QMIX: Monotonic Value Function Factorization for Deep Multi-agent Reinforcement Learning

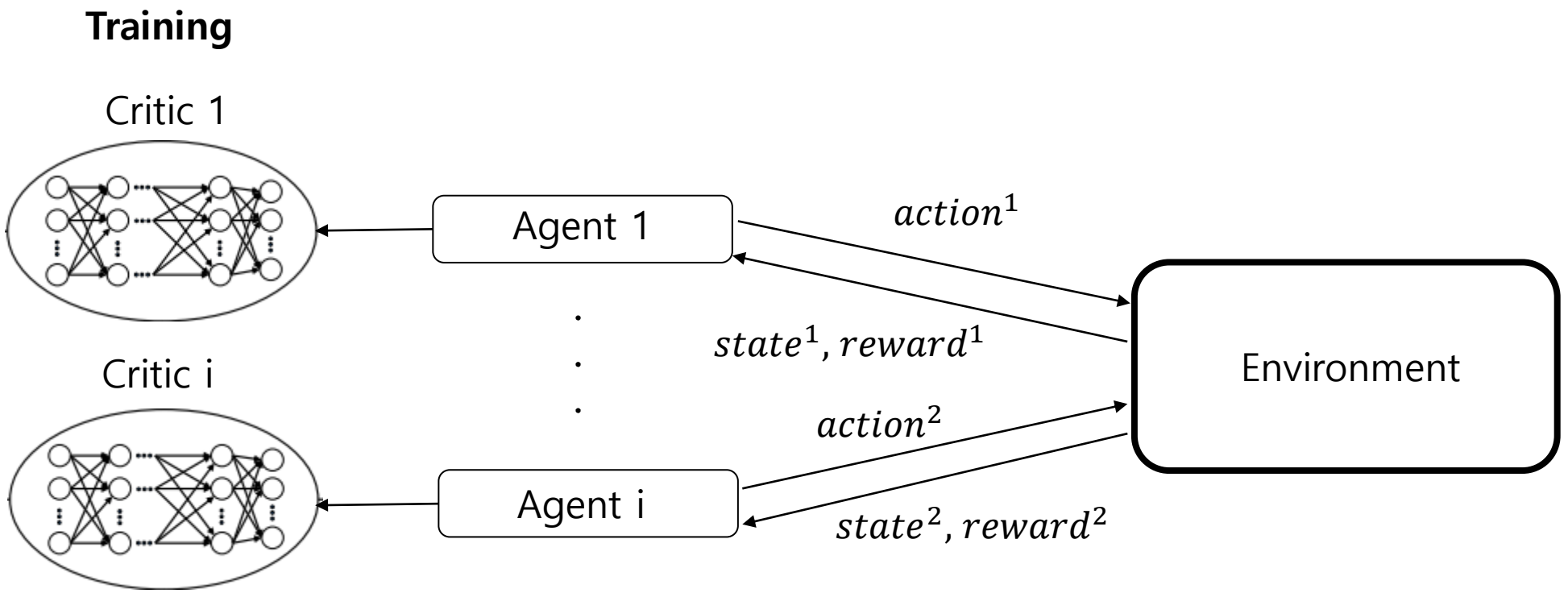
Tabish Rashid, et al
The Journal of Machine Learning (2018)

