YONSEI UNIVERSITY

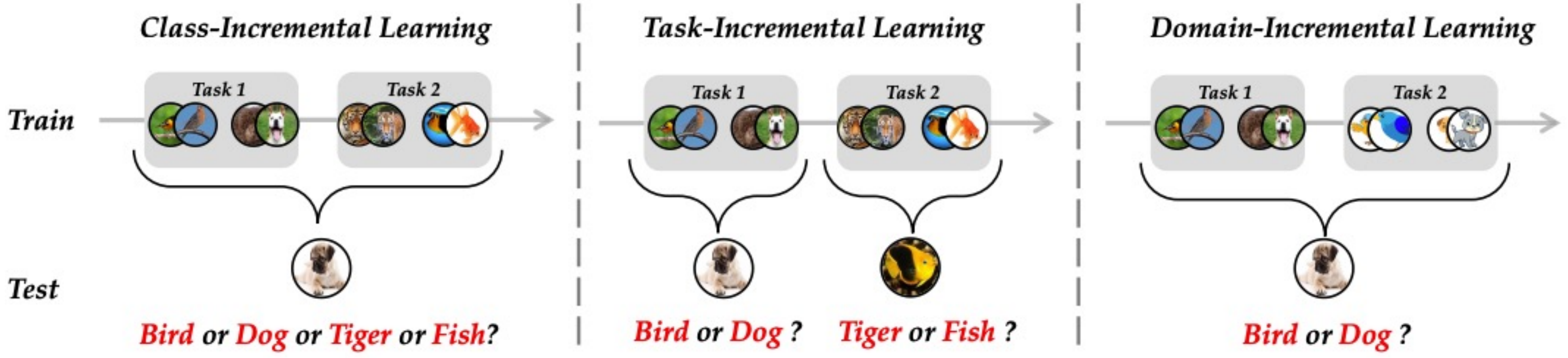
A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning

Shi Won Kim,
Digital Healthcare Lab,
Department of Digital Analytics,
Yonsei University College of Computing

Severance

Background

- Incremental learning (= continual learning)
  - A learning system that continually acquires the knowledge of **incoming new classes**
  - Assume **a sequence of training tasks** from a continuous data stream
  - Build a universal classifier for all seen classes with limited resources



[1] Zhou, Da-Wei et al. "Deep Class-Incremental Learning: A Survey." ArXiv abs/2302.03648 (2023).

Background

- Catastrophic forgetting
  - **Forgetting the characteristics of former classes** when trained with new class instances*
  - Leads to drastic degradation in the performance of old tasks (prediction of old classes)
  - A common problem with gradient-based learning methods
- Stability-plasticity dilemma
  - Stability denotes the ability to **maintain former knowledge**
  - Plasticity represents the ability to **learn new patterns**
  - Acquire knowledge from the current task and preserve knowledge from former tasks

* A single observation or record of data

Background (CIL Taxonomy)

- Data-centric
  - Data replay (real or synthetic) → utilize former data (exemplar set) as **rehearsal memory**
  - **User privacy issues** when saving exemplars from the history (← exemplar-free manner)
- Model-centric
  - Dynamic networks → **backbone expansion** (DER, FOSTER, MEMO, etc.)
  - Require **large memory budgets** (unsuitable for CIL on edge devices)
- Algorithm-centric
  - **Knowledge distillation** → logit, feature, relational distillation
  - Outperforms dynamic networks given limited memory (overturns with adequate memory)

[1] Zhou, Da-Wei et al. "Deep Class-Incremental Learning: A Survey." ArXiv abs/2302.03648 (2023).

MEMO: Memory-efficient Expandable Model

Introduction

- Typical CIL methods
  - Saving limited exemplars from former classes can boost the performance of CIL models
  - Saving backbones from the history pushes the performance towards the upper bound*
- Unfair comparison of CIL methods
  - Model-based methods train **multiple backbones** continually → extra memory budget
  - Should align the performance measure at **the same memory scale** for a fair comparison
- Contributions
  - Holistically evaluate different CIL methods at diverse memory budget scenarios
  - Propose a simple yet effective **MEMO** that extends diverse features with modest memory cost

* Saving all the streaming data for offline training (requires an unlimited memory budget for storage)

