AdaHAT: Adaptive Hard Attention to the Task in Task-Incremental Learning

Pengxiang Wang, Hongbo Bo, *Jun Hong*, Weiru Liu, and Kedian Mu

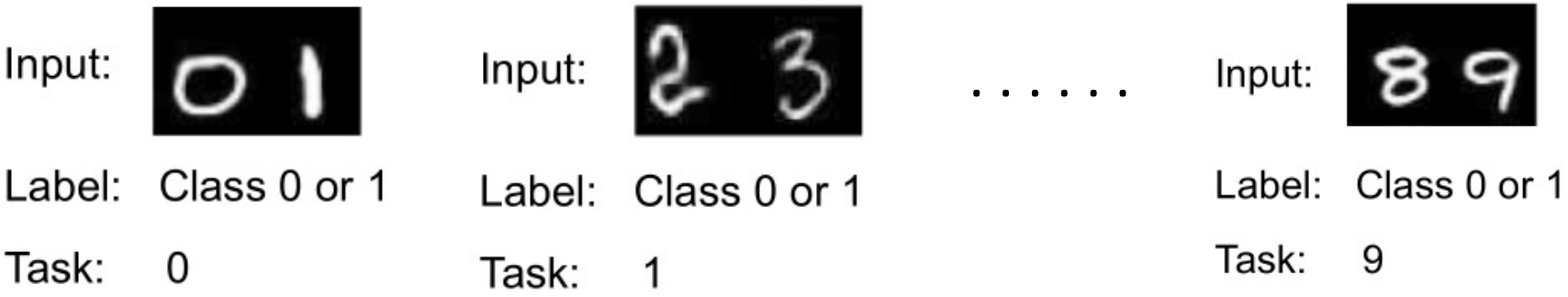
Peking University, China

University of Bristol, UK

University of the West of England, UK

Task-Incremental Learning

Incremental learning/continual learning/lifelong learning: A model learns a sequence of *distinct tasks* in an incremental manner

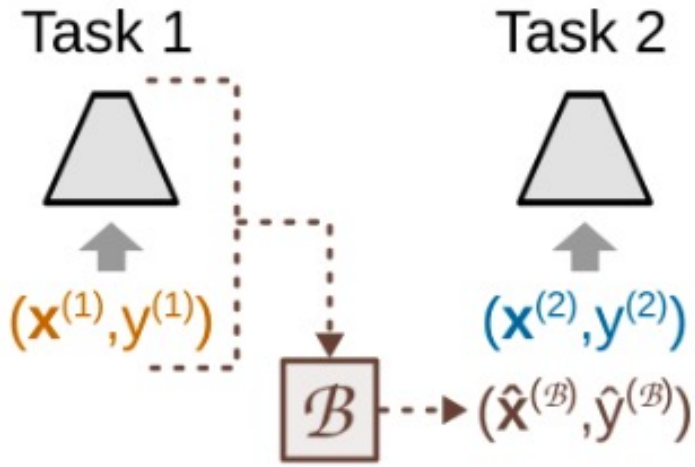


Key challenge - Catastrophic forgetting

- A neural network loses what it has learned in previous tasks after training on new tasks
- Specifically when learning a new task, the parameters are updated, which would not fit to the data distributions of previous tasks

Approaches to Addressing Catastrophic Forgetting

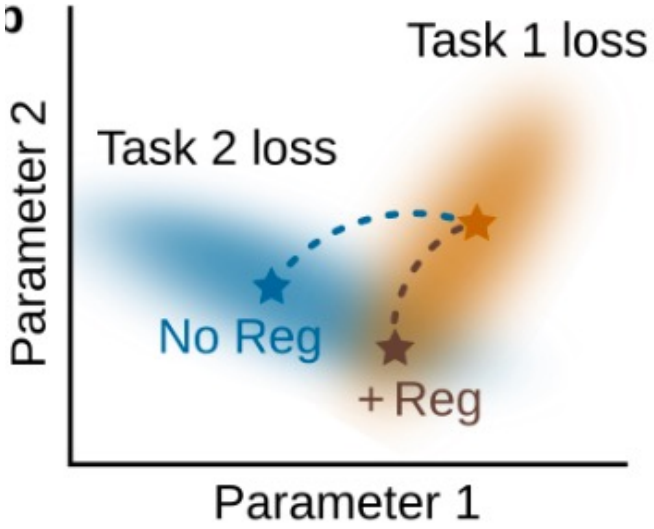
Replay: Complementing the training data of a new task with the data representative of the past tasks



