Continual Lifelong Learning: A Compacting, Picking and Growing Approach

Steven C. Y. Hung, Cheng-Hao Tu, Cheng-En Wu, Chein-Hung Chen, Yi-Ming Chan, and Chu-Song Chen

Institute of Information Science, Academia Sinica, Taipei, Taiwan

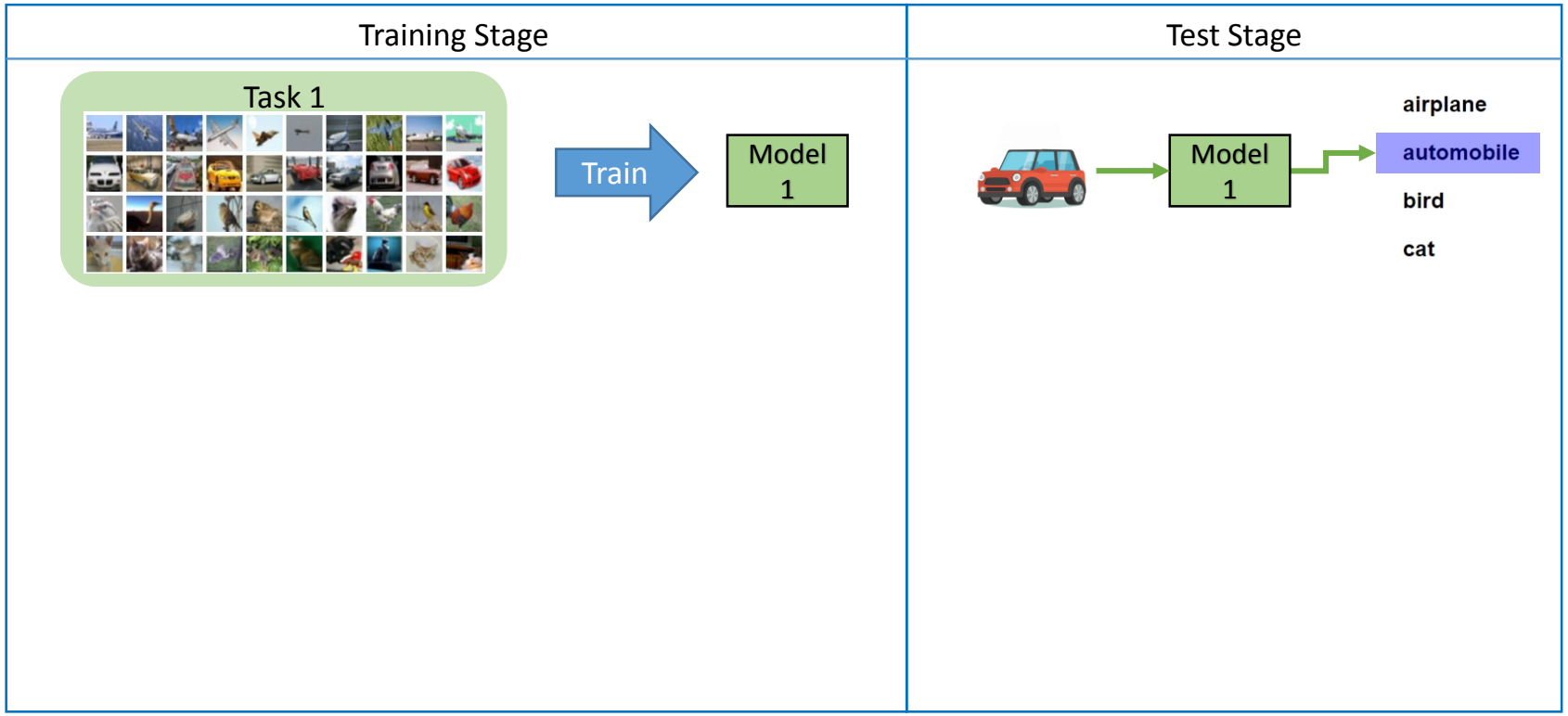
MOST Joint Research Center for AI Technology and All Vista Healthcare, Taipei, Taiwan

