Class Incremental Learning

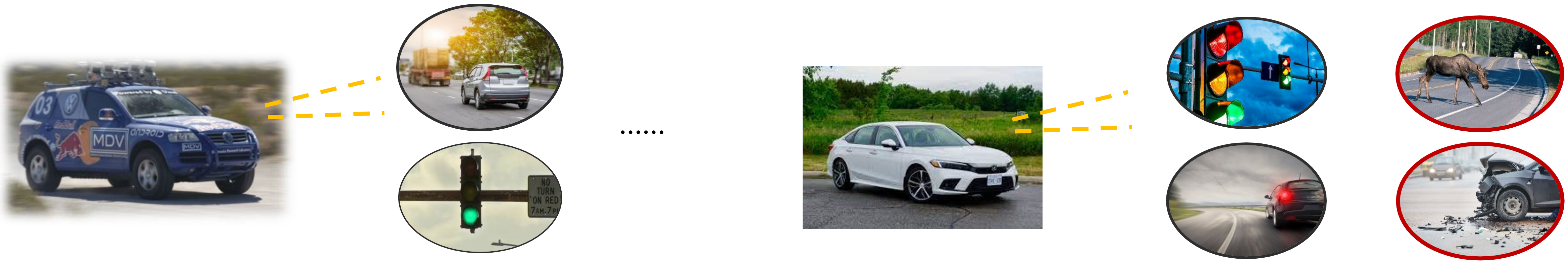
From *Backward* to *Forward* Compatible

Han-Jia Ye

School of Artificial Intelligence, Nanjing University

yehj@nju.edu.cn

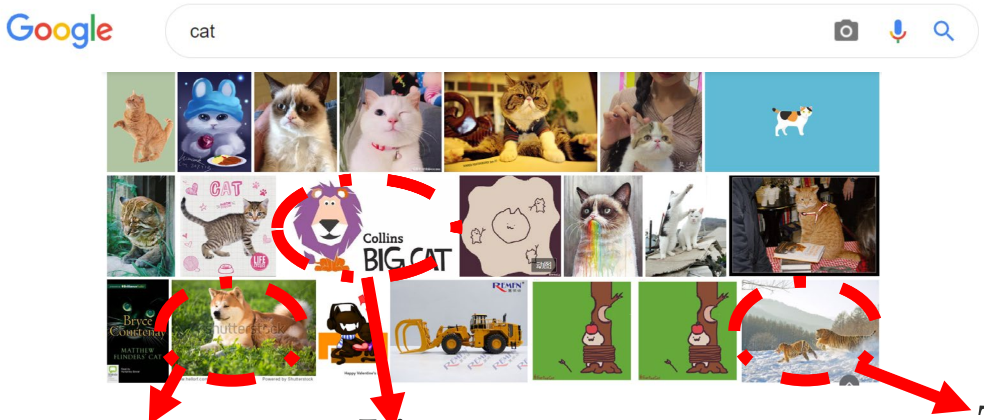
Closed and Open World



Assumed scenes

Real-world scenes

Self-Driving Vehicle



Dog

Lion

Tiger

Searching Engine

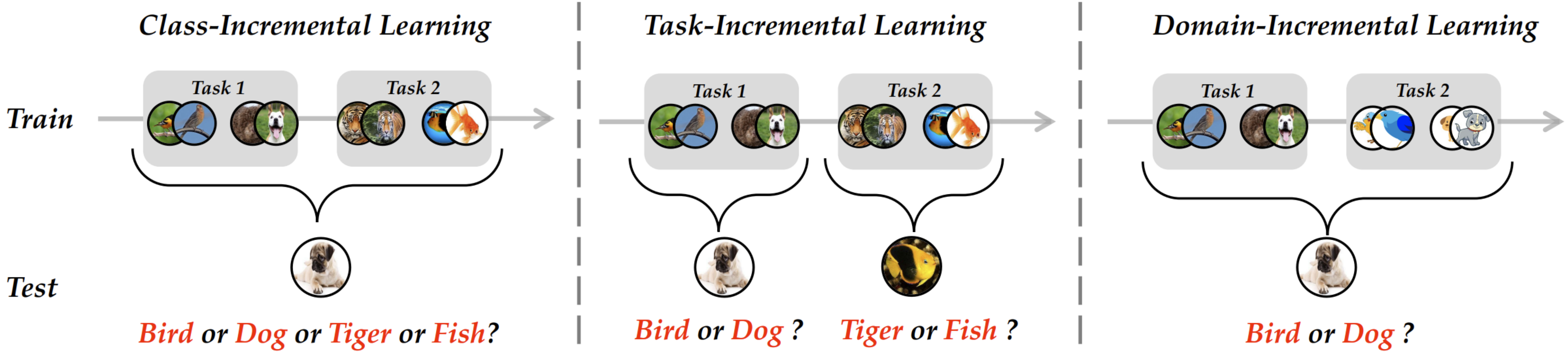
Authorized Users

Illegal Users

Face Recognition System

Incremental Learning

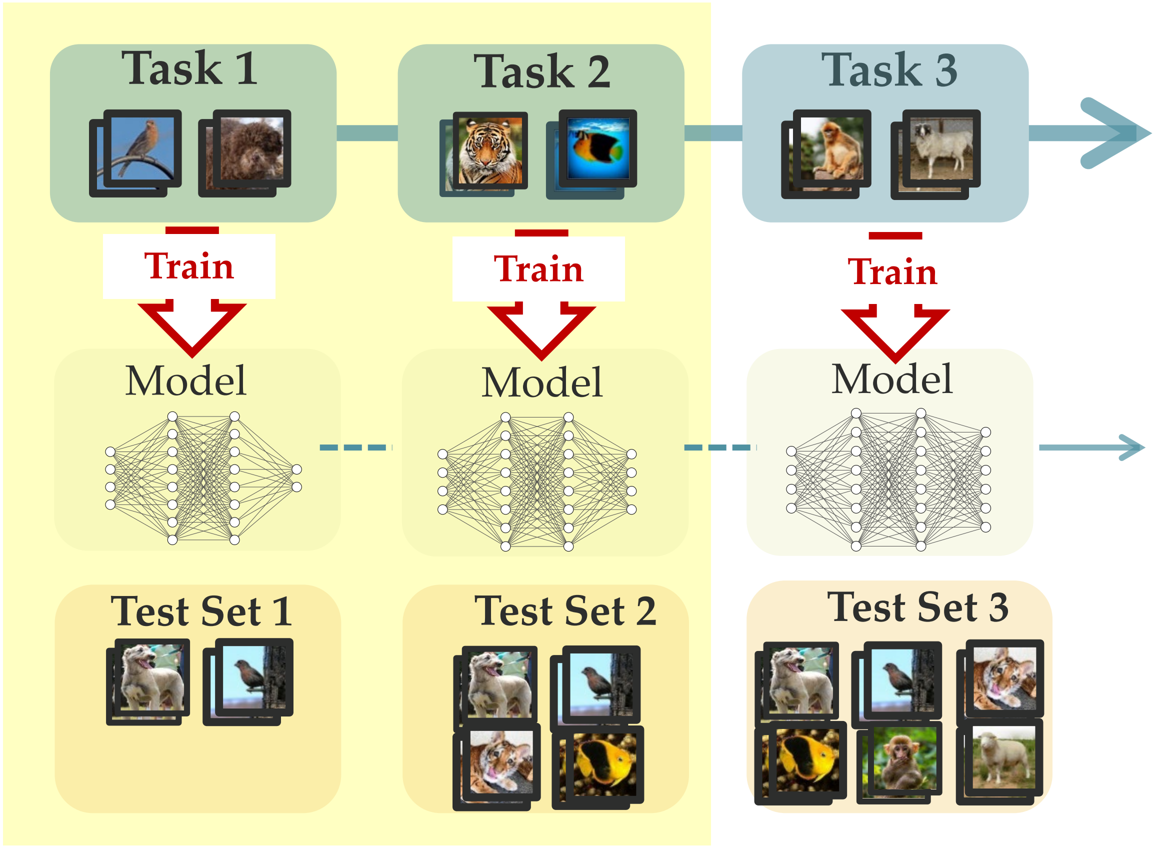
- Incremental Learning: continually adjust the model with new data

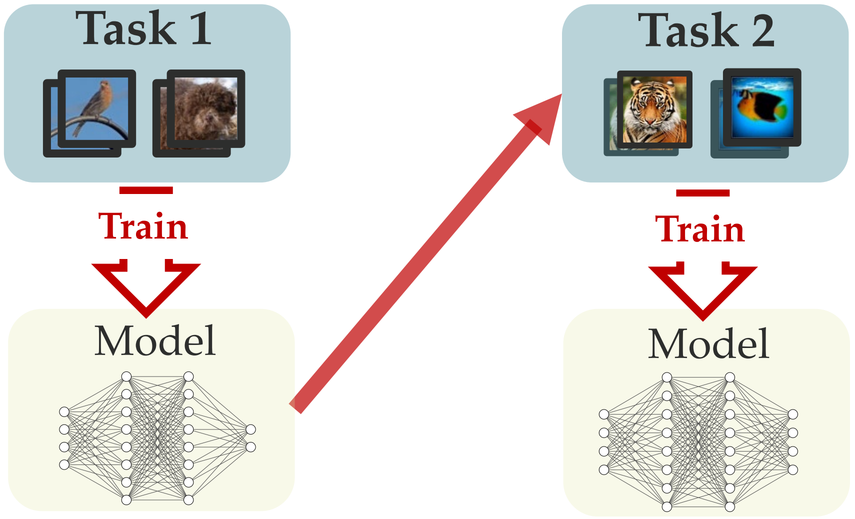


van de Ven, Tuytelaars, Tolias. *Three types of incremental learning*. NMI 2022.

Class-Incremental Learning

- Class-Incremental Learning： enable the model to tackle new classes

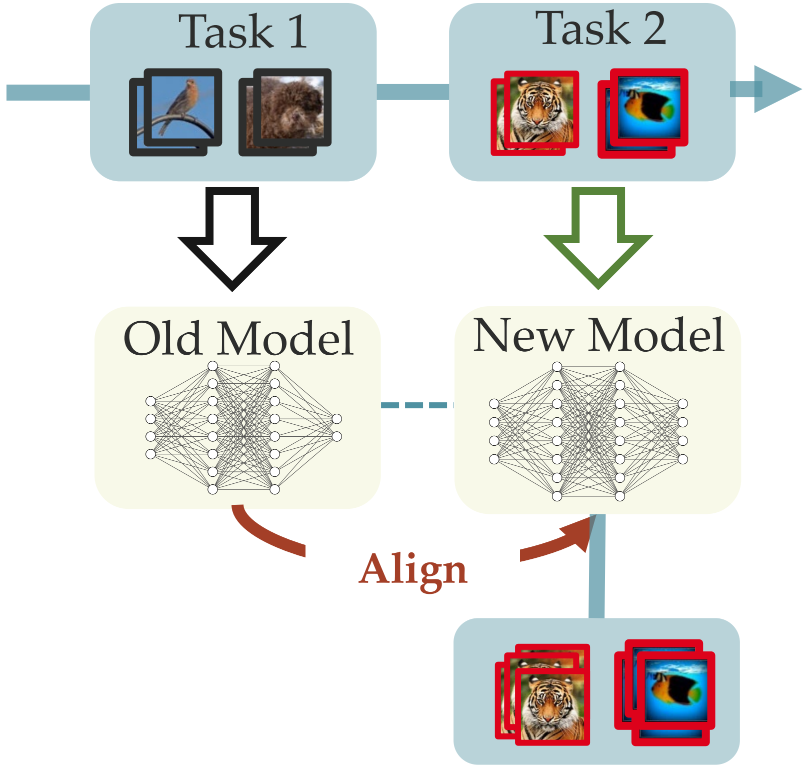




Finetune the current model with new classes?

How to expand capacity without forgetting?

