

Modelling the Dynamic Joint Policy of Teammates with Attention Multi-agent DDPG

Hangyu Mao, Zhengchao Zhang, Zhen Xiao*, Zhibo Gong

AAMAS 2019



北京大学
PEKING UNIVERSITY



HUAWEI
华为技术有限公司



北京大学

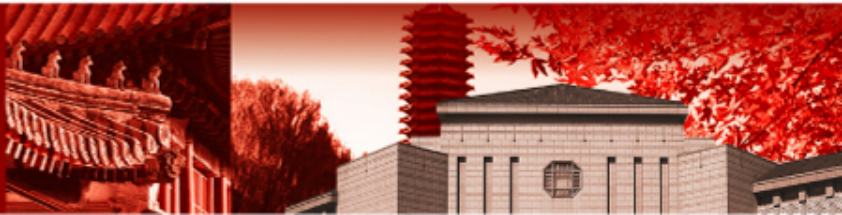


Outline

- **Research Problem**
- Background
- Design
- Evaluation
- Conclusion



北京大學



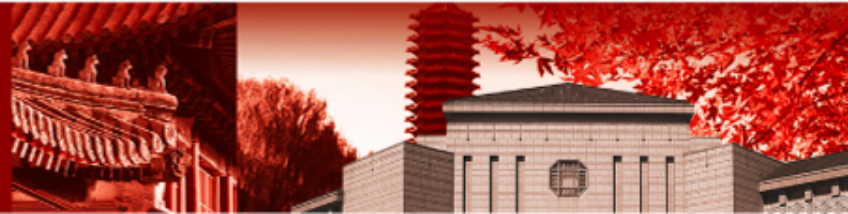
Adaptive Agent Modelling

- **Modelling the policies of teammates** has long been an interest for the Multi-agent Reinforcement Learning (MARL) community.
 - e.g., if the agent knows the policies of the teammates, it can adjust its own policy accordingly to achieve proper cooperation.
- However, teammates modelling is also a big challenge.
 - because **the agents are changing their policies continuously** while they are learning concurrently to adapt to each other.

Therefore, there is a great need to design **adaptive agent modelling** methods.

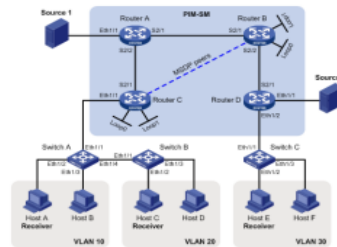
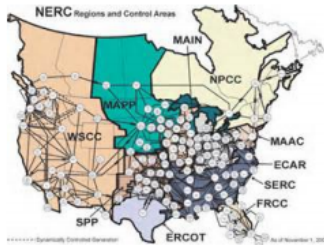


北京大學



Decentralized Policy

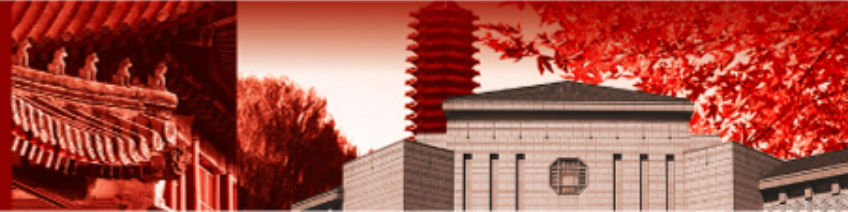
- Many real-world multi-agent tasks require **distributed policies**, since:
 - The agents may be located in different places
 - This may be required in the rules of the games
 -



Thus, there is also a great need to train **decentralized policy** for each agent.



北京大學

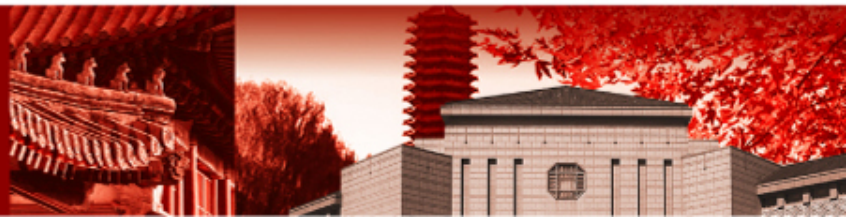


Research Problem

designing an **adaptive agent modelling** method that can train **decentralized policy**



北京大學

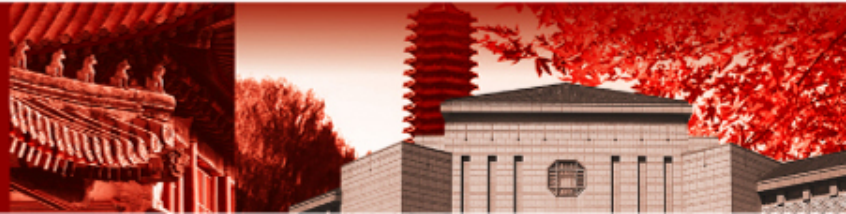


Outline

- Research Problem
- **Background**
- Design
- Evaluation
- Conclusion

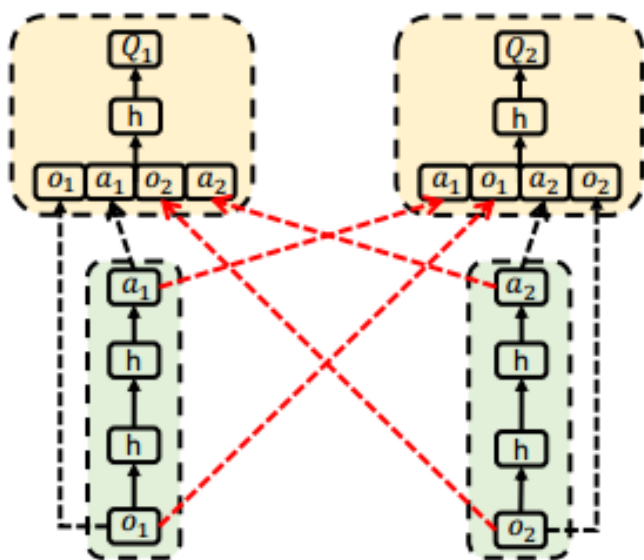


北京大学



MADDPG (our basic model)

- apply centralized critic to train decentralized policy



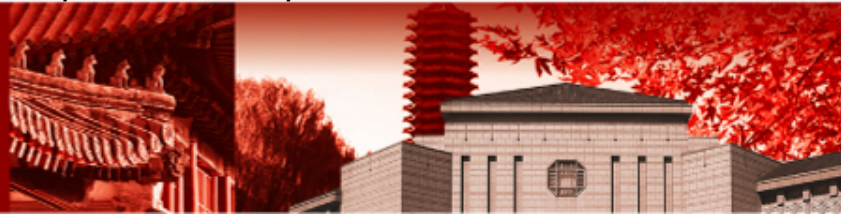
The **centralized critic** Q_i can get access to the observations and actions of all agents.
→ lay the necessary foundation to do agent modelling

The **independent actor** π_i can only get access to its own observation o_i .
→ learn decentralized policies for distributed execution

Lowe R, Wu Y, Tamar A, et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. NIPS2017.



