ResQ : A Residual Q Function-based Approach for Multi-Agent Reinforcement Learning Value Factorization

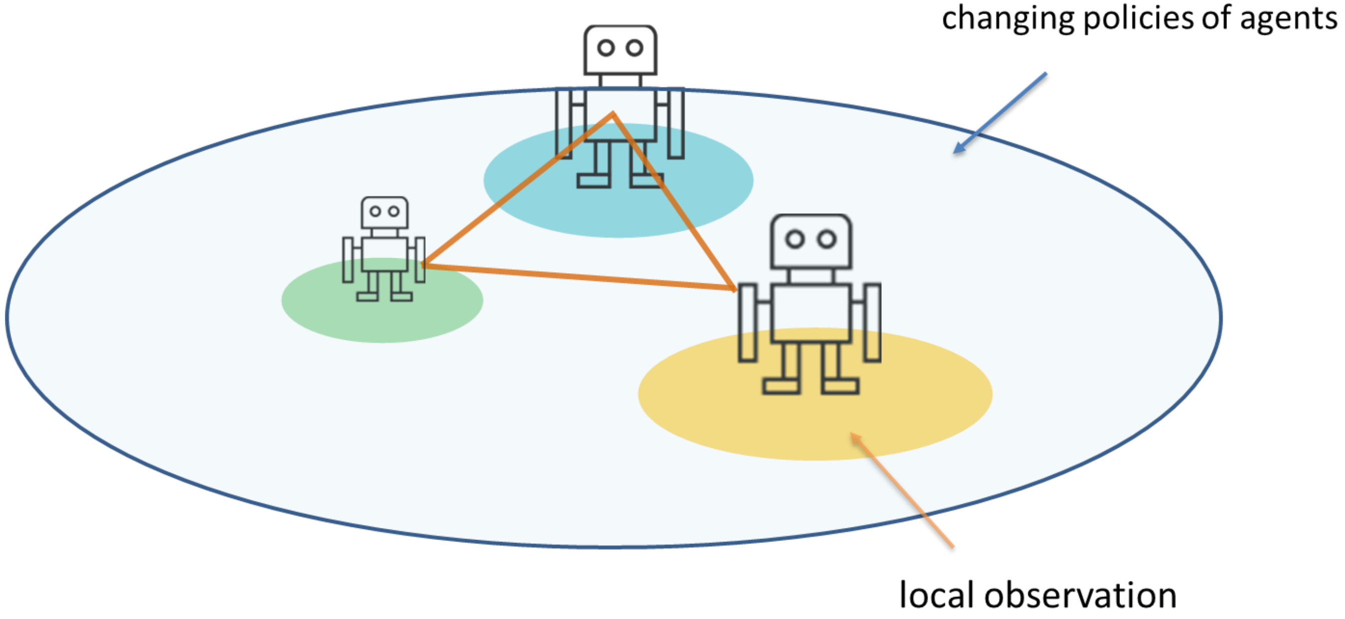
Siqi Shen†, Mengwei Qiu†, Jun Liu†, Weiquan Liu†, Yongquan Fu‡∗, Xinwang Liu‡, Cheng Wang†

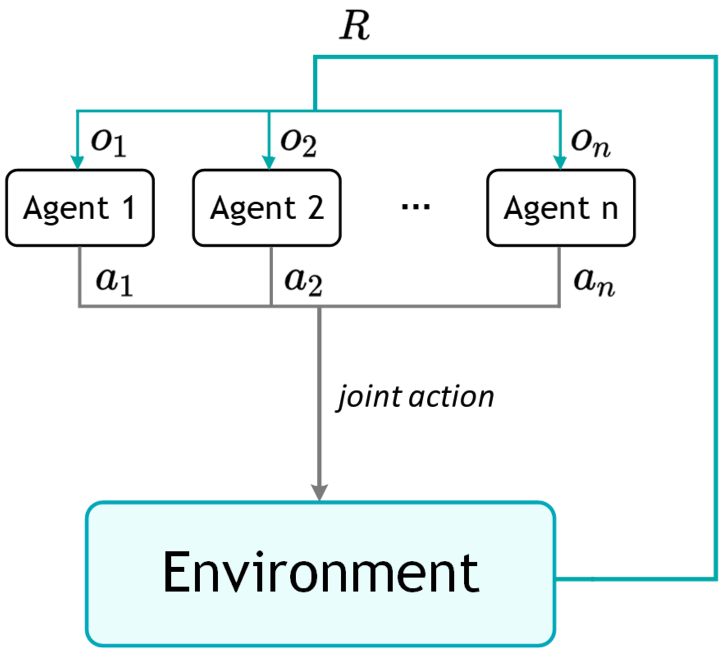
siqishen@xmu.edu.cn, mengweiqiu@stu.xmu.edu.cn, yongquanf@nudt.edu.cn

† Xiamen University
‡ National University of Defense Technology

NeurIPS 2022

Challenges in MARL





Centralized Training with Decentralized Execution paradigm (CTDE)

Value Factorization

- ***IGM theorem:***

$$\arg\max_{\mathbf{u}} Q_{\mathrm{jt}}(\boldsymbol{\tau}, \mathbf{u})$$
$$= (\arg\max_{u_1} Q_1(\tau_1, u_1), \ \ldots, \ \arg\max_{u_n} Q_n(\tau_n, u_n))$$