IIE8557-01 동적계획법과 강화학습

경영과학연구실 김변민

Challenge of Multi agent system

- 분산학습구조 에서는 타 agent 의 액션으로 인해 non-stationarity issue가 발생할 수 있음



Challenge of Multi agent system

- Fully Centralized 구조에서는 에이전트의 개수에 따라 Joint action space 가 지수적으로 증가할 수 있음

EX) N 개의 agent , k개의 action 일 때 joint action space = k^N

{Down}
{UP}
{Right}
{Left}
{Attack}

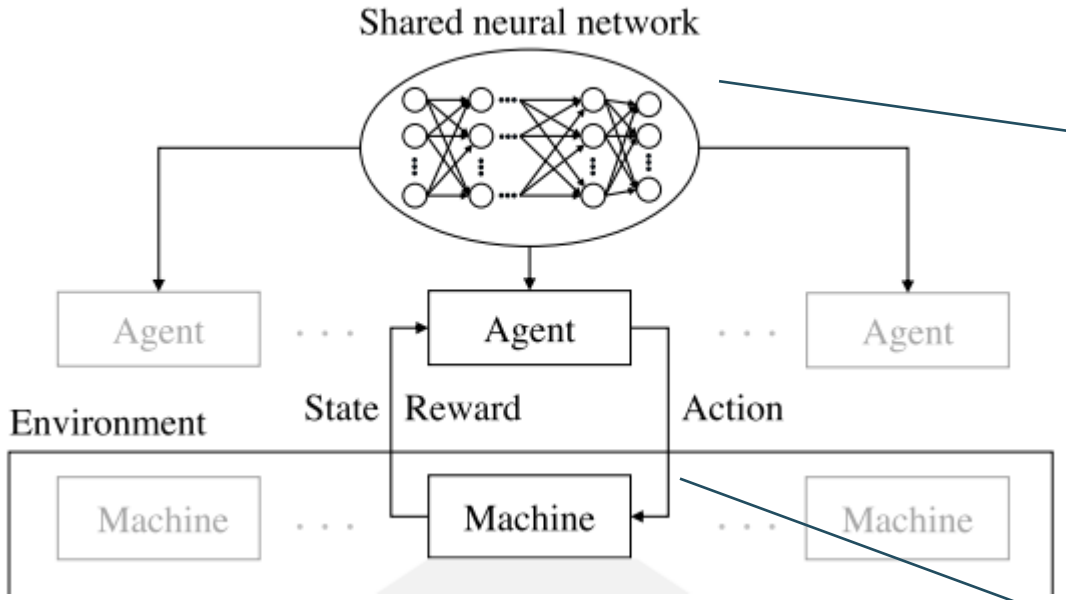
5 actions

{Down, up},
{UP, Attack},
{Right, up}
.
.

25 actions

CTDE (Centralized Training Decentralized Execution)

- Training 과정 중에는 모든 에이전트의 관측정보를 이용하고, 실행 과정에서는 각 에이전트 자신의 관측 정보만을 이용하여 실행함



Centralized Training

Decentralized Execution

DEC-POMDP (Decentralized partially observable Markov Decision process)

G = <S, U, P, r, Z, O, n,γ>

- s ∈ S : state
- u ∈ U : Joint action
- P(s'|s, u) : transition function
- r(s,u) : reward function
- a : agent
- z ∈ **Z : observation**
- **O(s,a) : observation function**
- τ^a ∈ τ **: action-observation history**
- γ : discount rate

CTDE (Centralized Training Decentralized Execution)의 제약조건

- 모든 에이전트는 부분 관측 정보만을 이용하게 됨
- 보상은 각 에이전트 개개인이 아닌 팀 보상으로 주어짐

팀 보상만 존재하는 멀티 에이전트 환경에서는 학습 단계에서 모든 에이전트의 관측 정보와 행동을 입력으로 하는 공동 행동가치함수 (Q_{total})를 계산할 수 있음

하지만 실행 단계에서는 각각의 에이전트는 자신의 관측 범위 내에 있는 정보만을 이용할 수 있어 공동 행동 가치 함수를 사용할 수 없음

CTDE 구조에서 agent별 action value function을 구할 수 있는 알고리즘을 개발하고자 함

Key Idea

- CTDE 구조에서 각 에이전트의 행동가치함수를 추정할 수 있는 방법을 개발함
- 간단한 방법을 통해 식 (1)을 만족하여 행동가치함수를 추정할 수 있는 환경을 구성함

