电子科技大学
University of Electronic Science and Technology of China

Recent Advances of Continual Learning

Wei Han

Data Mining Lab,
Big Data Research Center, UESTC
Email： weihan@std.uestc.edu.cn

Outline

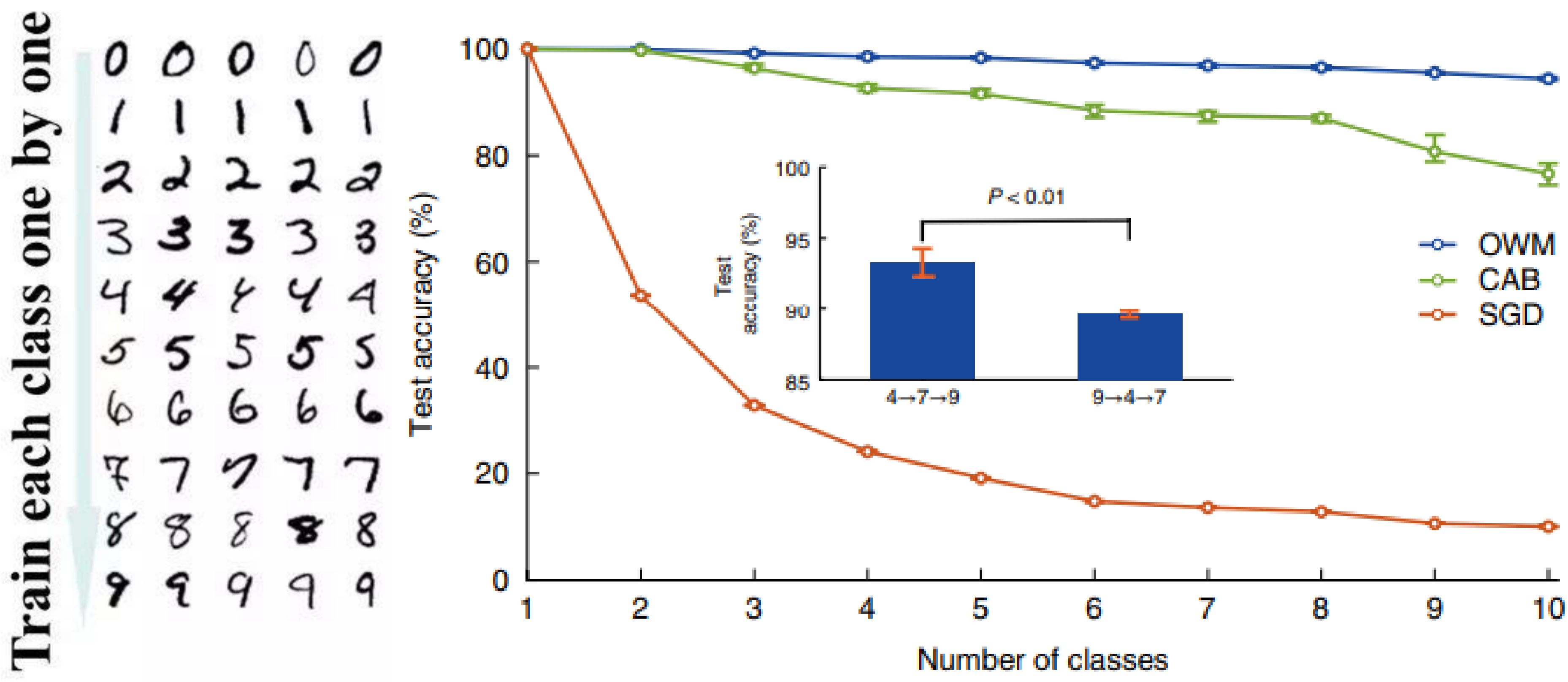
数据挖掘实验室
Data Mining Lab

- Preliminary: Taxonomy of Continual Learning
- Supervised Continual Learning
- Unsupervised Continual Learning
- Our Proposal

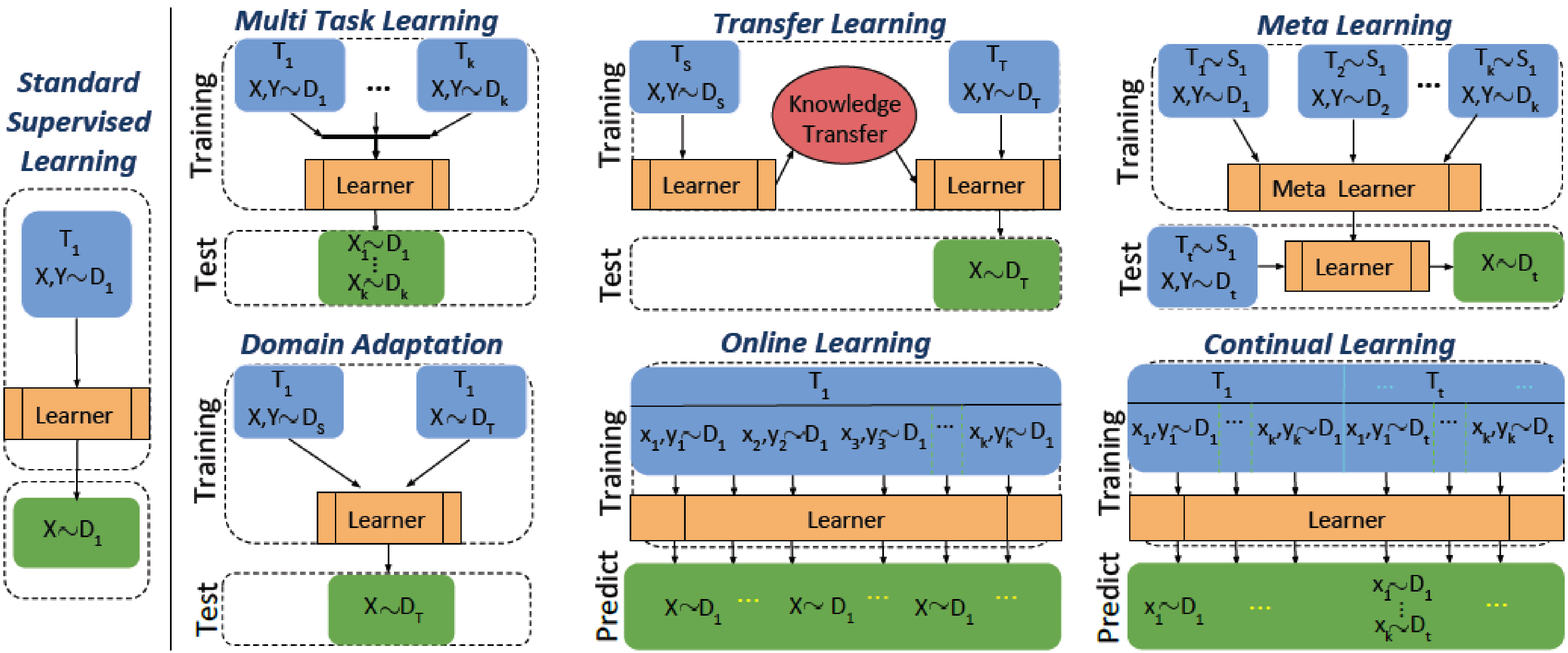
Preliminary:

Taxonomy of Continual Learning

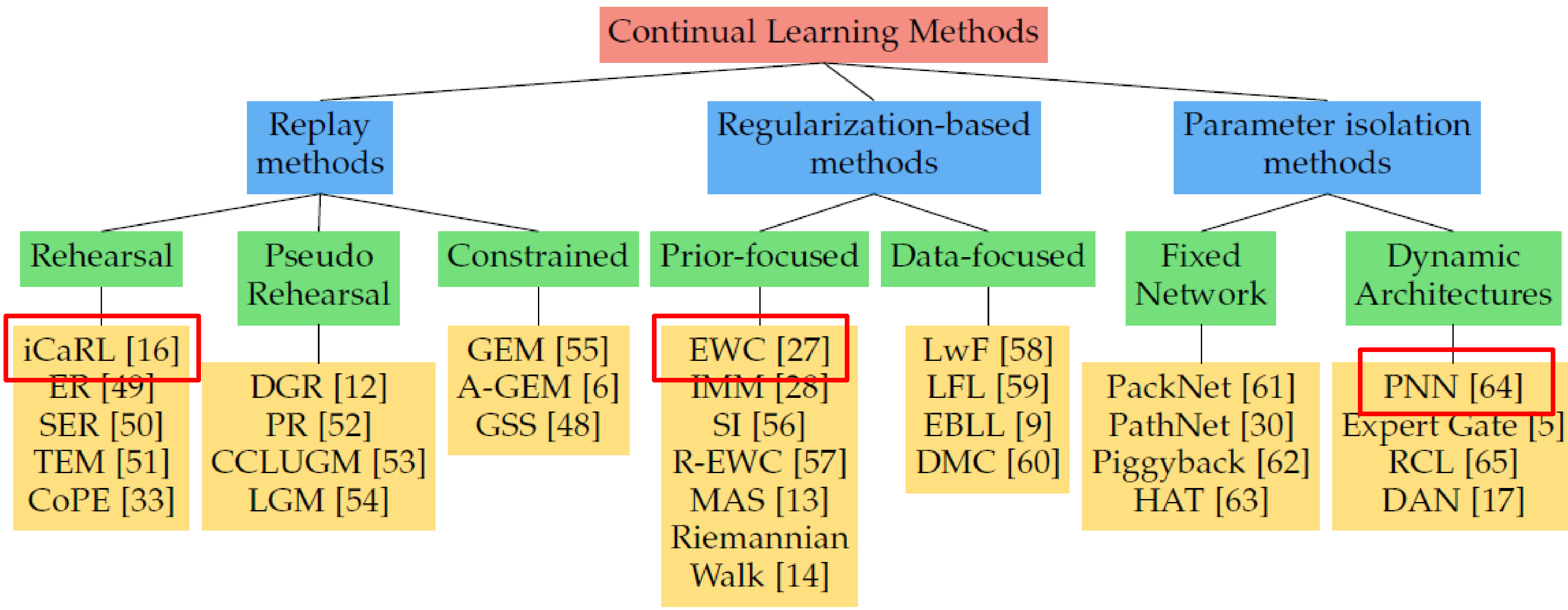
■ Continual Learning



■ Continual Learning



Taxonomy of Continual Learning



■ iCaRL (Replay-based)

Algorithm 2 iCaRL IncrementalTrain

input $X^s, \ldots, X^t$ // training examples in per-class sets
input K // memory size
require Θ // current model parameters
require $\boxed{\mathcal{P}} = (P_1, \ldots, P_{s-1})$ // current exemplar sets

