ASP-DAC 2025
30th Asia and South Pacific Design Automation Conference
Date: January 20-23, Tokyo

LightCL: Compact Continual Learning with Low Memory Footprint For Edge Device

Zeqing Wang, Fei Cheng*, Kangye Ji, Bohu Huang

School of Computer Science and Technology
Xidian University

2025/1/21

西安電子科技大學 XIDIAN UNIVERSITY
计算机科学与技术学院 SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY
国家示范性软件学院 NATIONAL PILOT SCHOOL OF SOFTWARE ENGINEERING

Outline

- Background
- Challenge
- Method
- Experimental Results
- Conclusion

1 Background

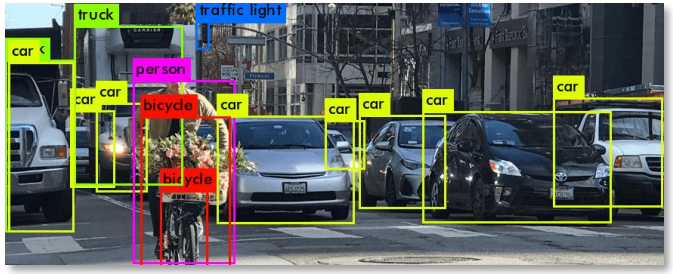
Edge devices are everywhere

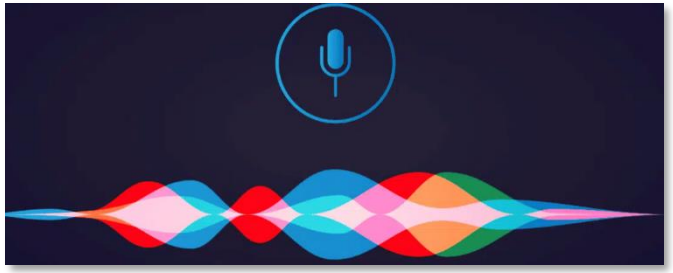
smartphone

drone

robot

AI empowers edge devices more intelligence



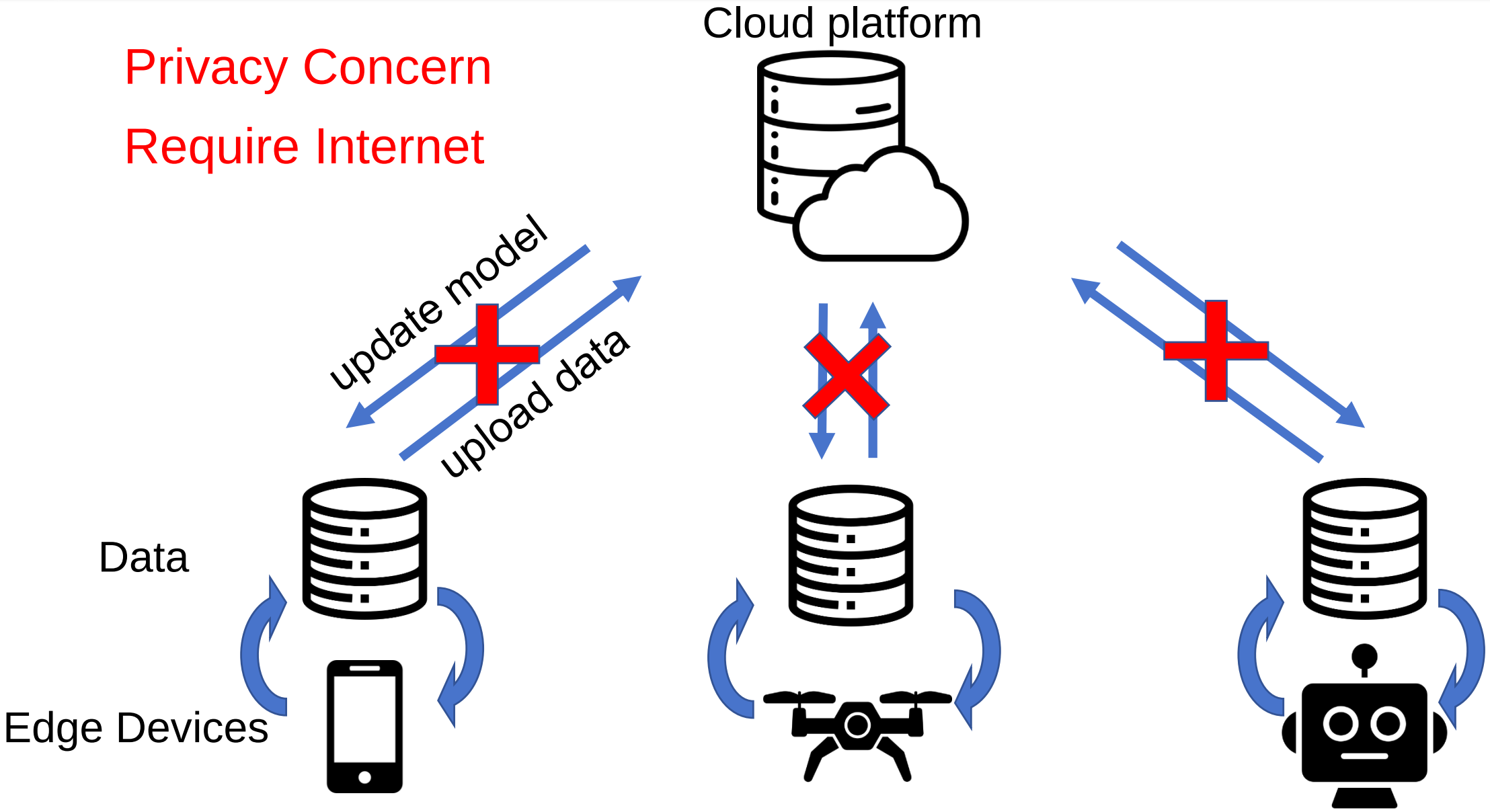


Continual Learning !