To appear in **NeurIPS 2019**

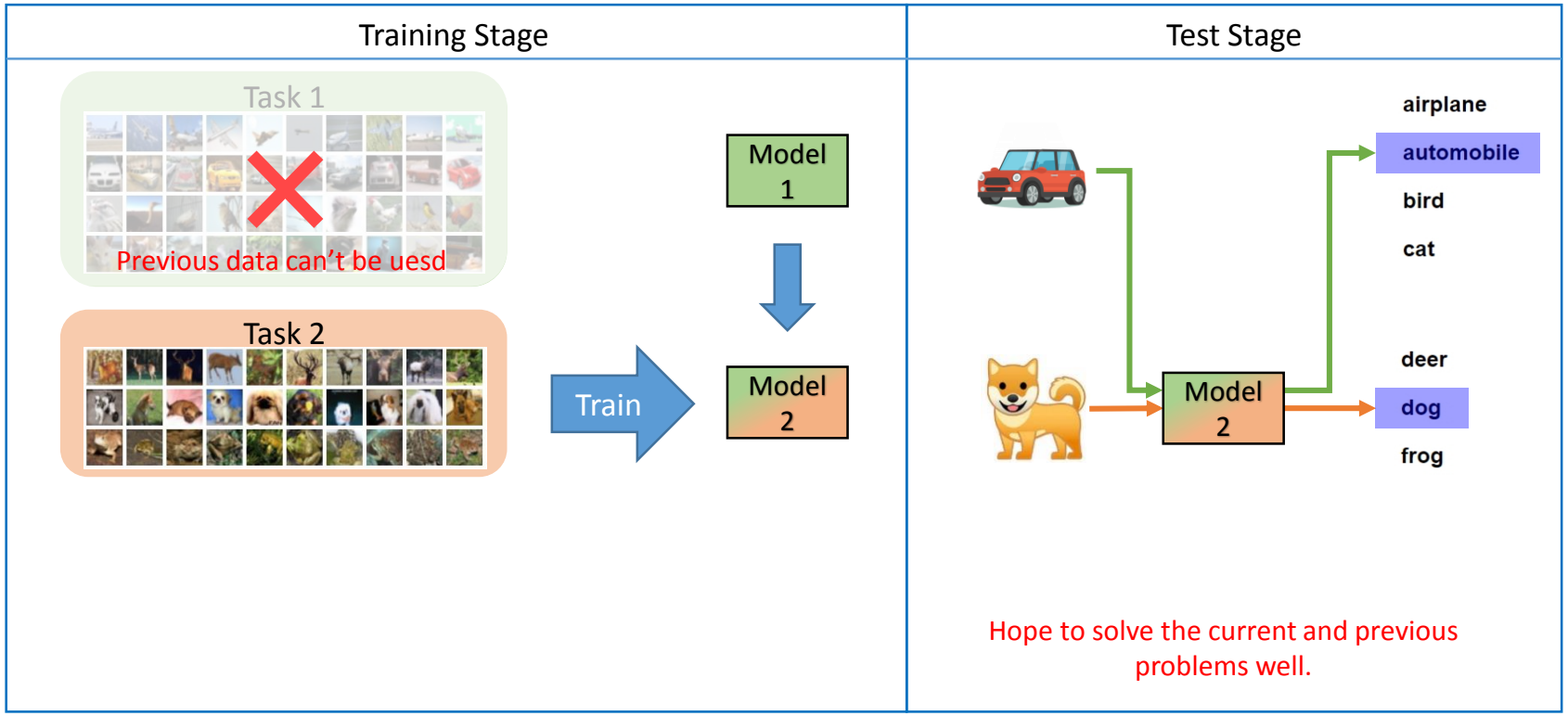
Continual lifelong learning

- Learn a model capable of handling unknown sequential tasks while keeping the performance on previously learned tasks.
- In continual lifelong learning, the training data of previous tasks are assumed non-available for the newly coming tasks.

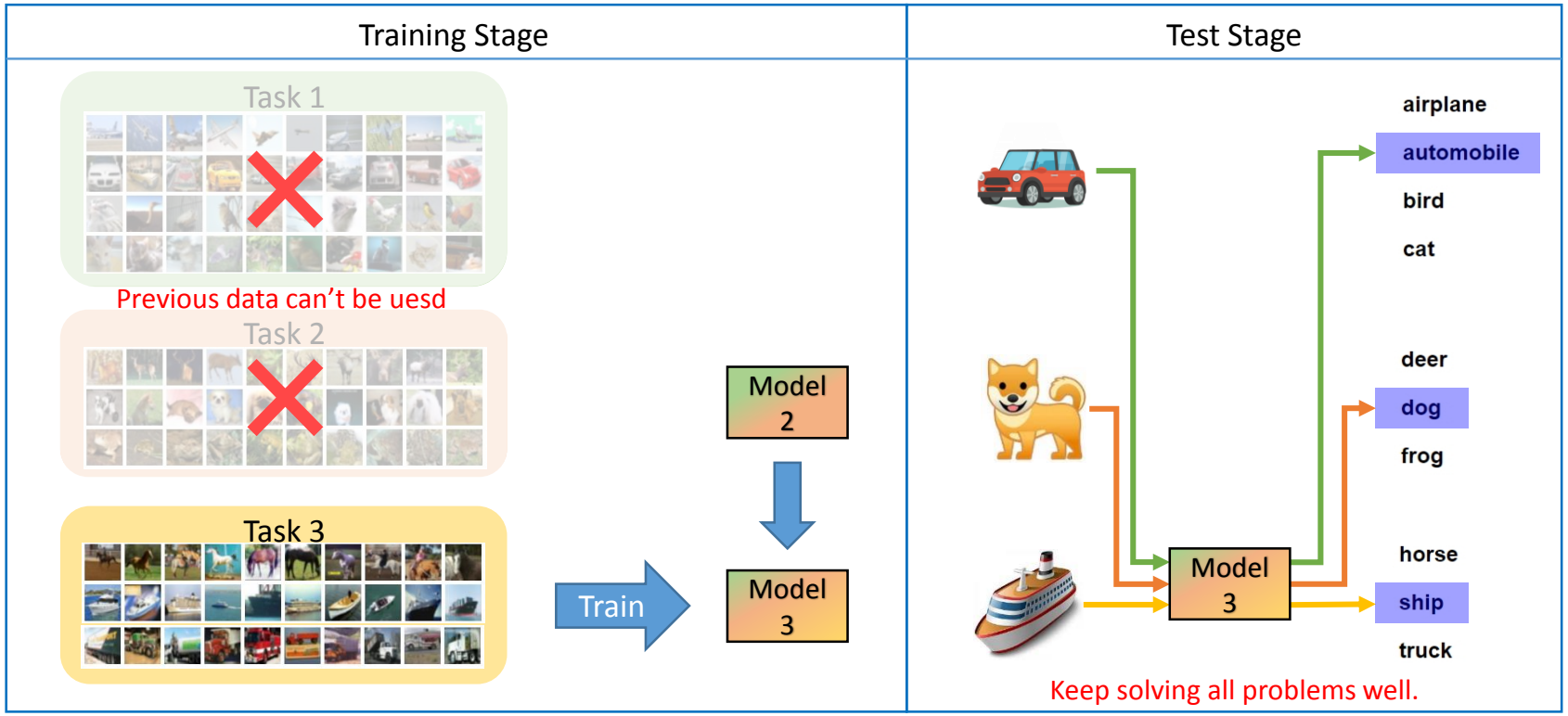
Continual lifelong learning illustration example



Continual lifelong learning illustration example



Continual lifelong learning illustration example



Catastrophic forgetting

- Although the model learned can be used as a pre-trained model, fine-tuning a model for the new task will force the model parameters to fit new data, which causes **catastrophic forgetting** on previous tasks.
- Lessen the catastrophic forgetting: one central issue of lifelong learning.

