Multi-Agent Network Randomization for Robust Knowledge Transfer in Deep Multi-Agent Reinforcement Learning

Dohyun Kim, Hoseong Jung, Jungho Bae*

Agency for Defense Development

00dh.kim@gmail.com

국 방 과 학 연 구 소
Agency for Defense Development

ICCAS 2024
THE INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS

Multi-Agent Reinforcement Learning

swarm drone control

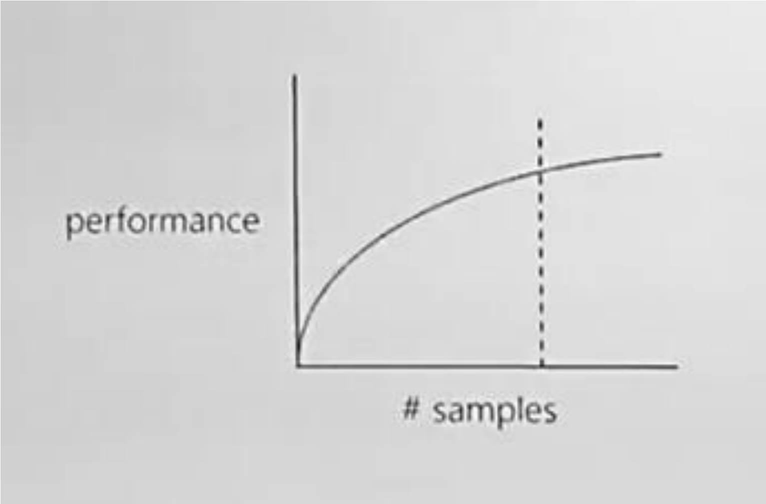
logistics robot collaboration

multi-agent game AI

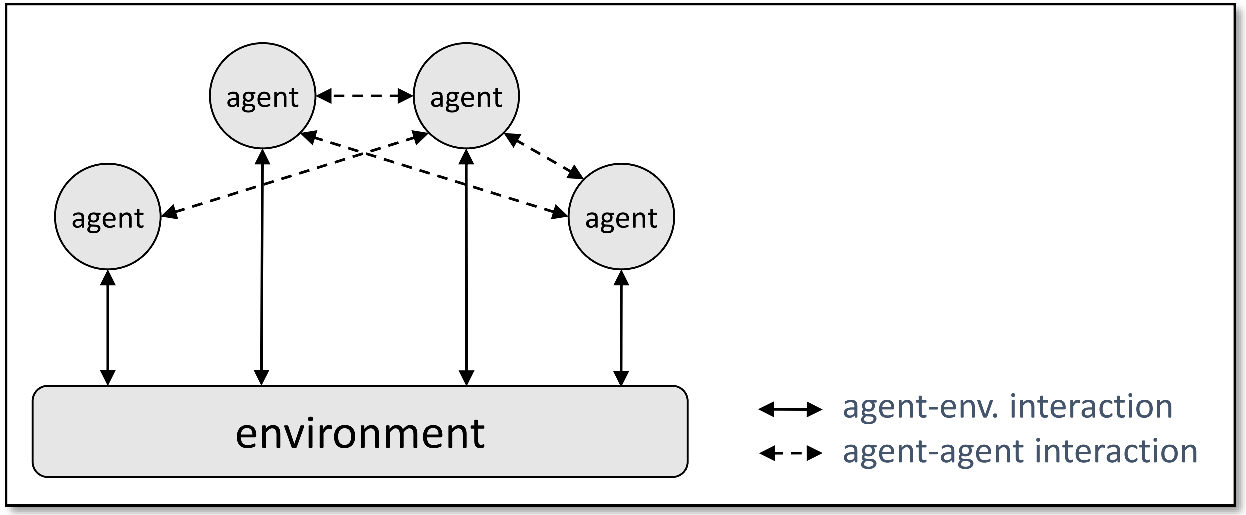
- Cooperative multi-agent reinforcement learning(MARL) is a framework for multiple agents to learn policies towards a common goal. [1]
- Problem in MARL: large state and action space
- Conventional solution: reward shaping(e.g., SMMAE), CTDE paradigm(e.g., QMIX), and so on.