VDN

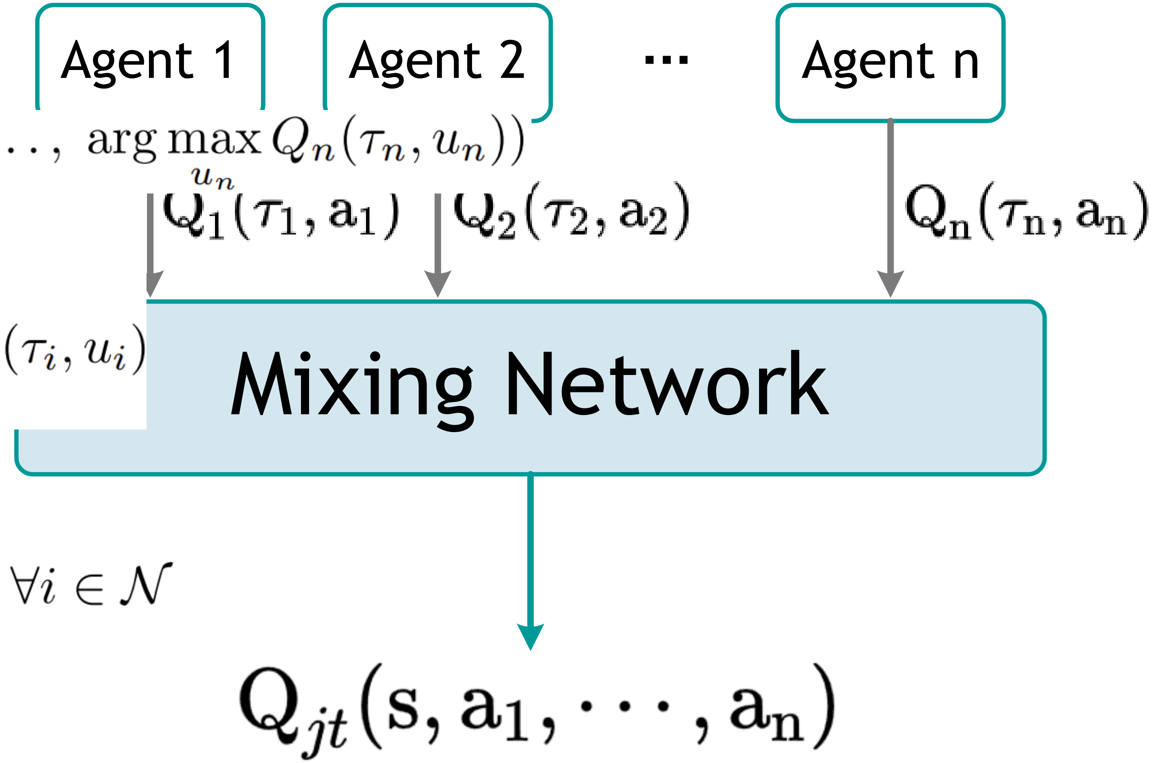
$$Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) = \sum_{i=1}^{N} Q_i(\tau_i, u_i)$$

QMIX

$$\frac{\partial Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u})}{\partial Q_i(\tau_i, u_i)} \geq 0, \quad \forall i \in \mathcal{N}$$

QTRAN

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



Sunehag et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In AAMAS, 2018.
Rashid et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In ICML, 2018.
Son et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In ICML, 2019.

Motivating Example

A one-step two agent game.

Agent 2

Agent 1

u_1 \ u_2	A	B	C
A	8	-12	-12
B	-12	0	0
C	-12	0	7.9

Q_{jt}

=

u_1 \ u_2	A	B	C
A	8	-12	-12
B	0	0	0
C	0	0	0

Q_{tot}

+

u_1 \ u_2	A	B	C
A	0	0	0
B	-12	0	0
C	-12	0	7.9

Q_r

Mask out these red numbers

Main Function
Easy to be factorized

Residual function
Store the mask-out values

$$Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = w_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) + w_r(\boldsymbol{\tau}, \boldsymbol{u}) Q_r(\boldsymbol{\tau}, \boldsymbol{u})$$

Q_{tot} shares the same greedy optimal policy as Q_{jt}.

ResQ

$$Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = w_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) + w_r(\boldsymbol{\tau}, \boldsymbol{u}) Q_r(\boldsymbol{\tau}, \boldsymbol{u})$$

Q_{tot} shares the same greedy optimal policy as Q_{jt}.

Main Function:

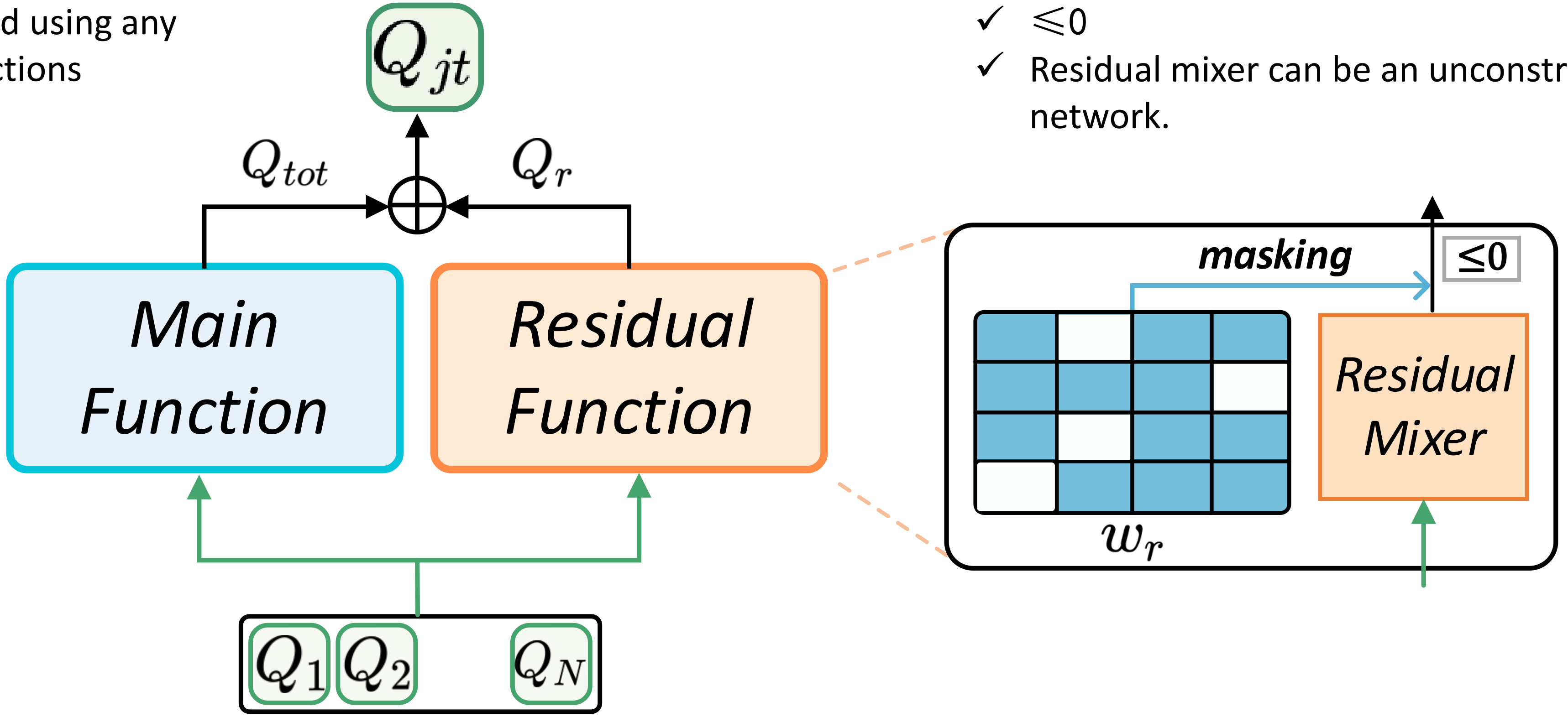
✓ For easy-to-factorize parts.