- In incremental learning, we want
  - Expand model's recognition ability for new classes
  - Resist catastrophic forgetting on old classes



➢ Parameter regularization [Kirkpatrick et al. PNAS'17] [Friedemann et al. ICML'17]

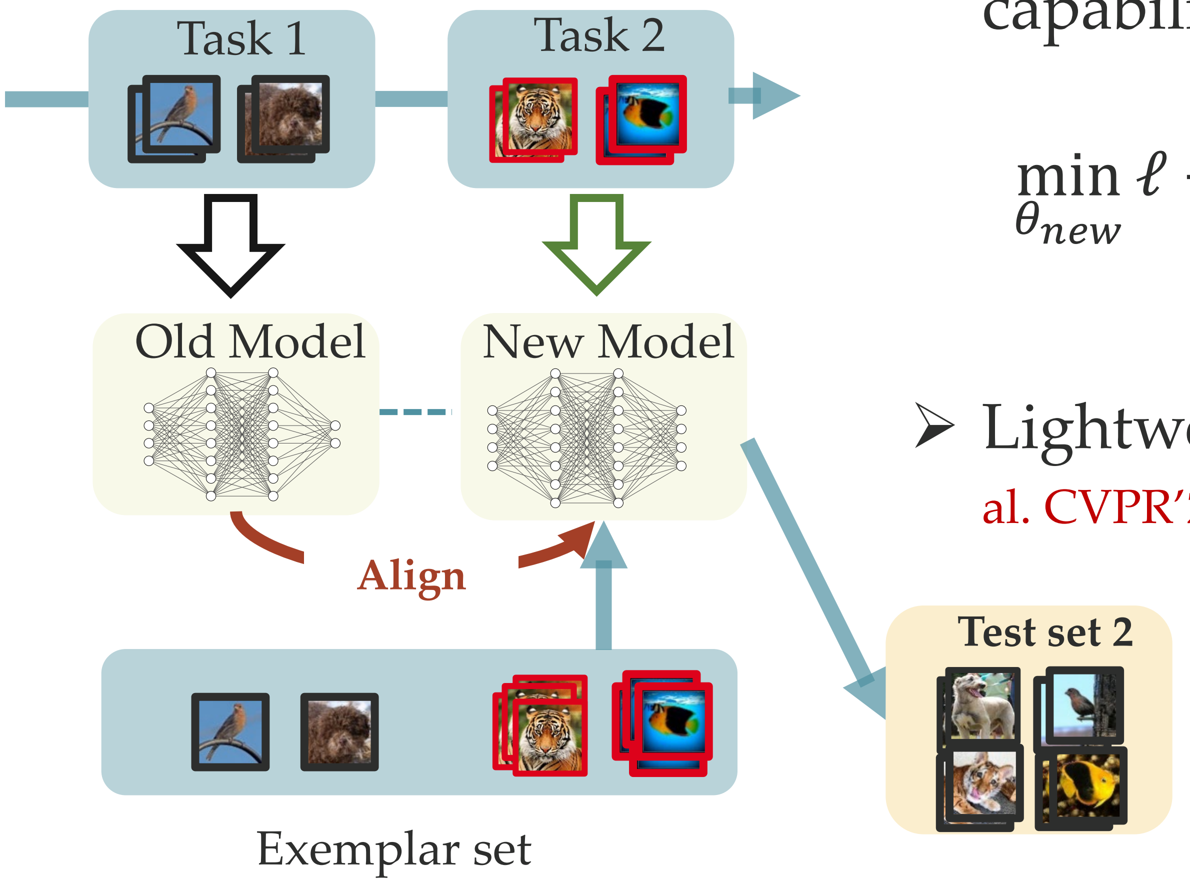
$$\min_{\theta_{new}} \ell + SIM(\boldsymbol{\theta}_{old}, \boldsymbol{\theta}_{new})$$

➢ Knowledge distillation [Li et al. TPAMI'17]

$$\min_{\theta_{new}} \ell + \sum_{\boldsymbol{x}_{new}} SIM\left(f_{\boldsymbol{\theta}_{old}}(\boldsymbol{x}_{new}), f_{\boldsymbol{\theta}_{new}}(\boldsymbol{x}_{new})\right)$$

Model cannot **balance** between old and new classes

How to expand capacity without forgetting?

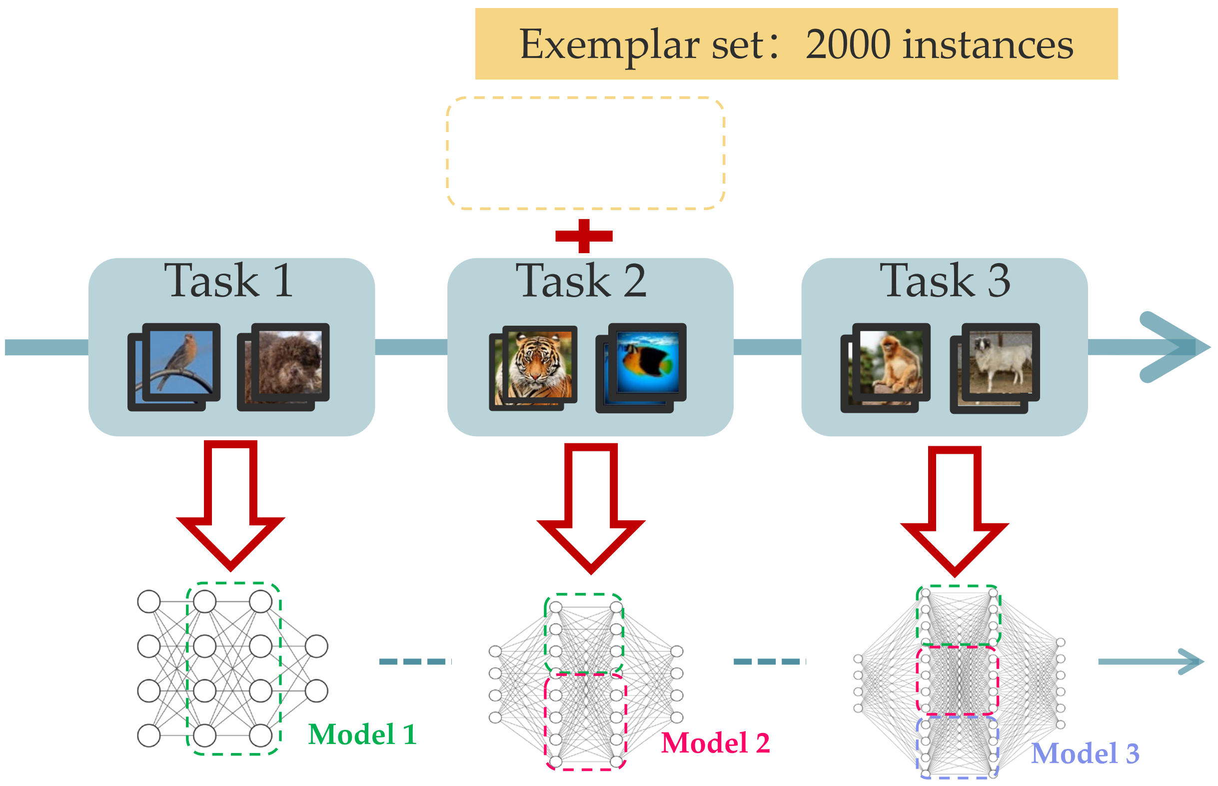


- Save limited instances to **replicate** old model's capability [Rebuffi et al. CVPR'17]

$$\min_{\theta_{new}} \ell + \sum_{x \in x_{new} \cup x_{old}} SIM\left(f_{\theta_{old}}(x), f_{\theta_{new}}(x)\right)$$

- Lightweight rectification [Wu et al. CVPR'19] [Zhao et al. CVPR'20]

Direct reuse of feature representation



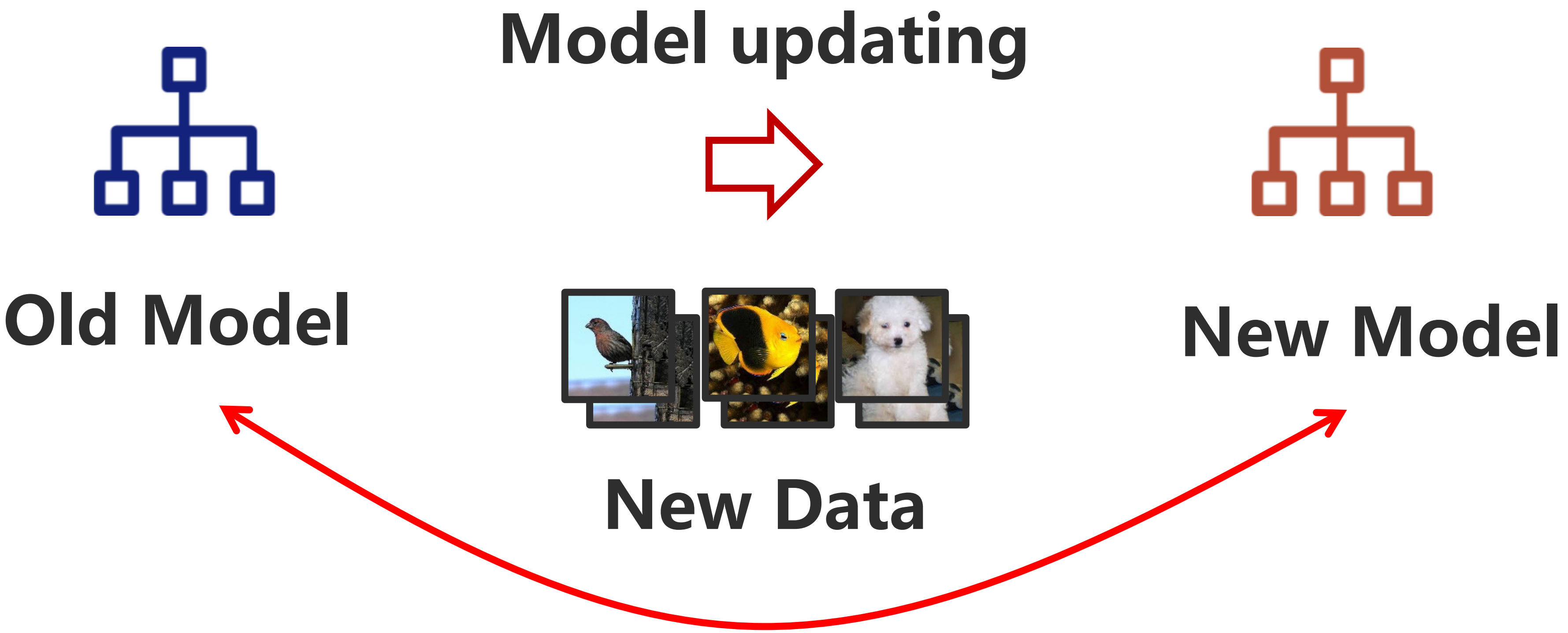
- Save model and concatenate features
  [Yan et al. CVPR'21] [Wang et al. ECCV'22]

$$f(x) = W^{\top}\text{Concat}[\phi_1(x), \phi_2(x), \ldots \phi_b(x)]$$

Save and freeze old model, only train new model.

Calibrate among multiple models via exemplar set.

Compatibility among Models



Are they "Compatible"?