北京大學



Unsolved Problem

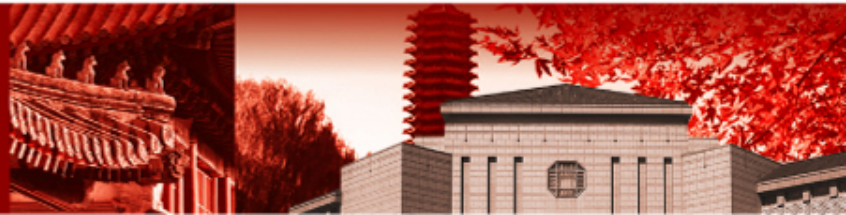
- MADDPG adopts a *fully-connected neural network* as the centralized critic, which is not very adaptive.

adaptive agent modelling?

→ Attention Mechanism!



北京大學



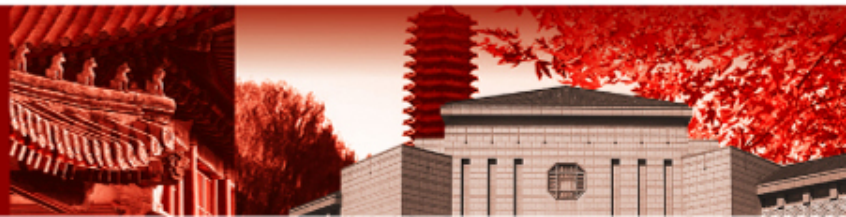
Soft Attention Mechanism

- The inputs are several source vectors $[S_1, S_2, \dots, S_k, \dots, S_K]$ and one target vector T
- The information contained in S_k can be encoded into a contextual vector C **adaptively** according to the normalized importance score w_k as follows:
$$w_k = \frac{\exp(f(T, S_k))}{\sum_{i=1}^K \exp(f(T, S_i))} ; C = \sum_{k=1}^K w_k S_k$$
- Besides, the attention weight vector $W \triangleq [w_1, w_2, \dots, w_k, \dots, w_K]$ can also be seen as **a probability distribution** because $\sum_{k=1}^K w_k \equiv 1$.

In our method, we will design an **attentional centralized critic** to generate **a probability distribution** in an **adaptive** manner!



北京大學

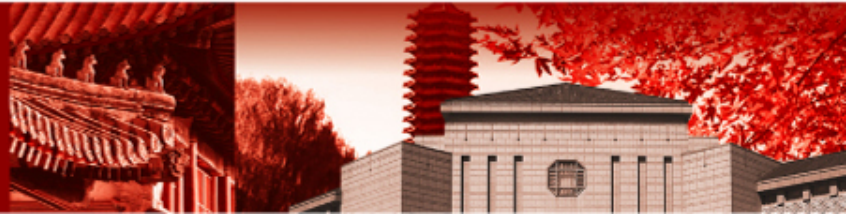


Outline

- Research Problem
- Background
- **Design**
- Evaluation
- Conclusion

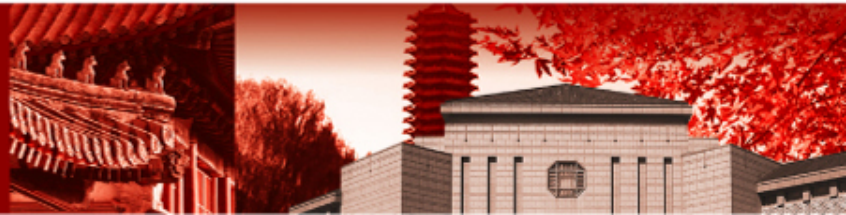


北京大学



Key Variables

\vec{a}	The joint action of all agents.
a_i	The local action of agent i .
\vec{a}_{-i}	The joint action of teammates of agent i .
The action set \vec{A} , A_i , \vec{A}_{-i} are denoted similarly.	
The observation (history) \vec{o} , o_i , \vec{o}_{-i} are denoted similarly.	
The policy $\vec{\pi}$, π_i , $\vec{\pi}_{-i}$ are denoted similarly.	
s'	The next state after s .
\vec{o}' , o'_i , \vec{o}'_{-i} , \vec{a}' , a'_i , and \vec{a}'_{-i} are denoted similarly.	
$\vec{\pi}_{-i}$	The joint policy of teammates of agent i .
$\vec{\pi}_{-i}(\vec{a}_{-i} s)$	The probability value for generating \vec{a}_{-i} under policy $\vec{\pi}_{-i}$. $\sum_{\vec{a}_{-i} \in \vec{A}_{-i}} \vec{\pi}_{-i}(\vec{a}_{-i} s) = 1$.
$\vec{\pi}_{-i}(\vec{A}_{-i} s)$	The probability distribution over the joint action space \vec{A}_{-i} under policy $\vec{\pi}_{-i}$.



Attentional Critic for Adaptability

- Multi-agent Q-value function

- We define *the multi-agent Q-value function relative to the joint policy of teammates* as previous work [10]

Mathematically, $Q_i^{\pi_i|\bar{\pi}^{-i}}(s, a_i)$ can be calculated by¹

$$Q_i^{\pi_i|\bar{\pi}^{-i}}(s, a_i) = \mathbb{E}_{\vec{a}_{-i} \sim \bar{\pi}_{-i}} [Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (6)$$

$$= \sum_{\vec{a}_{-i} \in \vec{A}_{-i}} [\bar{\pi}_{-i}(\vec{a}_{-i}|s) Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (7)$$

- Like single-agent setting, our objective is to find the optimal policy $\pi_i^* = \arg \max_{\pi_i} Q_i^{\pi_i|\bar{\pi}^{-i}}(s, a_i)$

¹The detailed derivation can be found in [10].

