Neighborhood Cognition Consistent
Multi-Agent Reinforcement Learning

Hangyu Mao,[1,2] Wulong Liu,[2] Jianye Hao,[3,2] Jun Luo[2]
Dong Li,[2] Zhengchao Zhang,[1] Jun Wang,[4] Zhen Xiao[1]
[1]Peking University, [2]Noah's Ark Lab, Huawei
[3]Tianjin University, [4]University College London
{hy.mao, zhengchaozhang, xiaozhen}@pku.edu.cn
{liuwulong, haojianye, jun.luo1, lidong106}@huawei.com
jun.wang@cs.ucl.ac.uk

诺亚方舟实验室
Noah's Ark Lab

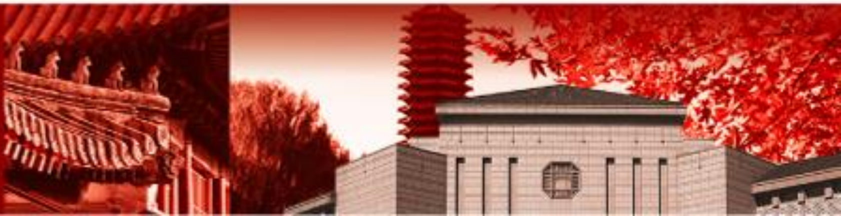
英国伦敦大学学院
University College London

Learning Agent Communication under Limited Bandwidth by Message Pruning

Hangyu Mao,[1] Zhengchao Zhang,[1] Zhen Xiao,[1] Zhibo Gong,[2] Yan Ni[1]
[1]Peking University, [2]Huawei Technologies Co., Ltd.
{hy.mao, zhengchaozhang, xiaozhen, niyan.ny}@pku.edu.cn, gongzhibo@huawei.com

Outline

- **Motivation**
- Design
- Evaluation
- Conclusion



Many Stories of DRL

Win the best human Go → Go Zero → Zero

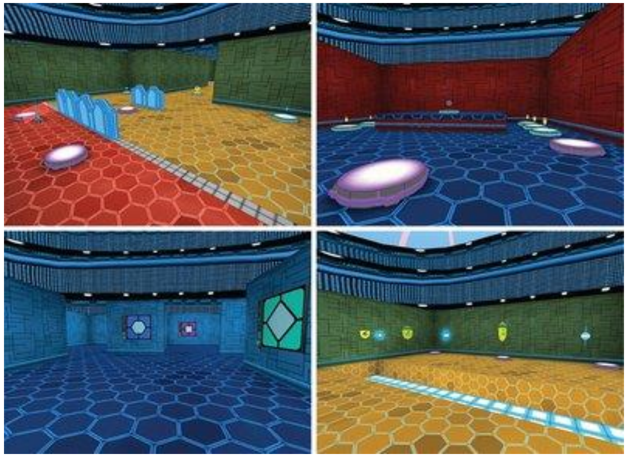
Reduce data center cooling bill by 40%

Alibaba Group 阿里巴巴集团 | 阿里技术

从虚拟世界走进现实应用

强化学习在阿里的技术演进与业务创新

Reinforcement Learning Beyond Games: To Make a Difference in Alibaba



Playing Atari games

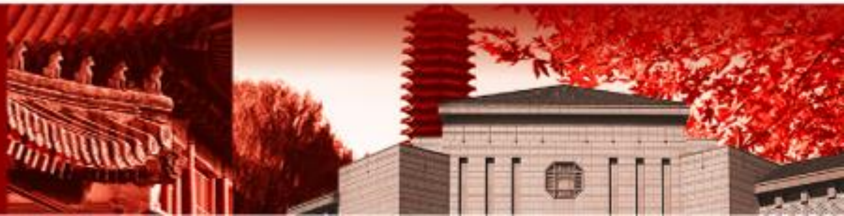
Berkeley helicopter

Relatively Backward of MARL

five decentralized cooperative agents

only one centralized agent

what is the next one

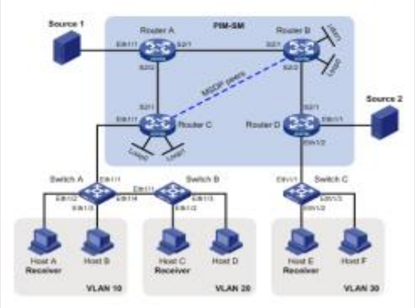


Focus of This Research

more agents in real systems

Abilene/Internet2 Network

Unmanned Aerial Vehicle



Smart Grid/WiFi Network

Autopilot/Unmanned Warship

possible answer:

more agents

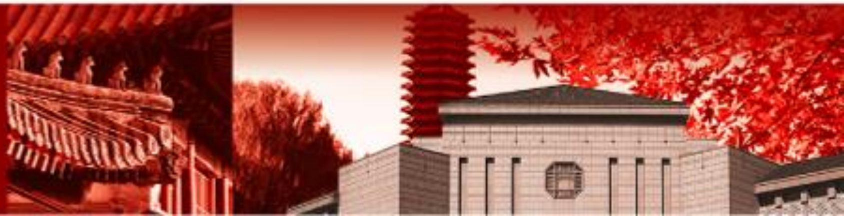
Focus of This Research

But how to coordinate more agents?

