QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning

Tabish Rashid [*1] **Mikayel Samvelyan** [*2] **Christian Schroeder de Witt** [1]
Gregory Farquhar [1] **Jakob Foerster** [1] **Shimon Whiteson** [1]

[Submitted on 30 Mar 2018 (v1), last revised 6 Jun 2018 (this version, v2)]

발표자 : 박우성

RL-paper-study 10th

Centralized Learning

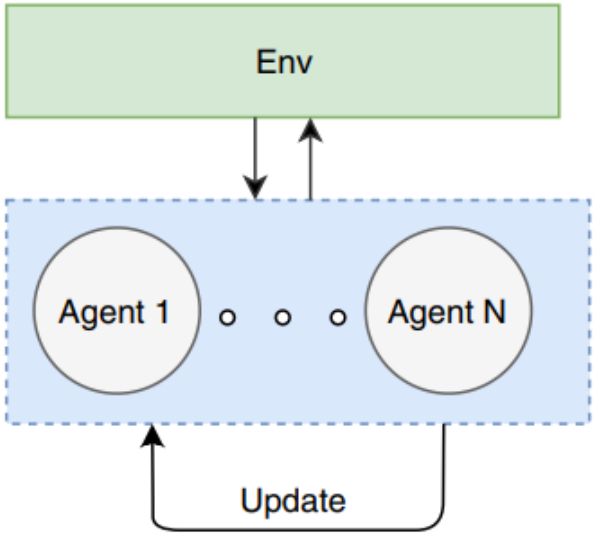


Figure 1

global state 정보를 사용할 수 있습니다.

communication constraints가 해제됩니다.

decentralised 정책을 추출하기 위한 최선의 전략이 불분명합니다.

이 경우 에이전트에게 숨겨진 추가 상태 정보에 대한 액세스 권한이 부여되는 경우가 많습니다.

에이전트에게 숨겨져 있고 에이전트 간 통신 제약을 제거합니다.

Decentralized Learning

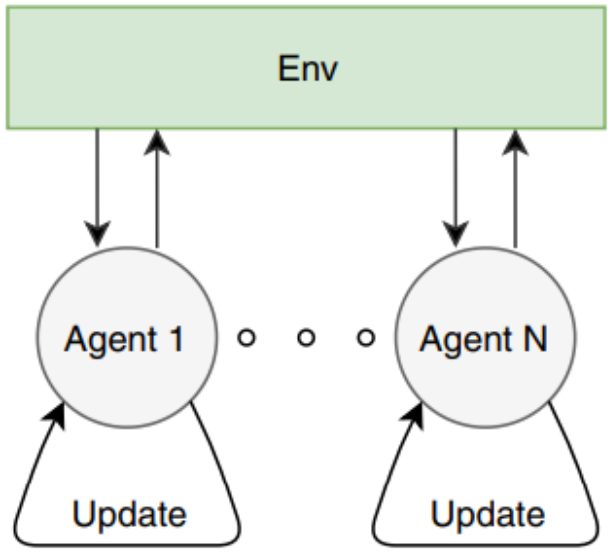


Figure 2

partial observability 및/또는 통신 제약으로 인해 각 에이전트의 로컬 작업 관찰 기록에만 의존하는 분산형 정책 학습이 필요합니다.

한편으로 에이전트 행동의 효과를 제대로 포착하려면 global state와 joint action을 조건으로 하는 centralised action-value 함수 Q_tot가 필요합니다.

반면에 이러한 함수는 에이전트가 많을 때 학습하기 어렵고, 학습할 수 있다고 하더라도 각 에이전트가 개별 관찰을 기반으로 개별 행동만을 선택할 수 있는 decentralised 정책을 추출할 수 있는 명확한 방법이 없습니다.

