Reinforcement Learning China Summer School

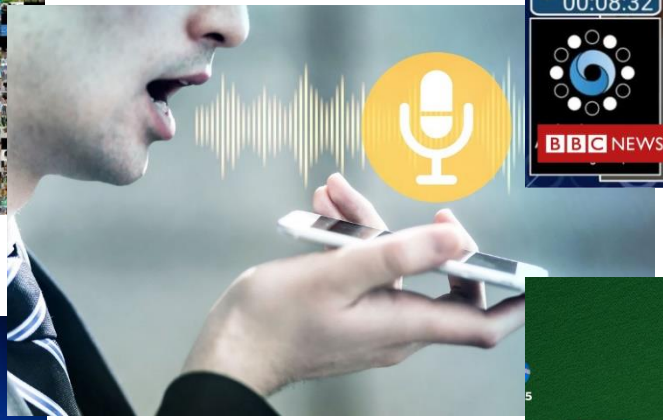
RLChina 2020

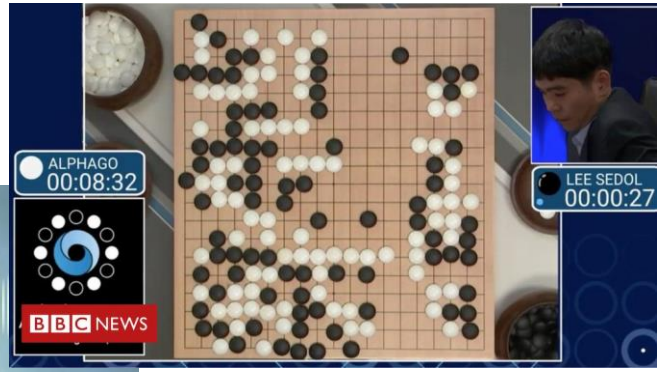
Deep Multi-Agent Reinforcement Learning

Chongjie Zhang
Institute of Interdisciplinary Information Sciences
Tsinghua University

August 6, 2020

Recent AI Breakthrough





What's Next for AI

- Shifting from pattern recognition to decision-making/control
- Shifting from single-agent to multi-agent settings

Drone Delivery

Smart Grids

Home Robots

Autonomous Vehicles

Multi-robot assembly

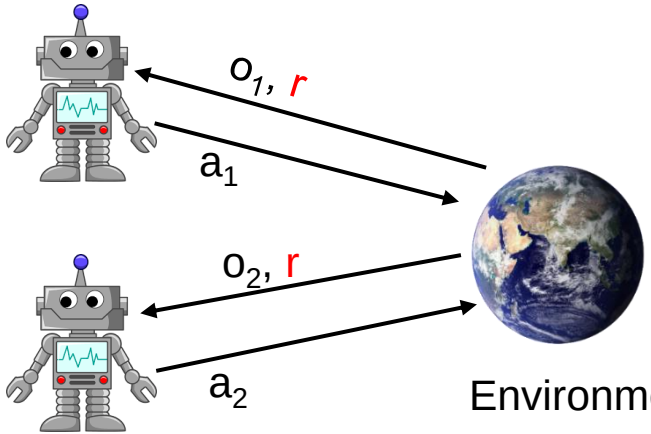
Video Games

Types of Multi-Agent Systems

- Cooperative
  - Working together and coordinating their actions
  - Maximizing a shared team reward
- Competitive
  - Self-interested: maximizing an individual reward
  - Opposite rewards
  - Zero-sum games
- Mixed
  - Self-interested with different individual rewards (not opposite)
  - General-sum games

Cooperative Multi-Agent Systems (MAS)

- A group of agents work together to optimize team performance
- Model: decentralized partially observable Markov decision process (Dec-POMPD)
  - Multi-agent sequential decision-making under uncertainty
  - Extension of MDPs and POMDPs
- At each step, each agent *i* takes an action and receives:
  - A local observation o_i
  - A joint immediate reward *r*



Dec-POMDP

- Model
  - Agent: $i \in I = \{1, 2, \dots, N\}$
  - State: $s \in S$
  - Action: $a_i \in A,\ \boldsymbol{a} \in A^N$
  - Transition function: $P(s' \mid s, \boldsymbol{a})$
  - Reward: $R(s, \boldsymbol{a})$
  - Observation: $o_i \in \Omega$
  - Observation function: $o_i \in \Omega \sim O(s, i)$

Dec-POMDP

- Objective: to find policies for agents to jointly maximize the expected cumulative reward
- A local policy π_i for each agent i: mapping its observation-action history τ_i to its action
  - Action-observation history: $\tau_i \in T = (\Omega \times A)^*$
  - State is unknown, so beneficial to remember the history
- Joint policy $\boldsymbol{\pi} = < \pi_1, \ldots, \pi_n >$
- Value function: $Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}, \boldsymbol{\pi}\right]$
- Policy $\boldsymbol{\pi}(\boldsymbol{\tau}) = \text{argmax}_{\boldsymbol{a}}\, Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})$

Multi-Agent Reinforcement Learning (MARL)

- MARL is promising for solving Dec-POMDP problems
  - The environment model is often unknown

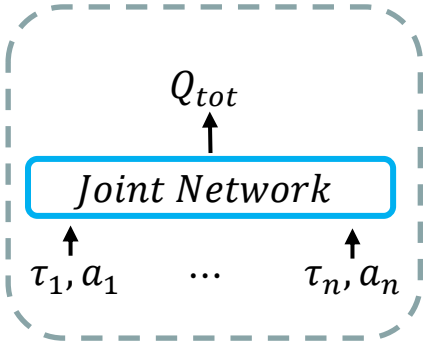
- MARL: learning policies for multiple agents
  - Where agents are interacting
  - Learning by interacting with other agents and the environment

Outline

- Value-Based Methods
  - Paradigm: Centralized Training and Decentralized Execution
  - Basic methods: VDN, QMIX, QPLEX
  - Theoretical analysis
  - Extensions
- Policy Gradient Methods
  - Paradigm: Centralized Critic and Decentralized Actors
  - Method: Decomposable Off-Policy Policy Gradient (DOP)
- Goals:
  - To give an brief introduction to (cooperative) MARL
  - To excite you about MARL

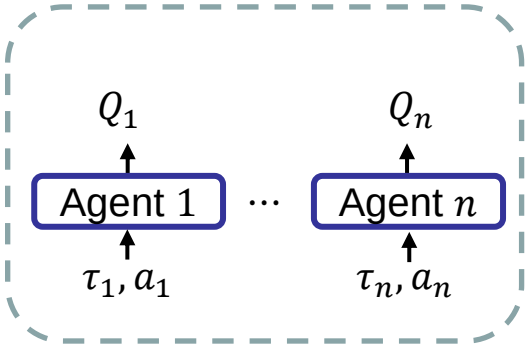
Multi-Agent Reinforcement Learning (MARL)

- How to learn joint policy $\boldsymbol{\pi}$ or value function Q_{tot}?



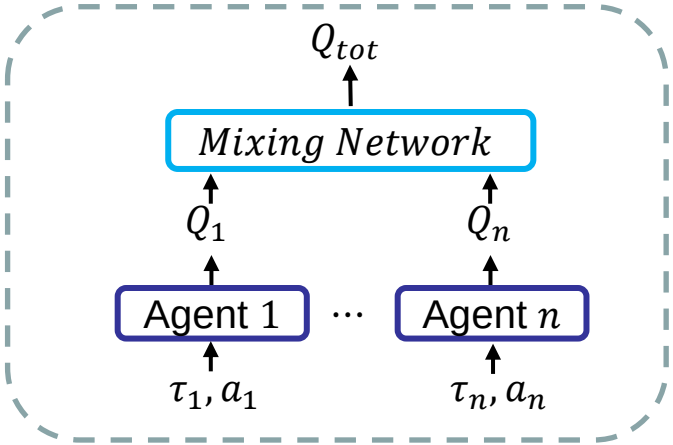
Centralized Value Functions

Scalability



Decentralized Value Functions

Non-stationarity
Credit assignment

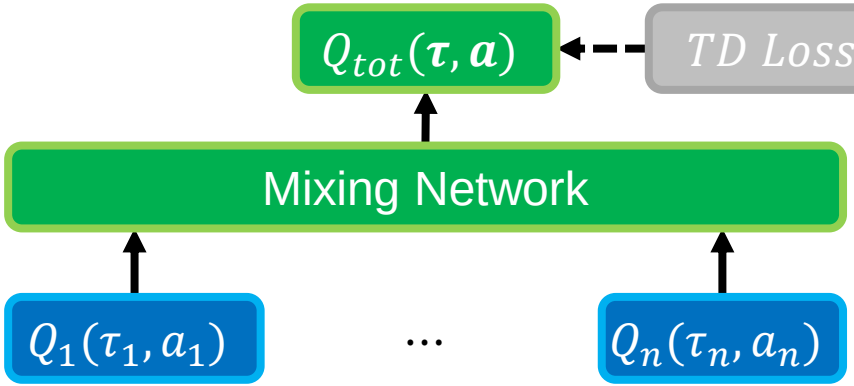


Factorized Value Functions

Centralized training
Decentralized execution

Factorized Value Function Learning

- Paradigm: centralized training with decentralized execution



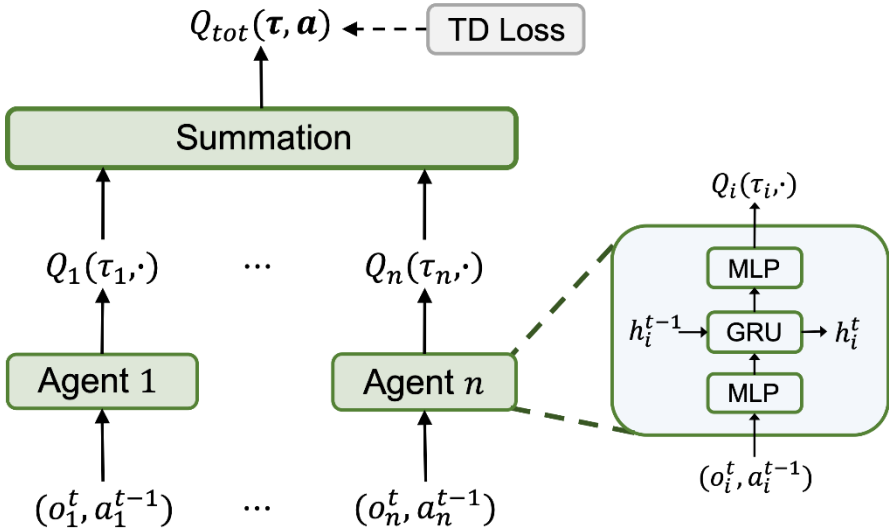
$$\mathcal{L}(\theta) = \mathbb{E}_{(\boldsymbol{\tau},\boldsymbol{a},r,\boldsymbol{\tau}')\in D}\left[\left(r + \gamma V\left(\boldsymbol{\tau}';\theta^{-}\right) - Q\left(\boldsymbol{\tau}, a;\theta\right)\right)^2\right]$$

$$V\left(\boldsymbol{\tau}';\theta^{-}\right) = \max_{\boldsymbol{a}'} Q\left(\boldsymbol{\tau}', \boldsymbol{a}';\theta^{-}\right)$$

- Individual-Global Maximization (IGM) Principle
  - Consistent action selection between joint and individuals
  - $\underset{\boldsymbol{a}}{\text{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\text{argmax}_{a_1} Q_1(\tau_1, a_1), \ldots, \text{argmax}_{a_n} Q_n(\tau_n, a_n)\right)$

Value Decomposition Networks (VDN)

- VDN: $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_i Q_i(\tau_i, a_i)$
- Sufficient for IGM constraint
  - $\underset{\boldsymbol{a}}{\text{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \begin{pmatrix} \text{argmax}_{a_1} Q_1(\tau_1, a_1) \\ \dots \\ \text{argmax}_{a_n} Q_n(\tau_n, a_n) \end{pmatrix}$



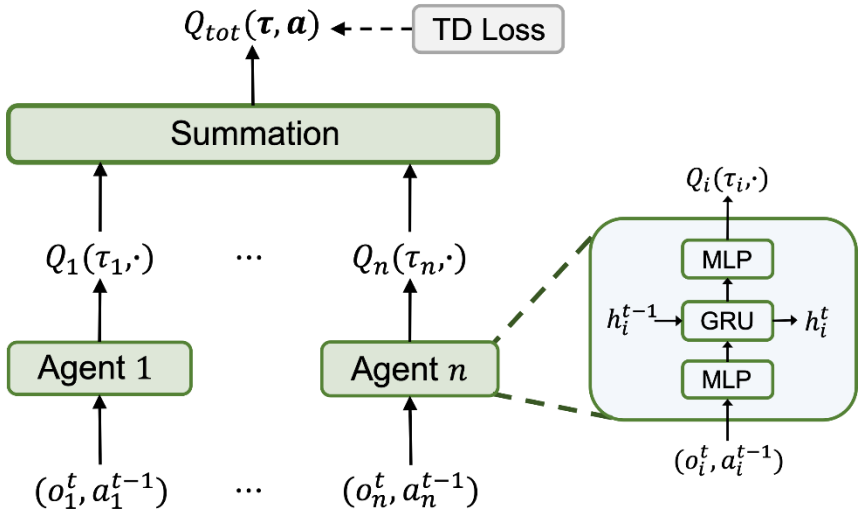
- No specific reward for each agent
- Implicit credit assignment through gradient backpropagation

[Sunehag et. al., 2017]

Learned Kiting Strategy in Starcraft II

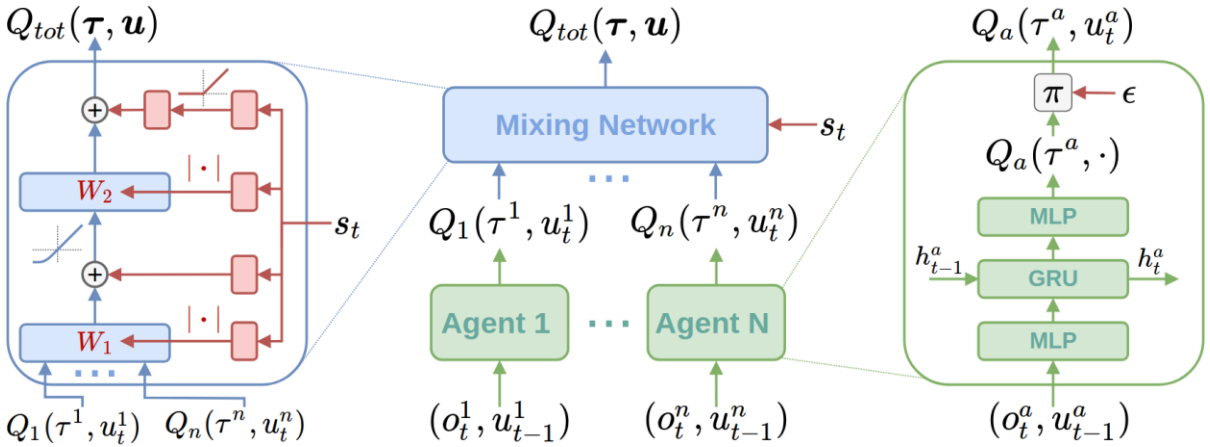
Why VDN Works?