Scalability issues - computational efficiency and storage capacity

Approaches to Addressing Catastrophic Forgetting

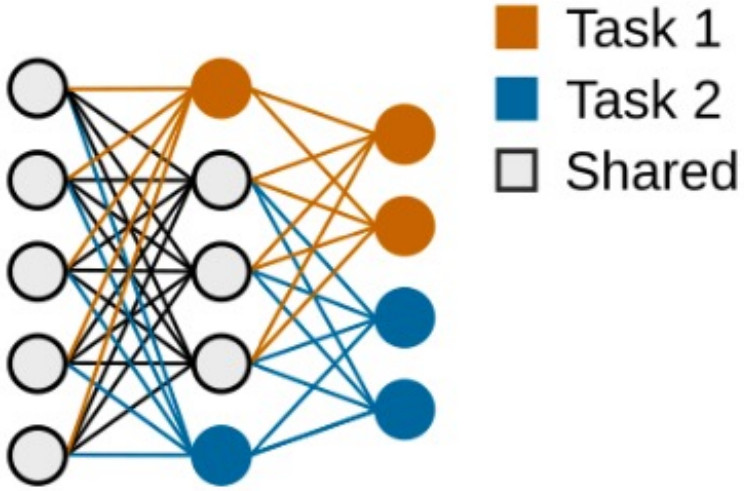
Parameter regularisation: Adding regularisation/penalty terms to the loss to prevent major changes in those parameters important for previous tasks when learning new tasks



Issues: network capacity – reduced parameter space available to new tasks; manually devise regularisation terms; leans towards plasticity of the network

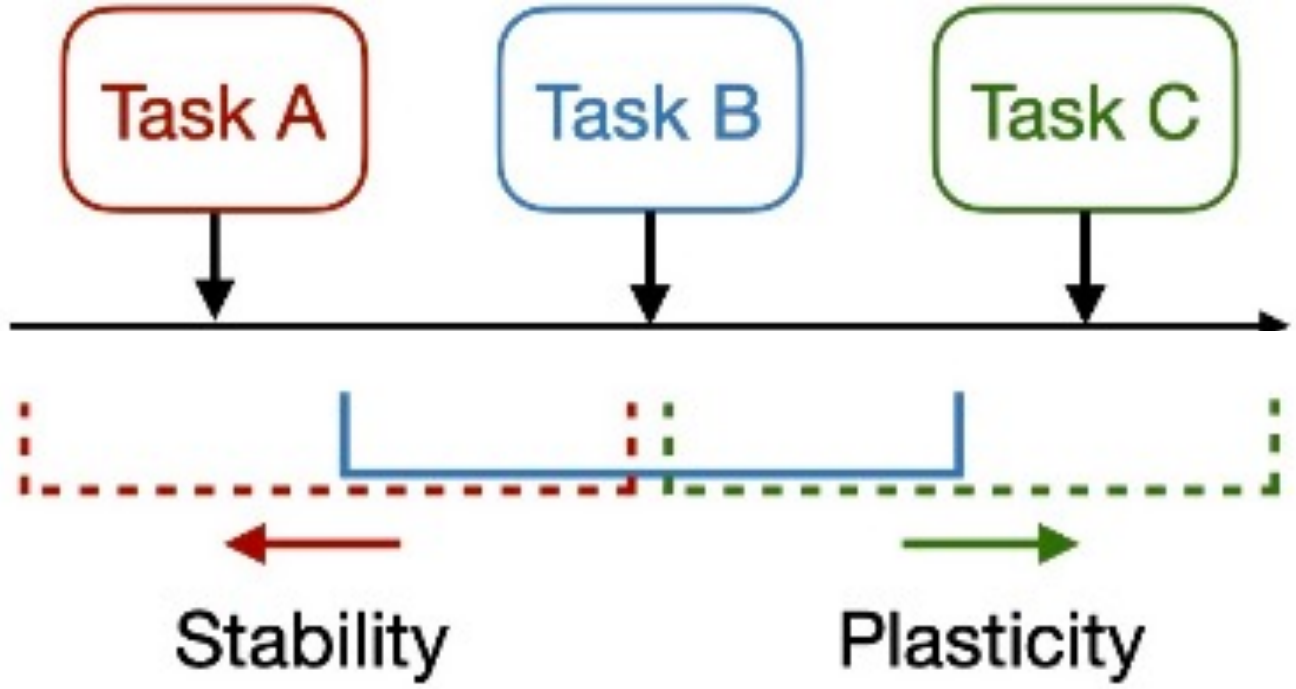
Approaches to Addressing Catastrophic Forgetting