Communication

But real applications has limited bandwidth!

Message pruning

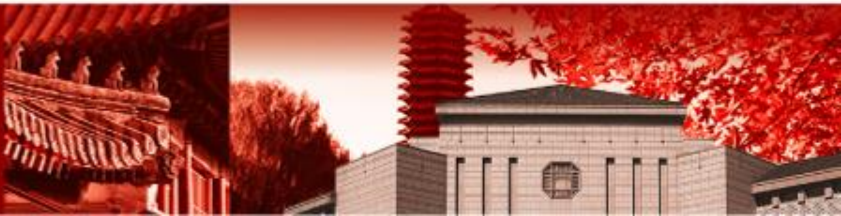


Focus of This Research

We focus on addressing the limited bandwidth problem in multi-agent communication by message pruning.

Outline

- Motivation
- **Design**
- Evaluation
- Conclusion



ACML (w/o Comm.)

- ACML combines the merits of the existing methods.

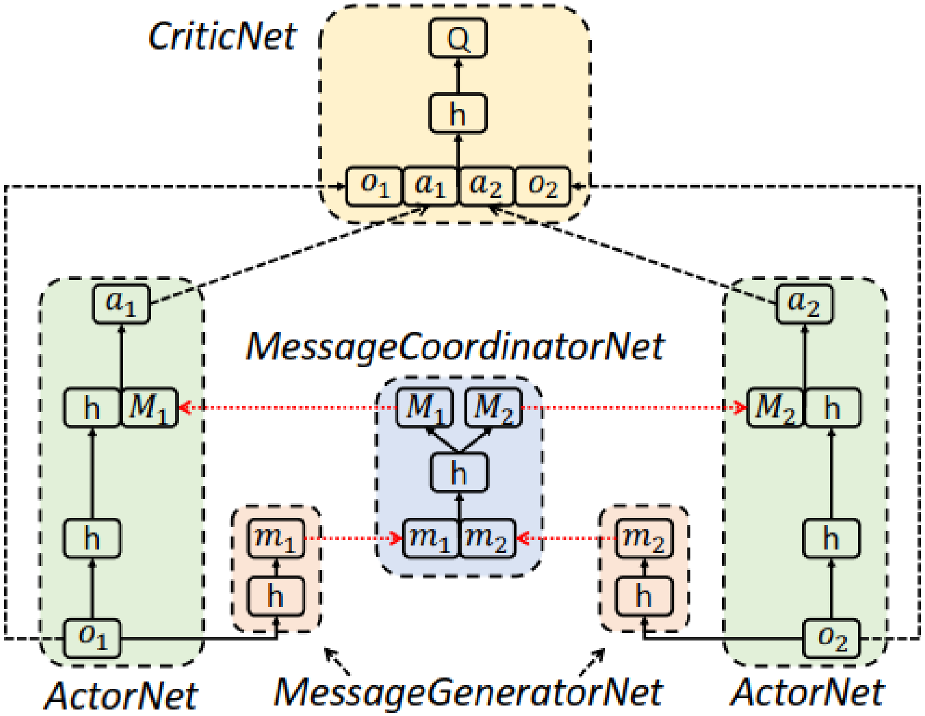


Figure 1. The proposed ACML. For clarity, we illustrate this model with a two-agent example. All components are implemented by DNN. The red arrows indicate the message exchange process.

However,

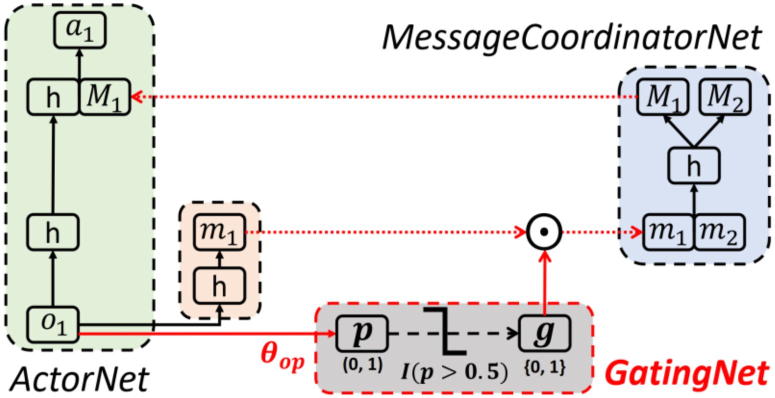
the agents have to send messages continuously (in order to generate one action),

regardless of whether the messages are beneficial to the performance of the agent team.

Gated-ACML

- Gated-ACML applies a gating mechanism to adaptively identify less beneficial messages (for the agent team) and thus to adaptively prune these messages.

Figure 2: The actor part of Gated-ACML. For clarity, we only show one agent's structure; we do not show the critic part because it is the same as that of ACML.



- To make the above design work, **a suitable p must be trained for each observation**, otherwise Gated-ACML may degenerate to ACML in the extreme case where $I(p > 0.5) \equiv 1$.
- However, as the indicator function $g \leftarrow I(p > 0.5)$ is **non-differentiable**, it makes the end-to-end backpropagation method inapplicable.
- To bypass the training of the non-differentiable indicator function, **we train the input p directly by the auxiliary task technique** (ICLR 2016), which provides training signal for p explicitly.