- Scalable maximization operator for action selection
  - Because of the consistency of individual-global maximization
- Parameter sharing among agents
- Implicit credit assignment
- Cons:
  - Not necessary for IGM
  - Limited representation



QMIX: A Monotonic Mixing Network

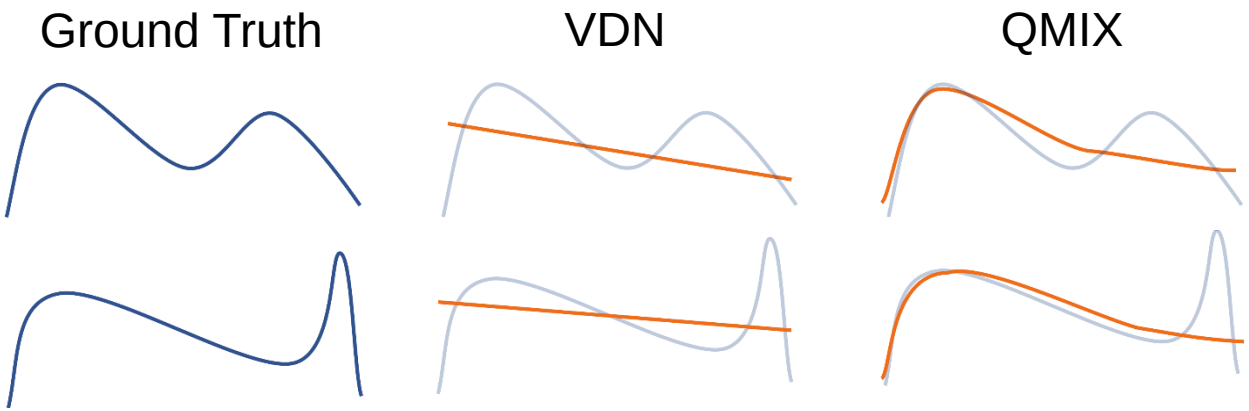
- Monotonic function: $\frac{\partial Q_{tot}}{\partial Q_i} > 0$
- Weights of the mixing network are restricted to be non-negative



[Rashid et. al., 2018]

Representational Complexity

- QMIX's representation is also limited
  - Monotonic condition is sufficient, but not necessary for IGM
- Sketch Illustration



Empirical Results on Matrix Games

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8**	-12	-12
$\mathcal{A}^{(2)}$	-12	0	0
$\mathcal{A}^{(3)}$	-12	0	0

(a) Payoff of matrix game.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.5	-5.0	-5.0
$\mathcal{A}^{(2)}$	-5.0	-3.5	-3.5
$\mathcal{A}^{(3)}$	-5.0	**-3.5**	-3.5

(b) Q_{tot} of VDN.

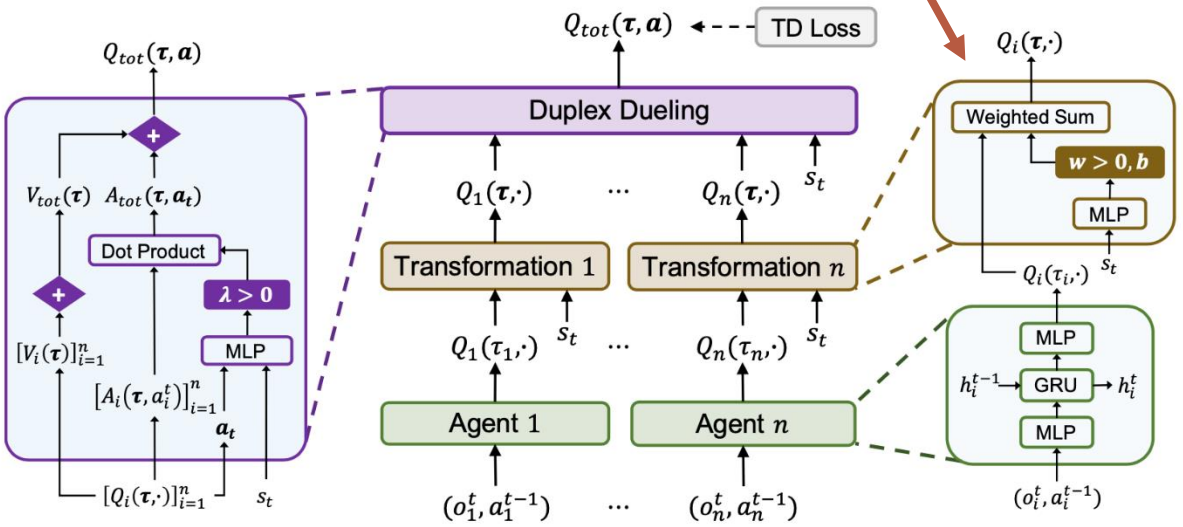
a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-7.8	-7.8	-7.8
$\mathcal{A}^{(2)}$	-7.8	-0.0	**-0.0**
$\mathcal{A}^{(3)}$	-7.8	-0.0	-0.0

(c) Q_{tot} of QMIX.

QPLEX: A Duplex Dueling Multi-Agent Q-Learning

- Three Modules of QPLEX
  - Individual Action-Value Function
  - Transformation
  - Duplex Dueling

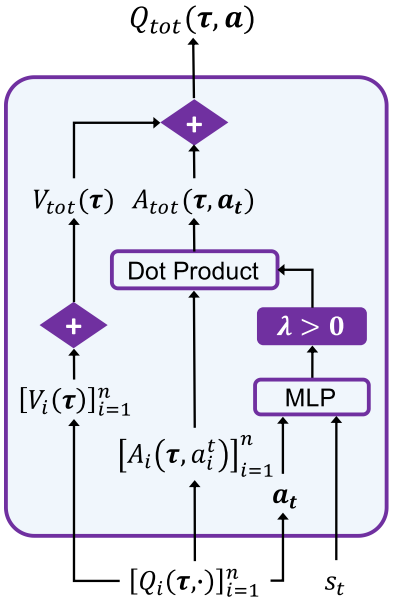
$$Q_i(\boldsymbol{\tau}, a_i) = w_i(\boldsymbol{\tau}) Q_i(\tau_i, a_i) + b_i(\boldsymbol{\tau})$$



[Wang et. al., 2020]

QPLEX: Duplex Dueling Mixing Network

- Input: $\{Q_1(\boldsymbol{\tau},\cdot),\cdots,Q_n(\boldsymbol{\tau},\cdot)\}$
- Individual Dueling:
  - $V_i(\boldsymbol{\tau}) = \max_{a_i'} Q_i(\boldsymbol{\tau}, a_i')$
  - $A_i(\boldsymbol{\tau}, a_i) = Q_i(\boldsymbol{\tau}, a_i) - V_i(\boldsymbol{\tau})$
- Joint Dueling:
  - $V_{tot}(\boldsymbol{\tau}) = \sum_{i=1}^{n} V_i(\boldsymbol{\tau})$
  - $A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{n} \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) A_i(\boldsymbol{\tau}, a_i)$, where $\lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) > 0$
  - $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau}) + A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_i V_i(\boldsymbol{\tau}) + \sum_{i=1}^{n} \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) A_i(\boldsymbol{\tau}, a_i)$



Representation Capacity of QPLEX

Theorem: *The joint action-value function class that QPLEX can realize is equivalent to what is induced by the IGM principle.*

Individual-Global Maximization Principle:

$$\underset{\boldsymbol{a}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \begin{pmatrix} \operatorname{argmax}_{a_1} Q_1(\tau_1, a_1) \\ \dots \\ \operatorname{argmax}_{a_n} Q_n(\tau_n, a_n) \end{pmatrix}$$

QPLEX Realizes the IGM Constraint

- Sufficient Condition for IGM

$$\underset{\boldsymbol{a}}{\operatorname{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \operatorname{argmax}_{\boldsymbol{a}} \sum_{i} V_i(\boldsymbol{\tau}) + \sum_{i=1}^{n} \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) A_i(\boldsymbol{\tau}, a_i) \qquad \text{(QPLEX Definition)}$$

$$= \operatorname{argmax}_{\boldsymbol{a}} \textstyle\sum_{i=1}^{n} \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) A_i(\boldsymbol{\tau}, a_i)$$

$$= \left(\operatorname{argmax}_{a_1} A_1(\boldsymbol{\tau}, a_1), \dots, \operatorname{argmax}_{a_n} A_n(\boldsymbol{\tau}, a_n)\right) \qquad (\forall i, \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) > 0, A_i(\tau_i, a_i) \leq 0, \text{ and } \max_{a_i} A_i(\tau_i, a_i) = 0)$$

$$= \left(\operatorname{argmax}_{a_1} Q_1(\boldsymbol{\tau}, a_1), \dots, \operatorname{argmax}_{a_n} Q_n(\boldsymbol{\tau}, a_n)\right) \qquad (\forall i, Q_i(\boldsymbol{\tau}, a_i) = V_i(\boldsymbol{\tau}) + A_i(\boldsymbol{\tau}, a_i))$$

$$= \left(\operatorname{argmax}_{a_1} w_1(\boldsymbol{\tau}) Q_1(\tau_1, a_1) + b_1(\boldsymbol{\tau}), \dots, \operatorname{argmax}_{a_n} w_n(\boldsymbol{\tau}) Q_n(\tau_n, a_n) + b_n(\boldsymbol{\tau})\right)$$

$$= \left(\operatorname{argmax}_{a_1} Q_1(\tau_1, a_1), \dots, \operatorname{argmax}_{a_n} Q_n(\tau_n, a_n)\right) \qquad (\forall i, w_i(\boldsymbol{\tau}) > 0)$$

QPLEX Realizes the IGM Constraint

- Necessary Condition for IGM

for $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}), \exists \{Q_1(\tau_1, a_1), \dots, Q_n(\tau_n, a_n)\}$, s. t.

$\text{argmax}_{\boldsymbol{a}} Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\text{argmax}_{a_1} Q_1(\tau_1, a_1), \dots, \text{argmax}_{a_n} Q_n(\tau_n, a_n)\right)$

$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau}) + A_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$

$= \sum_i V_i(\boldsymbol{\tau}) + \sum_{i=1}^{n} \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) A_i(\boldsymbol{\tau}, a_i)$

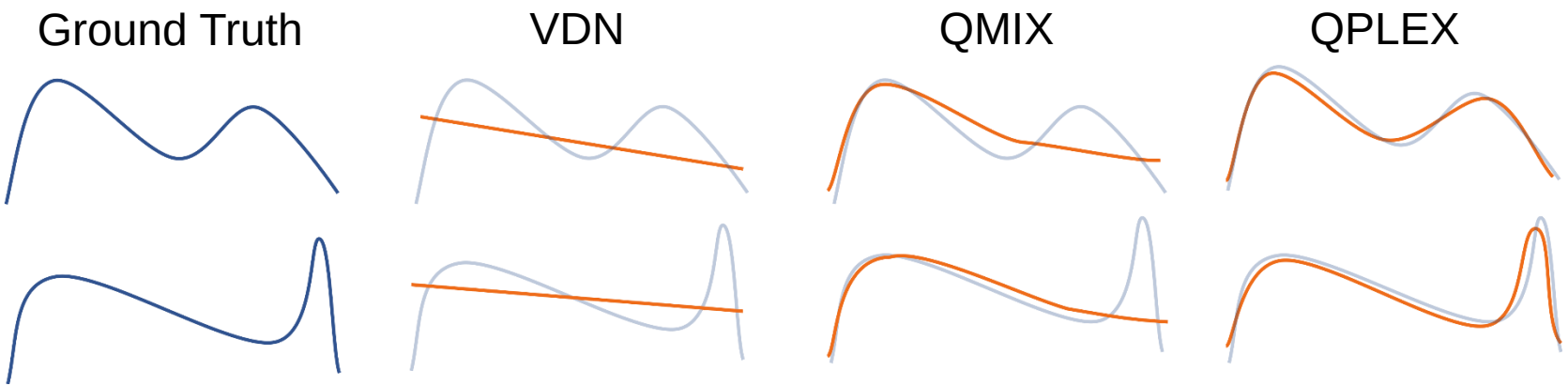
$(\exists \{Q_1(\boldsymbol{\tau}, a_1), \dots, Q_n(\boldsymbol{\tau}, a_n)\}, \exists \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) > 0$
$V_i(\boldsymbol{\tau}) = \max_{a_i} Q_i(\boldsymbol{\tau}, a_i),$
$A_i(\boldsymbol{\tau}, a_i) = Q_i(\boldsymbol{\tau}, a_i) - V_i(\boldsymbol{\tau}))$

$\forall Q_i(\boldsymbol{\tau}, a_i), \exists w_i(\boldsymbol{\tau}) > 0$ and $b_i(\boldsymbol{\tau})$, s. t. $Q_i(\boldsymbol{\tau}, a_i) = w_i(\boldsymbol{\tau}) Q_i(\tau_i, a_i) + b_i(\boldsymbol{\tau})$

In QPLEX, $\lambda_i(\boldsymbol{\tau}, \boldsymbol{a}), w_i(\boldsymbol{\tau}) > 0$ and $b_i(\boldsymbol{\tau})$ are represented by neural networks, and thus $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ can be realized by QPLEX with $\{Q_1(\tau_1, a_1), \dots, Q_n(\tau_n, a_n)\}$.