1 Background

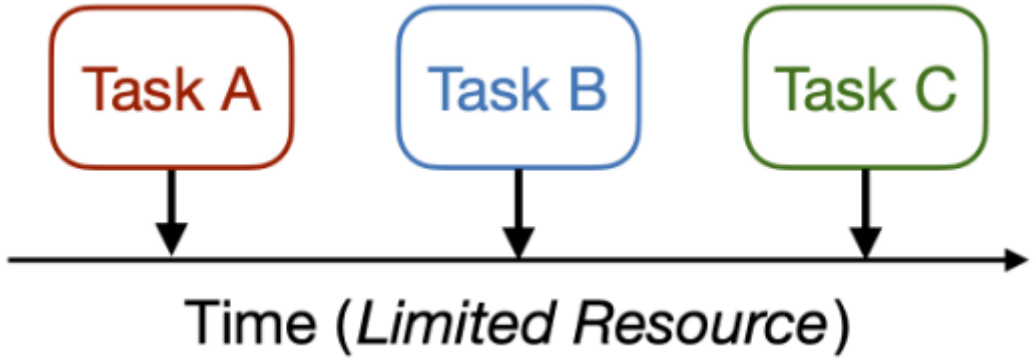
Privacy Concern

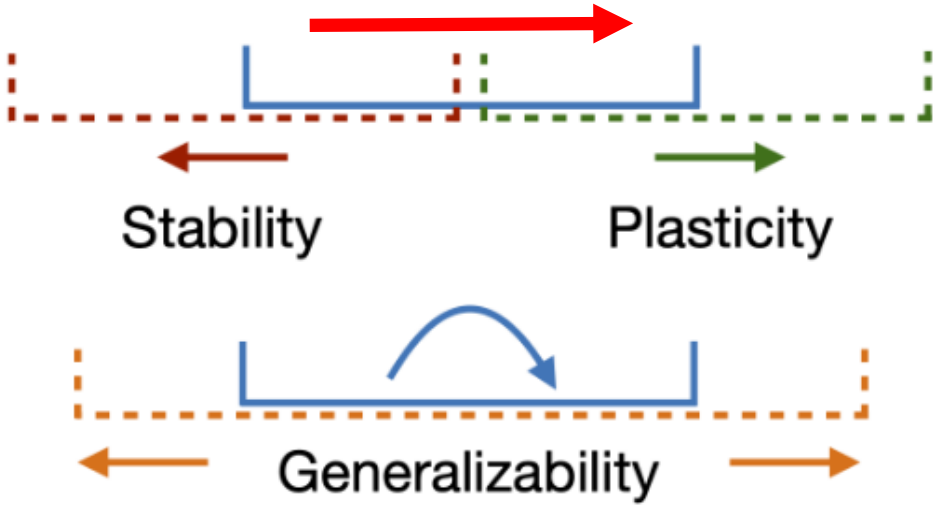
Require Internet



2 Challenge

- **Catastrophic Forgetting**
  - Model training on a new task tends to "forget" the knowledge learned from previous tasks
  - Key idea is to make trade-off between **learning plasticity** and **memory stability** for gaining **generalizability**

