Compacting, Picking and Growing for Unforgetting Continual Learning

Steven C. Y. Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen,
Yi-Ming Chan, and Chu-Song Chen
Institute of Information Science, Academia Sinica, Taipei, Taiwan

NeurIPS 2019

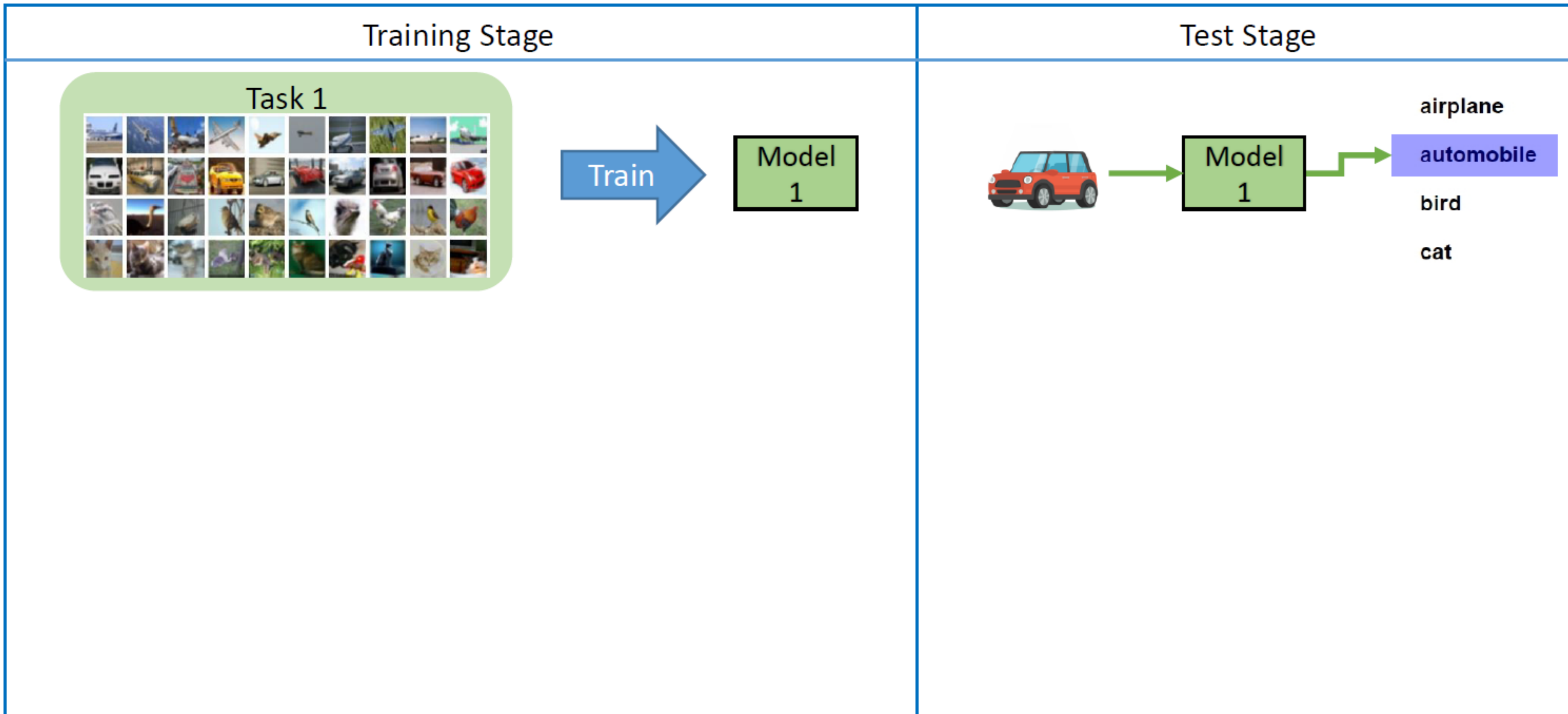
Presenter: Cheng-Hao Tu

Introduction – Continual Learning

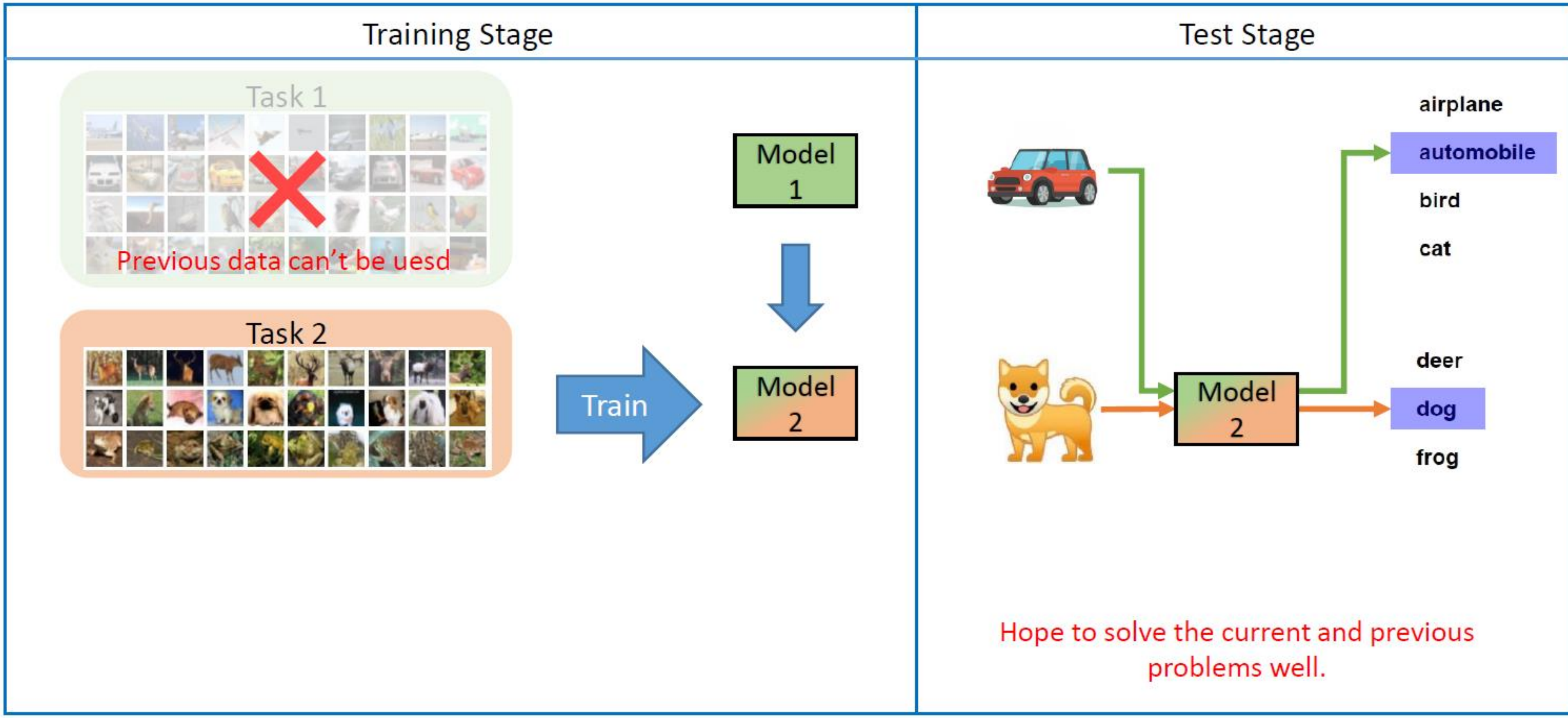
- Continual learning aims at learning an unknown sequence of tasks while keeping the performance of previously learned ones.

- The training data of previous learned tasks are assumed to be unavailable for new tasks.
  - **EX: storage constraints or privacy reasons**