Since the outcome of a_i taken in s is dependent on \vec{a}_{-i}

2016-ICML-Opponent modeling in deep reinforcement learning

Single-agent Q-value function:

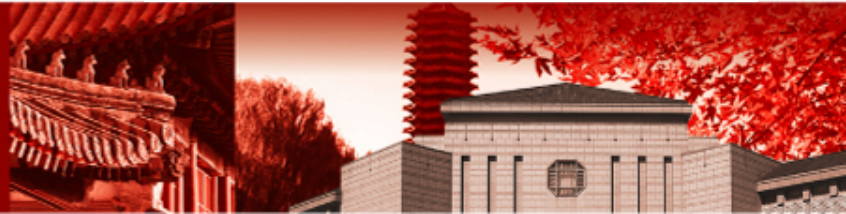
In practice, the Q-value function $Q^\pi(s, a)$ is defined as

$$Q^\pi(s, a) = \mathbb{E}_\pi [G|S = s, A = a] \quad (1)$$

then the optimal policy is derived by $\pi^* = \arg \max_\pi Q^\pi(s, a)$.



北京大學



Attentional Critic for Adaptability

- Multi-agent Q-value function

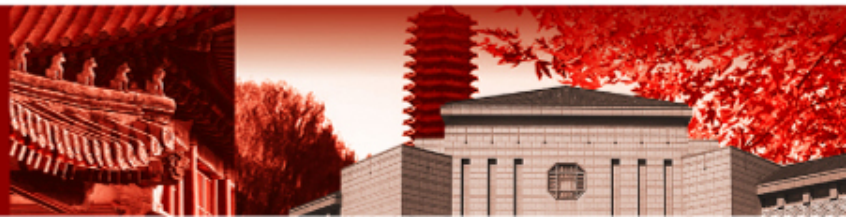
$$Q_i^{\pi_i|\bar{\pi}^{-i}}(s, a_i) = \mathbb{E}_{\vec{a}_{-i} \sim \bar{\pi}^{-i}}[Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (6)$$

$$= \sum_{\vec{a}_{-i} \in \vec{A}_{-i}} [\bar{\pi}^{-i}(\vec{a}_{-i}|s) Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (7)$$

Equation 7 implies that in order to estimate $Q_i^{\pi_i|\bar{\pi}^{-i}}(s, a_i)$, the critic network of agent i should have the abilities:

- (1) to estimate $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ for each $\vec{a}_{-i} \in \vec{A}_{-i}$;
- (2) to calculate the expectation of all $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ ².

²The expectation is equivalent to the weighted summation, and the weight of $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ is $\bar{\pi}^{-i}(\vec{a}_{-i}|s)$ as shown in Equation 7.



Attentional Critic for Adaptability

- (1) To estimate $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ for each $\vec{a}_{-i} \in \vec{A}_{-i}$
- We design a **K-head Module** where $K = |\vec{A}_{-i}|$.
 - As shown in Figure 3, the K-head Module generates K **action conditional Q-value** $Q_i^k(s, a_i | \vec{a}_{-i}; w_i)$ for each \vec{a}_{-i} to approximate the true $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$
 - Specifically, $Q_i^k(s, a_i, \vec{a}_{-i}; w_i)$ is generated using a_i and all observations $\langle o_i, \vec{o}_{-i} \rangle = \vec{o} \triangleq s$
 - As for the information about \vec{a}_{-i} , it is provided by an additional hidden vector $h_i(w_i)$, which will be introduced shortly
 - This is why we use $Q_i^k(s, a_i | \vec{a}_{-i}; w_i)$ instead of $Q_i^k(s, a_i, \vec{a}_{-i}; w_i)$ to represent the defined **action conditional Q-value**.

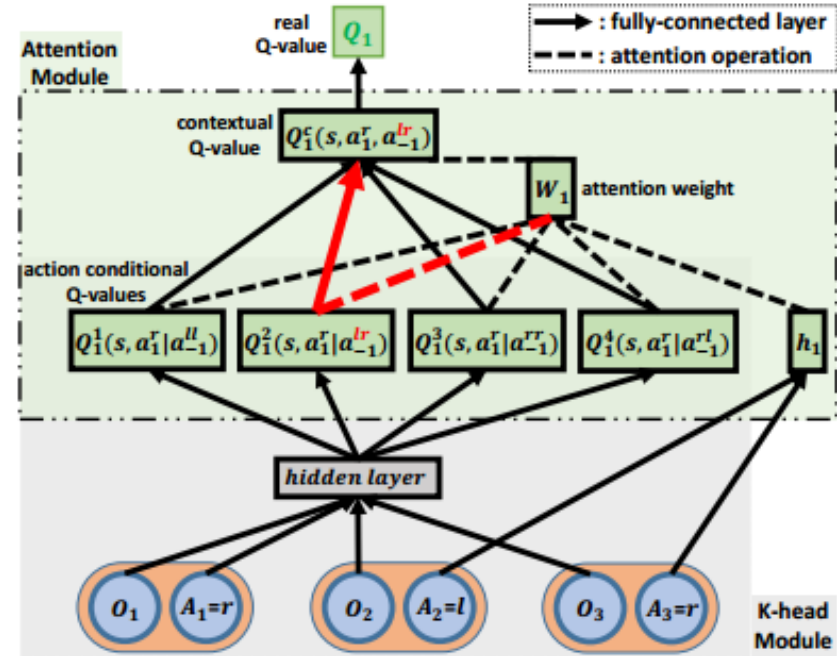


Figure 3: The attention critic of ATT-MADDPG. For clarity, we only show the detailed generation of Q_1 using a three-agent example: the discrete action space is $\{l, r\}$, and the agents prefer to take the actions r, l , and r , respectively. In this case, the second **action conditional Q-value** Q_1^2 will contribute more weights to the computation of the **contextual Q-value** Q_1^c , as indicated by thicker red links. We call Q_i the **real Q-value**, Q_i^c the **contextual Q-value**, and Q_i^k the **action conditional Q-value**. The difference is that Q_i^c and Q_i^k are multi-dimensional vectors, while Q_i is the real scalar Q-value used in Equation 10, 11, and 12.

Equation 7 implies that in order to estimate $Q_i^{\pi_i | \vec{\pi}_{-i}}(s, a_i)$, the critic network of agent i should have the abilities:

- (1) to estimate $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ for each $\vec{a}_{-i} \in \vec{A}_{-i}$;
- (2) to calculate the expectation of all $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ ².