Representational Complexity

- Sketch Illustration



Empirical Results on Matrix Games

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8**	-12	-12
$\mathcal{A}^{(2)}$	-12	0	0
$\mathcal{A}^{(3)}$	-12	0	0

(a) Payoff of matrix game.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8.0**	-12.1	-12.1
$\mathcal{A}^{(2)}$	-12.2	-0.0	-0.0
$\mathcal{A}^{(3)}$	-12.1	-0.0	-0.0

(a) Q_{tot} of QPLEX.

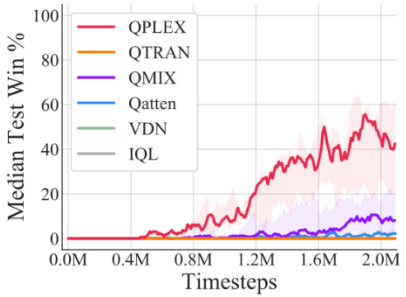
a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.5	-5.0	-5.0
$\mathcal{A}^{(2)}$	-5.0	-3.5	-3.5
$\mathcal{A}^{(3)}$	-5.0	**-3.5**	-3.5

(b) Q_{tot} of VDN.

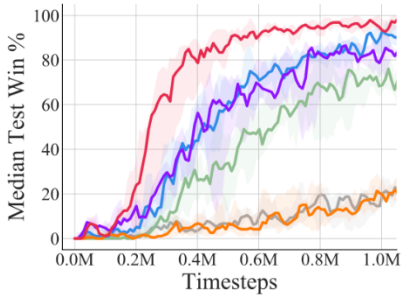
a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-7.8	-7.8	-7.8
$\mathcal{A}^{(2)}$	-7.8	-0.0	**-0.0**
$\mathcal{A}^{(3)}$	-7.8	-0.0	-0.0

(c) Q_{tot} of QMIX.

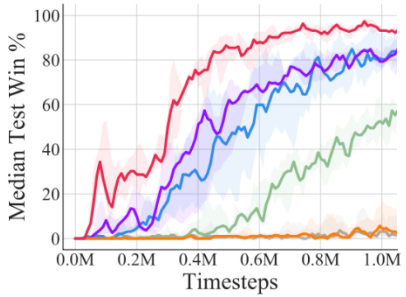
StarCraft II Benchmark: Online Learning



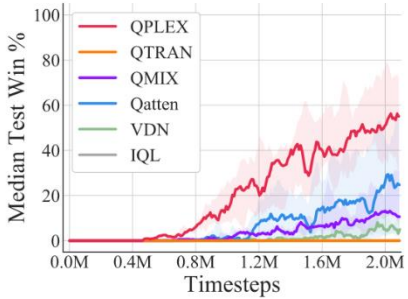
(a) 5s10z



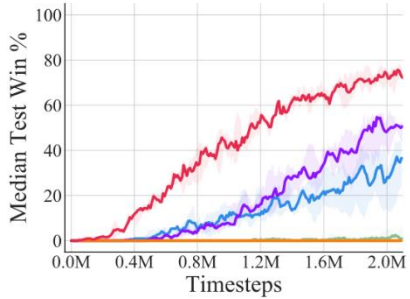
(b) 1c3s5z



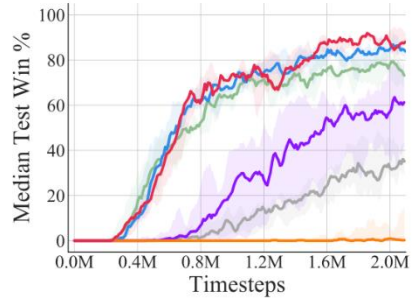
(c) 3s5z



(a) 1c3s8z_vs_1c3s9z



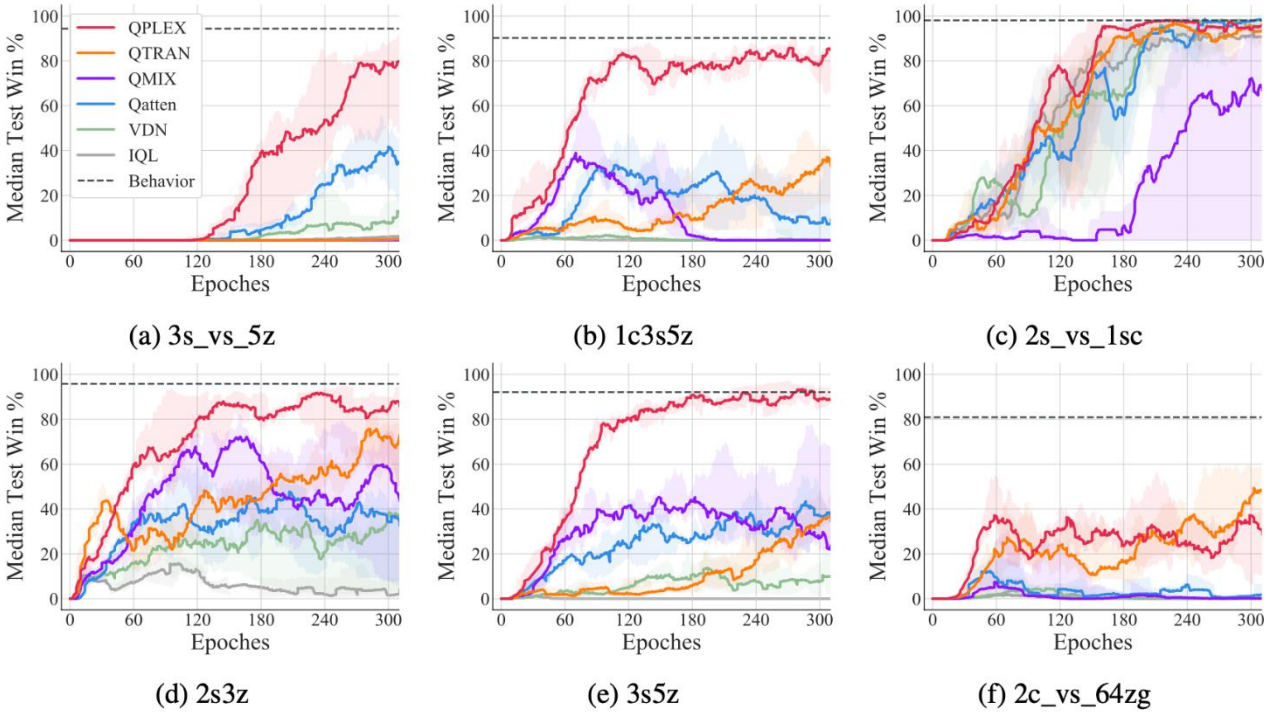
(b) 7sz



(c) 3s_vs_5z

StarCraft II Benchmark: Offline Learning

Data collected by a behavior policy learned by QMIX



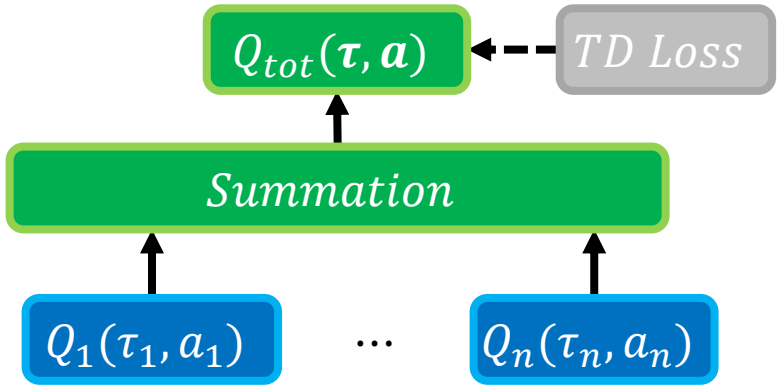
Outline

- Value-Based Methods
  - Paradigm: Centralized Training and Decentralized Execution
  - Basic methods: VDN, QMIX, QPLEX
  - **Theoretical analysis**
  - Extensions

- Policy Gradient Methods
  - Paradigm: Centralized Critic and Decentralized Actors
  - Method: Decomposable Off-Policy Policy Gradient (DOP)

Theoretical Analysis on Linear Value Factorization

- Linear Value Factorization
  - Example methods: VDN, Qatten, …
- A theoretical analysis framework
  - Multi-agent fitted Q-iteration
- Analysis
  - Implicit credit assignment
  - Convergence



[Wang et. al. , 2020]

Fitted Q-Iteration (FQI) Framework

- Multi-Agent MDP (MMDP): assume full observability
- Fitted Q-Iteration (FQI) for MMDP
  - Bellman optimality operator $\mathcal{T}$:
    - $(\mathcal{T}Q)_{tot}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma \mathbb{E}_{s'}\left[\max_{\mathbf{a}'} Q_{tot}(s', \mathbf{a}')\right]$
  - Given a dataset $D = \{(s, \mathbf{a}, r, s')\}$.
  - Iteratively optimization (empirical Bellman error minimization):
    - $Q^{(t+1)} \leftarrow \operatorname{argmax}_{Q} \mathbb{E}_{(s,\mathbf{a},r,s')\sim D}\left[\left(r + \gamma \max_{\mathbf{a}'} Q_{tot}^{(t)}(s', \mathbf{a}') - Q_{tot}(s', \mathbf{a}')\right)^2\right]$

Basic Assumptions

- Assumption 1: deterministic dynamics
  - $P(\cdot\,|s, \mathbf{a})$ is deterministic
- Assumption 2: adequate and factorizable dataset
  - Empirical probability is factorizable
  - $p_D(\mathbf{a}|s) = \prod_i p_D(a_i|s)\,,\ \sum_{a_i} p_D(a_i|s) = 1,\ p_D(a_i|s) > 0$

Fitted Q-Iteration with Linear Value Factorization

- The action-value function class $\mathcal{Q}^{LVD}$:
  - $\mathcal{Q}^{LVD} = \{Q \big| Q_{tot}(\cdot, \mathbf{a}) = \sum_{i=1}^{n} Q_i(\cdot, a_i), \forall \mathbf{a} \text{ and } [\forall Q_i]_{i=1}^{n}\}$
- Given an adequate and factorizable dataset D.
- Iteratively optimization framework:
  - $Q^{(t+1)} \leftarrow \underset{Q \in \mathcal{Q}^{LVD}}{\operatorname{argmax}} \sum_{(s,\mathbf{a})} p_D(\mathbf{a}|s) \left(y^{(t)}(s, \mathbf{a}) - \sum_{i=1}^{n} Q_i(s, a_i)\right)^2$
  - $y^{(t)}(s, \mathbf{a}) = r + \gamma \max_{\mathbf{a}'} Q_{tot}^{(t)}(s', \mathbf{a}')$

Theoretical Analysis of Linear Value Factorization

Theorem 1. (Closed-form solution of FQI-LVD)

- A single iteration of empirical Bellman operator $Q^{(t+1)} = \mathcal{T}_D^{LVD} Q^{(t)}$:
  - $Q_i^{(t+1)}(s, a_i) = \mathbb{E}_{a'_{-i}}\left[y^{(t)}(s, a_i \oplus a'_{-i})\right] - \frac{n-1}{n}\mathbb{E}_{\mathbf{a}'}\left[y^{(t)}(s, \mathbf{a}')\right] + w_i(s)$
  - $a_i \oplus a'_{-i} = \mathbf{a}' = (a'_1, \ldots, a'_{i-1}, a_i, a'_{i+1}, \ldots, a'_n)$
  - $\forall \mathbf{w} = [w_i]_{i=1}^n$ s.t. $\sum_{i=1}^n w_i(s) = 0$

Proof sketch:

- regarded as a weighted linear least squares problem with $n|S||A|$ variables and $|S||A|^n$ data points.
  - $Q^{(t+1)} \leftarrow \underset{Q \in \mathcal{Q}^{LVD}}{\operatorname{argmax}} \sum_{(s,\mathbf{a})} p_D(\mathbf{a}|s)\left(y^{(t)}(s, \mathbf{a}) - \sum_{i=1}^n Q_i(s, a_i)\right)^2$
- Construct a solution and use pseudo inverse to prove

Theoretical Analysis of Linear Value Factorization

- Theorem (Closed-form solution of FQI-LVD)
  - A single iteration of empirical Bellman operator $Q^{(t+1)} = \mathcal{T}_D^{LVD} Q^{(t)}$:
    - $Q_i^{(t+1)}(s, a_i) = \mathbb{E}_{a'_{-i}}\left[y^{(t)}(s, a_i \oplus a'_{-i})\right] - \frac{n-1}{n}\mathbb{E}_{\mathbf{a}'}\left[y^{(t)}(s, \mathbf{a}')\right]$ ~~$+ w_i(s)$~~
    - $\forall \mathbf{w} = [w_i]_{i=1}^n$ s.t. $\sum_{i=1}^n w_i(s) = 0$

1: Will be eliminated when considering the joint action-value function.

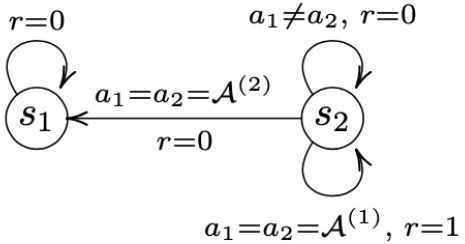
2: Do not change the greedy action selection.