Architecture: ***learning*** to dedicate specific parts of the network for each task



Network capacity issue – reduced parameter space available to new tasks; manually tune hyperparameters; sacrifice the plasticity over the stability

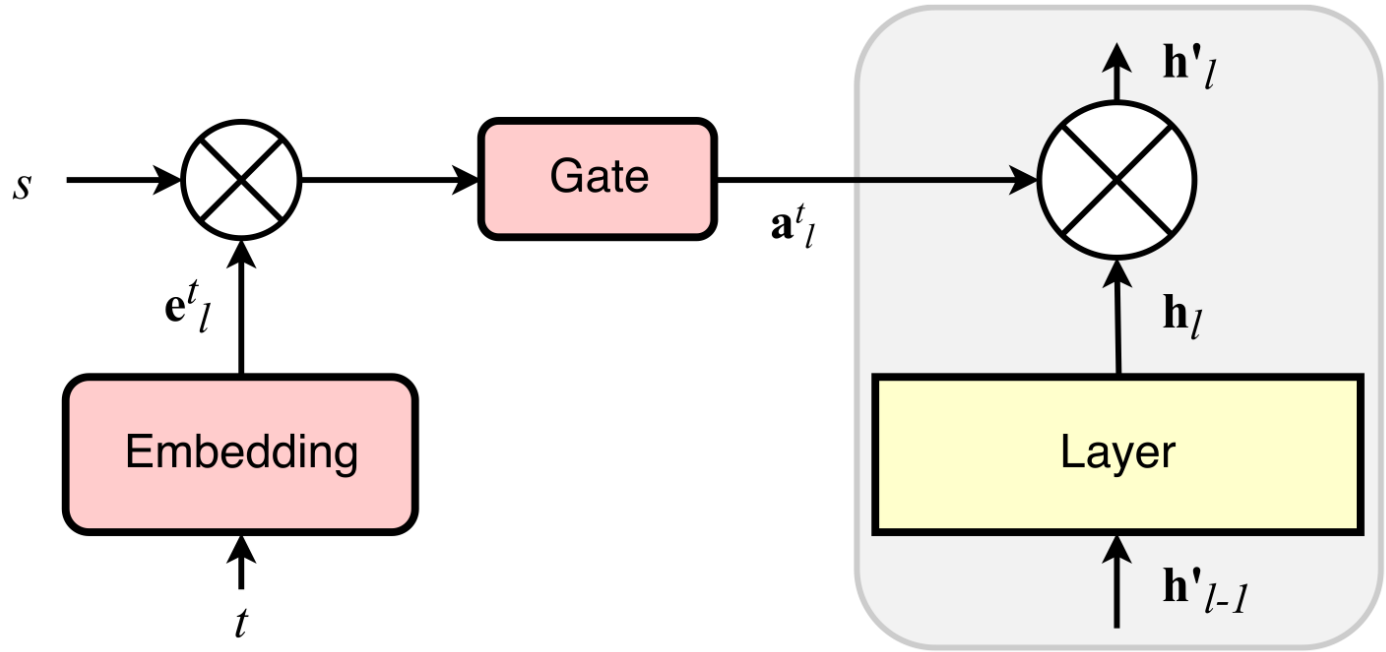
Critical to Balance between Memory Stability and Learning Plasticity



