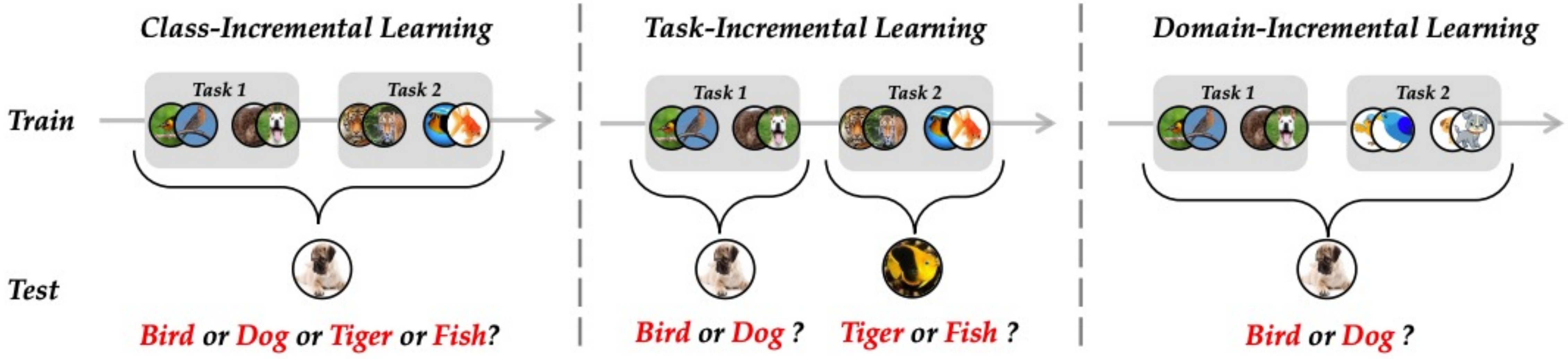
Introduction to Incremental Learning

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

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 (Problem Settings)

- Class overlapping
  - Blurry CIL (class-incremental learning) → **old classes re-emerge in later tasks**
  - Closer to the real-world setting but weakens the learning difficulty
  - Typical CIL setting assumes no overlapping classes for robustness
- Exemplar set (= rehearsal memory)
  - An extra collection of **data from former tasks** for rehearsal
  - The model reviews the exemplar set to resist forgetting of old classes

1) Fix the number of exemplars per class (k) → $k \times n$ total exemplars
2) Fix the size of the exemplar set (k) → k total exemplars, $\frac{k}{n}$ exemplars per class
3) Exemplar-free methods (= non-rehearsal)

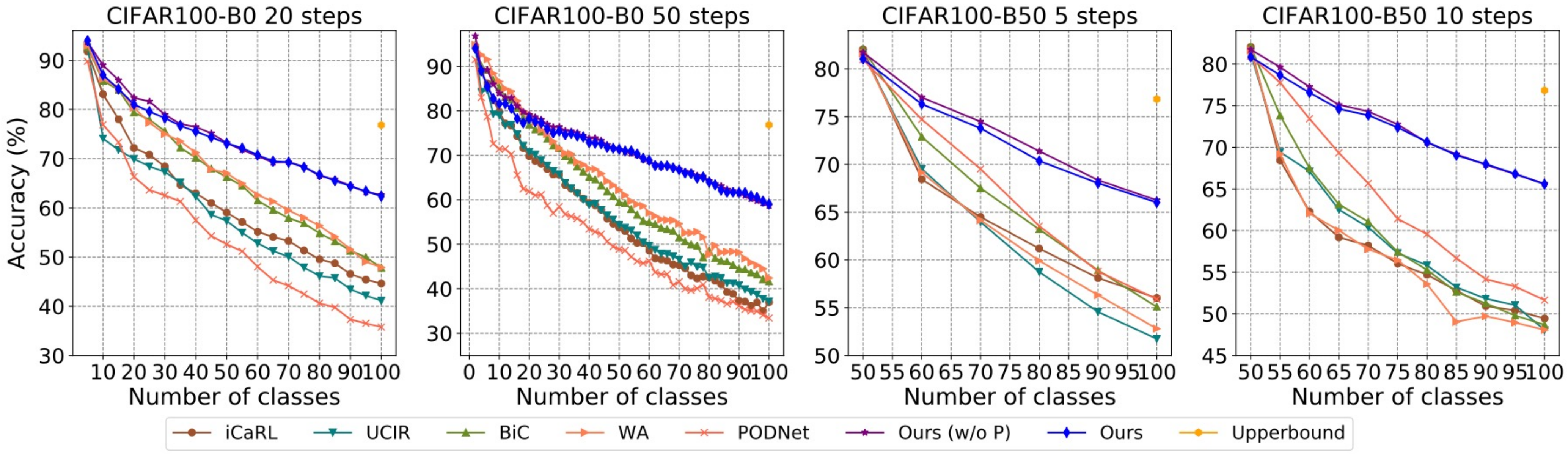
Background (CIL Taxonomy)