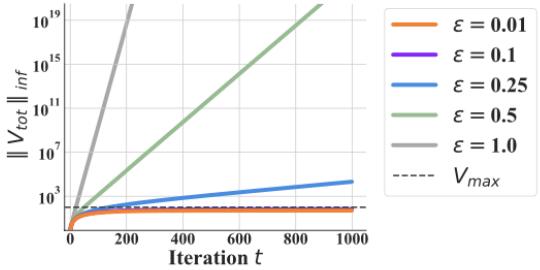
Theoretical Analysis of Linear Value Factorization

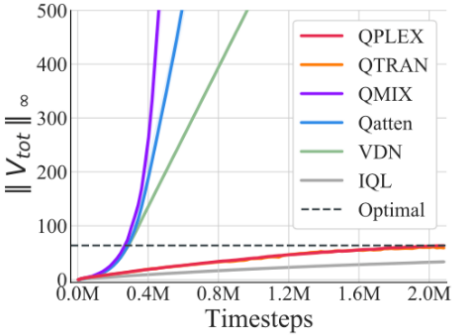
- Implicit **counterfactual** credit assignment mechanism of FQI-LVD
  - A single iteration of empirical Bellman operator $Q^{(t+1)} = \mathcal{T}_D^{LVD} Q^{(t)}$:
    - $Q_i^{(t+1)}(s, a_i) = \underbrace{\mathbb{E}_{a'_{-i}}\left[y^{(t)}(s, a_i \oplus a'_{-i})\right]}_{\text{Evaluation of } a_i} - \frac{n-1}{n} \underbrace{\mathbb{E}_{\mathbf{a}'}\left[y^{(t)}(s, \mathbf{a}')\right]}_{\text{Baseline}}$

Convergence Analysis of Linear Value Factorization

- Theorem 2. *With uniform data distribution, there exists MMDPs, FQI-LVD diverges to infinity from any arbitrary initialization.*
- Theorem 3. *With ϵ-greedy exploration, FQI-LVD has local convergence when ϵ is sufficient small.*



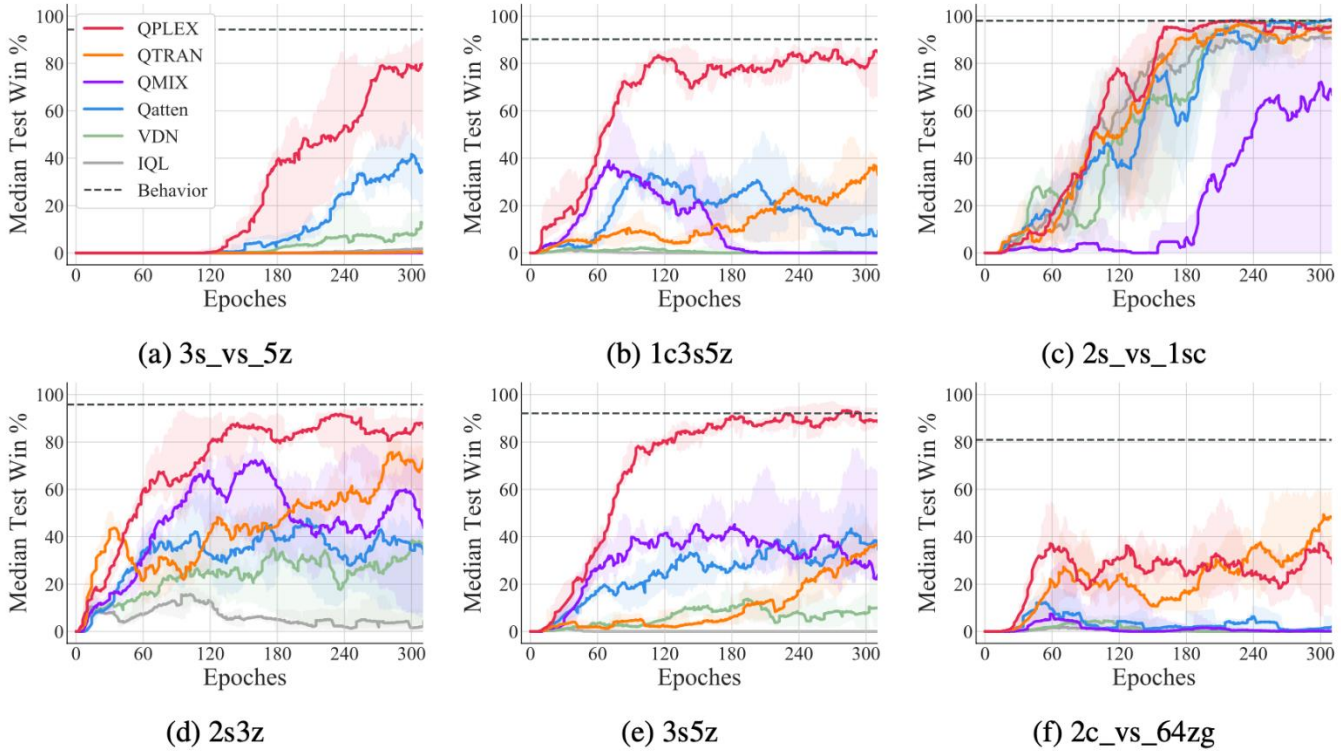




- Multi-agent Q-learning with linear value decomposition structure requires on-policy samples to maintain numerical stability.

Offline Learning on Starcraft Benchmark

Data collected by a behavior policy learned by QMIX



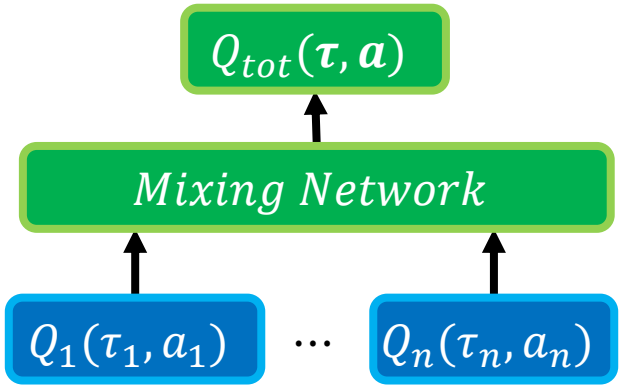
Outline

- Value-Based Methods
  - Paradigm: Centralized Training and Decentralized Execution
  - Basic methods: VDN, QMIX, QPLEX
  - Theoretical analysis
  - **Extensions**

- Policy Gradient Methods
  - Paradigm: Centralized Critic and Decentralized Actors
  - Method: Decomposable Off-Policy Policy Gradient (DOP)

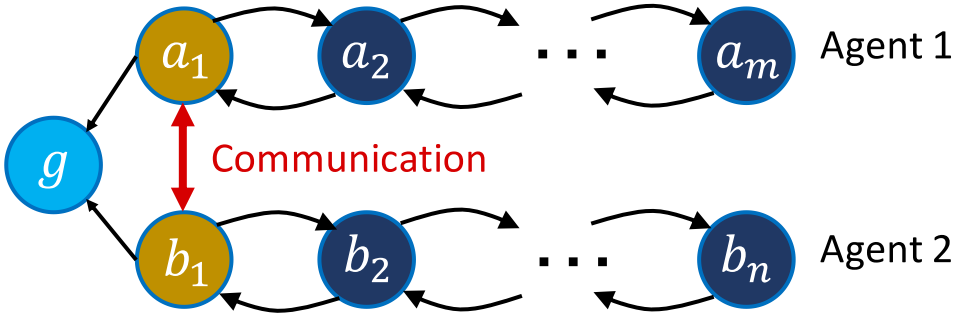
Challenges of Value Factorization Learning

- Uncertainty
  - Full value factorization ➔ miscoordination



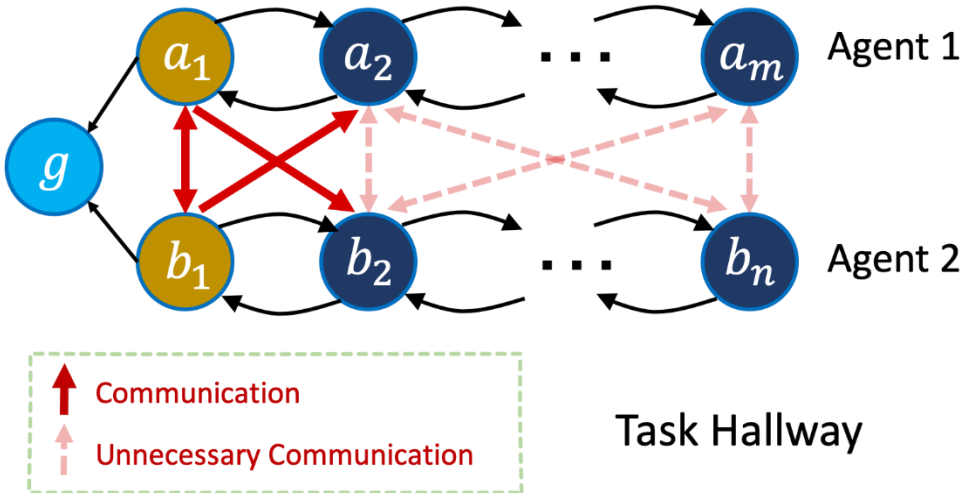
Limitations of Full Value Factorization

- Can cause miscoordinations during execution
  - Need Communication!

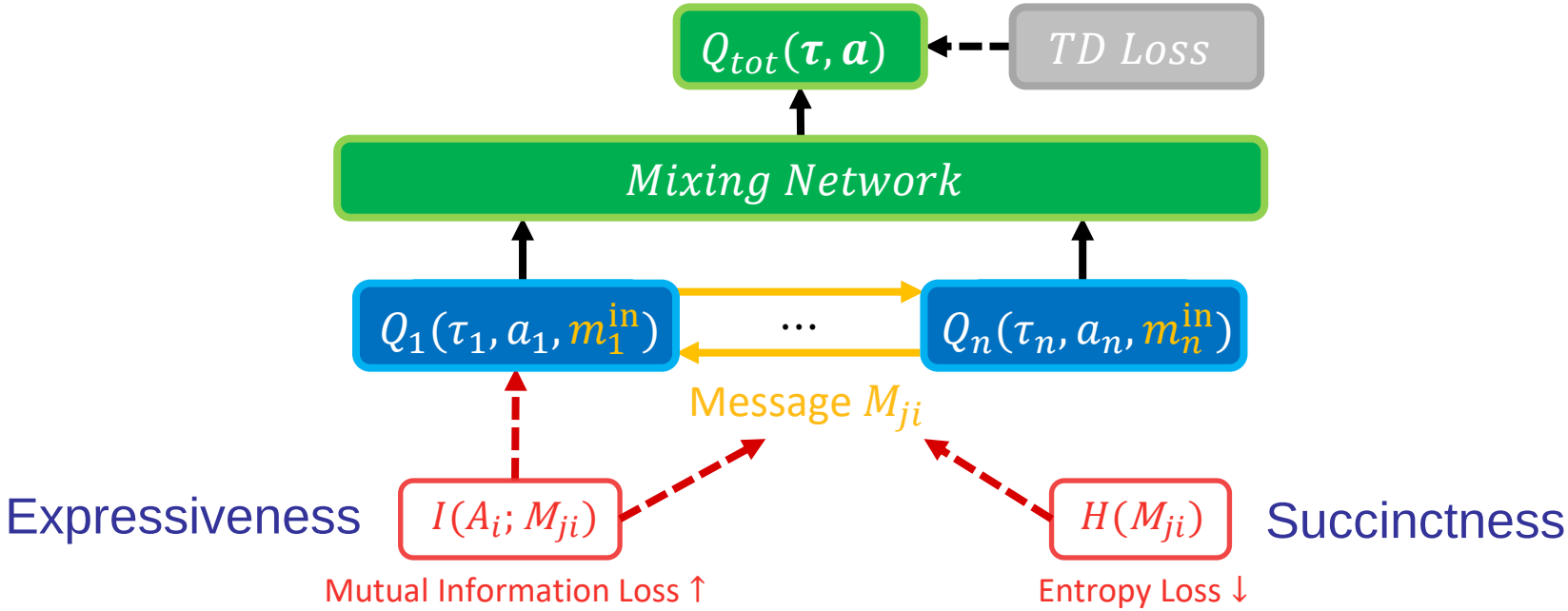


Nearly Decomposable Q-Value Learning (NDQ)

- Allowing communication, but minimized
- Learn when, what, and with whom to communicate

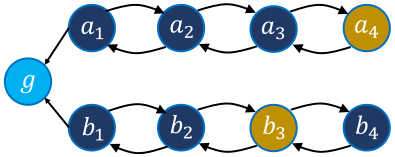


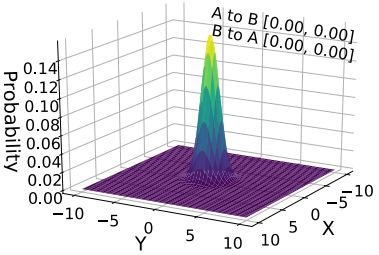
NDQ Framework: Communication Optimization



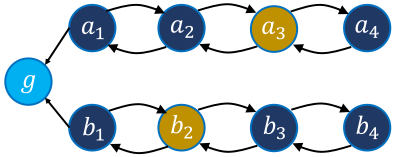
[Wang et. al. , 2020]

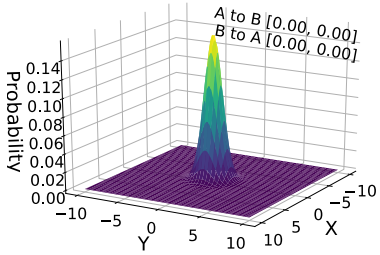
Learning Optimal Communication Protocol



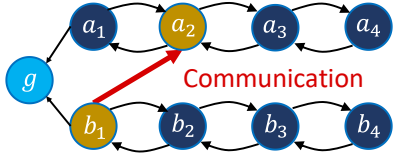


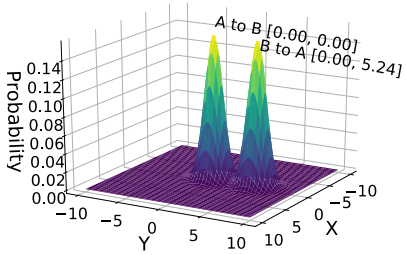
$t = 1$



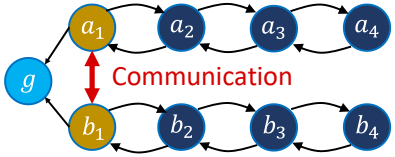


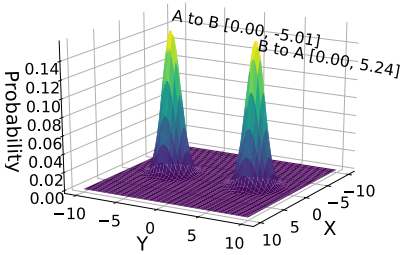
$t = 2$





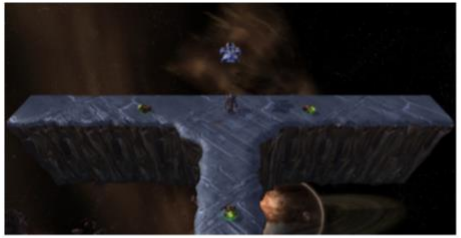
$t = 3$



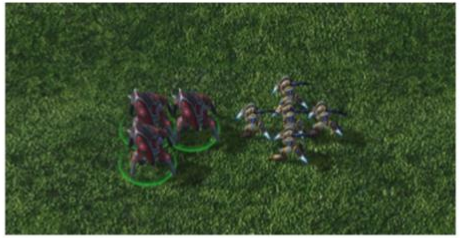


$t = 4$

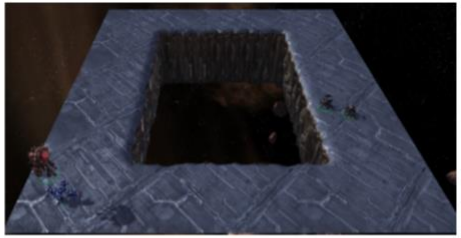
Experiments: Micro-Management in Starcraft II