[1] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

Sample Efficiency in Multi-Agent RL



sample efficiency with a limited budget [2]

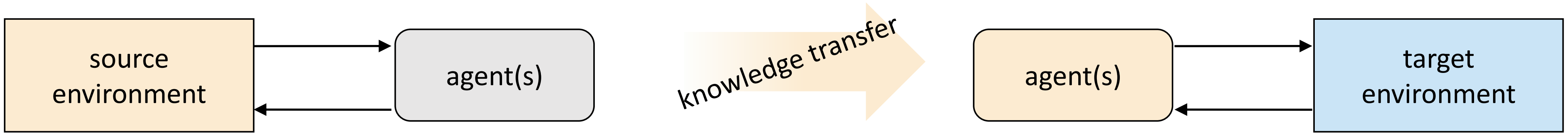


Multi-agent setting framework

- Sample efficiency: how well a model can perform with limited training data
- In multi-agent settings, the state and action space grow exponentially with the number of agents.

[2] H. Van Seijen, "Sample Efficient RL," presented at the DLRL Summer School, Edmonton, AB, Canada, Jul. 24 - Aug. 2, 2019. [Online]. Available: https://www.youtube.com/watch?v=DT2MbIeec4U

Transfer Learning

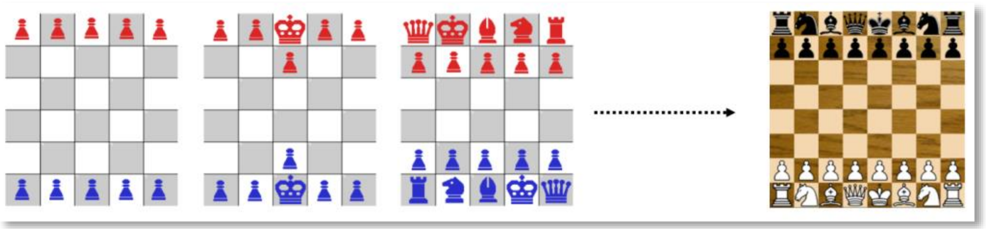


※ *task*, *scenario*, and *environment* are all same terms in this announcement.

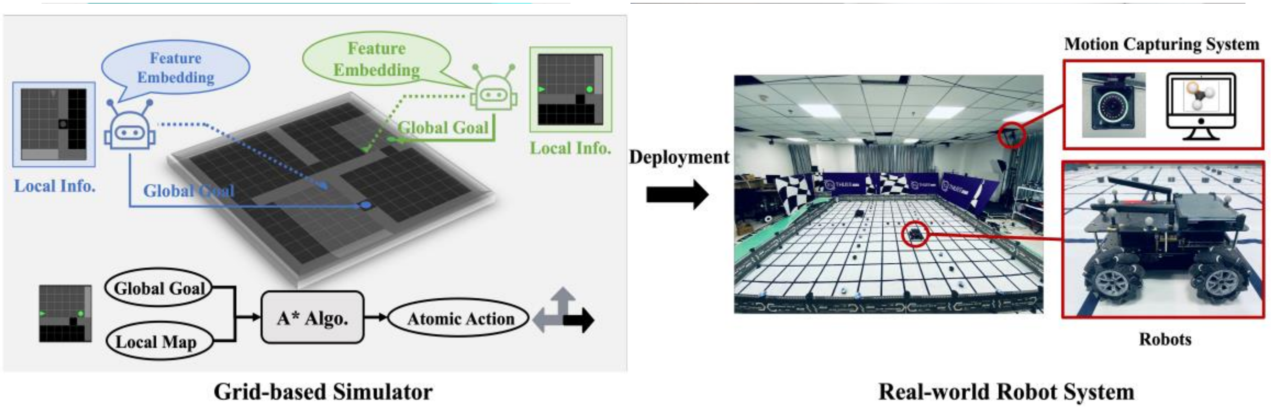
- Transfer learning refers to an approach that knowledge gathered in a source task is utilized in a target task. [3]
- In particular, learning a new environment from scratch requires a large number of samples.

[3] Z. Zhu, Y. Zhang, S. Li, and X. Wang, "Transfer learning in deep reinforcement learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Transfer Learning Examples



