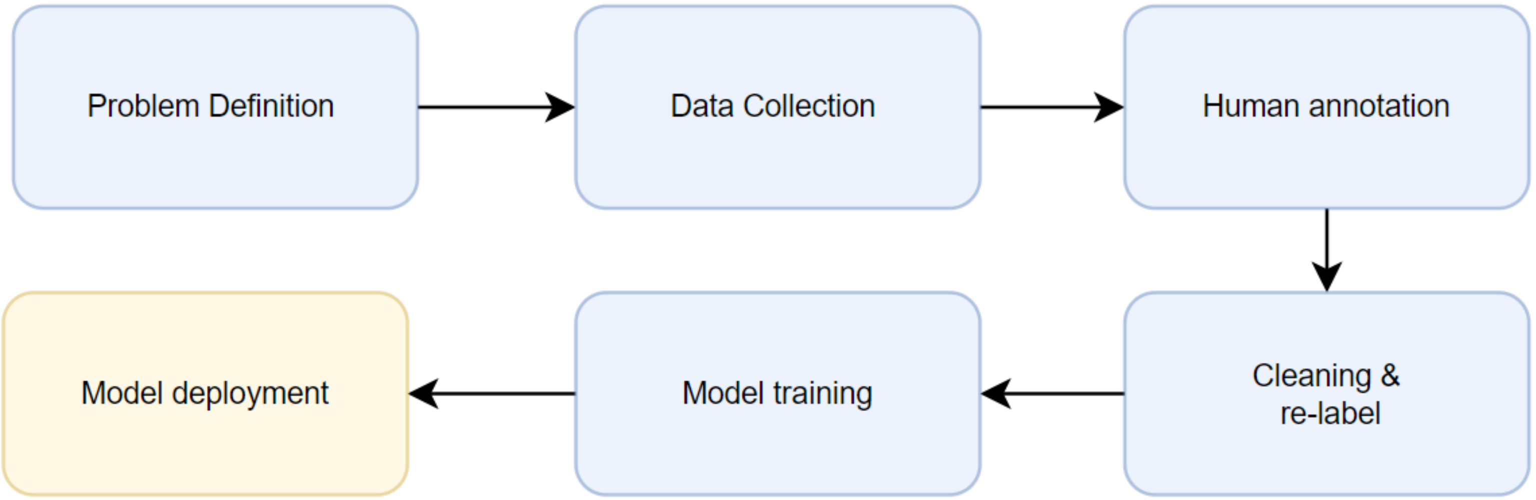
A Meta-Learning Approach for Few-Shot Class Incremental Learning