(a) 3b_vs_1h1m

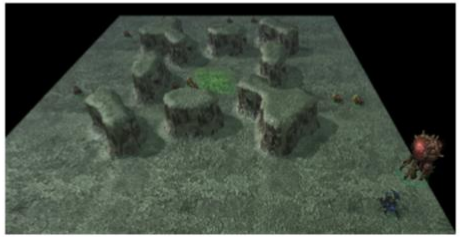


(b) 3s_vs_5z



(c) 1o2r_vs_4r

(d) 5z_vs_1ul



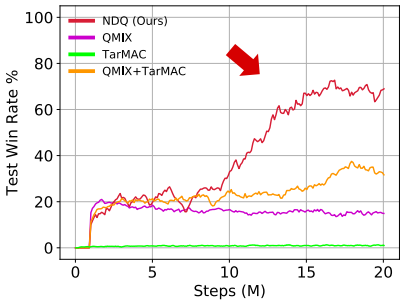
(e) 1o10b_vs_1r

(f) MMM

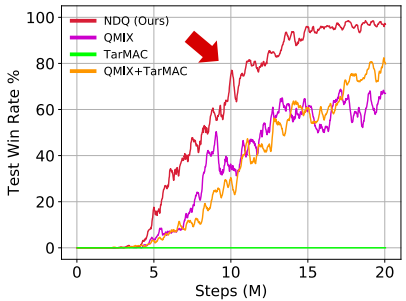
https://sites.google.com/view/ndq

SC2 benchmark: without Message Drop

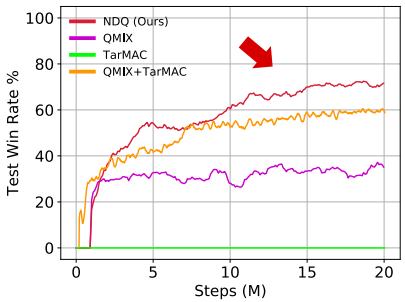
3b vs 1h1m



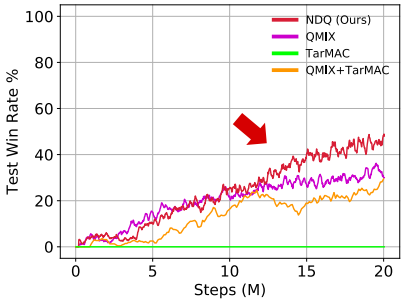
3s vs 5z



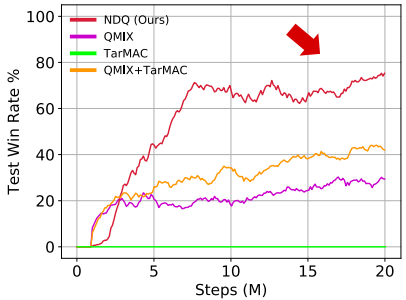
1o2r vs 4r



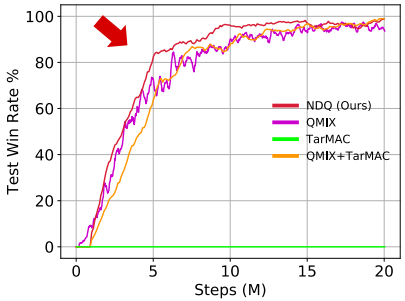
5z vs 1ul



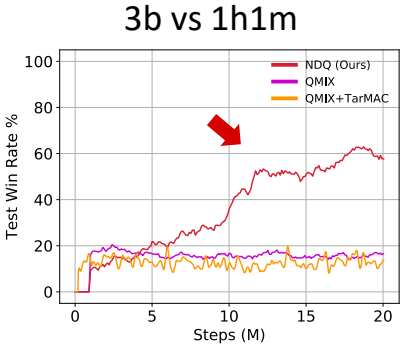
1o10b vs 1r

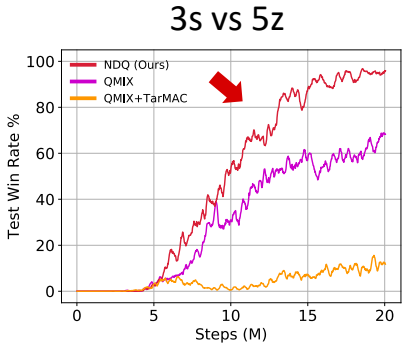


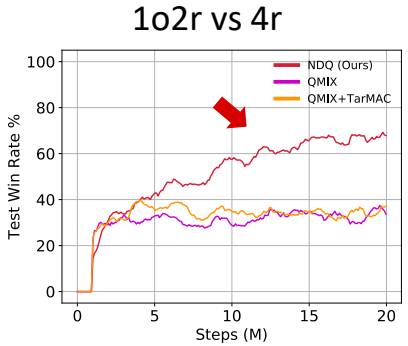
MMM

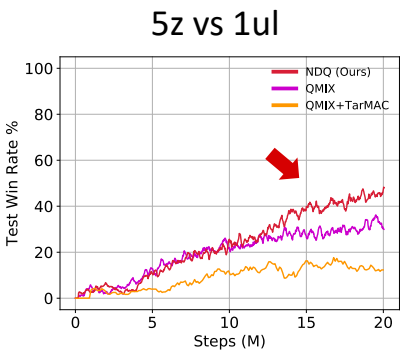


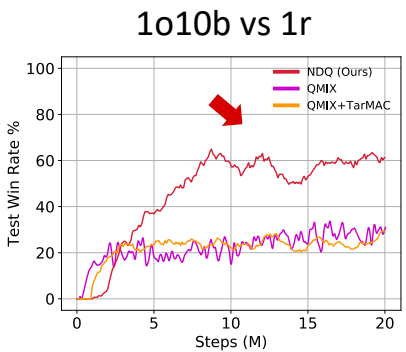
SC2 benchmark: 80% Messages Dropped

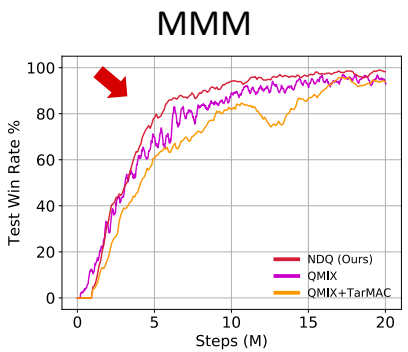






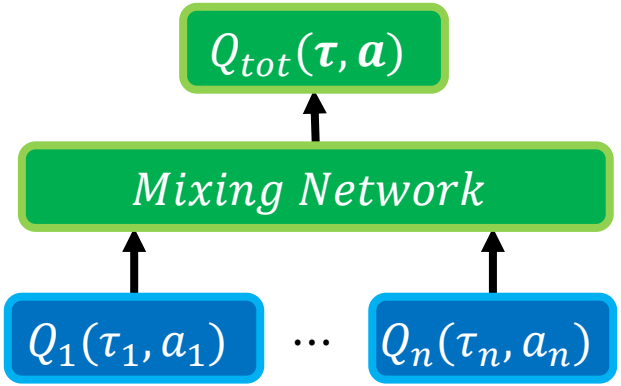






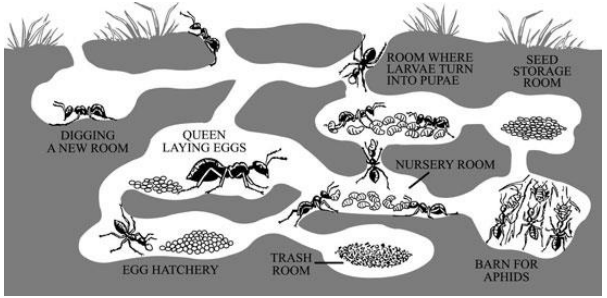
Challenges of Value Factorization Learning

- Uncertainty
  - Full value factorization ➔ miscoordination
- Complex tasks require diverse or heterogeneous agents
  - Shared value network ➔ ineffective learning



Why dynamic shared learning?

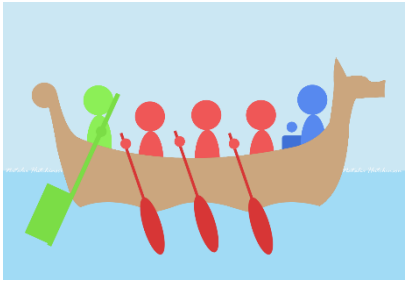
- Complex cooperative tasks require **diverse** behaviors among agents



- Learning a single shared policy network for agents[1-4]
  - Lack of diversity and requiring a high-capacity neural network
  - May result in slow, ineffective learning
- Learning independent policy networks is not efficient
  - Some agents perform similar sub-tasks, especially in large systems

[1] Rashid, et. al. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. (ICML 2018)
[2] Vinyals, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. (Nature 2019)
[3] Baker, et al. Emergent tool use from multi-agent autocurricula. (ICLR 2020)
[4] Lowe, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. (NeurIPS 2017)

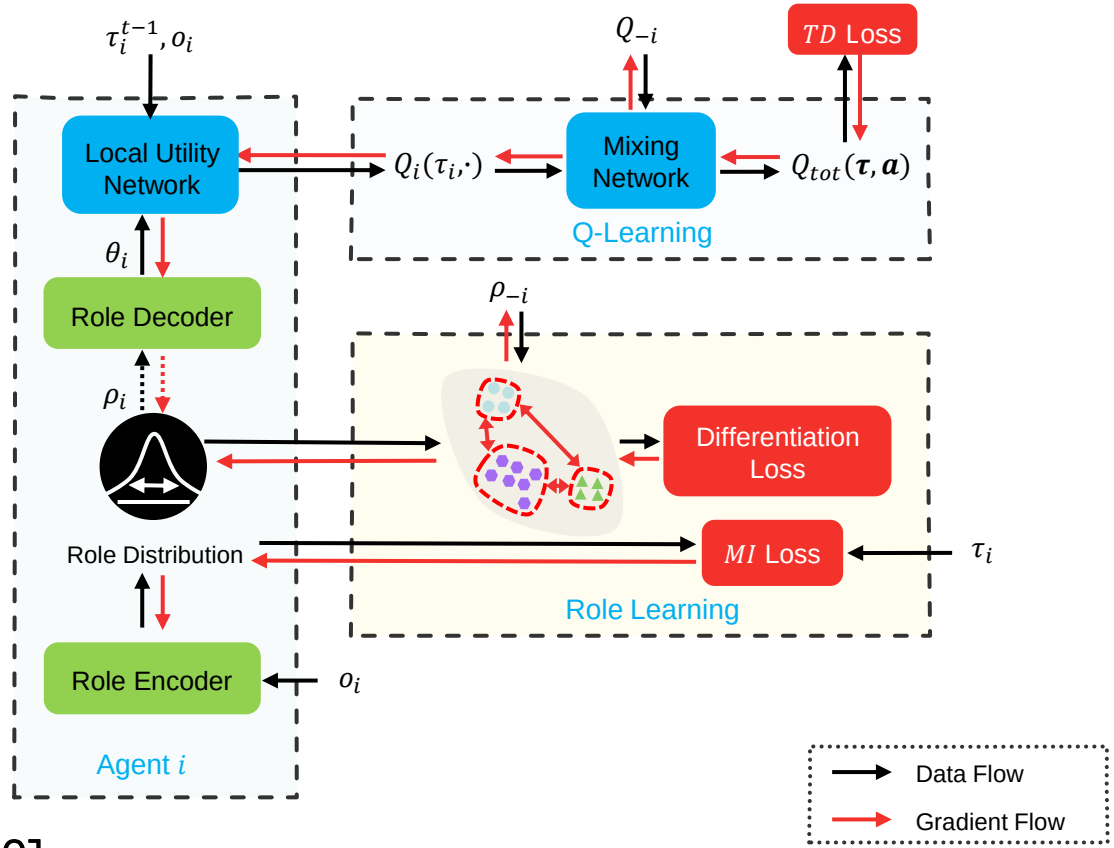
ROMA: Multi-Agent Reinforcement Learning with Emerging Roles



- Agents with similar roles have similar policies and share their learning
  - Similar roles ⟺ similar subtasks ⟺ similar behaviors
- Inferring an agent's roles based on the local observations and execution trajectories
- Conditioning agents' policies on their roles
- An agent can change its roles in different situations