Gated-ACML

- Because we want to prune the messages on the premise of maintaining the performance (
), we design the following auxiliary task.

① Let p indicate the probability that $\Delta Q(o) = Q(o, a^C) - Q(o, a^I)$ is larger than T.

② In this setting, the true label of this auxiliary task can be formulated as:

$$Y(o_i) = \mathrm{I}\big(Q\big(< o_i, a_i^C >, < \vec{o}_{-i}, \vec{a}_{-i}^C >\big) - Q\big(< o_i, a_i^I >, < \vec{o}_{-i}, \vec{a}_{-i}^C >\big) > T\big)$$

③ Then we can train p by minimizing the following loss function:

$$L_{\theta_{op}}(o_i) = -E_{o_i}[Y(o_i)\log p\big(o_i|\theta_{op}\big) + (1 - Y(o_i))\log(1 - p\big(o_i|\theta_{op}\big))]$$

The insight: If $\Delta Q(o)$ is really larger than T (i.e., a^C can obtain at least T Q-values that a^I, and $Y(o) = 1$), the network should try to generate a probability p that is larger than $T_p = 0.5$ to encourage communication.

Key Implementation

- The training method relies on correct labels of the auxiliary task.

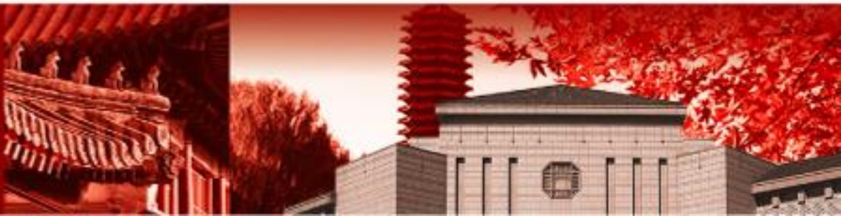
  $$Y(o_i) = \mathrm{I}\left(Q\left(< o_i, a_i^C >, < \vec{o}_{-i}, \vec{a}_{-i}^C >\right) - Q\left(< o_i, a_i^I >, < \vec{o}_{-i}, \vec{a}_{-i}^C >\right) > T\right)$$

- $Q(o, a^C)$ and $Q(o, a^I)$ can be estimated by setting g=1 and g=0, respectively.

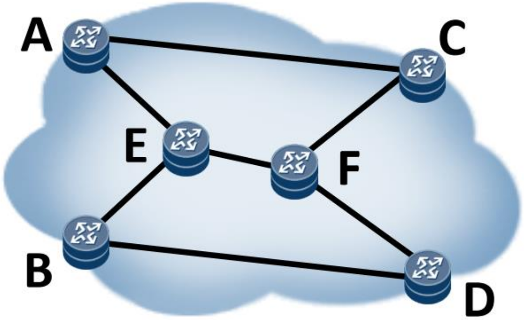
- For T, we propose two methods to set a fixed T and a dynamic T.
  - The moving average to set a **dynamic** $\boldsymbol{T}$:
    - $T_t = (1-\beta)T_{t-1} + \beta\left(Q_t\left(< o_i, a_i^C >, < \vec{o}_{-i}, \vec{a}_{-i}^C >\right) - Q_t\left(< o_i, a_i^I >, < \vec{o}_{-i}, \vec{a}_{-i}^C >\right)\right)$
    - Advantage: $Y(o)$ becomes an adaptive training label even for the same observation o. This is very important for the dynamically changing environments.
  - Pre-calculating to set a fixed T:
    - First, sort the $\Delta Q(o)$ of the latest K observations encountered during training, resulting in $L_{\Delta Q(o)}$.
    - Then, set T by splitting $L_{\Delta Q(o)}$ **in terms of the index**. For example, if we want to prune $T_m\%$ messages, we set $T = L_{\Delta Q(o)}[K \times T_m\%]$.
    - Advantage: the actual number of pruned messages is ensured to be close to the desired $T_m\%$.

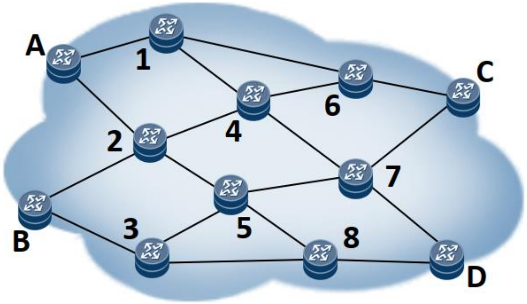
Outline

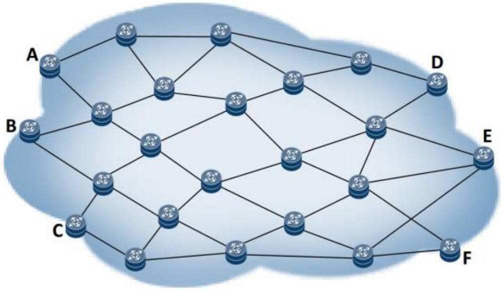
- Motivation
- Design
- **Evaluation**
- Conclusion

