KOREA UNIVERSITY 1905

Artificial Intelligence and Mobility Lab

Game AI(StarCraftII)

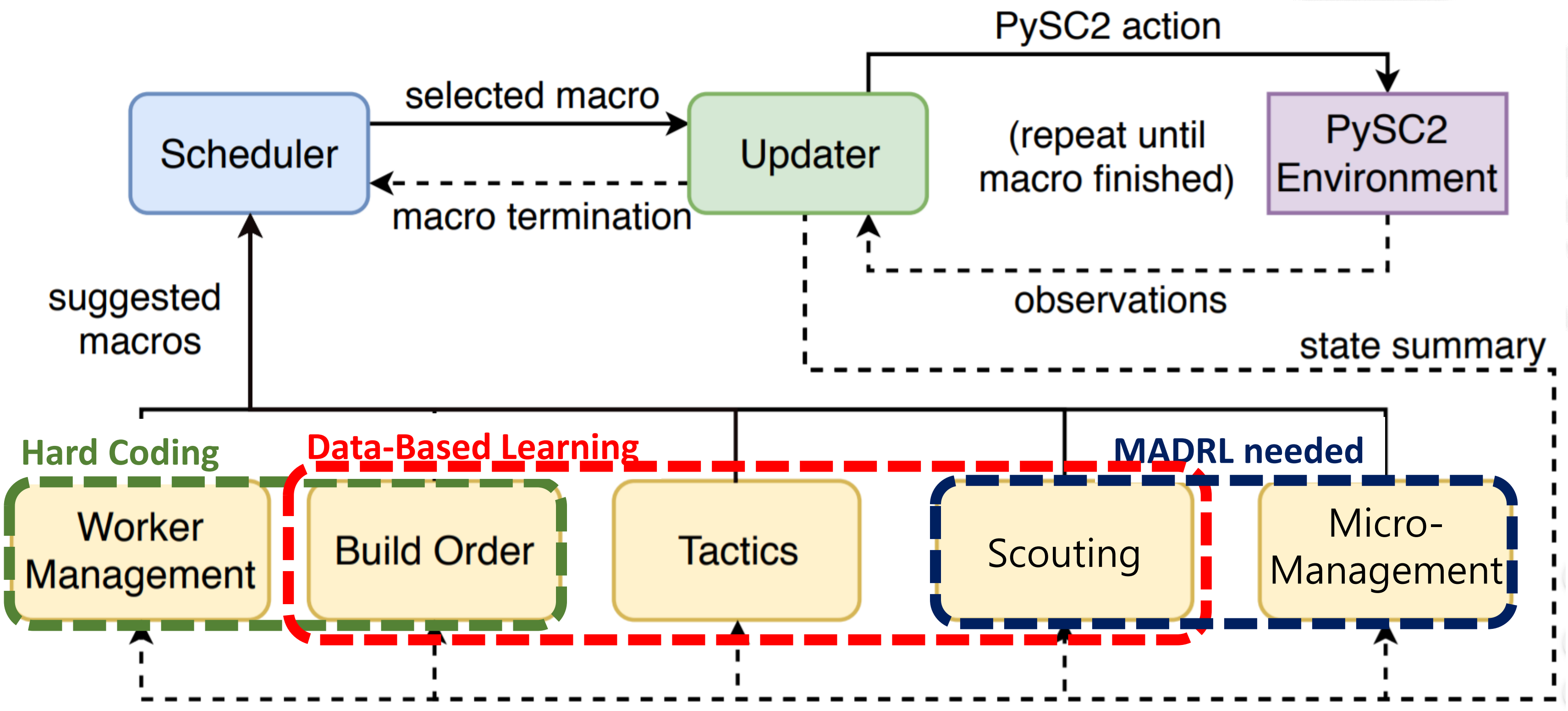
:Implementation of G2ANet and other algorithms

Won Joon Yun

Korea University, School of Electrical Engineering

ywjoon95@korea.ac.kr

- How to approach StarCraftII with artificial intelligence
  - **Macro-management**: Macro-management corresponds to high-level strategic considerations that solve problems such as build, tech tree, resource allocation, and reconnaissance using collected information (resource, population, information of enemy, *etc.*).
  - **Micro-management**: It is to operate single or multiple units. This is the battle scenario.



D. Lee et al, *Modular Architecture for StarCraft II with Deep Reinforcement Learning*, AAAI 2018

- Libraries
  - PySC2 provided by DeepMind
    - Provides environment applicable to StarCraft such as Action Space and State Space.
    - Information can be extracted from mini-maps, screens, etc.
  - S2Client-Proto provided by Blizzard
    - Library that can transmit/receive protocols sent from PySC2
  - SMAC
    - Provides various environments(i.e., Map) for multi-agent reinforcement learning
- Windows/Linux all applicable!