- 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).

DER: Dynamically Expandable Representation

DER: Dynamically Expandable Representation for Class Incremental Learning



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

Introduction

- Main idea
  - Settings → **backbone expansion** (dynamic networks) with **rehearsal memory**
  - Freeze the previously learned representation
  - Augment it with additional feature dimensions from a new learnable feature extractor
  - **Super feature** → capable of increasing its dimensionality to accommodate new classes
- Contributions
  - Dynamically expandable representation and a two-stage strategy for CIL
  - **Auxiliary loss** to promote the newly added feature module to learn novel classes effectively
  - **Model pruning step** to learn compact features and remove model redundancy

Methods

- Input process
  - Use **rehearsal memory** to prevent the loss of information from the previous steps
- Feature extraction
  - Construct a feature extractor for each training step (ResNet-50 as backbone)
  - Current features are combined with the previously obtained features → **super feature**
  - Differential channel-level mask to prune the filters in the feature extractor (compact model)
- Classification
  - DER loss = training loss + auxiliary loss + sparsity loss
  - **Auxiliary loss** → classification of old and new classes (previous steps vs. current step)

DER: Dynamically Expandable Representation

for each incremental step $t = 1, \dots, T$ **do**

Feature Extraction:

append(D_t, M_{t-1}) $(M_0 = \emptyset)$

Model Pruning:

$F_t^P \leftarrow$ add_mask(F_t)

$loss_{spr} \leftarrow \frac{\sum_{c=1}^{|c|} \|mask_{c-1}\| \, \|mask_c\|}{\sum_{c=1}^{|c|} ch_{c-1} ch_c}$

Auxiliary Loss:

$p_a \leftarrow \mathrm{Softmax}(H_t^a(F_t^P))$

$loss_{aux} \leftarrow -\sum_{i=1}^{|t|+1} y_a^i \log(p_a^i)$

$y_a = \{1 \dots |t|$ **if** new class **else** $0\}$

$\Phi_t^P \leftarrow \mathrm{concatenate}([\Phi_{t-1}^p, F_t^P])$ $(\Phi_1^P = F_1^P)$

Classification:

