QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning

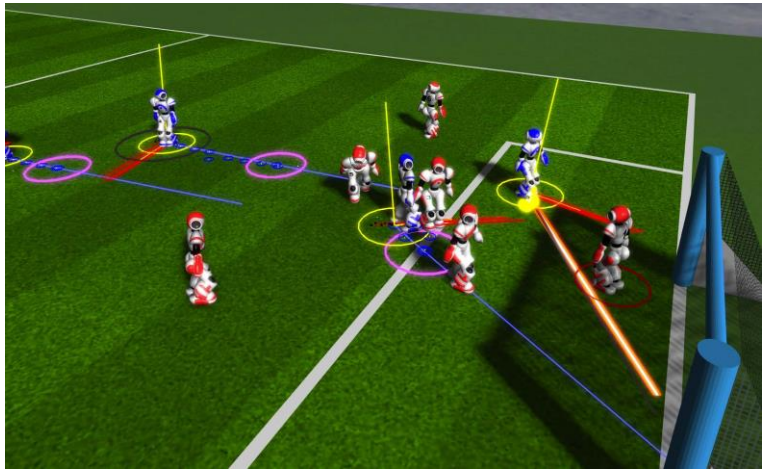
Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, Yung Yi

School of Electrical Engineering, KAIST

KAIST LANADA

Cooperative Multi-Agent Reinforcement Learning

Drone Swam Control



Cooperation Game

Network Optimization

- Distributed multi-agent systems with a **shared reward**
- Each agent has an individual, **partial observation**
- **No communication** between agents
- The goal is to maximize the shared reward

Background

- Fully centralized training
  - Not applicable to distributed systems

$$Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}),\ \pi_{jt}(\boldsymbol{\tau}, \boldsymbol{u})$$

- Fully decentralized training
  - Non-stationarity problem

$$Q_i(\tau_i, u_i),\ \pi_i(\tau_i, u_i)$$

- Centralized training with decentralized execution
  - **Value function factorization**[1,2]**,** Actor-critic method[3,4]
  - Applicable to distributed systems
  - No non-stationarity problem

$$Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) \rightarrow Q_i(\tau_i, u_i), \pi_i(\tau_i, u_i)$$

[1] Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., and Graepel, T. *Value decomposition networks for cooperative multi-agent learning based on team reward*. In Proceedings of AAMAS, 2018.
[2] Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. *QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning.* In Proceedings of ICML, 2018.
[3] Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. *Counterfactual multi-agent policy gradients. In Proceedings of AAAI, 2018.*
[4] Lowe, R., WU, Y., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. *Multi-agent actor-critic for mixed cooperative-competitive environments.* In Proceedings of NIPS, 2017.

Previous Approaches

- VDN (Additivity assumption)
  - Represent the joint Q-function as a sum of individual Q-functions

$$Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = \sum_{i=1}^{N} Q_i(\tau_i, u_i)$$

- QMIX (Monotonicity assumption)
  - The joint Q-function is monotonic in the per-agent Q-functions

$$\frac{\partial Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u})}{\partial Q_i(\tau_i, u_i)} \geq 0$$

- They have **limited representational complexity**

Agent 1 \ Agent 2	A	B	C
A	**8**	-12	-12
B	-12	0	0
C	-12	0	0

Non-monotonic matrix game

$Q_1(u_1)$ \ $Q_2(u_2)$	-3.14	**-2.29**	-2.41
-2.29	-5.42	-4.57	-4.70
-1.22	-4.35	-3.51	-3.63
-0.75	-3.87	**-3.02**	-3.14