PySC2 (Deepmind)

- It provides same environment as the actual StarCraft. Also it is a python library that can execute all StarCraft commands such as observation of game screen or minimap, troop, attack, and reconnaissance.
- It also provides a state space and action space for states and actions that can be done within Starcraft, which is good for reinforcement learning.
- The environment provided by PySC2 are mini games (for battle, reconnaissance, production, and build), which are essential conditions for winning the game.

- List of mini games provided by PySC2
  - CollectMineralsAndGas: Environment for efficient resource extraction
  - BuildMarines: the environment for production
  - MoveToBeacon: environment to move to destination
  - FindAndDefeatZerglings: Recon and battle environment
  - DefeatZerglingsAndBanelings: The environment in battle

- A library that delivers commands from PySC2 to StarCraftII and also delivers events from StarCraftII to PySC2.
- In StarCraftII, users can also play games with RL Agent. (Supports all functions of StarCraftII)
- It supports mapping between StarCraftII and PySC2 in real time.

- Supports multi-agent reinforcement learning in combat scenarios using the two libraries described before (Pysc2, S2client-proto).
- supports 23 maps of various battle scenarios.

M. Samvelyan, *The StarCraft Multi-Agent Challenge,* AAMAS 2019, May 13-17, 2019, Montréal, Canada

- State(Observation) : $O = \{o_t^1, o_t^2, \ldots, o_t^k, \ldots, o_t^K\}$, (K: Number of Agents)

- o_t^k Configuration

> - Agent movement and location characteristics (where to go, height characteristics, path)
> - Enemy characteristics (whether the enemy can attack the agent, the enemy's health, enemy's x coordinates, enemy's y coordinates, shield amount and unit type)
> - Allies' characteristics (allies' health, allies' x coordinate, allies' y coordinate, shield amount, unit type"
> - Unique characteristics of the agent (current health, shield amount, unit type)

SMAC – Action Space

- Action: $A = \{a_t^1, a_t^2, \dots, a_t^k, \dots, a_t^K\}$, (K: Number of Agents)

- a_t^k Configuration

> - Stop
> - Select one of four directions (*i.g.,* N / S / E / W) to move
> - Select target to attack and attack
> - Choosing whom companions to heal

- Reward: $R = \{r_t^1, r_t^2, \ldots, r_t^k, \ldots, r_t^K\}$, (K: Number of Agents)

- r_t^k Configuration

> - **Positive reward [+]:** A positive reward in proportion to the remaining stamina and energy remaining after destroying the agent
> - **Negative reward [-]:** negative reward when agent dies
> - **Common:** Positive reward for less time, positive reward for victory, negative reward for defeat

SMAC – Agent / Enemy / Time Steps per Map

Map Name	Number of Agents	Number of Enemies	Time Step
3m	3	3	60
8m	8	8	120
25m	25	25	150
5m_vs_6m	5	6	70
8m_vs_9m	8	9	120
10m_vs_11m	10	11	150
27m_vs_30m	27	30	180
MMM	10	10	150
MMM2	10	12	180
2s3z	5	5	120
3s5z	8	8	150
3s5z_vs_3s6z	8	9	170
3s_vs_3z	3	3	150
3s_vs_4z	3	4	200
3s_vs_5z	3	5	250
1c3s5z	9	9	180
2m_vs_1z	2	1	150
corridor	6	24	400
6h_vs_8z	6	8	150
2s_vs_1sc	2	1	300
so_many_baneling	7	32	100
bane_vs_bane	24	24	200
2c_vs_64zg	2	64	400

Name	Ally Units	Enemy Units	Type	Challenge
5m_vs_6m	5 Marines	6 Marines	Asymmetric, Homogeneous	Focusing fire
3s_vs_5z	3 Stalkers	5 Zealots	Asymmetric, Heterogeneous	Kite enemy
2c_vs_64zg	2 Colossi	64 Zerglings	Asymmetric, Heterogeneous	Large action space
bane_vs_bane	4 Banelings, 20 Zerglings	4 Banelings, 20 Zerglings	Symmetric, Heterogeneous	Baneling blasts properly
3s5z_vs_3s6z	3 Stalkers, 5 Zealots	3 Stalkers, 6 Zealots	Asymmetric, Heterogeneous	Medivac absorbs fire
MMM2	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 3 Marauders, 8 Marines	Asymmetric, Heterogeneous	Circuitous tactics