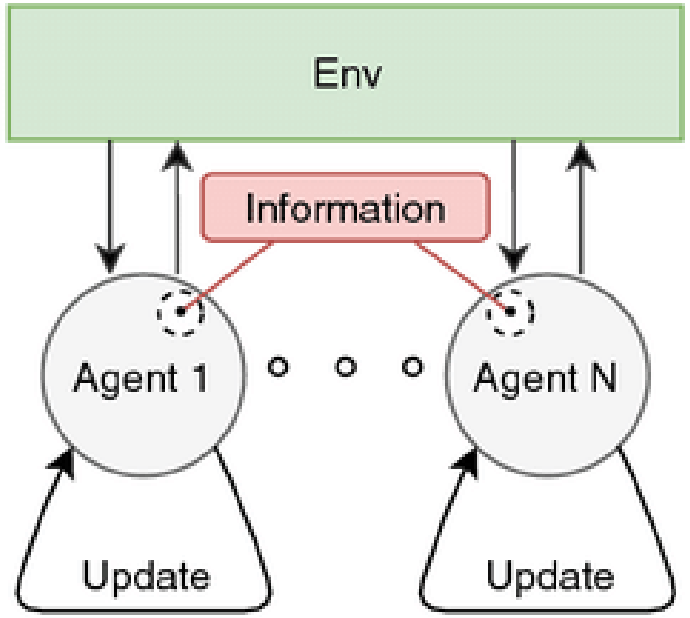


Figure 3

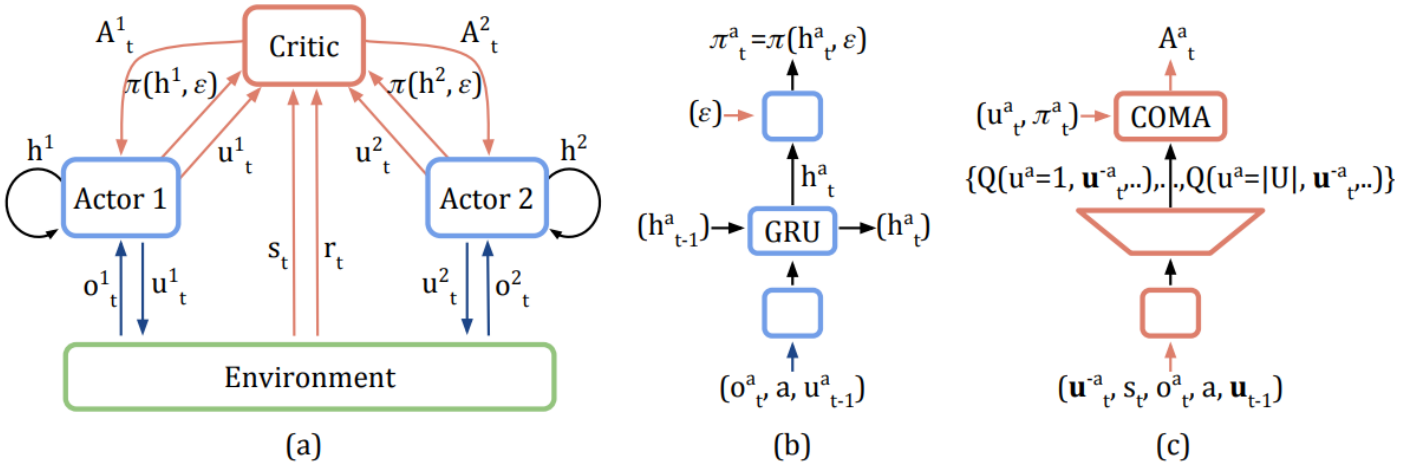


Figure 4

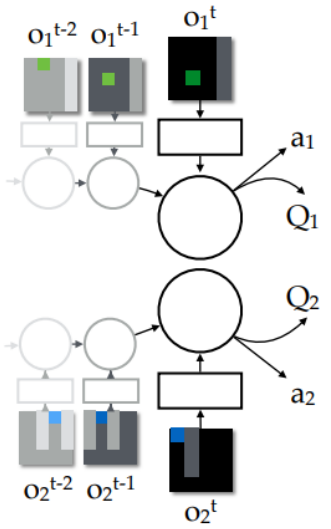


Figure 1: Independent agents architecture showing

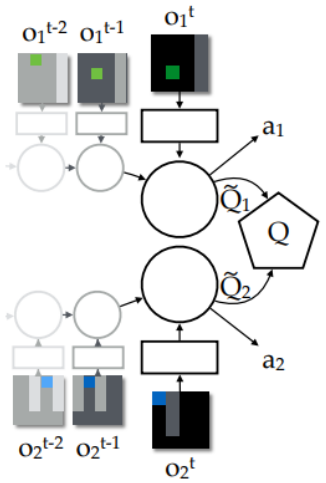


Figure 5

Figure 2: Value-decomposition individual architecture showing how local observations enter the networks of two agents over time (three steps shown),

Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 330–337, 1993.

그러나 이 접근 방식은 에이전트 간의 상호 작용을 명시적으로 나타낼 수 없으며

각 에이전트의 학습이 다른 에이전트의 학습 및 탐색과 혼동되기 때문에 수렴되지 않을 수 있습니다.

Counterfactual Multi-Agent Policy Gradients

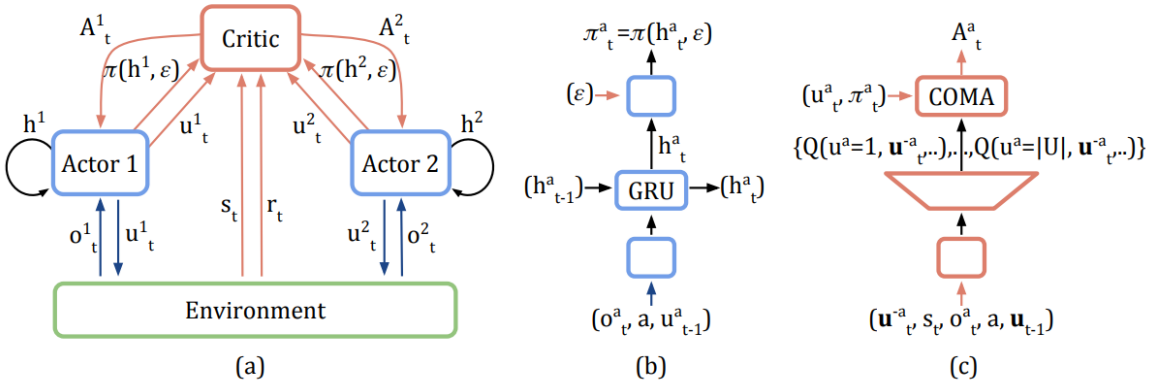


Figure 4

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

그러나 이를 위해서는 표본 비효율적일 수 있는 on-policy 학습이 필요하며,

에이전트가 소수가 아닌 경우 완전히 centralised critic를 훈련시키는 것은 매우 어렵다.

Value-Decomposition Networks For Cooperative Multi-Agent Learning

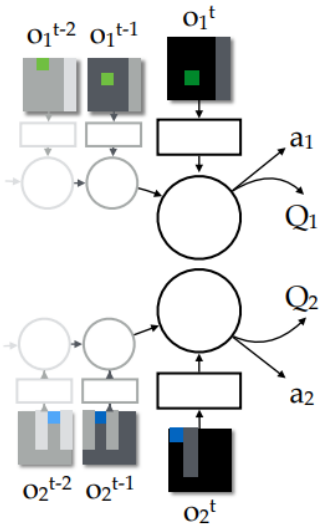


Figure 1: Independent agents architecture showing

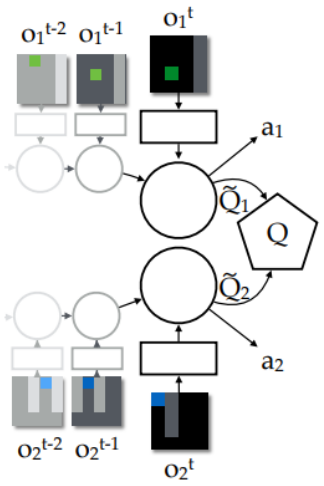


Figure 2: Value-decomposition individual architecture showing how local observations enter the networks of two agents over time (three steps shown),

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., 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 the 17th International Conference on Autonomous Agents and Multiagent Systems*, 2017.