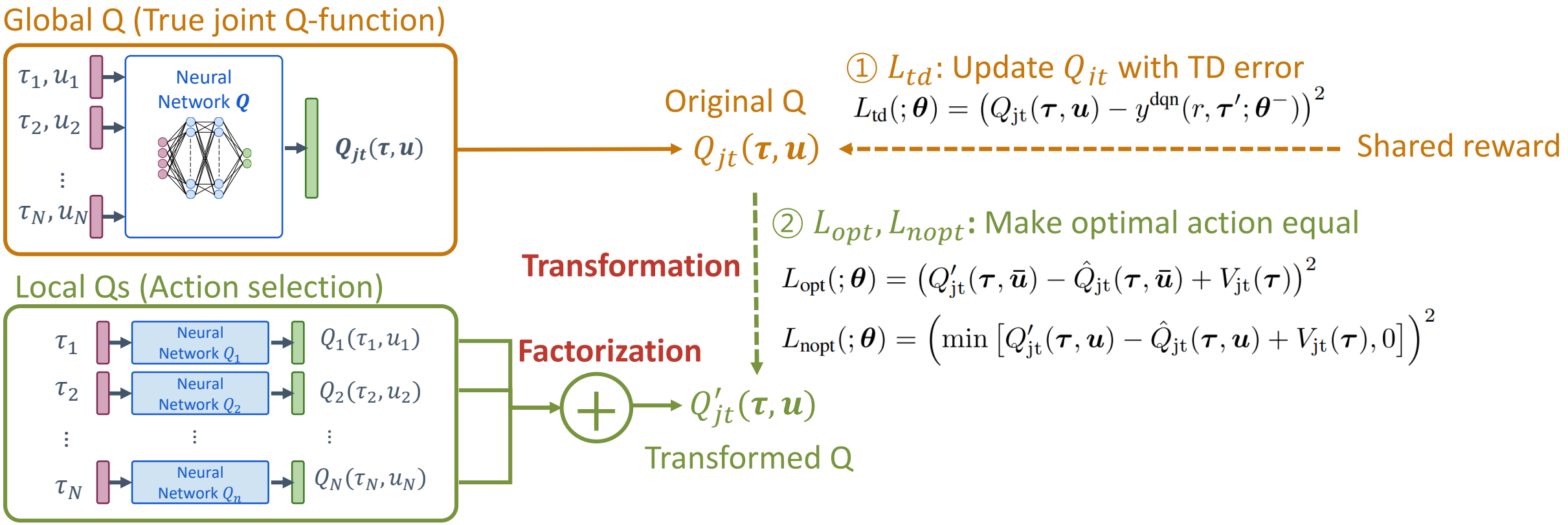
VDN Result Q_{jt}

$Q_1(u_1)$ \ $Q_2(u_2)$	-0.92	0.00	**0.01**
-1.02	-8.08	-8.08	-8.08
0.11	-8.08	0.01	**0.03**
0.10	-8.08	0.01	0.02

QMIX Result Q_{jt}

QTRAN: Learning to Factorize with Transformation

- Instead of direct value factorization, we factorize the **transformed joint Q-function**
- Additional objective function for transformation
  - The original joint Q-function and the transformed Q-function have the **same optimal policy**
  - The transformed joint Q-function is **linearly factorizable**
  - Argmax operation for the original joint Q-function is not required



Theoretical Analysis

- The objective functions make $Q'_{jt} - Q_{jt}$ zero for optimal action (L_{opt}), and positive for the rest (L_{nopt})
  → **Then, optimal actions are the same (Theorem 1)**
- Our theoretical analysis demonstrates that QTRAN handles a **richer class of tasks (= IGM condition)**

$$\arg\max_{\boldsymbol{u}} Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = \begin{pmatrix} \arg\max_{u_1} Q_1(\tau_1, u_1) \\ \vdots \\ \arg\max_{u_N} Q_N(\tau_N, u_N) \end{pmatrix}$$

Agent 2

Agent 1	A	B	C
A	**8.00**	-12.02	-12.02
B	-12.00	0.00	0.00
C	-12.00	0.00	-0.01

QTRAN Result Q_{jt}

$Q_2(u_2)$

$Q_1(u_1)$	**4.16**	2.29	2.29
3.84	**8.00**	6.13	6.12
-2.06	2.10	0.23	0.23
-2.25	1.92	0.04	0.04