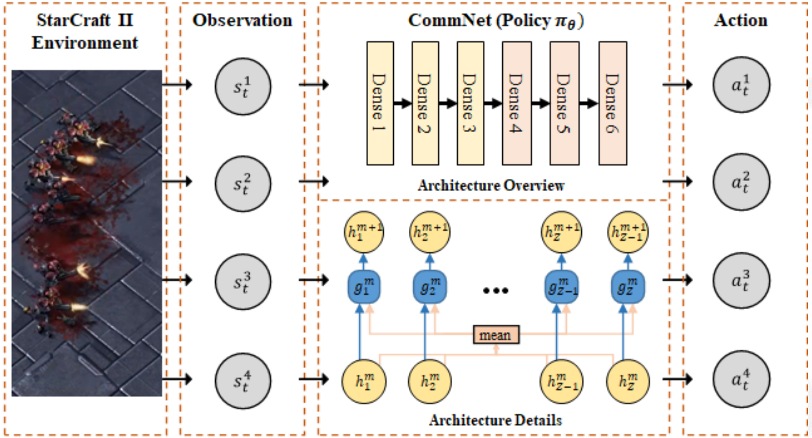
https://youtu.be/VZ7zmQ_obZ0

Battle Scenario: Learning Patterns with different algorithm

- In the case of COMA, it shows similar patterns to those learned with single agent.
- In the case of QMIX, agents communicate well with each other.

1. Communication problem
   - If the agent's strategies do not influence each other, it can be like a traditional single agent DRL.
2. Large-scale problem(Massive Agent Problem)
   - If N single agents capable of performing M actions exist, the dimension of the action value function to be dealt with increases exponentially to N^M.

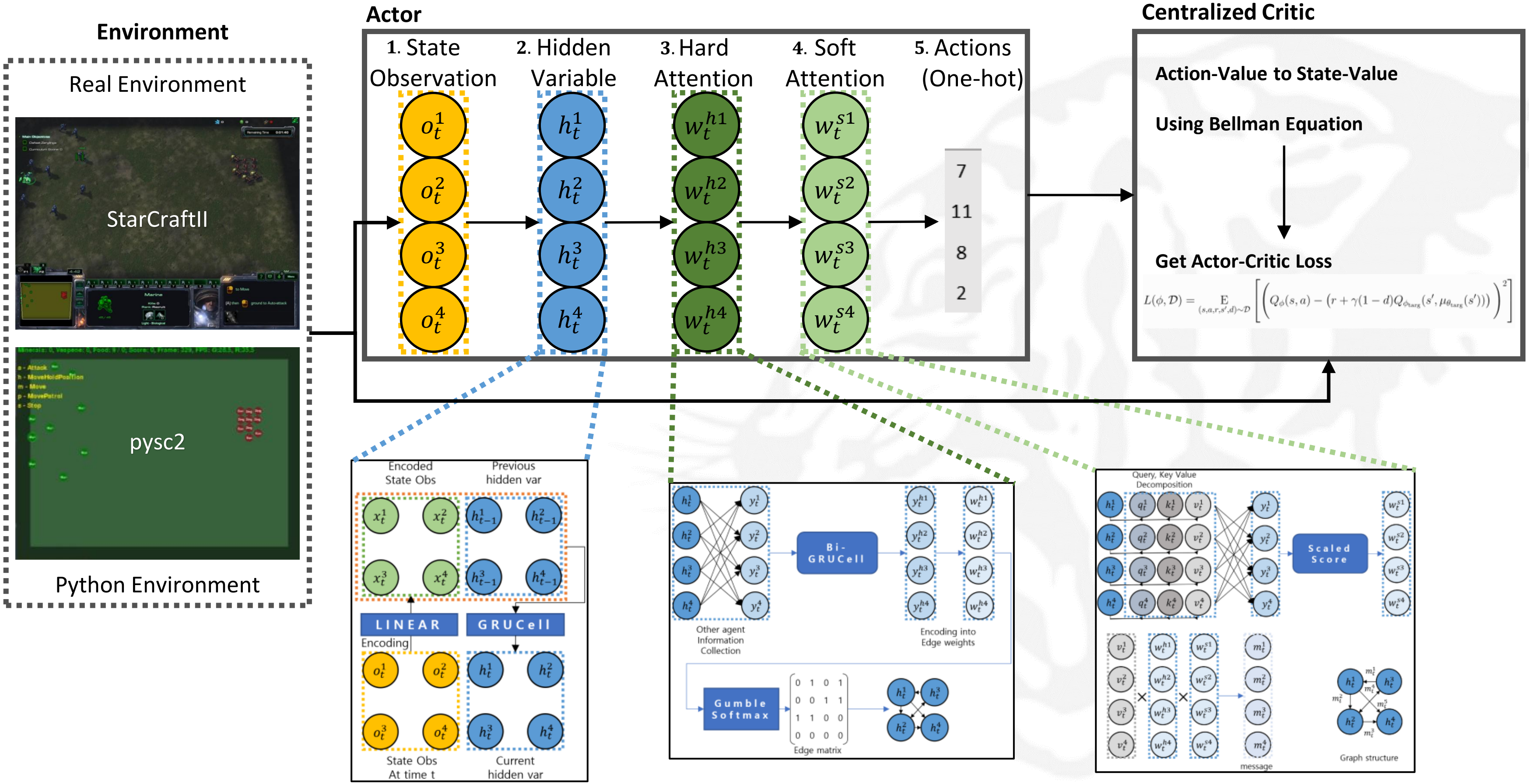
(Solution) Value Decomposition Network [VDN, QMIX, QTRAN, Qatten]