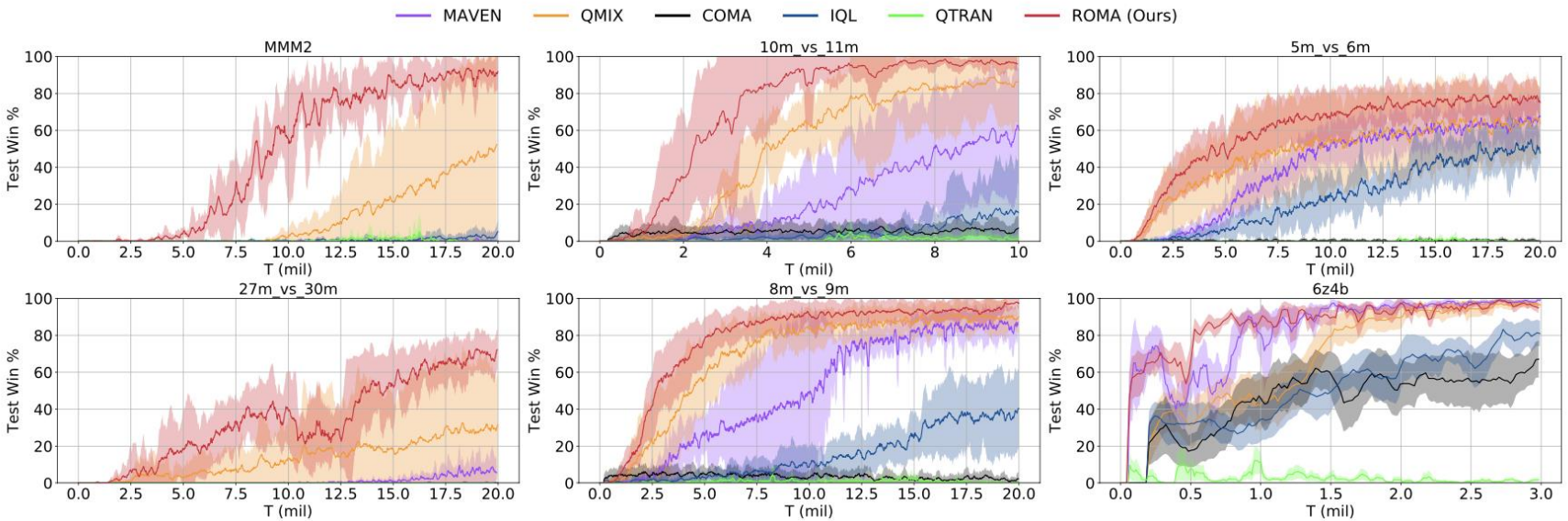
[Wang et. al. , 2020]

ROMA Framework



[Wang et. al. , 2020]

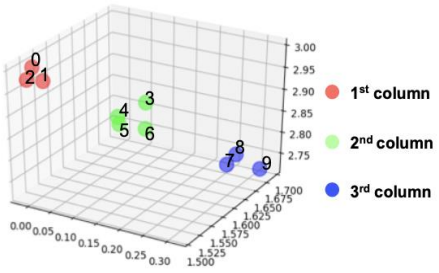
The SMAC Challenge in Starcraft II



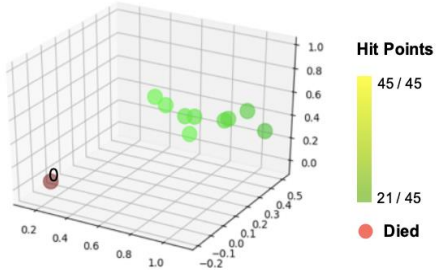
https://sites.google.com/view/romarl

Starcraft II: 27 Marines vs 30 Marines

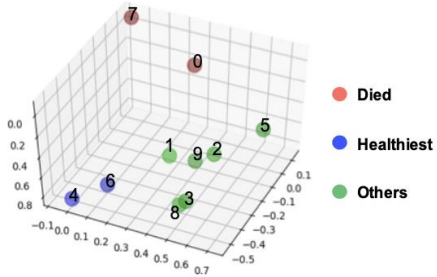
Dynamic Roles



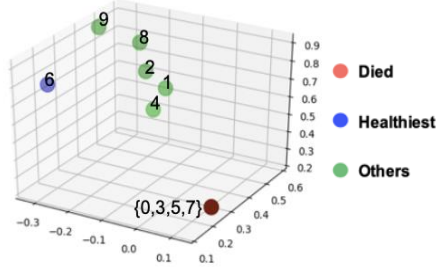
$t = 1$



$t = 8$

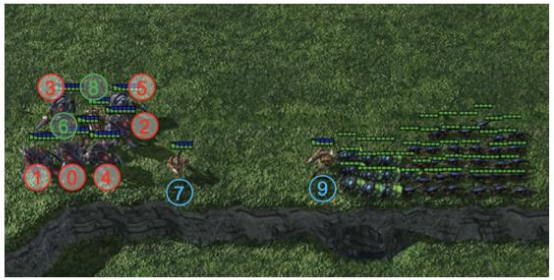


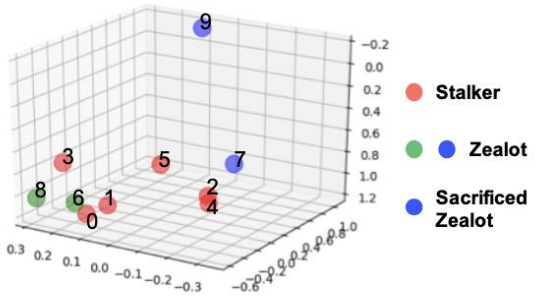
$t = 19$



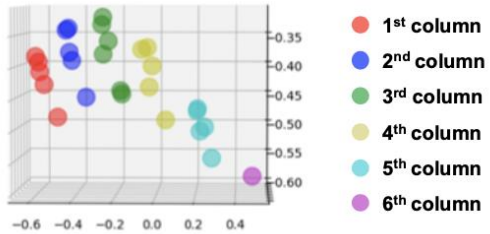
$t = 27$

Specialized Roles

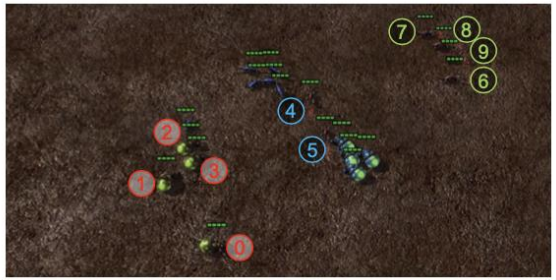


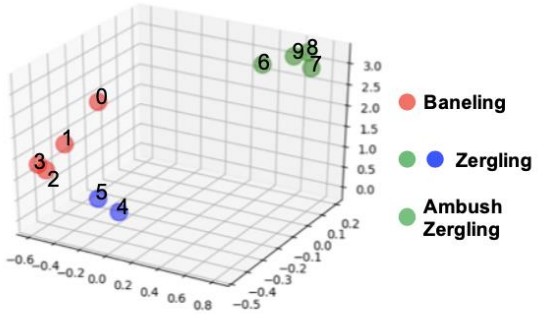


(a) Strategy: sacrificing Zealots 9 and 7 to minimize Banelings' splash damage.

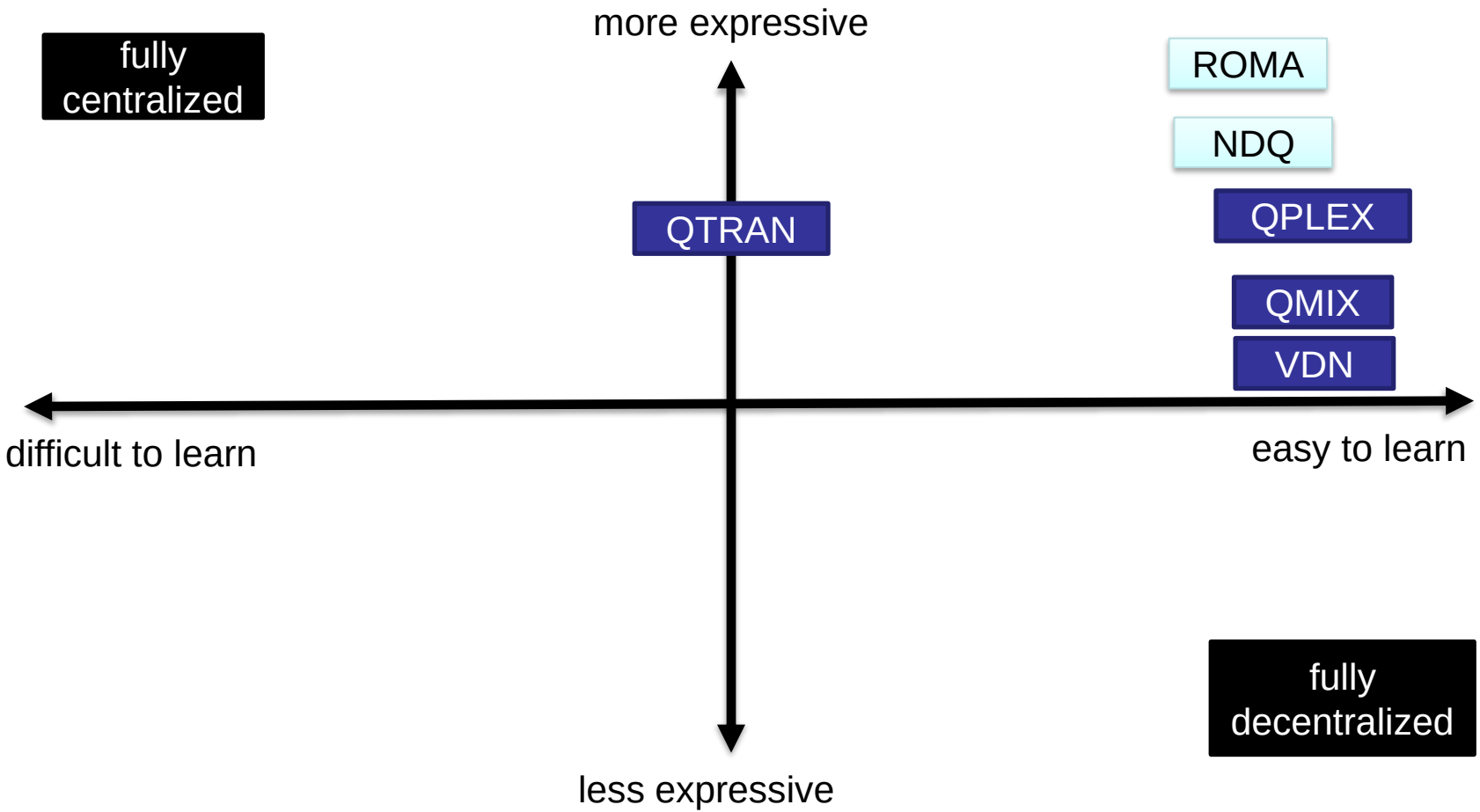


(b) Strategy: forming an offensive concave arc quickly





(c) Strategy: green Zerglings hide away and Banelings kill most enemies by explosion.


more expressive
fully centralized
ROMA
NDQ
QPLEX
QTRAN
QMIX
VDN
difficult to learn
easy to learn
fully decentralized
less expressive

Outline

- Value-Based Methods
  - Paradigm: Centralized Training and Decentralized Execution
  - Basic methods: VDN, QMIX, QPLEX
  - Theoretical analysis
  - Extensions

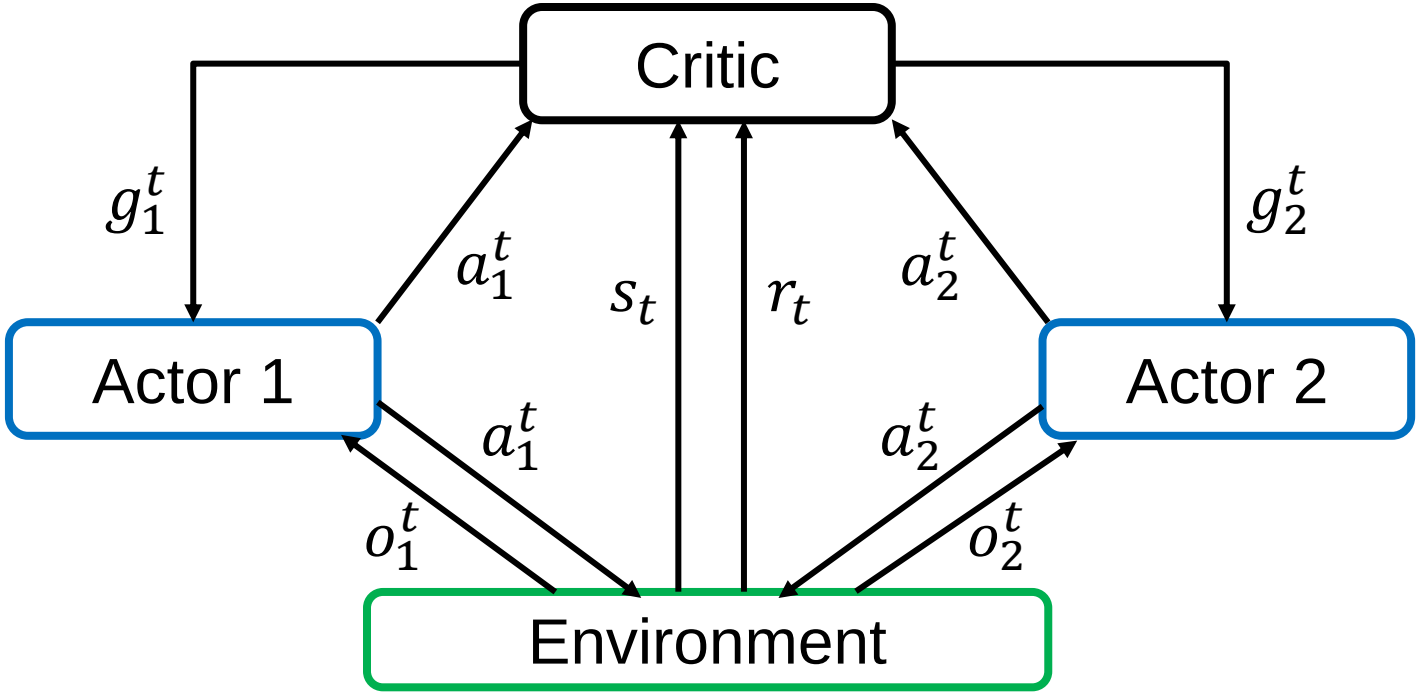
- **Policy Gradient Methods**
  - Paradigm: Centralized Critic and Decentralized Actors
  - Method: Decomposed Off-Policy Policy Gradient (DOP)

Value-Based Methods

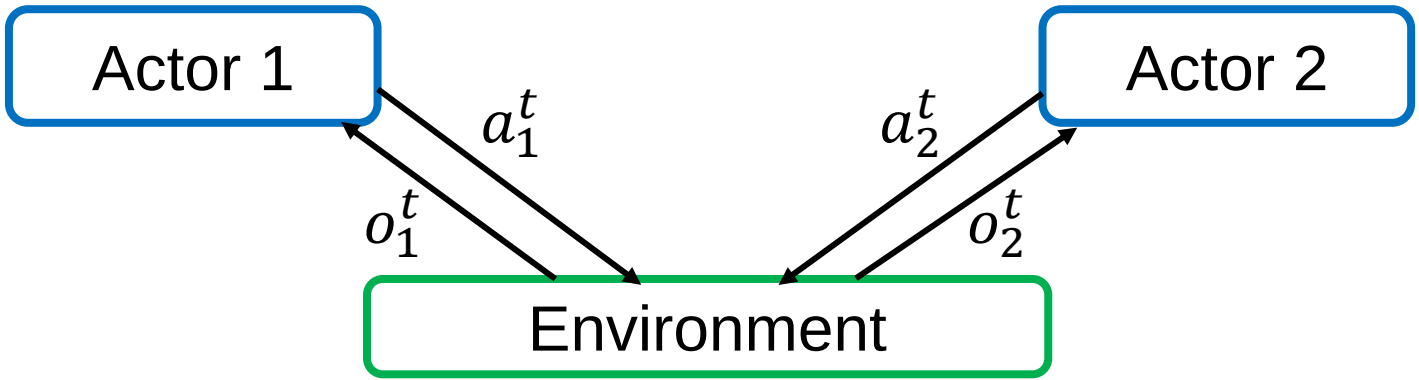
- Significantly contribute to the recent progress of MARL.
  - VDN, QMIX, QPLEX, QTRAN, Qatten, NDQ, ROMA, …
- Drawbacks:
  - Lack of stability;
  - Limited in discrete action spaces.