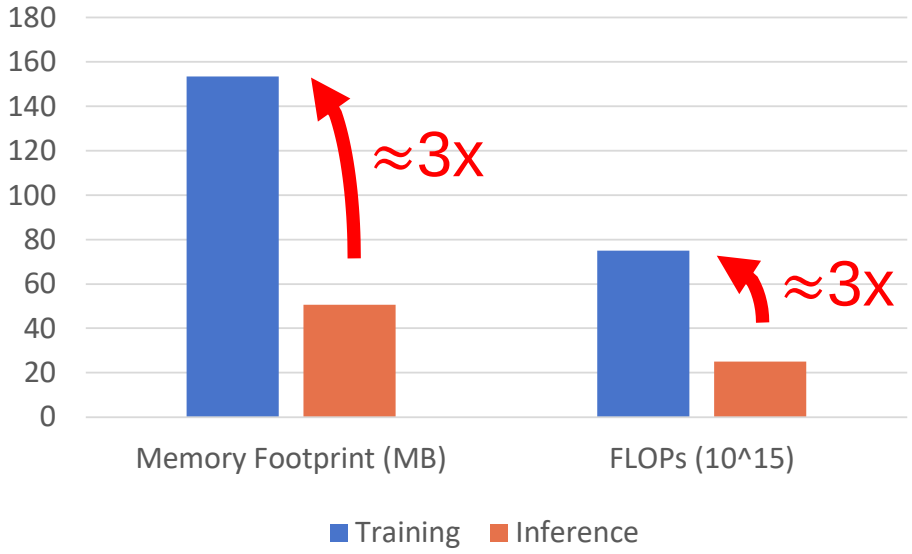


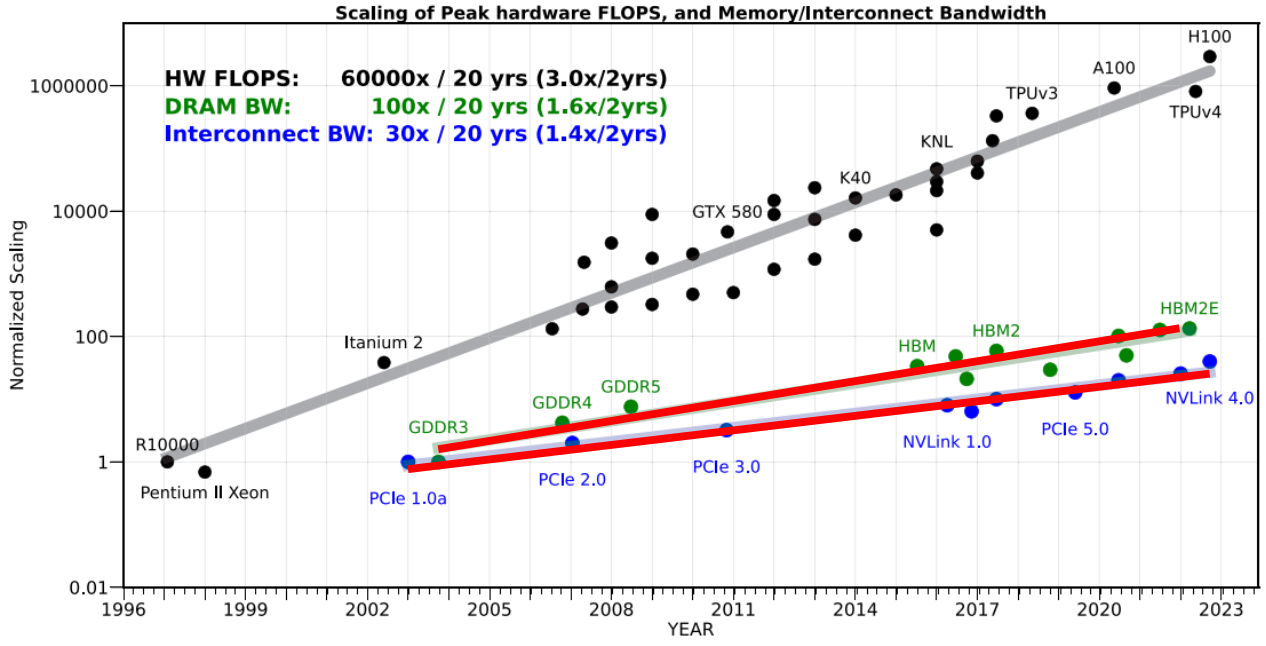


A Comprehensive Survey of Forgetting in Deep Learning Beyond Continual Learning [Wang *et al*., TPAMI 2024]

2 Challenge

- **Limited Resource in Edge Device**
  - Limited computational power and memory
  - Training consumes more than inference
  - Memory has become the primary bottleneck in AI applications

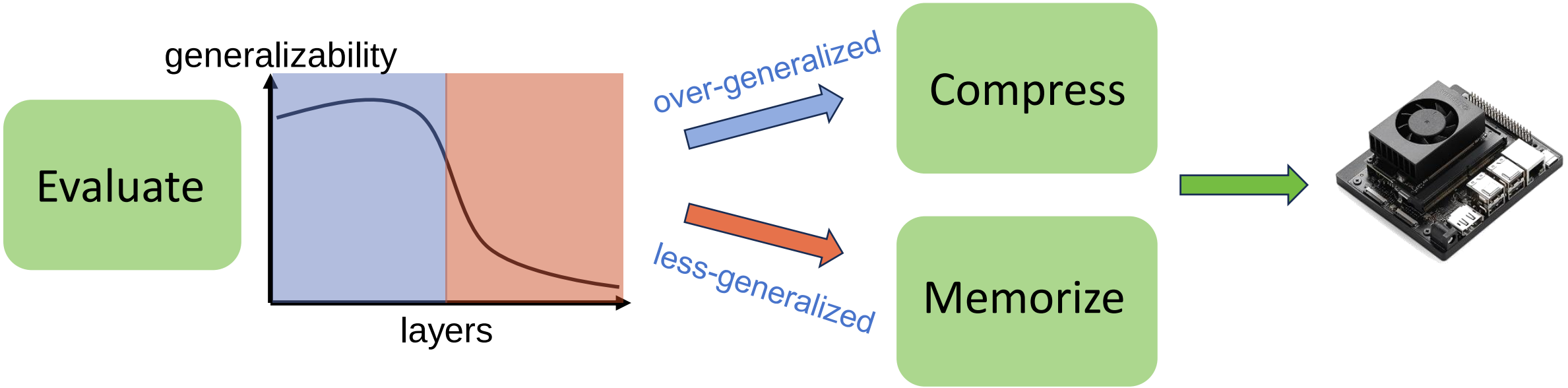




AI and Memory Wall [GHOLAMI *et al.*, Micro 2024]