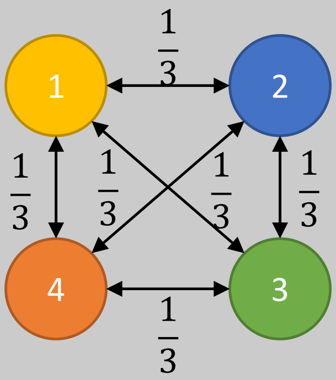
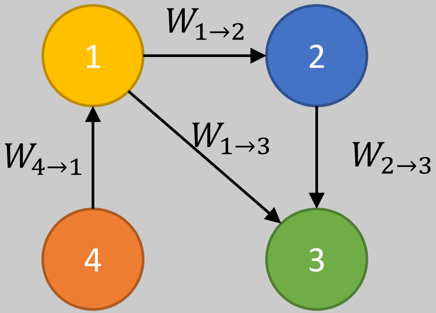
- Features 1. Allows sharing of elements between Hidden Variables within the policy.
- Features 2. In a policy neural network, Mean is a computing tool for collaboration.
  - ➢ Lost information exists because only means operation is taken

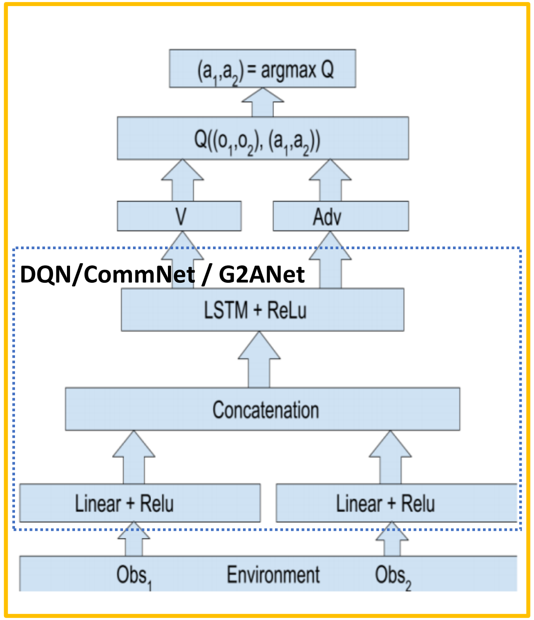
Learning Multiagent Communication with Backpropagation NIPS, 2016

Solution for Communication: *G2ANet*

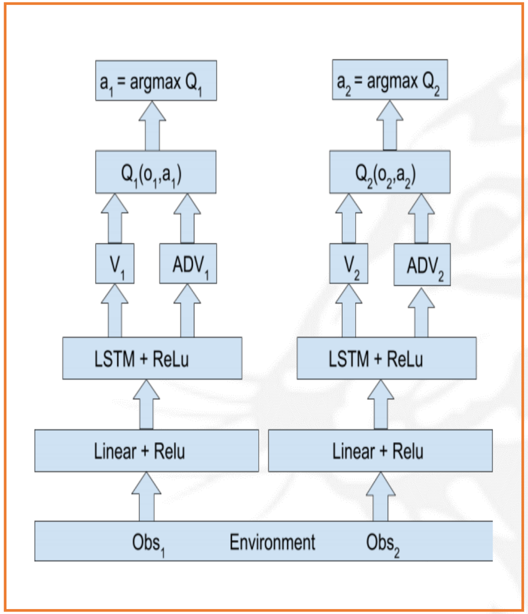


Performance Evaluation: *CommNet* versus. *G2ANet*

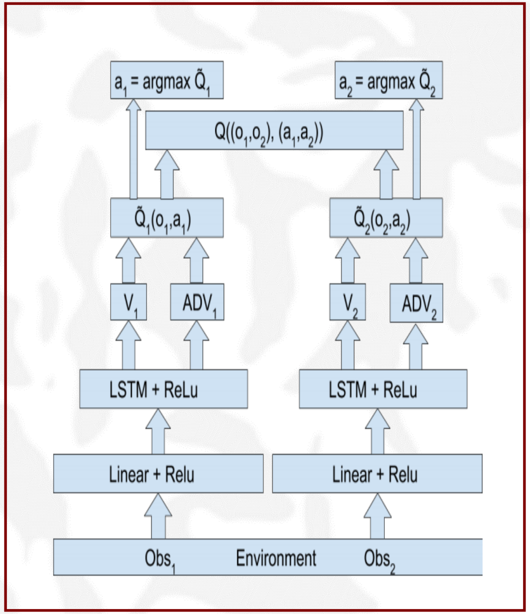
CommNet		G2ANet
		
Mean operation for all agents.	<	Select agent to communicate via hard attention
Cannot control the weight of message	<	Can control the weight of message via soft attention
Fast	>	Slow(complex neural network)