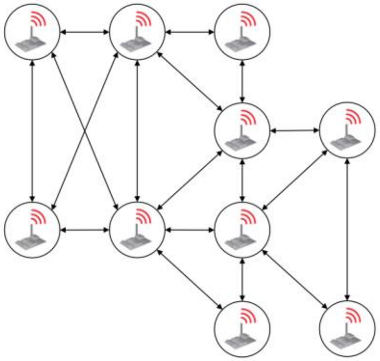


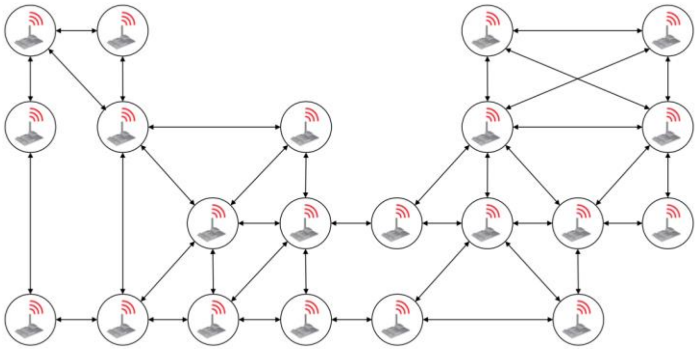
Environments

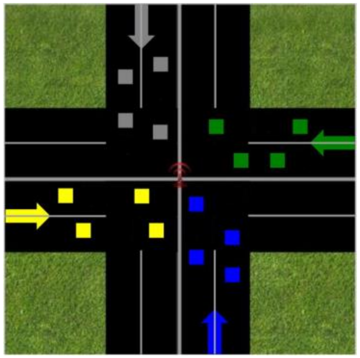












Results

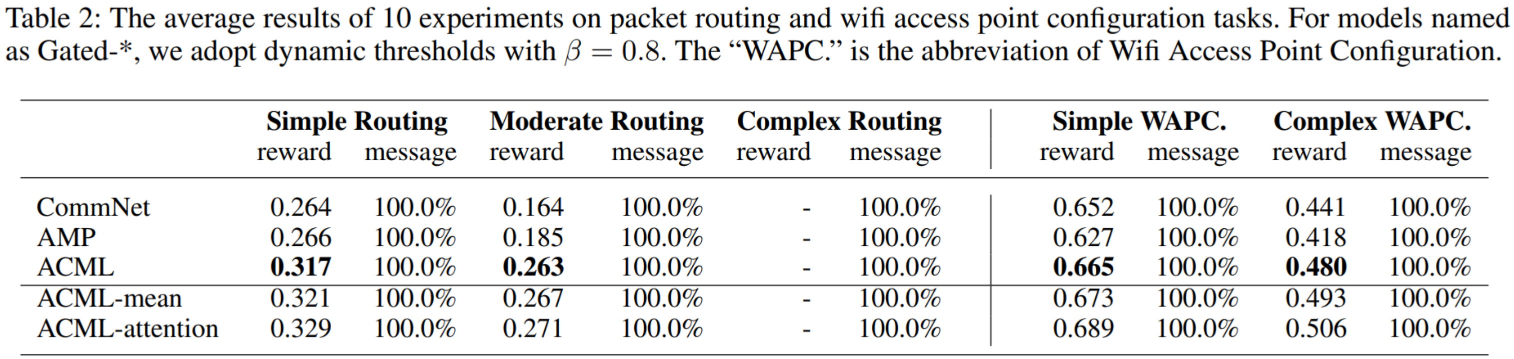
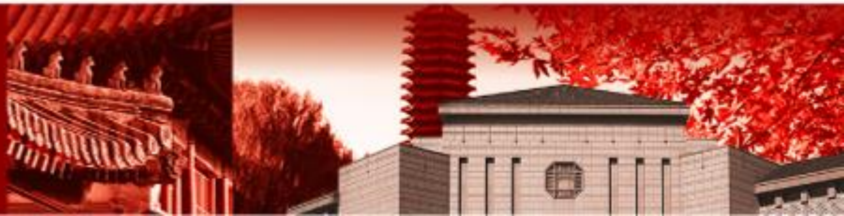


Table 2: The average results of 10 experiments on packet routing and wifi access point configuration tasks. For models named as Gated-*, we adopt dynamic thresholds with $\beta = 0.8$. The "WAPC." is the abbreviation of Wifi Access Point Configuration.

	Simple Routing		**Moderate Routing**		**Complex Routing**		**Simple WAPC.**		**Complex WAPC.**	
	reward	message	reward	message	reward	message	reward	message	reward	message
CommNet	0.264	100.0%	0.164	100.0%	-	100.0%	0.652	100.0%	0.441	100.0%
AMP	0.266	100.0%	0.185	100.0%	-	100.0%	0.627	100.0%	0.418	100.0%
ACML	**0.317**	100.0%	**0.263**	100.0%	-	100.0%	**0.665**	100.0%	**0.480**	100.0%
ACML-mean	0.321	100.0%	0.267	100.0%	-	100.0%	0.673	100.0%	0.493	100.0%
ACML-attention	0.329	100.0%	0.271	100.0%	-	100.0%	0.689	100.0%	0.506	100.0%

ACML (i.e., w/o message pruning) works better than baselines.