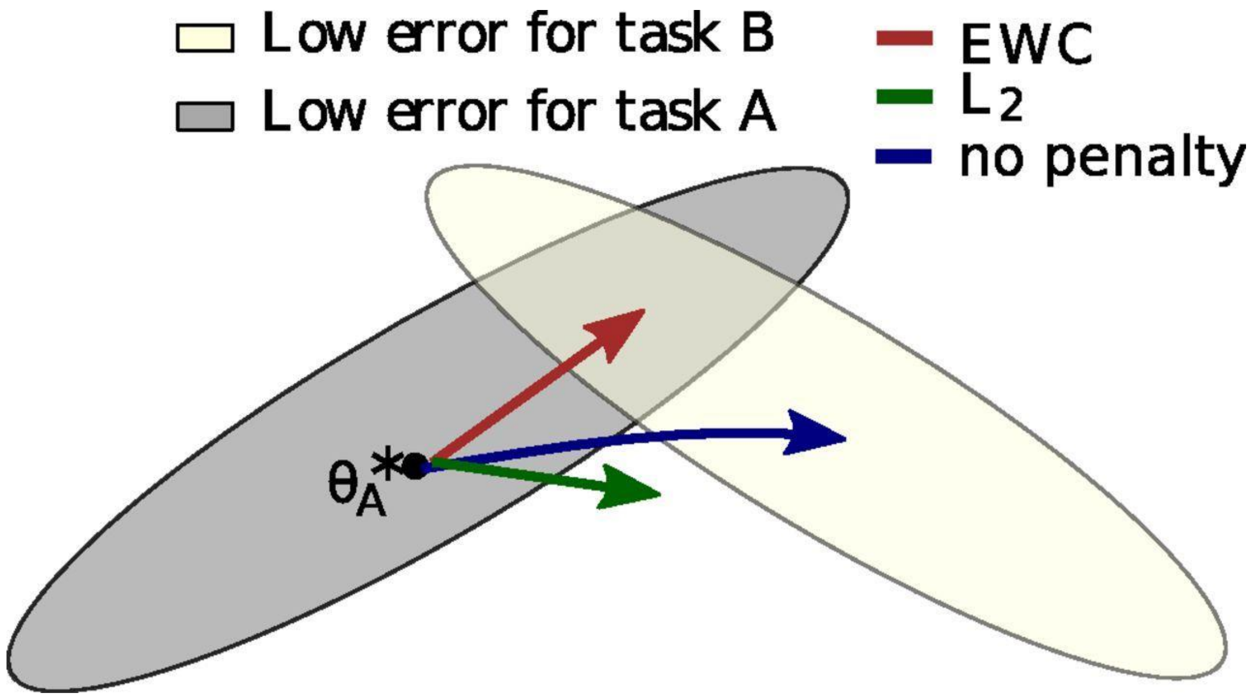
$\Theta \leftarrow \text{UpdateRepresentation}(X^s, \ldots, X^t; \mathcal{P}, \Theta)$

$m \leftarrow K/t$ // number of exemplars per class

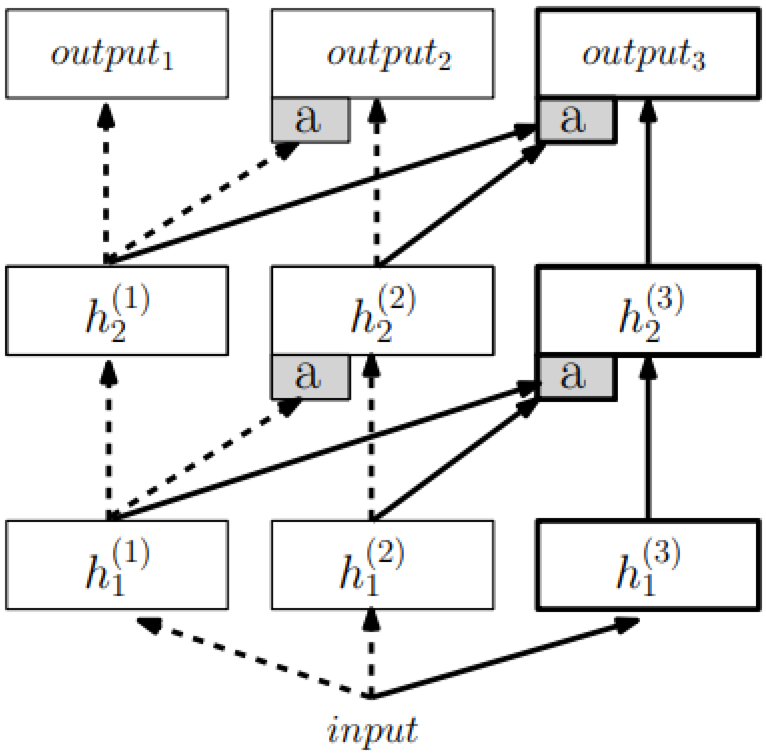
for $y = 1, \ldots, \underline{s-1}$ **do**
  $P_y \leftarrow \text{ReduceExemplarSet}(P_y, m)$
end for
for $y = \underline{s, \ldots, t}$ **do**
  $P_y \leftarrow \text{ConstructExemplarSet}(X_y, m, \Theta)$
end for

$\mathcal{P} \leftarrow (P_1, \ldots, P_t)$ // new exemplar sets

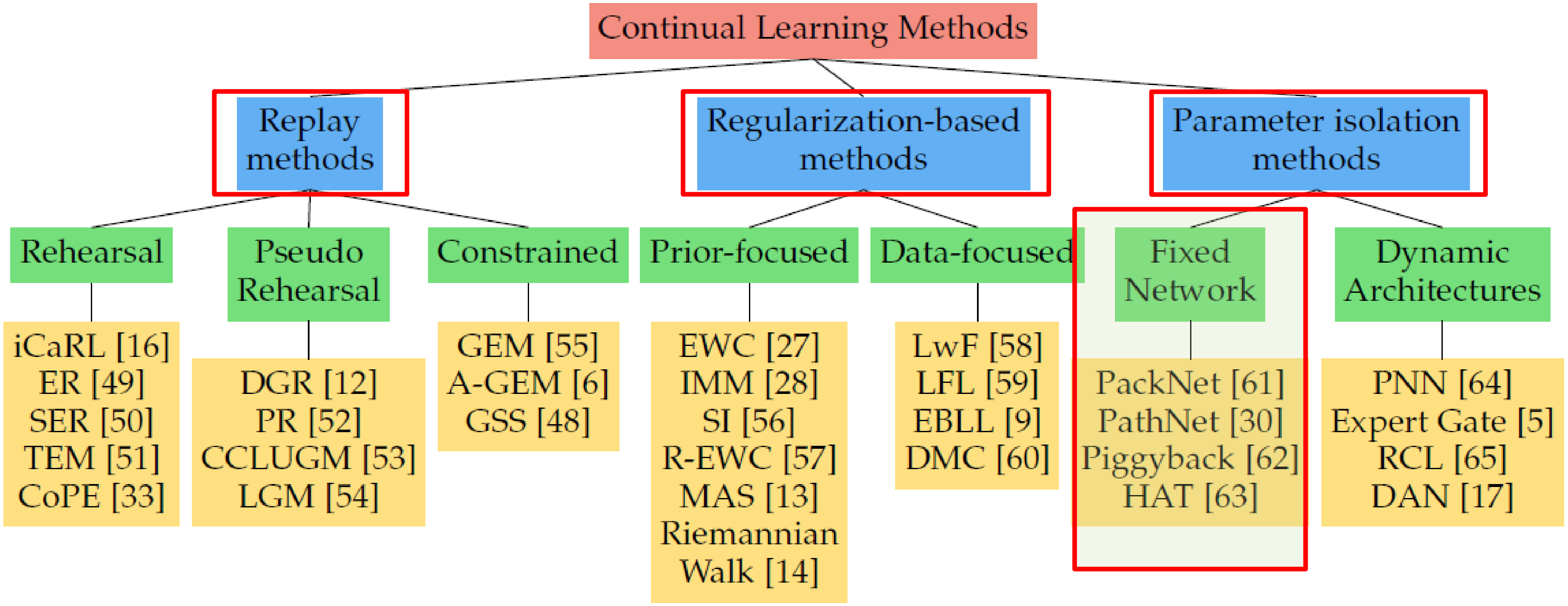
- EWC (Regularization-based)



- PNN (Parameter Isolation)