Combinatorially Centralized Architecture

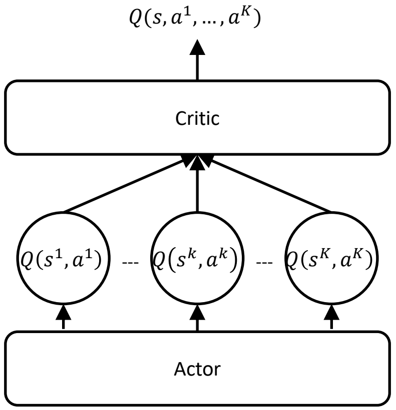


Independent Agents Architecture (IQL)

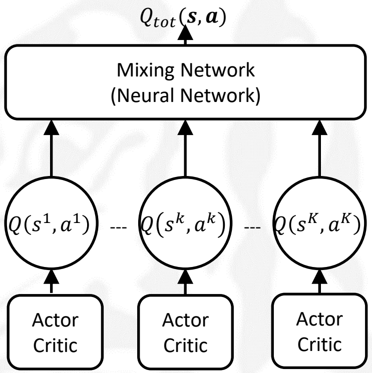


Value-Decomposition Individual Architecture

Actor-Critic Architecture



Centralized Critic



Value Decomposition Network

→ min: $Q(s, a^1, \dots, a^K) - TD_{ERROR}$

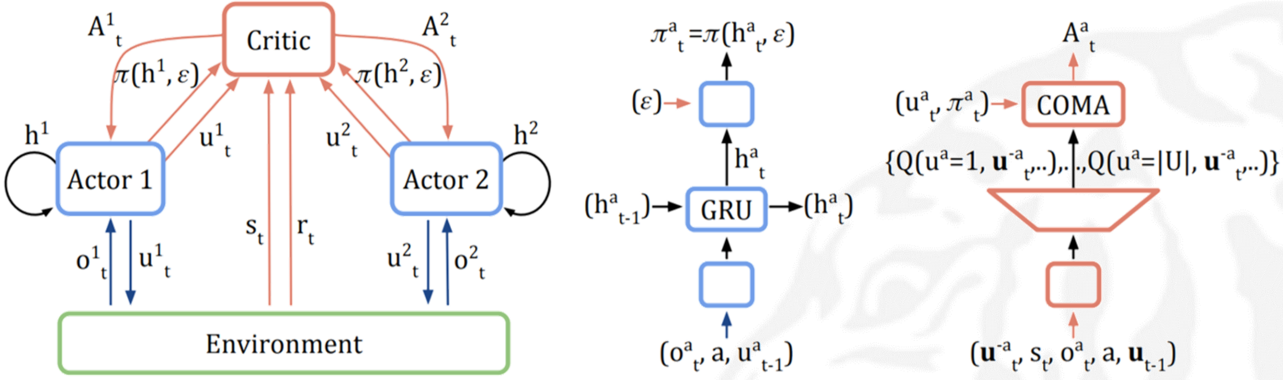
DQN
CommNet: With mean operation
G2ANet: With graph attention network
COMA: with counterfactual method

→ $Q_{tot}(\boldsymbol{s}, \boldsymbol{a}) - TD_{ERROR}$

→ s.t. $Q_{tot}(\boldsymbol{s}, \boldsymbol{a}) = NN_k(Q(s, a^k))$

VDN: Linear Combination
QMIX: With Neural Net
QTRAN: With Complex Neural Net

Artificial Intelligence and Mobility Lab



- Features 1. Centralized Critic: Only the size of the entire V value (or Q value) is evaluated
- Features 2. Counterfactual baseline

VDN

Assumption. $r(\mathbf{s}, \mathbf{a}) = r_1(o^1, a^1) + r_2(o^2, a^2)$

Theorem.

$$\begin{aligned} Q^\pi(\mathbf{s}, \mathbf{a}) &= \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r(\mathbf{s}_t, \mathbf{a}_t) | \mathbf{s}_1 = \mathbf{s}, \mathbf{a}_1 = \mathbf{a}; \pi] \\ &= \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r_1(o_t^1, a_t^1) | \mathbf{s}_1 = \mathbf{s}, \mathbf{a}_1 = \mathbf{a}; \pi] + \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r_2(o_t^2, a_t^2) | \mathbf{s}_1 = \mathbf{s}, \mathbf{a}_1 = \mathbf{a}; \pi] \\ &=: \bar{Q}_1^\pi(\mathbf{s}, \mathbf{a}) + \bar{Q}_2^\pi(\mathbf{s}, \mathbf{a}) \end{aligned}$$

All Agents do action greedy!