Li Gu

July 18, 2022 @CBG Canada

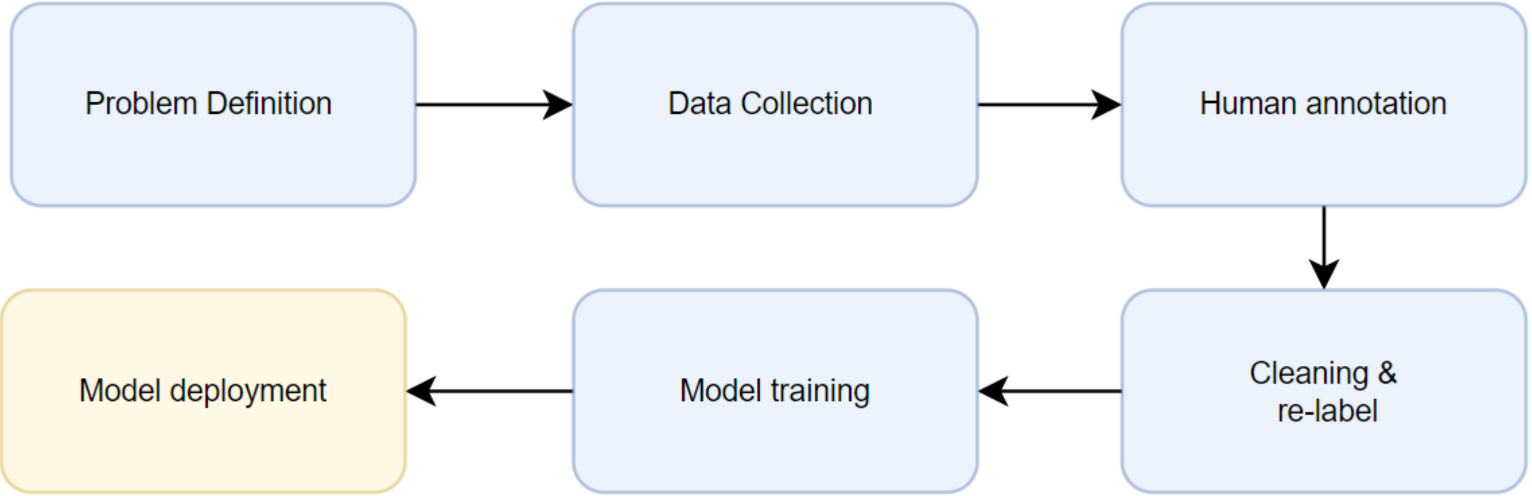
Classic Machine Learning Pipeline

- Classic Machine learning works well in a closed world
- Fixed set of classes, lots of labeled data per class, balanced dataset



Drawbacks of classic ML paradigm

Drawback 1: Lack of adaptation capabilities once deployed



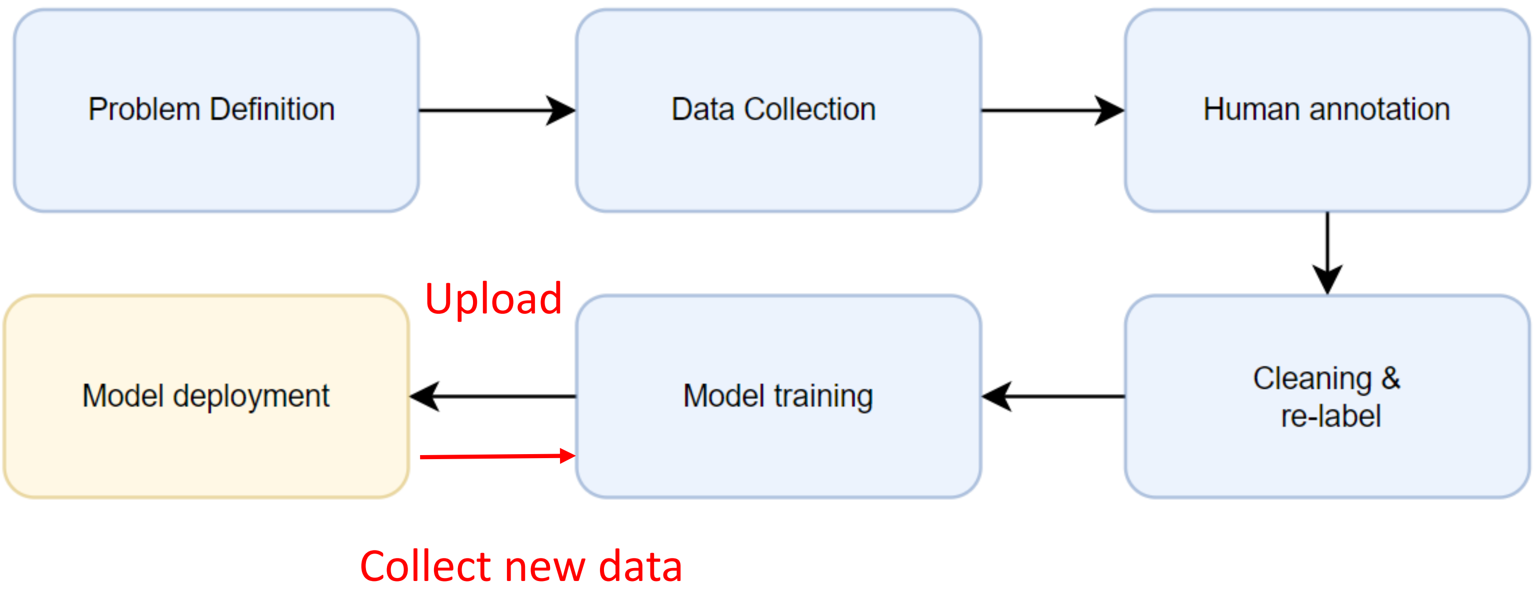
New concepts ?
New environments ?