✓ Can be modelled using any monotonic functions

Residual Function:

✓ ⩽0

✓ Residual mixer can be an unconstrained network.



We focus on $Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) + w_r(\boldsymbol{\tau}, \boldsymbol{u}) Q_r(\boldsymbol{\tau}, \boldsymbol{u})$

ResQ can viewed as a generalization of QTran, Weight QMIX, QPLEX, DDN, and DMIX

Theoretical Analysis of ResQ

Satisfy the IGM Theorem without representation limitations

Theorem 1. *A joint state-action function*

$$Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) + w_r(\boldsymbol{\tau}, \boldsymbol{u})Q_r(\boldsymbol{\tau}, \boldsymbol{u}) \tag{5}$$

is factorized by $[Q_i(\tau_i, u_i)]_{i=1}^N$, *if* $Q_r(\boldsymbol{\tau}, \boldsymbol{u}) \leq 0$, $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$ *and* $[Q_i(\tau_i, u_i)]_{i=1}^N$ *satisfy the monotonicity conditions (2), and*

$$w_r(\boldsymbol{\tau}, \boldsymbol{u}) = \begin{cases} 0 & \boldsymbol{u} = \bar{\boldsymbol{u}}, & \text{(6a)} \\ 1 & \boldsymbol{u} \neq \bar{\boldsymbol{u}}, & \text{(6b)} \end{cases}$$

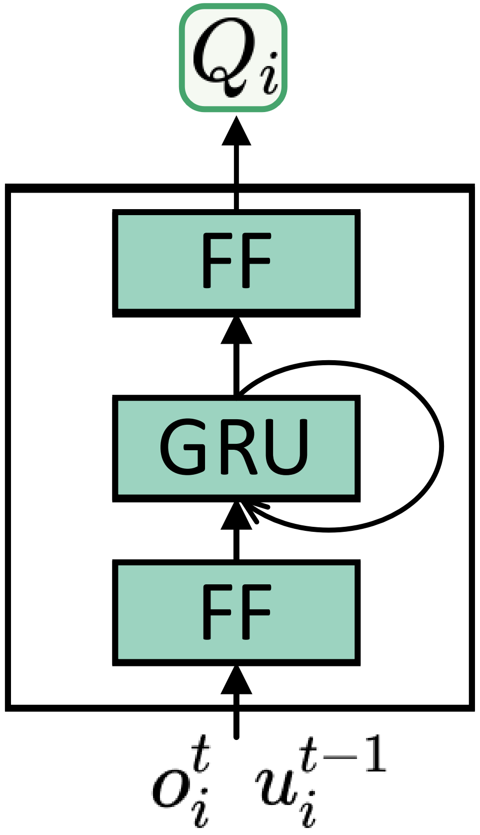
Theorem 2. *For any joint state-action function* $Q(\boldsymbol{\tau}, \boldsymbol{u})$, *we can find* $Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) = Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) + w_r(\boldsymbol{\tau}, \boldsymbol{u})Q_r(\boldsymbol{\tau}, \boldsymbol{u})$ *that*

$$\bar{u} = \arg\max_{u} Q(\boldsymbol{\tau}, \boldsymbol{u}) = \arg\max_{u} Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) \tag{7}$$