Results

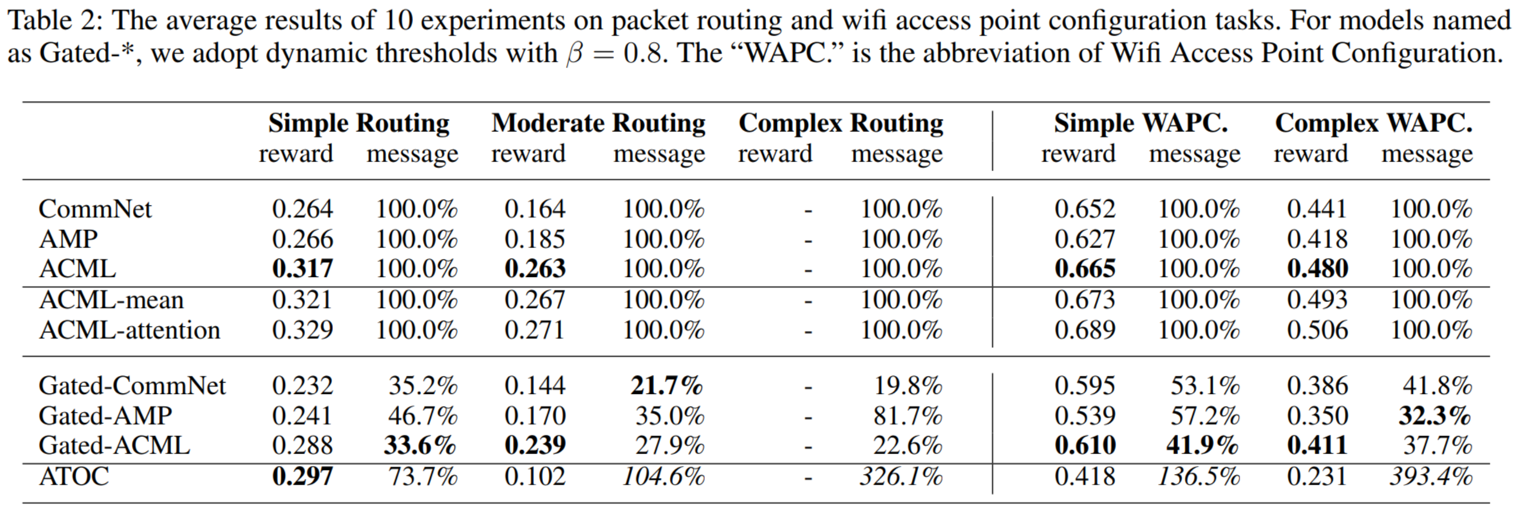
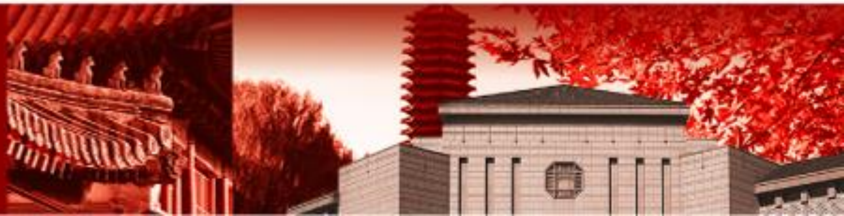


Table 2: The average results of 10 experiments on packet routing and wifi access point configuration tasks. For models named as Gated-*, we adopt dynamic thresholds with $\beta = 0.8$. The "WAPC." is the abbreviation of Wifi Access Point Configuration.

	Simple Routing		**Moderate Routing**		**Complex Routing**		**Simple WAPC.**		**Complex WAPC.**	
	reward	message	reward	message	reward	message	reward	message	reward	message
CommNet	0.264	100.0%	0.164	100.0%	-	100.0%	0.652	100.0%	0.441	100.0%
AMP	0.266	100.0%	0.185	100.0%	-	100.0%	0.627	100.0%	0.418	100.0%
ACML	**0.317**	100.0%	**0.263**	100.0%	-	100.0%	**0.665**	100.0%	**0.480**	100.0%
ACML-mean	0.321	100.0%	0.267	100.0%	-	100.0%	0.673	100.0%	0.493	100.0%
ACML-attention	0.329	100.0%	0.271	100.0%	-	100.0%	0.689	100.0%	0.506	100.0%
Gated-CommNet	0.232	35.2%	0.144	**21.7%**	-	19.8%	0.595	53.1%	0.386	41.8%
Gated-AMP	0.241	46.7%	0.170	35.0%	-	81.7%	0.539	57.2%	0.350	**32.3%**
Gated-ACML	0.288	**33.6%**	**0.239**	27.9%	-	22.6%	**0.610**	**41.9%**	**0.411**	37.7%
ATOC	**0.297**	73.7%	0.102	*104.6%*	-	*326.1%*	0.418	*136.5%*	0.231	*393.4%*

Gated-ACML (w/ dynamic T) can prune a lot of messages with little impact on performance.