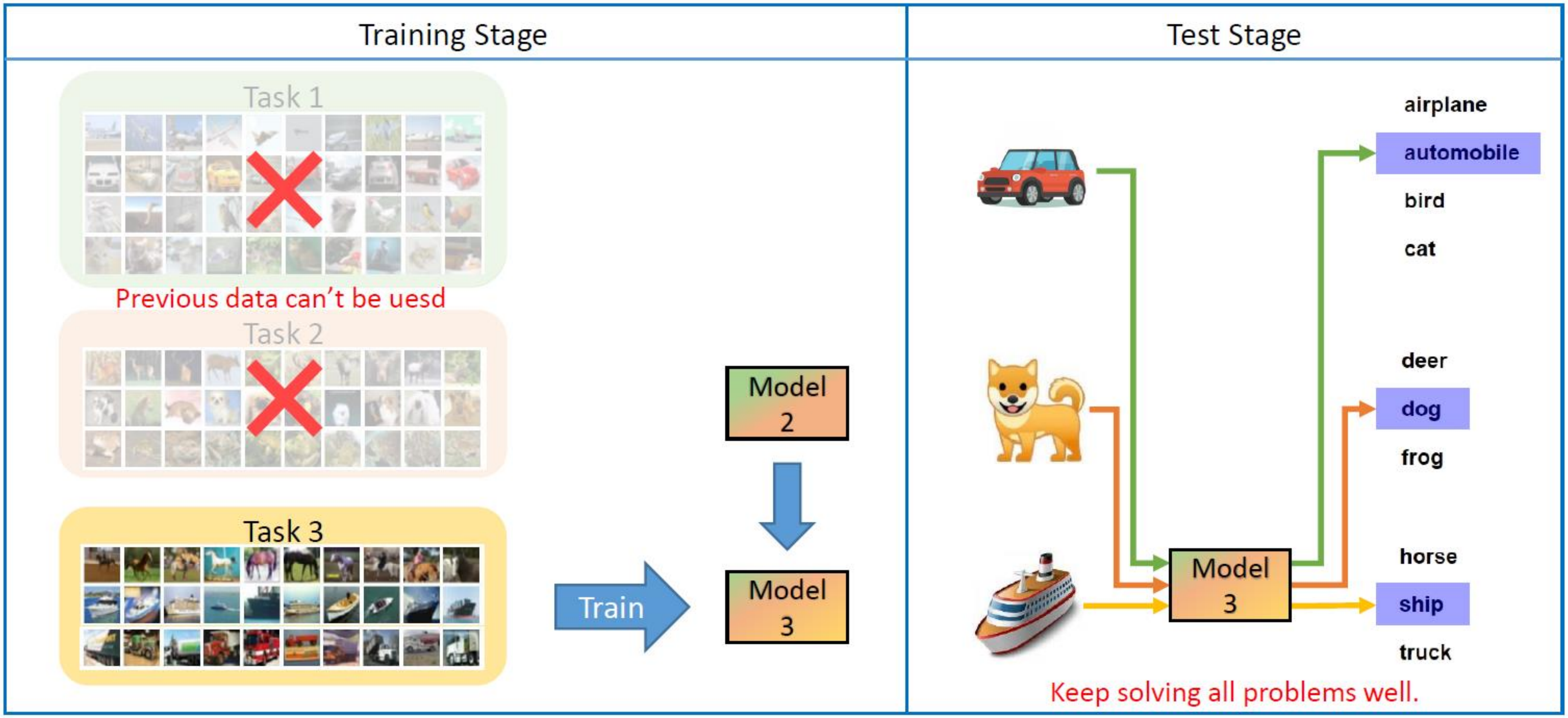
Introduction – Illustration Example



Introduction – Illustration Example



Introduction – Illustration Example



Introduction – Continual Learning

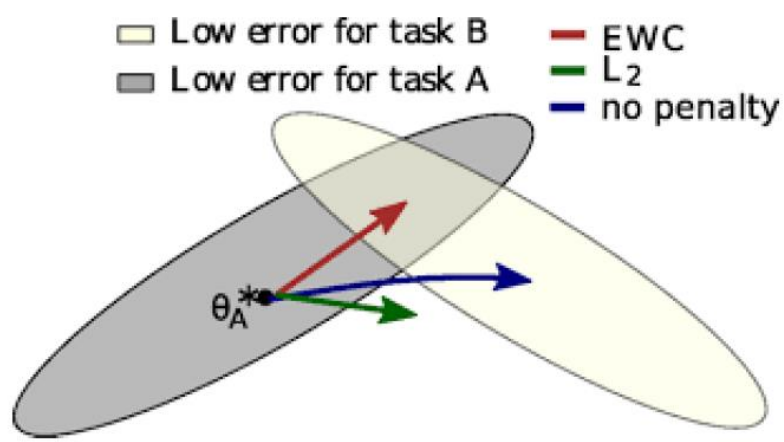
- Main issue: **Catastrophic forgetting**
  - Training only on data of the new task will force parameters to fit on the new data.
  - For example: Fine-tuning a model trained on a previous task will degrade its performance on the previously learned task.

Related Work – Memory Replay

- **Memory Reply**
  - Use extra models or memories to keep the data information of previous tasks.
  - Reduce the forgetting by jointly training with the replayed data.
- Data Preserving [CVPR17, ICLR19, AAAI19]
- Generative Models [NeurIPS17, CVPR19]

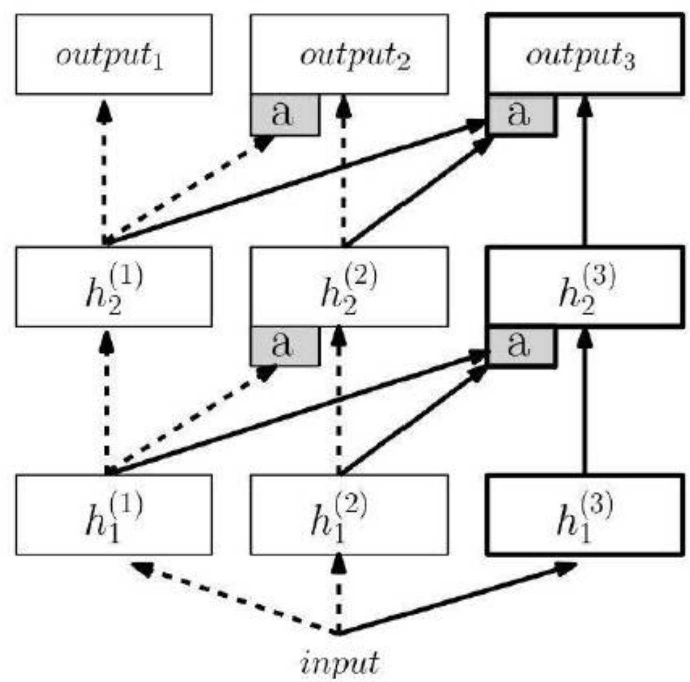
Related Work – Model Regularization

- **Model Regularization**
  - Restrict the update of model weights.
  - Alleviate forgetting but cannot exactly guarantee the accuracy of previous tasks.
- EWC [PNAS 2017]
- LwM [CVPR19]



Related Work – Dynamic Structure

- **Dynamic Structure**
  - Adapt the architecture with new tasks.
  - Forgetting can be avoided by keeping the weights unchanged.
- PackNet [CVPR18]
- DAN [TPAMI 2018]
- ProgressiveNet [DeepMind 2016]