Due to network capacity – rather than simply preventing catastrophic forgetting, need to balance the trade-off between memory stability and learning plasticity

Hard Attention to the Task (HAT)

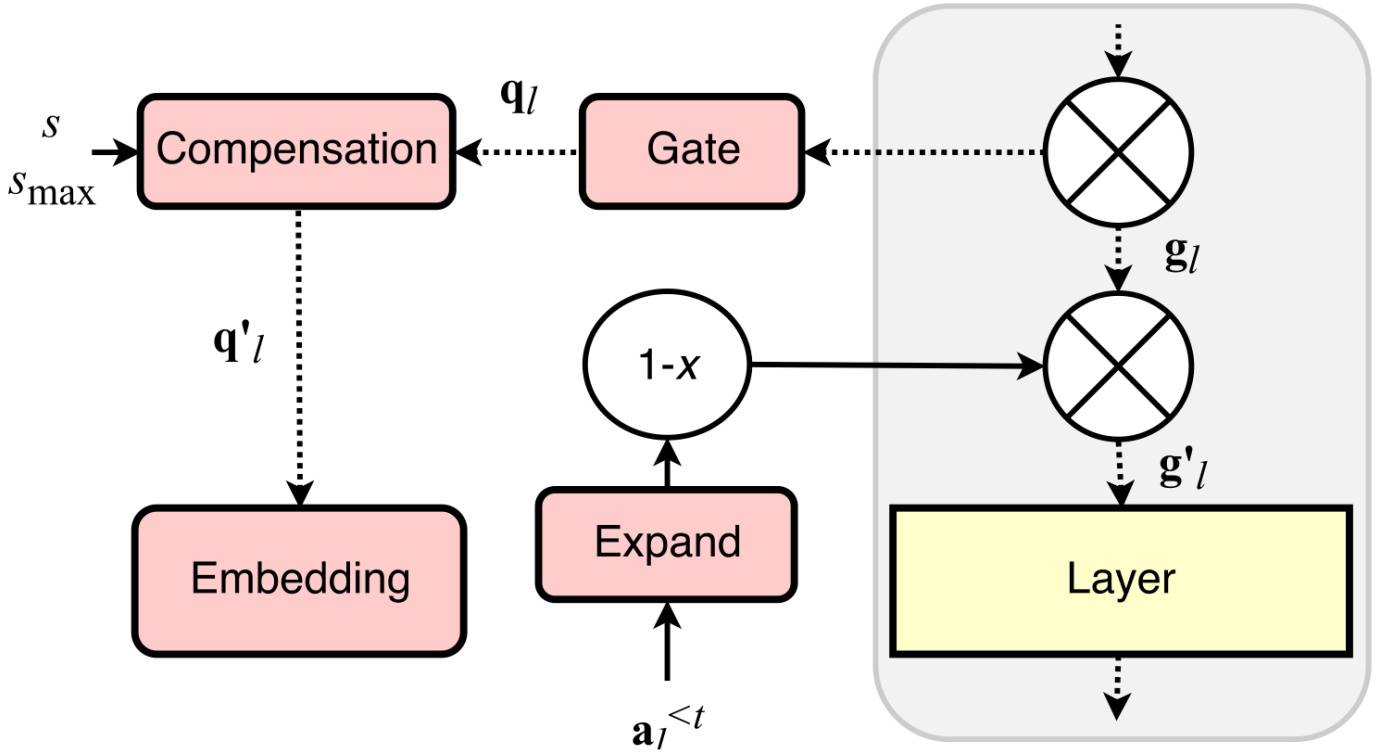
An ICML 2018 paper, state of the art, highly influential (1,000+ citations), proposed a hard *attention mechanism*



Learns a layer-wise hard attention vector, concurrently with learning every task; Uses hard attention masks (acts as "inhibitory synapses") to activate or deactivate the output of the units of every layer in the forward pass.