Our work

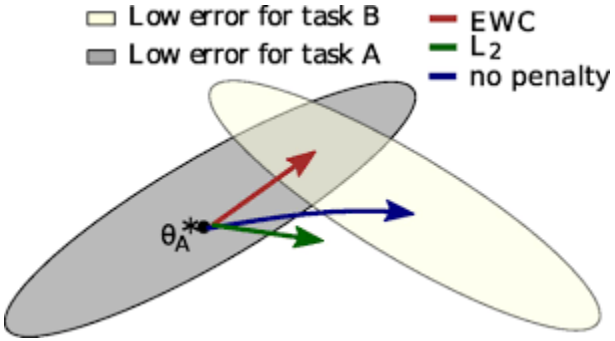
- Another issue:
  - As humans have the ability to continually acquire, fine-tune and transfer knowledge and skills throughout their lifespan, in lifelong learning, we would hope that the experience accumulated from previous tasks is helpful to learn a new task.

- Characteristics of our method
  - **Avoid forgetting**
  - **Expand with shrinking**
  - **Compact knowledge base**

Previous works (I)

Network regularization

- Main idea: restrict the update of model weights.
  - EWC [PNAS17] regularize the network weights and hope to search the common convergence for the current and previous tasks.



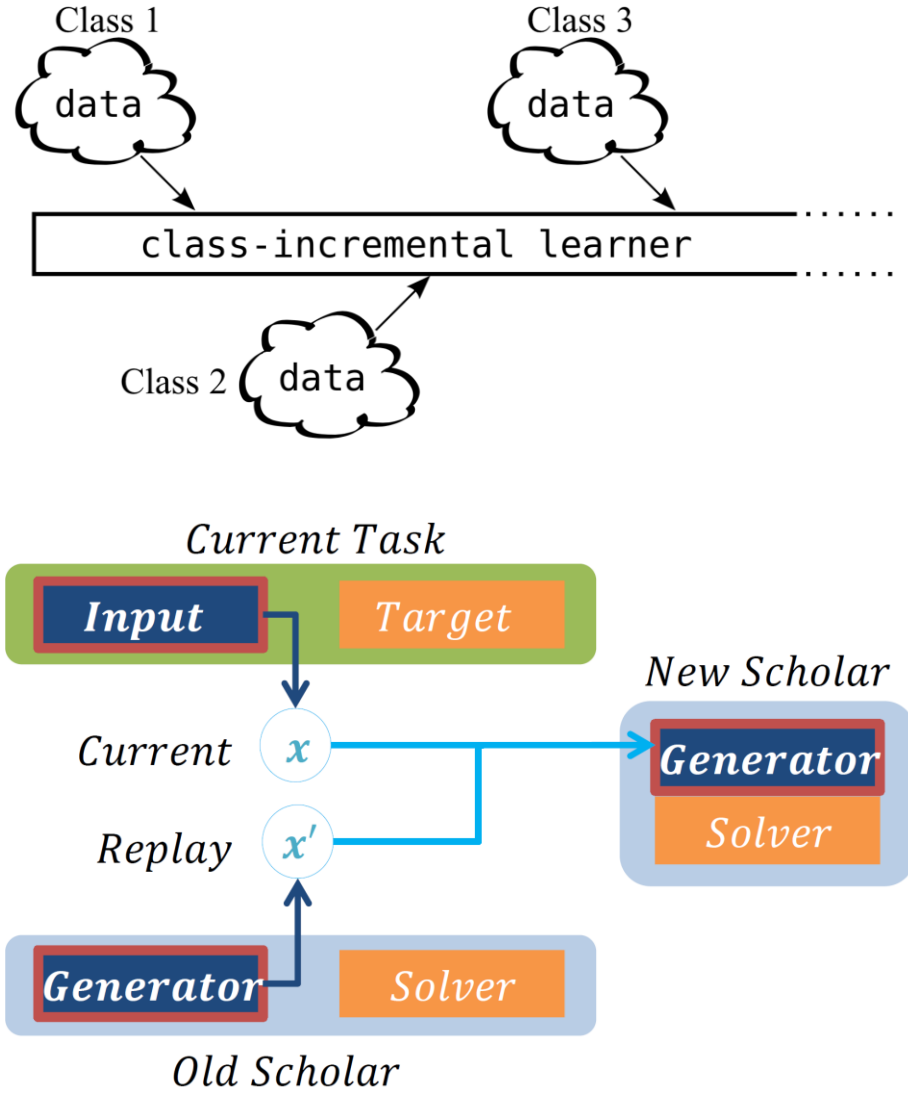
  - Online EWC [ICML18] and EWC++ [ECCV18] improve the efficiency of EWC.
  - Learning without Memorizing (LwM) [CVPR19] builds an attention map, and enforces that the attention region of the previous and concurrent models are consistent.

- Alleviate catastrophic forgetting but cannot guarantee the accuracy of previous tasks exactly.

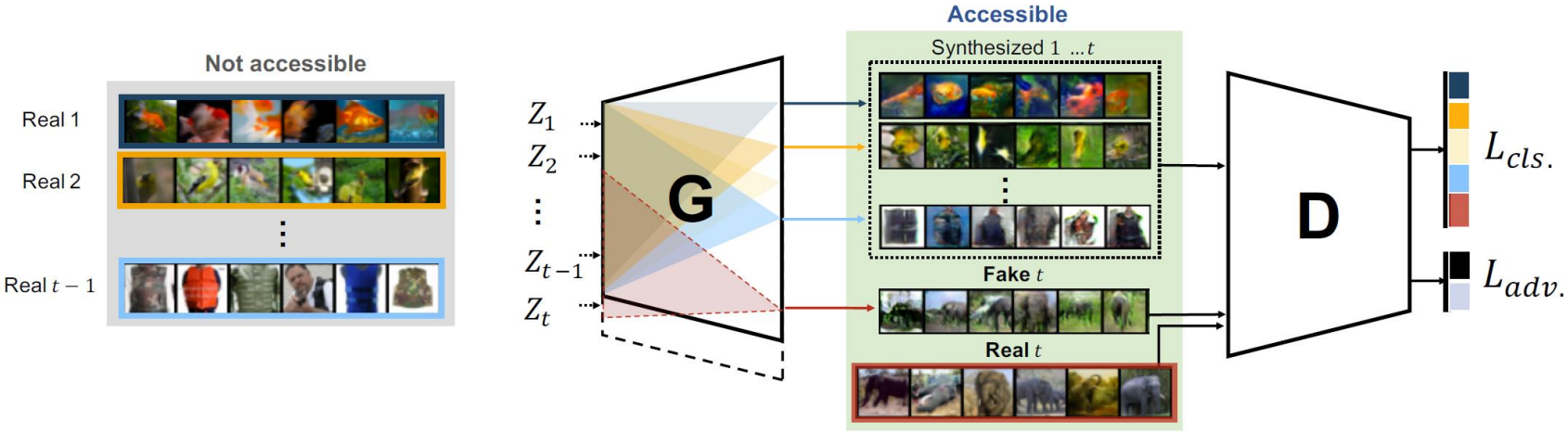
Previous works (II)