It requires desirable **compatibility** for a model from closed world to open world

Two Kinds of Compatible

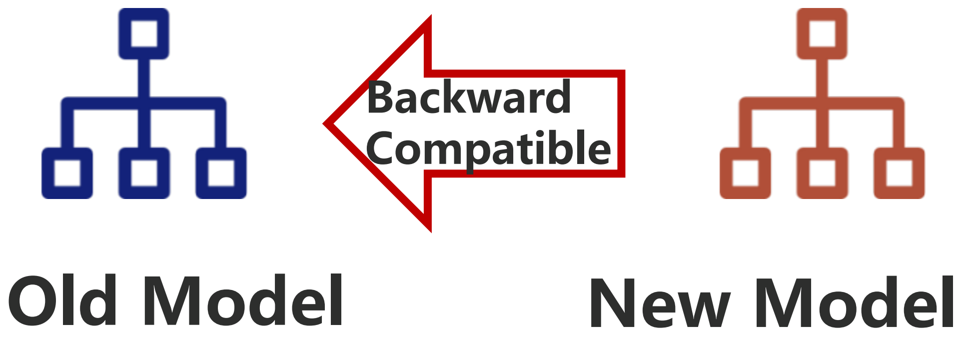
Backward Compatible

Forward Compatible

“Backward” Compatible

Backward Compatible

- Make modifications (like puting a patch) on the current model to maintain old class performance

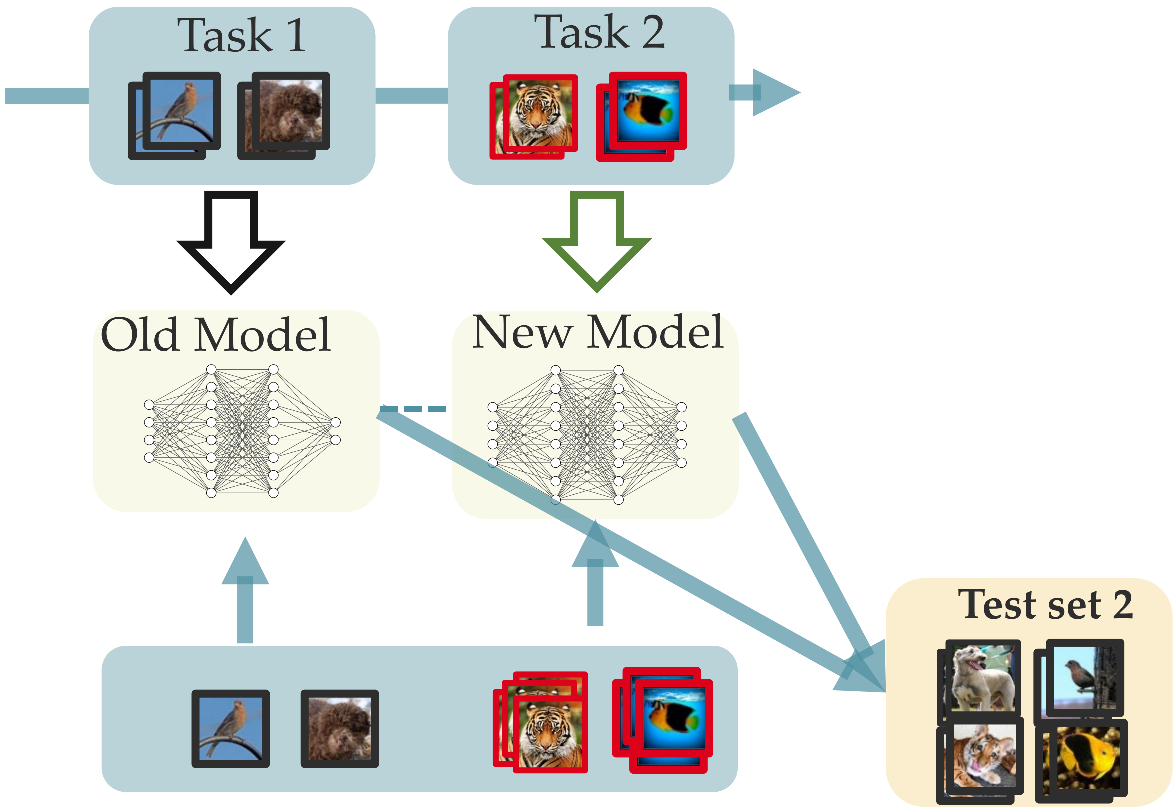


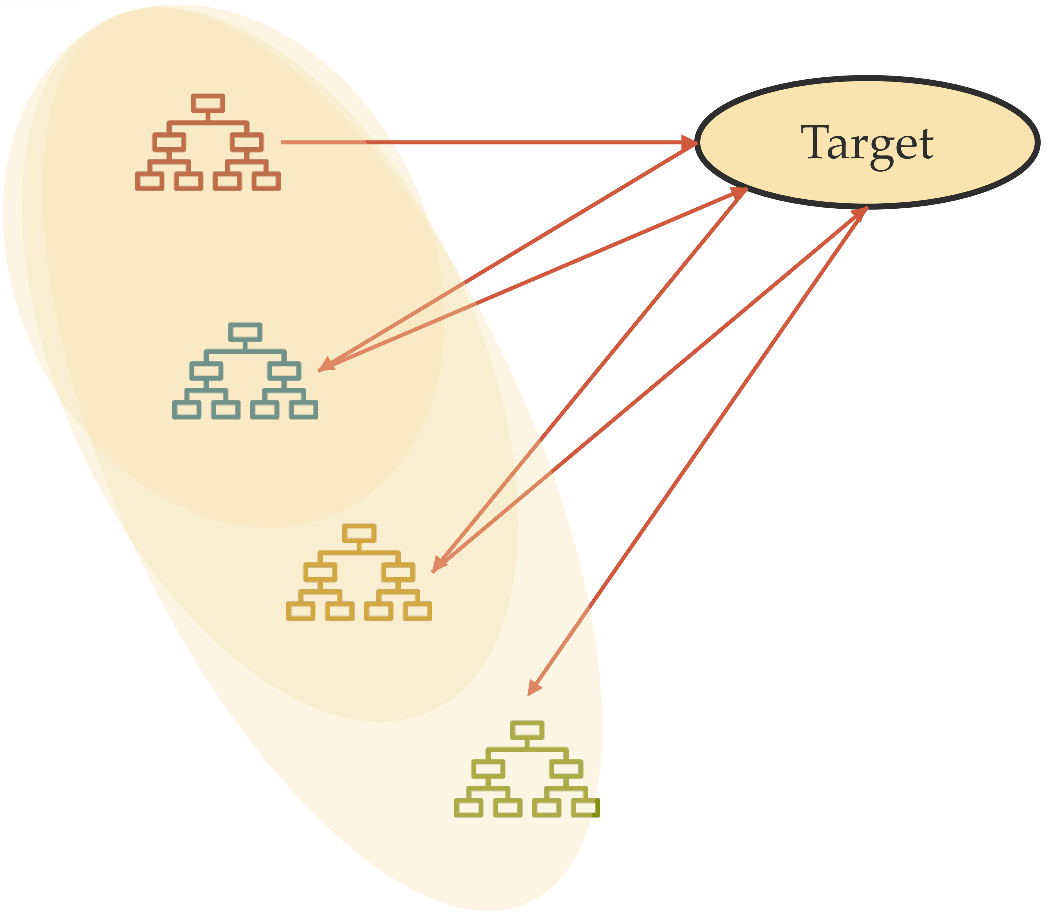
- Backward compatible with ***full*** model reuse/updates (ECCV 2022)
- Backward compatible with ***partial*** model updates (ICLR 2023)
- Backward compatible with ***few*** updates (CoRR 2023)

Incremental model boosting

➢ Save model and concatenate features

[Yan et al. CVPR'21] [Wang et al. ECCV'22]

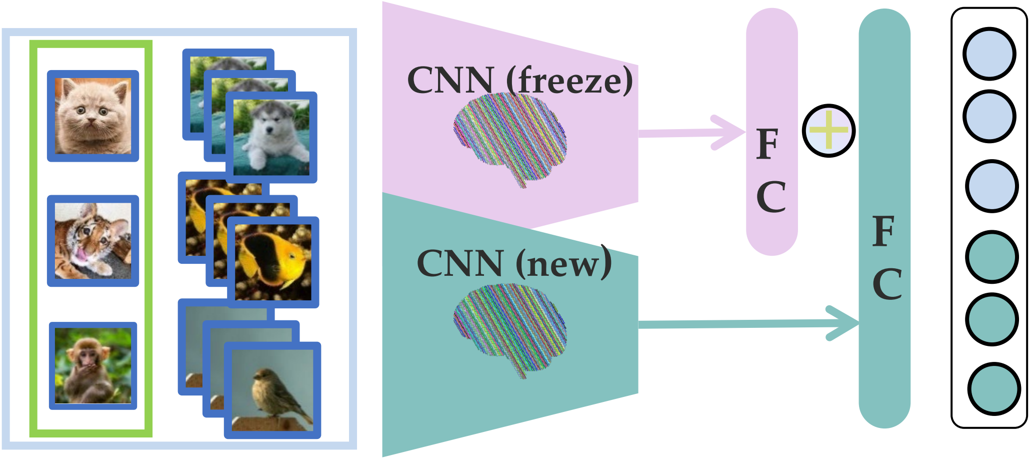




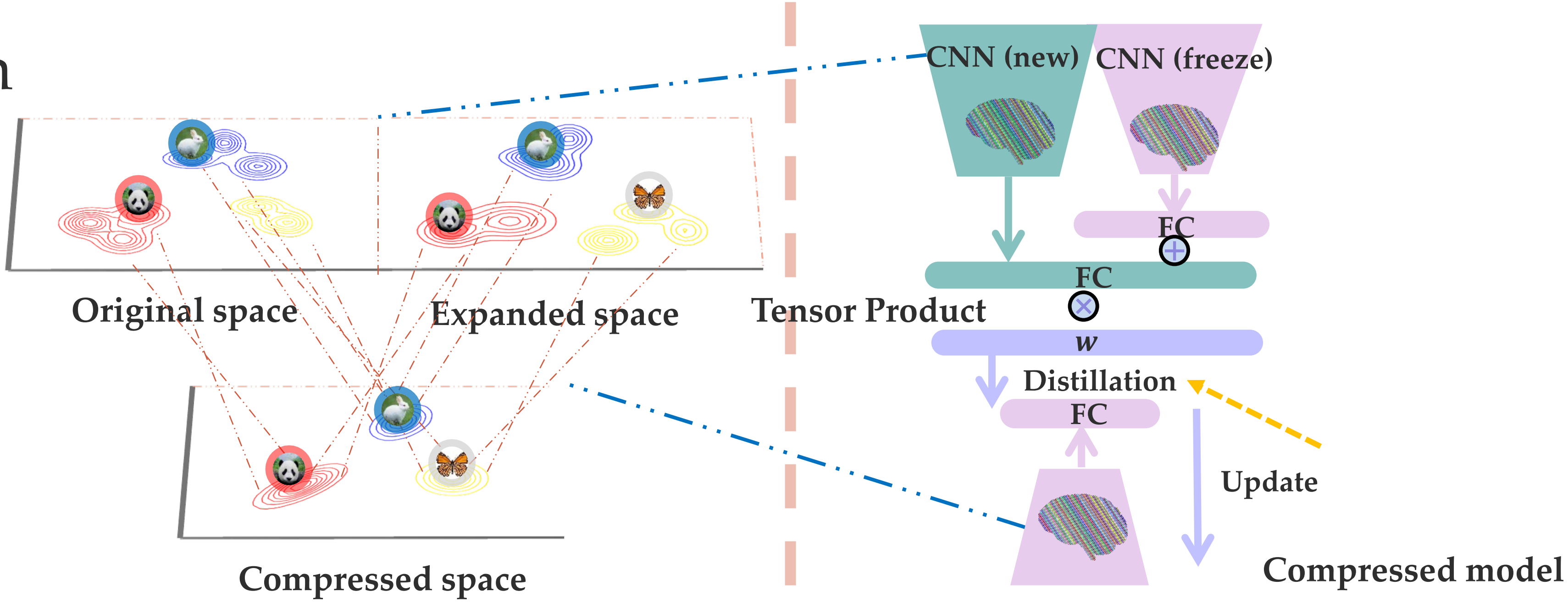
Wang, Zhou, Ye, Zhan. *FOSTER: Feature Boosting and Compression for Class-Incremental Learning*. ECCV 2022.