Hard Attention to the Task (HAT)

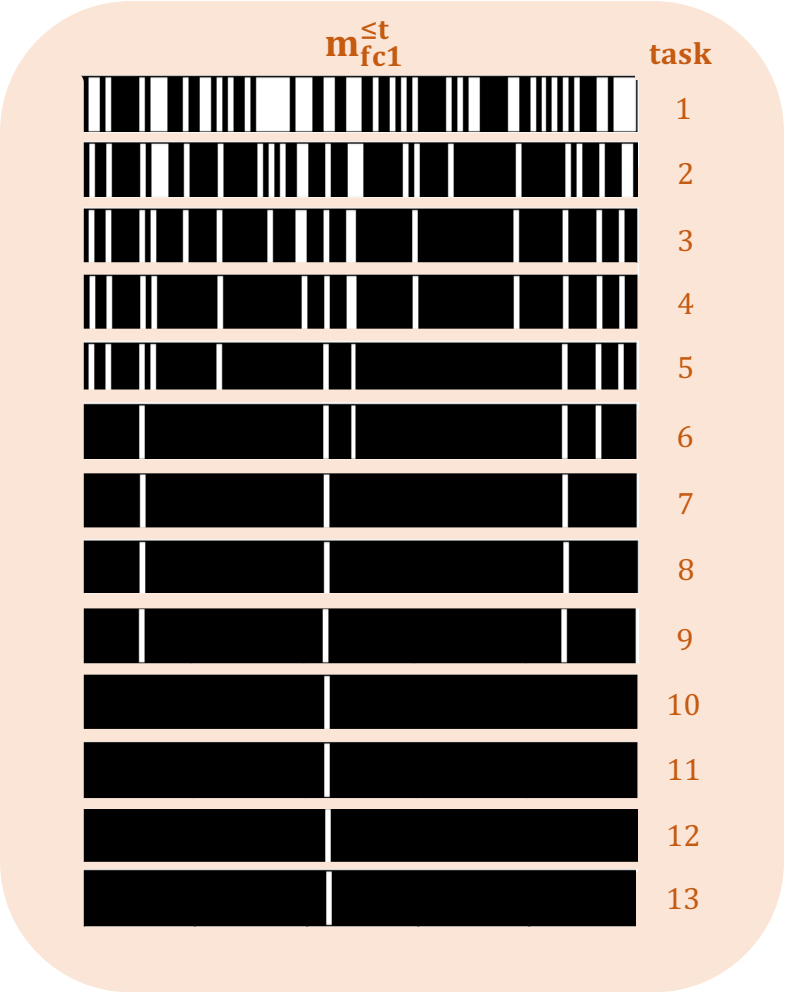
Conditions learning a new task (the gradients) on the attention vectors of the previous tasks in the backward pass



The network capacity rapidly saturated (< 10 tasks), with the performance dropping dramatically - leans towards stability