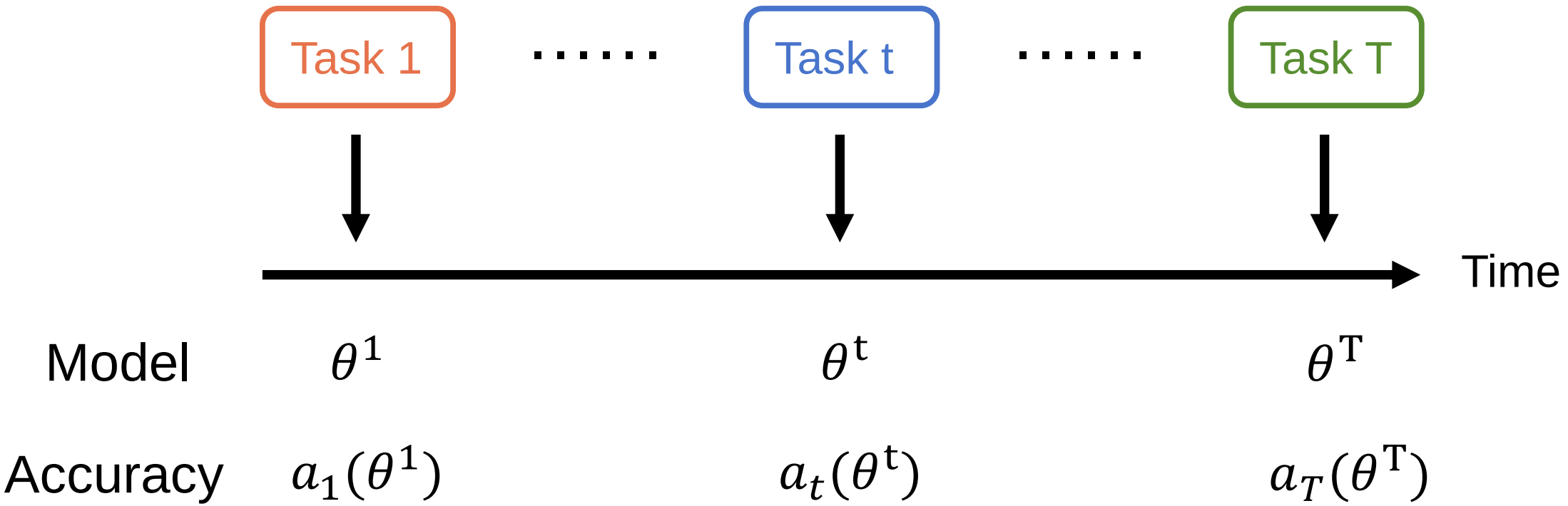
3 Method

- **Motivation**
  - CL process has the potential to leverage previous knowledge when training on new tasks ⟹ Redundancy in training



3 Method

- **CL Setting**



3 Method

- **Analysis of Generalizability**
  - Memory stability (**MS**) and Learning plasticity (**LP**) are two different characteristics of **generalizability**
  - **MS** denotes the loss of previous knowledge
  - **LP** denotes the adaptation to new knowledge

$$\theta^{new_i} = \begin{cases} \theta_j^{new_i} = \theta_j^T & \text{if } j \neq i, \\ \theta_i^{new_i} = \theta_i^{T-1} & \text{otherwise.} \end{cases} \quad (1)$$

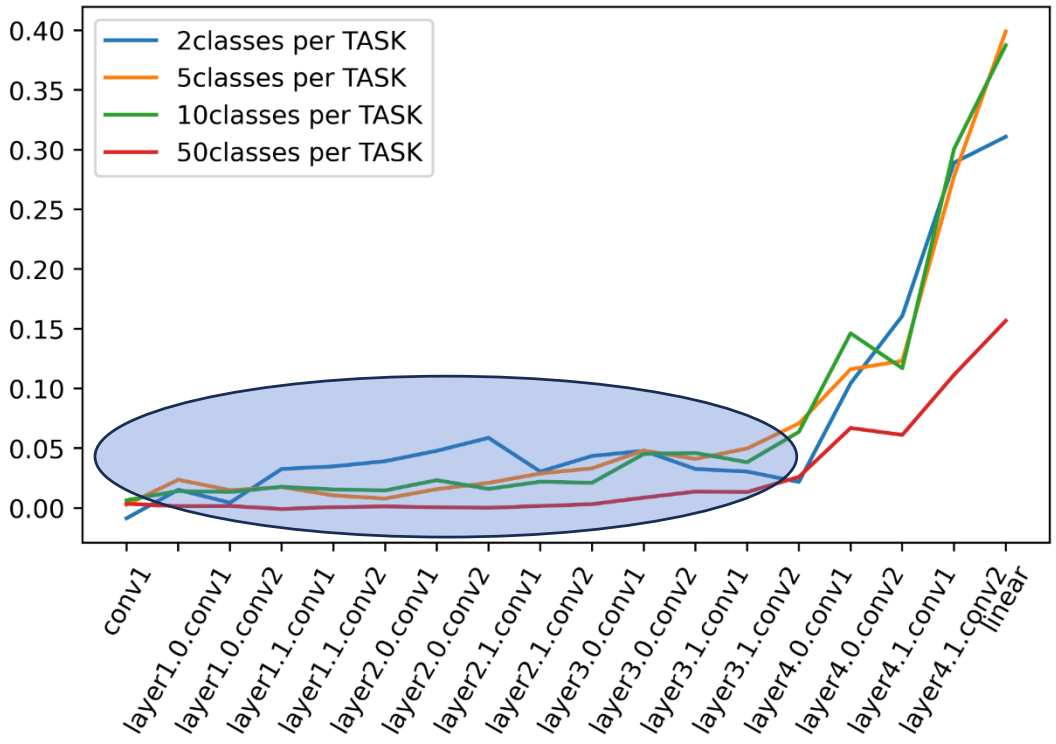
$$\begin{cases} MS = a_{T-1}(\theta^{new_i}) - a_{T-1}(\theta^T), \\ LP = a_T(\theta^{new_i}) - a_T(\theta^T). \end{cases} \quad (2)$$

A Comprehensive Survey of Forgetting in Deep Learning Beyond Continual Learning [Wang *et al*., TPAMI 2024]