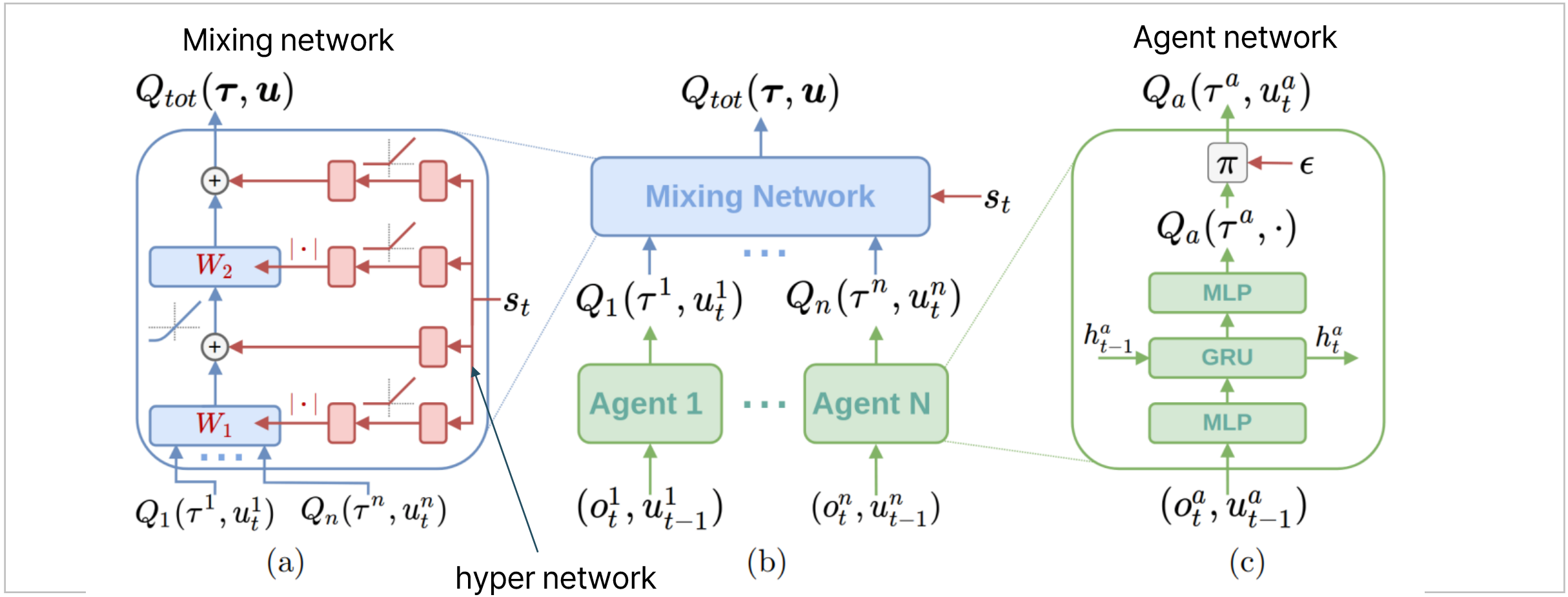
$$\underset{\mathbf{u}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}. \quad (1)$$

해당 식을 만족해야만 공동행동가치 함수를 이용한 각 에이전트의 행동가치함수가 추정 가능함

How to ensure this ?