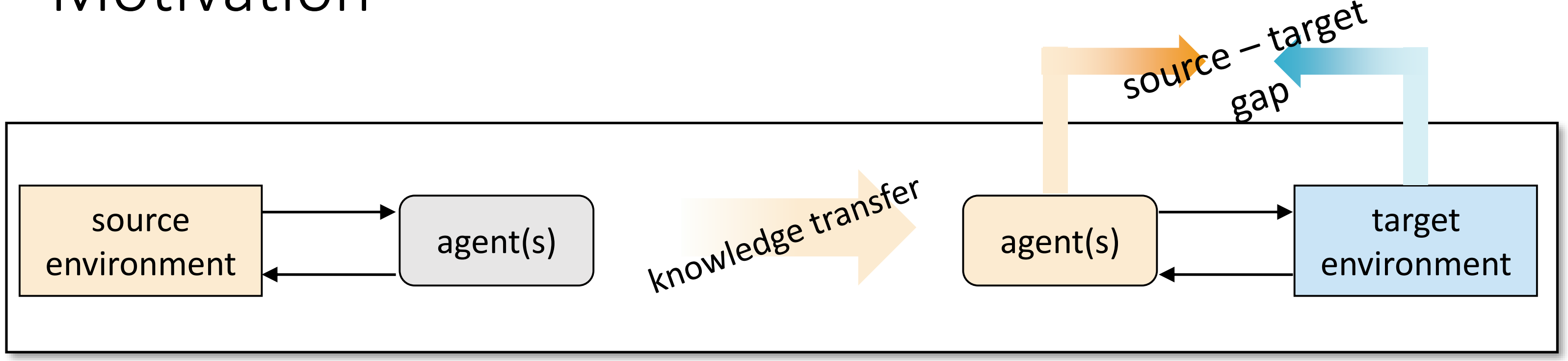
simplified chess board and original chess game [4]



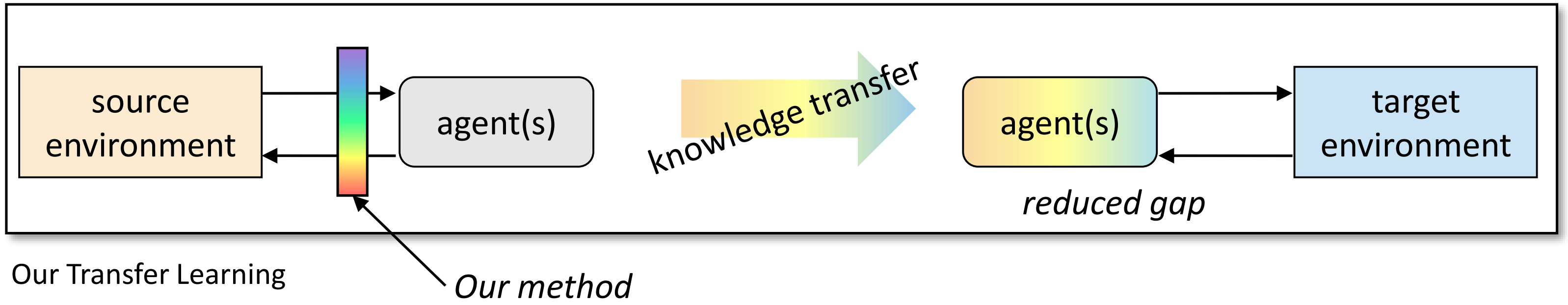
sim-to-real transfer for multi-robot exploration problem [5]

[4] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *Journal of Machine Learning Research*, vol. 21, no. 181, pp. 1–50, 2020.
[5] C. Yu, X. Yang, J. Gao, J. Chen, Y. Li, J. Liu, Y. Xiang, R. Huang, H. Yang, and Y. Wu, "Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration," *in Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1107-1115.

Motivation



Conventional Transfer Learning



Our Transfer Learning

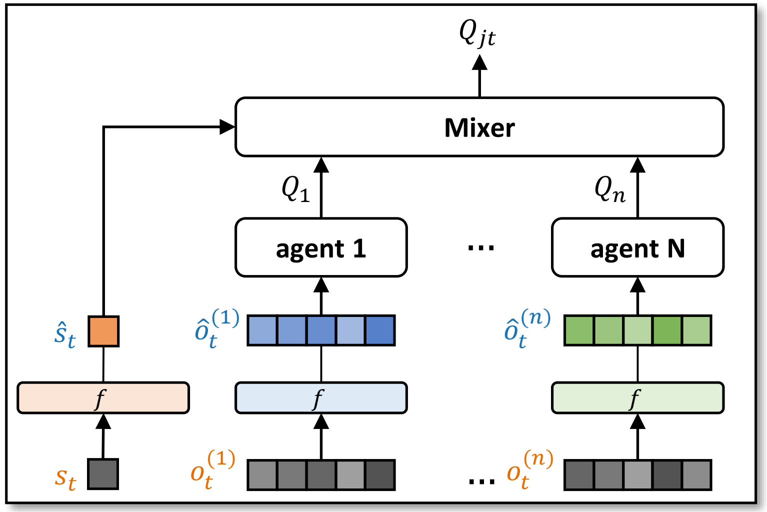
- There is a gap between the source task and the target task.
- Our method collecting task-invariant knowledge

Method

- Our proposed multi-agent network randomization(MANR) method is implemented by adding a random layer to the MARL framework.