Results

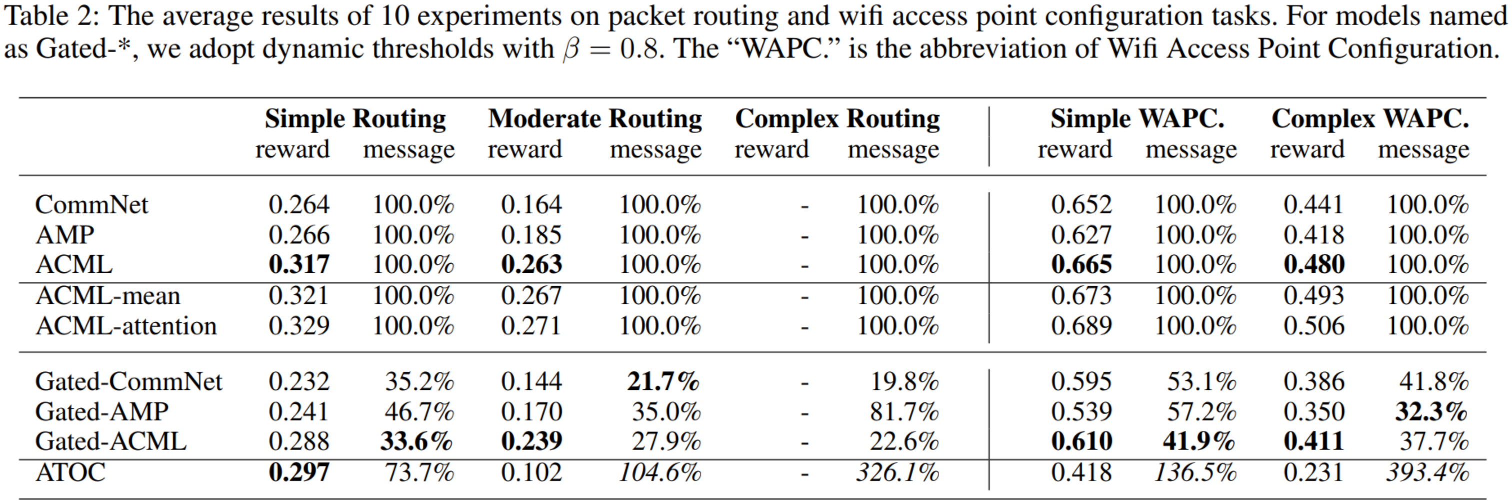


Table 2: The average results of 10 experiments on packet routing and wifi access point configuration tasks. For models named as Gated-*, we adopt dynamic thresholds with $\beta = 0.8$. The “WAPC.” is the abbreviation of Wifi Access Point Configuration.

	Simple Routing		**Moderate Routing**		**Complex Routing**		**Simple WAPC.**		**Complex WAPC.**	
	reward	message	reward	message	reward	message	reward	message	reward	message
CommNet	0.264	100.0%	0.164	100.0%	-	100.0%	0.652	100.0%	0.441	100.0%
AMP	0.266	100.0%	0.185	100.0%	-	100.0%	0.627	100.0%	0.418	100.0%
ACML	**0.317**	100.0%	**0.263**	100.0%	-	100.0%	**0.665**	100.0%	**0.480**	100.0%
ACML-mean	0.321	100.0%	0.267	100.0%	-	100.0%	0.673	100.0%	0.493	100.0%
ACML-attention	0.329	100.0%	0.271	100.0%	-	100.0%	0.689	100.0%	0.506	100.0%
Gated-CommNet	0.232	35.2%	0.144	**21.7%**	-	19.8%	0.595	53.1%	0.386	41.8%
Gated-AMP	0.241	46.7%	0.170	35.0%	-	81.7%	0.539	57.2%	0.350	**32.3%**
Gated-ACML	0.288	**33.6%**	**0.239**	27.9%	-	22.6%	**0.610**	**41.9%**	**0.411**	37.7%
ATOC	**0.297**	73.7%	0.102	*104.6%*	-	*326.1%*	0.418	*136.5%*	0.231	*393.4%*

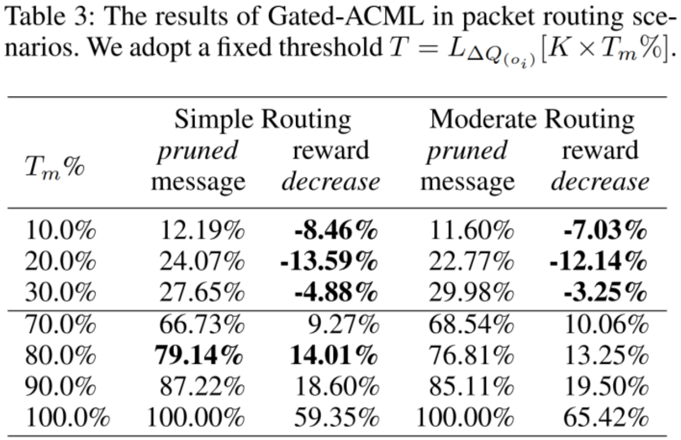
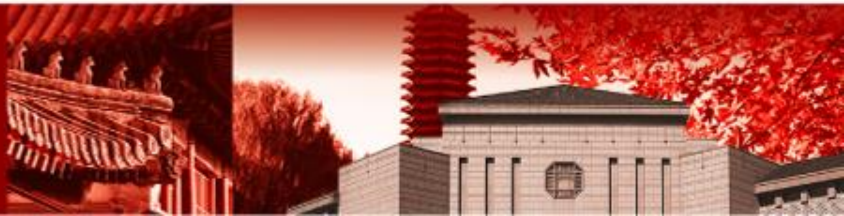