Qmix Architecture

- Agent network: 각 에이전트의 개별관찰o_t^a와 전 액션 u_{t-1}^a를 인풋으로 받아 에이전트 별 Q 값을 예측
- Mixing network : 각 에이전트에서 산출된 Q 값을 인풋으로 받아 전체 Q_{total} 값을 예측함
- Hyper network : global state를 인풋으로 넣어 Mixing network 의 가중치를 결정함



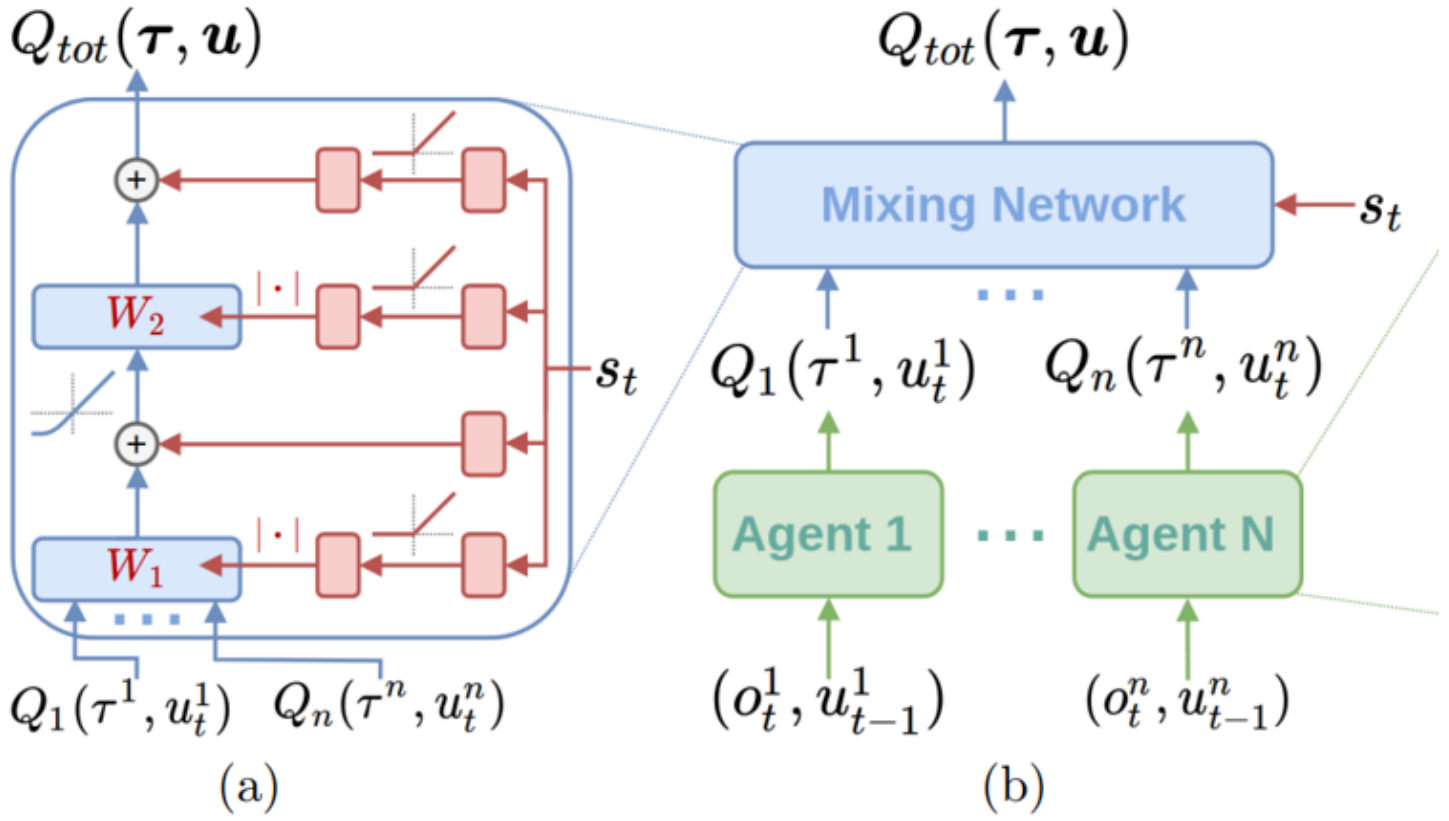
Mixing network

- Mixing network의 가중치가 양의 실수라 가정 할 때, $\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \ \forall a \in A,$ 의 좌변이 항상 양의 값을 갖게 됨
- 이러한 특성은 기여한 에이전트 모두 업데이트가 가능함을 의미함