$$\hat{o}^{(i)} = f^{(i)}\left(o^{(i)};\ \phi^{(i)}\right)$$

$$\phi^{(i)} = \mathrm{diag}\left(\phi_j^{(i)}\right), \qquad \phi_j^{(i)} \sim \mathrm{Uniform}(1-\delta, 1+\delta)$$



The MANR framework

Advantages:

- Increase the diversity of the data --> generalized and robust knowledge
- Reduce overfitting to specific observation patterns

Method

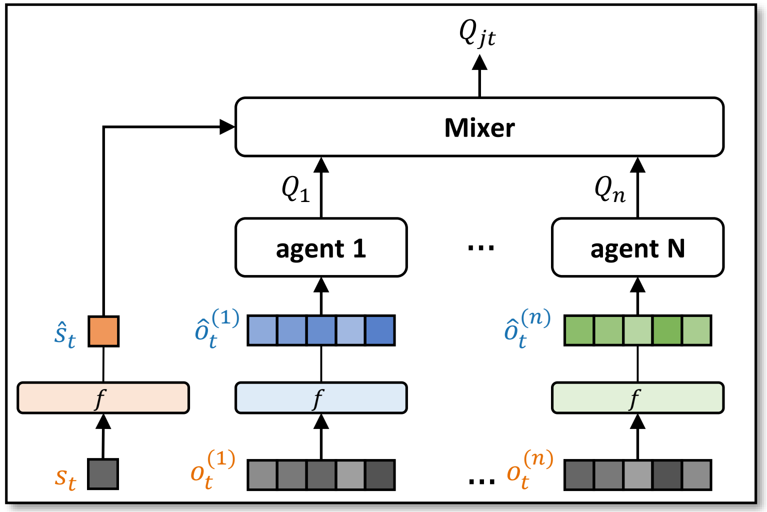
- QMIX [6]: individual agent networks + mixer network

individual agent networks

- input: observation, output: Q-values

mixer network

- combines Q-values to predict team rewards



The MANR framework

$$\mathcal{L}_{TD} = \mathbb{E}\left[\left(y - Q_{jt}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right]$$

[6] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

Method

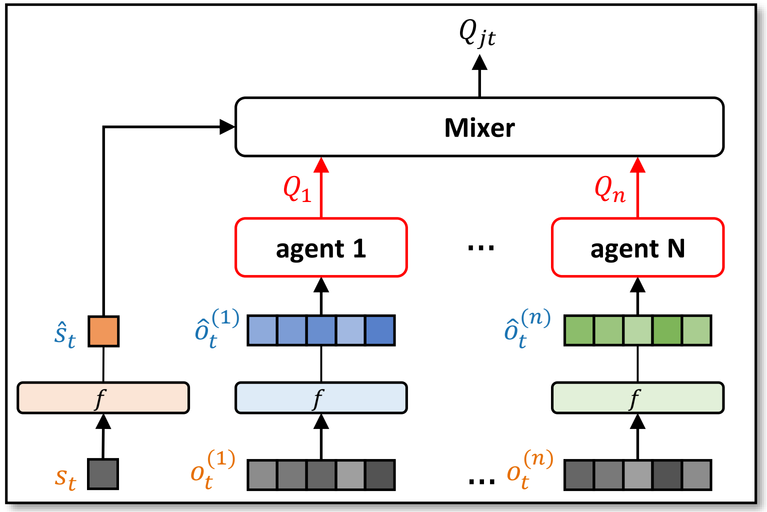
- QMIX [6]: individual agent networks + mixer network

individual agent networks

- input: observation, output: Q-values

mixer network

- combines Q-values to predict team rewards



The MANR framework

$$\mathcal{L}_{TD} = \mathbb{E}\left[\left(y - Q_{jt}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right]$$

[6] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

Method

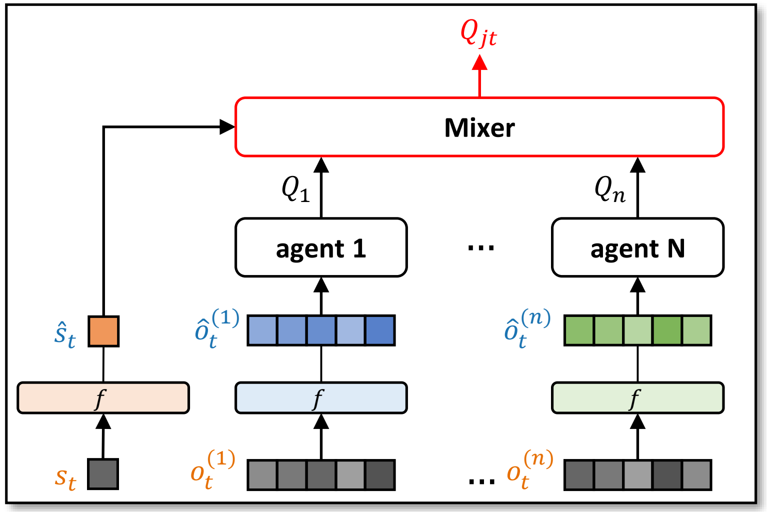
- QMIX [6]: individual agent networks + mixer network

individual agent networks

- input: observation, output: Q-values

mixer network

- combines Q-values to predict team rewards



The MANR framework

$$\mathcal{L}_{TD} = \mathbb{E}\left[\left(y - Q_{jt}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right]$$