Memory replay

- Main idea: use extra models to keep data information.
  - Data-preserving approaches (such as [CVPR17, ICLR19, AAAI19]) directly save important data or latent codes as an efficient form.
  - Generative Replay [39] introduces GANs to lifelong learning.
  - Memory Replay GANs (MeRGANs) [NeurIPS18] use replay data to enhance the generator quality.



Previous works (II) – cont.



- Dynamic Generative Memory (DGM) [CVPR19] uses neural masking to learn connection plasticity in conditional generative models, with a dynamic expansion.

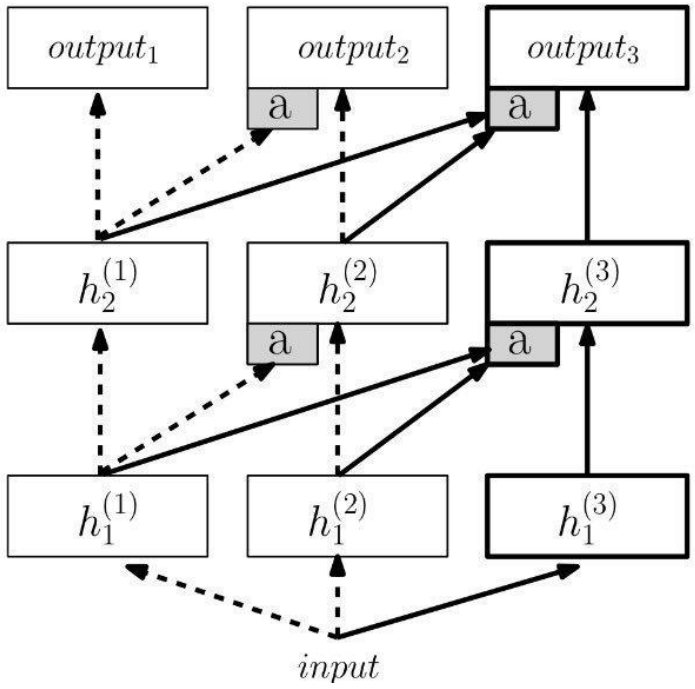
- Despite keeping data information, memory-replay approaches still cannot guarantee the exact performance of past tasks.

Previous works (III)

Dynamic architecture

- Main idea: adapt the architecture with new tasks.
  - ProgressiveNet [DeepMind 2016] expands the architecture for new tasks and keeps the function mappings via the preserved weights.

- **Turning data to weights.**



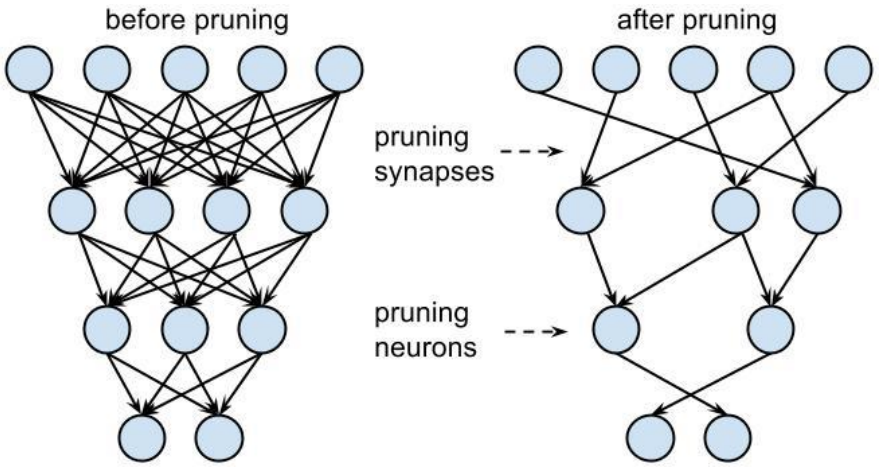
Previous works (III) – cont.

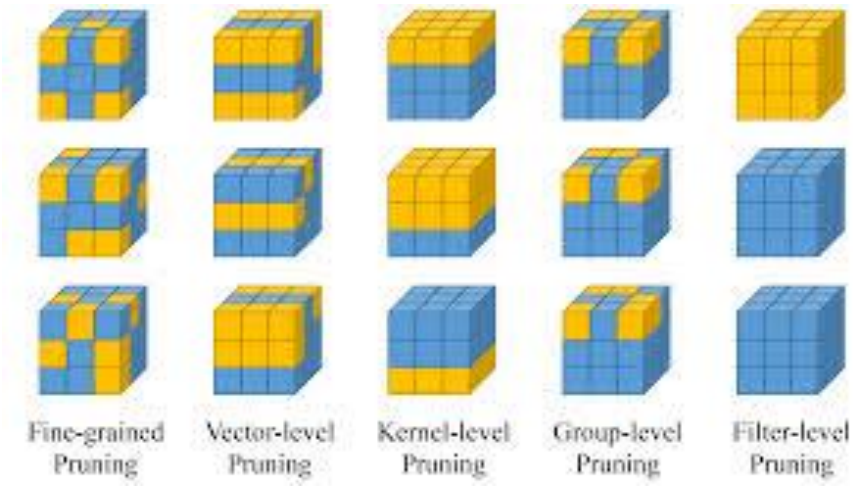
- LwF [IEEE TPAMI18] divides the model into two parts, shared and task-specific.
- DAN [IEEE TPAMI19] extends the architecture per new task, while each layer in the new-task model is a linear combination of the original filters of the base model.