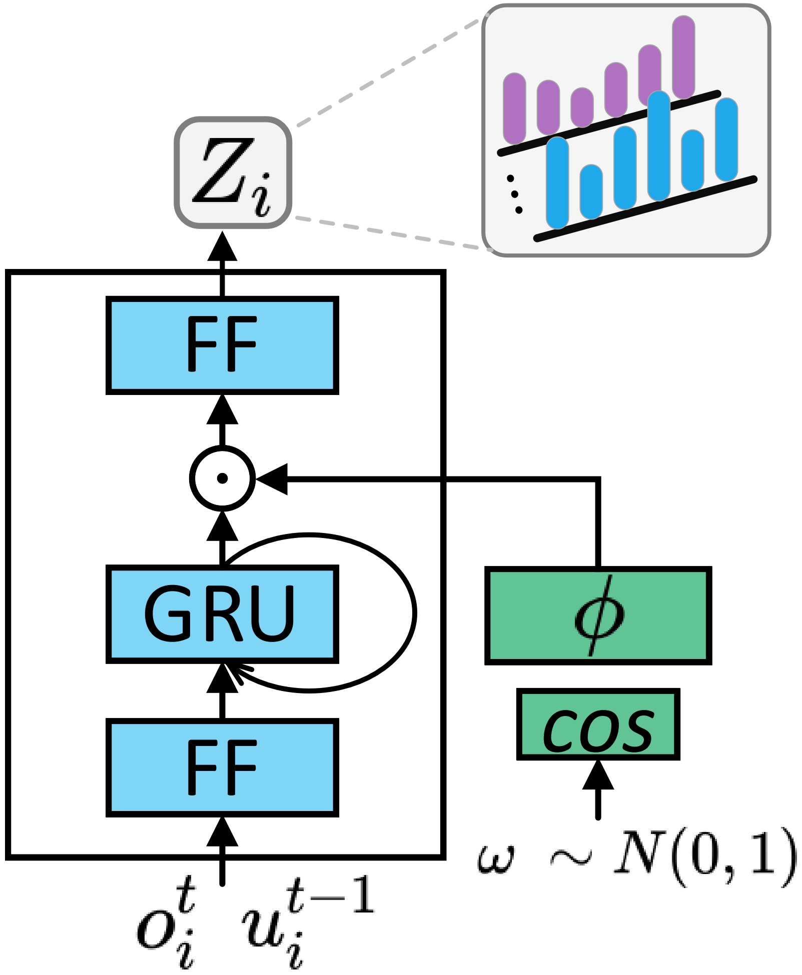
$$Q(\boldsymbol{\tau}, \boldsymbol{u}) = Q_{jt}(\boldsymbol{\tau}, \boldsymbol{u}) \quad \forall \boldsymbol{u} \neq \bar{\boldsymbol{u}} \tag{8}$$

$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$ *monotonically increases with* $[Q_i(\tau_i, u_i)]_{i=1}^N$, $w_r(\boldsymbol{\tau}, \boldsymbol{u})$ *satisfies (6), and* $Q_r(\boldsymbol{\tau}, \boldsymbol{u}) \leq 0$.

Extending ResQ to Distributional RL



- Deterministic agent network



- Stochastic agent network

Extending ResQ to Distributional RL

- ***DIGM theorem:***

$$\arg\max_{\mathbf{u}} \mathbb{E}[Z_{jt}(\boldsymbol{\tau}, \boldsymbol{u})] = (\arg\max_{u_1} \mathbb{E}[Z_1(\tau_1, u_1)], \ \ldots, \ \arg\max_{u_n} \mathbb{E}[Z_n(\tau_n, u_n)])$$

DDN / DMIX

(Mean-Shape Decomposition)

$$\begin{aligned} Z &= \mathbb{E}[Z] + (Z - \mathbb{E}[Z]) \\ &= Z_{\text{mean}} + Z_{\text{shape}} \ , \end{aligned}$$

(DDN) $Z_{\text{mean}} = \sum_{k \in \mathbb{K}} Q_k, \ Z_{\text{shape}} = \sum_{k \in \mathbb{K}} (Z_k - Q_k)$

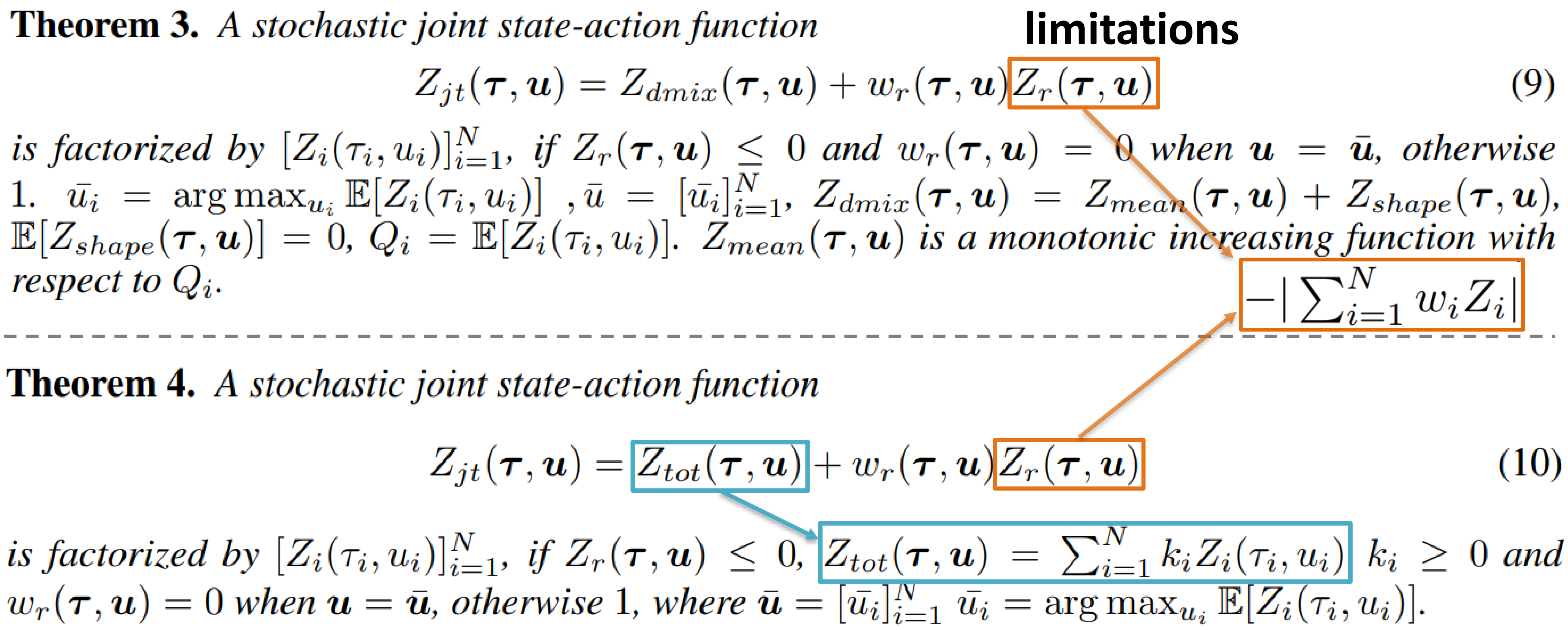
(DMIX) $Z_{\text{mean}} = M(Q_1, \ldots, Q_{\mathrm{K}} | s), \ Z_{\text{shape}} = \sum_{k \in \mathbb{K}} (Z_k - Q_k)$

DDN and DMIX suffer from representation limitations

Sun et al. DFAC framework: Factorizing the value function via quantile mixture for multi-agent distributional q-learning. In ICML, 2021.