[6] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

Method

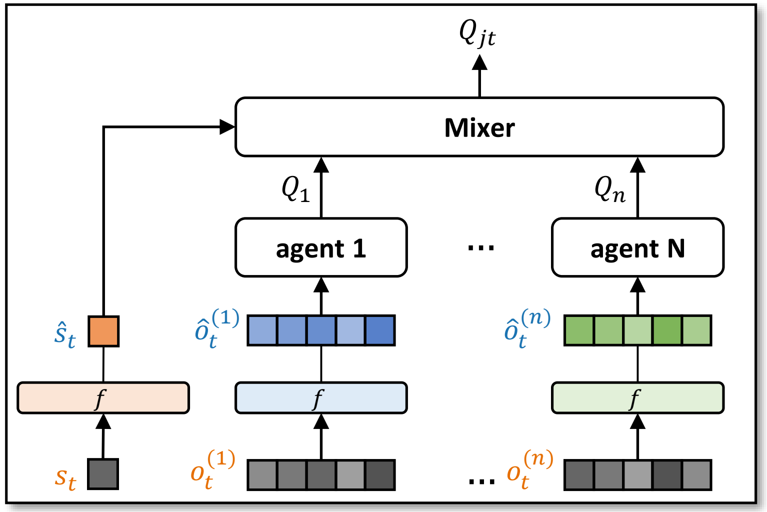
- QMIX [6]: individual agent networks + mixer network

individual agent networks

- input: observation, output: Q-values

mixer network

- combines Q-values to predict team rewards



The MANR framework

$$\mathcal{L}_{TD} = \mathbb{E}\left[\left(\boxed{y} - \boxed{Q_{jt}(\boldsymbol{s}, \boldsymbol{a})}\right)^2\right]$$

model's predicted Q-value

target Q-value

$$\boxed{y = r + \gamma \cdot Q_{jt}^{target}(\boldsymbol{s}', \boldsymbol{a}')}$$

[6] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

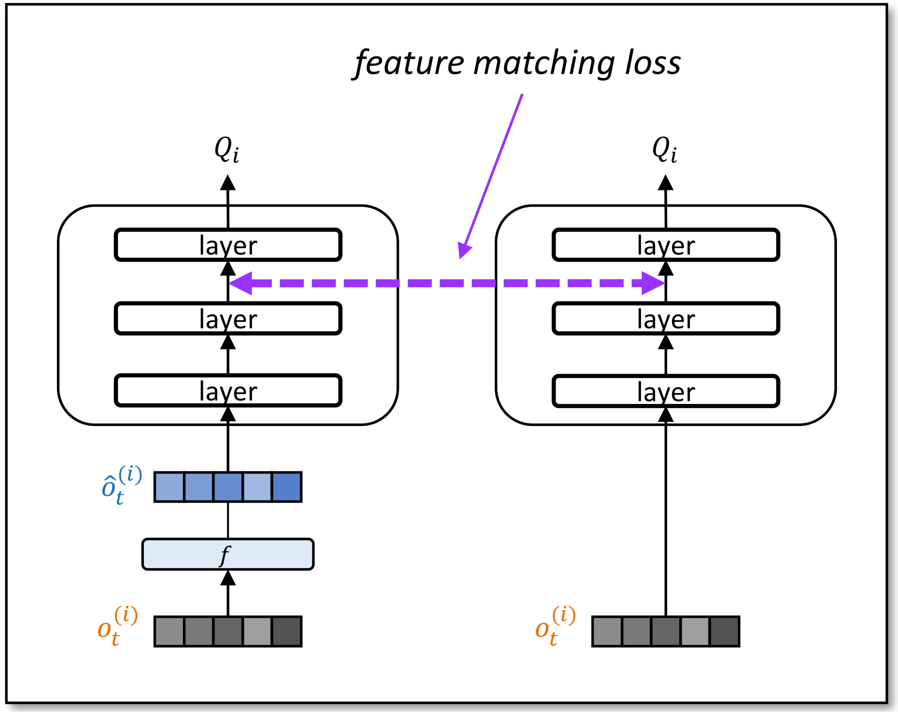
Method (Cont')