Preliminaries

- Problem definition
  - A sequence of $\boldsymbol{B}$ **training tasks** without overlapping classes
  - The aim is to acquire new knowledge while preserving the knowledge from former tasks
  - A fixed number of representative instances from the old classes → **exemplar set**
  - We can only access the current training data and the exemplar set
  - The incremental model is decomposed into the **embedding module** and **linear layers**

$$i.e., f(\mathbf{x}) = W^\top \phi(\mathbf{x}), \text{ where } \phi(\cdot) : \mathbb{R}^D \to \mathbb{R}^d, W \in \mathbb{R}^{d \times |\mathcal{Y}_b|}$$

Preliminaries

- Overcome forgetting in class-incremental learning
  - Knowledge distillation

$$\mathcal{L}(\mathbf{x}, y) = (1-\lambda) \underbrace{\sum_{k=1}^{|\mathcal{Y}_b|} -\mathbb{I}(y=k) \log \mathcal{S}_k(W^\top \phi(\mathbf{x}))}_{\textit{Cross Entropy}} + \lambda \underbrace{\sum_{k=1}^{|\mathcal{Y}_{b-1}|} -\mathcal{S}_k(\bar{W}^\top \bar{\phi}(\mathbf{x})) \log \mathcal{S}_k(W^\top \phi(\mathbf{x}))}_{\textit{Knowledge Distillation}}, \quad (1)$$

  - Feature aggregation

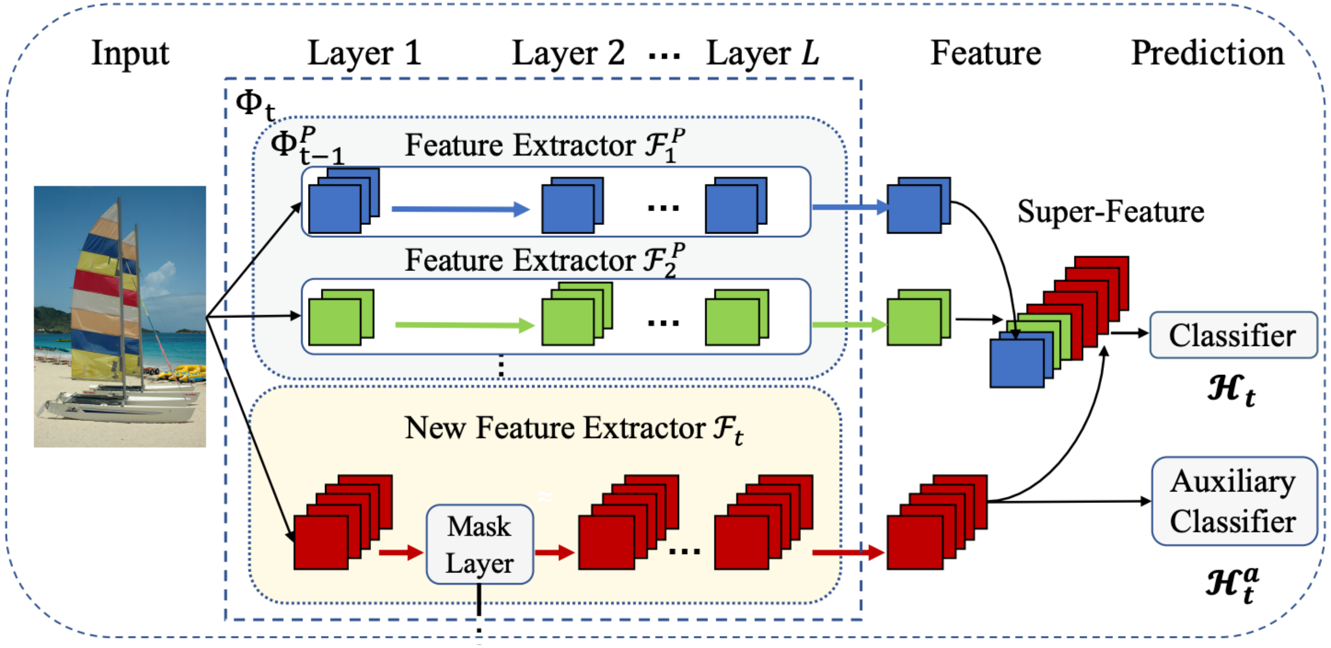
$$\mathcal{L}(\mathbf{x}, y) = \sum_{k=1}^{|\mathcal{Y}_b|} -\mathbb{I}(y=k) \log \mathcal{S}_k(W_{new}^\top [\bar{\phi}_{old}(\mathbf{x}), \phi_{new}(\mathbf{x})]) \,. \quad (2)$$