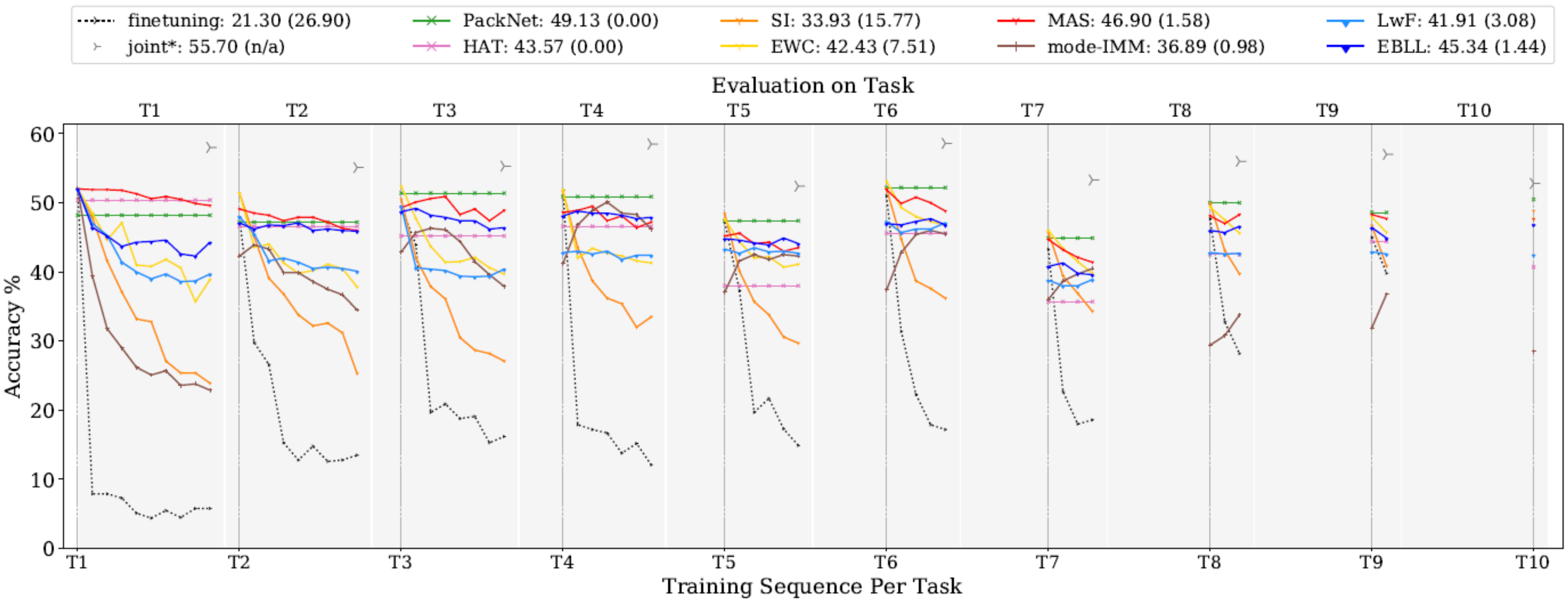


■ Taxonomy of Continual Learning

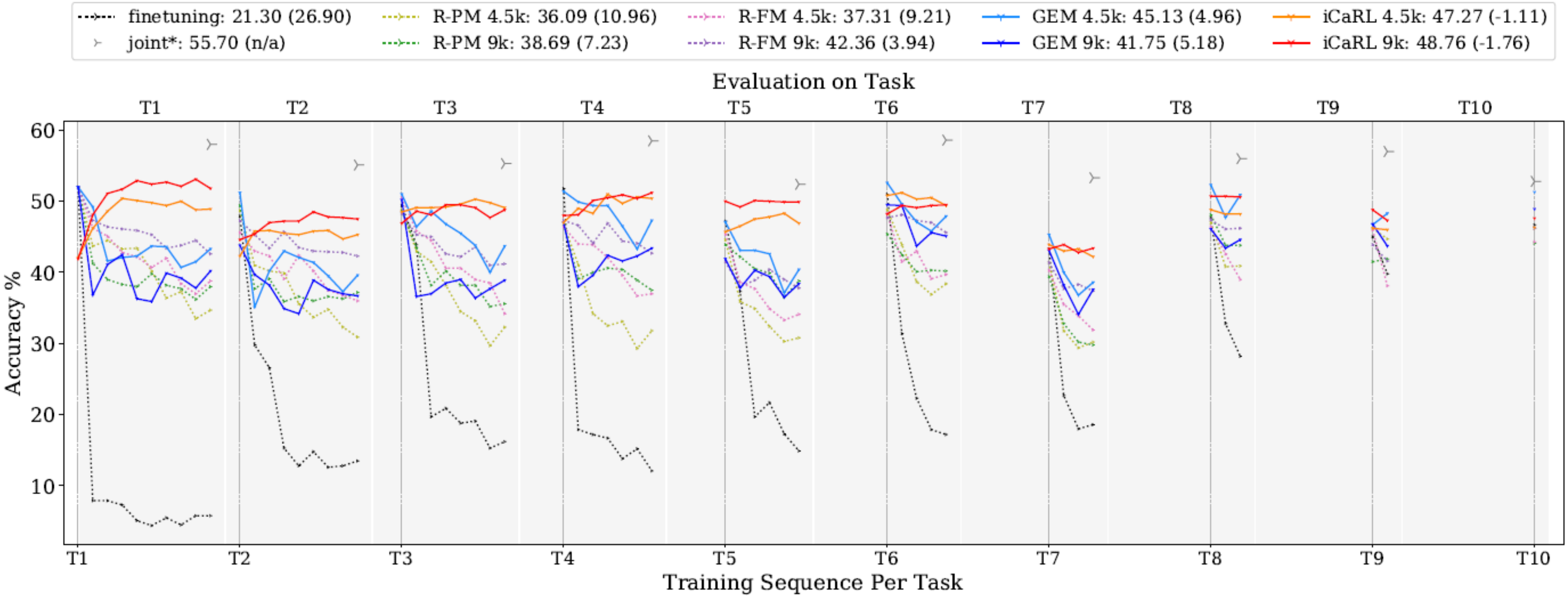


数据挖掘实验室
Data Mining Lab

■ Results of Continual Learning



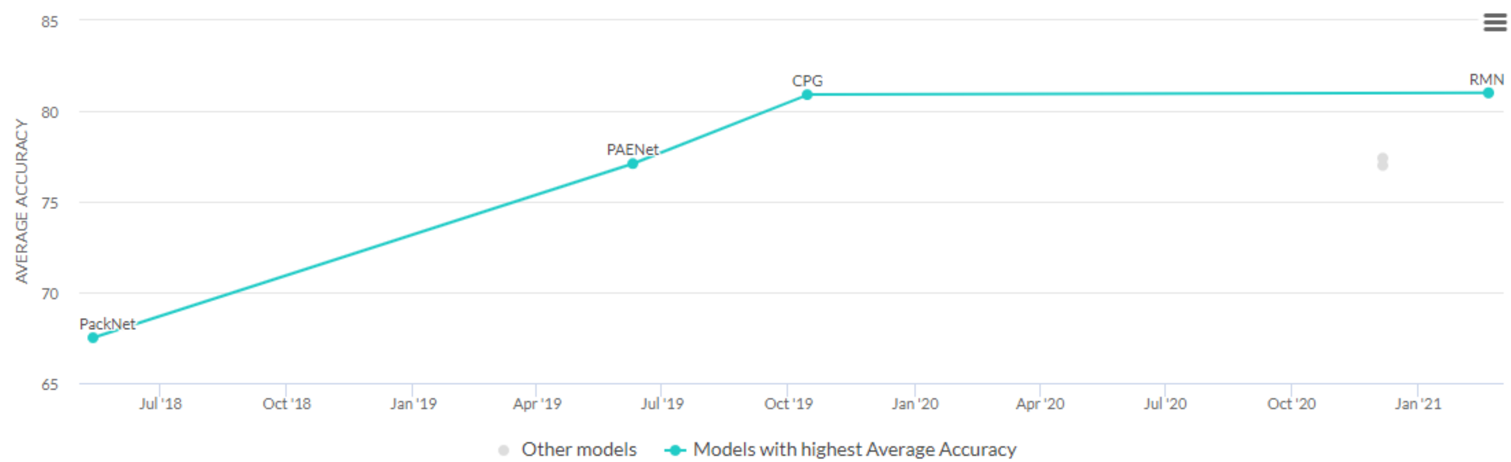
■ Results of Continual Learning



Continual Learning on Cifar100 (20 tasks)

Leaderboard Dataset

View Average Accuracy by Date



Filter: Strict Continual Learning untagged

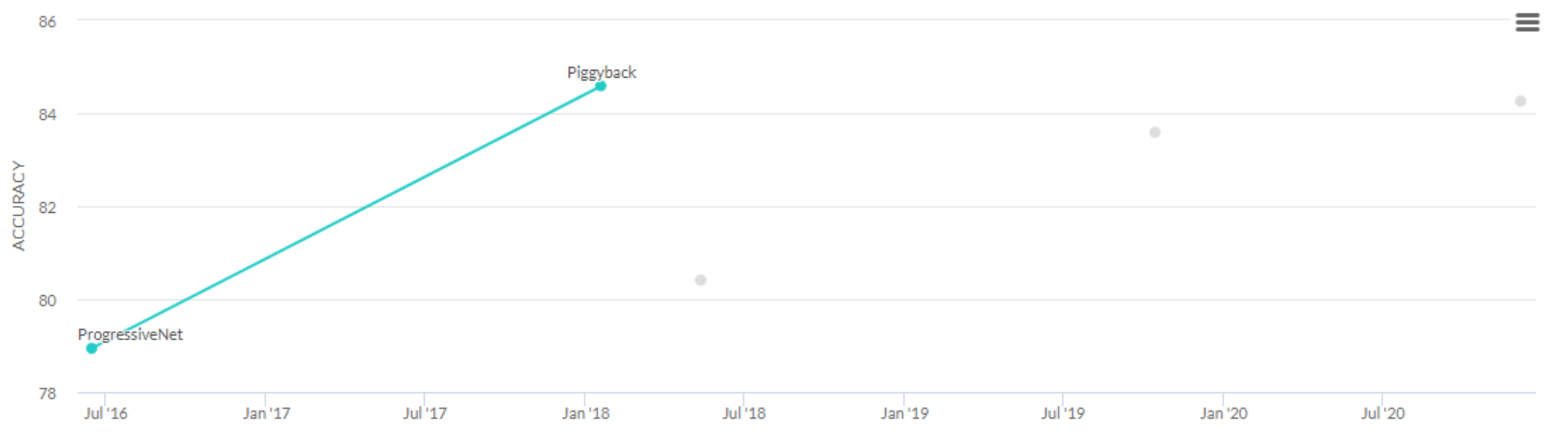
Edit Leaderboard

Rank	Model	Average Accuracy	Paper	Code	Result	Year	Tags
1	RMN	81	Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping			2021	Strict Continual Learning
2	CPG	80.9	Compacting, Picking and Growing for Unforgetting Continual Learning			2019	
3	CondConvContinual	77.4	EXTENDING CONDITIONAL CONVOLUTION STRUCTURES FOR ENHANCING MULTITASKING CONTINUAL LEARNING			2020	

Continual Learning on CUBS (Fine-grained 6 Tasks)