- feature matching loss between clean and randomized observation [7]

$$\mathcal{L}_{FM} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\left\|h(\hat{o}^{(i)};\theta) - h(o^{(i)};\theta)\right\|^2\right]$$

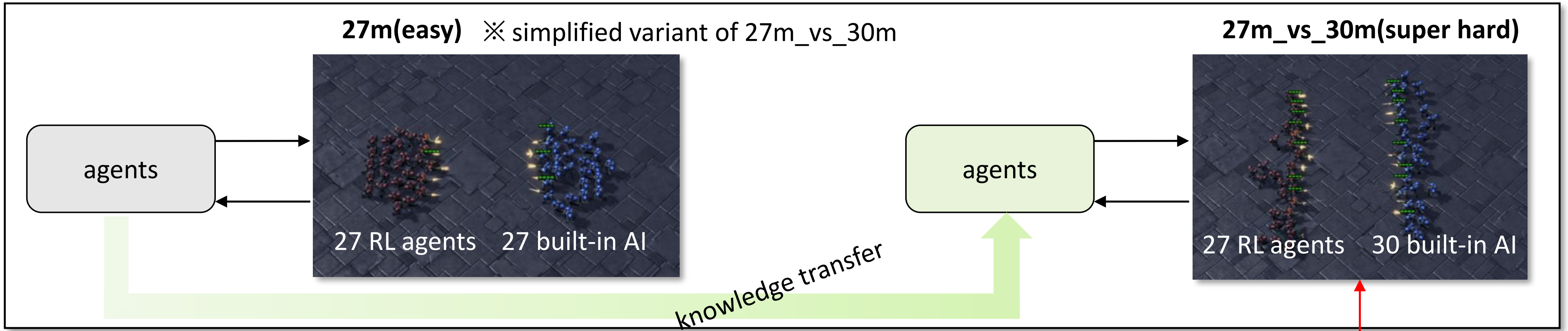
$$\mathcal{L} = \mathcal{L}_{TD} + \mathcal{L}_{FM}$$

- Effectiveness: more stable against injected randomness



[7] K. Lee, K. Lee, J. Shin, and H. Lee, "Network randomization: A simple technique for generalization in deep reinforcement learning," *arXiv preprint arXiv:1910.05396*, 2019.

Experimental Setup

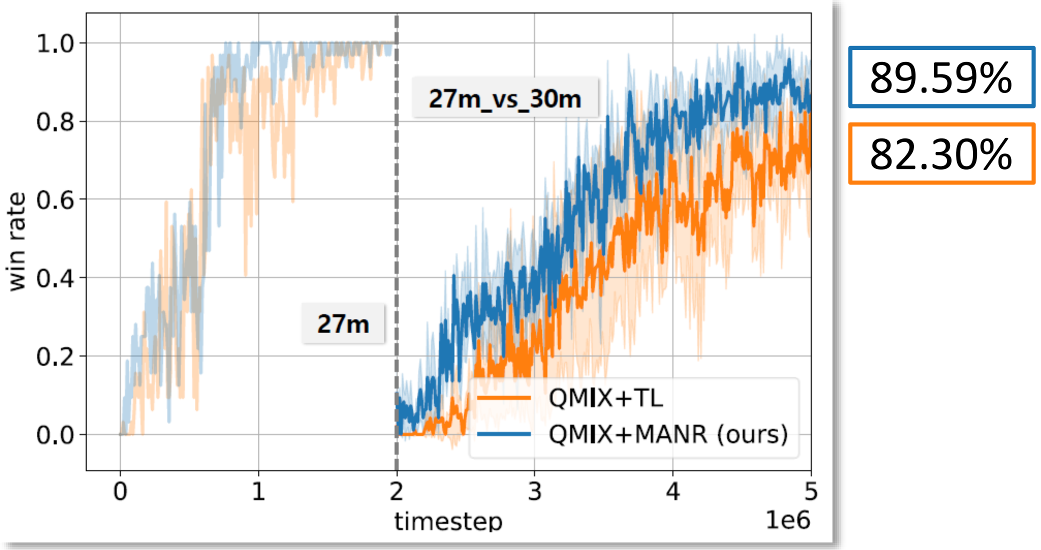


- StarCraft Multi-Agent Challenge(SMAC)
- target task: 27m_vs_30m most agents in the SMAC
- source task: learn general knowledge

Name	n_agents	Difficulty
2s_vs_1sc	2	Easy
2c_vs_64zg	2	Hard
3s_vs_5z	3	Hard
..	..	..
MMM2	10	Super Hard
bane_vs_bane	24	Hard
27m_vs_30m	**27**	**Super Hard**