Experimental Setup

- Dataset
  - **CIFAR100 (ResNet32)**
  - ImageNet100/1000 (ResNet18)
  - Dataset split → Base-x and Inc-y
  - Class order → random seed 1993 (the common setting in CIL)
- Implementation details
  - Batch size = 128
  - Epochs = 170
  - SGD momentum
  - Initial learning rate = 0.1
  - Learning rate decay = 0.1 (80 and 150 epochs)

Methods

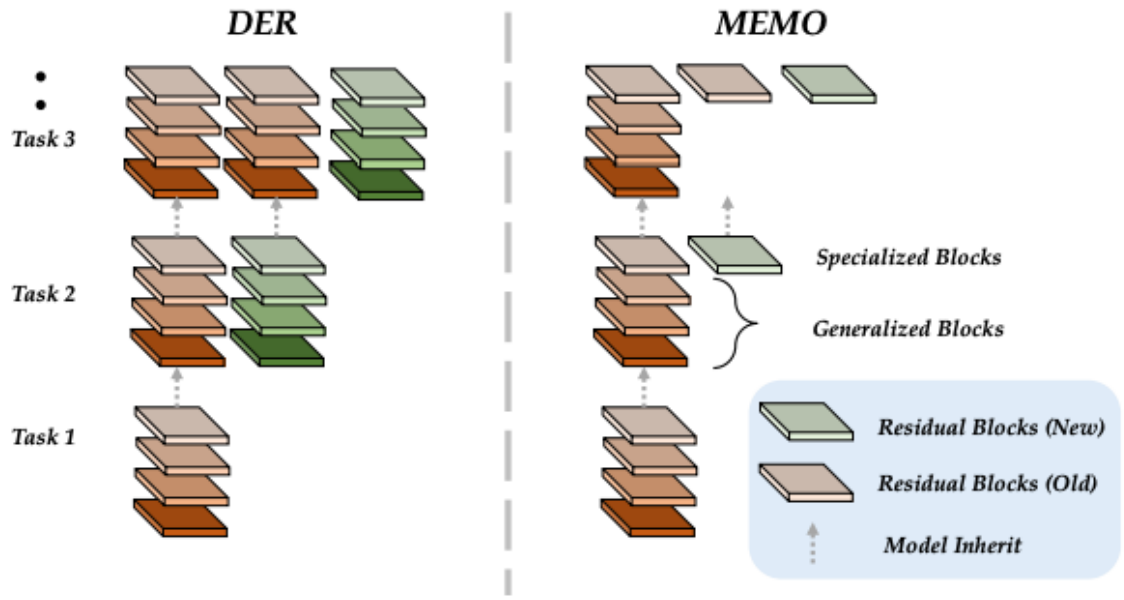
- Benchmark backbone
  - DER (dynamically expandable representation) [2]
  - Creates a new backbone per new task → requires saving all the embeddings during inference



[2] Yan, Shipeng et al. "DER: Dynamically Expandable Representation for Class Incremental Learning." (CVPR) (2021).