Leaderboard Dataset

View Accuracy by Date



Filter: untagged

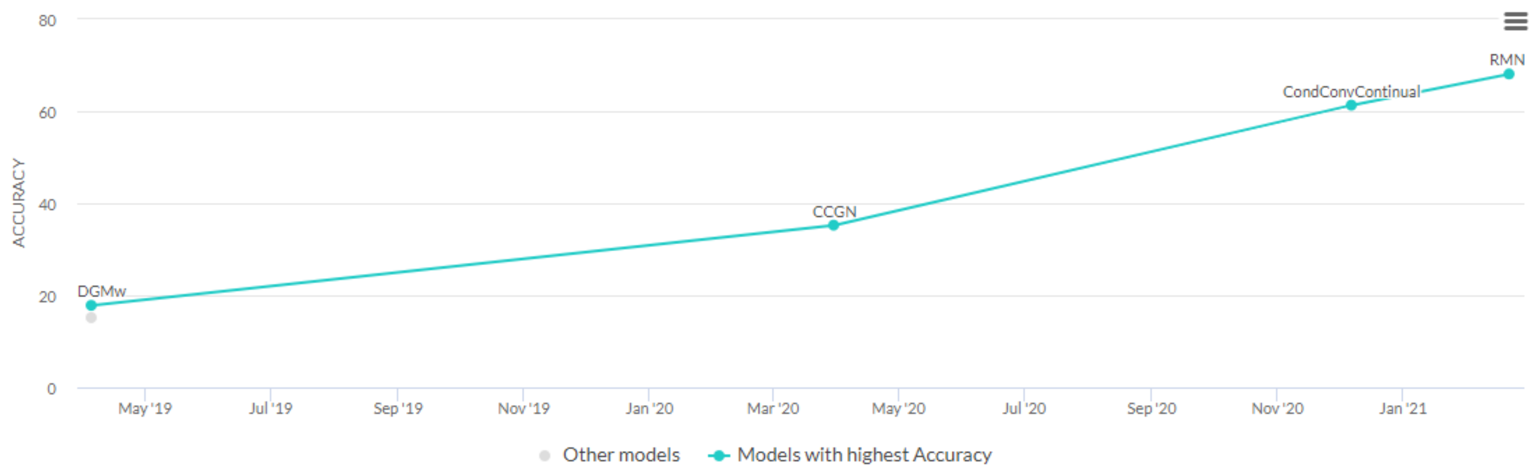
Edit Leaderboard

Rank	Model	Accuracy	Pretrained	Paper	Code	Result	Year	Tags
1	**Piggyback**	84.59	Yes	Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights			2018	
2	**CondConvContinual**	84.26	Yes	EXTENDING CONDITIONAL CONVOLUTION STRUCTURES FOR ENHANCING MULTITASKING CONTINUAL LEARNING			2020	
3	**CPG**	83.59	Yes	Compacting, Picking and Growing for Unforgetting Continual Learning			2019	

Continual Learning on ImageNet-50 (5 tasks)

Leaderboard Dataset

View Accuracy by Date



Filter: untagged

Edit Leaderboard

Rank	Model	Accuracy	Paper	Code	Result	Year	Tags
1	**RMN**	68.1	Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping			2021	
2	**CondConvContinual**	61.32	EXTENDING CONDITIONAL CONVOLUTION STRUCTURES FOR ENHANCING MULTITASKING CONTINUAL LEARNING			2020	
3	**CCGN**	35.24	Conditional Channel Gated Networks for Task-Aware Continual Learning			2020	

Preliminary

数据挖掘实验室
Data Mining Lab

Oxford 102 Flowers					
Method	Task 1	Task 2	Task 3	Task 4	Avg.
Finetuning	10.0 (-20.3)	5.1 (-17.1)	6.7 (-13.6)	17.3 (0.0)	9.8
Freezing	30.3 (0.0)	39.8 (0.0)	32.0 (0.0)	33.1 (0.0)	33.8
Joint	54.6 (+24.3)	58.9 (+11.5)	57.7 (+4.5)	47.0 (0.0)	54.6
EWC [14]	12.1 (-18.2)	11.6 (-38.1)	9.3 (-24.4)	25.8 (0.0)	14.7
HAT [41]	17.2 (-12.7)	19.3 (-28.5)	28.6 (+1.4)	31.6 (0.0)	24.2
PackNet [26]	32.0 (0.0)	53.7 (0.0)	43.6 (0.0)	37.9 (0.0)	41.8
TFM w/o FN	36.4 (0.0)	54.1 (0.0)	38.6 (0.0)	39.0 (0.0)	42.0
TFM	36.4 (0.0)	53.8 (0.0)	45.5 (0.0)	37.6 (0.0)	**43.3**

CUBS 200 Birds					
Method	Task 1	Task 2	Task 3	Task 4	Avg.
Finetuning	7.4 (-30.2)	2.6 (-30.0)	29.7 (-3.4)	43.1 (0.0)	20.7
Freezing	37.6 (0.0)	35.1 (0.0)	35.4 (0.0)	38.4 (0.0)	36.6
Joint	48.7 (+11.1)	52.1 (+6.0)	50.7 (+1.5)	51.9 (0.0)	50.8
EWC [14]	16.2 (-21.4)	19.0 (-21.2)	24.2 (-14.0)	41.7 (0.0)	25.3
HAT [41]	18.7 (-1.8)	19.4 (-0.4)	28.5 (-0.6)	31.2 (0.0)	24.4
PackNet [26]	35.3 (0.0)	42.8 (0.0)	44.4 (0.0)	45.9 (0.0)	42.1
TFM w/o FN	42.9 (0.0)	44.1 (0.0)	48.3 (0.0)	49.1 (0.0)	46.1
TFM	42.9 (0.0)	43.1 (0.0)	49.9 (0.0)	48.8 (0.0)	**46.2**

Stanford 40 Actions					
Method	Task 1	Task 2	Task 3	Task 4	Avg.
Finetuning	24.4 (-10.5)	26.5 (-7.7)	17.6 (-16.8)	28.9 (0.0)	24.4
Freezing	34.9 (0.0)	29.4 (0.0)	30.1 (0.0)	30.5 (0.0)	31.2
Joint	45.7 (+10.8)	40.3 (+4.8)	43.2 (-1.1)	40.2 (0.0)	42.4
EWC [14]	24.2 (-10.7)	28.2 (-2.0)	25.2 (-5.6)	34.3 (0.0)	28.0
HAT [41]	25.7 (-1.0)	25.5 (-2.7)	30.1 (-2.1)	34.4 (0.0)	28.9
PackNet [26]	32.5 (0.0)	32.9 (0.0)	36.7 (0.0)	34.3 (0.0)	34.1
TFM w/o FN	35.3 (0.0)	38.3 (0.0)	39.2 (0.0)	38.0 (0.0)	37.7
TFM	35.3 (0.0)	37.2 (0.0)	42.0 (0.0)	37.2 (0.0)	**38.0**

Supervised Continual Learning

PiggyBack

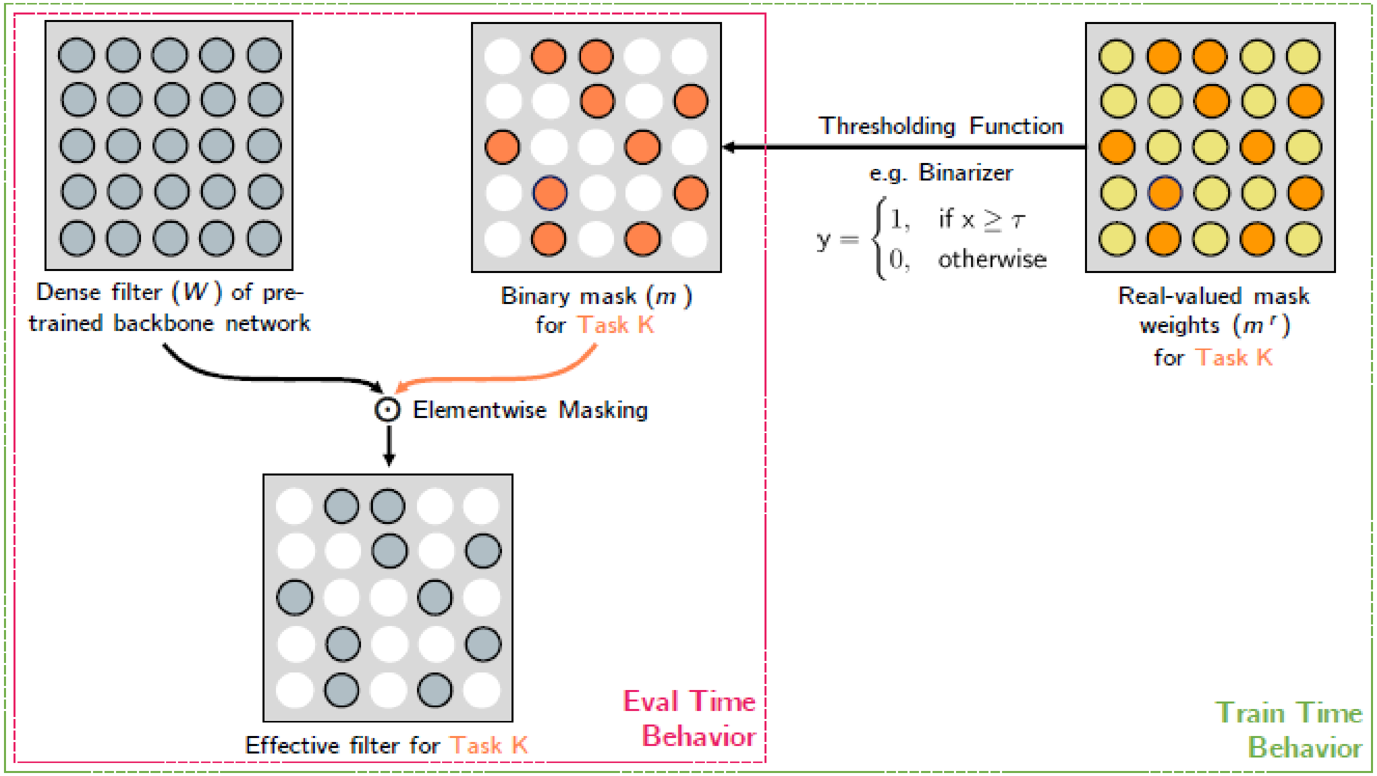


Fig. 1: Overview of our method for learning piggyback masks for fixed backbone networks. During training, we maintain a set of real-valued weights m^r which are passed through a thresholding function to obtain binary-valued masks m. These masks are applied to the weights W of the backbone network in an elementwise fashion, keeping individual weights active, or masked out. The gradients obtained through backpropagation of the task-specific loss are used to update the real-valued mask weights. After training, the real-valued mask weights are discarded and only the thresholded mask is retained, giving one network mask per task.