Deploy

Drawbacks of classic ML paradigm

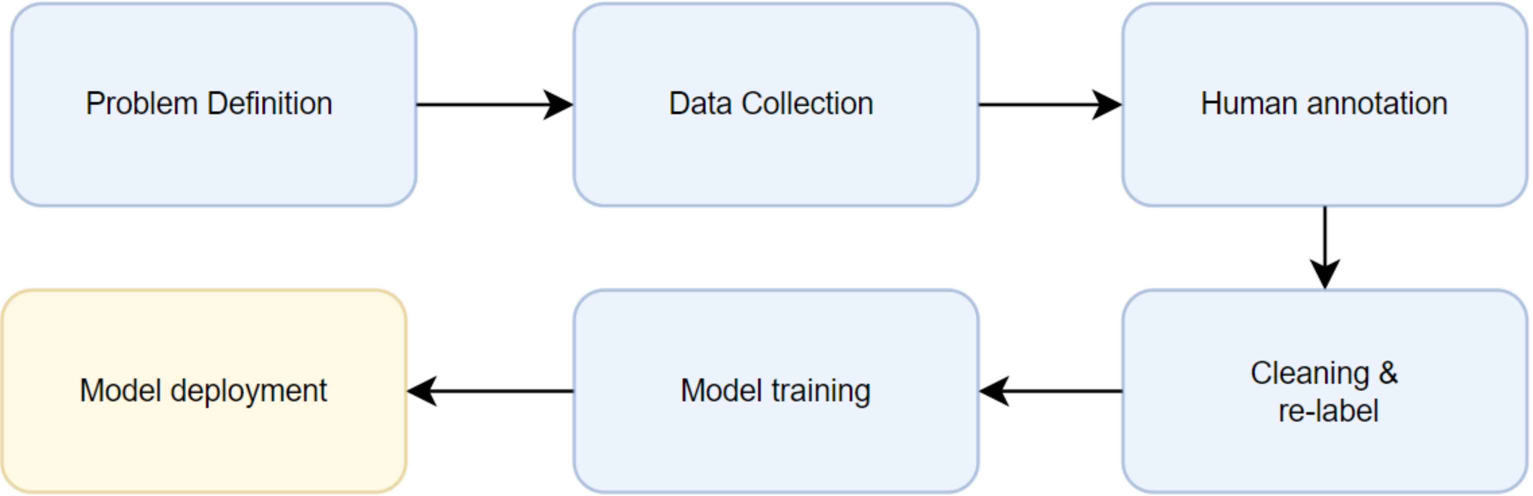
Drawback 1: Lack of adaptation capabilities once deployed

Naïve Solution: Keep collecting new data and re-train the model → Privacy-related regulations



Drawbacks of classic ML paradigm

Drawback 2: Data Hungary. Resource Inefficiency if collecting all new data locally