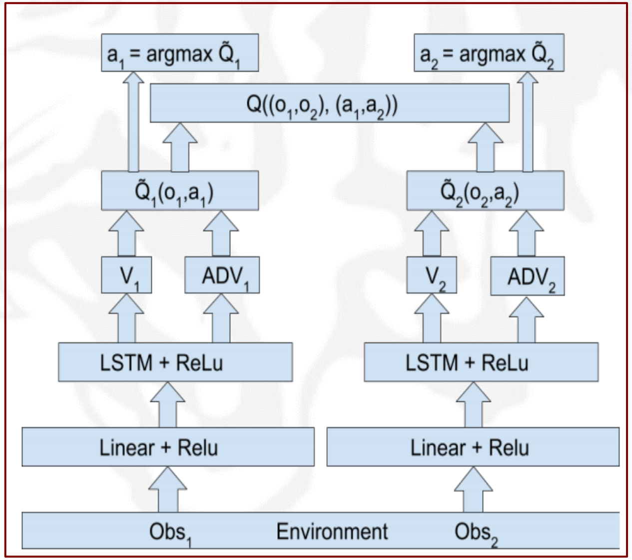
Formula.

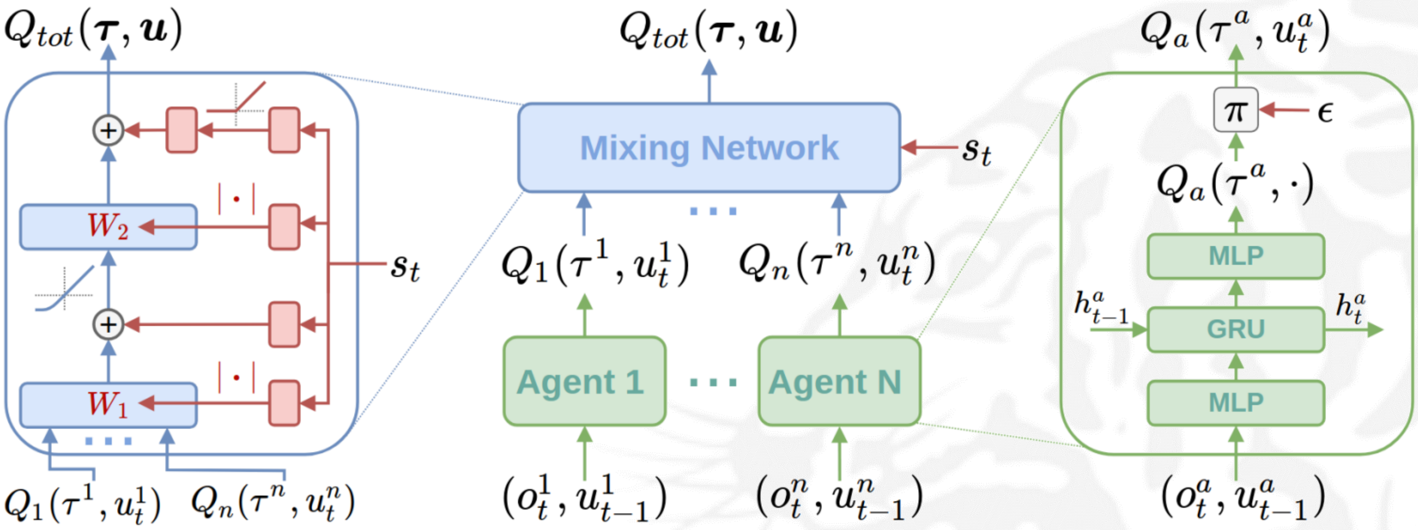
$$\boxed{Q((h^1, h^2, ..., h^d), (a^1, a^2, ..., a^d)) \approx \sum_{i=1}^{d} \tilde{Q}_i(h^i, a^i)}$$

QMIX: With Neural Net

QTRAN: With Complex Neural Net

Qatten: With Attention mechanism





In VDN… joint action-value Q function

$$Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \sum_{i=1}^{n} Q_i(\tau^i, u^i; \theta^i)$$