Extending ResQ to Distributional RL

Satisfy the DIGM Theorem without representation limitations



Theorem 3. *A stochastic joint state-action function*

$$Z_{jt}(\boldsymbol{\tau},\boldsymbol{u}) = Z_{dmix}(\boldsymbol{\tau},\boldsymbol{u}) + w_r(\boldsymbol{\tau},\boldsymbol{u})\boxed{Z_r(\boldsymbol{\tau},\boldsymbol{u})} \tag{9}$$

is factorized by $[Z_i(\tau_i,u_i)]_{i=1}^{N}$, *if* $Z_r(\boldsymbol{\tau},\boldsymbol{u}) \leq 0$ *and* $w_r(\boldsymbol{\tau},\boldsymbol{u}) = 0$ *when* $\boldsymbol{u} = \bar{\boldsymbol{u}}$, *otherwise* 1. $\bar{u_i} = \arg\max_{u_i} \mathbb{E}[Z_i(\tau_i,u_i)]$, $\bar{u} = [\bar{u_i}]_{i=1}^{N}$, $Z_{dmix}(\boldsymbol{\tau},\boldsymbol{u}) = Z_{mean}(\boldsymbol{\tau},\boldsymbol{u}) + Z_{shape}(\boldsymbol{\tau},\boldsymbol{u})$, $\mathbb{E}[Z_{shape}(\boldsymbol{\tau},\boldsymbol{u})] = 0$, $Q_i = \mathbb{E}[Z_i(\tau_i,u_i)]$. $Z_{mean}(\boldsymbol{\tau},\boldsymbol{u})$ *is a monotonic increasing function with respect to* Q_i.

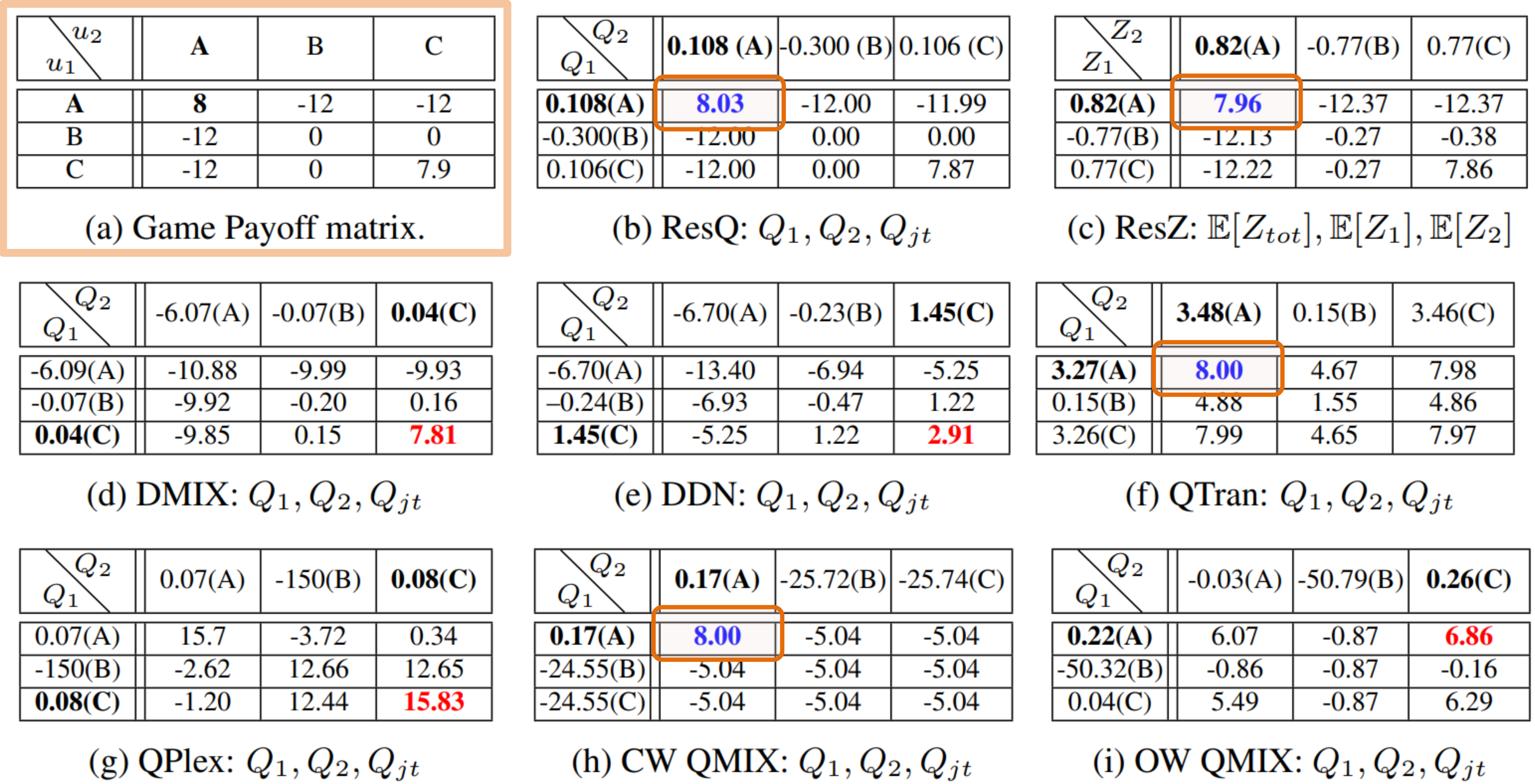
$\boxed{-|\sum_{i=1}^{N} w_i Z_i|}$

Theorem 4. *A stochastic joint state-action function*

$$Z_{jt}(\boldsymbol{\tau},\boldsymbol{u}) = \boxed{Z_{tot}(\boldsymbol{\tau},\boldsymbol{u})} + w_r(\boldsymbol{\tau},\boldsymbol{u})\boxed{Z_r(\boldsymbol{\tau},\boldsymbol{u})} \tag{10}$$