Attentional Critic for Adaptability

(2) To calculate the expectation of all $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$

- The weights $\vec{\pi}_{-i}(\vec{a}_{-i}|s)$ of each $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$ are also required (as indicated by Equation 7)

$$Q_i^{\pi_i|\vec{\pi}_{-i}}(s, a_i) = \mathbb{E}_{\vec{a}_{-i} \sim \vec{\pi}_{-i}}[Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (6)$$

$$= \sum_{\vec{a}_{-i} \in \vec{A}_{-i}} [\vec{\pi}_{-i}(\vec{a}_{-i}|s) Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (7)$$

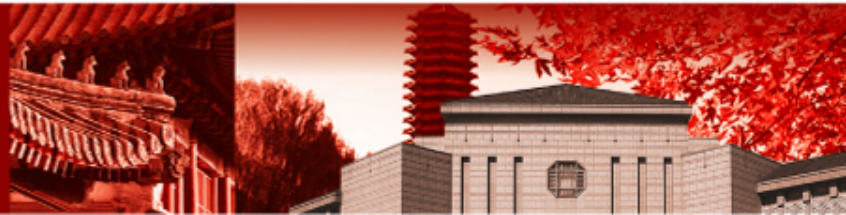


- However, it is hard to approximate these weights.
- On one hand, for different s , the teammates will take different \vec{a}_{-i} with different probabilities $\vec{\pi}_{-i}(\vec{a}_{-i}|s)$ based on the policy $\vec{\pi}_{-i}$.
- On the other hand, the policy $\vec{\pi}_{-i}$ is changing continuously, because the agents are learning concurrently to adapt to each other.

It's time for Attention!



北京大學



Attentional Critic for Adaptability

(2) To calculate the expectation of all $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$

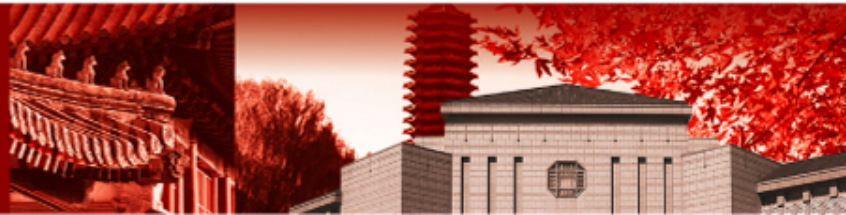
- Difficulties: $\rightarrow \vec{\pi}_{-i}(\vec{a}_{-i}|s)$ is different for different s . $\rightarrow \vec{\pi}_{-i}$ itself is changing continuously.
- We approximate all $\vec{\pi}_{-i}(\vec{a}_{-i}|s) \in \vec{\pi}_{-i}(\vec{A}_{-i}|s)$ *jointly* by a weight vector $W_i \triangleq [W_i^1, \dots, W_i^K]$.
 - That is to say, we use W_i to approximate the *probability distribution* $\vec{\pi}_{-i}(\vec{A}_{-i}|s)$ rather than approximating each *probability value* $\vec{\pi}_{-i}(\vec{a}_{-i}|s)$ separately.
 - A good W_i should satisfy the following conditions:
 - $\sum_{k=1}^K W_i^k \equiv 1$, such that $W_i \triangleq [W_i^1, \dots, W_i^K]$ **is a probability distribution indeed**;
 - $W_i \triangleq [W_i^1, \dots, W_i^K]$ **can change adaptively** when the joint policy of teammates $\vec{\pi}_{-i}$ is changed, such that W_i can really model the teammates' joint policy in an adaptive manner.

It's time for Attention: recall that the attention mechanism is intrinsically suitable for **generating a probability distribution in an adaptive manner.**

\rightarrow so we leverage it to design an Attention Module.



北京大學



Attentional Critic for Adaptability

(2) To calculate the expectation of all $Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})$

- We proposed Attention Module to do this task
- It works as follows:

Firstly, a hidden vector $h_i(w_i)$ is generated based on all actions of teammates (i.e., \vec{a}_{-i}).

Then, the attention weight vector $W_i(w_i)$ is generated by comparing $h_i(w_i)$ with all action conditional Q-values $Q_i^k(s, a_i | \vec{a}_{-i}; w_i)$. Specifically, we apply the dot score function [19] to calculate the element $W_i^k(w_i) \in W_i(w_i)$:

$$W_i^k(w_i) = \frac{\exp(h_i(w_i)Q_i^k(s, a_i | \vec{a}_{-i}; w_i))}{\sum_{k=1}^K \exp(h_i(w_i)Q_i^k(s, a_i | \vec{a}_{-i}; w_i))} \quad (8)$$

Lastly, the *contextual Q-value* $Q_i^c(s, a_i, \vec{a}_{-i}; w_i)$ is calculated as a weighted summation of W_i^k and Q_i^k :

$$Q_i^c(s, a_i, \vec{a}_{-i}; w_i) = \sum_{k=1}^K W_i^k(w_i)Q_i^k(s, a_i | \vec{a}_{-i}; w_i) \quad (9)$$