Loss function of Mixing network

$$\mathcal{L}(\theta) = \sum_{i=1}^{b} \left[\left(y_i^{tot} - Q_{tot}(\boldsymbol{\tau}, \mathbf{u}, s; \theta) \right)^2 \right]$$

$$y^{tot} = r + \gamma \max_{\mathbf{u}'} Q_{tot}(\boldsymbol{\tau}', \mathbf{u}', s'; \theta^-)$$



Monotonicity constraint

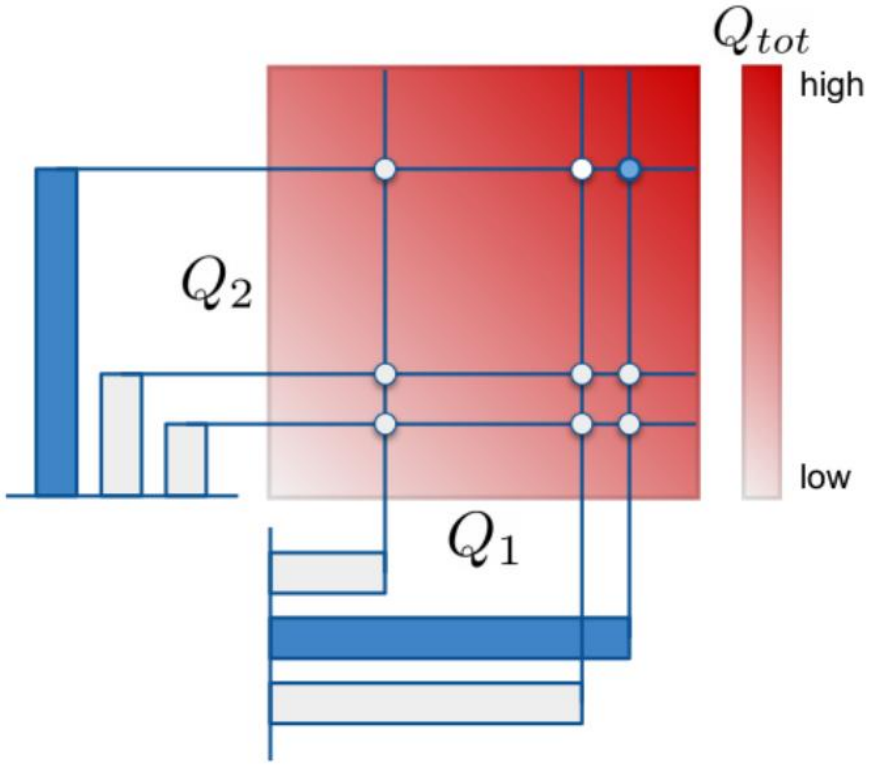
- 식(1)을 만족함으로써 최종적으로 식(2) 를 만족하게 되고 이러한 특성은 기여한 에이전트 모두 Q_{total} 을 통해 Q값의 업데이트가 가능함을 의미함

If...