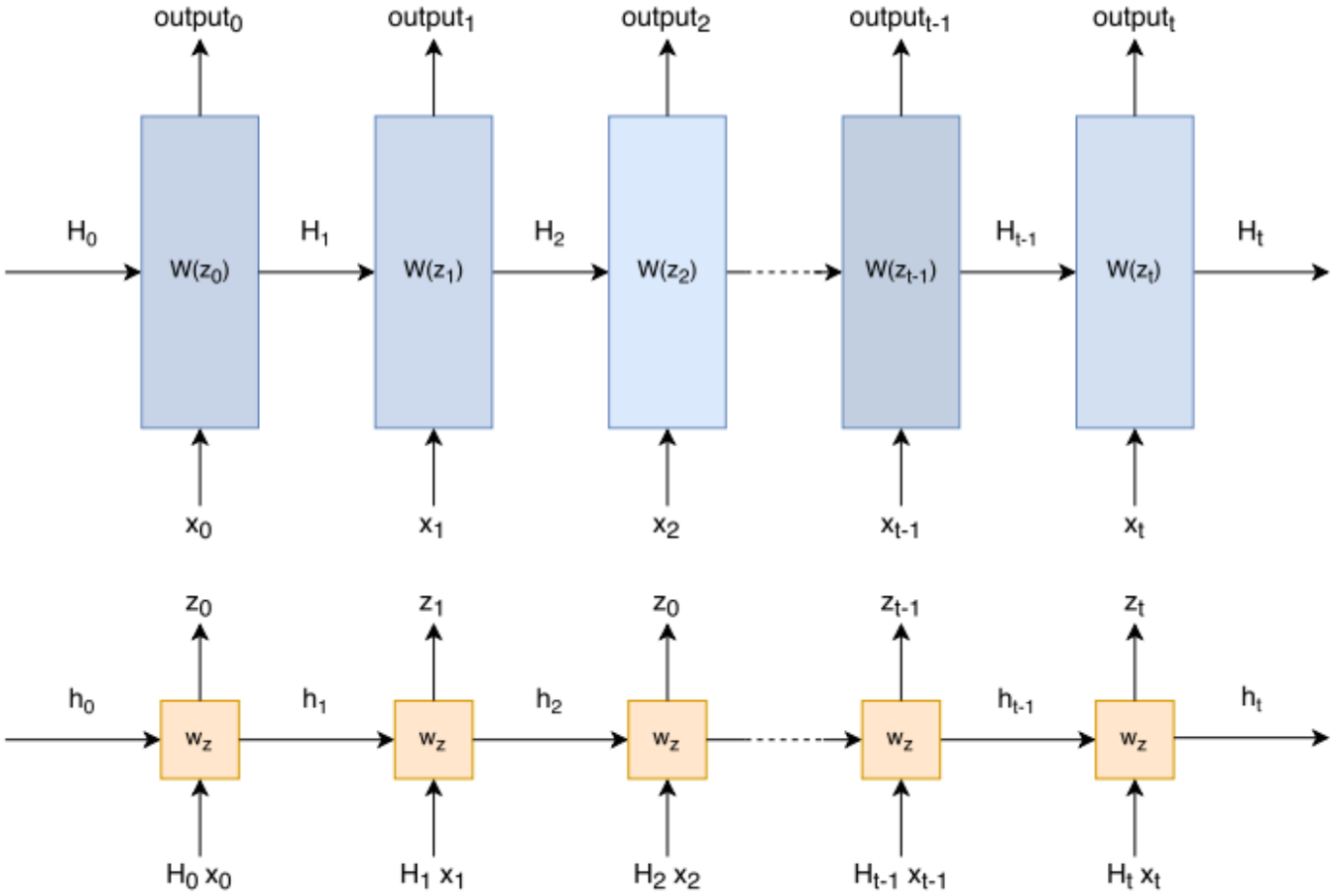
Figure 5

VDN은 표현할 수 있는 중앙 집중식 액션 값 함수의 복잡성을 심각하게 제한하고

훈련 중에 사용할 수 있는 추가 상태 정보를 무시한다.

HYPERNETWORKS

David Ha,* Andrew Dai, Quoc V. Le
Google Brain
{hadavid,adai,qvl}@google.com



QMIX는 신경망을 사용하여 centralised state를 다른 신경망의 가중치로 변환한다. (hypernetwork Ha et al., 2017)

이 두 번째 신경망은 가중치를 양수로 유지함으로써 입력에 대해 monotonic 하도록 제한된다.

(Dugas et al. (2009))은 신경망에 대한 이러한 기능적 제약을 조사합니다.

QMIX

$$\underset{\mathbf{u}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}. \tag{4}$$

환경이 적대적이지 않은 한, 전체 행동 관찰 이력을 기반으로 하는 결정론적 최적 정책이 존재합니다.

따라서 우리는 최적의 공동 행동 가치 함수를 기반으로 deterministic greedy decentralised policies 과 deterministic greedy centralised policy 간의 일관성을 확립하기만 하면 됩니다. 탐욕적인 탈중앙화 정책이 Qa에 대한 최대값에 의해 결정될 때, Qtot에 대해 수행된 글로벌 최대값이 각 Qa에 대해 수행된 개별 최대값 연산 집합과 동일한 결과를 산출하는 경우 일관성이 유지됩니다:

QMIX

$$\underset{\mathbf{u}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}. \tag{4}$$

이를 통해 각 에이전트 a는 자신의 Qa와 관련하여 탐욕스러운 행동을 선택하는 것만으로 탈중앙화 실행에 참여할 수 있습니다.

부수적으로 (4)가 만족되면, 정책 외 학습 업데이트에 필요한 Qtot의 최대값을 구하는 것은 기하급수적으로 많은 공동 행동에 대한 Qtot을 완전히 평가하지 않고도 간단하게 추적할 수 있습니다.

QMIX

$$\underset{\mathbf{u}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}. \tag{4}$$

VDN의 표현은 (4)를 만족시키기에 충분합니다. 그러나 QMIX는 이 표현이 (4)를 만족하는 더 큰 단조 함수군으로 일반화될 수 있다는 관찰에 기반합니다.

이 맥락에서 단조로움은 Qtot과 각 Qa 사이의 관계에 대한 제약 조건으로 정의됩니다:

QMIX

$$\underset{\mathbf{u}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}. \tag{4}$$

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \ \forall a \in A, \tag{5}$$

which is sufficient to satisfy (4), as the following theorem shows.

Theorem 1 *If* $\forall a \in A \equiv \{1, 2, ..., n\}$, $\frac{\partial Q_{tot}}{\partial Q_a} \geq 0$ *then*

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

Appendix A. Monotonicity

Theorem 2 *If* $\forall a \in A \equiv \{1, 2, ..., n\}$, $\frac{\partial Q_{tot}}{\partial Q_a} \geq 0$ *then*

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

Proof Since $\frac{\partial Q_{tot}}{\partial Q_a} \geq 0$ for $\forall a \in A$, the following holds for any $(u^1, \dots, u^n)$ and the *mixing network* function $Q_{tot}(\cdot)$ with n arguments:

$$\begin{aligned}
&Q_{tot}(Q_1(\tau^1, u^1), \dots, Q_a(\tau^a, u^a), \dots, Q_n(\tau^n, u^n)) \\
&\leq Q_{tot}(\max_{u^1} Q_1(\tau^1, u^1), \dots, Q_a(\tau^a, u^a), \dots, Q_n(\tau^n, u^n)) \\
&\dots \\
&\leq Q_{tot}(\max_{u^1} Q_1(\tau^1, u^1), \dots, \max_{u^a} Q_a(\tau^a, u^a), \dots, Q_n(\tau^n, u^n)) \\
&\dots \\
&\leq Q_{tot}(\max_{u^1} Q_1(\tau^1, u^1), \dots, \max_{u^a} Q_a(\tau^a, u^a), \dots, \max_{u^n} Q_n(\tau^n, u^n)).
\end{aligned}$$