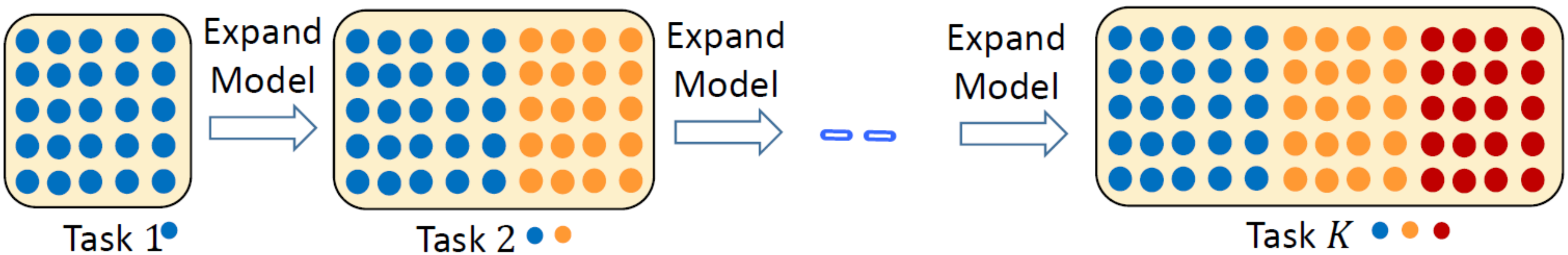


Objective of Our Work

- Avoid forgetting
- Maintain the compactness of our model
- Utilize knowledge learned from previous tasks

Methodology – Motivation

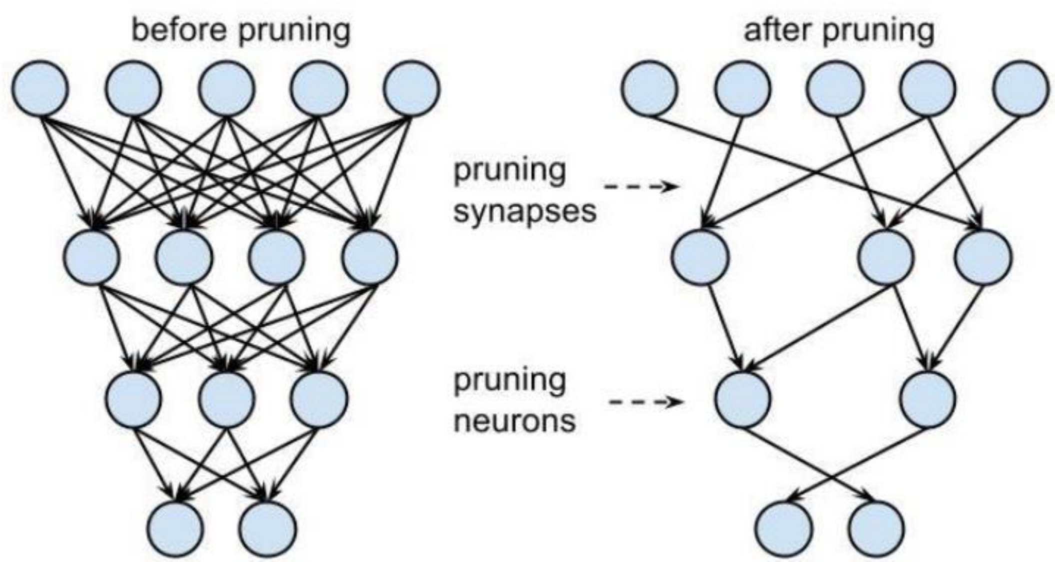
- ProgressiveNet [DeepMind 2016] expands the network structure every time a new task arrives and results in a **redundant structure**.



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

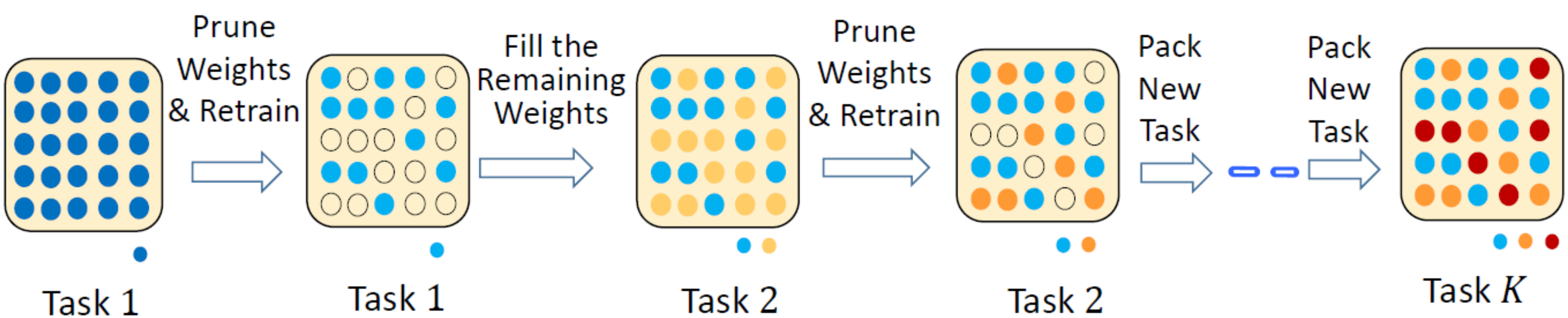
Methodology – Motivation

- According to deep-net compression [ICLR16], there is much redundant in a network, and removing these weights does not affect the performance too much.