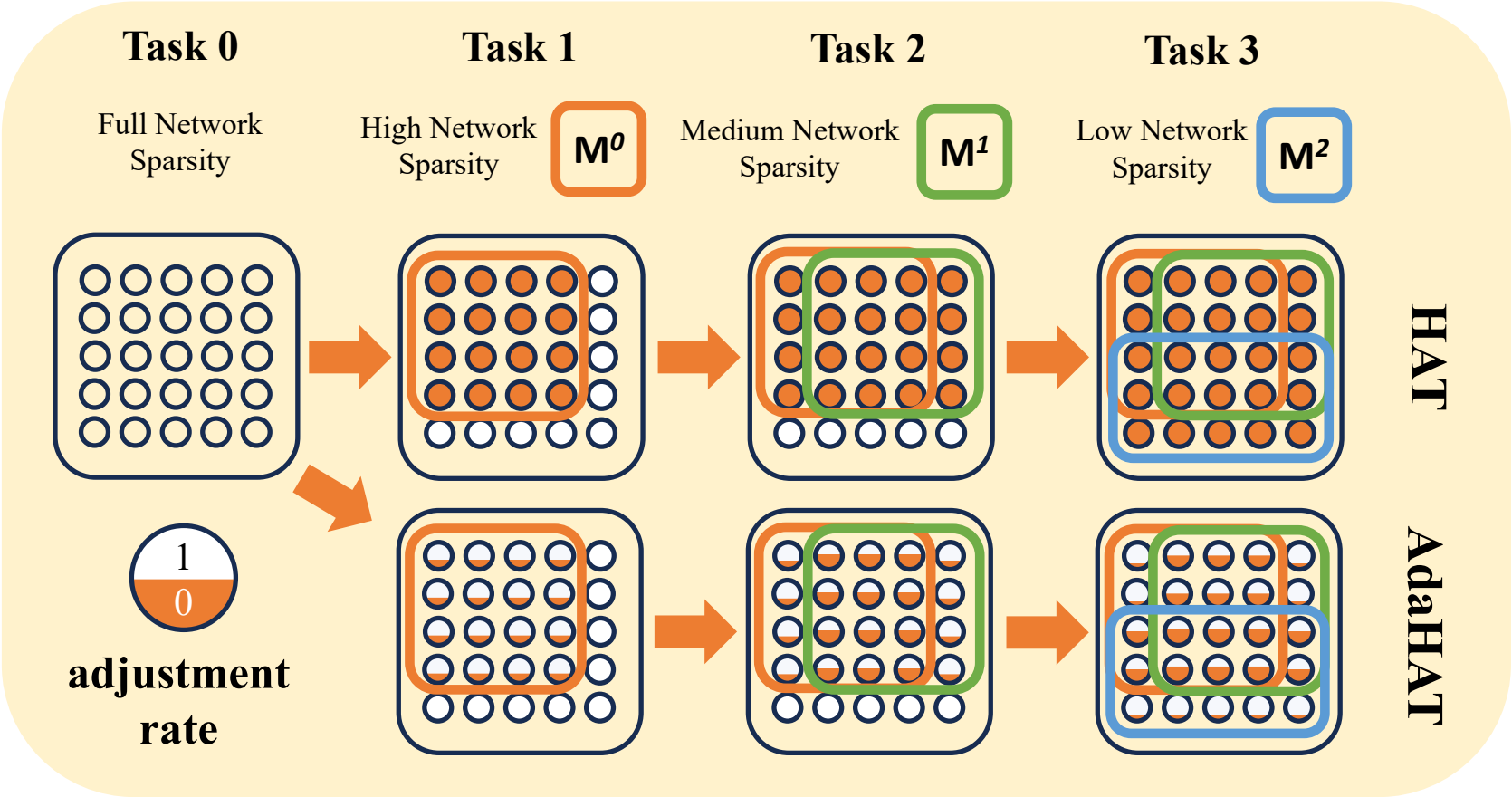
Limitations of HAT

- Network capacity runs out after a number of tasks (< 10), hence no active parameters for new tasks – not suitable for long sequences
- Sacrifice plasticity for stability
- Not adaptive to task sequences- manually tune hyperparameters to allocate network capacity, but in CL typically no prior knowledge on the number of tasks in a sequence