Table 3: The results of Gated-ACML in packet routing scenarios. We adopt a fixed threshold $T = L_{\Delta Q_{(o_i)}}[K \times T_m\%]$.

	Simple Routing		Moderate Routing	
$T_m\%$	*pruned* message	reward *decrease*	*pruned* message	reward *decrease*
10.0%	12.19%	**-8.46%**	11.60%	**-7.03%**
20.0%	24.07%	**-13.59%**	22.77%	**-12.14%**
30.0%	27.65%	**-4.88%**	29.98%	**-3.25%**
70.0%	66.73%	9.27%	68.54%	10.06%
80.0%	**79.14%**	**14.01%**	76.81%	13.25%
90.0%	87.22%	18.60%	85.11%	19.50%
100.0%	100.00%	59.35%	100.00%	65.42%

Gated-ACML (w/ fixed T) can ensure the number of prune messages is close to the desired $T_m\%$.

北京大學



Results

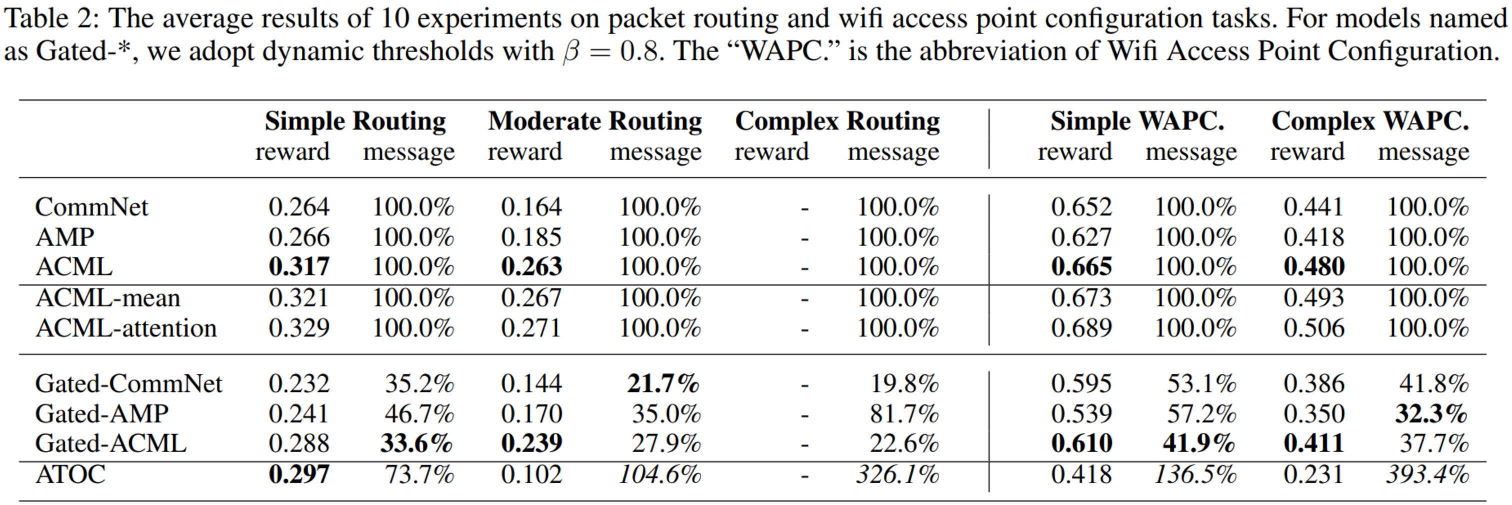


Table 2: The average results of 10 experiments on packet routing and wifi access point configuration tasks. For models named as Gated-*, we adopt dynamic thresholds with $\beta = 0.8$. The "WAPC." is the abbreviation of Wifi Access Point Configuration.