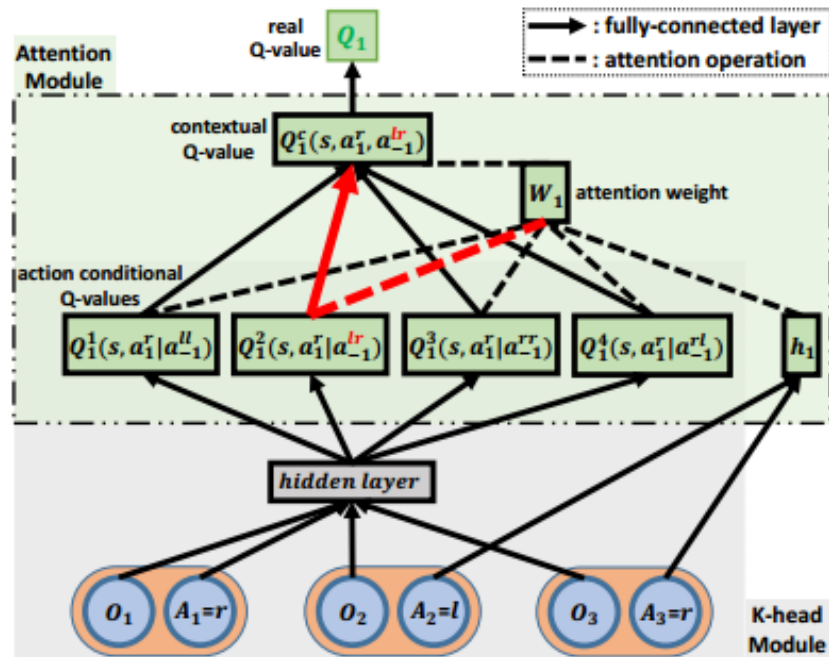


Figure 3: The attention critic of ATT-MADDPG. For clarity, we only show the detailed generation of Q_1 using a three-agent example: the discrete action space is $\{l, r\}$, and the agents prefer to take the actions r , l , and r , respectively. In this case, the second action conditional Q-value Q_1^2 will contribute more weights to the computation of the contextual Q-value Q_1^c , as indicated by thicker red links. We call Q_i the real Q-value, Q_i^c the contextual Q-value, and Q_i^k the action conditional Q-value. The difference is that Q_i^c and Q_i^k are multi-dimensional vectors, while Q_i is the real scalar Q-value used in Equation 10, 11, and 12.




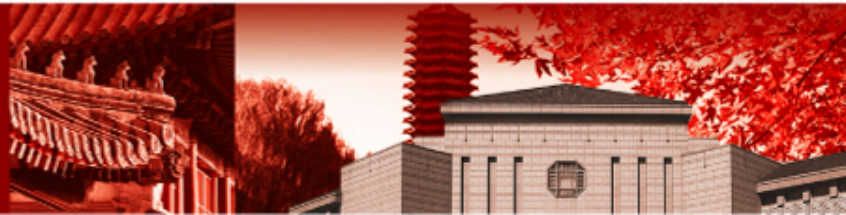
Attentional Critic for Adaptability

- Summary

$$Q_i^{\pi_i|\vec{\pi}_{-i}}(s, a_i) = \mathbb{E}_{\vec{a}_{-i} \sim \vec{\pi}_{-i}}[Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (6)$$

$$= \sum_{\vec{a}_{-i} \in \vec{A}_{-i}} [\vec{\pi}_{-i}(\vec{a}_{-i}|s) Q_i^{\pi_i}(s, a_i, \vec{a}_{-i})] \quad (7)$$

$$Q_i^c(s, a_i, \vec{a}_{-i}; w_i) = \sum_{k=1}^K W_i^k(w_i) Q_i^k(s, a_i | \vec{a}_{-i}; w_i) \quad (9)$$




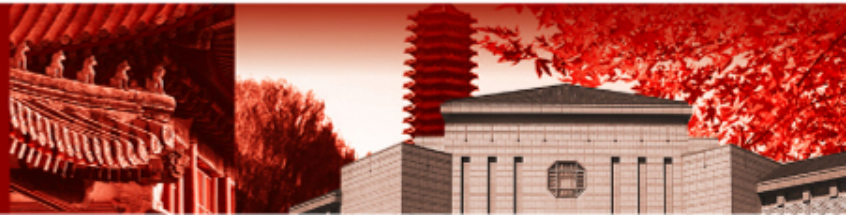
Key Implementation

- **K-head Module:** there is no need to set $K = |\vec{A}_{-i}|$, such that our method is feasible for large discrete action space and even continuous action space.
 - only a small set of actions are crucial in most cases, and the conclusion is suitable for both continuous [26] and discrete [31] action space environments.
 - *We argue that if $Q_i^k(s, a_i | \vec{a}_{-i}; w_i)$ could group similar \vec{a}_{-i} (namely, representing different but similar \vec{a}_{-i} using one Q-value head), it will be much more efficient.*
 - As deep neural network is an universal function approximator [4, 12, 25], we expect that our method can possess this ability.
 - *Further experiments also indicate that our hypothesis is reasonable.*
 - In the experiments, we test $K=2, 4, 8, 12$ and 16 , they all work well.
- **Attention Module:** transforming the multi-dimensional *contextual Q-value* Q_i^c into a scalar *real Q-value* Q_i using a fully-connected layer with one output neuron.
- **Parameter Updating Method:** similar to MADDPG.

See the paper for the details.



北京大學

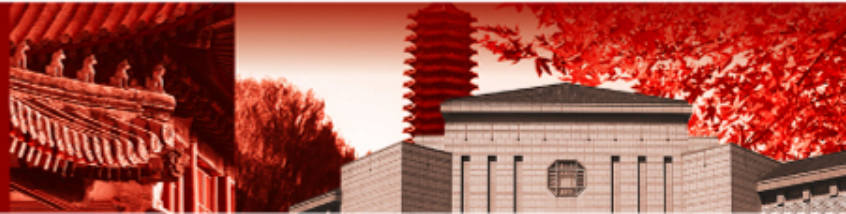


Outline

- Research Problem
- Background
- Design
- **Evaluation**
- Conclusion



北京大學

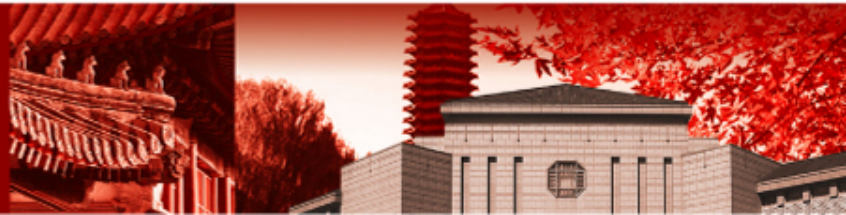


Experimental Settings

- Environments
 - The packet routing environments
 - The benchmark environments
- Baselines
 - [1] MADDPG: the centralized critic is a fully-connected network
 - [2] PSMADDPGV2: same as MADDPG, but sharing parameters among homogeneity agents
 - Khead-MADDPG: the ablation model that directly merges the branches of K-head Module to generate the real Q-value, and there is no attention mechanism in this model.
 - Some rule-based methods: WCMP, Greedy-Pursuit, etc...

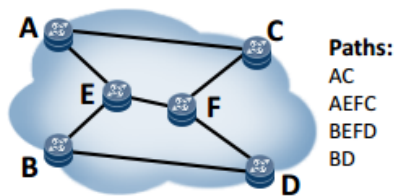
[1] Lowe R, Wu Y, Tamar A, et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. NIPS 2017.