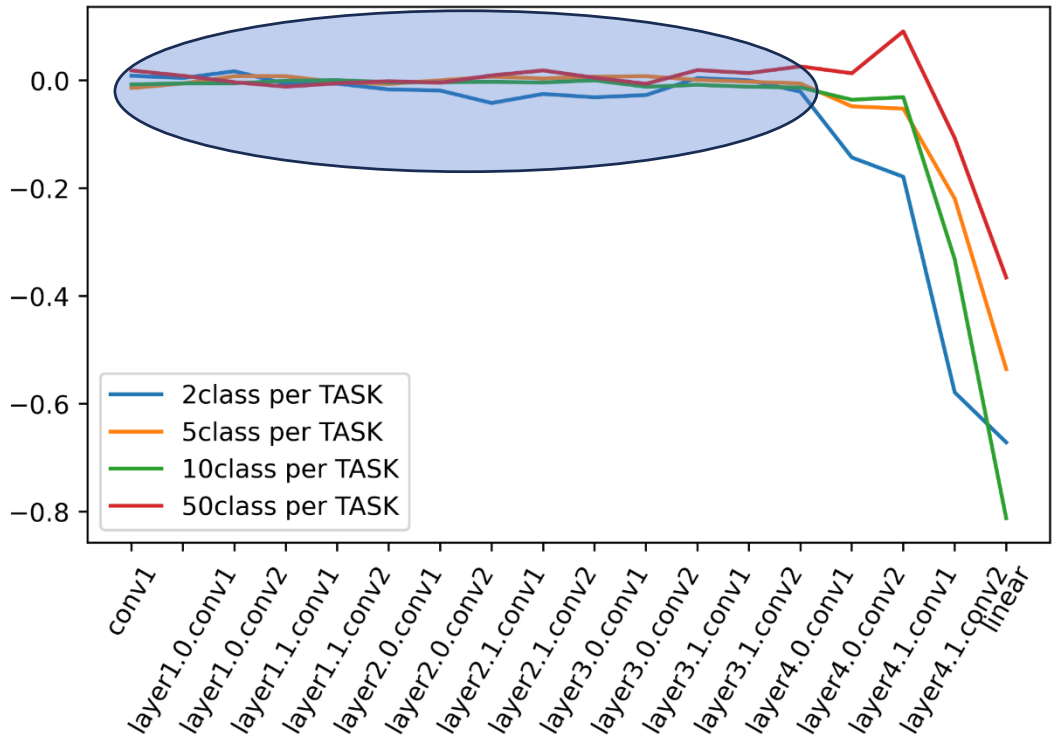
3 Method

- **Analysis of Generalizability**
  - During CL, lower and middle layers have strong generalizability, and deeper layers have less generalizability