Therefore, the maximiser of the mixing network function is:

$$(\max_{u^1} Q_1(\tau^1, u^1), \dots, \max_{u^a} Q_a(\tau^a, u^a), \dots, \max_{u^n} Q_n(\tau^n, u^n)).$$

Thus,

$$\begin{aligned}
\max_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) &\coloneqq \max_{\mathbf{u}=(u^1,\dots,u^n)} Q_{tot}(Q_1(\tau^1, u^1), \dots, Q_n(\tau^n, u^n)) \\
&= Q_{tot}(\max_{u^1} Q_1(\tau^1, u^1), \dots, \max_{u^n} Q_n(\tau^n, u^n)).
\end{aligned}$$

Letting $\mathbf{u}_* = (u^1_*, \dots, u^n_*) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}$, we have that:

$$\begin{aligned}
Q_{tot}(Q_1(\tau^1, u^1_*), \dots, Q_n(\tau^n, u^n_*)) &= Q_{tot}(\max_{u^1} Q_1(\tau^1, u^1), \dots, \max_{u^n} Q_n(\tau^n, u^n)) \\
&= \max_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u})
\end{aligned}$$

Hence, $\mathbf{u}_* = \operatorname{argmax}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u})$, which proves (4). ■

QMIX

Figure 6

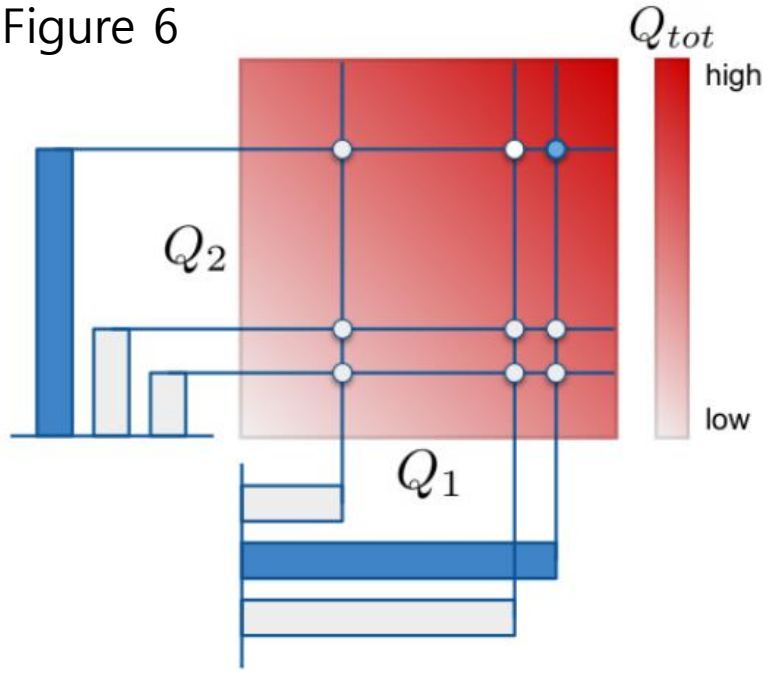


Figure 2: The discrete per-agent action-value scores Q_a are fed into the monotonic function $Q_{tot}(Q_1, Q_2)$. The maximum Q_a for each agent is shown in blue, which corresponds to the maximum Q_{tot} also shown in blue. The constraint (4) is satisfied due to the monotonicity of Q_{tot}.

그림 2는 세 가지 가능한 행동을 하는 두 명의 에이전트가 있는 예시에서 Qtot과 개별 Qa 함수 간의 관계와 단조로움이 어떻게 분산 가능한 인수 최대값으로 이어지는지 설명합니다.

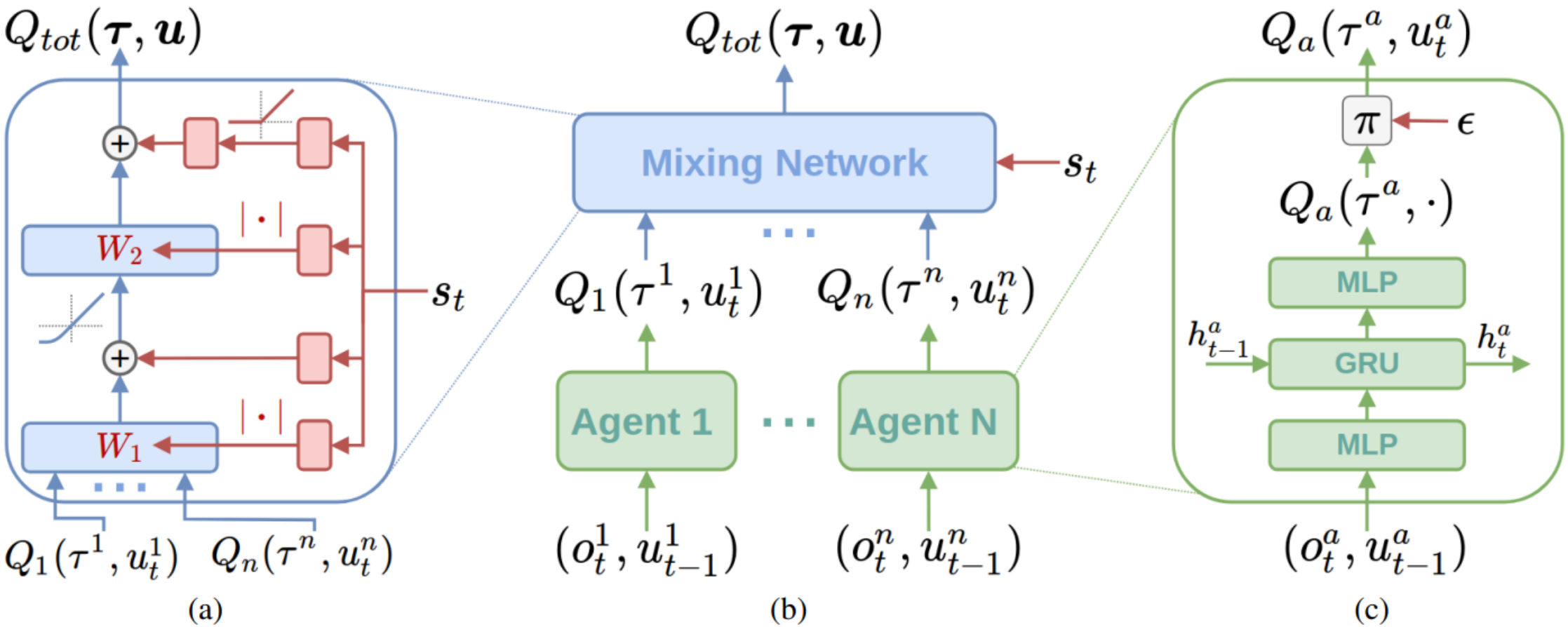


Figure 2. (a) Mixing network structure. In red are the hypernetworks that produce the weights and biases for mixing network layers shown in blue. (b) The overall QMIX architecture. (c) Agent network structure. Best viewed in colour.