- By architecture expansion, catastrophic forgetting is considerably lessened or avoided.
- However, the model is monotonically increased and a redundant structure is yielded.

Motivation of our method (I)

- Perform model compression for the current task, so that a condensed model is established for the old tasks.
  - According to deep-net compression [ICLR16], there is much redundancy in a neural network, and removing the redundant (usually small) weights does not affect the network performance.



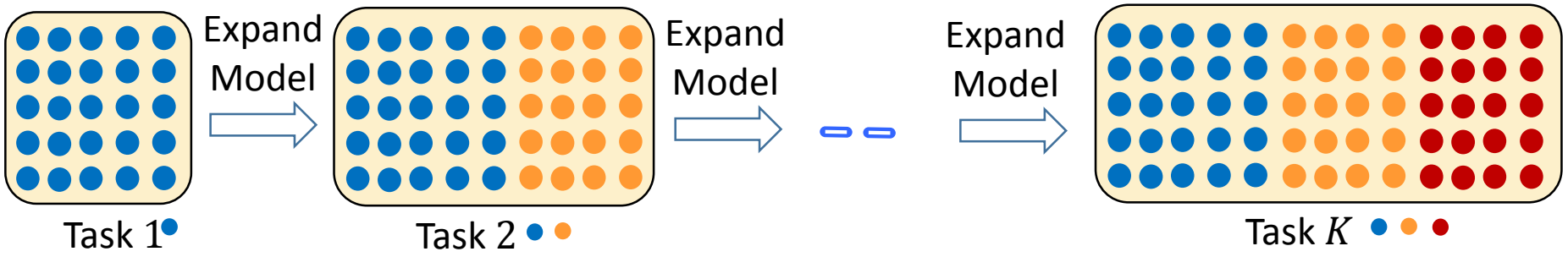


Motivatino of our method (II)

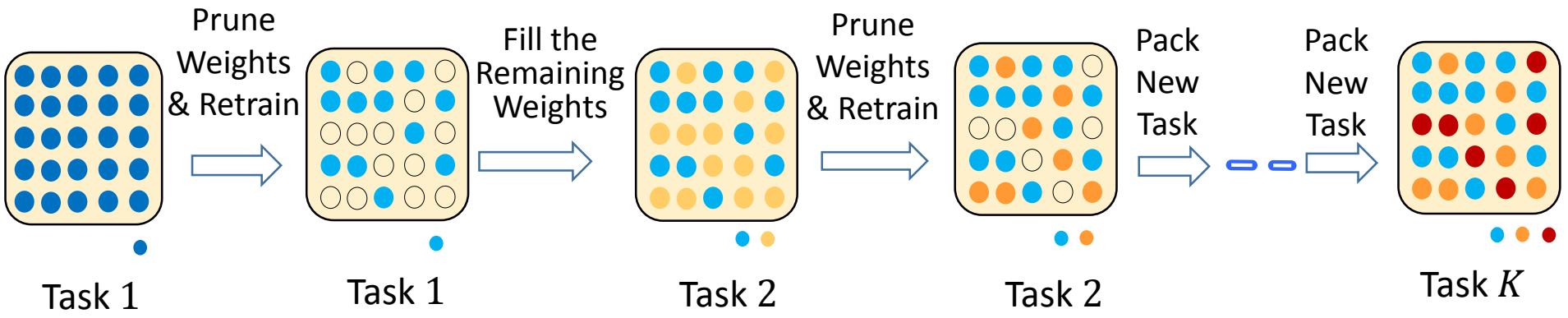
- Our approach exploits this property, which compresses the current task by deleting neglectable weights.

- This yields a compressing and growing loop for a sequence of tasks.