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \ \forall a \in A, \ (1)$$

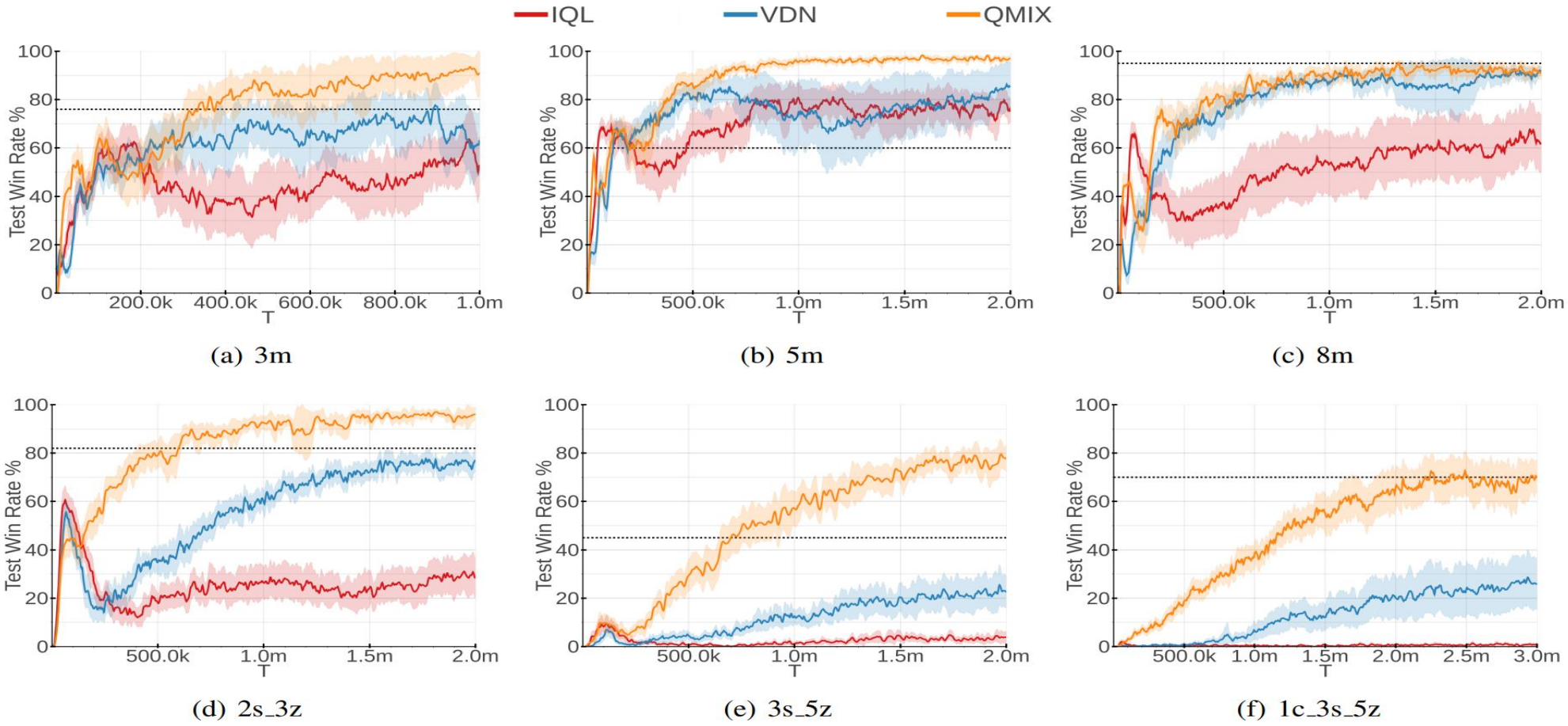
Then

$$\underset{\mathbf{u}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}. \ (2)$$