[2] Chu et al. Parameter Sharing Deep Deterministic Policy Gradient for Cooperative Multi-agent Reinforcement Learning. Arxiv 2017.



Results of Packet Routing

- Achieve better performance & better scalability
- Stay robust at a wide range of K to achieve good results



(a) The small topology.

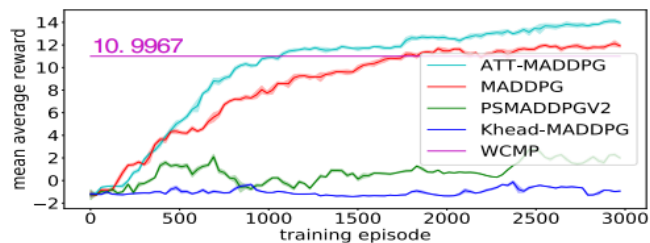


Figure 5: The average rewards on small topology.

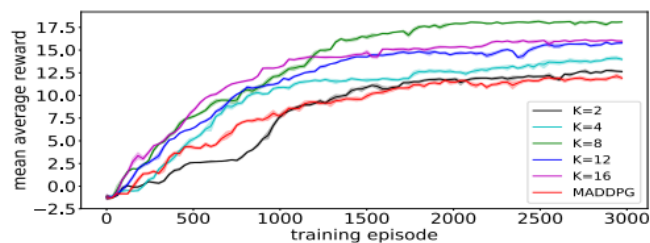
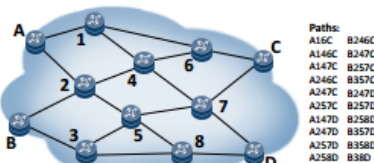


Figure 7: The robustness test on small topology.



(b) The large topology.

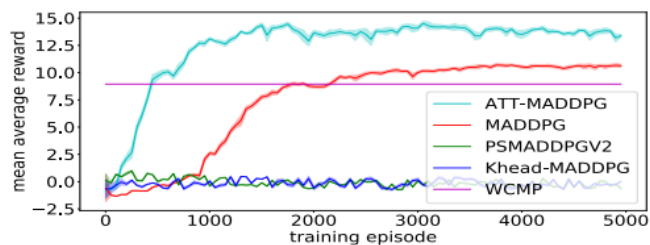


Figure 6: The average rewards on large topology.

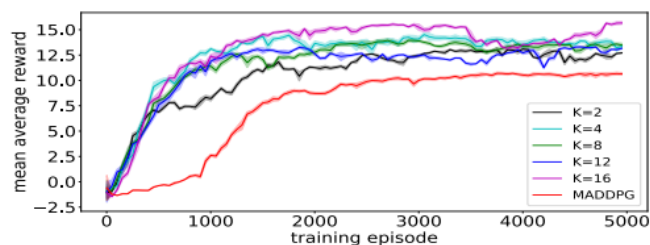
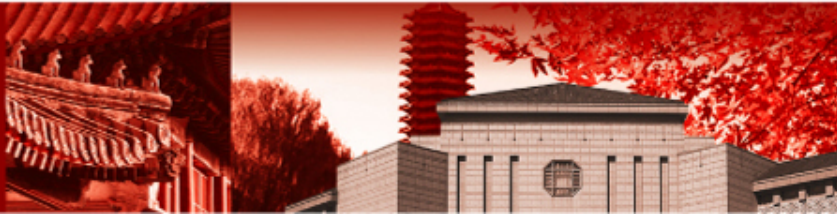


Figure 8: The robustness test on large topology.



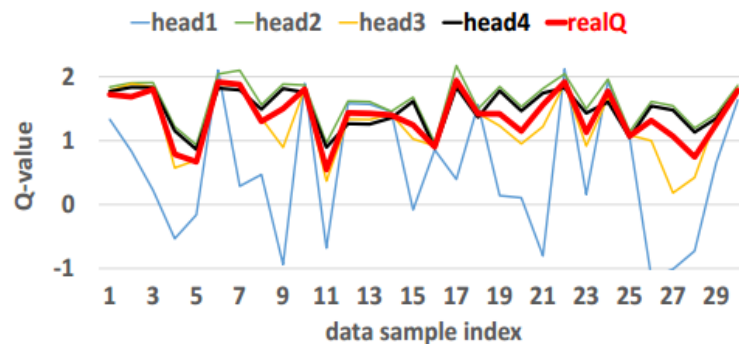
北京大学



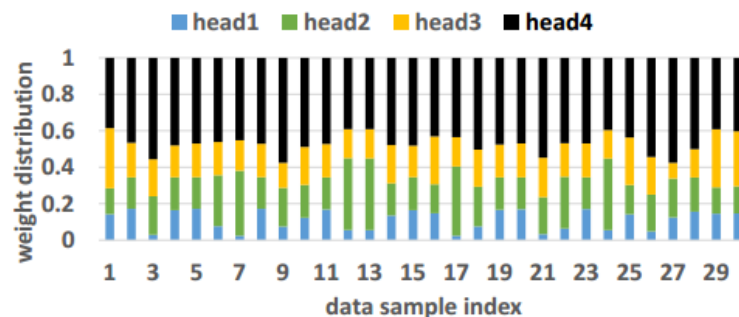
Results of Packet Routing

- If $Q_i^k(s, a_i | \vec{a}_{-i}; w_i)$ could group similar \vec{a}_{-i} , it will be much more efficient.
 - The analysis on the Q-values and the attention weights also indicates that our hypothesis is reasonable.

See the paper for the details.



(a) The different heads' Q-values.

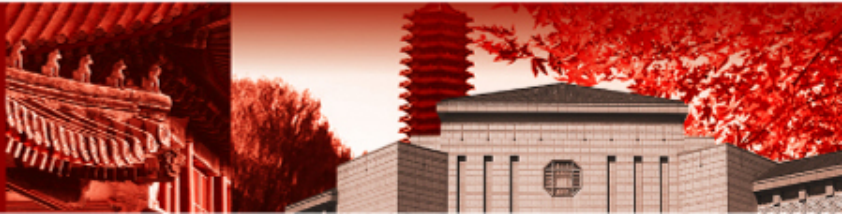


(b) The attention weights.

Figure 9: The Q-values and attention weights generated by router B in the small topology.