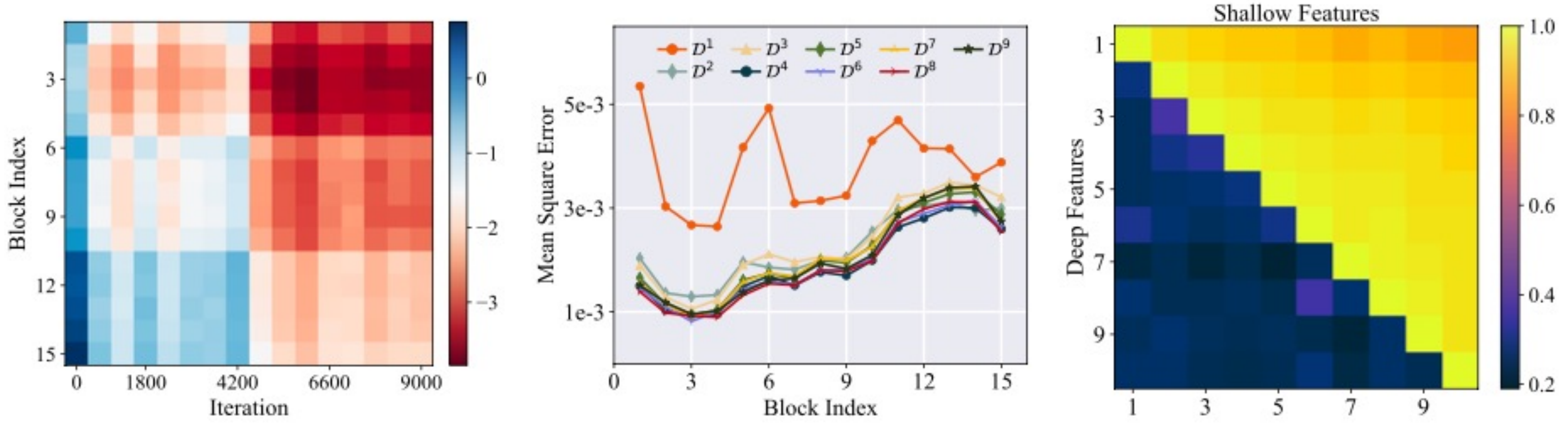
Methods

- MEMO (proposed model)
  - **Shallow layers** tend to learn similar features among all tasks (**generalized features**)
  - **Deep layers** yield very different characteristics from task to task (**specialized features**)
  - Decompose the network structure and only expand the specialized blocks



Methods

- Comparison of shallow and deep layers
  a) The gradient of different residual blocks in a single task when optimizing Eq. 1
  b) MSE per block between the first and last epoch for every incremental stage
  c) CKA (Centered Kernel Alignment) based similarity between the feature maps (10 steps) [3]

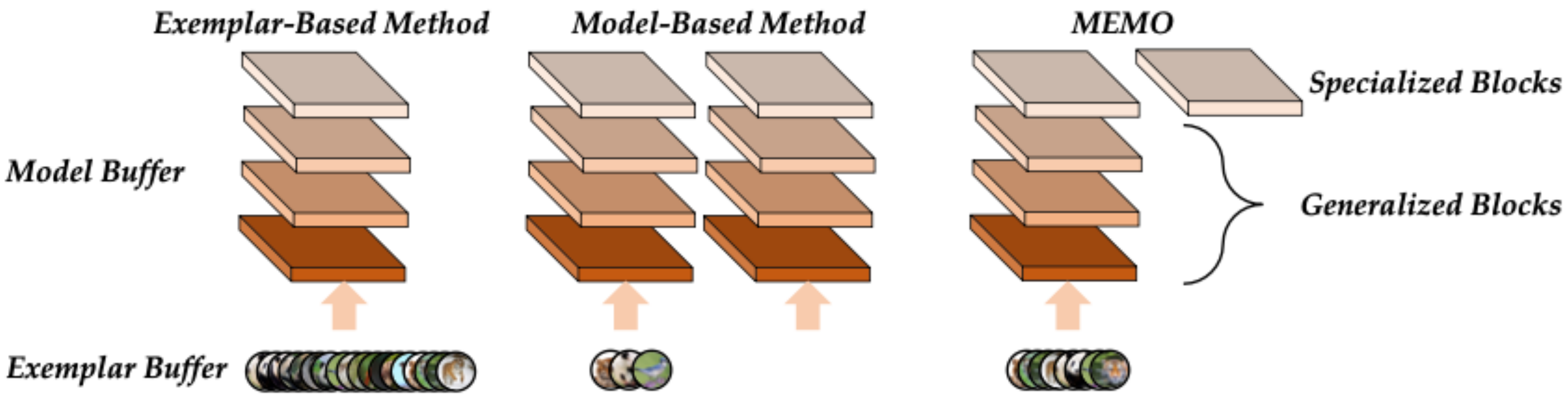


(a) Gradient norm (log scale) (b) Shift of different blocks (c) CKA between backbones

[3] Kornblith, Simon et al. "Similarity of Neural Network Representations Revisited." ArXiv abs/1905.00414 (2019).