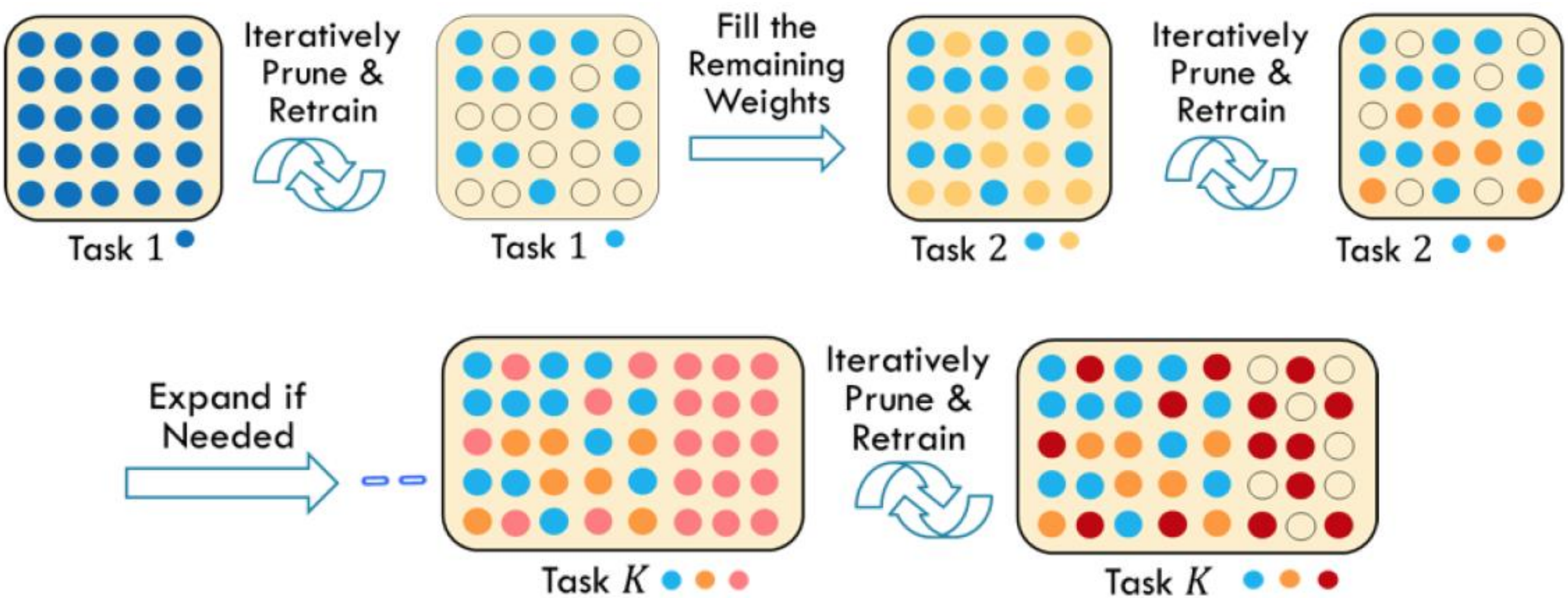
Methodology – Motivation

- PackNet [CVPR18] exploits this property and compresses the model after learning a new task.