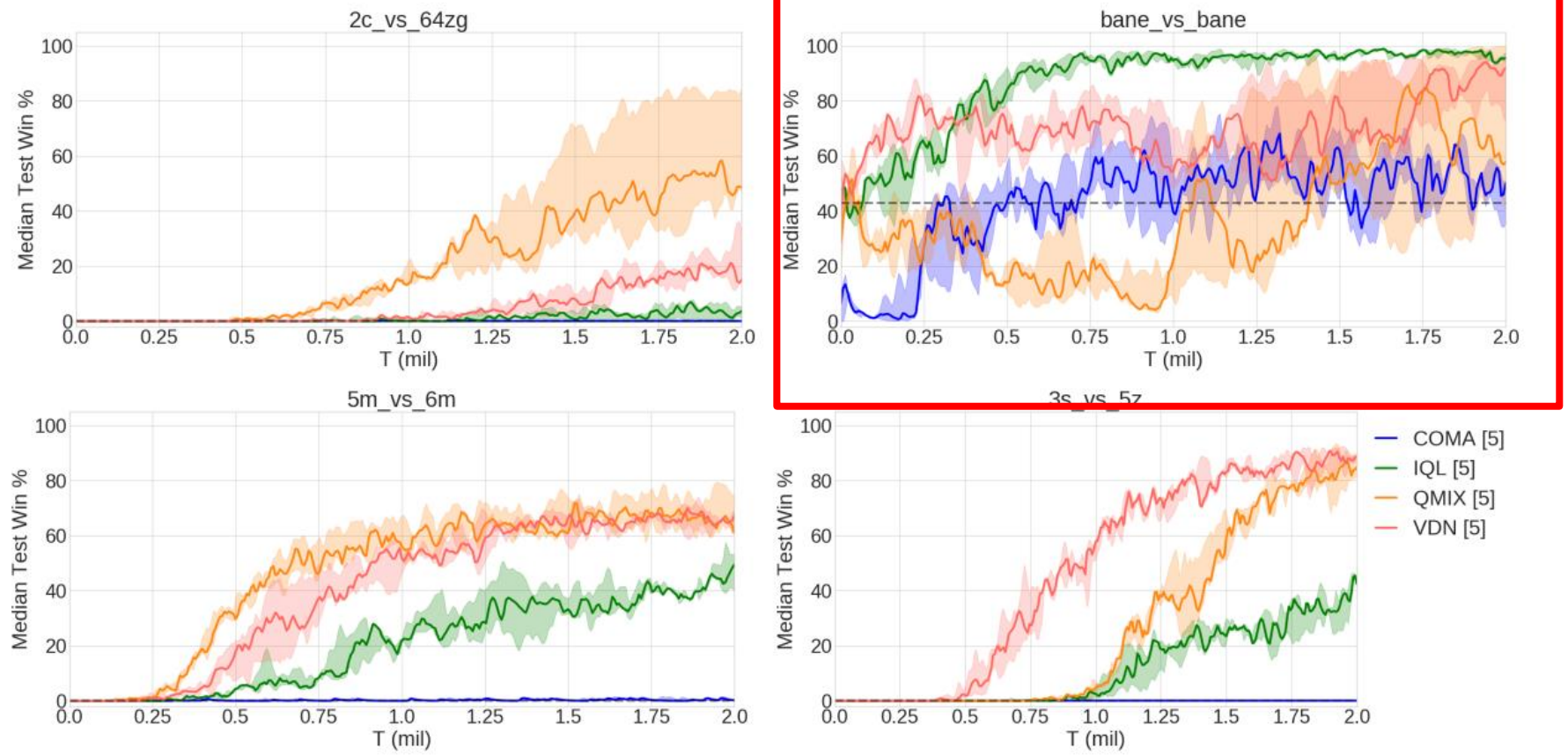
Result

- Starcraft 2 환경에서 각 유닛 별로 전투를 시켜 승리하도록 학습
- 모든 경우에서 Qmix 에서 가장 우수한 성능을 보임



Result

- Bane vs Bane 에서는 independent Q learning 이 가장 좋은 성능을 보임 (agent 수가 1 개)
- 이외에는 VDN 과 Qmix 가 우수한 성능을 보임



Conclusions

- 다른 멀티 에이전트 방식보다 Qmix 방식이 스타크래프트 2 환경에서 더 우수한 성능을 보임
- 특히 현재 가장 많이 쓰이는 방식은 VDN 과 COMA 방식을 능가했다는 점에서 의의가 있음
- 듀얼 스토커 프로젝트에서 Qmix 알고리즘 적용가능여부에 대해 고민하고자 함

Q & A