Full Model Reuse for Backward Compatible

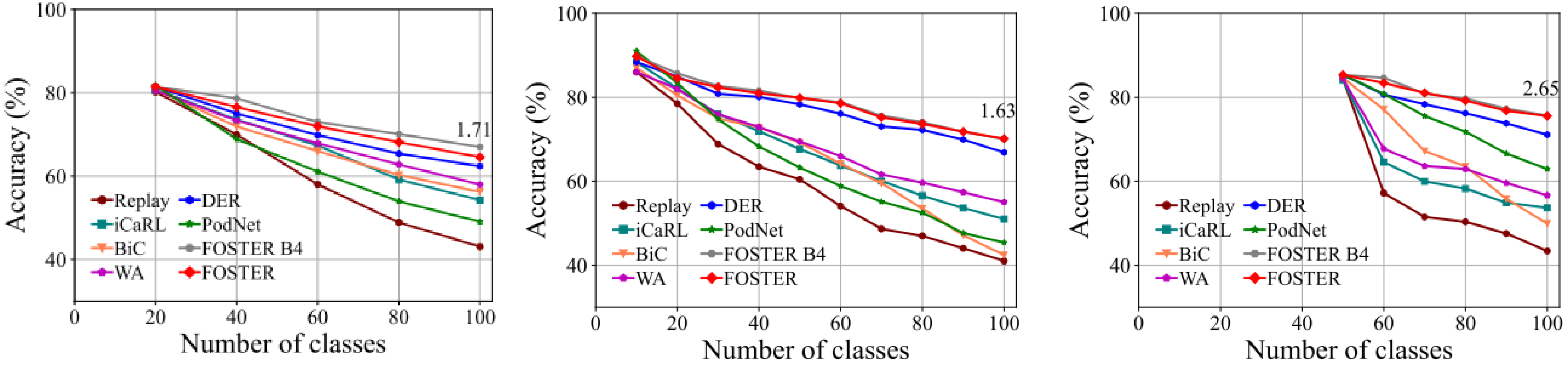
- Feature expansion



- Feature compression



Empirical evaluations



(a) CIFAR-100 B0 5 steps (b) ImageNet-100 B0 10 steps (c) ImageNet-100 B50 5 steps

Input *freeze CNN* *new CNN* *Input* *freeze CNN* *new CNN* *Input* *freeze CNN* *new CNN*

- FOSTER outperforms DER (which requires saving all historical backbones) even only using a single backbone

Challenges in feature reuse

Unfair

Exemplar-based method	2000 exemplars	BP	
Model-based method	2000 exemplars	BP * N	

Current comparison protocol

Exemplar-based method	2000 exemplars	*Δ* exemplars	BP
Replay-based method	2000 exemplars	BP * N	

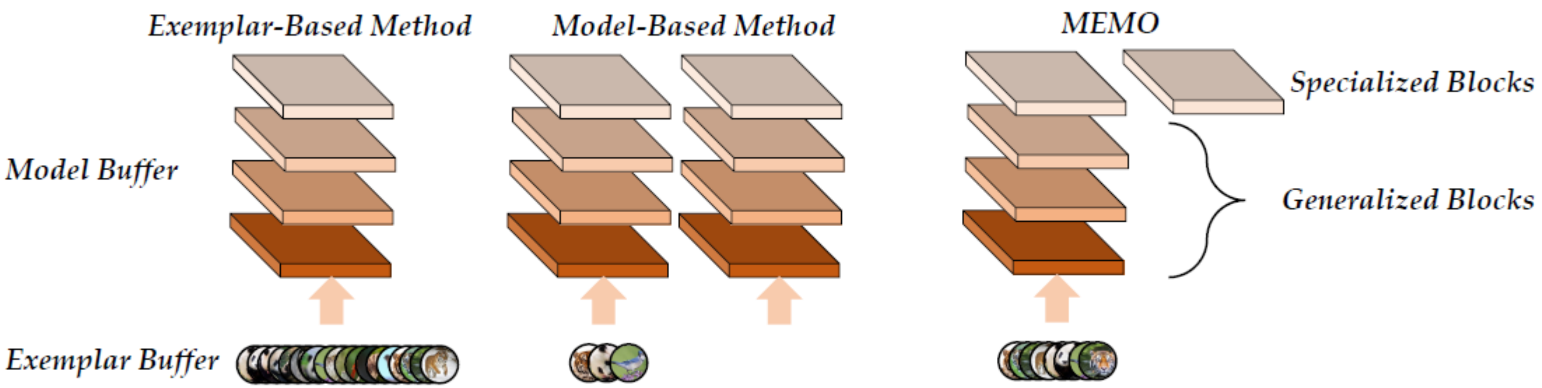
Fair protocol

ResNet32 603 CIFAR images

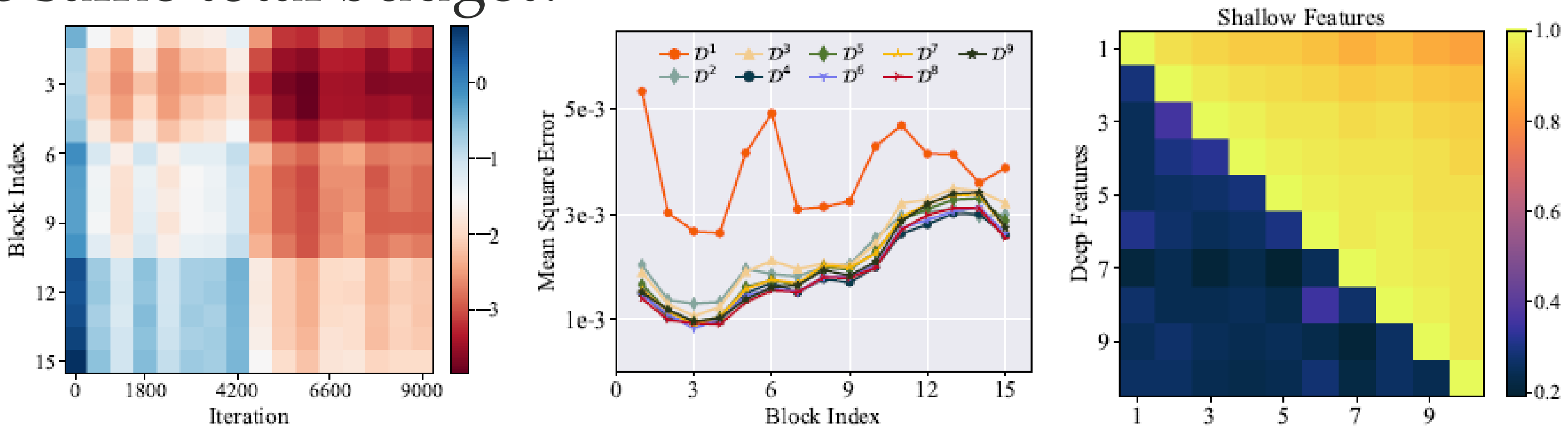
ResNet18 297 ImageNet images

How to assign memory budget for data and model to better reuse representations **given the same total budget**?

Partial model reuse for Backward Compatible



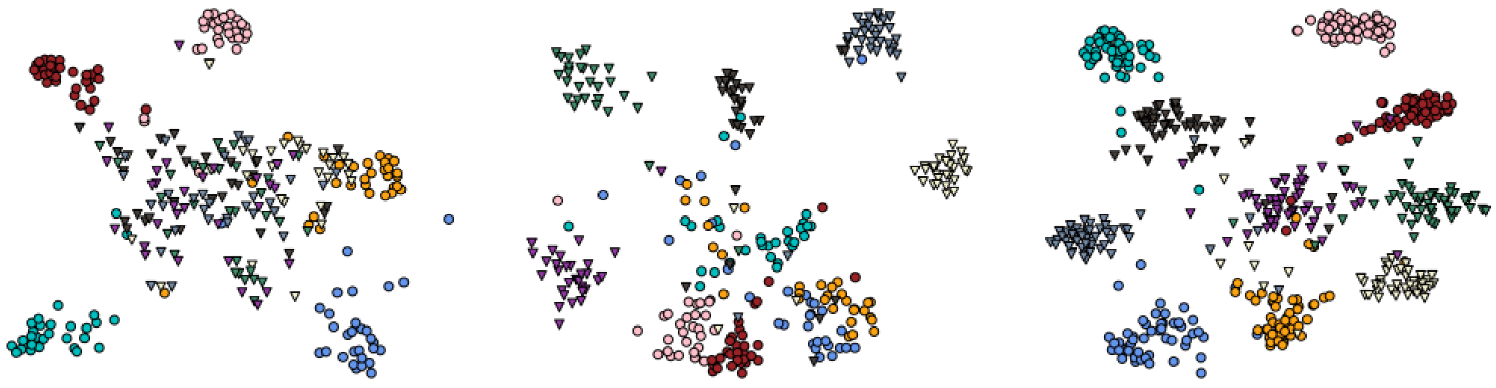
- How to assign memory budget for data and model to better reuse representations given the same total budget?



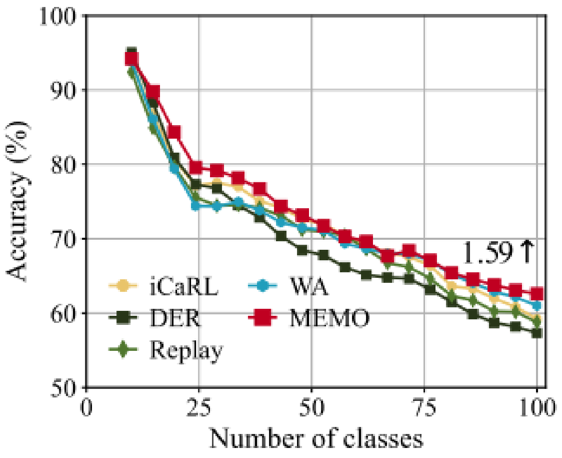
(a) Gradient norm (log scale) (b) MSE of different blocks (c) CKA between backbones

Zhou, Wang, Ye, Zhan. *A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning*. ICLR 2023.

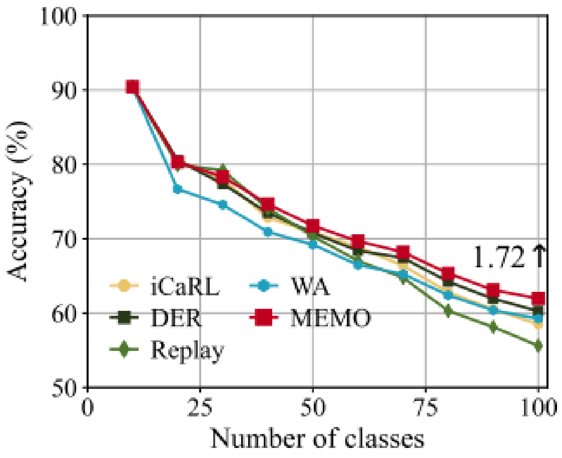
Empirical evaluations



(a) $\phi_{s1}(\phi_g(\mathbf{x}))$ (b) $\phi_{s2}(\phi_g(\mathbf{x}))$ (c) $[\phi_{s1}(\phi_g(\mathbf{x})), \phi_{s2}(\phi_g(\mathbf{x}))]$



(a) CIFAR100 Base0 Inc5



(b) CIFAR100 Base0 Inc10

- When sharing shallow features, deep features learn **task-specific representations**
- When concatenating deep features of different tasks, we obtain representations **for all tasks**
- When all algorithms are aligned to the same memory cost, our method **improves the performance for free**