Illustration of ProgressiveNet & PackNet

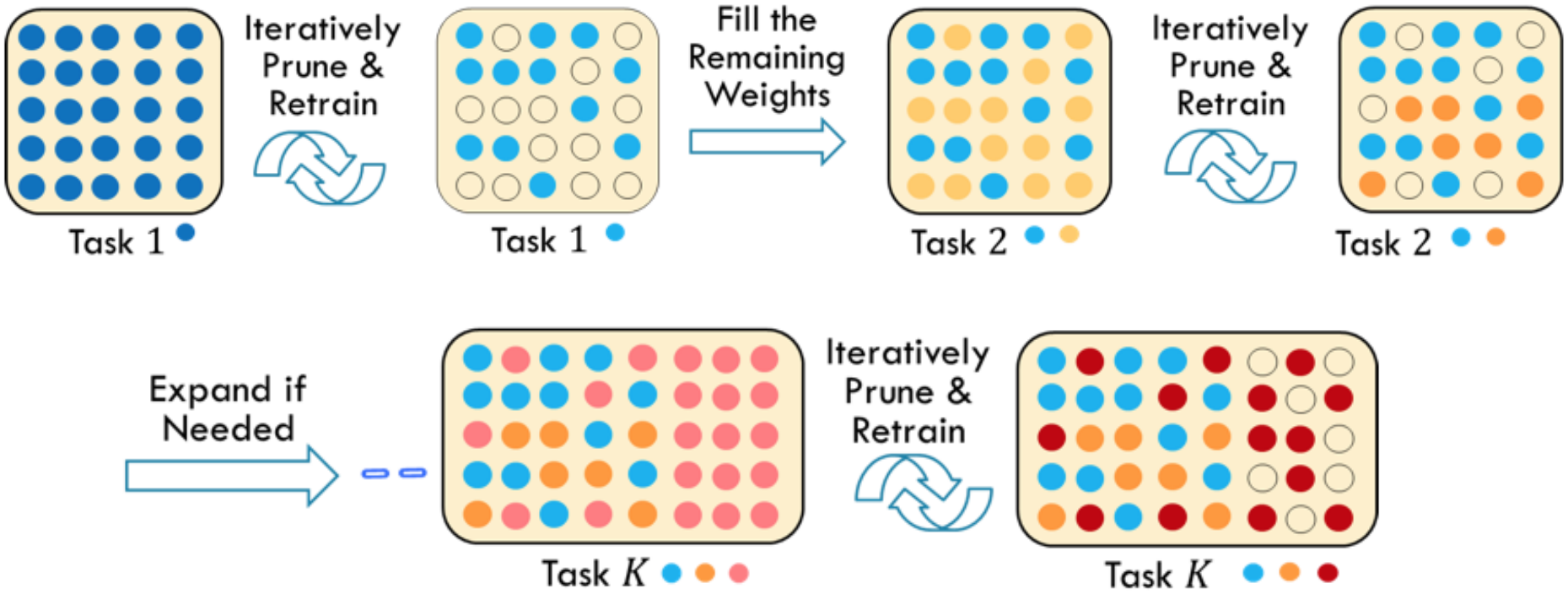


Progressive NeuralNet [DeepMind 2016] (√ Avoid forgetting; × Compactness; √ Extensible)



PackNet [CVPR18] (√ Avoid forgetting; √ Compactness; × Extensible)

Illustration of our approach-v1: PAE (ICMR 2019)

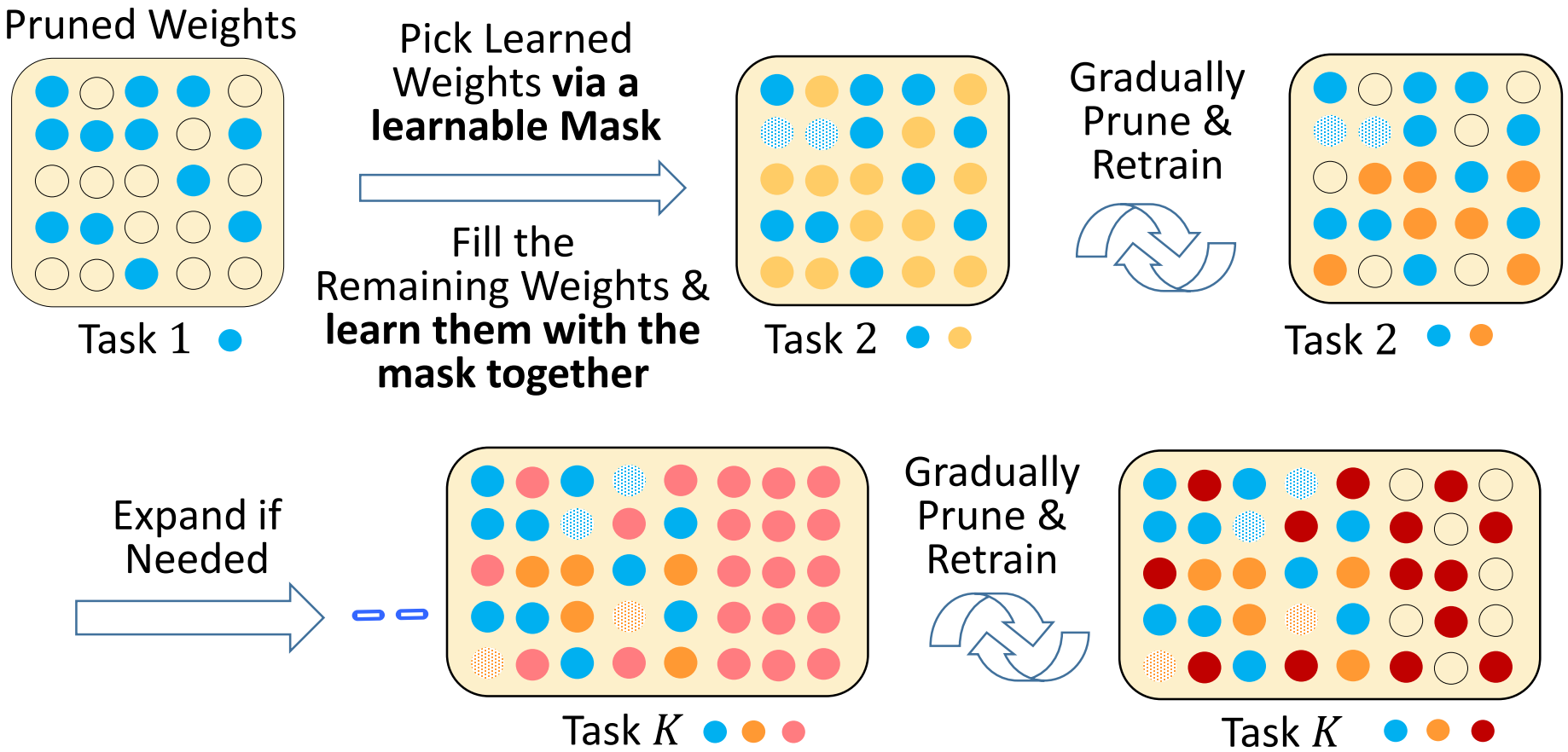


Pack & Expand (PAE) [icmr19] **(**√ Avoid forgetting; √ Compactness; √ Extensible**)**

- The weights previously learned serve as a knowledge base. There could be the case that only a part of the previous weights are suitable for the new task.
- However, PAE always re-use all previous weights, which would not be the most favorable choice for the new task.