List of SMAC scenarios

Result: Performance



Win rate result graph

- We validate our method based on QMIX algorithm
- **QMIX+MANR (ours)** : QMIX combined with our proposed method
- **QMIX+TL (baseline)** : the application of vanilla transfer learning to QMIX

Robustness Definition

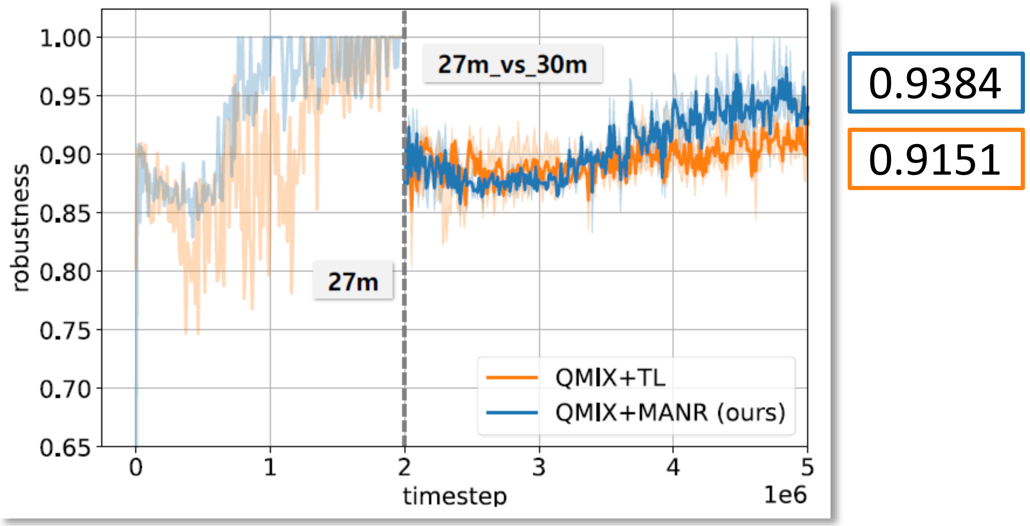
- Robustness : degree of stability in predictive performance despite variations in input data. [8]
- We need a task independent metrics.

$$robustness = \exp\left(-\frac{\sqrt{\mathbb{V}\left(\boldsymbol{R}\right)}}{\mathbb{E}(\boldsymbol{R})}\right), \boldsymbol{R} \textit{ is a set of returns.}$$

- We define robustness as the standard deviation normalized by the mean over the returns of multiple episodes.

[8] H. B. Braiek and F. Khomh, "Machine Learning Robustness: A Primer," *arXiv preprint arXiv:2404.00897*, 2024.