is factorized by $[Z_i(\tau_i,u_i)]_{i=1}^{N}$, *if* $Z_r(\boldsymbol{\tau},\boldsymbol{u}) \leq 0$, $\boxed{Z_{tot}(\boldsymbol{\tau},\boldsymbol{u}) = \sum_{i=1}^{N} k_i Z_i(\tau_i,u_i)}$ $k_i \geq 0$ *and* $w_r(\boldsymbol{\tau},\boldsymbol{u}) = 0$ *when* $\boldsymbol{u} = \bar{\boldsymbol{u}}$, *otherwise* 1, *where* $\bar{\boldsymbol{u}} = [\bar{u_i}]_{i=1}^{N}$ $\bar{u_i} = \arg\max_{u_i} \mathbb{E}[Z_i(\tau_i,u_i)]$.

Experiments —— Matrix game



u_1 \ u_2	A	B	C
A	**8**	-12	-12
B	-12	0	0
C	-12	0	7.9

(a) Game Payoff matrix.

Q_1 \ Q_2	**0.108 (A)**	-0.300 (B)	0.106 (C)
0.108(A)	**8.03**	-12.00	-11.99
-0.300(B)	-12.00	0.00	0.00
0.106(C)	-12.00	0.00	7.87

(b) ResQ: Q_1, Q_2, Q_{jt}

Z_1 \ Z_2	**0.82(A)**	-0.77(B)	0.77(C)
0.82(A)	**7.96**	-12.37	-12.37
-0.77(B)	-12.13	-0.27	-0.38
0.77(C)	-12.22	-0.27	7.86

(c) ResZ: $\mathbb{E}[Z_{tot}], \mathbb{E}[Z_1], \mathbb{E}[Z_2]$

Q_1 \ Q_2	-6.07(A)	-0.07(B)	**0.04(C)**
-6.09(A)	-10.88	-9.99	-9.93
-0.07(B)	-9.92	-0.20	0.16
0.04(C)	-9.85	0.15	**7.81**

(d) DMIX: Q_1, Q_2, Q_{jt}

Q_1 \ Q_2	-6.70(A)	-0.23(B)	**1.45(C)**
-6.70(A)	-13.40	-6.94	-5.25
–0.24(B)	-6.93	-0.47	1.22
1.45(C)	-5.25	1.22	**2.91**

(e) DDN: Q_1, Q_2, Q_{jt}

Q_1 \ Q_2	**3.48(A)**	0.15(B)	3.46(C)
3.27(A)	**8.00**	4.67	7.98
0.15(B)	4.88	1.55	4.86
3.26(C)	7.99	4.65	7.97

(f) QTran: Q_1, Q_2, Q_{jt}

Q_1 \ Q_2	0.07(A)	-150(B)	**0.08(C)**
0.07(A)	15.7	-3.72	0.34
-150(B)	-2.62	12.66	12.65
0.08(C)	-1.20	12.44	**15.83**

(g) QPlex: Q_1, Q_2, Q_{jt}

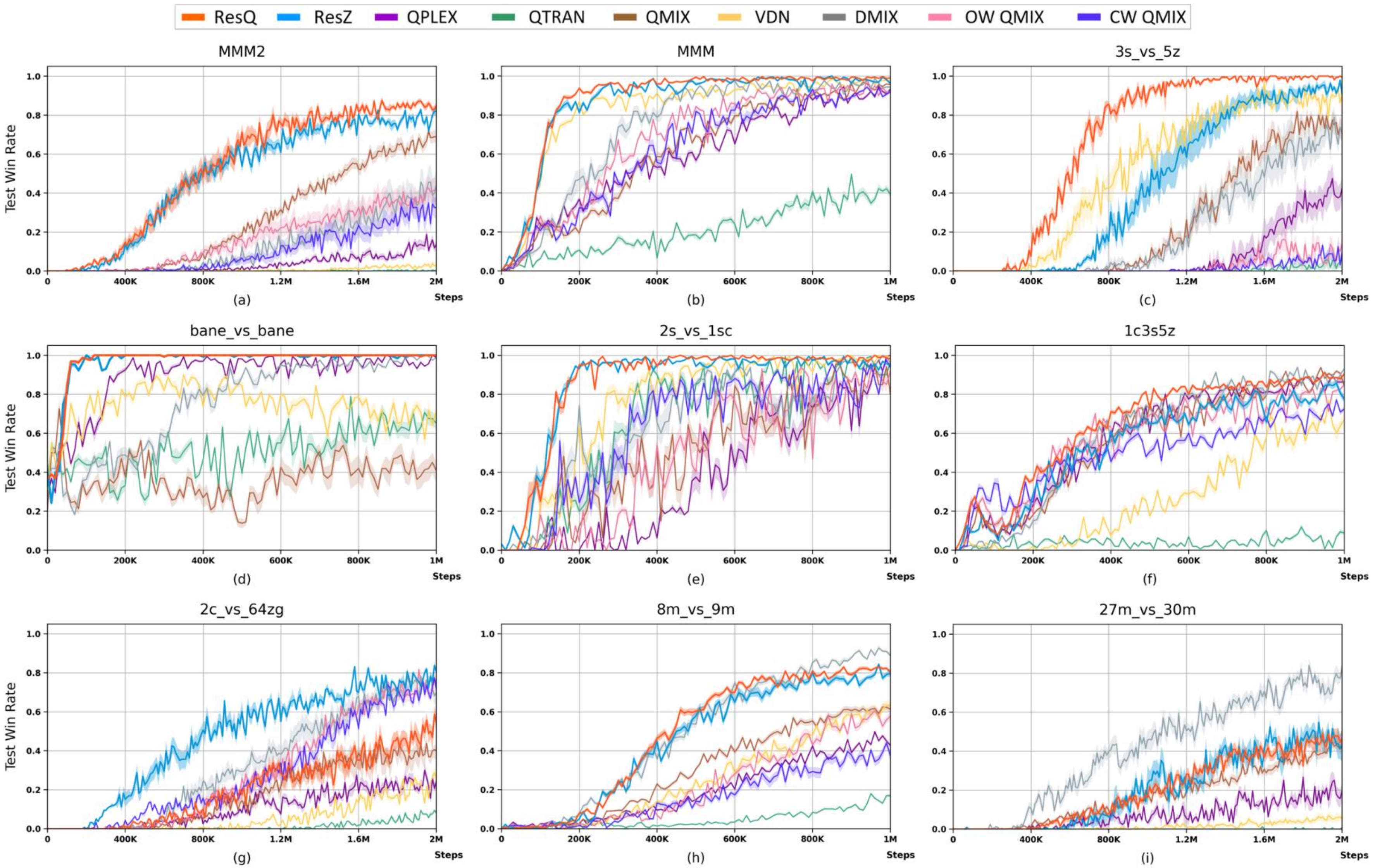
Q_1 \ Q_2	**0.17(A)**	-25.72(B)	-25.74(C)
0.17(A)	**8.00**	-5.04	-5.04
-24.55(B)	-5.04	-5.04	-5.04
-24.55(C)	-5.04	-5.04	-5.04