MS

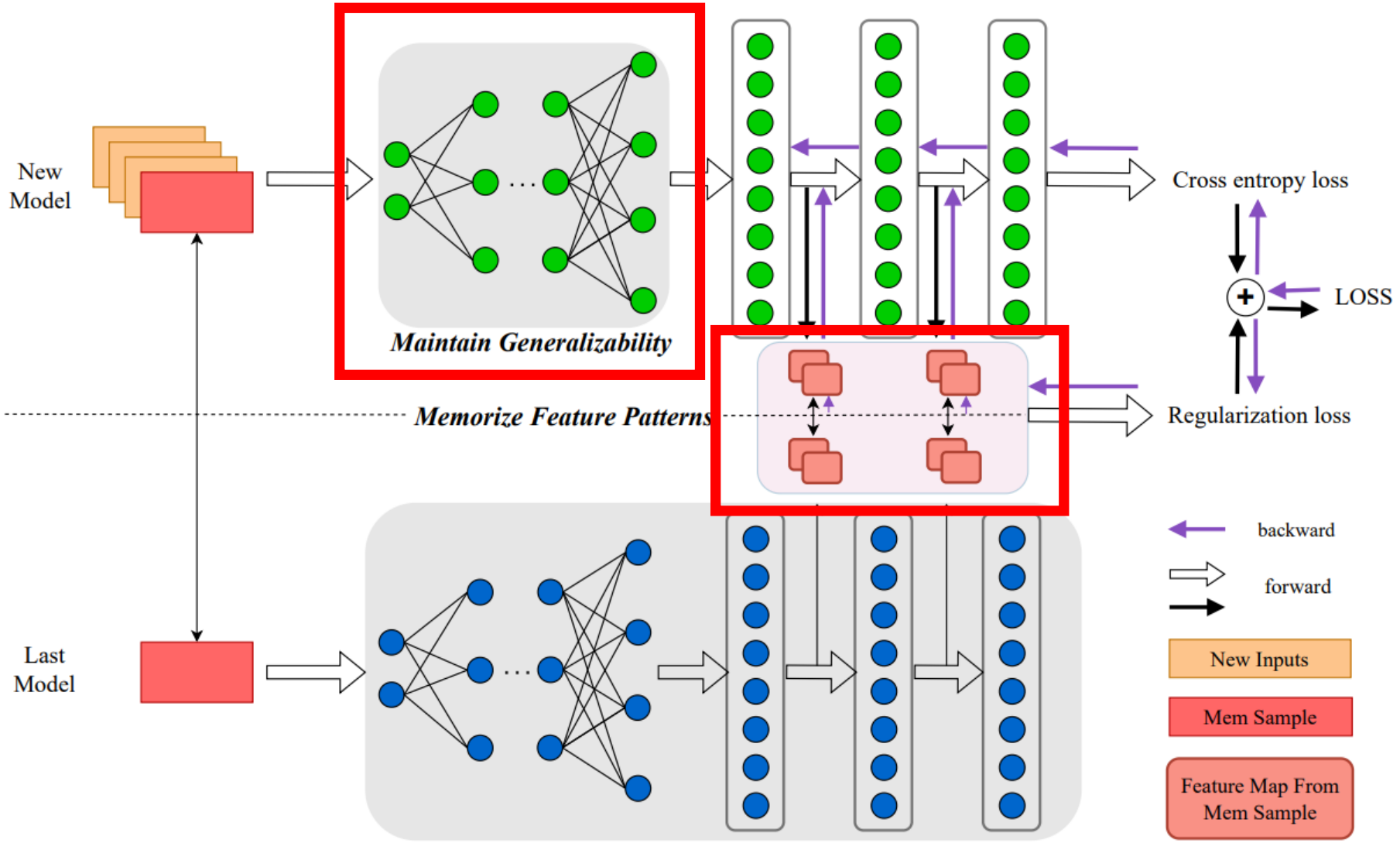


LP



3 Method

- **Proposed Architecture**



3 Method

- **Maintain Generalizability**
  - Freezing lower and middle layers to maintain generalizability during CL
  - Reduce resource consumption and maintain generalized knowledge
- **Memorize Feature Patterns**
  - Feature maps consume lower consumption in deeper layers
  - The output mode of feature mapping is not related to sample information, but to the location of feature map
  - **Regulate features important for previous tasks during learning new task without accessing to previous samples**

3 Method

- **Memorize Feature Patterns**
  - **Select references**
    - Recognize important positions by $\boldsymbol{l1}$-norm with certain ratio
    - All important positions are stored in $\boldsymbol{I} = \{ (\boldsymbol{i}, \boldsymbol{j}) \}$
    - Select a few samples from the new task to run it on the model and get feature map standards, called $\boldsymbol{FS}$, as references
  - **Regulation**

$$\mathcal{L}_f(FM, FS) = \beta \sum_{(i,j) \in I} (FM_{i,j} - FS_{i,j})^2.$$

  - **Whole loss function**

$$\mathcal{L}(\theta^t) = \mathcal{L}_c(f(\theta^t, x_t), y_t) + \mathcal{L}_f(FM, FS).$$

4 Experimental Results

- **Scenarios**
  - Task incremental learning (TIL) and Class incremental learning (CIL) scenarios
- **Models**
  - ResNet-18 and ResNet-50 (pretrained on the ImageNet32×32)
- **Datasets**
  - Split CIFAR-10 (5 tasks) and Split Tiny-ImageNet (10 tasks) datasets
- **Metrics**

$$AA = \frac{1}{T}\sum_{i=1}^{T} a_i\left(\theta^T\right)$$

Average Accuracy

Forward
&
Backward

Training FLOPs

Parameters
Gradients
Feature Maps

Memory Footprint

4 Experimental Results

- **Main Result**

Method	Buffer size	Sparsity	Split CIFAR-10 CIL(↑)	Split CIFAR-10 TIL(↑)	Split CIFAR-10 FLOPs $\times 10^{15}$(↓)	Split CIFAR-10 Mem (MB)(↓)	Split TinyImageNet CIL(↑)	Split TinyImageNet TIL(↑)	Split TinyImageNet FLOPs $\times 10^{16}$(↓)	Split TinyImageNet Mem (MB)(↓)
JOINT	/	0.0	94.67±0.15	99.04±0.07	37.5	153.4	61.38±0.12	83.14±0.52	120	357.9
SGD	/	0.0	34.21±3.06	86.19±1.32	7.5	153.4	10.63±0.36	34.24±0.84	12	357.9
EWC [14]	/	0.0	40.42±3.81	89.77±3.01	7.5	196.0	11.17±1.55	34.70±1.55	12	400.5
PNN [25]	/	0.0	/	94.62±0.52	10.5	323.9	/	73.94±3.56	26.4	528.4
FDR [2]	15	0.0	25.64±2.52	85.54±3.08	11	185.4	10.80±0.71	34.42±0.38	17.7	485.7
ER [5]	15	0.0	54.44±2.82	92.25±1.92	11	185.4	11.82±0.31	36.76±0.57	17.7	485.7
DER++ [3]	15	0.0	40.75±2.55	85.68±4.35	14.5	217.3	7.98±1.84	35.88±0.41	23.3	613.5
SparCL$_{ER}$ [30]	15	0.9	36.98±2.42	85.33±2.49	1.1	106.6	10.01±0.55	33.01±0.78	1.8	407.0
SparCL$_{DER++}$ [30]	15	0.9	35.58±0.84	85.28±1.00	1.5	140.6	9.12±1.02	31.86±1.77	2.3	536.8
LightCL	**15**	**0.0**	**55.53±0.35**	**96.28±1.08**	**5.4**	**92.8**	**12.79±0.30**	**37.39±0.63**	**8.6**	**128.0**
		0.9	**38.31±3.93**	**86.88±1.69**	**2.3**	**25.0**	**8.95±0.35**	**33.94±0.52**	**3.7**	**87.1**

'/' indicates the corresponding item is not needed (Buffer is not required in JOINT, SGD, EWC, and PNN) or cannot run (PNN cannot run under CIL settings). FLOPs and Mem are calculated theoretically with the precision to one decimal place.