More active parameters become static as more tasks come in

HAT versus AdaHAT



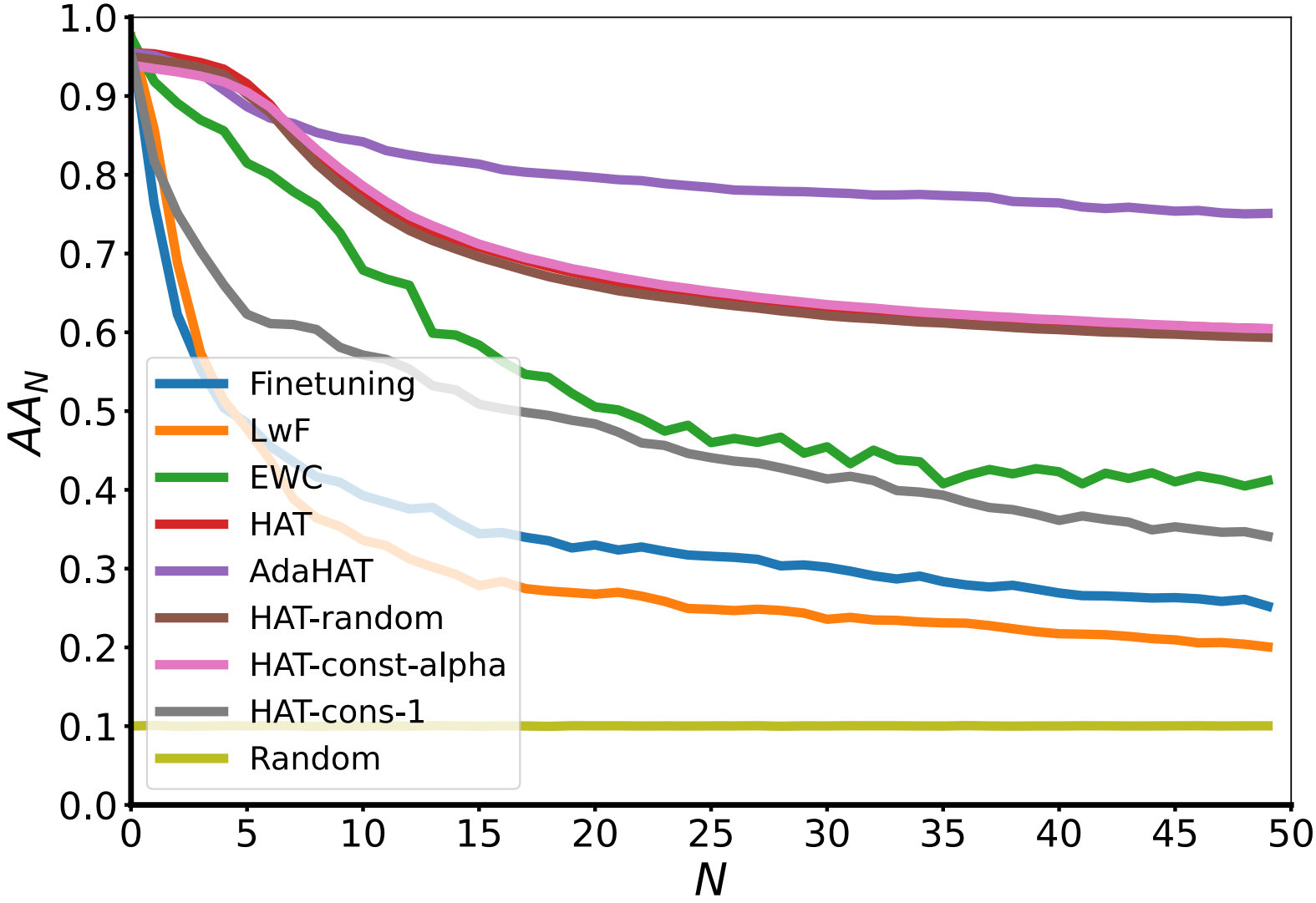
Adaptive updates to static parameters, taking into account their importance and the network capacity, to reuse them