- Policy gradient methods hold a promise
- Paradigm: centralized critic with decentralized actors

Centralized Critic with Decentralized Actors



Centralized Critic with **Decentralized Actors**



Centralized Critic with Decentralized Actors

- Single-agent

$$g = \mathbb{E}_{\pi}\left[Q^{\pi}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s)\right]$$

- Multi-agent (Centralized Critics with Decentralized Actors)

$$g = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_i Q^{\pi}(s,\boldsymbol{a})\,\nabla_{\theta}\log\pi_i(a_i|\tau_i)\right]$$

- Add a baseline to reduce variance

$$g = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_i (Q^{\pi}(s,\boldsymbol{a}) - b(s,\boldsymbol{a}_{-i}))\nabla_{\theta}\log\pi_i(a_i|\tau_i)\right]$$

Independent of a_i

Counterfactual Baseline to Assign Credit

- Gradient: $g = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_i (Q^{\pi}(s, \boldsymbol{a}) - b(s, \boldsymbol{a}_{-i}))\nabla_{\theta} \log \pi_i(a_i|\tau_i)\right]$
- A problem
  - $Q^{\pi}(s, \boldsymbol{a})$ is an estimation of the global return;
  - Not tailored to a specific agent
- COMA method: counterfactual baseline:
  - $b(s, \boldsymbol{a}_{-i}) = \sum_{a_i} \pi_i(a_i|o_i) Q^{\pi}(s, \boldsymbol{a}_{-i}, a_i)$
  - Simultaneously achieving
    - Variance reduction
    - Credit assignment

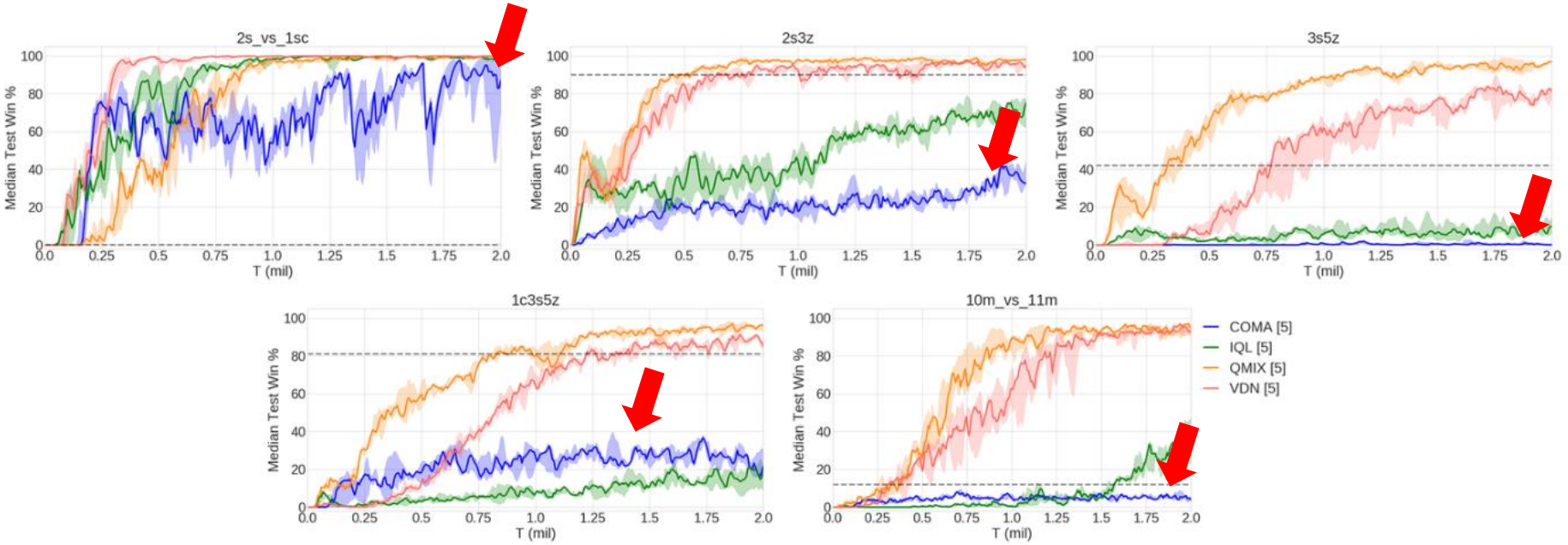
[Foerster et. al., 2018]

Policy-Based Methods

- Centralized Critic with Decentralized Actors
  - COMA[6] (stochastic policy gradients)
  - MADDPG[7] (deterministic policy gradients)
  - Some extensions
    - MAAC[8]

[6] Foerster, J.N., Farquhar, G., Afouras, T., Nardelli, N. and Whiteson, S., Counterfactual multi-agent policy gradients. *(AAAI 2018)*.
[7] Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Abbeel, O.P. and Mordatch, I., Multi-agent actor-critic for mixed cooperative-competitive environments. *(NIPS 2017)*.
[8] Iqbal, S. and Sha, F., Actor-attention-critic for multi-agent reinforcement learning. *(ICML 2019)*.

Unsatisfactory Performance



Why?

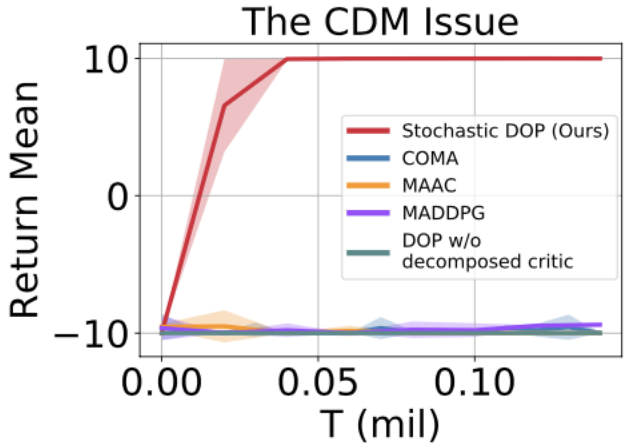
- On-policy learning of stochastic PG
  - E.g., COMA
- Lack credit assignment mechanism of deterministic PG
  - E.g., MADDPG, MAAC
- Centralized-Decentralized Mismatch (CDM)
  - Centralized critics introduce the influence of other agents.

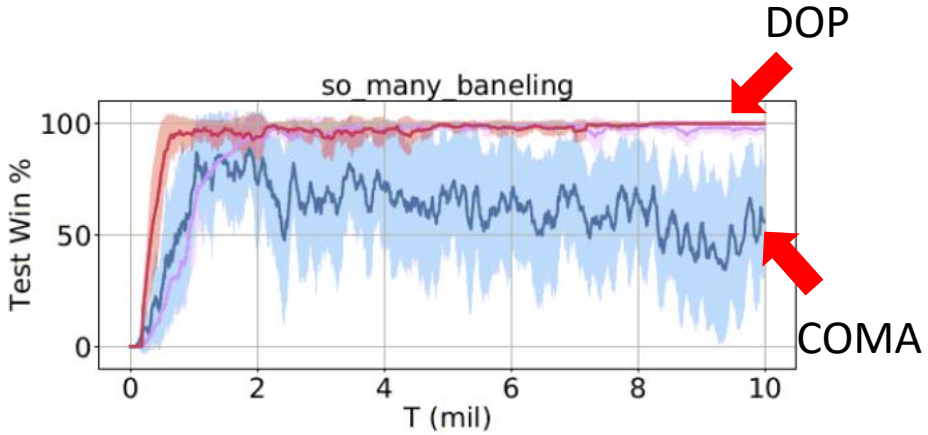
Centralized-Decentralized Mismatch (CDM)

$$g = \mathbb{E}_{\boldsymbol{\pi}}[\sum_i Q_{tot}^{\boldsymbol{\pi}}(\tau, a_1, a_2, \cdots, a_n)\nabla_{\theta_i} \log \pi_i(a_i|\tau_i)]$$

- The joint critic introduces the influence from other agents.
  - Assume the optimal action under $\boldsymbol{\tau}$ is $\boldsymbol{a}^* = (a_1^*, \cdots, a_n^*)$.
  - It is possible that $\mathbb{E}_{\boldsymbol{\pi}_{-i}}[Q_{tot}^{\boldsymbol{\pi}}(\tau, \boldsymbol{a}_{-i}, a_i^*)] < 0$, leading to the decrease of $\pi_i(a_i^*|\tau_i)$. (e.g. when $\boldsymbol{\pi}_{-i}$ is suboptimal)
  - Suboptimality reinforces each other by propagating though the joint critic.
  - Leading to large variance

Centralized-Decentralized Mismatch

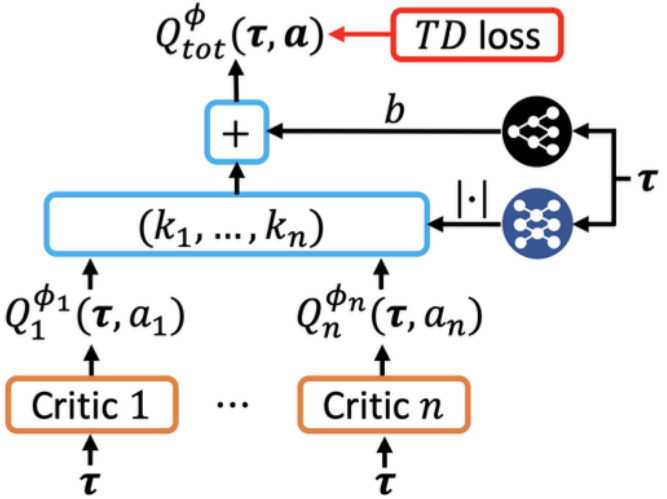




10 agents, 10 actions. Rewarded 10 if all agents take the first action; otherwise -10.

DOP: Off-Policy Decomposed Policy Gradient

- Introducing linearly decomposed critic
  - $Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \cdot) = \sum_i k_i(\boldsymbol{\tau}) Q_i(\boldsymbol{\tau}, \cdot) + b(\boldsymbol{\tau})$
  - where $k_i(\boldsymbol{\tau}) > \mathbf{0}$

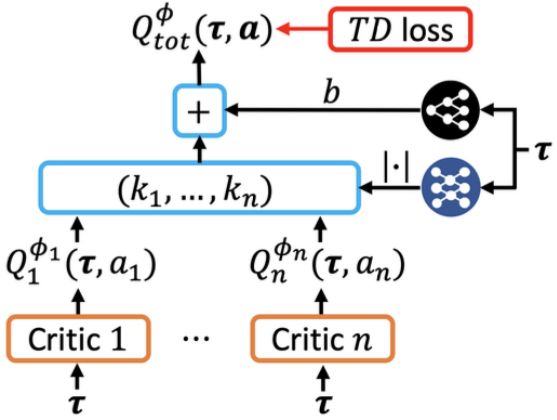


- Benefits:
  - Simple policy update rules
  - Tractable off-policy learning
  - Convergence guarantees
  - Addressing centralized-decentralized mismatch (CDM)

DOP Policy Gradient Theorem

Policy gradient theorem

$$\nabla J(\theta) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_i \nabla_\theta \log \pi_i(a_i|\tau_i) k_i(\boldsymbol{\tau}) Q_i\ (\boldsymbol{\tau}, a_i)\right]$$



Proof

$$\begin{aligned} U_i(\boldsymbol{\tau}, a_i) &= Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) - \textstyle\sum_x \pi_i(x|\tau_i)\, Q_{tot}^{\boldsymbol{\pi}}\big(\boldsymbol{\tau}, (x, \boldsymbol{a}_{-i})\big) \\ &= \textstyle\sum_j k_j(\boldsymbol{\tau})\, Q_j\ (\boldsymbol{\tau}, a_j) - \sum_x \pi_i(x|\tau_i)\left[\sum_{j\neq i} k_j(\boldsymbol{\tau}) Q_j\ (\boldsymbol{\tau}, a_j) + k_i(\boldsymbol{\tau}) Q_i\ (\boldsymbol{\tau}, x)\right] \\ &= k_i(\boldsymbol{\tau})[Q_i\ (\boldsymbol{\tau}, a_i) - \textstyle\sum_x \pi_i(x|\tau_i)\, Q_i\ (\boldsymbol{\tau}, x)] \end{aligned}$$

$$\begin{aligned} \nabla J(\theta) &= \mathbb{E}_{\boldsymbol{\pi}}[\textstyle\sum_i \nabla_\theta \log \pi_i(a_i|\tau_i)\, U_i\ (\boldsymbol{\tau}, a_i)] \\ &= \mathbb{E}_{\boldsymbol{\pi}}\big[\textstyle\sum_i \nabla_\theta \log \pi_i(a_i|\tau_i) k_i(\boldsymbol{\tau})\big(Q_i\ (\boldsymbol{\tau}, a_i) - \sum_x \pi_i(x|\tau_i)\, Q_i\ (\boldsymbol{\tau}, x)\big)\big] \\ &= \mathbb{E}_{\boldsymbol{\pi}}[\textstyle\sum_i \nabla_\theta \log \pi_i(a_i|\tau_i) k_i(\boldsymbol{\tau}) Q_i\ (\boldsymbol{\tau}, a_i)] \end{aligned}$$

Off-Policy Learning

- Evaluate value functions using off-policy data:
  - Estimate $Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})$ using data collected by a behavior policy β.
- Popular techniques in single-agent settings:
  - Importance sampling:
    - Require computing $\prod_i \frac{\pi_i(a_i|\tau_i)}{\beta_i(a_i|\tau_i)}$ in multi-agent settings.
    - The variance grows exponentially with the number of agents.