■ Learnable hard attention

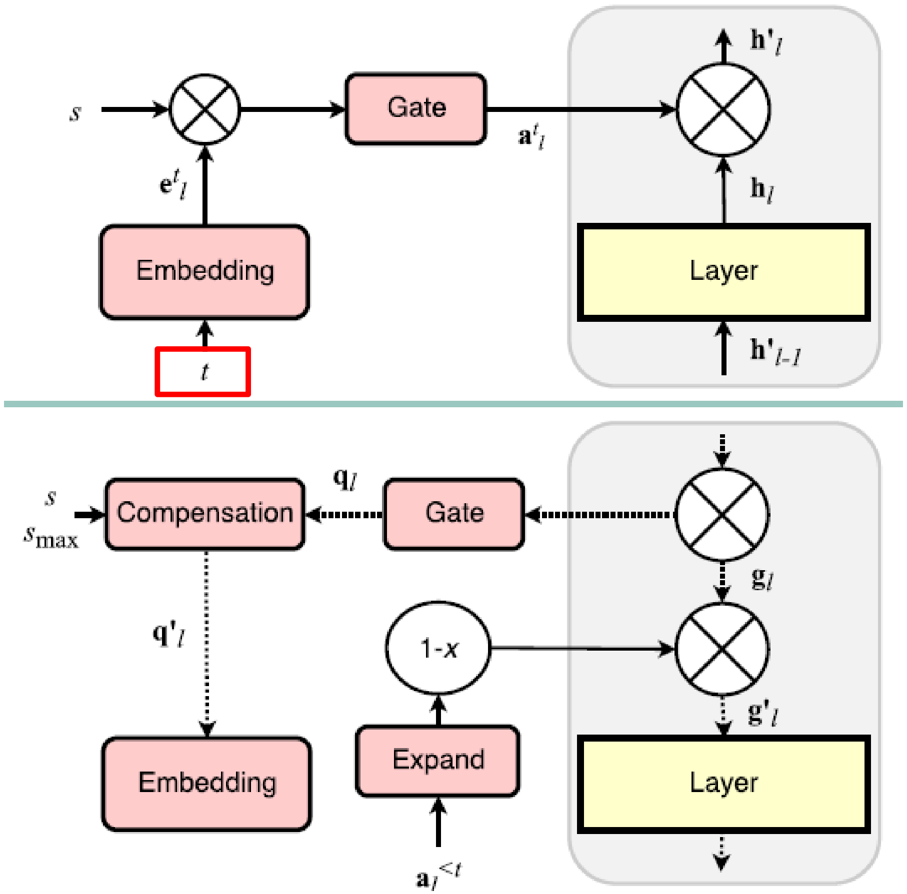
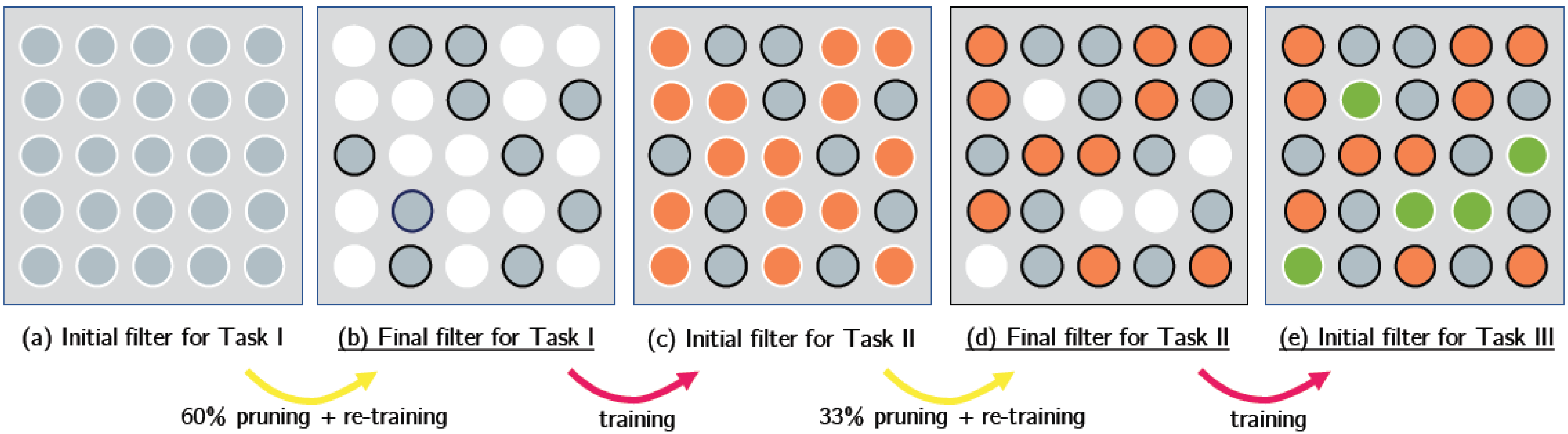


Figure 1. Schematic diagram of the proposed approach: forward (top) and backward (bottom) passes.

$$s = \frac{1}{s_{\max}} + \left(s_{\max} - \frac{1}{s_{\max}}\right)\frac{b-1}{B-1},$$

■ Pruning as Masks



Compacting, Picking and Growing

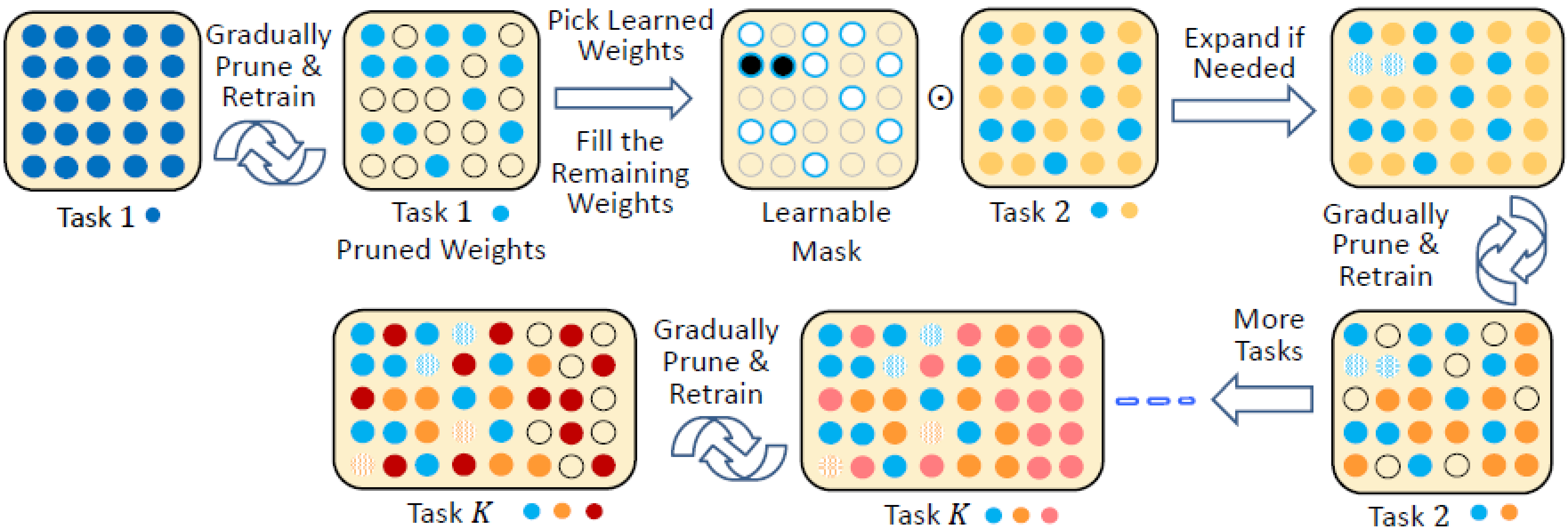


Figure 1: Compacting, Picking, and Growing (CPG) continual learning. Given a well-trained model, gradual pruning is applied to compact the model to release redundant weights. The compact model weights are kept to avoid forgetting. Then a learnable binary weight-picking mask is trained along with previously released space for new tasks to effectively reuse the knowledge of previous tasks. The model can be expanded for new tasks if it does not meet the performance goal. Best viewed in color.

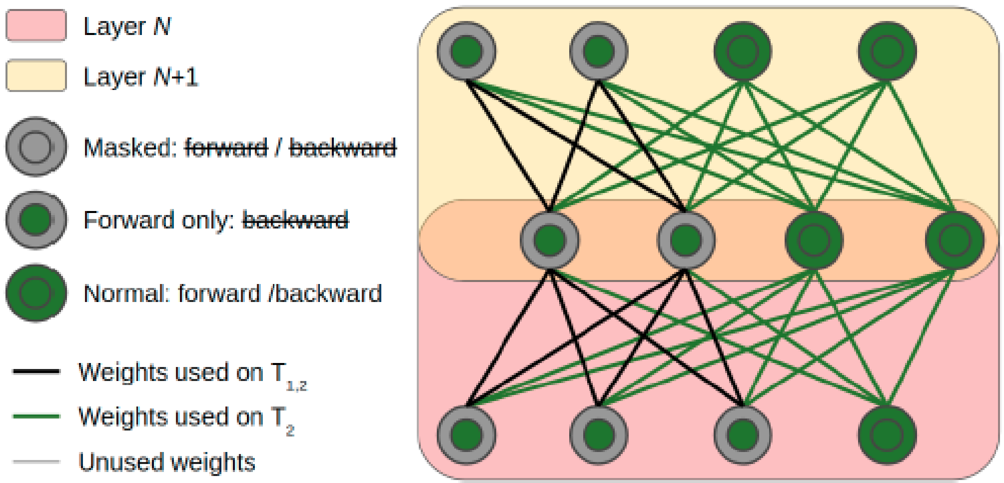
Ternary Feature Map

■ Ternary state: ‘*used*’, ‘*learnable*’, ‘*unused*’

- Binary mask

$$y = (Wx) \odot m^{t,l}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}} = (m_i^{t,l} \wedge m_j^{t,l-1}) x_j \frac{\partial \mathcal{L}}{\partial y_i}$$



- Ternary mask

$$y = (Wx) \odot n^{t,l} \qquad n_i^{t,l} = \begin{cases} 1, & \text{if } \exists\, s \leq t \ : m_i^{s,l} = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}} = (n_j^{t,l-1} n_i^{t,l} - n_j^{t-1,l-1} n_i^{t-1,l}) x_j \frac{\partial \mathcal{L}}{\partial y_i}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}} = (m_i^{t,l} \vee m_j^{t,l-1}) x_j \frac{\partial \mathcal{L}}{\partial y_i}$$

Ternary Feature Map

- Task-specific feature normalization

$$\hat{x}_{l,i} = \gamma_{t,l,i}\, x_{l,i} + \beta_{t,l,i}$$

- Growing Ternary Feature Map

$$n_j^{t,l} = \begin{cases} 1, & \text{for current task, if } 1 \leq j \leq I{+}N \\ 0, & \text{for previous tasks, if } I < j \leq I{+}N \end{cases}$$

$$m_j^{t,l} = \begin{cases} 0, & \text{for current task, if } 1 \leq j \leq I \\ 1, & \text{for current task, if } I < j \leq I{+}N \\ 0, & \text{for previous tasks, if } I < j \leq I{+}N \end{cases}$$