collect data locally

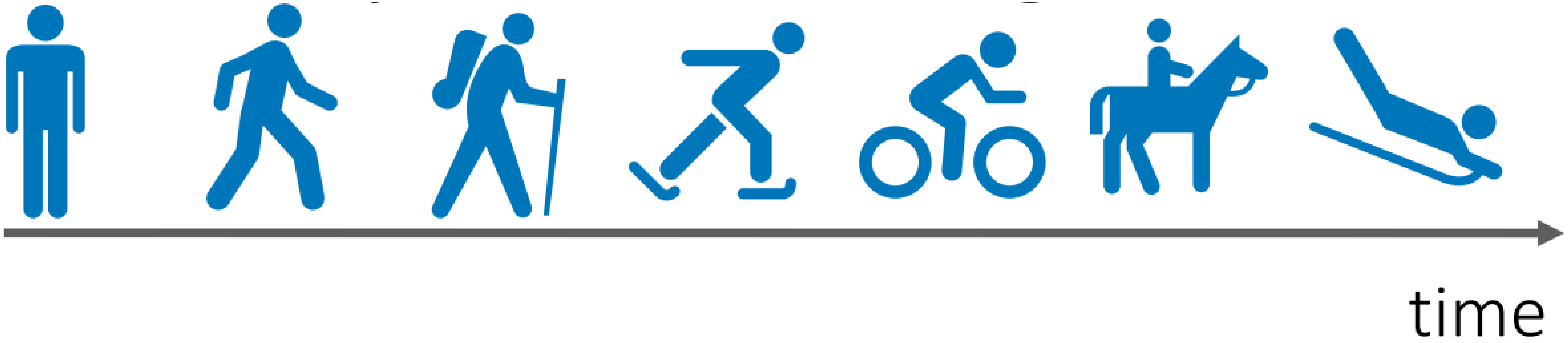
Multi-sensor in self-driving car

- 50GB/s streaming data.
- ~30240 TB of data after only a week.
- Impossible to re-train from scratch and to adapt fast.

CORe50: a New Dataset and Benchmark for Continuous Object Recognition. [Lomonaco V. CoRL2017]

Drawbacks of classic ML paradigm

Drawback 3: Humans can accumulate knowledge and learn in a dynamically changing environment



What is Lifelong Learning / Continual Learning?

Problem Definition

- Learn a sequence of task, $T_1, T_2, \ldots, T_N \ldots$ incrementally. Each task T_t has a small training dataset $D_t = \{x_{t,i}, y_{t,i}\}_{i=1}^{n_t}$
- These small datasets may be of different types and from different domains
- At time step t, the learner can only access D_t instead of all datasets

Lifelong machine learning. [Chen, Z., & Liu, B. (2018).]
An introduction to Lifelong supervised learning [A Sodhani. 2022]

What is Lifelong Learning / Continual Learning?

Problem Definition

- Learn a sequence of task, $T_1, T_2, \ldots, T_N \ldots$ incrementally. Each task T_t has a small training dataset $D_t = \{x_{t,i}, y_{t,i}\}_{i=1}^{n_t}$
- These small datasets may be of different types and from different domains
- At time step t, the learner can only access D_t instead of all datasets

Goal

- Accumulate the knowledge in dynamic environments across the lifetime
  - Learning of the new task T_{N+1} should not result in degradation of accuracy for previous N tasks.
- Transfer the accumulated knowledge to improve their performance on both previous and future tasks
  - Forward transfer → Enable fast adaptation

Lifelong machine learning. [Chen, Z., & Liu, B. (2018).]
An introduction to Lifelong supervised learning [A Sodhani. 2022]

Lifelong Learning is a popular research topic

← **Thread**

Timothée Lesort
@TLesort

[0/N] Hi all, I have made a thread 🧶 with of all continual learning papers accepted @CVPR 2022. (I did not expect it to be that long when I started… N=45 😅)
I will be in person at CVPR2022 I hope to see you there. 😎