Off-Policy Evaluation with Linear Decomposition

- Evaluate value functions using off-policy data:
  - Estimate $Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})$ using data collected by a behavior policy β.
- Popular techniques in single-agent settings:
  - Tree-backup (Sutton & Barto (2018 edition), section 7.5) :
    - Require computing $\mathbb{E}_{\boldsymbol{\pi}}[Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \cdot)]$ in multi-agent settings;
    - Need a summation for every joint action; the complexity is exponential;
    - Fortunately, using linearly decomposed critics, this expectation can be computed in **linear time**:

$$\mathbb{E}_{\boldsymbol{\pi}}[Q_{tot}^{\boldsymbol{\pi}}(\tau, \cdot)] = \sum_i k_i(\boldsymbol{\tau}) \mathbb{E}_{\pi_i}[Q_i(\boldsymbol{\tau}, \cdot)] + b(\boldsymbol{\tau})$$

Other Properties of DOP

- Policy Improvement Theorem

Theorem 2. *[Stochastic DOP policy improvement theorem] For any pre-update policy $\boldsymbol{\pi}^o$ which is updated by Eq. 9 to $\boldsymbol{\pi}$, let $\pi_i(a_i|\tau_i) = \pi_i^o(a_i|\tau_i) + \beta_{a_i,\boldsymbol{\tau}}\delta$, where $\delta > 0$ is a sufficiently small number. If it holds that $\forall \tau, a_i', a_i, Q_i^{\phi_i}(\boldsymbol{\tau}, a_i) > Q_i^{\phi_i}(\boldsymbol{\tau}, a_i') \iff \beta_{a_i,\boldsymbol{\tau}} \geq \beta_{a_i',\boldsymbol{\tau}}$, then we have*

$$J(\boldsymbol{\pi}) \geq J(\boldsymbol{\pi}^o),$$

i.e., the joint policy is improved by the update.

- Attenuating Centralized-Decentralized Mismatch

Theorem 3. *Denote r.v.s* $g_1 = \nabla_{\theta_i} \log \pi_i(a_i|\tau_i;\theta_i) Q_{tot}^{\phi}(\boldsymbol{\tau}, \mathbf{a})$, $g_2 = k_i(\boldsymbol{\tau}) \nabla_{\theta_i} \log \pi_i(a_i|\tau_i;\theta_i) Q_i^{\phi_i}(\boldsymbol{\tau}, a_i)$, *under any* $\boldsymbol{\tau}$ *we have*

$$\frac{\mathrm{Var}_{\pi_i}(g_2)}{\mathrm{Var}_{\boldsymbol{\pi}}(g_1)} = O(\frac{1}{n}). \tag{11}$$

DOP Deterministic Policy Gradient

- A similar theorem can be derived for the deterministic case:

$$\nabla J(\theta) = \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{D}}\left[\sum_i \nabla_\theta \pi_i(\tau_i)\, \nabla_{a_i} k_i(\boldsymbol{\tau}) Q_i\,(\boldsymbol{\tau}, a_i)|_{a_i=\pi_i(\tau_i)}\right]$$

Proof. Drawing inspirations from the single-agent case [silver et. al, 2014].

$$\begin{aligned}\nabla J(\theta) &= \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{D}}[\nabla_\theta Q_{tot}\,(\boldsymbol{\tau}, \boldsymbol{a}\,)] \\ &= \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{D}}\left[\textstyle\sum_i \nabla_\theta k_i(\boldsymbol{\tau}) Q_i\,(\boldsymbol{\tau}, a_i)|_{a_i=\pi_i(\tau_i)}\ \right] \\ &= \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{D}}\left[\sum_i \nabla_\theta \pi_i(\tau_i)\, \nabla_{a_i} k_i(\boldsymbol{\tau}) Q_i\,(\boldsymbol{\tau}, a_i)|_{a_i=\pi_i(\tau_i)}\right]\end{aligned}$$

DOP Deterministic Policy Gradient

- Sufficient Representational Capability

Theorem 5. *For* $\forall \boldsymbol{\tau}, \boldsymbol{a} \in \{\boldsymbol{a} | \ ||\boldsymbol{a} - \boldsymbol{\pi}(\boldsymbol{\tau})|| \leq \delta\}$, *there are infinite tuples of feasible* $Q_i^{\phi_i}(\boldsymbol{\tau}, a_i)$, *s.t.*

$$|Q_{tot}^{\phi}(\boldsymbol{\tau}, \boldsymbol{a}) - Q_{tot}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})| \leq 2Ln\delta = O(n\delta), \tag{14}$$

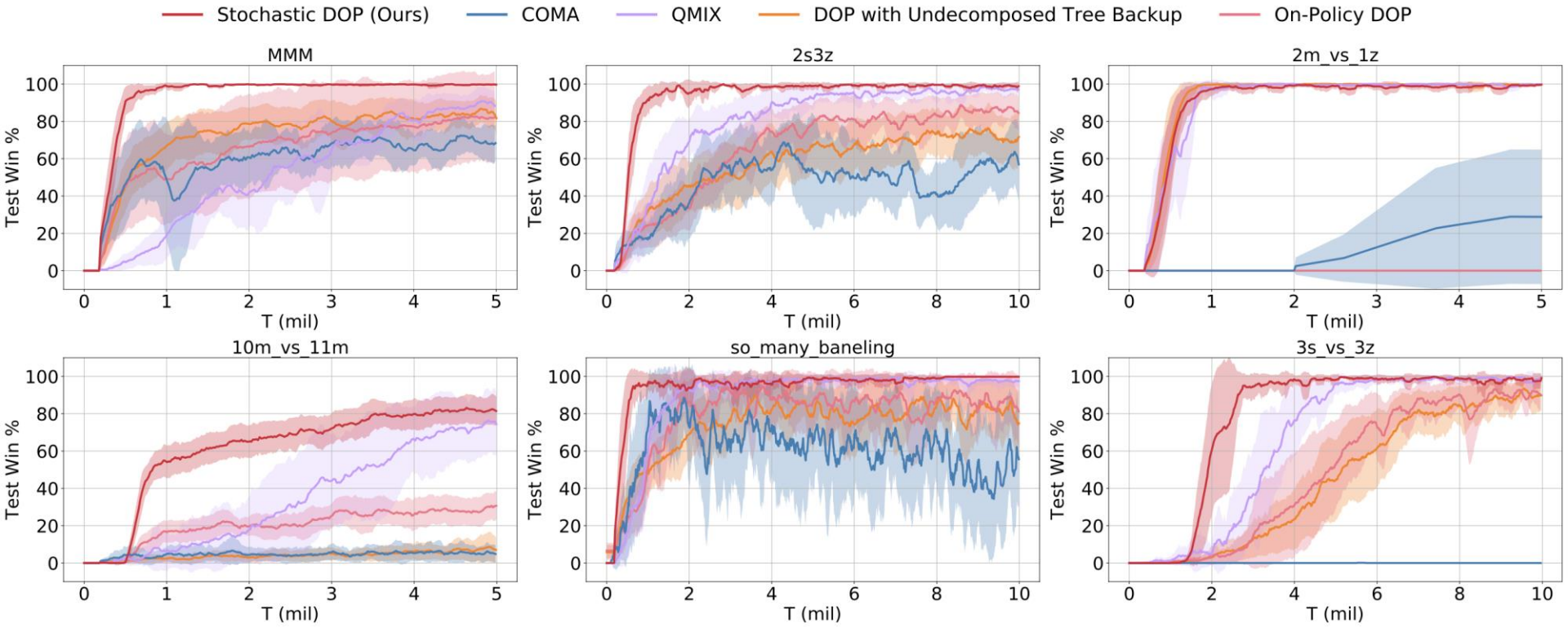
where $Q_{tot}^{\phi}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{n} k_i(\boldsymbol{\tau}) Q_i^{\phi_i}(\boldsymbol{\tau}, a_i) + b(\boldsymbol{\tau})$.

- Attenuating Centralized-Decentralized Mismatch

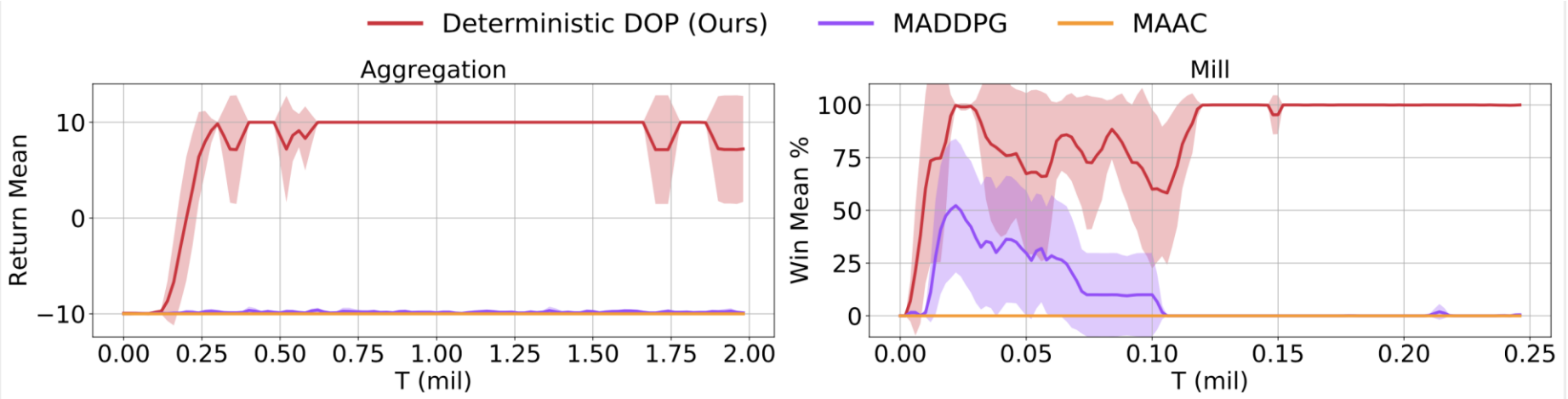
Theorem 6. *Denote r.v.s* $g_1 = \nabla_{\theta_i} \pi_i(\tau_i; \theta_i) \nabla_{a_i} Q_{tot}^{\phi}(\boldsymbol{\tau}, \boldsymbol{a})$, $g_2 = \nabla_{\theta_i} \pi_i(\tau_i; \theta_i) k_i(\boldsymbol{\tau}) \nabla_{a_i} Q_i^{\phi_i}(\boldsymbol{\tau}, a_i)$. *Use* μ_i *to denote the distribution of* a_i', *which is the action of agent* i *accompanied by an exploration noise* $\epsilon \sim P_\epsilon$, *and use* $\boldsymbol{\mu}$ *to denote the joint distribution of all* a_i'. *Under any* $\boldsymbol{\tau}$ *we have:*

$$\frac{\mathrm{Var}_{\mu_i}(g_2)}{\mathrm{Var}_{\boldsymbol{\mu}}(g_1)} = O(\frac{1}{n}). \tag{15}$$

State of the art on SC2 Benchmark



Continuous Action Spaces



Multi-Agent Particle Environment (MPE)

Summary

- Value-Based Methods
  - Paradigm: centralized training with decentralized execution
  - Methods: VDN, QMIX, QPLEX, NDQ, ROMA
- Policy Gradient Methods
  - Paradigm: centralized critic and decentralized actors
  - DOP: off-policy decomposed multi-agent policy gradient
- Take-away
  - Value factorization or decomposition is very useful
  - Dynamic shared learning + communication for complex tasks
  - MARL plays a critical role for AI, but is at the early stage

Challenges in MARL

- Exploration (e.g., sparse interaction)
- Scalability (e.g., number of agents and large action spaces)
- Hierarchical learning (e.g., long horizon problems)
- Decentralized or semi-centralized training
- Non-stationary environments
- Adversarial environments
- Mixed environments (e.g., social dilemma)
- Communication emergence
- Theoretical analysis
- …

References

- Sunehag, et. al.. Value-decomposition networks for cooperative multi-agent learning. *(AAMAS 2018)*
- Rashid, et. al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. (ICML 2018)
- Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, Chongjie Zhang. QPLEX: Duplex Dueling Multi-Agent Q-Learning. arXiv:2008.01062. 2020
- Wang, Jianhao, Zhizhou Ren, Beining Han, and Chongjie Zhang. "Towards Understanding Linear Value Decomposition in Cooperative Multi-Agent Q-Learning." arXiv preprint arXiv:2006.00587 (2020).
- Wang, T., Wang, J., Zheng, C. and Zhang, C., 2019. Learning Nearly Decomposable Value Functions Via Communication Minimization. (ICLR 2020)
- Wang, Tonghan, Heng Dong, Victor Lesser, and Chongjie Zhang. "ROMA: Multi-Agent Reinforcement Learning with Emergent Roles." In Proceedings of the 37th International Conference on Machine Learning. 2020.

References

- Son, Kyunghwan, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning." arXiv preprint arXiv:1905.05408 (2019).
- Yang, Yaodong, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. "Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning." arXiv preprint arXiv:2002.03939 (2020).
- Foerster, J.N., Farquhar, G., Afouras, T., Nardelli, N. and Whiteson, S., Counterfactual multi-agent policy gradients. (AAAI 2018).
- Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Abbeel, O.P. and Mordatch, I., Multi-agent actor-critic for mixed cooperative-competitive environments. (NIPS 2017).
- Iqbal, S. and Sha, F., Actor-attention-critic for multi-agent reinforcement learning. (ICML 2019).
- Wang, Yihan, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. "Off-Policy Multi-Agent Decomposed Policy Gradients." arXiv preprint arXiv:2007.12322 (2020).