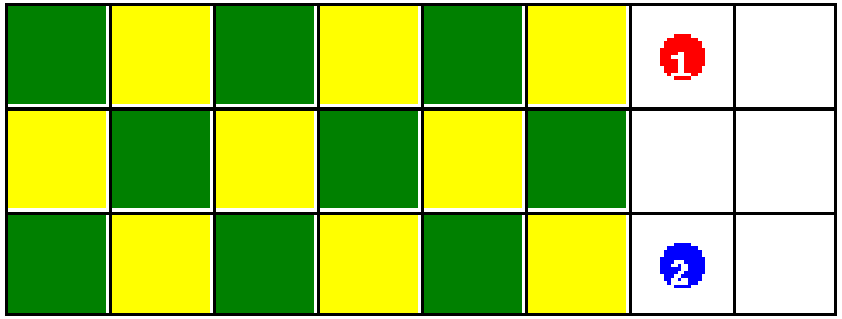
Environment

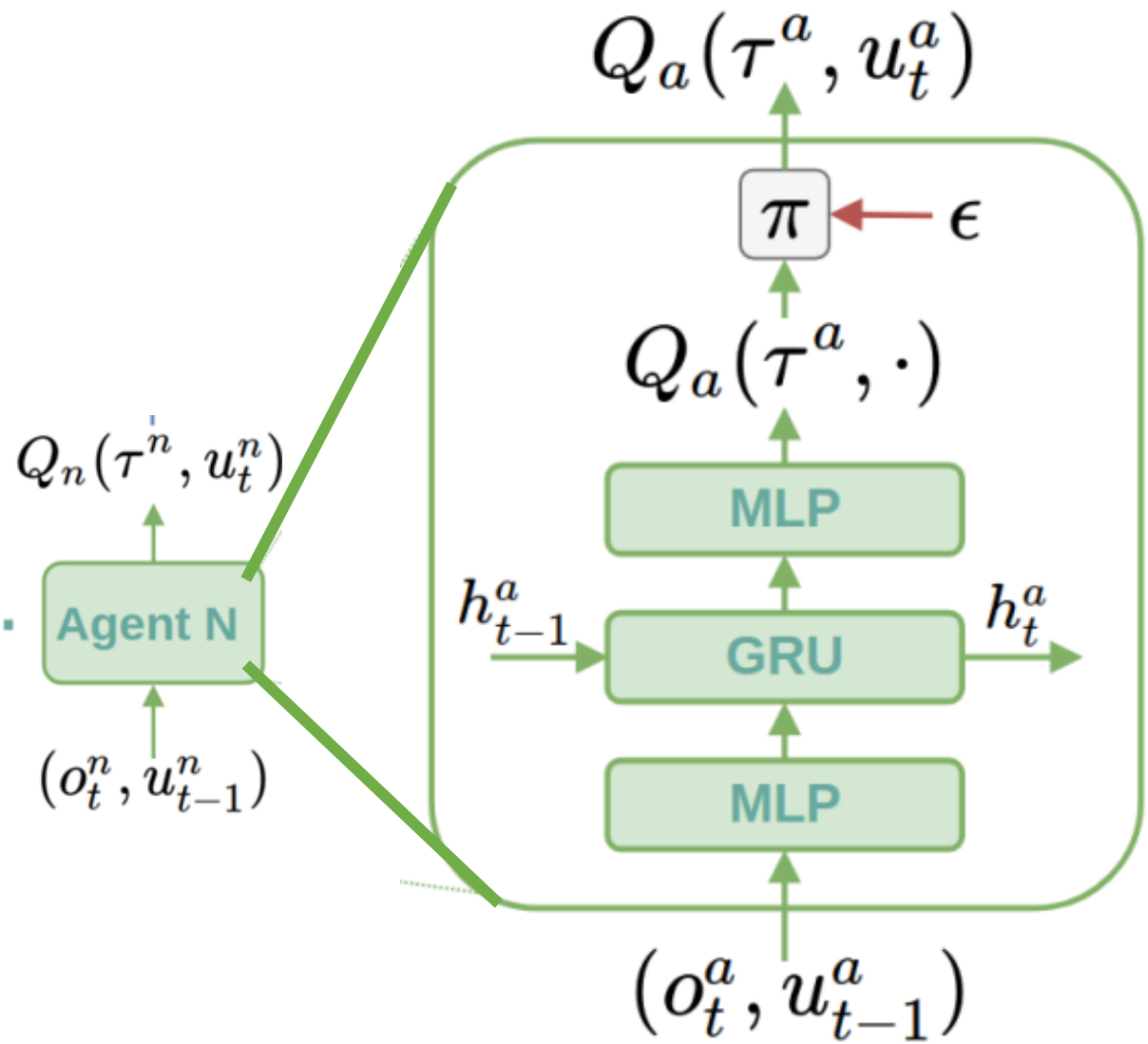
Checkers

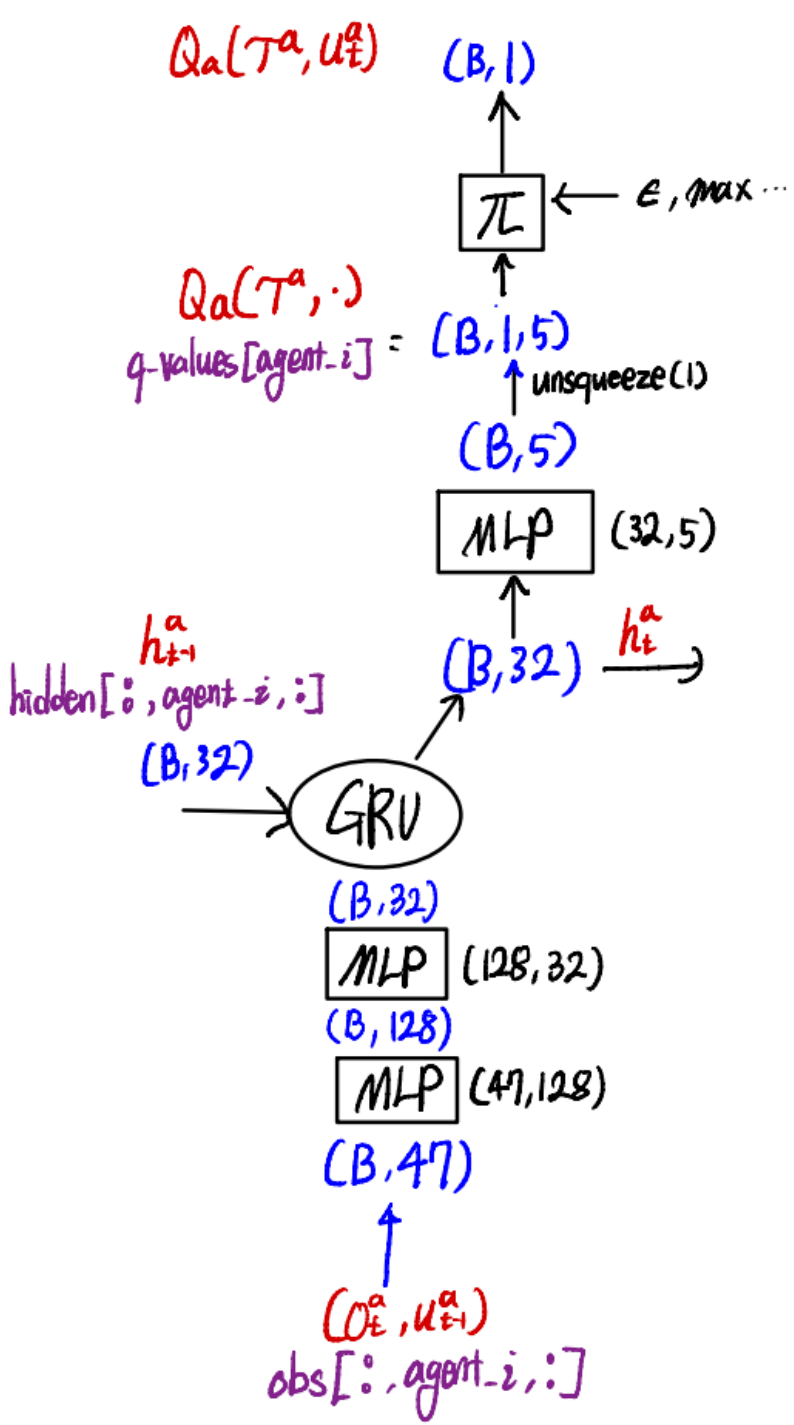
The map contains apples and lemons. The first player (red) is very sensitive and scores 10 for the team for an apple (green square) and −10 for a lemon (yellow square). The second (blue), less sensitive player scores 1 for the team for an apple and −1 for a lemon. There is a wall of lemons between the players and the apples. Apples and lemons disappear when collected, and the