teams. To this end, we visualize the emergence and evolution process of roles on the map `10m_vs_11m`, which features 10 ally Marines facing 11 enemy Marines. In Fig. 3, we show the roles at the first time step of the battle (screenshot can be found in Fig. 4) at four different stages during the training. At this moment, agents need to form an offensive concave arc quickly. We can see that ROMA gradually learns to allocate roles according to relative positions of agents. Such roles and the corresponding differentiation in the individual policies help agents form the offensive arc more efficiently. Since setting up an attack formation is critical for winning the game, a connection between the specialization of the roles at the first time step and the improvement of the win rate can be observed.

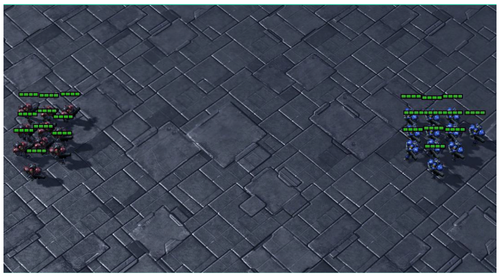


Figure 4. Screenshot of `10m_vs_11m`, t=1.

D. Related Works

Multi-agent reinforcement learning holds the promise to solve many real-world problems and has been making vigorous progress recently. To avoid otherwise exponentially large state-action space, factorizing MDPs for multi-agent systems is proposed (Guestrin et al., 2002). Coordination graphs (Bargiacchi et al., 2018; Yang et al., 2018; Grover et al., 2018; Kipf et al., 2018) and explicit communication (Sukhbaatar et al., 2016; Hoshen, 2017; Jiang & Lu, 2018; Singh et al., 2019;

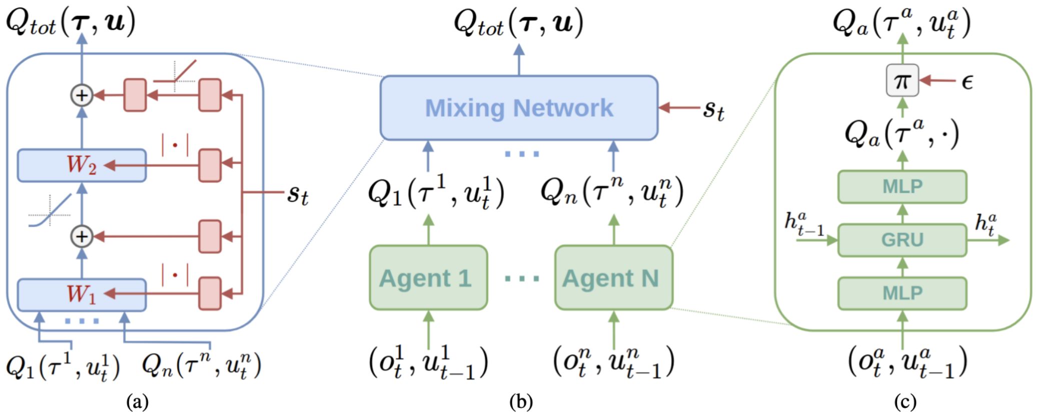


Figure 5. The framework of QMIX, reproduced from the original paper (Rashid et al., 2018). (a) The architecture of the mixing network (blue), whose weights and biases are generated by a hyper-net (red) conditioned on the global state. (b) The overall QMIX structure. (c) Local utility network structure.

Das et al., 2019; Singh et al., 2019; Kim et al., 2019) are studied to model the dependence between the decision-making processes of agents. Training decentralized policies is faced with two challenges: the issue of non-stationarity (Tan, 1993) and reward assignment (Foerster et al., 2018; Nguyen et al., 2018). To resolve these problems, Sunehag et al. (2018) propose a value decomposition method called VDN. VDN learns a global action-value function, which is factored as the sum of each agent's local Q-value. QMIX (Rashid et al., 2018) extends VDN by representing the global value function as a learnable, state-condition, and monotonic combination of the local Q-values. In this paper, we use the mixing network of QMIX. The framework of QMIX is shown in Fig. 5.

The StarCraft II unit micromanagement task is considered as one of the most challenging cooperative multi-agent testbeds for its high degree of control complexity and environmental stochasticity. Usunier et al. (2017) and Peng et al. (2017) study this problem from a centralized perspective. In order to facilitate decentralized control, we test our method on the SMAC benchmark (Samvelyan et al., 2019), which is the same as in (Foerster et al., 2017; 2018; Rashid et al., 2018; Mahajan et al., 2019).

References

Bargiacchi, E., Verstraeten, T., Roijers, D., Nowé, A., and Hasselt, H. Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In *International Conference on Machine Learning*, pp. 491–499, 2018.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., and Pineau, J. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pp. 1538–1546, 2019.

Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H., Kohli, P., and Whiteson, S. Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1146–1155. JMLR. org, 2017.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Grover, A., Al-Shedivat, M., Gupta, J. K., Burda, Y., and Edwards, H. Evaluating generalization in multiagent systems using agent-interaction graphs. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1944–1946. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Guestrin, C., Koller, D., and Parr, R. Multiagent planning with factored mdps. In *Advances in neural information processing systems*, pp. 1523–1530, 2002.

Hoshen, Y. Vain: Attentional multi-agent predictive modeling. In *Advances in Neural Information Processing Systems*, pp. 2701–2711, 2017.

Jiang, J. and Lu, Z. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, pp. 7254–7264, 2018.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., and Yi, Y. Learning to schedule communication in multi-agent reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697, 2018.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pp. 7611–7622, 2019.

Nguyen, D. T., Kumar, A., and Lau, H. C. Credit assignment for collective multiagent rl with global rewards. In *Advances in Neural Information Processing Systems*, pp. 8102–8113, 2018.

Peng, P., Wen, Y., Yang, Y., Yuan, Q., Tang, Z., Long, H., and Wang, J. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

Rashid, T., Samvelyan, M., Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4292–4301, 2018.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Singh, A., Jain, T., and Sukhbaatar, S. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Sukhbaatar, S., Fergus, R., et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pp. 2244–2252, 2016.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Usunier, N., Synnaeve, G., Lin, Z., and Chintala, S. Episodic exploration for deep deterministic policies: An application to starcraft micromanagement tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Yang, Z., Zhao, J., Dhingra, B., He, K., Cohen, W. W., Salakhutdinov, R. R., and LeCun, Y. Glomo: unsupervised learning of transferable relational graphs. In *Advances in Neural Information Processing Systems*, pp. 8950–8961, 2018.