Our approach-v2: Compacting, Picking & Growing (CPG)

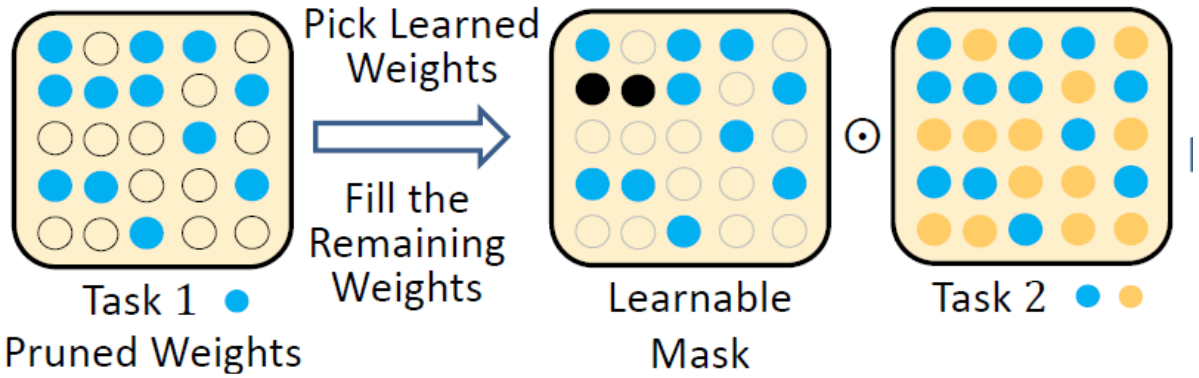
(NeurIPS 2019)



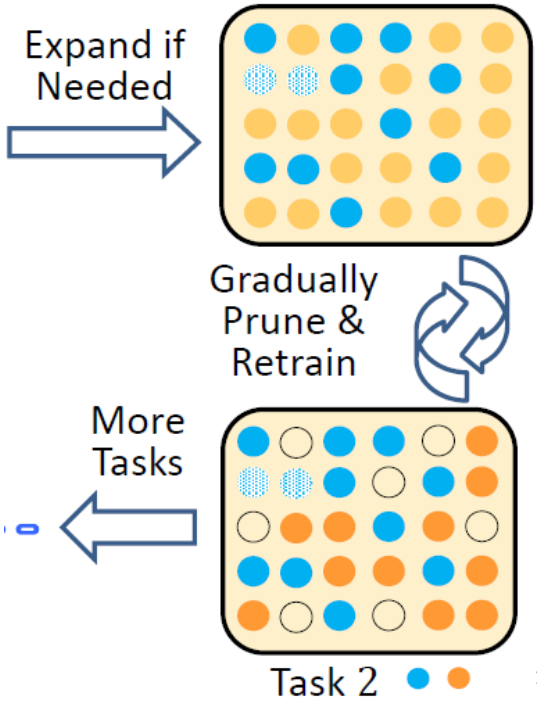
Compacking Picking & Growing (CPG) [under submission] (√ Avoid forgetting; √ Compactness; √ Extensible; √ Exploiting previous knowledge better)

CPG

- Old-weights picking and new-weights adapting.

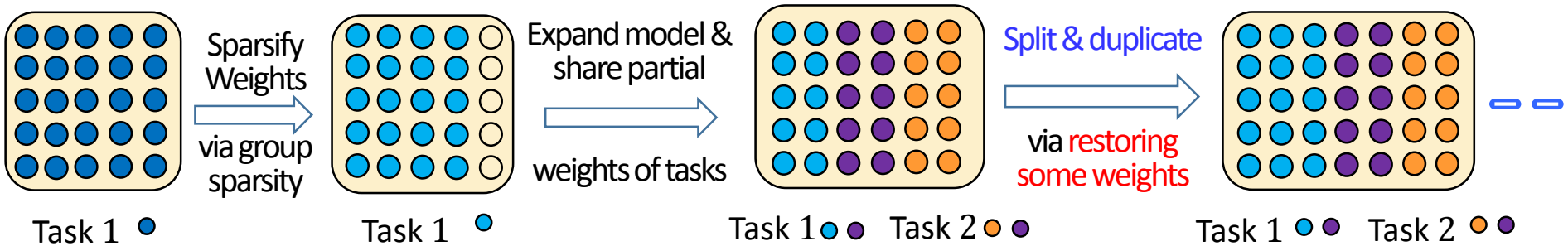


- Compression: after leaned, pruning the current task weights to consolidate the model.
  - Sparse on both old-weigh-picking and new-weights-learning sides.
  - Release weights: for forthcoming tasks.
  - New capacity: new or released weights can be used for new tasks.



Related work

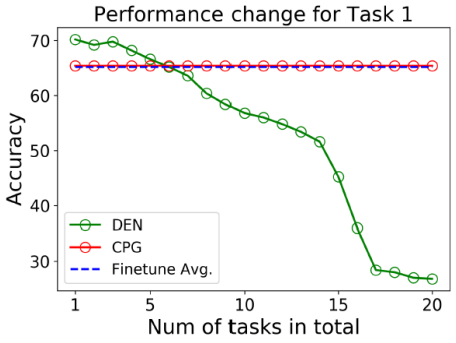
- DEN [ICLR18] reduces the weights of the previous tasks via sparse-regularization.
  - However, DEN does not ensure non-forgetting since part of the old-tasks weights are selected and modified under the sparse setting.
  - A "Split & Duplication" step is introduced to further ‘restore’ some of the old weights modified, so as to lessen the forgetting effect.



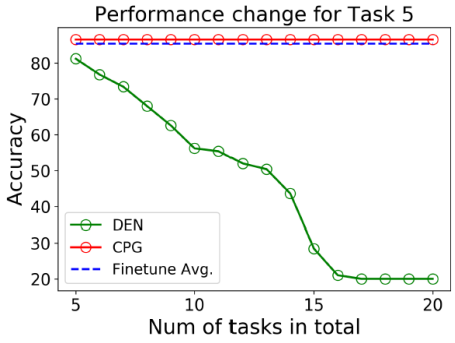
Dynamic Expansion Net (DEN)

20 tasks on CIFAR100 dataset (I)

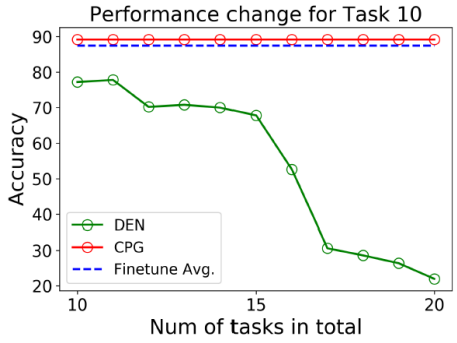
- Divide CIFAR-100 into 20 tasks. Each has 5 classes, 2500 training and 500 testing images.
- VGG16-BN model (VGG16 with batch normalization layers) is employed to train the 20 tasks sequentially.
- Compared our CPG to DEN [ICLR18], which also has a compression-expansion loop. (both expands 1.09x on model parameters.)