11:23 AM · May 26, 2022 · Twitter Web App

DEEPMIND IS HIRING A

Continual Learning Research Engineer - London

Avalanche: an End-to-End Library for Continual Learning

Powered by ContinualAI

Avalanche powered by ContinualAI

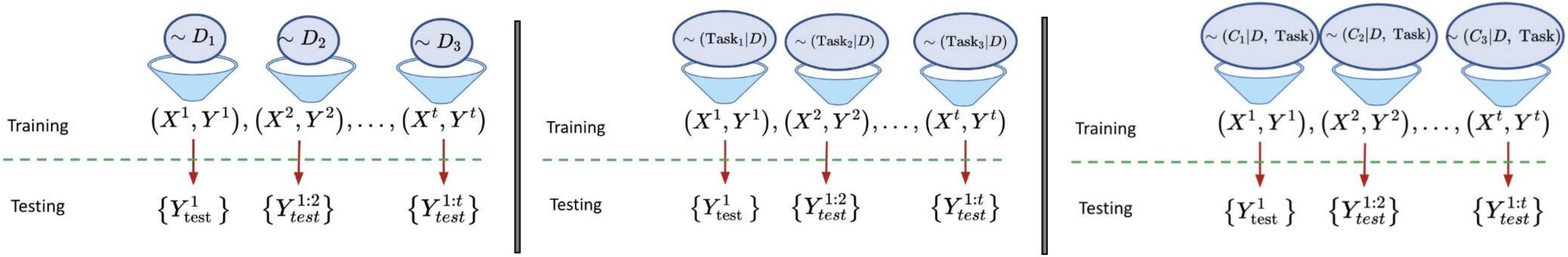
Avalanche is an *End-to-End Continual Learning Library* based on **PyTorch**, born within **ContinualAI** with the unique goal of providing a **shared** and **collaborative** open-source (MIT licensed) **codebase** for *fast prototyping*, *training* and *reproducible evaluation* of continual learning algorithms.

Continual Learning: On Machines that can Learn Continually

A University of Pisa, ContinualAI and AIDA Doctoral Academy Course

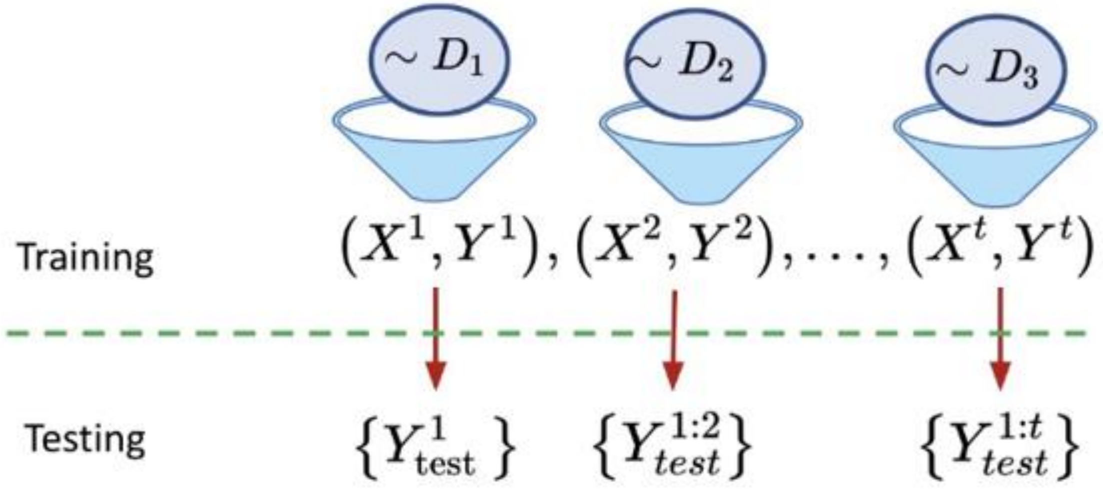
Università di Pisa ContinualAI AIDA Artificial Intelligence Doctoral Academy