environment resets when all apples are eaten. It is important that the sensitive agent eats the apples while the less sensitive agent should leave them to its teammate but clear the way by eating obstructing lemons.

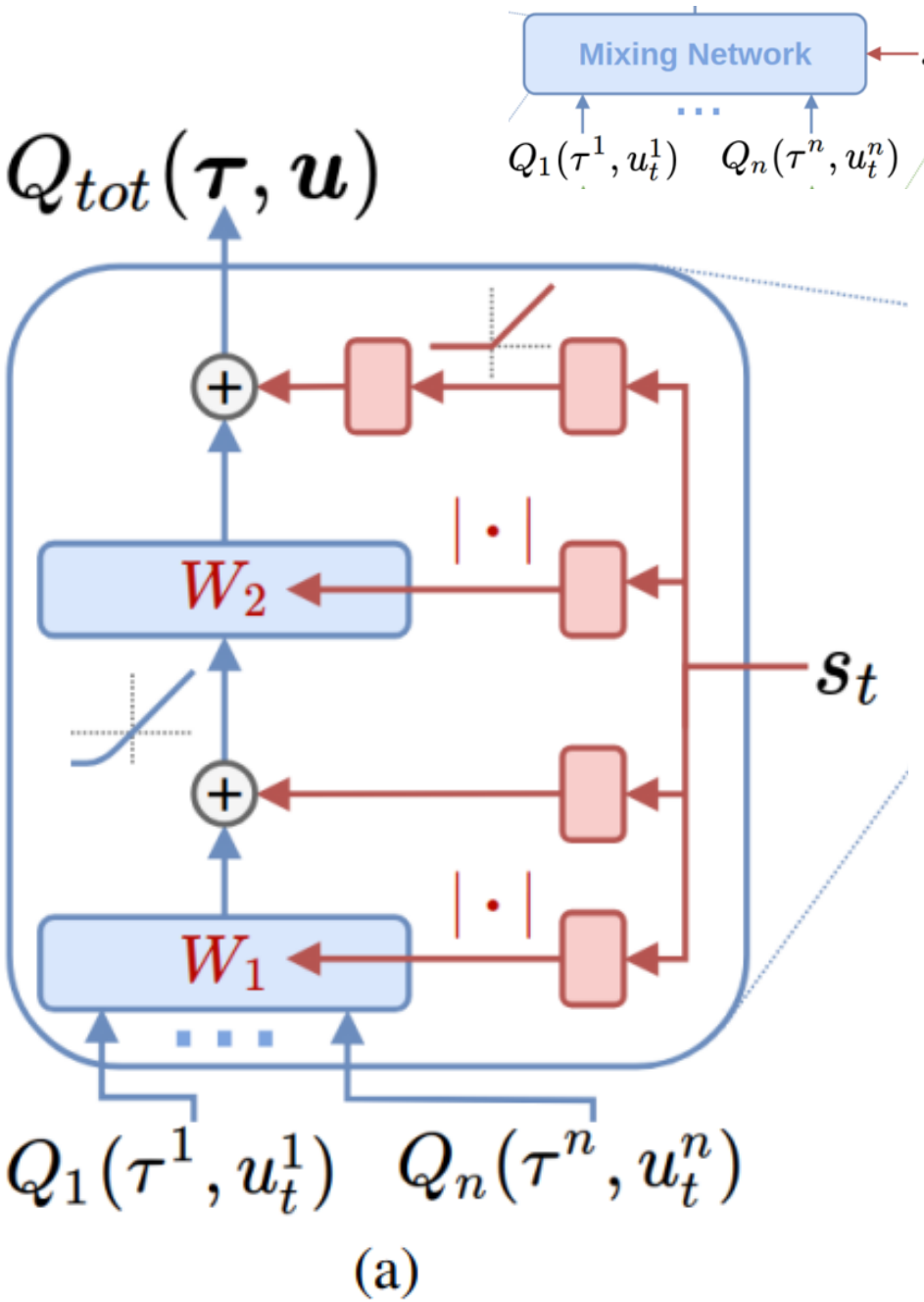
- Reference Paper : Value-Decomposition Networks For Cooperative Multi-Agent Learning (Section 4.2)
- Action Space: `0: Down, 1: Left, 2: Up , 3: Right, 4: Noop`
- Agent Observation : `Agent Coordinate + 3x3 mask around the agent +  Steps in env.`
- Best Score: `NA`