北京大學



Results of Benchmark Tasks

- We see that ATT-MADDPG can obtain more rewards than all baselines (both RL-based and rule-based) in both environments.
- It indicates that our method asserts itself with general applicability and good performance.

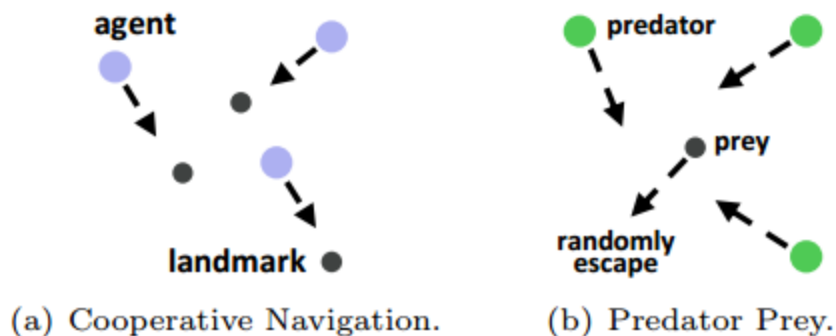


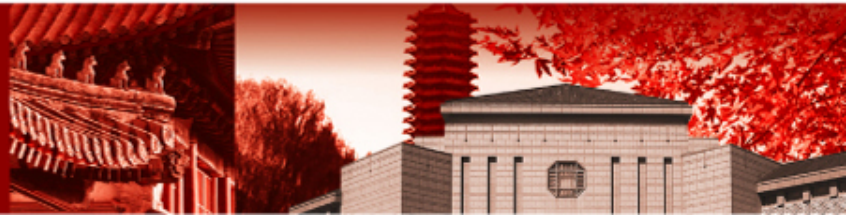
Figure 10: The benchmark environments.

Table 2: The average final stable rewards.

	Co. Na.	Pr. Pr.
ATT-MADDPG, $K=2$	-1.279	3.986
ATT-MADDPG, $K=4$	-1.268	3.589
ATT-MADDPG, $K=8$	-1.322	3.012
ATT-MADDPG, $K=12$	-1.353	3.170
ATT-MADDPG, $K=16$	-1.317	3.004
PSMADDPGV2	-1.586	2.473
MADDPG	-1.767	1.920
GreedyPursuit	-2.105	1.903
Khead-MADDPG	-2.825	1.899



北京大學



Results of Benchmark Tasks

- Figure 11 shows a convergent joint policy learned by ATT-MADDPG under the cooperative navigation task.
 - In the beginning: move to the middle of two landmarks
 - After a while: learn the closest landmark and move directly to that landmark
- These behaviors indicate that the agents really learnt a cooperative joint policy.

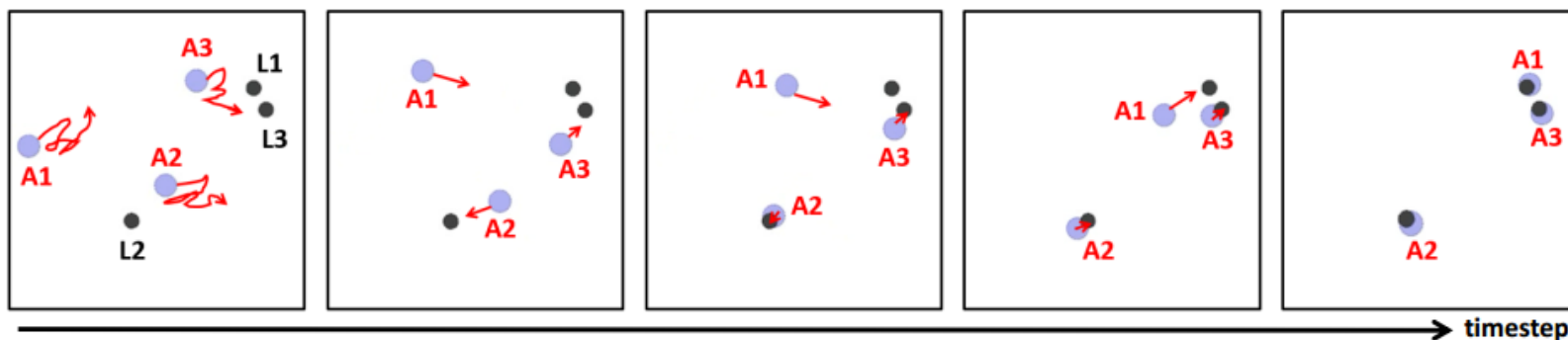
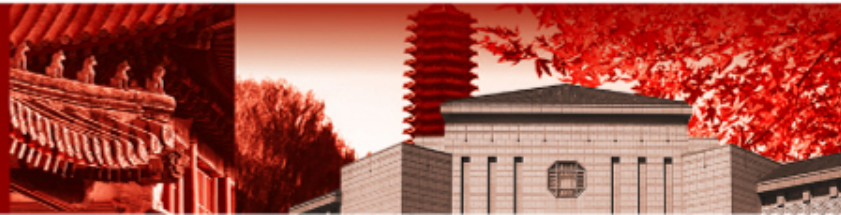


Figure 11: A convergent joint policy learned by ATT-MADDPG under an instance of the cooperative navigation task. L1, L2 and L3 represent different landmarks. A1, A2 and A3 stand for different agents. The red arrows indicate the agents' actions. Note that one picture stands for several timesteps.

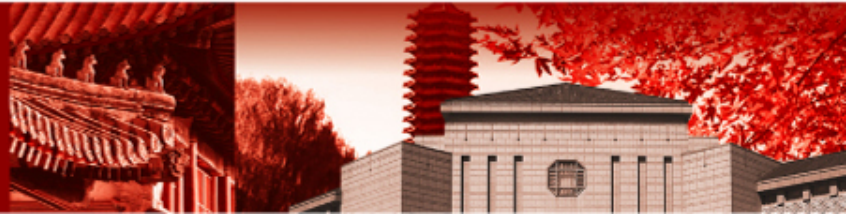


Outline

- Research Problem
- Background
- Design
- Evaluation
- **Conclusion**



北京大學

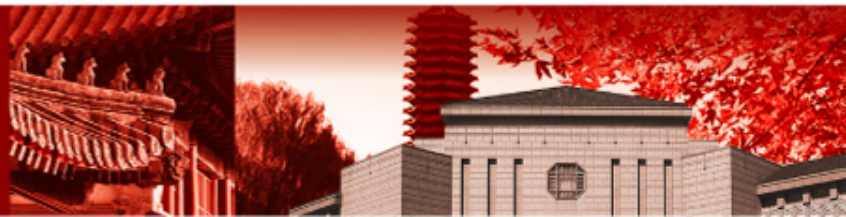


Conclusion @ Method

- This paper presents an actor-critic RL method to **model and exploit teammates' policies** in cooperative distributed multi-agent setting.
- Our method **embeds an attention mechanism into a centralized critic**, which introduces a special structure to explicitly model the dynamic joint policy of teammates in an adaptive manner.
- Consequently, all agents will cooperate with each other efficiently.