Optimal joint action-value Q function

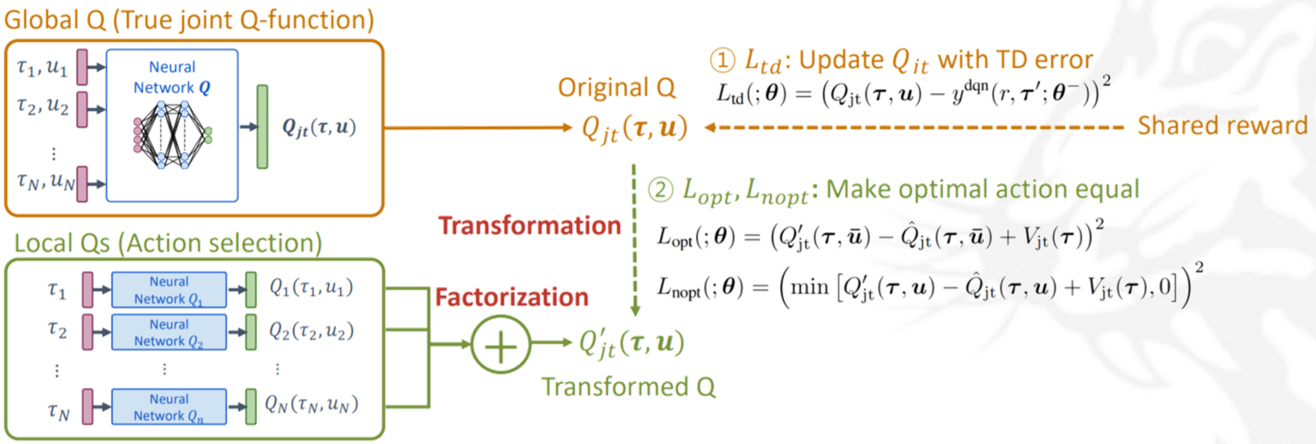
$$\operatorname*{argmax}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix} \quad s.t. \quad \frac{\partial Q_{tot}}{\partial Q_a} \geq 0$$

QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning Proc. ICML 2018

QTRAN

With VDN and QMIX assumption

$$Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \sum_{i=1}^{n} Q_i(\tau^i, u^i; \theta^i) \quad , \quad \operatorname*{argmax}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix} \quad s.t. \quad \frac{\partial Q_{tot}}{\partial Q_a} \geq 0$$



Theorem 1

$$\sum_{i=1}^{N} Q_i(\tau_i, u_i) - Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) + V_{\mathrm{jt}}(\boldsymbol{\tau}) = \begin{cases} 0 & \boldsymbol{u} = \bar{\boldsymbol{u}} \\ \geq 0 & \boldsymbol{u} \neq \bar{\boldsymbol{u}} \end{cases}$$

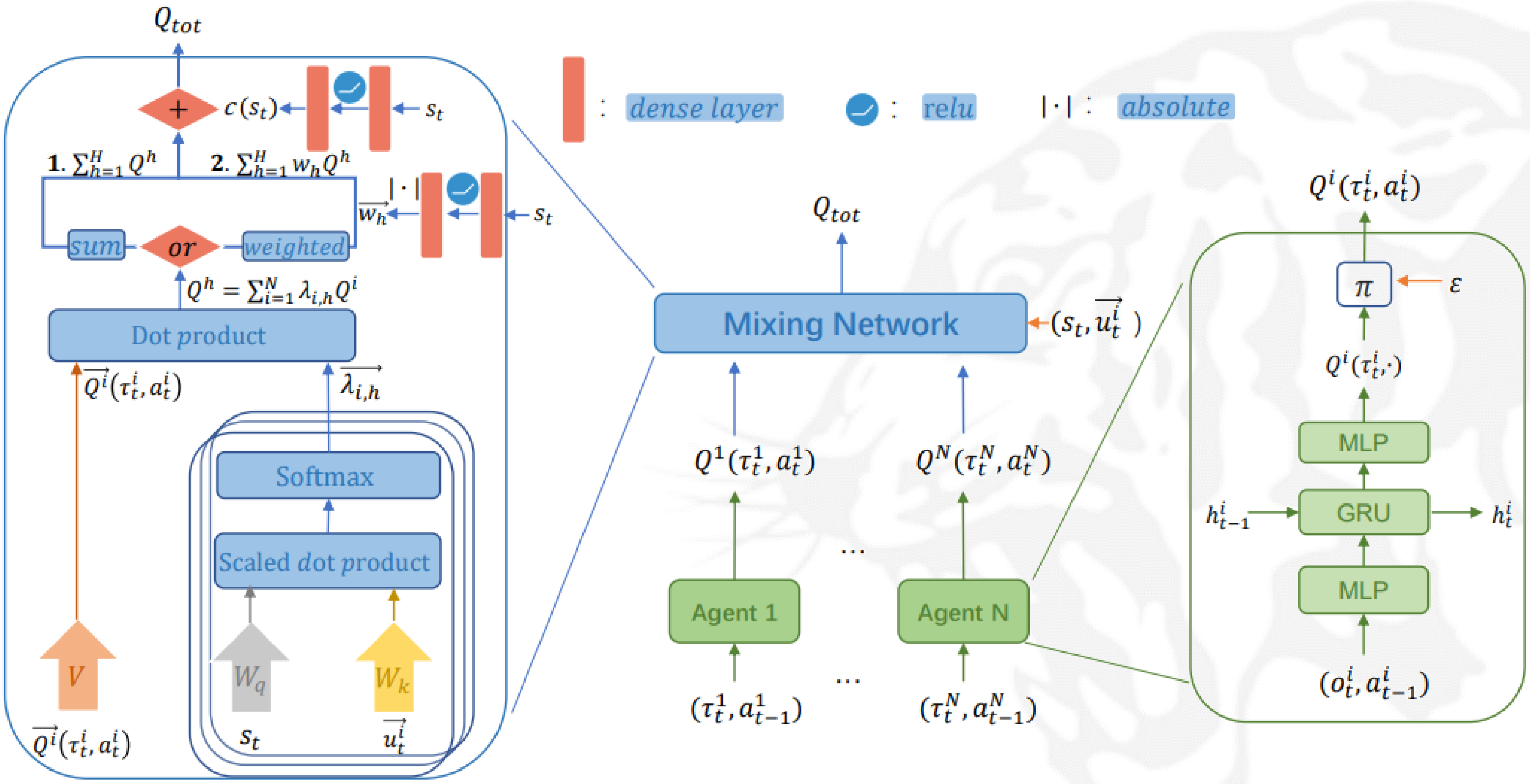
where $V_{\mathrm{jt}}(\boldsymbol{\tau}) = \max_{\boldsymbol{u}} Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) - \sum_{i=1}^{N} Q_i(\tau_i, \bar{u}_i).$