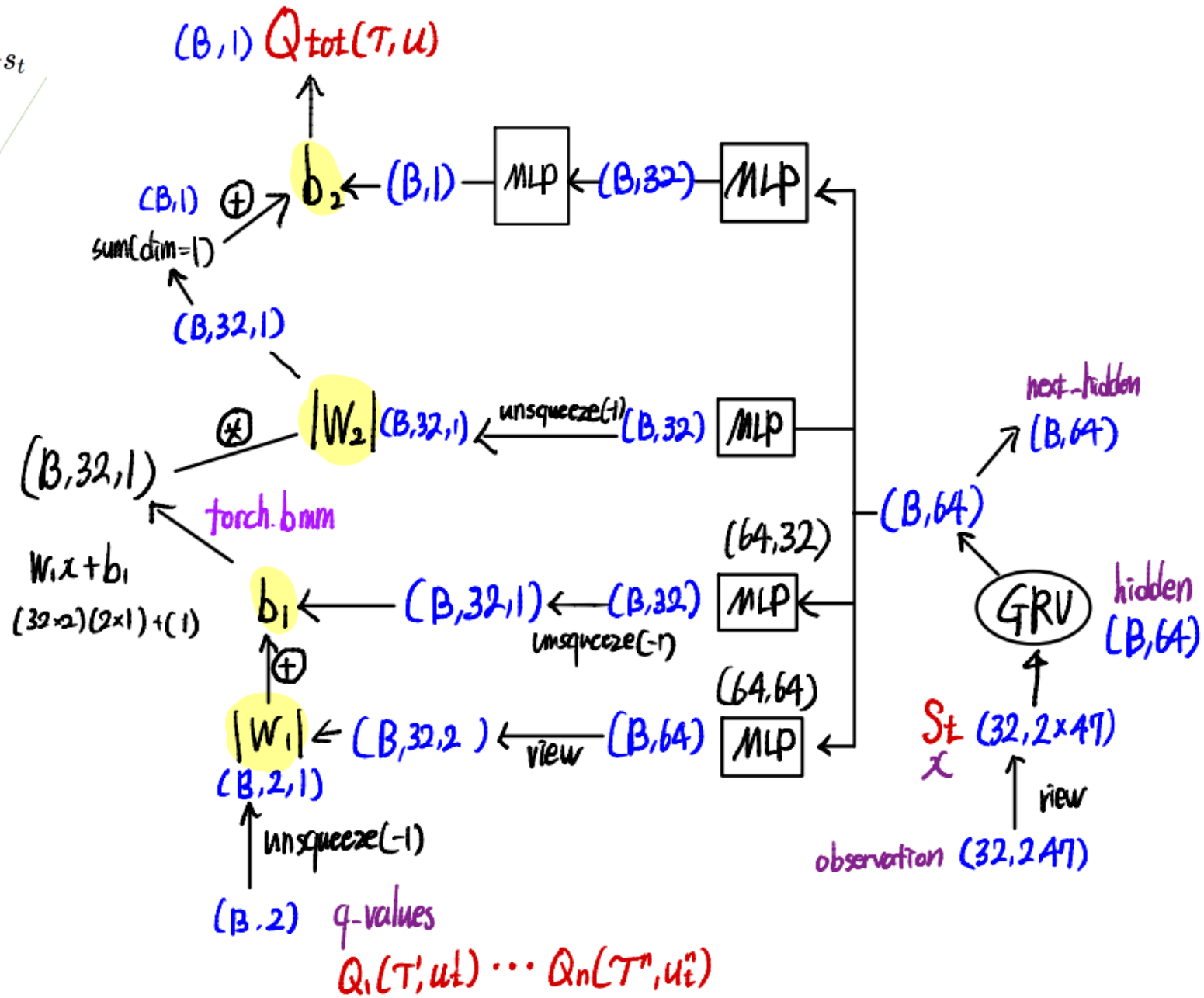
Figure 7




$Q_a(\tau^a, u_t^a)$
π
ϵ
$Q_a(\tau^a, \cdot)$
MLP
h_{t-1}^a
GRU
h_t^a
MLP
$Q_n(\tau^n, u_t^n)$
Agent N
(o_t^n, u_{t-1}^n)
(o_t^a, u_{t-1}^a)


$Q_a(\tau^a, u_t^a)$
(B,1)
π
ϵ, max ···
$Q_a(\tau^a, \cdot)$
q_values[agent_i] =
(B,1,5)
unsqueeze(1)
(B,5)
MLP
(32,5)
h_{t-1}^a
hidden[:, agent_i, :]
(B,32)
h_t^a
(B,32)
GRU
(B,32)
MLP
(128,32)
(B,128)
MLP
(47,128)
(B,47)
(o_t^a, u_{t-1}^a)
obs[:, agent_i, :]


Mixing Network
s_t
$Q_1(\tau^1, u_t^1)$ $Q_n(\tau^n, u_t^n)$
$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$
W_2
W_1
s_t
$Q_1(\tau^1, u_t^1)$ $Q_n(\tau^n, u_t^n)$
(a)


(B,1) Qtot(τ,u)
b2
(B,1)
MLP
(B,32)
MLP
(B,1)
sum(dim=1)
(B,32,1)
|W2|
(B,32,1)
unsqueeze(-1)
(B,32)
MLP
(B,32,1)
torch.bmm
W1x+b1
(32×2)(2×1)+(1)
b1
(B,32,1)
(B,32)
unsqueeze(-1)
MLP
(64,32)
(B,64)
next_hidden
(B,64)
GRU
hidden
(B,64)
|W1|
(B,32,2)
view
(B,64)
MLP
(64,64)
St
x
(32,2×47)
view
observation (32,247)
(B,2,1)
unsqueeze(-1)
(B,2) q-values
Q1(τ1,u1t) ··· Qn(τn,unt)