Methods

- Memory-efficient expandable model
  - *Exemplar-based methods* train a single model with the most exemplars
  - *Model-based methods* train a new model per new task with the least exemplars
  - **MEMO** trains a new specialized block (strikes a trade-off between the two methods)
  - The first to address the memory-efficient problem in CIL from the model buffer perspective



Methods

- Modification of the model structure
  - Decompose the embedding module into specialized and generalized blocks

$$\phi(\mathbf{x}) = \phi_s(\phi_g(\mathbf{x}))$$

  - Feature aggregation (the modified loss function as Eq. 3)

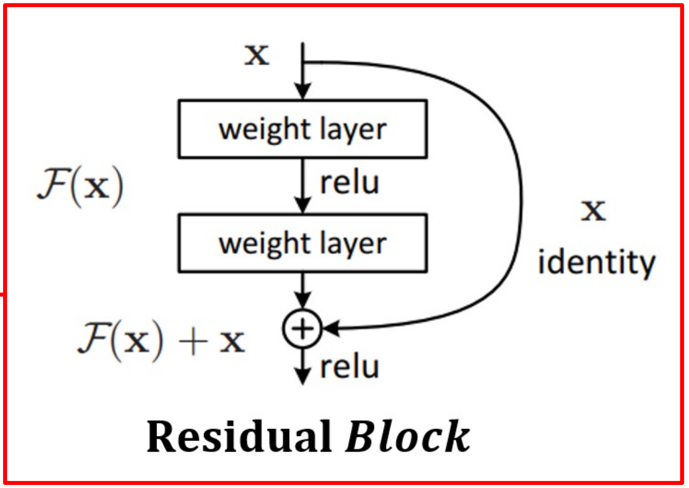
$$\mathcal{L}(\mathbf{x}, y) = \sum_{k=1}^{|\mathcal{Y}_b|} -\mathbb{I}(y = k) \log \mathcal{S}_k(W_{new}^\top [\bar{\phi}_{old}(\mathbf{x}), \phi_{new}(\mathbf{x})]) \,. \tag{2}$$

$$\mathcal{L}(\mathbf{x}, y) = \sum_{k=1}^{|\mathcal{Y}_b|} -\mathbb{I}(y = k) \log \mathcal{S}_k(W_{new}^\top [\phi_{s_{old}}(\phi_g(\mathbf{x})), \phi_{s_{new}}(\phi_g(\mathbf{x}))]) \,. \tag{3}$$

Methods

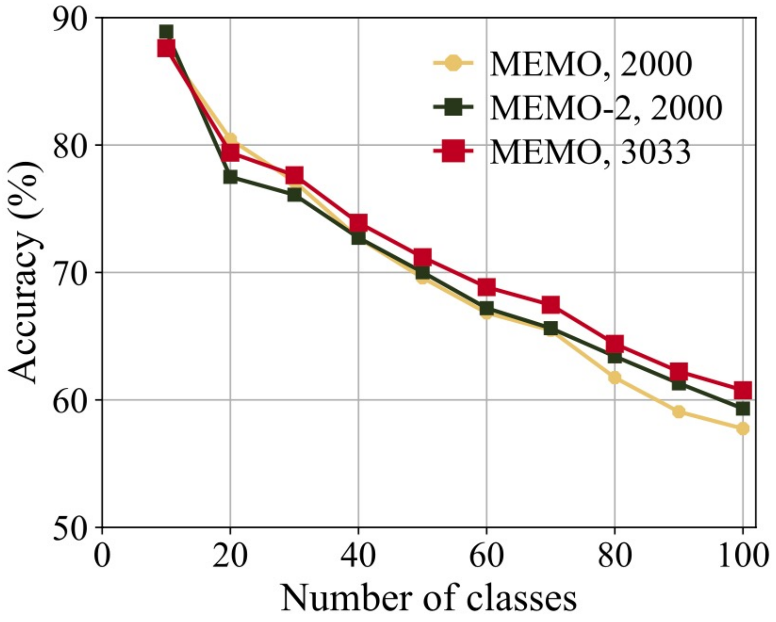
- How to define the specialized and generalized blocks
  - e.g., ResNet32

```
class ResNet(nn.Module):
    def __init__(self, block, num_blocks, num_classes=10):
        super(ResNet, self).__init__()
        self.in_planes = 16

        self.conv1 = nn.Conv2d(3, 16, kernel_size=3, stride=1, padding=1, bias=False)
        self.bn1 = nn.BatchNorm2d(16)
        self.layer1 = self._make_layer(block, 16, num_blocks[0], stride=1)
        self.layer2 = self._make_layer(block, 32, num_blocks[1], stride=2)
        self.layer3 = self._make_layer(block, 64, num_blocks[2], stride=2)
        self.linear = nn.Linear(64, num_classes)

    def resnet32():
        return ResNet(BasicBlock, [5, 5, 5])
```