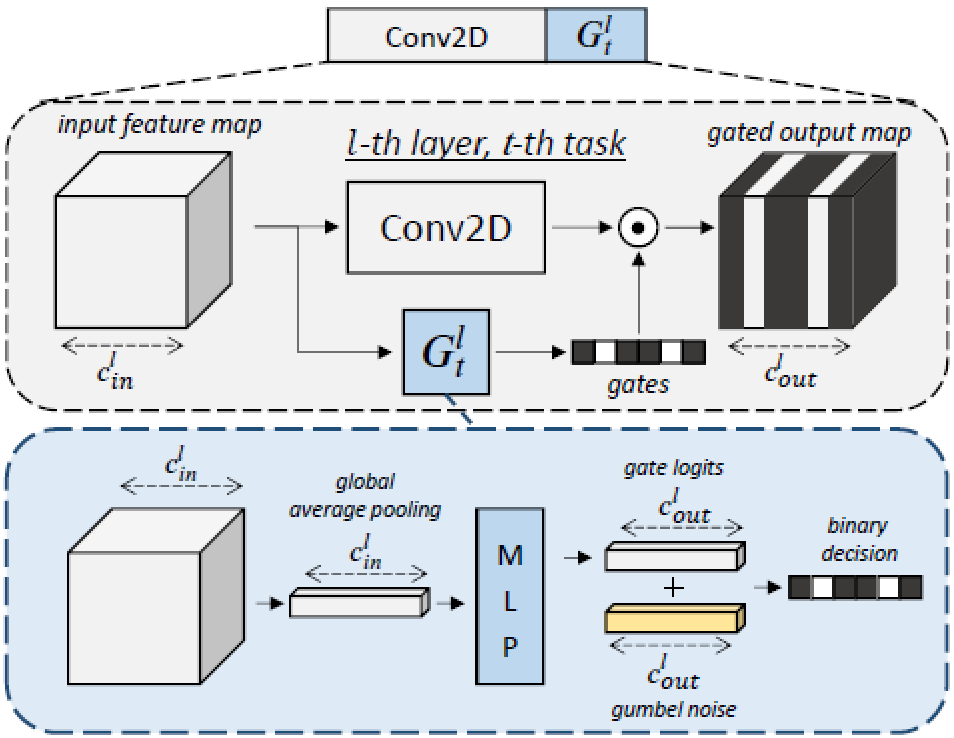
Summary and Discussion

- Differentiable Mask?
- Knowledge Reuse?
- Expandable?
- Computational Complexity (upon *weights*/*features*)

Unsupervised Continual Learning

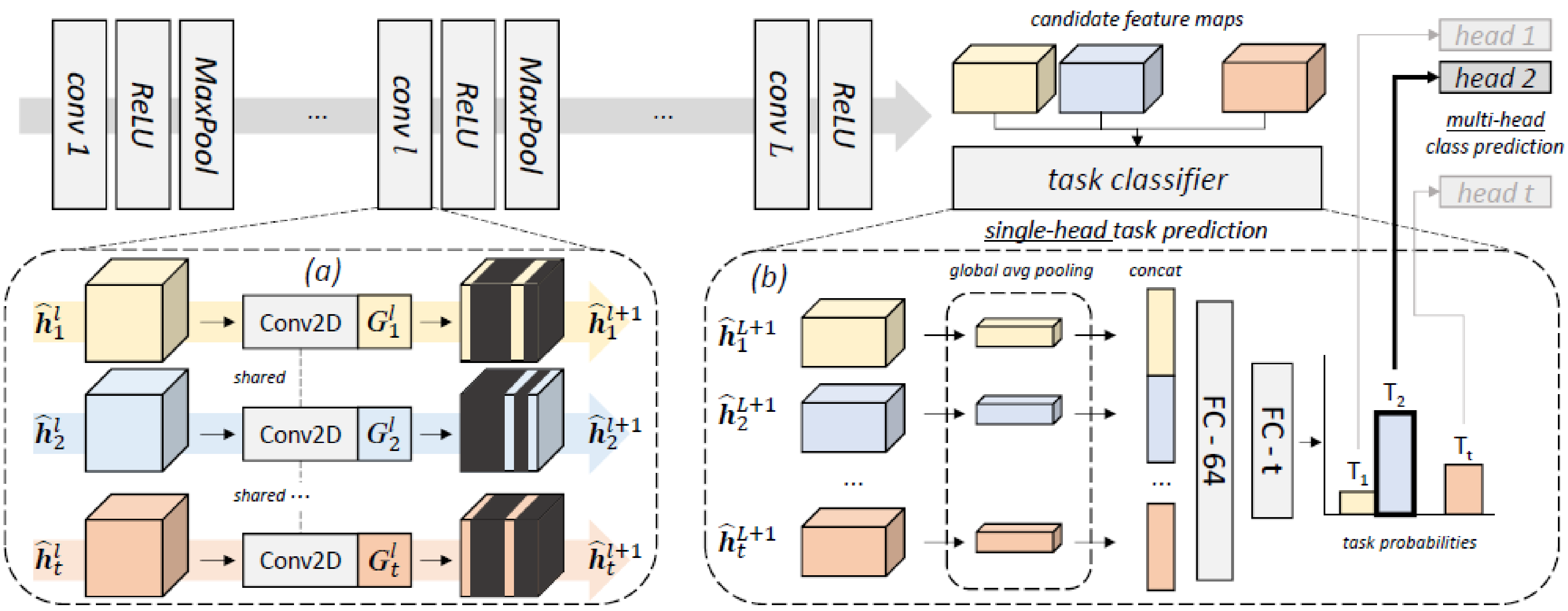
Conditional Channel Gated Networks

■ Conditional Channel Gate



Conditional Channel Gated Networks

■ Task Classifier



■ Conditional Convolution

$$\mathrm{CondConv}_n(\mathbf{x}) = \sigma((\alpha_1 \mathbf{W}_1 + \alpha_2 \mathbf{W}_2 + \cdots + \alpha_n \mathbf{W}_n) * \mathbf{x}),$$

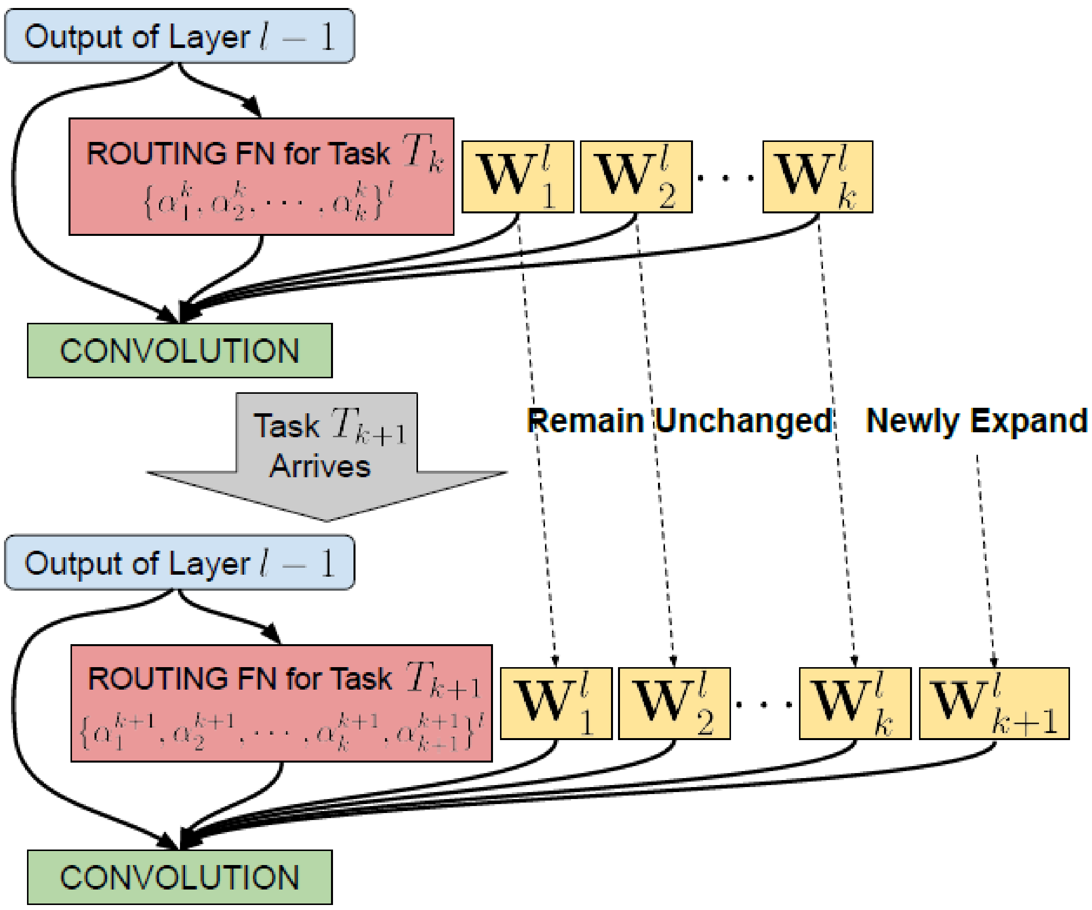
$$\alpha_i = \mathrm{Sigmoid}(\mathrm{GlobalAveragePool}(\mathbf{x})\mathbf{R}_i + \mathbf{b}_i),$$

$$M_{k+1}(\mathbf{x}) = \underline{C_{k+1}}(\mathcal{B}(\mathbf{x}; \underline{\mathcal{W}_{1:(k+1)}}, \underline{\mathcal{A}_{k+1}})),$$

$$\underline{\operatorname*{arg\,min}_{\mathcal{W}_{k+1}, \mathcal{A}_{k+1}, C_{k+1}}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{k+1}} l_{k+1}(M_{k+1}(\mathbf{x}), \mathbf{y})$$

数据挖掘实验室
Data Mining Lab

■ Conditional Convolution



Optimal Relevance Mapping

- What have they done?

$$\mathbb{M}_{\mathbb{P}_k} \approx \prod_k \mathcal{L}_R \mathcal{N}(\mu_k, \sigma_k^2)$$

$$\mathcal{L}_R(x_k;\beta) = \frac{1}{1+\exp(-(\beta(x_k - 0.5)}$$

- What did they claim?

Our proposed approach builds on the following **Optimal Overlap Hypothesis**: *For a strictly continually trained deep neural network, catastrophic forgetting and remembering can be minimized, without additional memory or data, by learning optimal representational overlap, such that the representational overlap is reduced for unrelated tasks and increased for tasks that are similar.*

Optimal Relevance Mapping

What have they said?