Three prominent scenarios in lifelong learning



(a) Domain-Incremental **(b)** Task-Incremental **(c)** Class-Incremental

Three prominent scenarios in lifelong learning

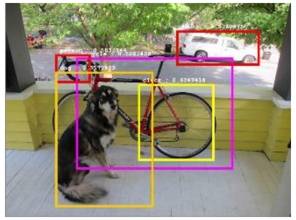


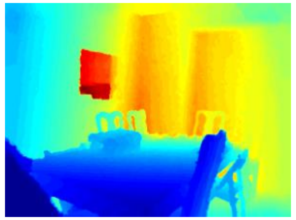
(a) Domain-Incremental

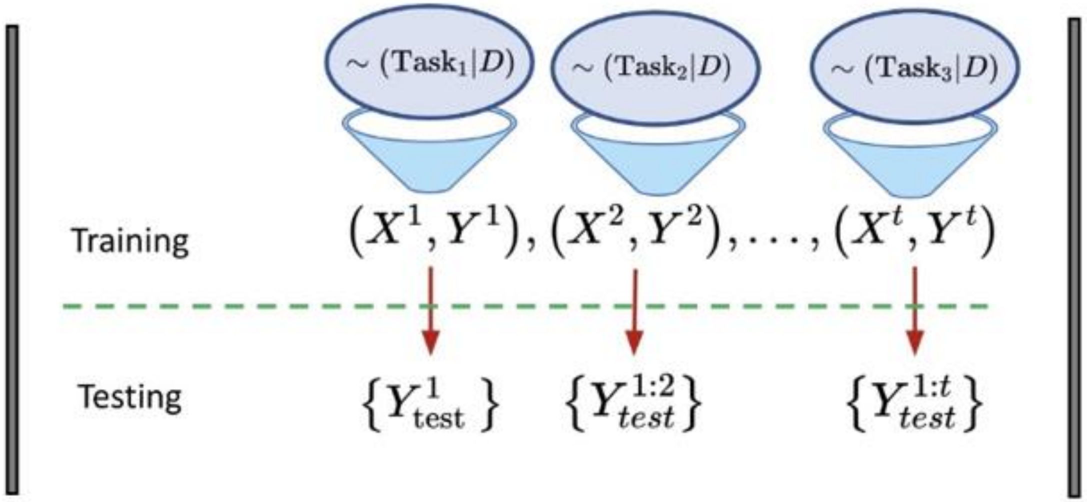
Problem
- Recommend k courses from N candidates to a school

Dataset
- Each task contains one specific student’s data.
- Covariate shift across students. Only $P(x)$ changes