reduce at most 6.16×

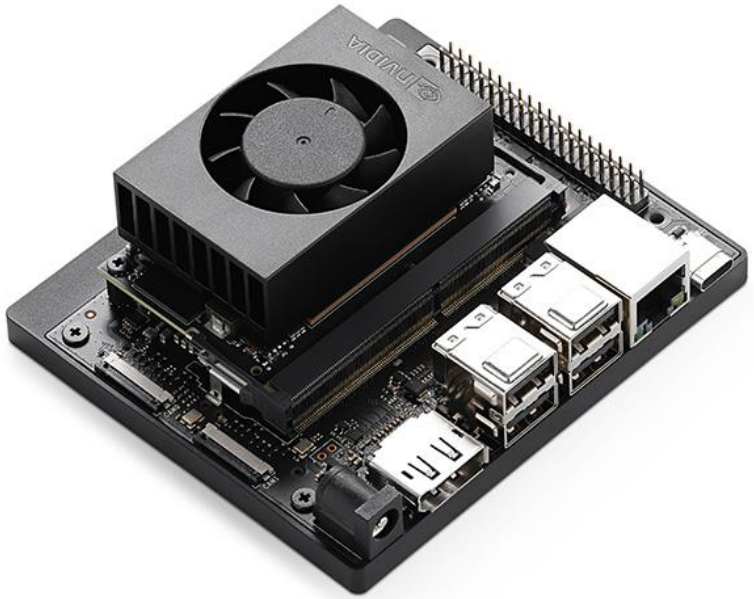
4 Experimental Results

- **Other Results**
  - Training from scratch
  - First task as pretraining process and LightCL is conducted on the subsequent tasks
  - Still outstanding in ResNet-18 and ResNet-50

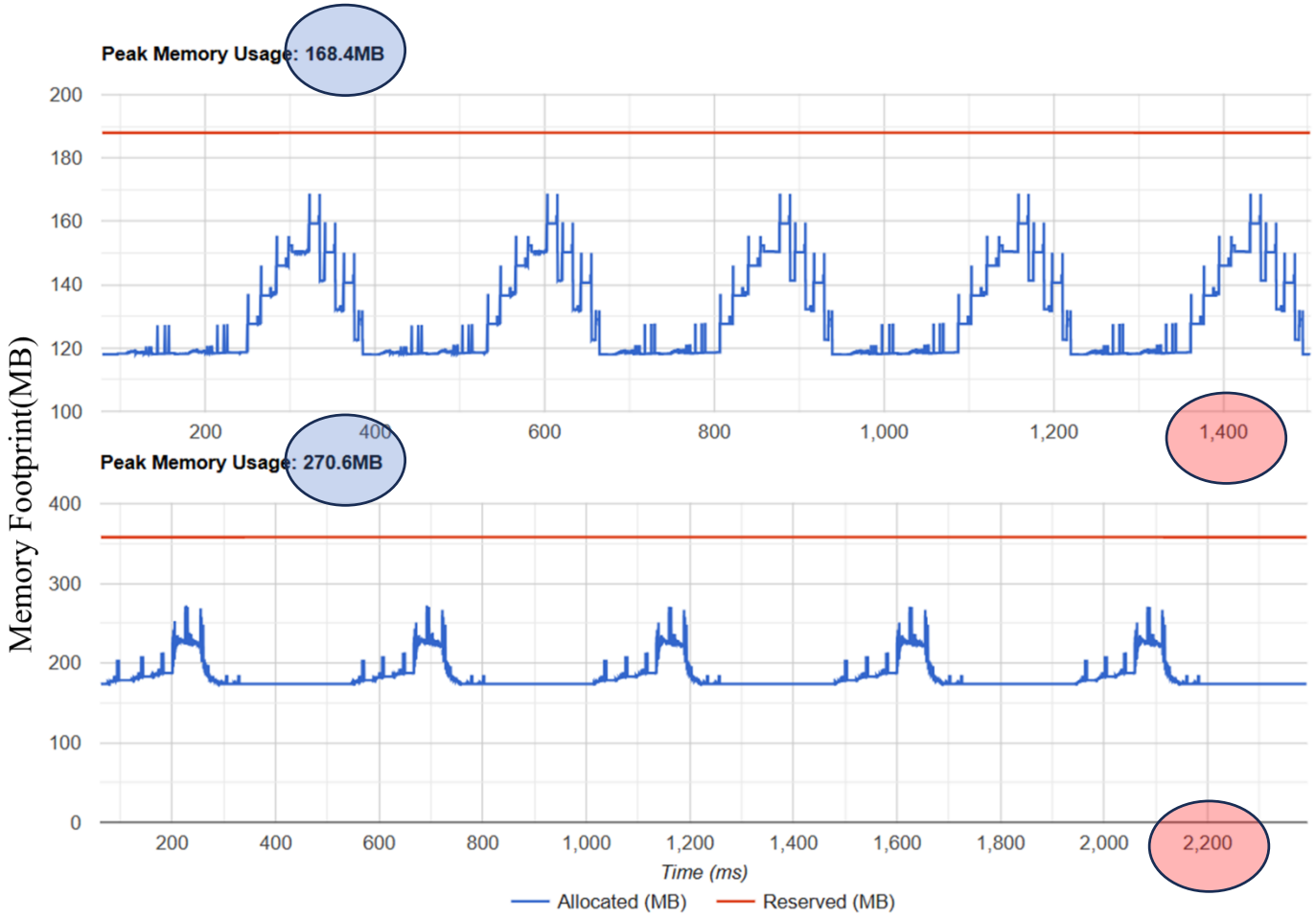
Method	Split CIFAR-10		Split TinyImageNet	
	CIL(↑)	TIL(↑)	CIL(↑)	TIL(↑)
ResNet-18				
JOINT	92.20±0.15	98.31±0.12	59.99±0.19	82.04±0.10
SGD	21.00±1.61	65.78±3.52	7.28±0.04	24.99±0.27
EWC	21.32±0.17	66.13±0.17	8.58±0.22	24.47±0.30
PNN	/	95.13±0.72	/	61.88±1.00
FDR	26.19±3.14	77.61±0.68	8.58 ±0.21	25.20±0.29
ER	35.90±1.53	82.40±1.64	8.76±0.23	26.29±1.18
DER++	36.25±1.99	82.05±1.63	6.53±1.44	27.21±1.85
$\text{SparCL}_{\text{ER}}$	30.09±1.60	75.12±0.63	7.70±0.13	25.24±0.55
$\text{SparCL}_{\text{DER++}}$	34.36±1.32	78.56±2.88	8.16±0.58	26.53±0.54
LightCL	**37.86±2.16**	**85.69±0.98**	**9.03±0.16**	**33.74±0.68**
$\text{LightCL}_{0.9}$	**35.72±2.67**	**83.37±1.10**	**8.76±0.63**	**36.86±0.29**
ResNet-50				
JOINT	94.66±0.15	98.93±0.09	61.04±0.42	82.70±0.42
SGD	14.16±0.39	72.05±1.52	8.55±0.20	29.99±0.23
ER	24.19±1.43	76.65±1.50	8.81±0.62	31.38±0.06
DER++	26.42±1.67	75.13±0.69	6.23±1.23	27.22±2.70
LightCL	**35.08±0.50**	**81.49±0.92**	**9.23±0.67**	**33.81±1.47**