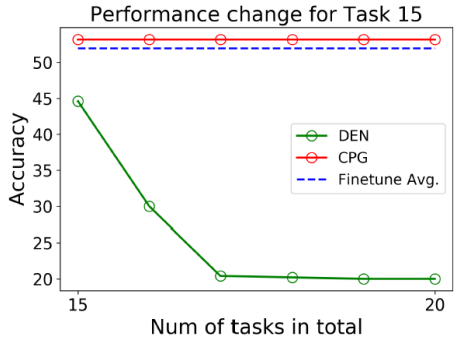
(a) Task-1



(b) Task-5



(c) Task-10



(d) Task-15

20 tasks on CIFAR100 dataset (III)

- Compare CPG with PackNet (without expansion) and our previous approach PAE, where they pick always all of the previous weights.
  - To verify whether the `picking' step in CPG is useful for selecting more useful knowledge.

Methods	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Avg.	Exp. (×)	Red. (×)
PackNet	66.4	80.0	76.2	78.4	80.0	79.8	67.8	61.4	68.8	77.2	79.0	59.4	66.4	57.2	36.0	54.2	51.6	58.8	67.8	83.2	67.5	1	0
PAE	67.2	77.0	78.6	76.0	84.4	81.2	77.6	80.0	80.4	87.8	85.4	77.8	79.4	79.6	51.2	68.4	68.6	68.6	83.2	88.8	77.1	2	0
CPG	65.2	76.6	79.8	81.4	86.6	84.8	83.4	85.0	87.2	89.2	90.8	82.4	85.6	85.2	53.2	74.4	70.0	73.4	88.8	94.8	80.9	1.5	0.41

20 tasks on CIFAR100 dataset (II)

- The performance of CPG and individual models on CIFAR-100 twenty tasks.
  - scratch: train from random weights.
  - fine-Avg: average accuracy of fine-tuning from a previous model randomly selected and repeats the process 5 times.
  - fine-Max: maximal accuracy of these 5 random trials.

Methods	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Avg.	Exp. (×)	Red. (×)
Scratch	65.8	78.4	76.6	82.4	82.2	84.6	78.6	84.8	83.4	89.4	87.8	80.2	84.4	80.2	52.0	69.4	66.4	70.0	87.2	91.2	78.8	20	0
fine-Avg	65.2	76.1	76.1	77.8	85.4	82.5	79.4	82.4	82.0	87.4	87.4	81.5	84.6	80.8	52.0	72.1	68.1	71.9	88.1	91.5	78.6	20	0
fine-Max	65.8	76.8	78.6	80.0	86.2	84.8	80.4	84.0	83.8	88.4	89.4	83.8	87.2	82.8	53.6	74.6	68.8	74.4	89.2	92.2	80.2	20	0
CPG avg	65.2	76.6	79.8	81.4	86.6	84.8	83.4	85.0	87.2	89.2	90.8	82.4	85.6	85.2	53.2	74.4	70.0	73.4	88.8	94.8	80.9	1.5	0.41
CPG max	67.0	79.2	77.2	82.0	86.8	87.2	82.0	85.6	86.4	89.6	90.0	84.0	87.2	84.8	55.4	73.8	72.0	71.6	89.6	92.8	81.2	1.5	0
CPG top	66.6	77.2	78.6	83.2	88.2	85.8	82.4	85.4	87.6	90.8	91.0	84.6	89.2	83.0	56.2	75.4	71.0	73.8	90.6	93.6	81.7	1.5	0

Fine-grained tasks [cvpr18, eccv18]

- Six tasks, including ImageNet, CUBS, Standford Cars, Flowes, Wikiart, and Sketch.
- Unlike the CIFAR-100 case, the first task is ImageNet, which serves as a strong base for fine-tuning of the others.

Dataset	Train from Scratch	Finetune	Prog. Net	PackNet	Piggyback	CPG
ImageNet	76.16	-	76.16	75.71	76.16	75.81
CUBS	40.96	82.83	78.94	80.41	81.59	83.59
Stanford Cars	61.56	91.83	89.21	86.11	89.62	92.80
Flowers	59.73	96.56	93.41	93.04	94.77	96.62
Wikiart	56.50	75.60	74.94	69.40	71.33	77.15
Sketch	75.40	80.78	76.35	76.17	79.91	80.33
Model Size (MB)	554	554	563	115	121	121

Facial informatics tasks

- Starting from a face-recognition model, add sequentially the gender, expression and age tasks.

Task	Train from Scratch	Finetune	CPG
Face	$99,417 \pm 0.367$	-	99.300 ± 0.384
Gender	83.70	90.80	89.66
Expression	57.64	62.54	63.57
Age	46.14	57.27	57.66
Exp. ($\times$)	4	4	1
Red. ($\times$)	0	0	0.003

Conclusions and Future Works for Continual Lifelong Learning

- We introduce a new approach, CPG, for continual lifelong learning, which
  - prevents forgetting,
  - maintains the model compactness when growing,
  - can select and reuse previous knowledge to yield better models for new tasks,
  - is sustainable and easy to be implemented.
- Future works:
  - The picking masks are allowed to be overlapped with tasks currently. We will examine non-overlapping picking masks to reduce the pick-mask storage.
  - Support “selectively forgetting” some previous tasks if needed.
  - Compress by removing filters.
  - Extend to lifelong learning without task boundaries.