- Understanding Catastrophic Forgetting

Task Loss

$$\underline{\log \mathcal{P}(\theta_1|D_1)} = \underline{\log \mathcal{P}(D_1|\theta_1)} + \log \mathcal{P}(\theta_1) - \log \mathcal{P}(D_1).$$

Prior info

$$\begin{aligned}\log \mathcal{P}(\theta_{1:2}|D_{1:2}) &= \log \mathcal{P}(D_2|\theta_2) + \underline{\log \mathcal{P}(\theta_1|D_1)} \\ &\quad - \log \mathcal{P}(D_2) \\ &= \log \mathcal{P}(D_2|\theta_2) + \log \mathcal{P}(D_1|\theta_1) \\ &\quad + \log \mathcal{P}(\theta_1) - \log \mathcal{P}(D_1) - \log \mathcal{P}(D_2).\end{aligned}$$

- How is it proved?

$$\mathcal{P}(\theta_1, \mathbb{M}_{\mathbb{P}1}|D_1) \propto \mathcal{P}(D_1|\theta_{\mathbb{M}_{\mathbb{P}1}})\mathcal{P}(\theta_{\mathbb{M}_{\mathbb{P}1}})$$

$$\begin{aligned}\mathcal{P}(\theta_{1:2}, \mathbb{M}_{\mathbb{P}2}|D_{1:2}) &\propto \mathcal{P}(D_2|\theta_{\mathbb{M}_{\mathbb{P}2}})\mathcal{P}(\theta_1, \mathbb{M}_{\mathbb{P}1}|D_1) \\ &\propto \mathcal{P}(D_2|\theta_{\mathbb{M}_{\mathbb{P}2}})\mathcal{P}(\theta_2^{''})\end{aligned}$$

Supervised Continual Learning

Algorithm 1 *RMN* Supervised Continual Learning

Input: data x, ground truth y for n tasks, prune parameter μ, corresponding task labels i paired with all x

Given: parameters $\mathbf{W}$ & initilaized relevance mappings $\mathbb{M}_{\mathbb{P}}$

for each task i **do**

  $f(x_i; \mathbf{W}, \mathbb{M}_{\mathbf{P_i}}) \Rightarrow \hat{y}_i = \sigma((W \odot \mathbb{M}_{P_i}) \odot x_i)$

  Compute Loss : $L(\hat{y}_i, y_i)$

  Optional: Add Sparsity Loss : $L(\hat{y}_i, y_i) + (\mathbb{M}_{P_i})_{l_0}$

  Backpropagate and optimize

  Prune $\mathbb{M}_{\mathbb{P}} \leq \mu$ only.

  Stabilize (fix) parameters in f where $\mathbb{M}_{\mathbb{P}} = 1$

end for

Inference: For data x and ground-truth task label i:

Output: $f(x, i; W)$

Optimal Relevanc

- Unsupervised

Algorithm 2 *RMN* Unsupervised Continual Learning

Input: data x, ground truth y, prune parameter μ
Given: parameters $\mathbf{W}$, $\mathbb{M}_{\mathbb{P}_{est\_j}}$ with $est\_j = 0$, Task Switch Detection Method *TSD*
for each task j **do**
  Filter input x on $f(x; \mathbf{W}, \mathbb{M}_{\mathbb{P}_{\mathbf{0 \ldots j-1}}})$
  $f(x; \mathbf{W}, \mathbb{M}_{\mathbb{P}_{\mathbf{est\_j}}}) \Rightarrow \hat{y}$
  Compute Loss : $L(\hat{y}, y)$
  if $TSD(x)$ is True **then**
    $est\_j{+}{+}$
    Add $\mathbb{M}_{\mathbb{P}_{est\_j}}$ to learn-able parameter list
    $f(x; \mathbf{W}, \mathbb{M}_{\mathbb{P}_{\mathbf{est\_j}}}) \Rightarrow \hat{y}$
    Re-Compute Loss : $L(\hat{y}, y)$
  end if
  Backpropagate and optimize
  Sample x_g from standard Gaussian distribution with same shape as x
  $f(x_g; \mathbf{W}, \mathbb{M}_{\mathbb{P}_{\mathbf{est\_j}}}) \Rightarrow \hat{y}$
  Compute Loss : $||\hat{y} - 0||_2^2$
  Backpropagate and optimize
  Prune $\mathbb{M}_{\mathbb{P}} \leq \mu$ only.
  Stabilize (fix) parameters in f where $\mathbb{M}_{\mathbb{P}} \approx 1$
end for
Inference: For data x:
Output: $max_k f(x, k; W)$

a Relevance modified Welsh's **t-test** on the KL divergence

- Is that real?

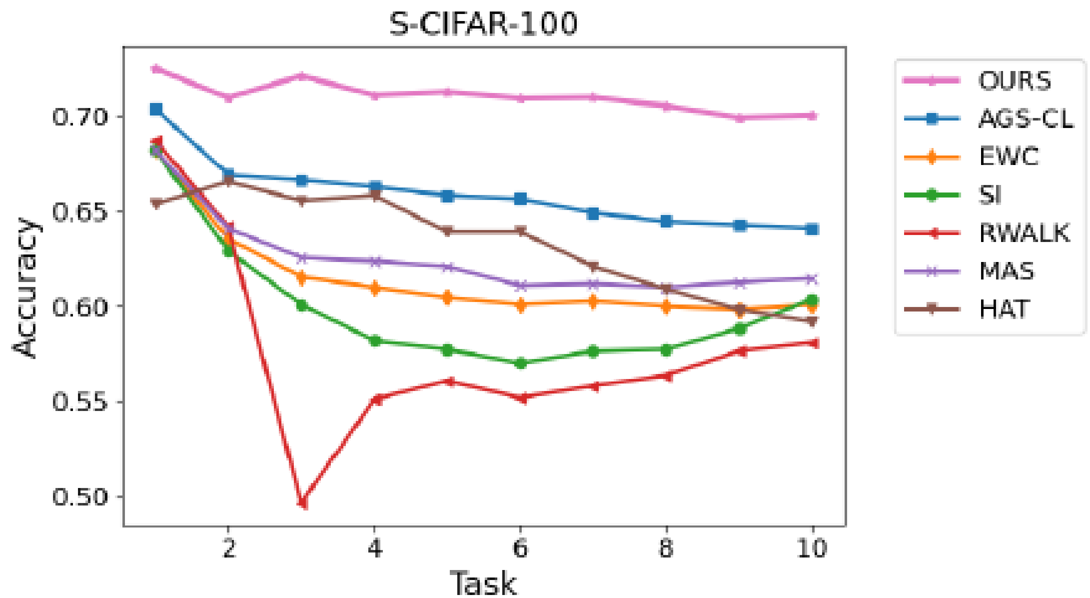


Figure 1. Average accuracy results on CIFAR-100 (10 tasks)

Summary and Discussion

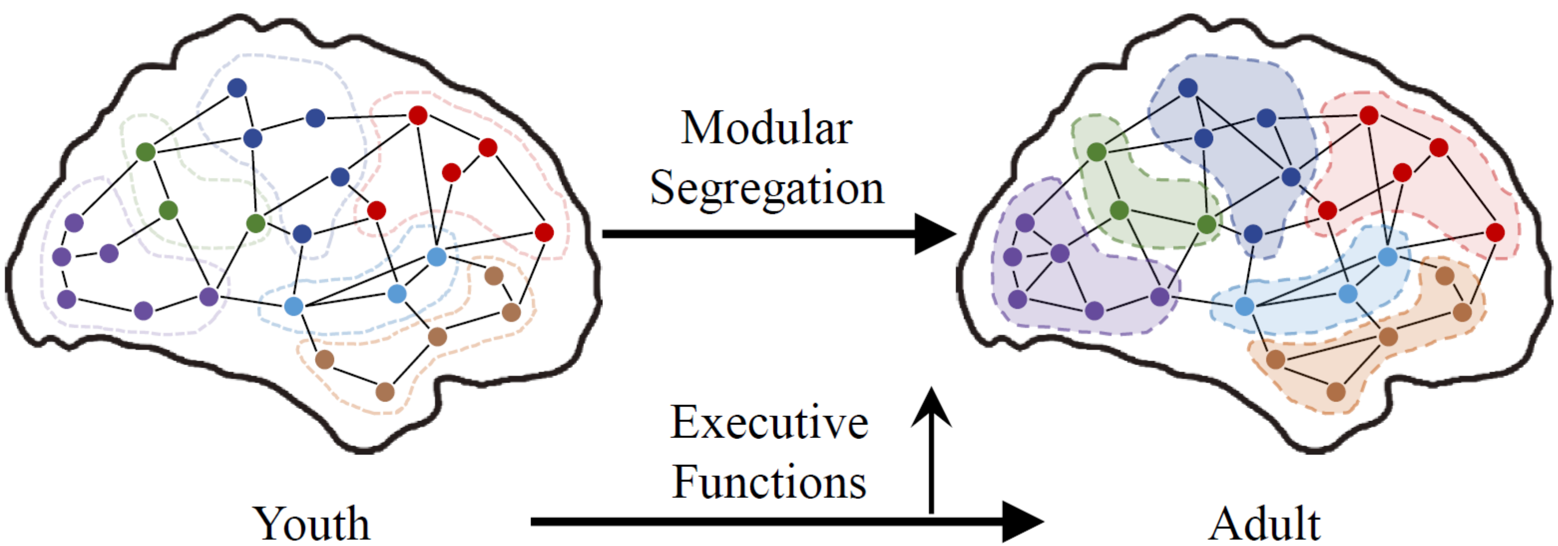
- Based on X (fc layer or t-test)
- Fix previous params? √
- Joint optimization? ×

Our proposals

Previous Work

■ Motivation

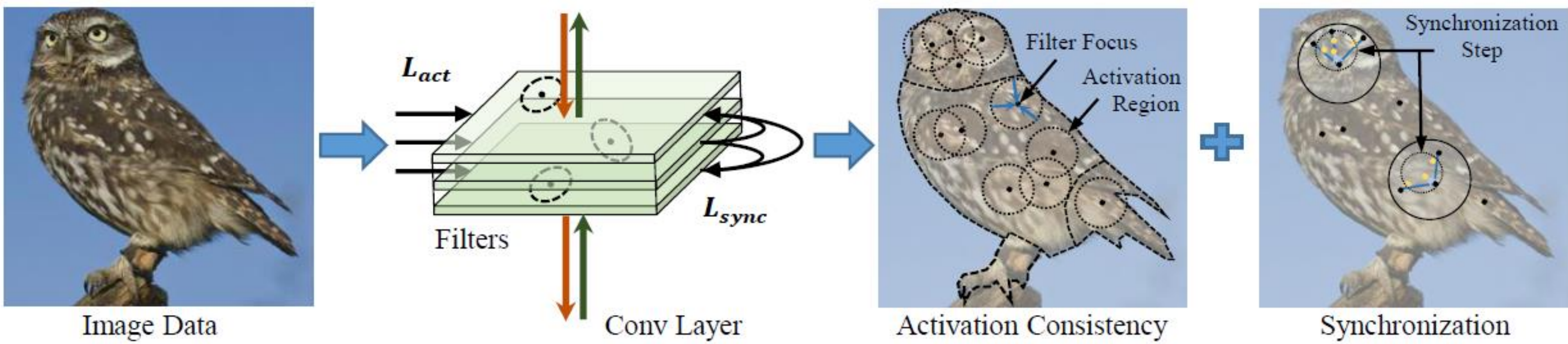
- Interpretation via Synchronization as Localization
- A Feature Map as an Entity -> Sync as Modules