4 Experimental Results

- **Other Results**

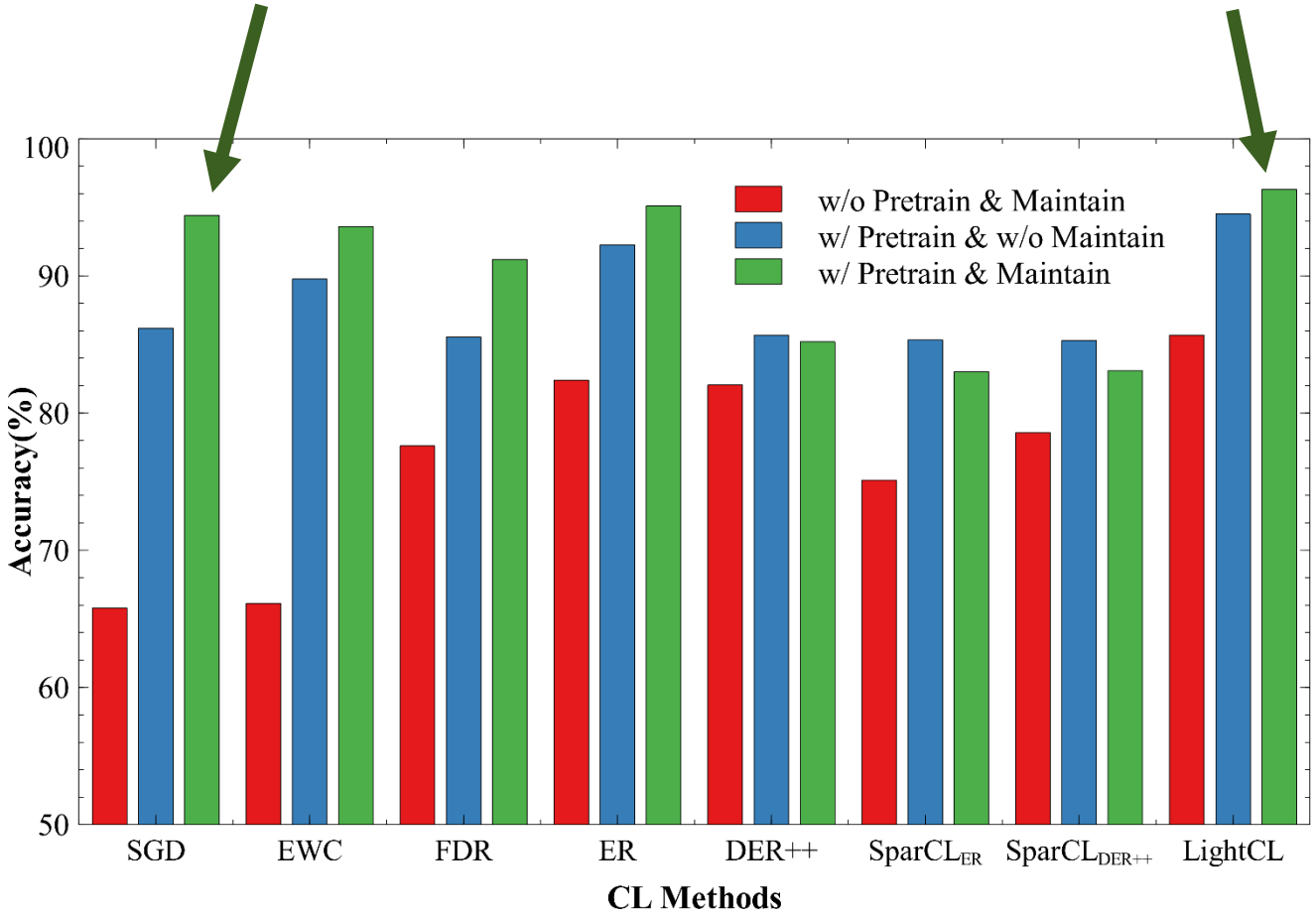


Training on Jetson Nano B01



4 Experimental Results

- **Ablation Study**
  - All CL methods can enhance performance with the pre-trained model
  - ***Maintain Generalizability*** can even further enhance performance
  - ***Memorize Feature Patterns*** can further increase accuracy



5 Conclusion

- It is the **first** study to propose two new metrics of learning plasticity (**LP**) and memory stability (**MS**) to quantitatively evaluate generalizability
- Lower and middle layers have more generalizability, while deeper layers hold the opposite.
- We propose a new method, LightCL, to overcome catastrophic forgetting through ***Maintain Generalizability*** and ***Memorize Feature Patterns***
- LightCL shows great improvement in delaying forgetting and memory efficiency

Q & A

Thank you for your attention!

Presenter: Zeqing Wang

Email: wang.zeqing@stu.xidian.edu.cn

Github