Backward compatible with few updates

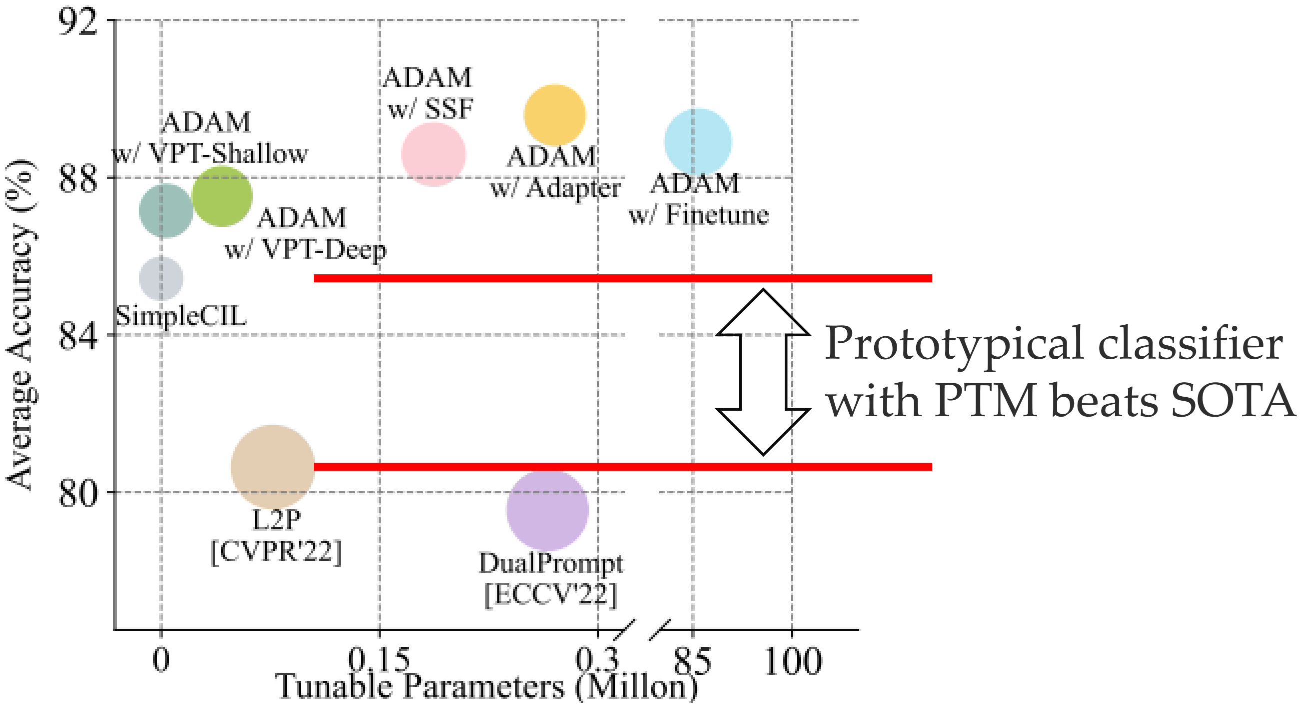
- ➢ The target of CIL is to obtain feature presentation for all tasks and resist forgetting
- ➢ Comparing to training from scratch, PTMs are born with **generalizable** features

What can PTMs bring to feature reuse?

"Training from scratch"

"Training from PTM"

$$\boldsymbol{p}_i = \frac{1}{K}\sum_{j=1}^{|\mathcal{D}^b|}\mathbb{I}(y_j = i)\phi(\mathbf{x}_j)$$



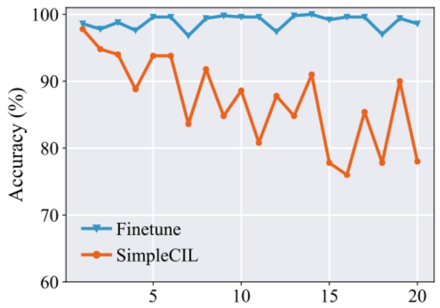
Do PTMs need incremental learning?

Zhou, Ye, Zhan, Liu. *Revisiting Class-Incremental Learning with Pre-Trained Models: Generalizability and Adaptivity are All You Need*. CoRR 2023.

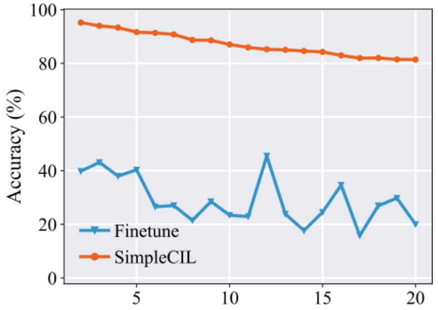
Backward compatible with few updates

Is (PTM + Prototypical Classifier) enough for any incremental learning task?

No! Adapting the model with downstream task can further enhance model's performance

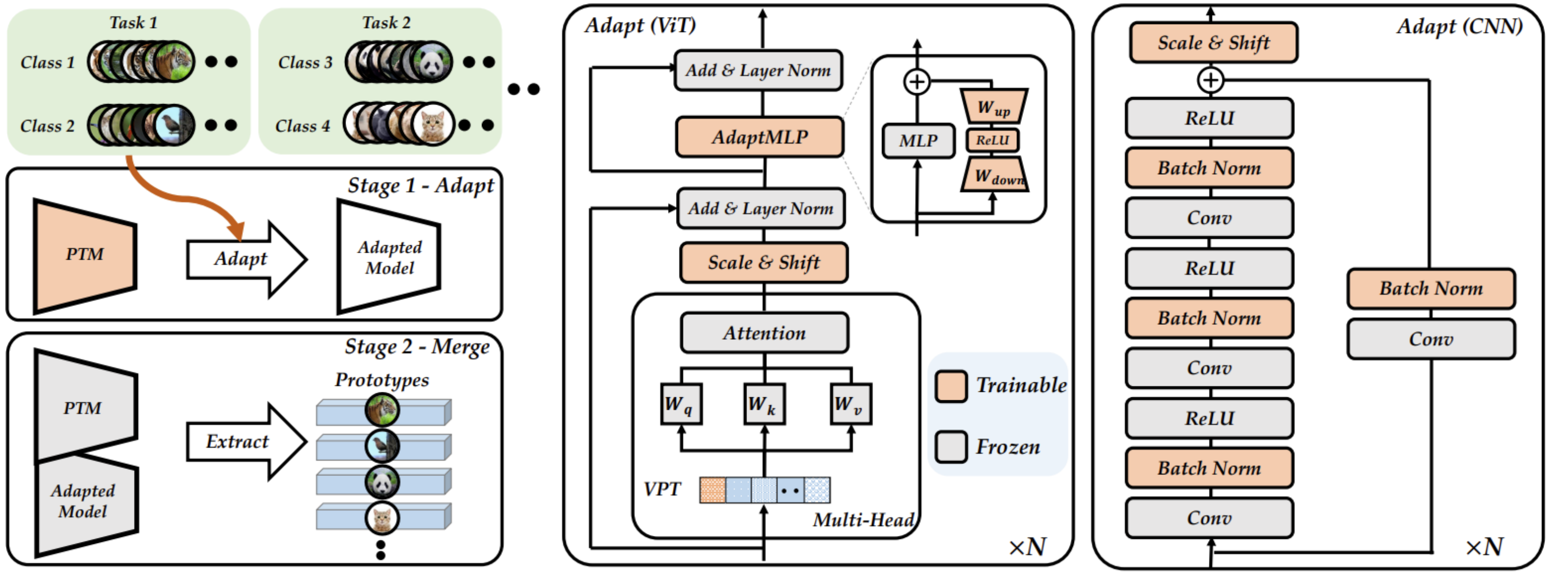


New class ACC



Old class ACC

How to combine PTM and adapted model's advantages?

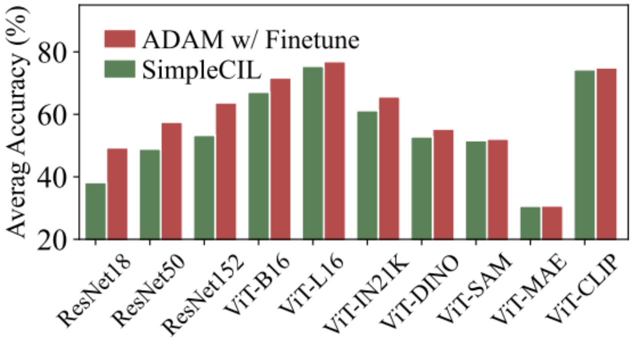


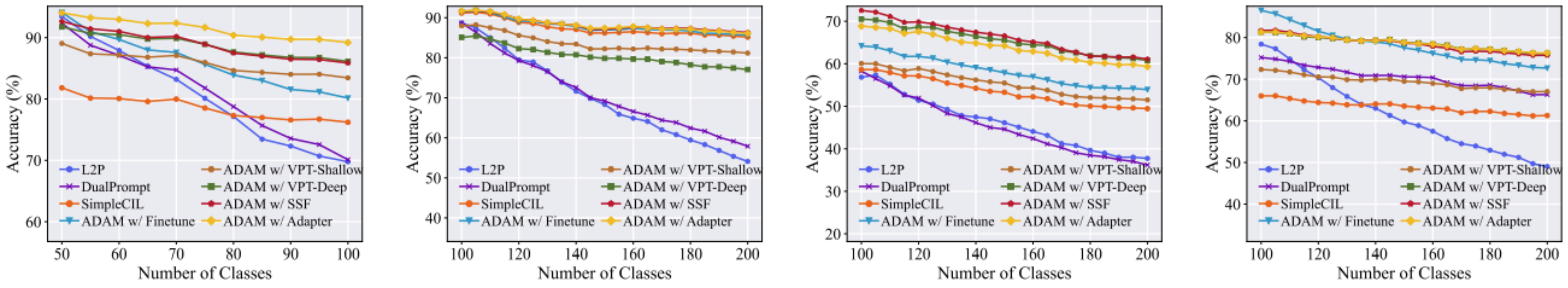
First stage: model adaptation and merge

Latter stages: prototypical classifier

Backward compatible with few updates

Method	CIFAR B0 Inc5 $\bar{\mathcal{A}}$	CIFAR B0 Inc5 $\mathcal{A}_B$	CUB B0 Inc10 $\bar{\mathcal{A}}$	CUB B0 Inc10 $\mathcal{A}_B$	IN-R B0 Inc5 $\bar{\mathcal{A}}$	IN-R B0 Inc5 $\mathcal{A}_B$	IN-A B0 Inc10 $\bar{\mathcal{A}}$	IN-A B0 Inc10 $\mathcal{A}_B$	ObjNet B0 Inc10 $\bar{\mathcal{A}}$	ObjNet B0 Inc10 $\mathcal{A}_B$	OmniBench B0 Inc30 $\bar{\mathcal{A}}$	OmniBench B0 Inc30 $\mathcal{A}_B$	VTAB B0 Inc10 $\bar{\mathcal{A}}$	VTAB B0 Inc10 $\mathcal{A}_B$
Finetune	38.90	20.17	26.08	13.96	21.61	10.79	21.60	10.96	19.14	8.73	23.61	10.57	34.95	21.25
Finetune Adapter [10]	60.51	49.32	66.84	52.99	47.59	40.28	43.05	37.66	50.22	35.95	62.32	50.53	48.91	45.12
LwF [38]	46.29	41.07	48.97	32.03	39.93	26.47	35.39	23.83	33.01	20.65	47.14	33.95	40.48	27.54
L2P [72]	85.94	79.93	67.05	56.25	66.53	59.22	47.16	38.48	63.78	52.19	73.36	64.69	77.11	77.10
DualPrompt [71]	87.87	81.15	77.47	66.54	63.31	55.22	52.56	42.68	59.27	49.33	73.92	65.52	83.36	81.23
SimpleCIL	87.57	81.26	92.20	**86.73**	62.58	54.55	60.50	49.44	65.45	53.59	79.34	73.15	85.99	84.38
ADAM w/ Finetune	87.67	81.27	91.82	86.39	70.51	62.42	61.57	50.76	61.41	48.34	73.02	65.03	**87.47**	80.44
ADAM w/ VPT-Shallow	90.43	84.57	92.02	86.51	66.63	58.32	57.72	46.15	64.54	52.53	79.63	73.68	87.15	**85.36**
ADAM w/ VPT-Deep	88.46	82.17	91.02	84.99	68.79	60.48	60.59	48.72	67.83	54.65	**81.05**	**74.47**	86.59	83.06
ADAM w/ SSF	87.78	81.98	91.72	86.13	68.94	60.60	**62.81**	**51.48**	**69.15**	**56.64**	80.53	74.00	85.66	81.92
ADAM w/ Adapter	**90.65**	**85.15**	**92.21**	**86.73**	**72.35**	**64.33**	60.53	49.57	67.18	55.24	80.75	74.37	85.95	84.35