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

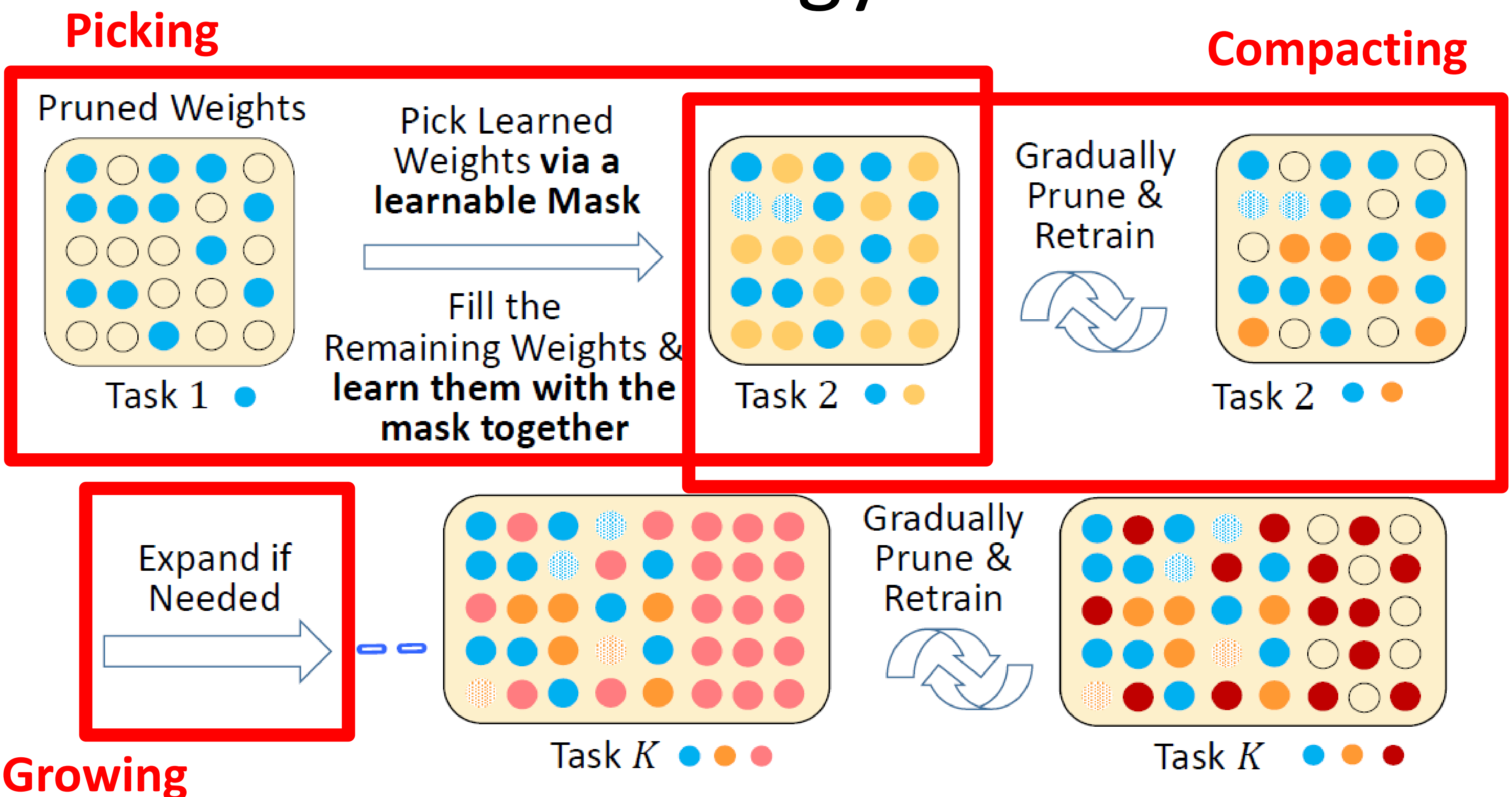
Methodology – Expand and Shrink



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

- However, fixed **weights of previous tasks accumulate** and dominate the outputs of future tasks.