Theorem 2. *The statement presented in* Theorem 1 *and the necessary condition of* Theorem 1 *holds by replacing* (4b) *with the following* (7): *if* $\boldsymbol{u} \neq \bar{\boldsymbol{u}}$,

$$\min_{u_i \in \mathcal{U}} \Big[Q'_{\mathrm{jt}}(\boldsymbol{\tau}, u_i, \boldsymbol{u}_{-i}) - Q_{\mathrm{jt}}(\boldsymbol{\tau}, u_i, \boldsymbol{u}_{-i}) + V_{\mathrm{jt}}(\boldsymbol{\tau}) \Big] = 0, \quad \forall i = 1, \ldots, N, \tag{7}$$

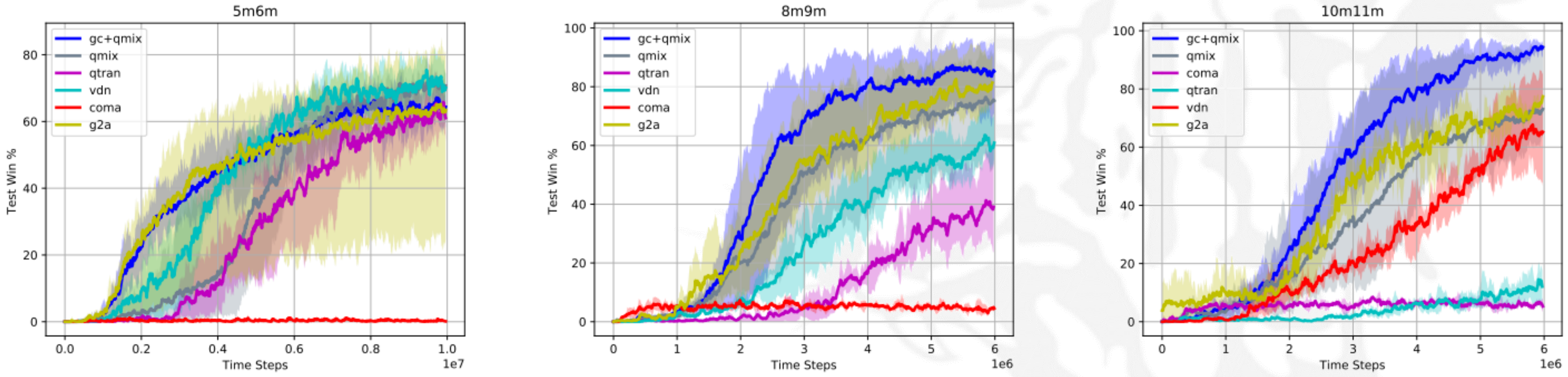
where $\boldsymbol{u}_{-i} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_N)$, i.e., *the action vector except for i's action.*

QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement learning, ICML
https://icml.cc/media/Slides/icml/2019/hallb(13-16-00)-13-17-05-5141-qtran_learning.pdf



Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning, arxiv, 2020

Performance Evaluation



Multi-Agent Reinforcement Learning With Graph Clustering, arxiv , August 2020

Thank you for your attention!

- More questions?
  - ywjoon95@korea.ac.kr