北京大學

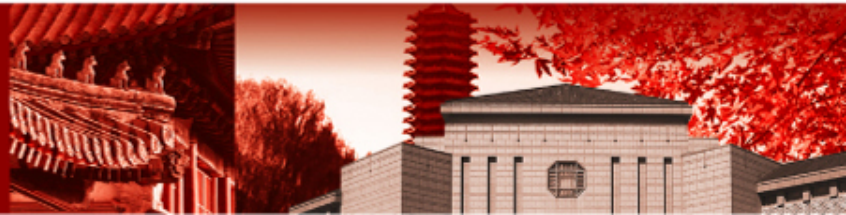


Conclusion @ Experiments

- We evaluate our method on both benchmark tasks and the real-world packet routing tasks.
- The results show that it not only outperforms the several RL-based methods and rule-based methods by a large margin, but also achieves good scalability and robustness.
- Moreover, to better understand our method, we also conduct thorough experiments:
 - (1) the ablation model illustrates that all components of the proposed model are necessary;
 - (2) the study on Q-values and attention weights demonstrates that our method has mastered a sophisticated attention mechanism indeed;
 - (3) the analysis of a concrete policy shows that the agents really learned a cooperative joint policy.



北京大學

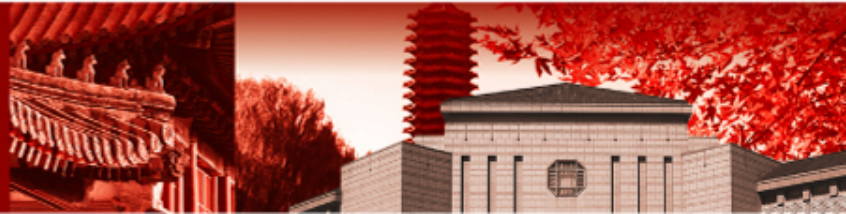


Thanks for listening!

Question?



北京大學

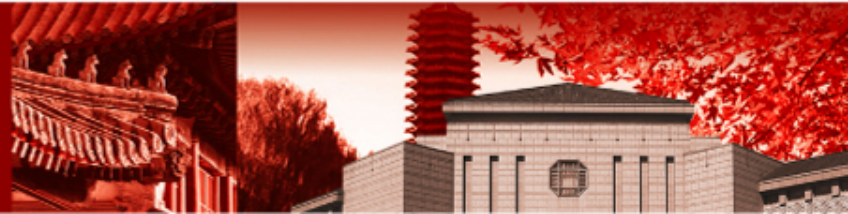


Some Discussion

- **Attention mechanism selectively attends to more important and relevant information from all observations and actions of all agents, and correspondingly ignores unimportant and irrelevant information**
- **That is why ATT-MADDPG outperforms other non-attentional methods, especially when the number of agents becomes large.**

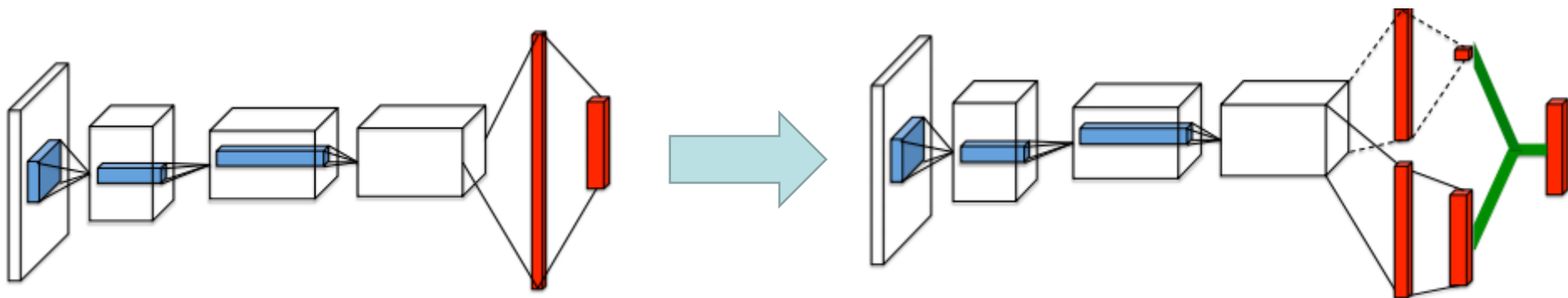


北京大學



Some Discussion

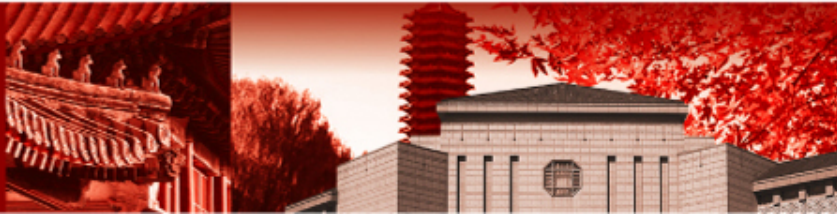
- **The Inductive Bias is crucial for the success of Deep Learning**
- **One promising direction is that trying to inject inductive bias into your network structure**
 - Generally, CNN, LSTM, Attention, etc...
 - Specially for RL, Dueling DQN, Value Iteration Network, etc...



2016-ICML(best paper)-Dueling Network Architectures for Deep Reinforcement Learning
2016-NIPS(best paper)-Value Iteration Networks



北京大學



ACKNOWLEDGMENTS

The authors would like to thank Zhihua Zhang, Xiangyu Liu, Yan Ni, Yuanxing Zhang, Shihan Xiao, and Shiru Ren for helpful suggestions. The authors would also like to thank the anonymous reviewers for their comments. This work was supported by the National Natural Science Foundation of China under Grant No.61572044 and 61872397. The contact author is Zhen Xiao.