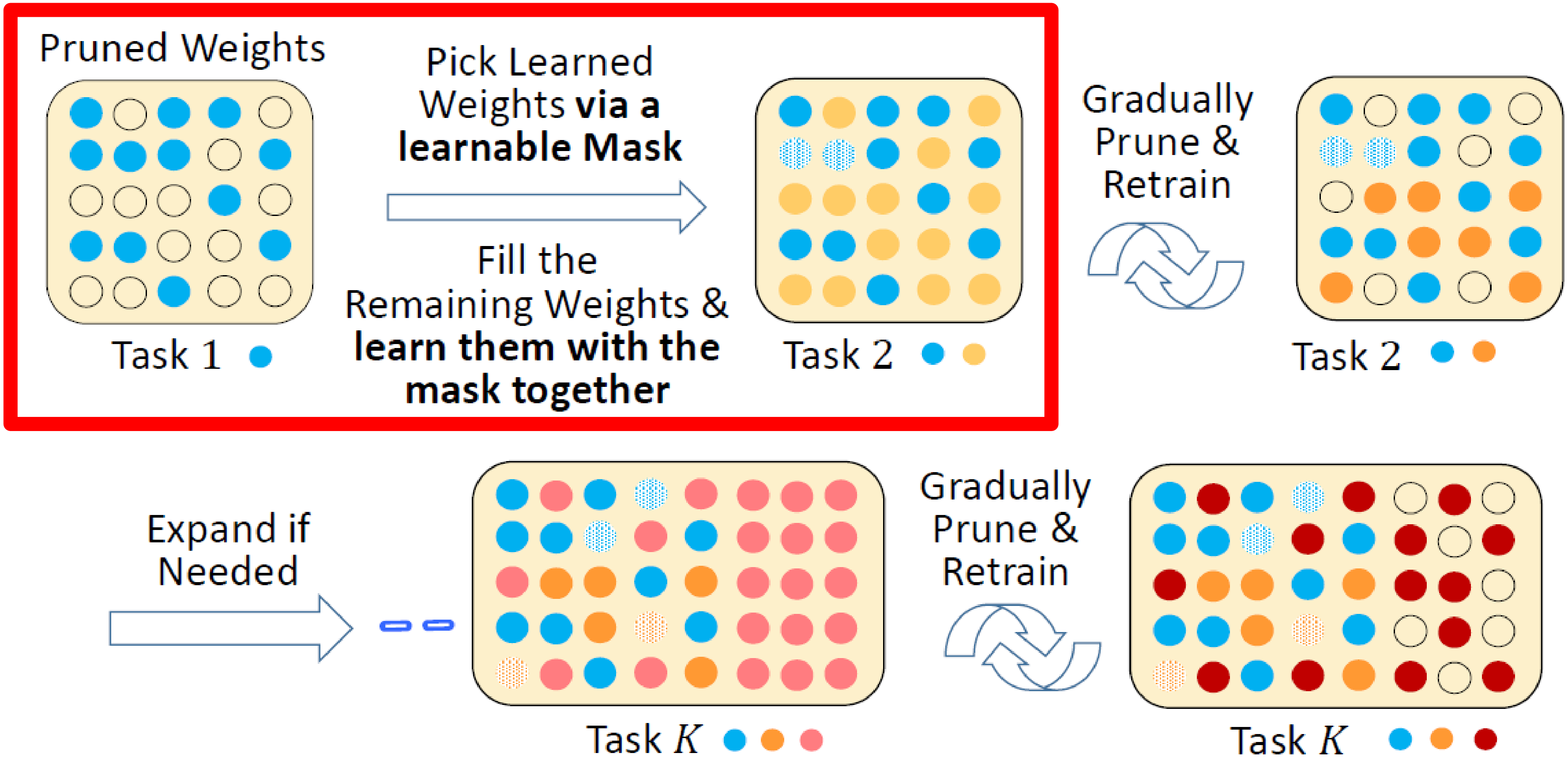
Methodology – CPG



Compacking Picking & Growing (CPG) [NeurIPS 2019] (√ Avoid forgetting; √ Compactness; √ Extensible; √ Exploiting previous knowledge better**)**