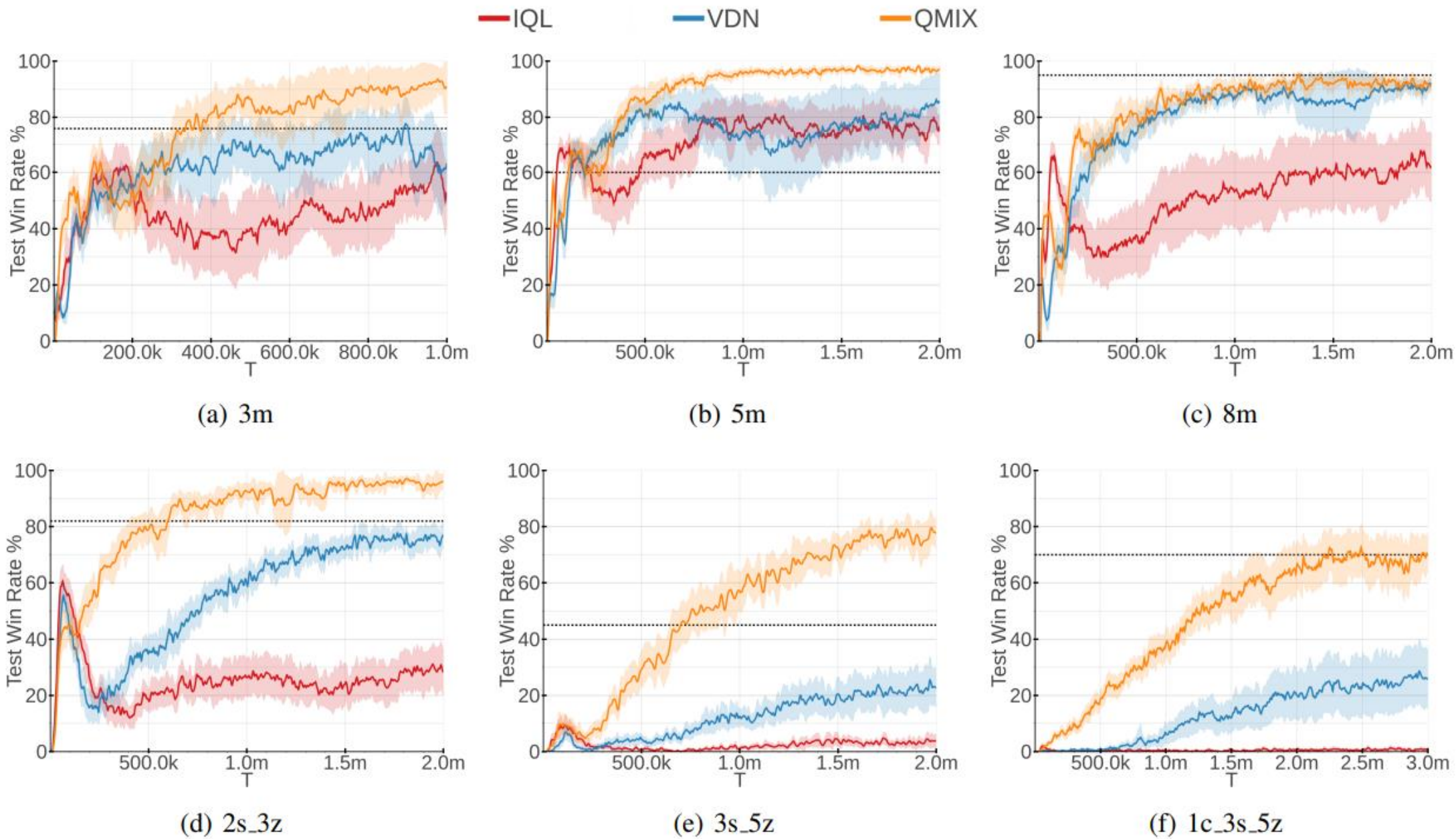
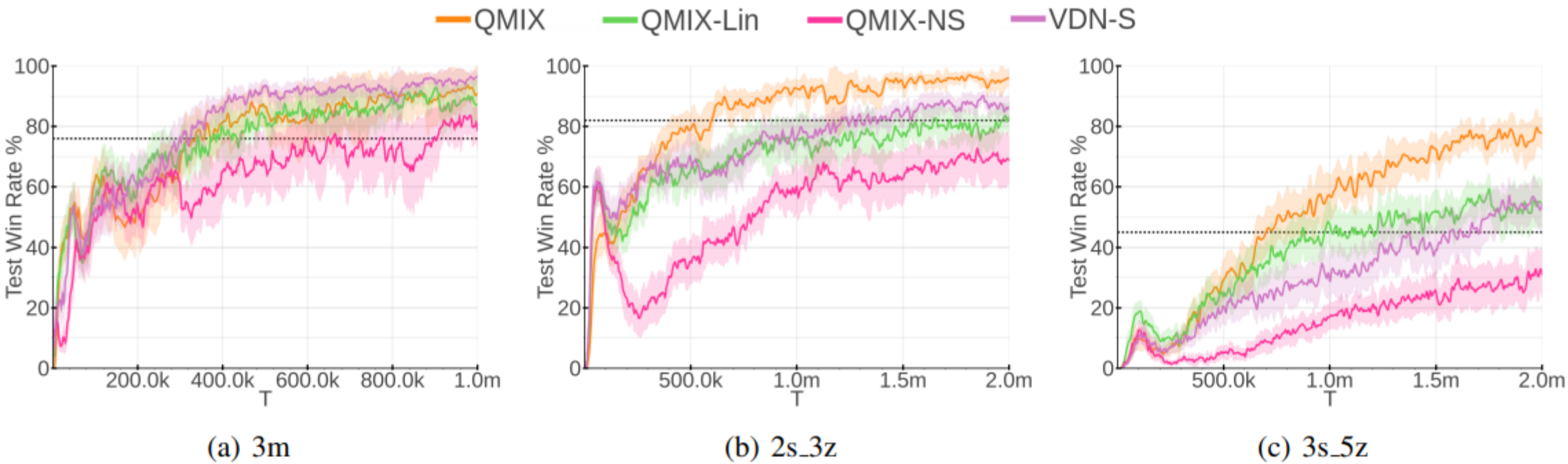


Figure 3. Win rates for IQL, VDN, and QMIX on six different combat maps. The performance of the heuristic-based algorithm is shown as a dashed line.



(a) 3m (b) 2s_3z (c) 3s_5z

Figure 4. Win rates for QMIX and ablations on 3m, 2s_3z and 3s_5z maps.

Figure 1 / Figure 2 / Figure 3
Multi-agent deep reinforcement learning: a survey - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Training-schemes-in-the-multi-agent-setting-Left-CTCE-holds-a-joint-policy-for-all_fig2_350893932 [accessed 4 Apr, 2023]

Figure 4
Counterfactual Multi-Agent Policy Gradients
Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, Shimon Whiteson

Figure 5
Value-Decomposition Networks For Cooperative Multi-Agent Learning
Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, Thore Graepel

Figure 6

Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning
Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, Shimon Whiteson

Figure 7

https://github.com/koulanurag/ma-gym/wiki/Environments#Checkers