AdaHAT Achieves Best Performance over Sequences of 20 Tasks

Dataset	Approach	AA(%)	FR (%)	BWT(%)	FWT (%)
Permuted MNIST	Finetuning	32.62 ± 1.60	-73.78 ± 1.84	-68.10 ± 1.68	63.51 ± 0.03
	LwF	26.95 ± 1.80	-80.35 ± 2.08	-72.59 ± 1.91	62.04 ± 0.09
	EWC	52.25 ± 2.46	-51.38 ± 2.83	-42.04 ± 2.67	58.12 ± 0.15
	HAT	67.64 ± 1.27	-33.70 ± 1.46	-0.11 ± 0.18	32.49 ± 1.12
	HAT-random	66.43 ± 1.21	-35.10 ± 1.39	-0.27 ± 0.49	31.40 ± 1.22
	HAT-const-alpha	68.08 ± 1.18	-33.20 ± 1.36	$-1 * e^{-3} \pm 0.00$	32.92 ± 1.23
	HAT-const-1	48.83 ± 4.35	-55.14 ± 5.02	-49.68 ± 4.40	62.26 ± 0.21
	AdaHAT	$\mathbf{79.90 \pm 2.40}$	$\mathbf{-19.43 \pm 2.76}$	-14.68 ± 2.48	59.96 ± 0.09
Split CIFAR-100	Finetuning	24.34 ± 0.73	-91.66 ± 1.32	-54.00 ± 1.00	53.10 ± 0.55
	LwF	34.56 ± 0.94	-70.91 ± 2.05	-48.03 ± 1.01	57.61 ± 0.40
	EWC	30.23 ± 1.61	-79.84 ± 3.13	-54.05 ± 1.28	59.20 ± 0.50
	HAT	32.44 ± 1.58	-74.71 ± 3.37	-45.59 ± 1.49	53.11 ± 0.34
	HAT-random	31.41 ± 1.29	-76.98 ± 2.45	-48.80 ± 1.33	52.76 ± 0.57
	HAT-const-alpha	32.16 ± 2.48	-75.04 ± 5.16	-44.49 ± 2.57	51.86 ± 0.82
	HAT-const-1	32.40 ± 1.40	-75.58 ± 3.08	-48.80 ± 1.72	56.30 ± 0.36
	AdaHAT	$\mathbf{38.74 \pm 2.24}$	$\mathbf{-62.37 \pm 4.64}$	-42.11 ± 2.02	56.33 ± 0.82

AdaHAT outperforms all baselines in terms of average accuracy (AA) and forgetting ratio (RF)

AdaHAT Achieves the Consistent Performance over Longer Sequences of 50 Tasks



Conclusion

HAT

- Tends to tilt the stability-plasticity trade-off towards stability
- Suffers from the insufficient network capacity problem in long sequence of tasks

AdaHAT

- Balances the trade-off in an adaptive manner
- Suits for long sequences of tasks
- Achieves better performance than the baselines, in particular HAT

Future work

- Explore and exploit more subtle information about previous tasks

Contact wangpengxiang@stu.pku.edu.cn for further technical discussions

Hard Conditioning Gradients in HAT

- Condition gradients on previous tasks by hard-clipping the gradients with the hard adjustment rate

$$g'_{l,ij} = a_{l,ij} \cdot g_{l,ij}$$

$$a_{l,ij} = 1 - \min\left(m_{l,i}^{<t}, m_{l-1,j}^{<t}\right) \in \{0,1\}$$

Cumulative Attention Vector

$$m_l^{\leq t} = \max\left(m_l^t, m_l^{t-1}\right) \in \{0,1\}$$

Soft Conditioning Gradients in AdaHAT

- Condition gradients on previous tasks by soft-clipping the gradients with the adaptive adjustment rate

$$g'_{l,ij} = a^{\star}_{l,ij} \cdot g_{l,ij}$$

$$a^{\star}_{l,ij} = 1 - \min\left(m_{l,i}^{<t,\mathrm{sum}}, m_{l-1,j}^{<t,\mathrm{sum}}\right) \in [0,1]$$

Summative Attention Vector

$$m_l^{\leq t,\mathrm{sum}} = \max\left(m_l^{t,\mathrm{sum}}, m_l^{t-1,\mathrm{sum}}\right) \in [0,1]$$