(h) CW QMIX: Q_1, Q_2, Q_{jt}

Q_1 \ Q_2	-0.03(A)	-50.79(B)	**0.26(C)**
0.22(A)	6.07	-0.87	**6.86**
-50.32(B)	-0.86	-0.87	-0.16
0.04(C)	5.49	-0.87	6.29

(i) OW QMIX: Q_1, Q_2, Q_{jt}

Experiments —— Starcraft II Multi-agent Challenge(SMAC)

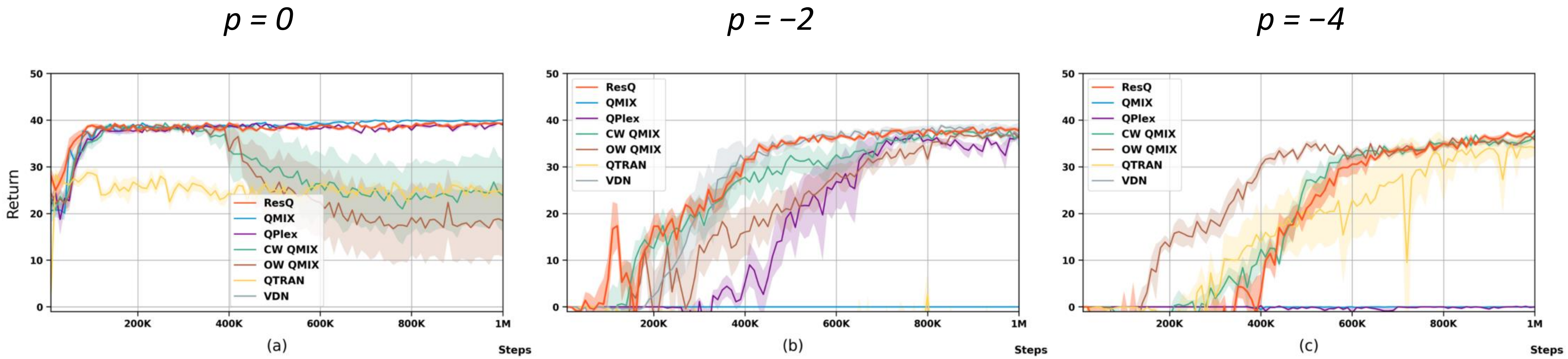


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Experiments —— Predator Prey

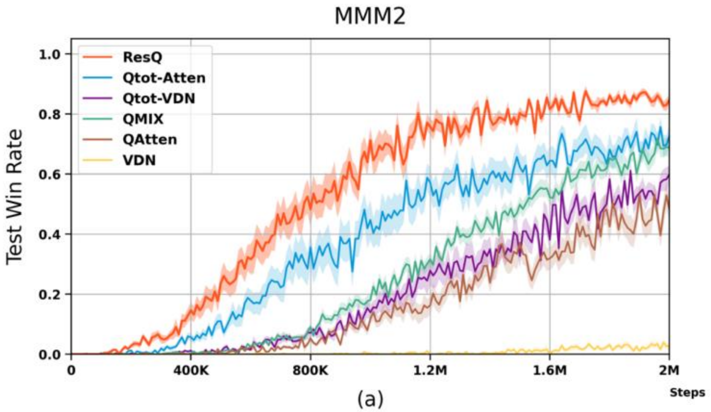


(a)

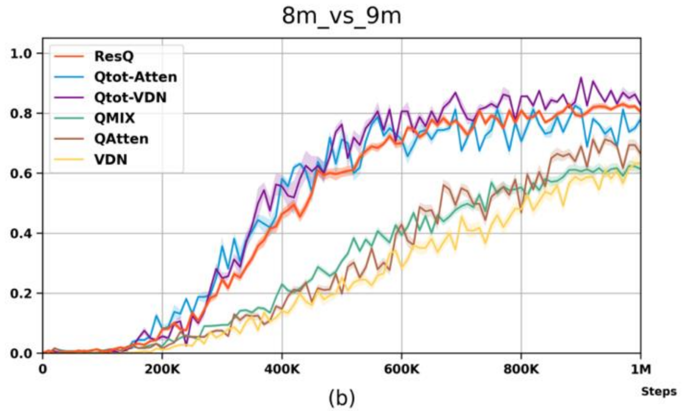
(b)