generate FC_t

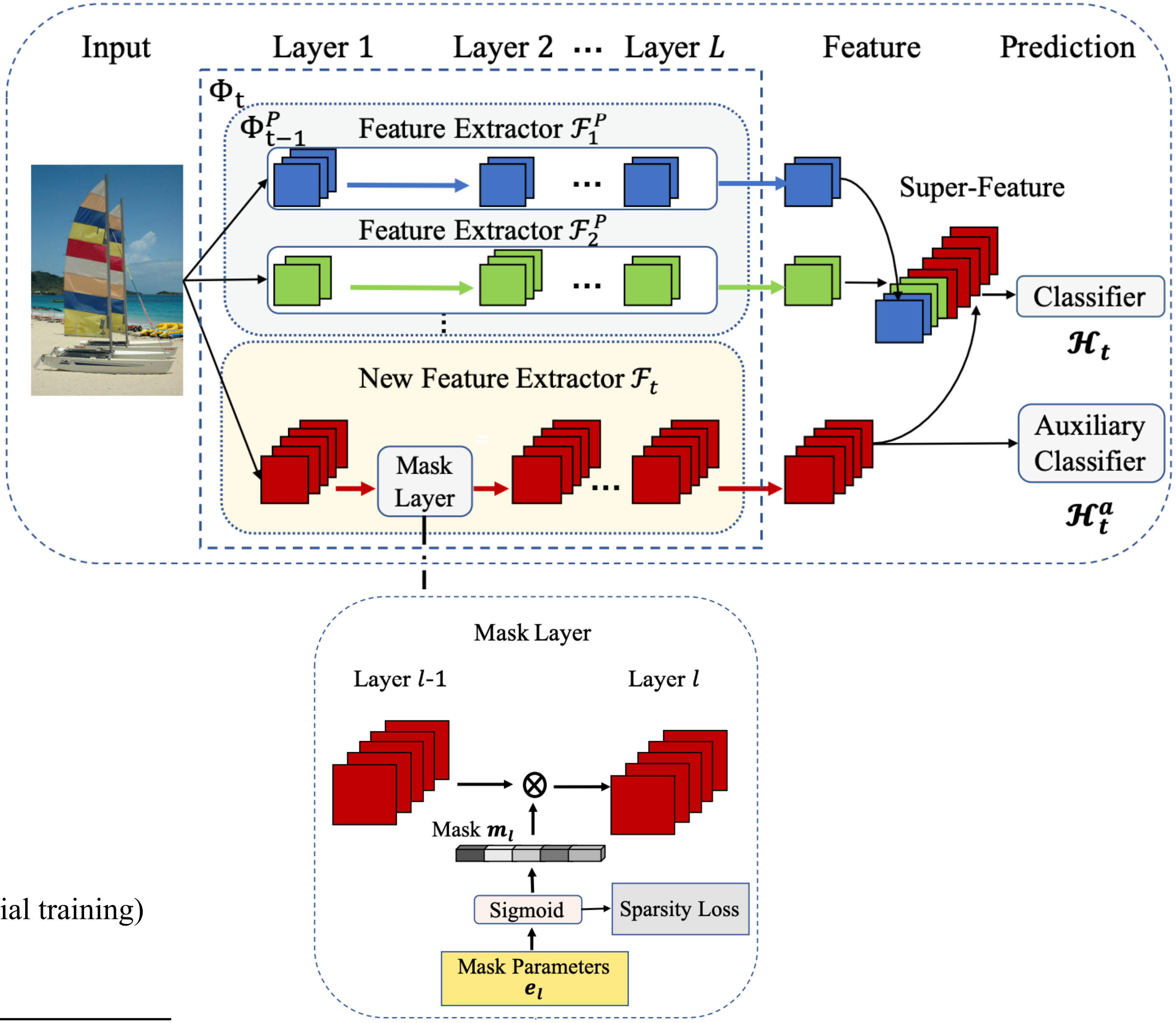
$p_t \leftarrow \mathrm{Softmax}(H_t(\Phi_t^P))$

$P_t \leftarrow \mathrm{argmax}(p_t)$

$loss_{clf} \leftarrow -\sum_{j=1}^{N} y_t^j \log(p_t^j)$

$loss_{total} \leftarrow loss_{clf} + \lambda_a loss_{aux} + \lambda_s loss_{spr}$ ($\lambda_a = 0$ in initial training)

$M_t \leftarrow$ construct_rehearsal_exemplar(m)



YONSEI UNIVERSITY
COLLEGE OF MEDICINE