Methodology – CPG

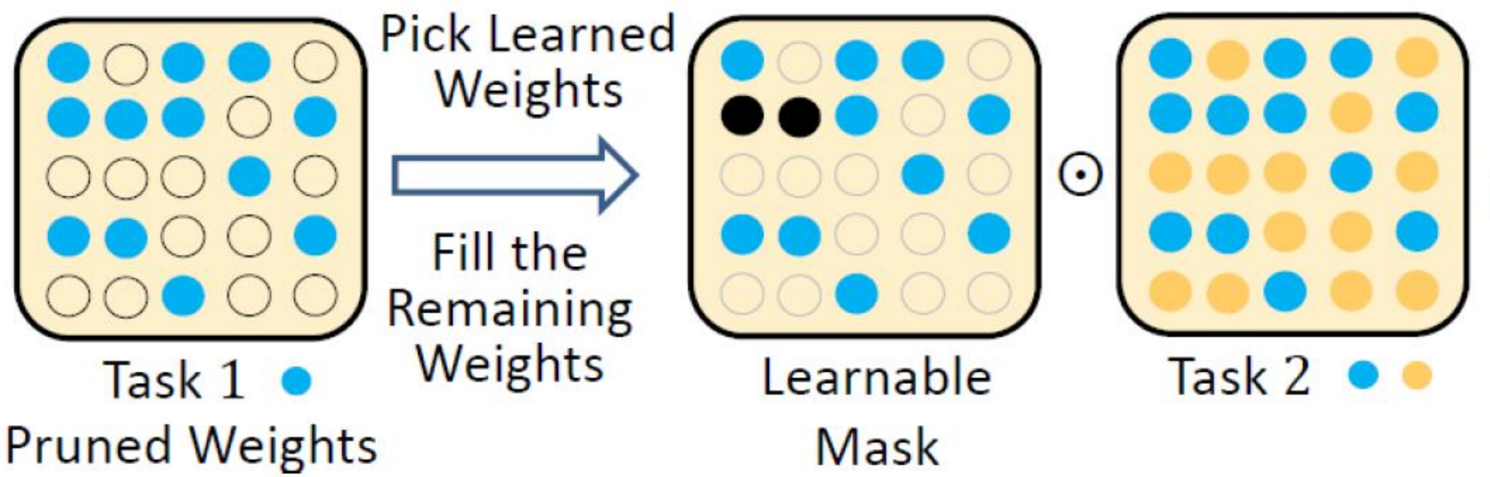
Picking



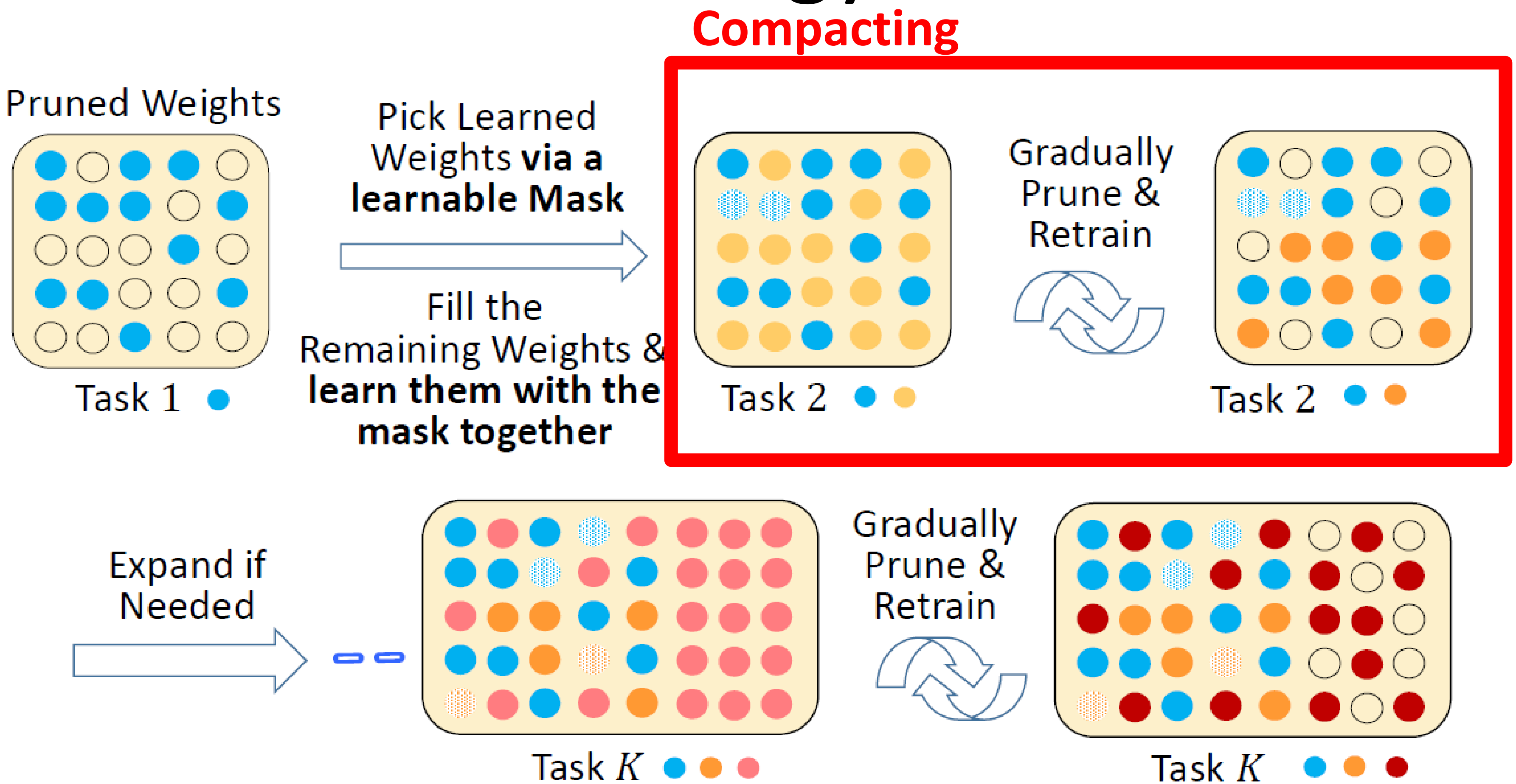
Compacking Picking & Growing (CPG) [NeurIPS 2019] (√ Avoid forgetting; √ Compactness; √ Extensible; √ Exploiting previous knowledge better**)**

Methodology – CPG (Picking)

- **Old-weights picking** and **new-weights adapting**.



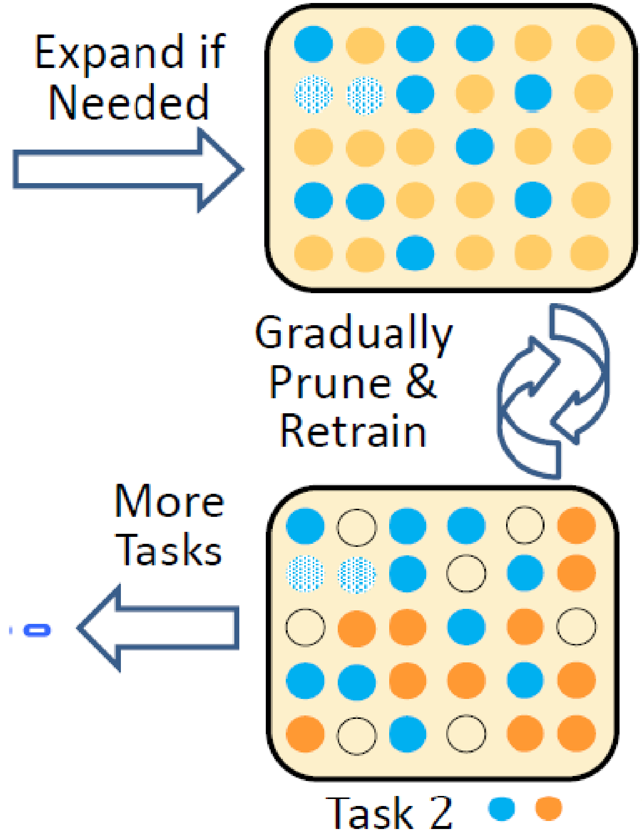
Methodology – CPG



Compacking Picking & Growing (CPG) [NeurIPS 2019] (√ Avoid forgetting; √ Compactness; √ Extensible; √ Exploiting previous knowledge better**)**

Methodology – CPG (Compacting)

- Compression: Prune the current task weights after learned.
  - New or released weights can be used for new tasks.
  - **Gradual pruning** to iteratively remove neglectable weights and retrain the model.



Methodology – CPG Summary

- Avoid forgetting ➔ By keeping the learned weights unchanged.

- Maintain the compactness of our model ➔ By expanding and shrinking loops.

- Utilize knowledge learned from previous tasks ➔ By picking the old-task weights.

Experiments

- Divide CIFAR-100 into 20 tasks, and each has 5 classes. We use VGG16-BN to train the 20 tasks sequentially.

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

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

Experiments

- Six tasks include ImageNet, CUBS, Stanford Cars, Flowers, Wikiart and Sketch. ResNet-50 is used.

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

Experiments

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

Conclusion

- We introduce a new approach, CPG, for continual learning which
  - Prevents forgetting.
  - Maintains the model compactness while growing.
  - Can select and reuse previous knowledge efficiently for new tasks.