QTRAN Result Q'_{jt}

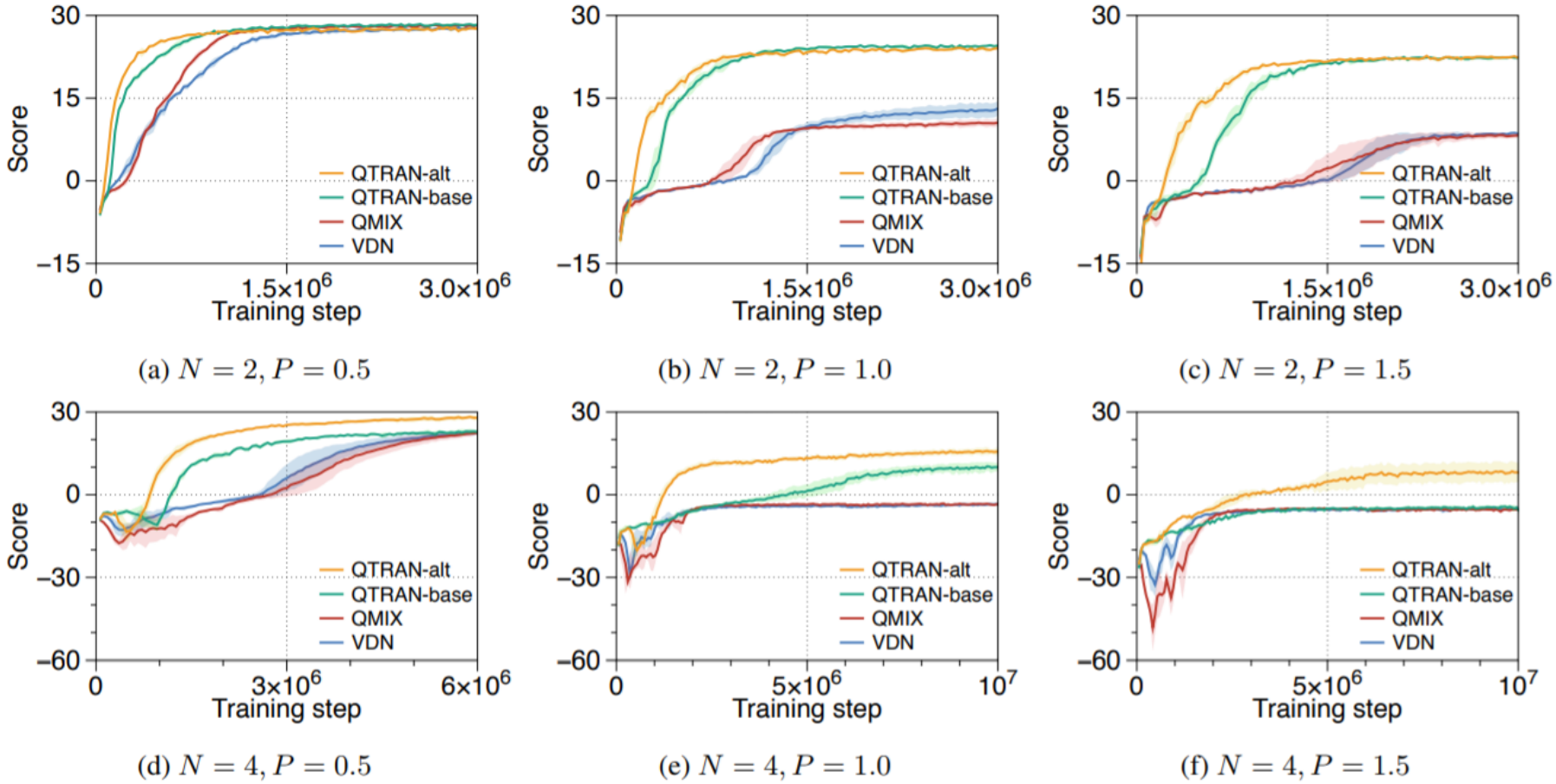
Agent 2

Agent 1	A	B	C
A	**0.00**	18.14	18.14
B	14.11	0.23	0.23
C	13.93	0.05	0.05

QTRAN Result $Q'_{jt} - Q_{jt}$

Results

- QTRAN outperforms VDN and QMIX by a **substantial margin**, especially so when the game exhibits **more severe non-monotonic characteristics**



(a) $N = 2, P = 0.5$ (b) $N = 2, P = 1.0$ (c) $N = 2, P = 1.5$

(d) $N = 4, P = 0.5$ (e) $N = 4, P = 1.0$ (f) $N = 4, P = 1.5$

Thank you!

Pacific Ballroom #58