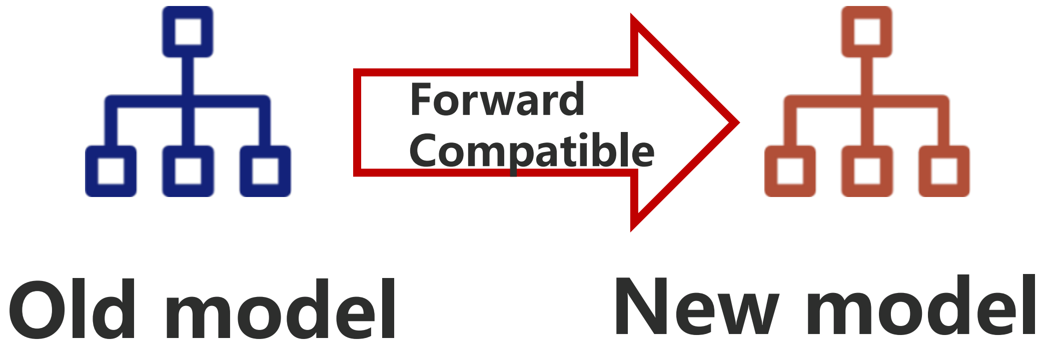


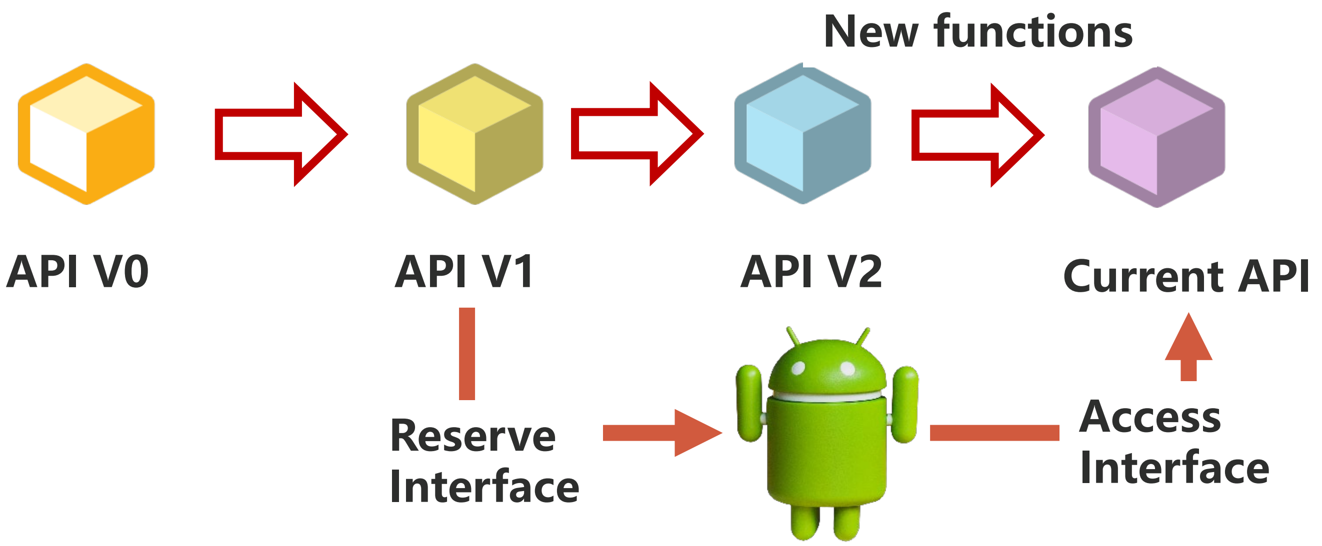
(a) CIFAR B50 Inc5 (b) CUB B100 Inc5 (c) ImageNet-A B100 Inc5 (d) ImageNet-R B100 Inc5

- Outperforms SOTA on 7 benchmark datasets and various settings
- Show substantial improvement on various pre-trained backbones

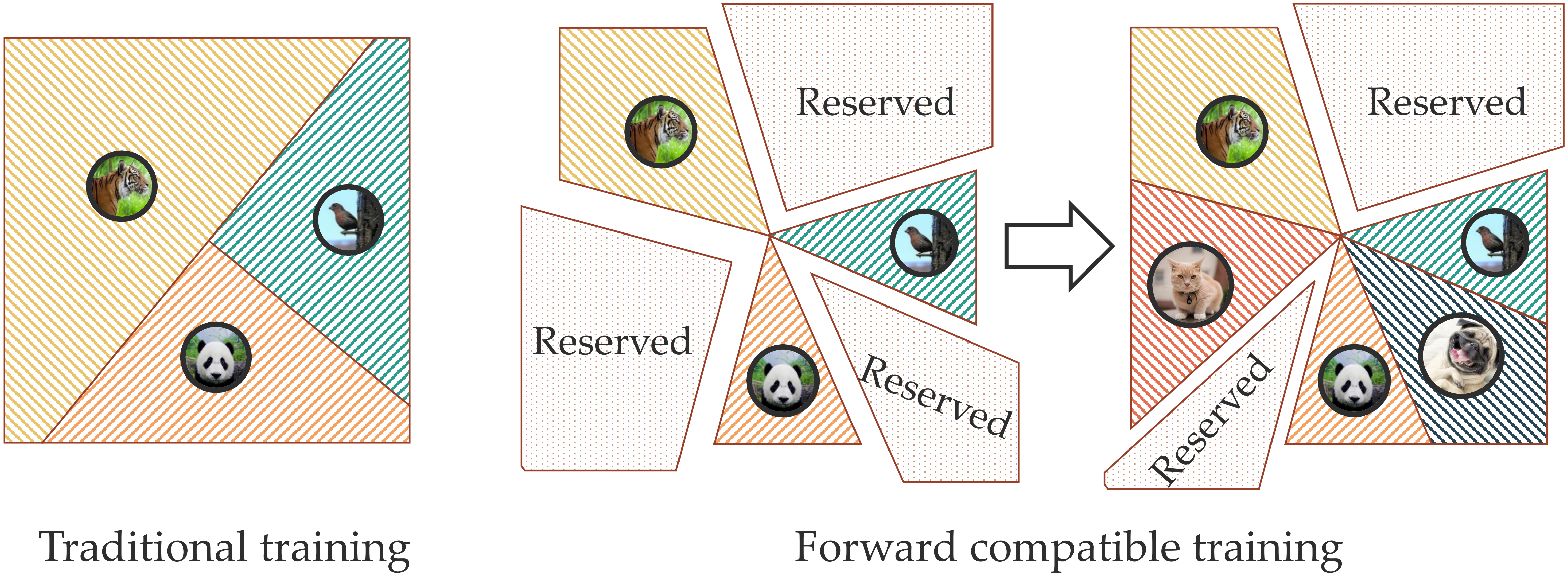
Forward Compatible

- Forward compatible
  - Reserve *interface* for future possible characteristics during the current training process





Reserve embedding space for new classes



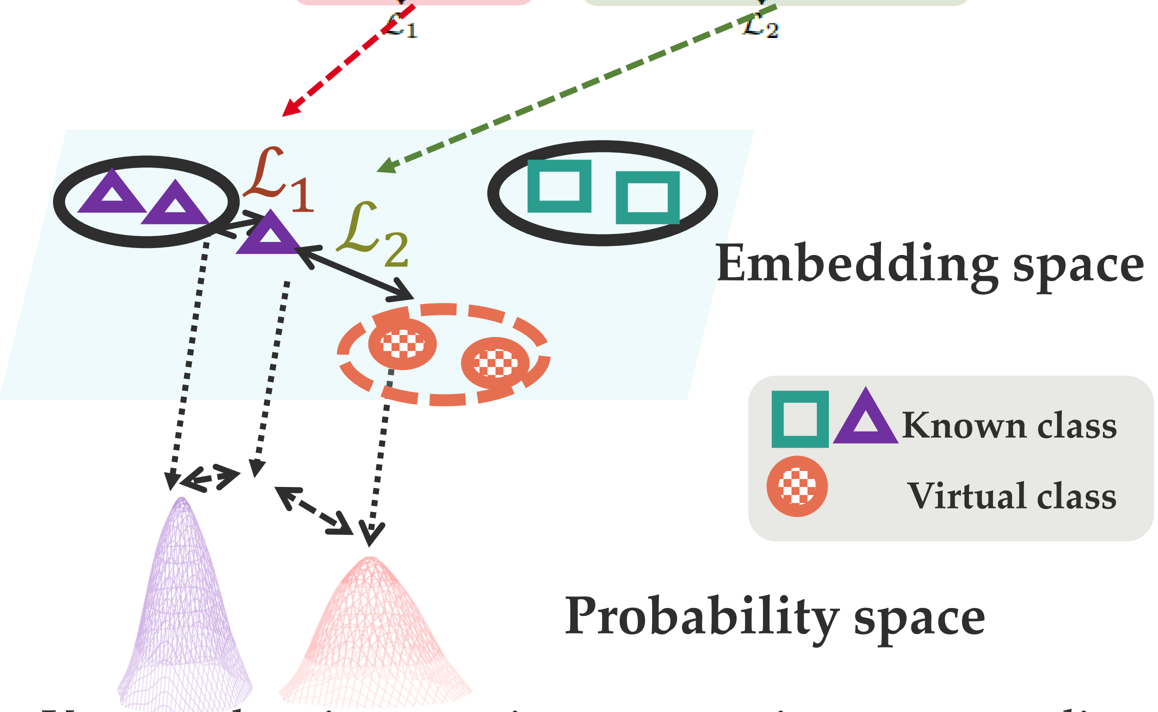
Zhou, Wang, Ye, Ma, Pu, Zhan. *Forward compatible few-shot class-incremental learning*. CVPR 2022

Forward compatible for few-shot CIL

- Core idea: reserve embedding space for new classes

Objective on known classes

$$\mathcal{L}_v(\mathbf{x},y) = \underbrace{\ell(f_v(\mathbf{x}),y)}_{\mathcal{L}_1} + \gamma\, \underbrace{\ell\left(\mathrm{Mask}(f_v(\mathbf{x}),y),\hat{y}\right)}_{\mathcal{L}_2}$$

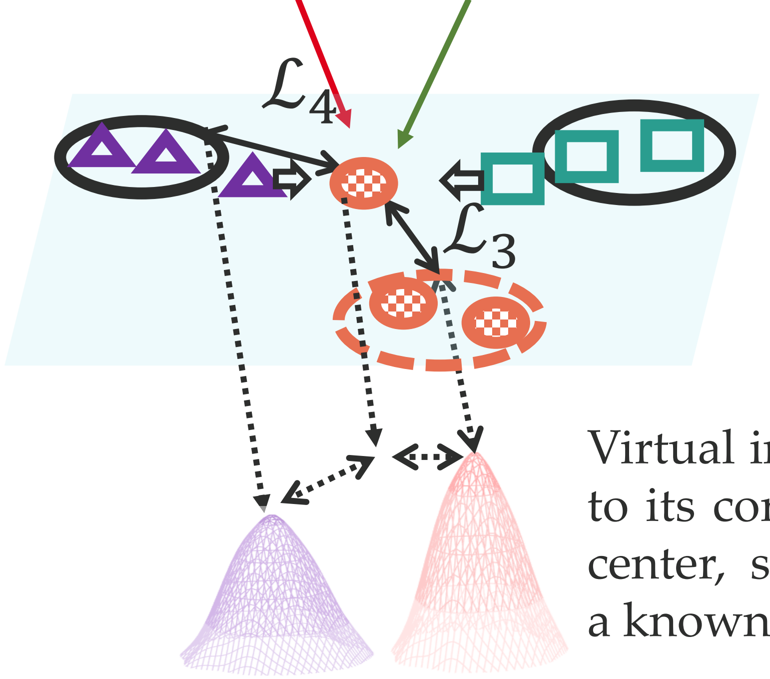


Known class instance is nearest to its corresponding class center, second nearest to a virtual class center