数据挖掘实验室
Data Mining Lab

■ Idea

- Minimize Entropy from several levels
- Sync (unsupervised) for Module Segmentation



Functional Module Level

- SynC in clustering

$$\frac{d\theta_i}{dt} = \omega_i + \frac{S}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i), (i = 1, \ldots, N)$$

$$Nb_\epsilon(x) = \{y \in \mathcal{D} | dist(y, x) \leq \epsilon\}$$

$$x_i(t+1) = x_i(t) + \frac{1}{|Nb_\epsilon(x(t))|} \cdot \sum_{y \in Nb_\epsilon(x(t))} \sin(y_i(t) - x_i(t))$$

- SynC in mini-batch

$$L_{sync} = \frac{1}{N}\sum_{x_i}\frac{1}{|\, Nb_\epsilon(x_i) \,|}\sum_{x_j \in Nb_\epsilon(x_i)} dist(x_i, x_j)$$

Understanding Sync Mechanism

数据挖掘实验室
Data Mining Lab

■ Sync in mini-batch

$$L_{sync} = \frac{1}{N}\sum_{x_i}\frac{1}{|Nb_\epsilon(x_i)|}\sum_{x_j \in Nb_\epsilon(x_i)} dist(x_i, x_j)$$

- **Understanding**

$$L_{sync} = \frac{1}{N}\sum_{x_i}\frac{1}{|Nb_\epsilon(x_i)|}\sum_{x_j \in Nb_\epsilon(x_i)} dist(x_i, x_j)$$

Word Embedding!

$$\cong -\sum_{i,j} p(i)p(j|i)\log q(j|i) = H(j|i)$$

$$p(i) = \frac{1}{N} \qquad q(j|i) = \frac{e^{-dist(x_i,x_j)/\tau_2}}{\sum_j e^{-dist(x_i,x_j)/\tau_2}}$$

$$p(j|i) = \frac{Nb_\epsilon(x_i,x_j)}{\sum_j Nb_\epsilon(x_i,x_j)} = \begin{cases} \frac{1}{|Nb_\epsilon(x_i)|}, & dist(x_i,x_j) < \epsilon \\ 0, & otherwise \end{cases}$$

Motivation

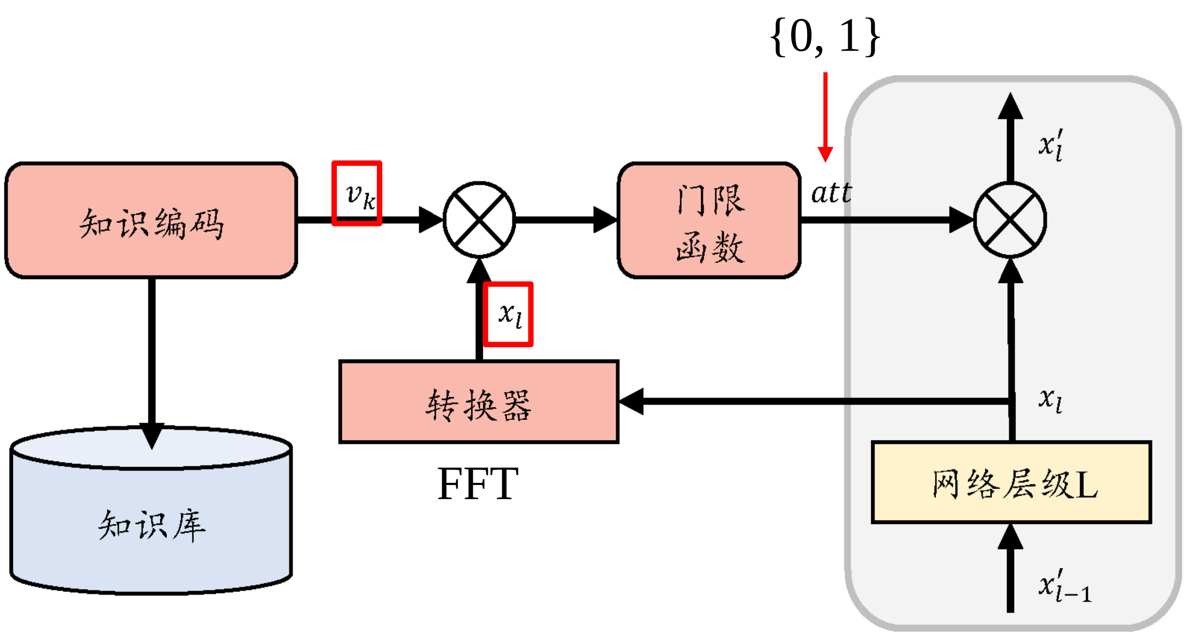
- Functional modules on **the whole structure**
- Unsupervised knowledge embedding
- Knowledge embedding management

Idea

- Fourier Transform
- Sync to regular representation space
- Attention mechanism

Attention Structure

- Knowledge embedding as Key vector
- Hard attention -> activate the knowledge or not



Proposal

- What's the cons?
  - 'Soft'-mask
  - No absolutely fixed parameter
  - Joint optimization

- So, Sync could be a promising way to it?
  - **Stable** functional module

What's the pros?

1. **Constant memory.** To avoid unbounded systems, the consumed memory should be constant w.r.t. the number of tasks or length of the data stream.

6. **Problem agnostic** continual learning is not limited to a specific setting (e.g. only classification).

7. **Adaptive** systems learn from available unlabeled data as well, opening doors for adaptation to specific user data.

8. **No test time oracle** providing the task label should be required for prediction.

9. **Task revisiting** of previously seen tasks should enable enhancement of the corresponding task knowledge.

10. **Graceful forgetting.** Given an unbounded system and infinite stream of data, selective forgetting of trivial information is an important mechanism to achieve balance between stability and plasticity.

Proposal

What’s the pros?

- Local, Disentangled, Interpretable, Comparable
- Knowledge management for Continual learning
- Multi-source, partial, model-based Transfer learning
- Zero-shot via compositional generalization
- Match with existing knowledge systems
- Interact with human beings
- ……

Proposal

数据挖掘实验室
Data Mining Lab

■ Plan

- Exan
- Adap
- Expe
- Gene g, comp , etc.)
- Theo tional)
- Perfo

Oxford 102 Flowers					
Method	Task 1	Task 2	Task 3	Task 4	Avg.
Finetuning	10.0 (-20.3)	5.1 (-17.1)	6.7 (-13.6)	17.3 (0.0)	9.8
Freezing	30.3 (0.0)	39.8 (0.0)	32.0 (0.0)	33.1 (0.0)	33.8
Joint	54.6 (+24.3)	58.9 (+11.5)	57.7 (+4.5)	47.0 (0.0)	54.6
EWC [14]	12.1 (-18.2)	11.6 (-38.1)	9.3 (-24.4)	25.8 (0.0)	14.7
HAT [41]	17.2 (-12.7)	19.3 (-28.5)	28.6 (+1.4)	31.6 (0.0)	24.2
PackNet [26]	32.0 (0.0)	53.7 (0.0)	43.6 (0.0)	37.9 (0.0)	41.8
TFM w/o FN	36.4 (0.0)	54.1 (0.0)	38.6 (0.0)	39.0 (0.0)	42.0
TFM	36.4 (0.0)	53.8 (0.0)	45.5 (0.0)	37.6 (0.0)	**43.3**

CUBS 200 Birds					
Method	Task 1	Task 2	Task 3	Task 4	Avg.
Finetuning	7.4 (-30.2)	2.6 (-30.0)	29.7 (-3.4)	43.1 (0.0)	20.7
Freezing	37.6 (0.0)	35.1 (0.0)	35.4 (0.0)	38.4 (0.0)	36.6
Joint	48.7 (+11.1)	52.1 (+6.0)	50.7 (+1.5)	51.9 (0.0)	50.8
EWC [14]	16.2 (-21.4)	19.0 (-21.2)	24.2 (-14.0)	41.7 (0.0)	25.3
HAT [41]	18.7 (-1.8)	19.4 (-0.4)	28.5 (-0.6)	31.2 (0.0)	24.4
PackNet [26]	35.3 (0.0)	42.8 (0.0)	44.4 (0.0)	45.9 (0.0)	42.1
TFM w/o FN	42.9 (0.0)	44.1 (0.0)	48.3 (0.0)	49.1 (0.0)	46.1
TFM	42.9 (0.0)	43.1 (0.0)	49.9 (0.0)	48.8 (0.0)	**46.2**

Stanford 40 Actions					
Method	Task 1	Task 2	Task 3	Task 4	Avg.
Finetuning	24.4 (-10.5)	26.5 (-7.7)	17.6 (-16.8)	28.9 (0.0)	24.4
Freezing	34.9 (0.0)	29.4 (0.0)	30.1 (0.0)	30.5 (0.0)	31.2
Joint	45.7 (+10.8)	40.3 (+4.8)	43.2 (-1.1)	40.2 (0.0)	42.4
EWC [14]	24.2 (-10.7)	28.2 (-2.0)	25.2 (-5.6)	34.3 (0.0)	28.0
HAT [41]	25.7 (-1.0)	25.5 (-2.7)	30.1 (-2.1)	34.4 (0.0)	28.9
PackNet [26]	32.5 (0.0)	32.9 (0.0)	36.7 (0.0)	34.3 (0.0)	34.1
TFM w/o FN	35.3 (0.0)	38.3 (0.0)	39.2 (0.0)	38.0 (0.0)	37.7
TFM	35.3 (0.0)	37.2 (0.0)	42.0 (0.0)	37.2 (0.0)	**38.0**

Plan

- Exam existing CL methods in the same setting
- Adapt from FcaNet, add HAT and Sync
- Experiments of CL and Interpretability(?)
- General Applications (transfer/zero-shot learning, compact existing networks & efficient networks, etc.)
- Theoretical guarantee of accuracy boundary (optional)
- Perform as a conference paper

Thank you