Methods

- How to define the specialized and generalized blocks
  - 3 groups of residual blocks → treated as the minimal unit when decoupling the network
    1) The last group as the specialized block (MEMO)
    2) The last two groups as specialized blocks (MEMO-2)



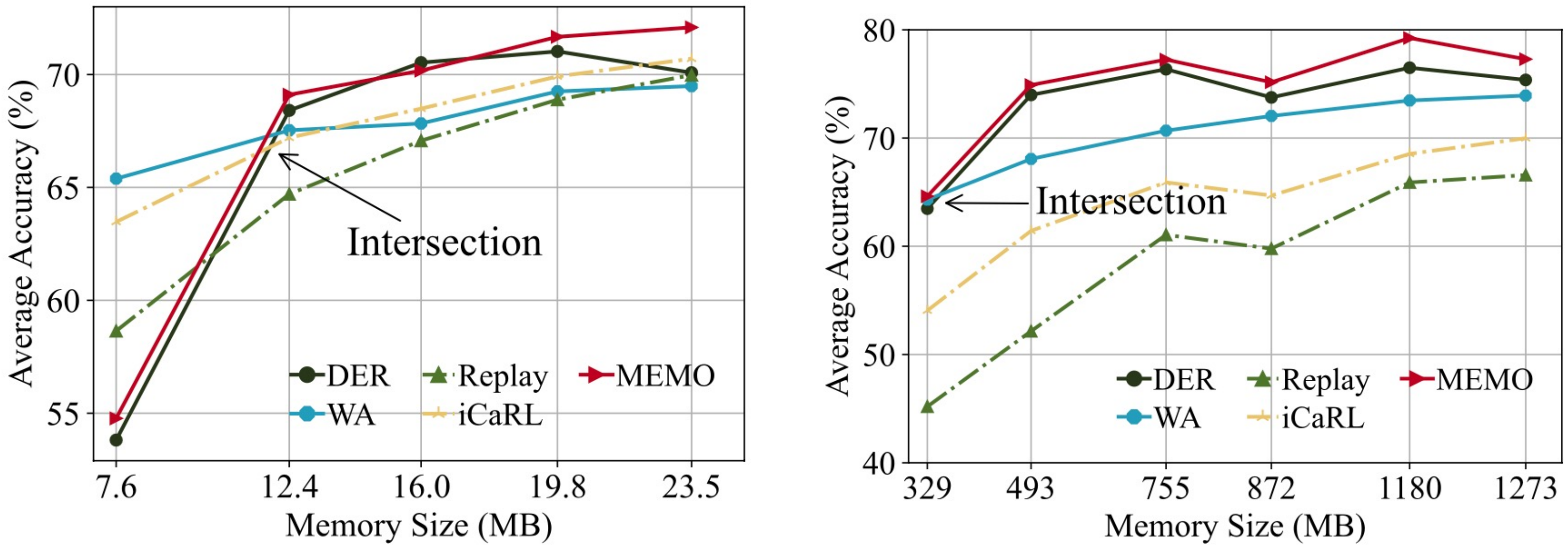
Methods

- How to fairly compare CIL methods
  - Exemplar- and model-based methods consume different memory sizes when being trained
  - These methods should be **aligned to the same memory cost** when comparing the results
- *A model or 603 exemplars*
  - e.g., ResNet32 to CIFAR conversion
  - A ResNet32 model contains 463,504 parameters (float) → 4 bytes per float
  - A CIFAR image consists of (32 x 32) pixels (int) → 3 bytes per integer
  - **A ResNet32 backbone** → (463,504 x 4) ÷ (32 x 32 x 3) ≈ **603 CIFAR instances**