Objective on virtual classes

$$\mathcal{L}_f(\mathbf{z}) = \underbrace{\ell(f_v(\mathbf{z}),\hat{y})}_{\mathcal{L}_3} + \gamma\, \underbrace{\ell\left(\mathrm{Mask}(f_v(\mathbf{z}),\hat{y}),\hat{\hat{y}}\right)}_{\mathcal{L}_4},$$

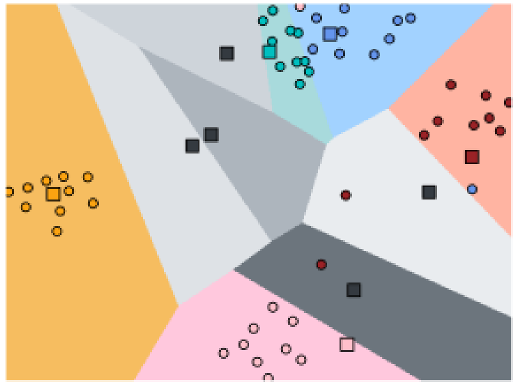
$$\mathbf{z} = g\left[\lambda h(\mathbf{x}_i) + (1-\lambda)h(\mathbf{x}_j)\right],$$



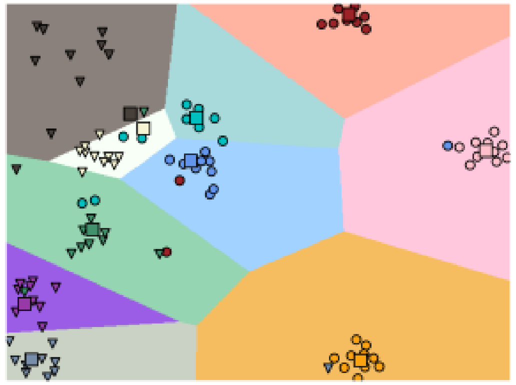
Virtual instance is nearest to its corresponding class center, second nearest to a known class center

Empirical evaluations

Method	Accuracy in each session (%) ↑											PD ↓	Δ PD
	0	1	2	3	4	5	6	7	8	9	10		
Finetune	68.68	43.70	25.05	17.72	18.08	16.95	15.10	10.06	8.93	8.93	8.47	60.21	**+41.25**
Pre-Allocated RPC† [32]	68.47	51.00	45.42	40.76	35.90	33.18	27.23	24.24	21.18	17.34	16.20	52.27	**+33.31**
iCaRL [33]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	47.52	**+28.56**
EEIL [8]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	46.57	**+27.61**
Rebalancing [21]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	48.81	**+29.85**
TOPIC [41]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.26	42.40	**+23.44**
SPPR [67]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	31.35	**+12.39**
Decoupled-NegCosine† [26]	74.96	70.57	66.62	61.32	60.09	56.06	55.03	52.78	51.50	50.08	48.47	26.49	**+7.53**
Decoupled-Cosine [45]	75.52	70.95	66.46	61.20	60.86	56.88	55.40	53.49	51.94	50.93	49.31	26.21	**+7.25**
Decoupled-DeepEMD [57]	75.35	70.69	66.68	62.34	59.76	56.54	54.61	52.52	50.73	49.20	47.60	27.75	**+8.79**
CEC [58]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	23.57	**+4.61**
FACT	**75.90**	**73.23**	**70.84**	**66.13**	**65.56**	**62.15**	**61.74**	**59.83**	**58.41**	**57.89**	**56.94**	**18.96**	



(a) Base session, 5 old classes & 5 virtual prototypes.

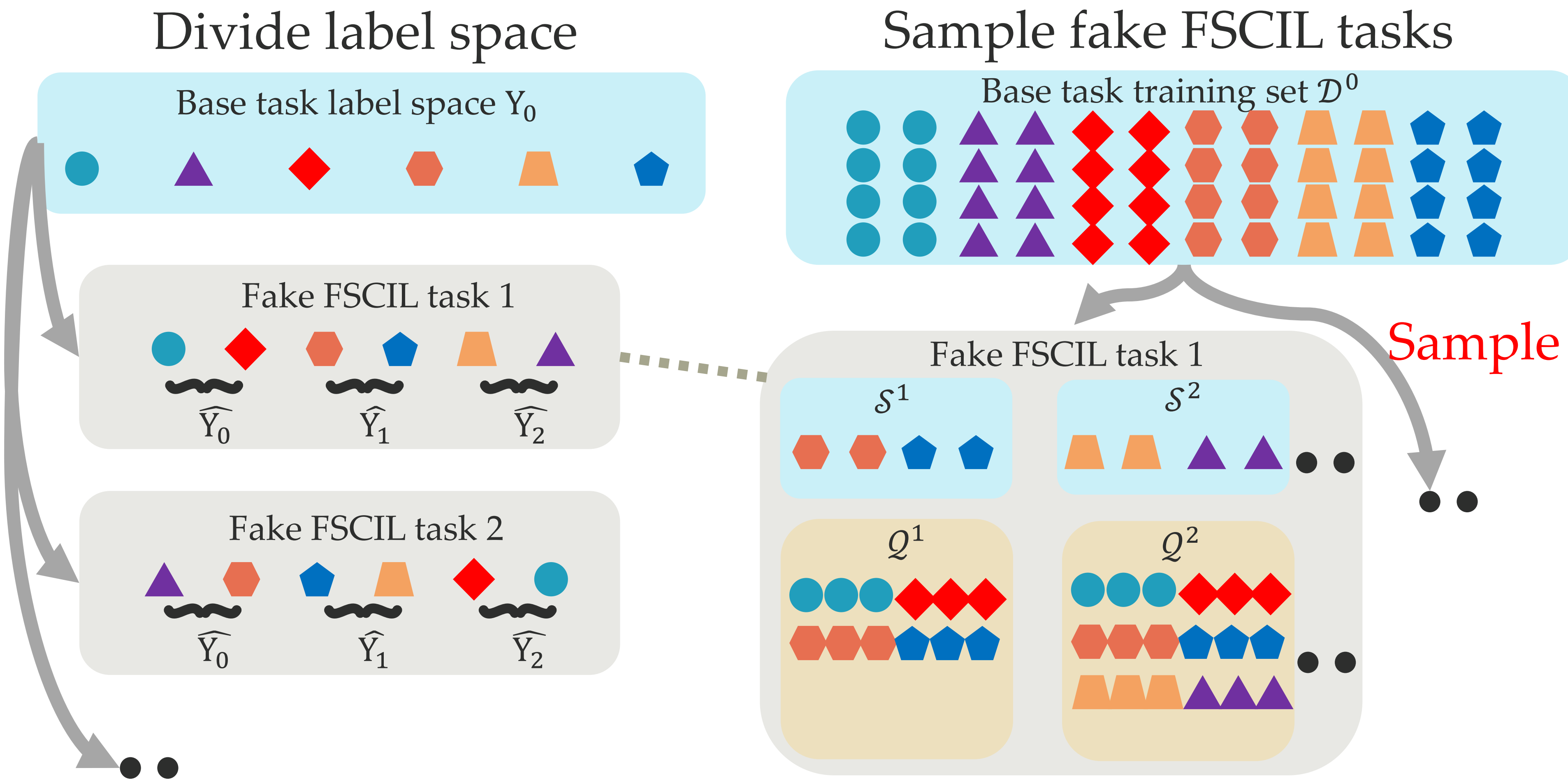


(b) Incremental session, 5 old classes & 5 new classes.

- On CUB200 (100 base classes, 10-way-5-shot setting), FACT outperforms SOTA by 4.5%
- **Reserved** embedding space (dark) helps the learning of new classes