	Simple Routing		**Moderate Routing**		**Complex Routing**		**Simple WAPC.**		**Complex WAPC.**	
	reward	message	reward	message	reward	message	reward	message	reward	message
CommNet	0.264	100.0%	0.164	100.0%	-	100.0%	0.652	100.0%	0.441	100.0%
AMP	0.266	100.0%	0.185	100.0%	-	100.0%	0.627	100.0%	0.418	100.0%
ACML	**0.317**	100.0%	**0.263**	100.0%	-	100.0%	**0.665**	100.0%	**0.480**	100.0%
ACML-mean	0.321	100.0%	0.267	100.0%	-	100.0%	0.673	100.0%	0.493	100.0%
ACML-attention	0.329	100.0%	0.271	100.0%	-	100.0%	0.689	100.0%	0.506	100.0%
Gated-CommNet	0.232	35.2%	0.144	**21.7%**	-	19.8%	0.595	53.1%	0.386	41.8%
Gated-AMP	0.241	46.7%	0.170	35.0%	-	81.7%	0.539	57.2%	0.350	**32.3%**
Gated-ACML	0.288	**33.6%**	**0.239**	27.9%	-	22.6%	**0.610**	**41.9%**	**0.411**	37.7%
ATOC	**0.297**	73.7%	0.102	*104.6%*	-	*326.1%*	0.418	*136.5%*	0.231	*393.4%*

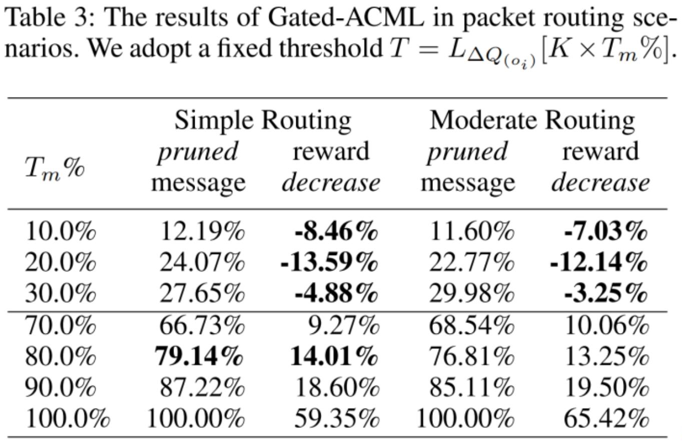
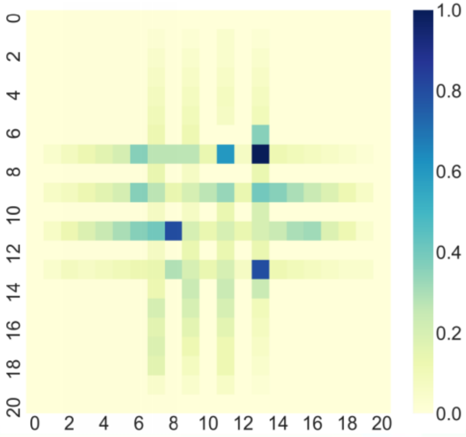


Table 3: The results of Gated-ACML in packet routing scenarios. We adopt a fixed threshold $T = L_{\Delta Q_{(o_i)}}[K \times T_m\%]$.

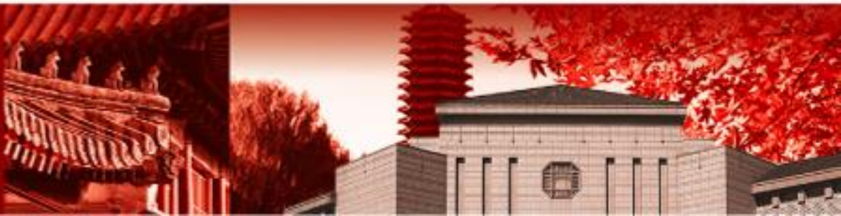
	Simple Routing		Moderate Routing	
$T_m\%$	*pruned* message	reward *decrease*	*pruned* message	reward *decrease*
10.0%	12.19%	**-8.46%**	11.60%	**-7.03%**
20.0%	24.07%	**-13.59%**	22.77%	**-12.14%**
30.0%	27.65%	**-4.88%**	29.98%	**-3.25%**
70.0%	66.73%	9.27%	68.54%	10.06%
80.0%	**79.14%**	**14.01%**	76.81%	13.25%
90.0%	87.22%	18.60%	85.11%	19.50%
100.0%	100.00%	59.35%	100.00%	65.42%



The messages are distributed near the junction where communication is critical for safety deriving.

Outline

- Motivation
- Design
- Evaluation
- **Conclusion**



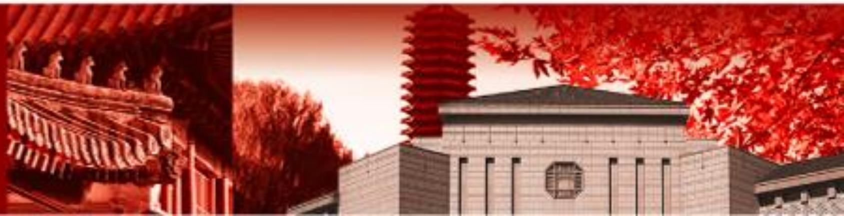
Conclusion

- We have proposed a gating mechanism, which consists of several key designs like auxiliary task with appropriate training signal, dynamic and fixed thresholds, to address the limited bandwidth that has been largely ignored by previous DRL methods.

- The gating mechanism prunes less beneficial messages in an adaptive manner, so that the performance can be maintained or even improved with much fewer messages. (as shown by the experiments on three tasks developed based on eight real-world scenarios.)

- Furthermore, it is applicable to several previous methods and multi-agent scenarios with good performance.

- To the best of our knowledge, it is the first method to achieve these in the multi-agent reinforcement learning community.



Thanks for Listening!

Question?