A MODEL OR 603 EXEMPLARS: TOWARDS MEMORY-EFFICIENT CLASS-INCREMENTAL LEARNING

Results

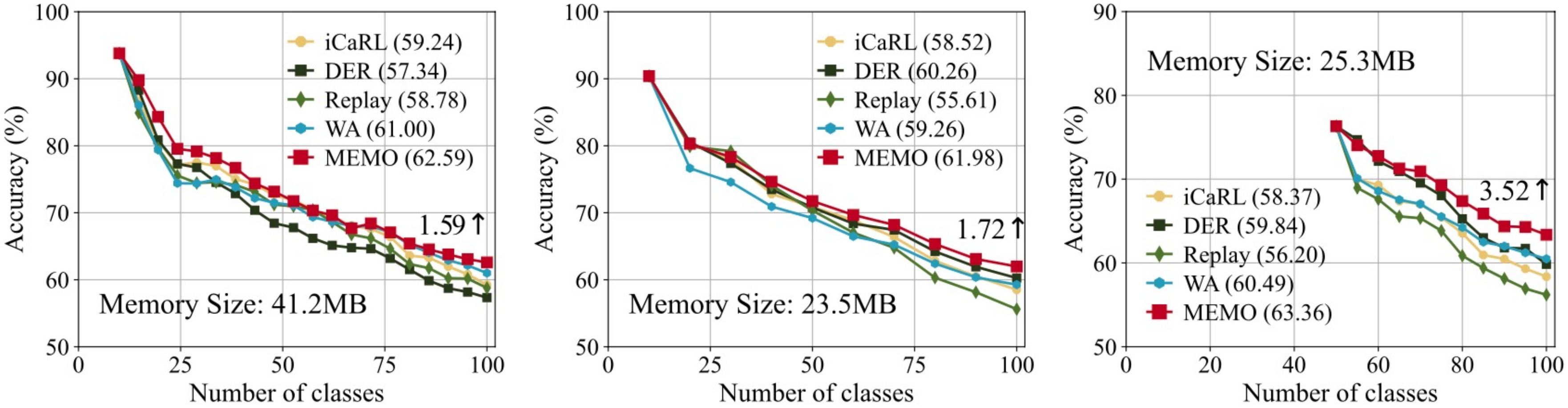
- Whether to use a larger model or more exemplars
  - Saving more exemplars is more effective when the total budget is limited
  - Model expansion is more effective when the total budget is ample



(a) CIFAR100, Base0 Inc10

(b) ImageNet100, Base50 Inc5

Results



(a) CIFAR100 Base0 Inc5 (b) CIFAR100 Base0 Inc10 (c) CIFAR100 Base50 Inc5

Figure. Comparison of the last accuracy under the same memory budget

Method	Accuracy in each session (%) ↑																				Average
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Replay	93.80	84.90	79.67	75.55	74.44	74.57	74.14	73.12	71.22	70.96	70.42	68.63	66.75	66.19	64.63	62.34	61.73	60.24	60.17	58.78	70.61
iCaRL	93.80	86.70	80.67	77.15	77.52	76.97	75.03	74.22	72.62	71.90	70.44	69.12	68.09	67.59	66.39	63.62	63.28	61.98	60.81	59.24	71.85
WA	93.80	86.10	79.40	74.40	74.36	74.97	73.80	72.18	71.49	71.10	69.35	68.65	68.00	67.97	67.08	65.20	64.04	62.89	62.14	61.00	71.39
DER	93.80	88.30	80.87	77.30	76.76	74.53	72.86	70.38	68.44	67.78	66.18	65.13	64.78	64.63	63.19	61.52	59.86	58.73	58.17	57.34	69.52
MEMO	93.80	89.80	84.33	79.55	79.16	78.17	76.71	74.38	73.16	71.74	70.33	69.62	67.68	68.41	67.07	65.41	64.53	63.78	63.07	62.59	73.16

Table. Incremental and average accuracy comparison under CIFAR100 Base0 Inc5 setting

Conclusion

- Firstly…
  - The improvement of DER over other methods is **not** so much as reported in the original paper
  - The improvement of DER than others under the fair comparison is **much less**
- Secondly…
  - MEMO outperforms DER **by a substantial margin** in most cases
  - MEMO is a simple yet effective way to organize CIL models with memory efficiency

Conclusion

- The effect of block sharing
  - Creating specialized features (deep layers) for new tasks is essential
  - Expanding generalized blocks is less efficient than saving more exemplars of equal size

*Given **the same memory budget**, we can further improve the performance*

*by **sharing the generalized blocks** and only **expanding the specialized blocks** for new tasks*

Conclusion

- Fair comparison of different CIL methods
  - Aligned the memory size at the same scale
  - Obtained the state-of-the-art performance *for free* in the fair comparison
- Limitations
  - There are other methods that do not save exemplars or models
  - The research only concentrates on the methods with extra memory

YONSEI UNIVERSITY
COLLEGE OF MEDICINE