(c)

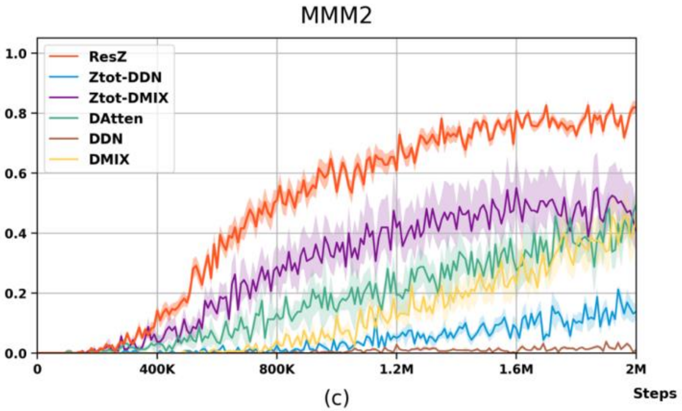
Experiments —— ablations



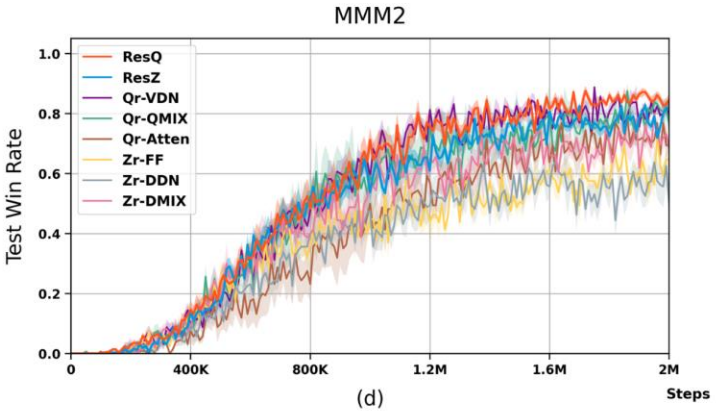
(a)



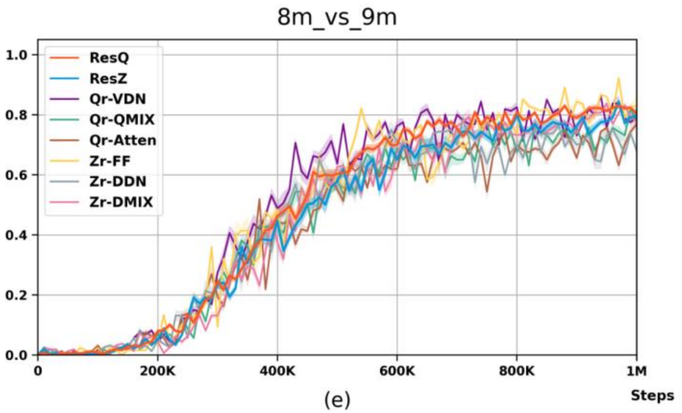
(b)



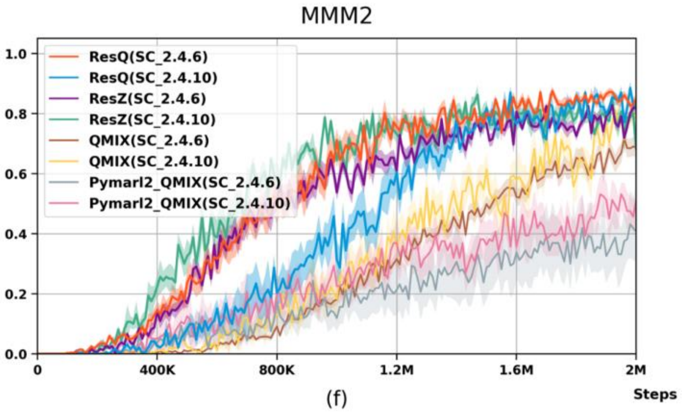
(c)



(d)



(e)



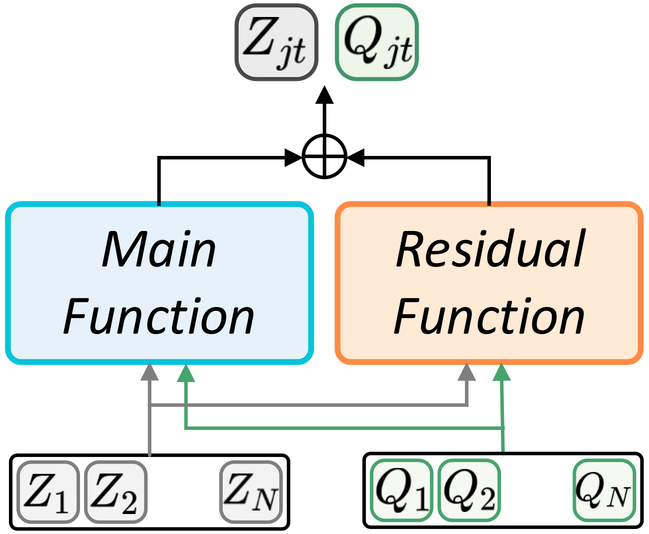
(f)

Summary

- ResQ, a residual function-based approach for Multi-Agent Reinforcement Learning value function factorization.
- Through extensive experiments, we show that ResQ can obtain promising results.

For more details, please check our project page:
https://github.com/xmu-rl-3dv/ResQ

Contact us:
siqishen@xmu.edu.cn
mengweiqiu@stu.xmu.edu.cn
yongquanf@nudt.edu.cn



Thanks for your attention!