Forward compatible for few-shot CIL

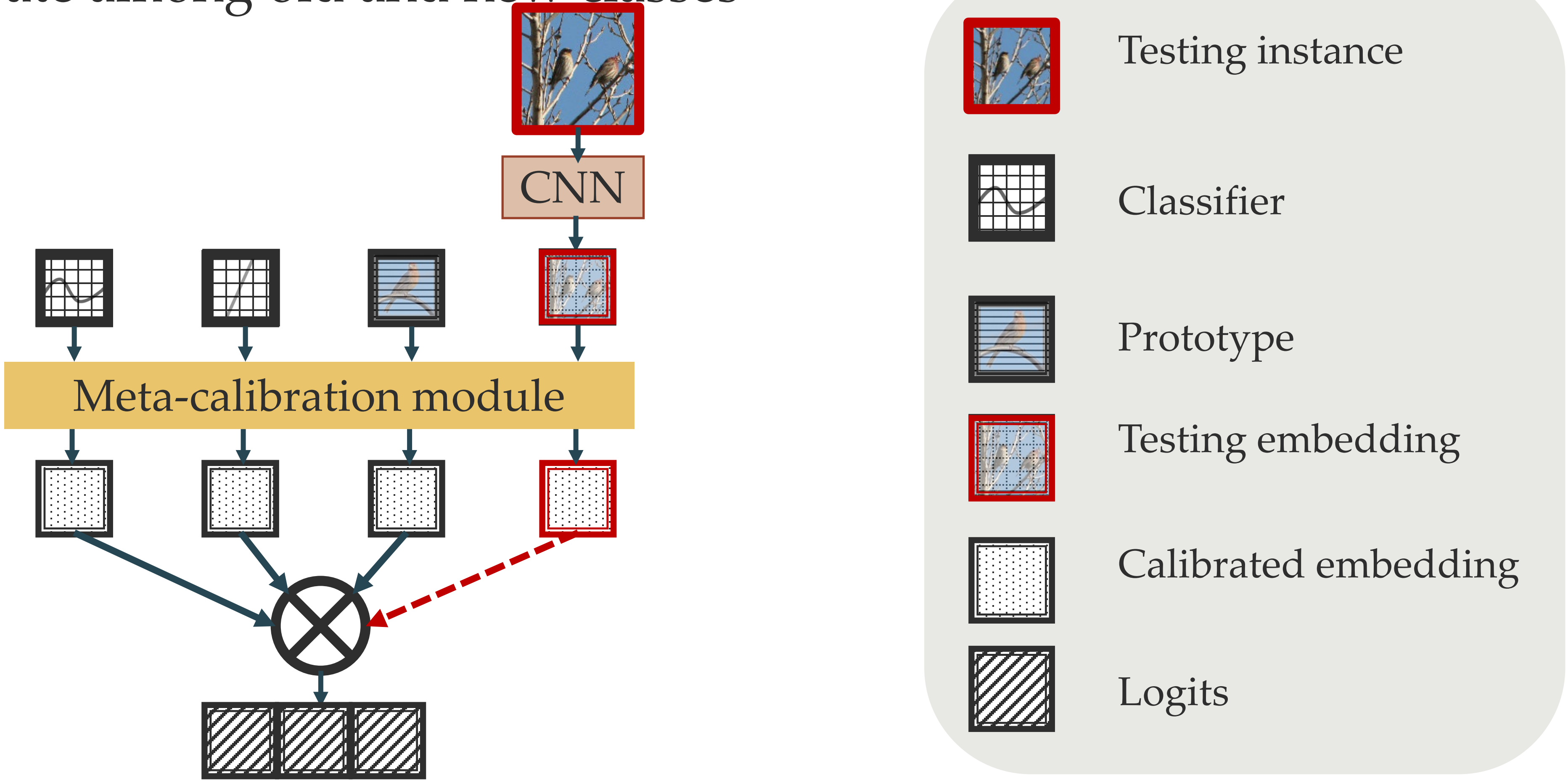
- Sample few-shot CIL meta-tasks



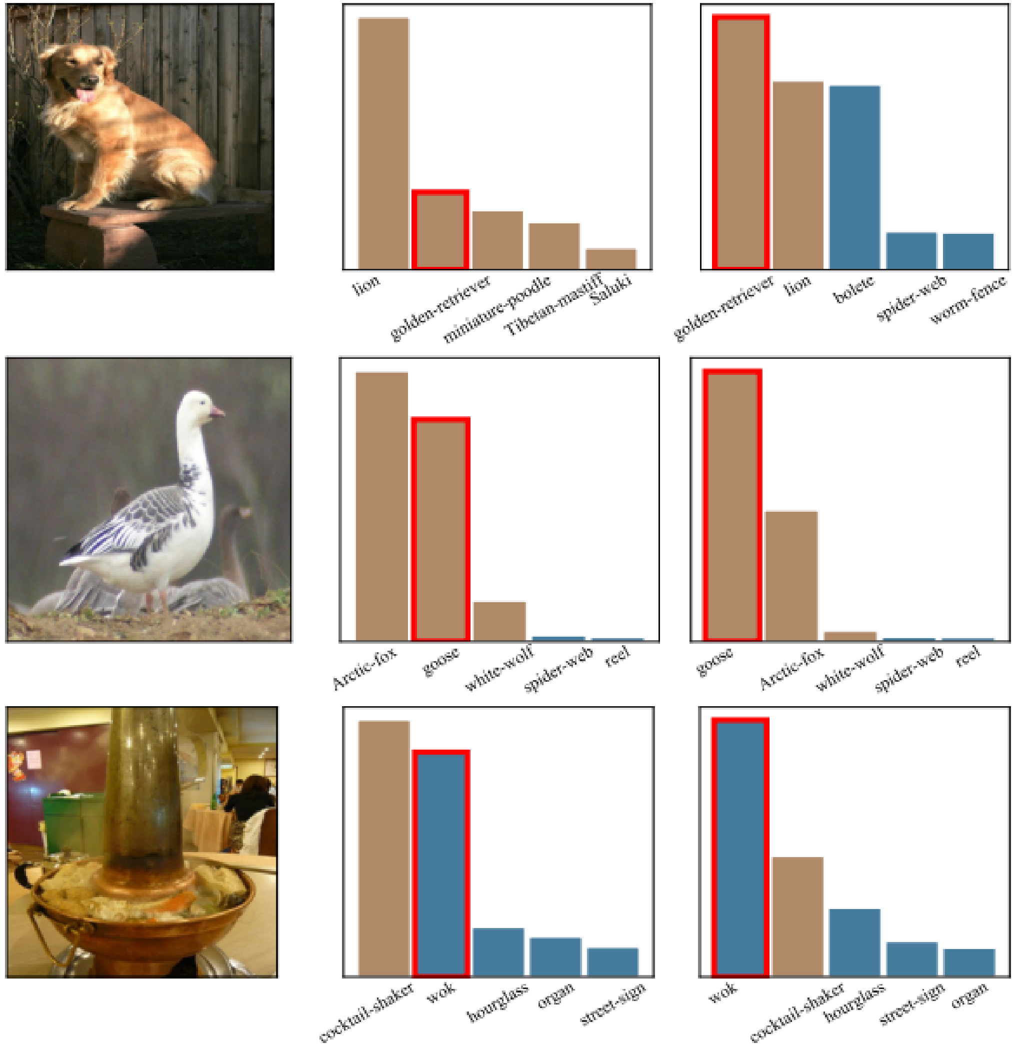
Zhou, Ye, Ma, Xie, Pu, Zhan. *Few-Shot Class-Incremental Learning by Sampling Multi-Phase Tasks*. TPAMI 2023.

Forward compatible for few-shot CIL

- Train calibration module via meta-learning
  - Learn to calibrate among old and new classes



Empirical evaluations

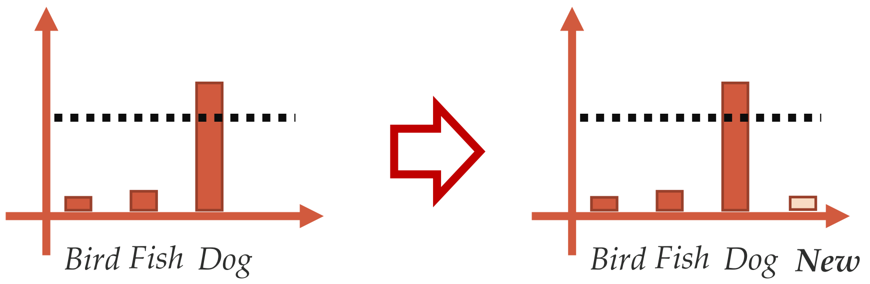


Methods	Base	Incremental	Harmonic Mean
Decoupled-Cosine	71.5	28.8	41.1
CEC	71.1	33.9	45.9
LIMIT	**73.6**	**41.8**	**53.3**

- Calibration module helps obtain instance-specific embedding and adapt the logits, which **rectifies the wrong predictions** of the model
- Since new classes are limited, model tends to predict new classes into seen classes, the calibration module can improve new class performance adaptively

Applications of compatibility

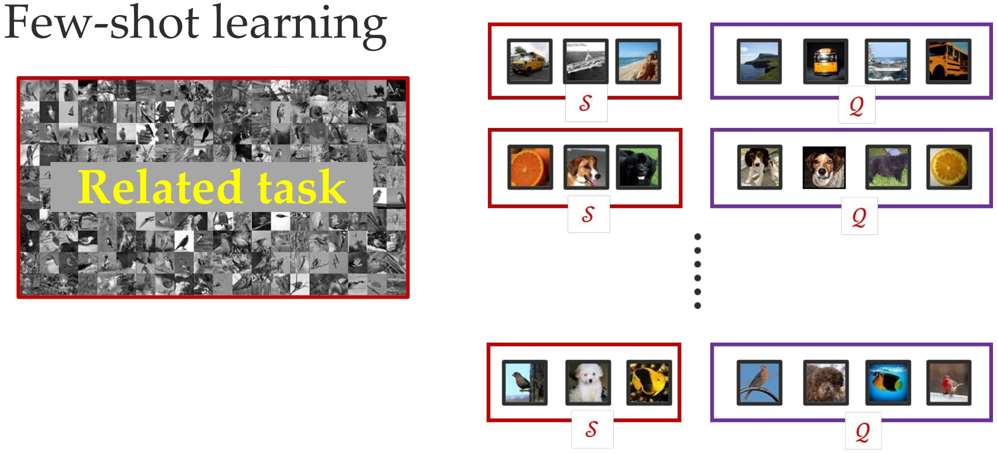
Setting threshold for long-tailed learning



Classifier re-scaling [50, 20]	$h_i = (W_i / n_i^{\tau})^T f(x)$
Classifier normalization [17]	$h_i = (W_i / \|W_i\|^{\tau})^T f(x)$
Class-aware bias [28]	$h_i = W_i^T f(x) - \tau \log(n_i)$
Feature disentangling [40]	$h_i = W_i^T (f(x) - \alpha \cos(f(x), d) \cdot d)$

Balancing old and new classes in imbalanced/FSL via classifier calibration

Few-shot learning



Summary

Incremental Learning

Using Transformer [Douillard et al. CVPR'22]

Tackles catastrophic forgetting
Rely on PTM or specific network structure

Feature concatenation [Yan et al. CVPR'21]

Model's memory cost increasers
as task number evolves

Knowledge distillation [Li et al. TPAMI'17]

Regularize performs on new tasks,
Shift the burden from old model to new model

Parameter regularization [Kirkpatrick et al. PNAS'17]

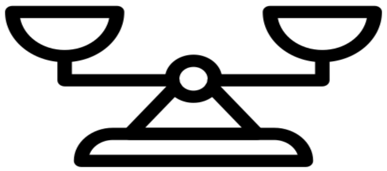
Parameter importance differs from task to
task, even being contradictory

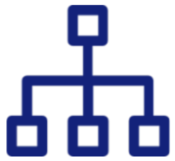
Thanks

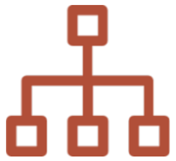
- Making modifications among models **compatible**

Backward compatible

Forward compatible



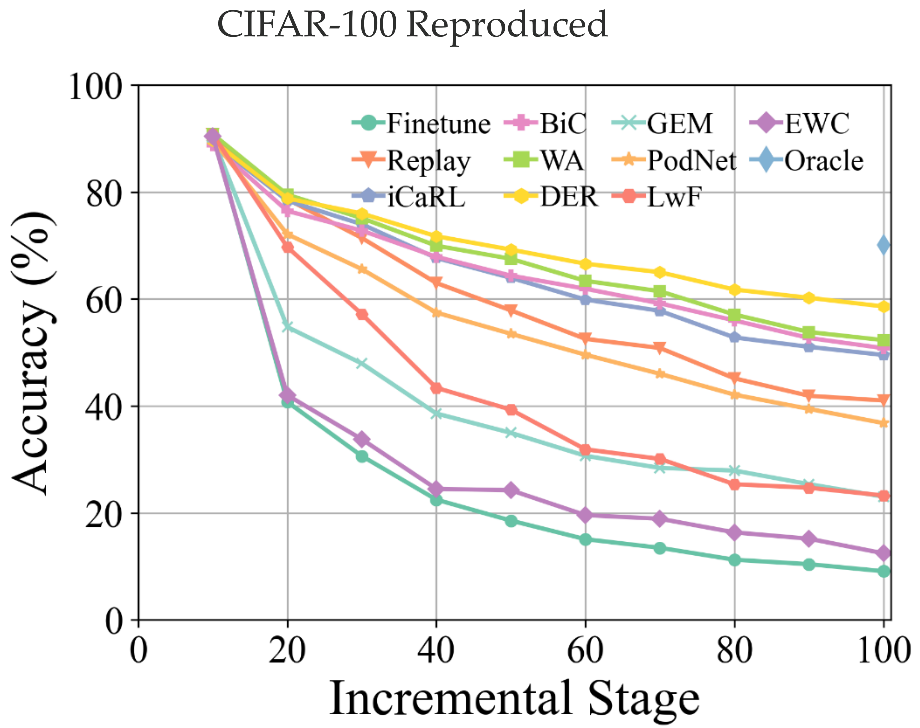




Old model

New model

Class Incremental Learning Toolbox



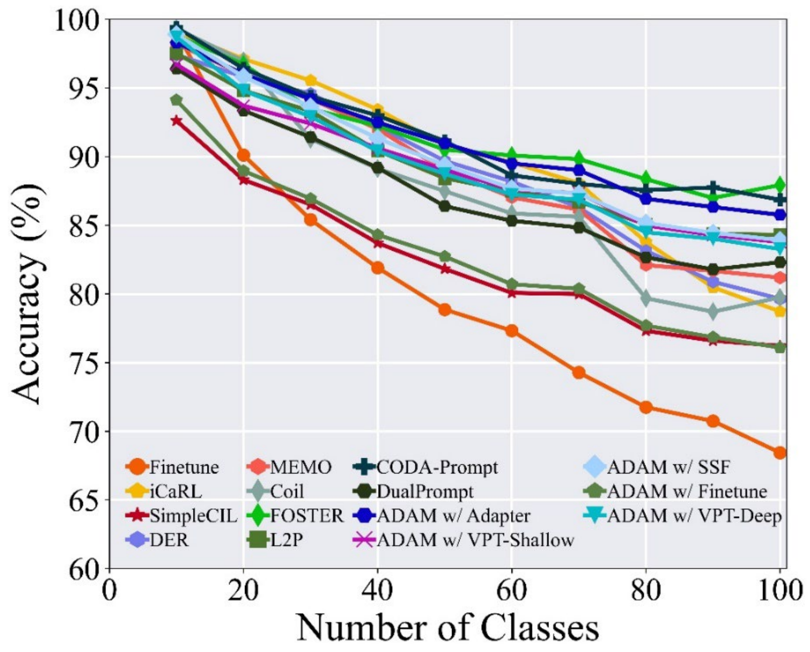
https://github.com/G-U-N/PyCIL

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, De-Chuan Zhan. *PyCIL: A Python Toolbox for Class-Incremental Learning.* **SCIENCE CHINA Information Sciences** 2023.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, Ziwei Liu. *Deep class-incremental learning: A survey.* CoRR 2023.

Pre-trained Continual Learning Toolbox

CIFAR-100 Reproduced



https://github.com/sun-hailong/LAMDA-PILOT

Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, De-Chuan Zhan. *PILOT: A Pre-Trained Model-Based Continual Learning Toolbox*. CoRR 2023.