Result: Robustness and Entropy



Robustness result graph

QMIX+MANR (Ours)	QMIX+TL (Baseline)
0.5401	0.3620

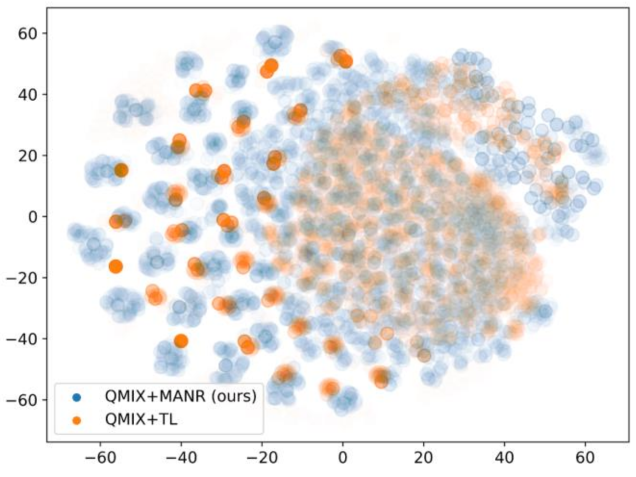
Table of entropy results

$$H(S) = -\sum_{i=1}^{n}\sum_{j=1}^{b} P(x_{ij}) \log P(x_{ij})$$

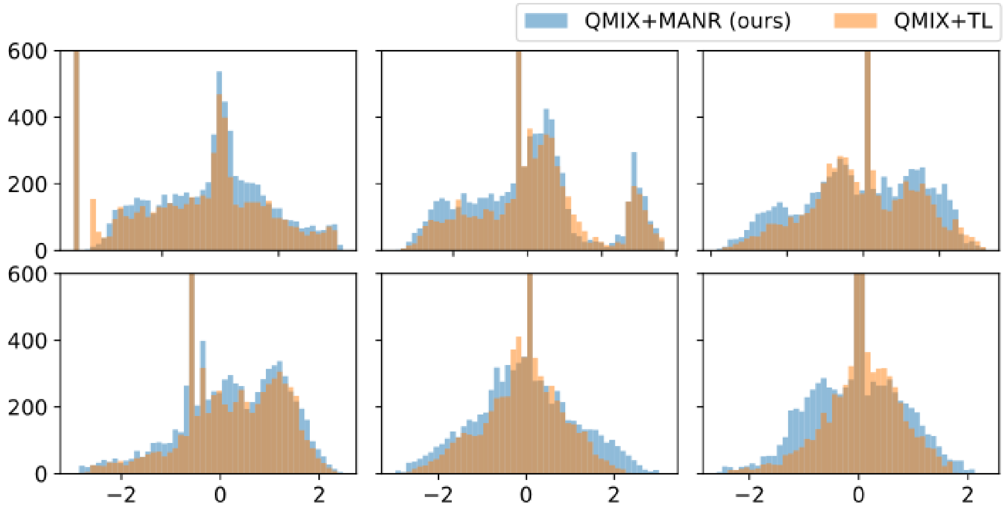
Entropy formula

- We compared robustness for **QMIX+MANR** and **QMIX+TL** with the same setup.
- Improved robustness and entropy suggest that our method makes the model more general.

Result: t-SNE and PCA



t-SNE clustering analysis



PCA visualization analysis

- We show t-SNE and PCA analysis to verity qualitative results.
- Two visualizations show the better generalization ability of our method compared to the baseline.

Conclusion

- We propose MANR method to improve the generalization ability for applying to MARL.
- The MANR method injects randomness into the training data by introducing random layers.
- Our method aims to
  - enhance robustness and facilitate knowledge transfer,
  - be compatible with almost MARL algorithms,
  - and be easy to implement.

Dohyun Kim, Hoseong Jung, Jungho Bae*

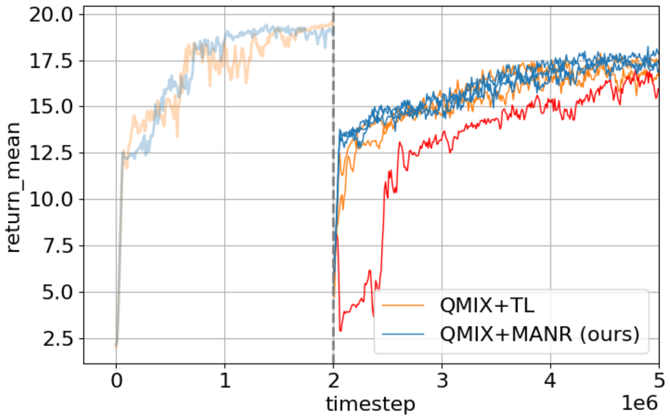
00dh.kim@gmail.com

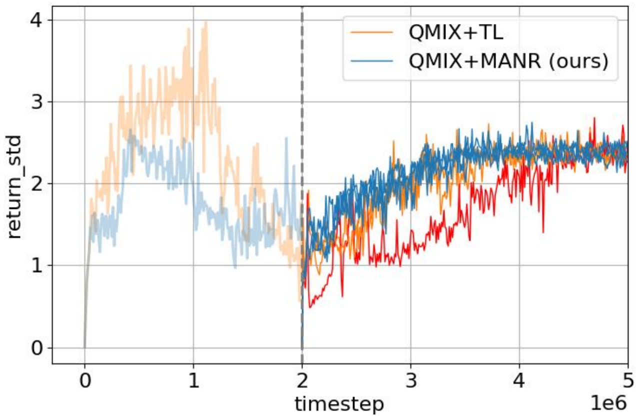
Conclusion

- We propose MANR method to improve the generalization ability for applying to MARL.
- The MANR method injects randomness into the training data by introducing random layers.
- Our method aims to
  - enhance robustness and facilitate knowledge transfer,
  - be compatible with almost MARL algorithms,
  - and be easy to implement.

Dohyun Kim, Hoseong Jung, Jungho Bae*
00dh.kim@gmail.com

Appendix


return_mean
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0
1
2
3
4
5
timestep
1e6
QMIX+TL
QMIX+MANR (ours)


return_std
4
3
2
1
0
0
1
2
3
4
5
timestep
1e6
QMIX+TL
QMIX+MANR (ours)

Appendix

domain randomization	multi-agent network randomization
• randomize elements of environments (e.g., lighting, textures, physical properties) • task-dependent method	• add random layer to network • task-independent method • increase robustness