Three prominent scenarios in lifelong learning







(b) Task-Incremental

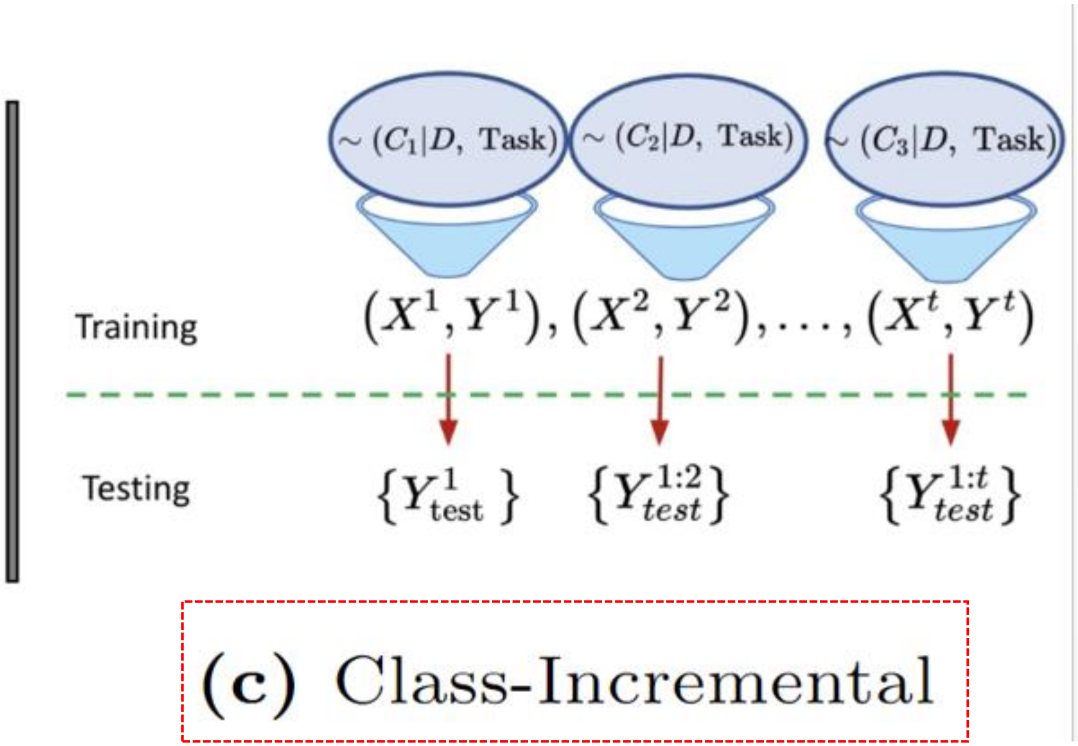
Problem

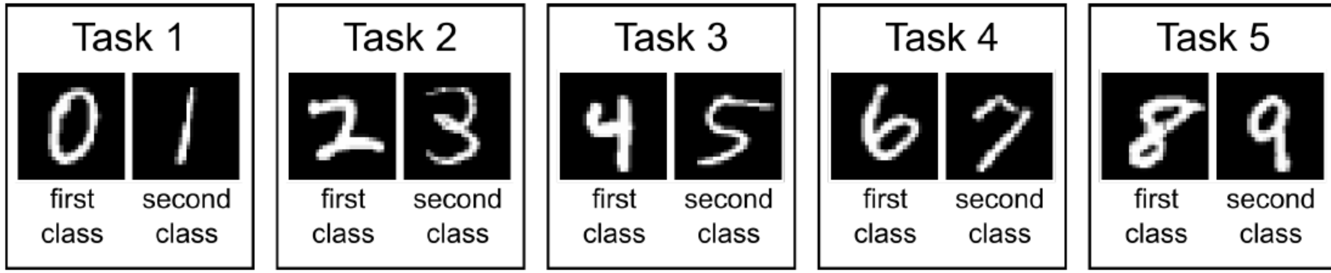
- Learn to handle object detection, segmentation and depth prediction at the same time

Dataset

- Encounter those three tasks sequentially

Three prominent scenarios in lifelong learning



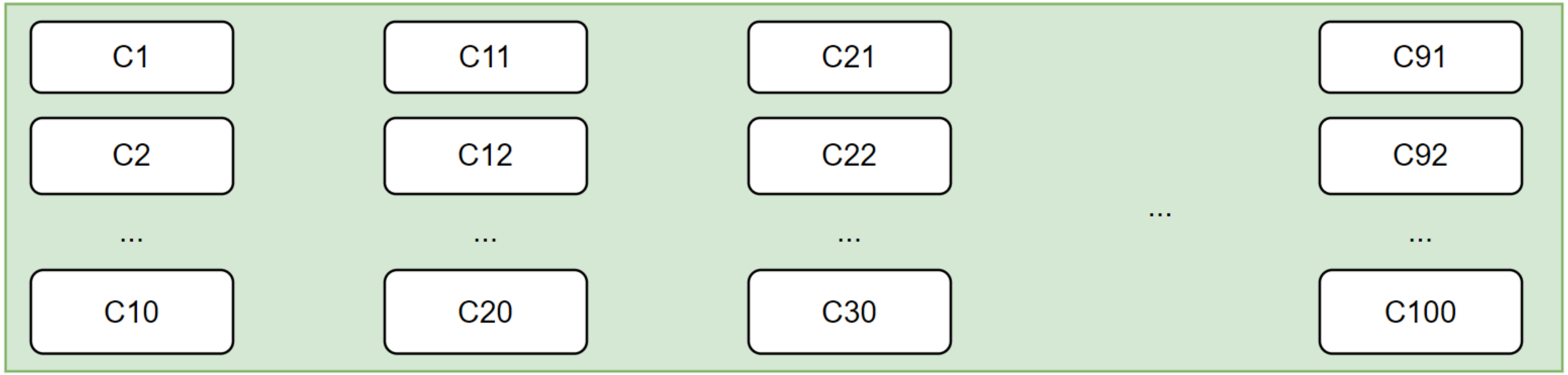


Problem
- Learn to recognize all 10 digits

Dataset
- Encounter non-overlapped classes sequentially during incremental learning

Three scenarios for continual learning [van de Ven. NeurIPS 2019 workshop]

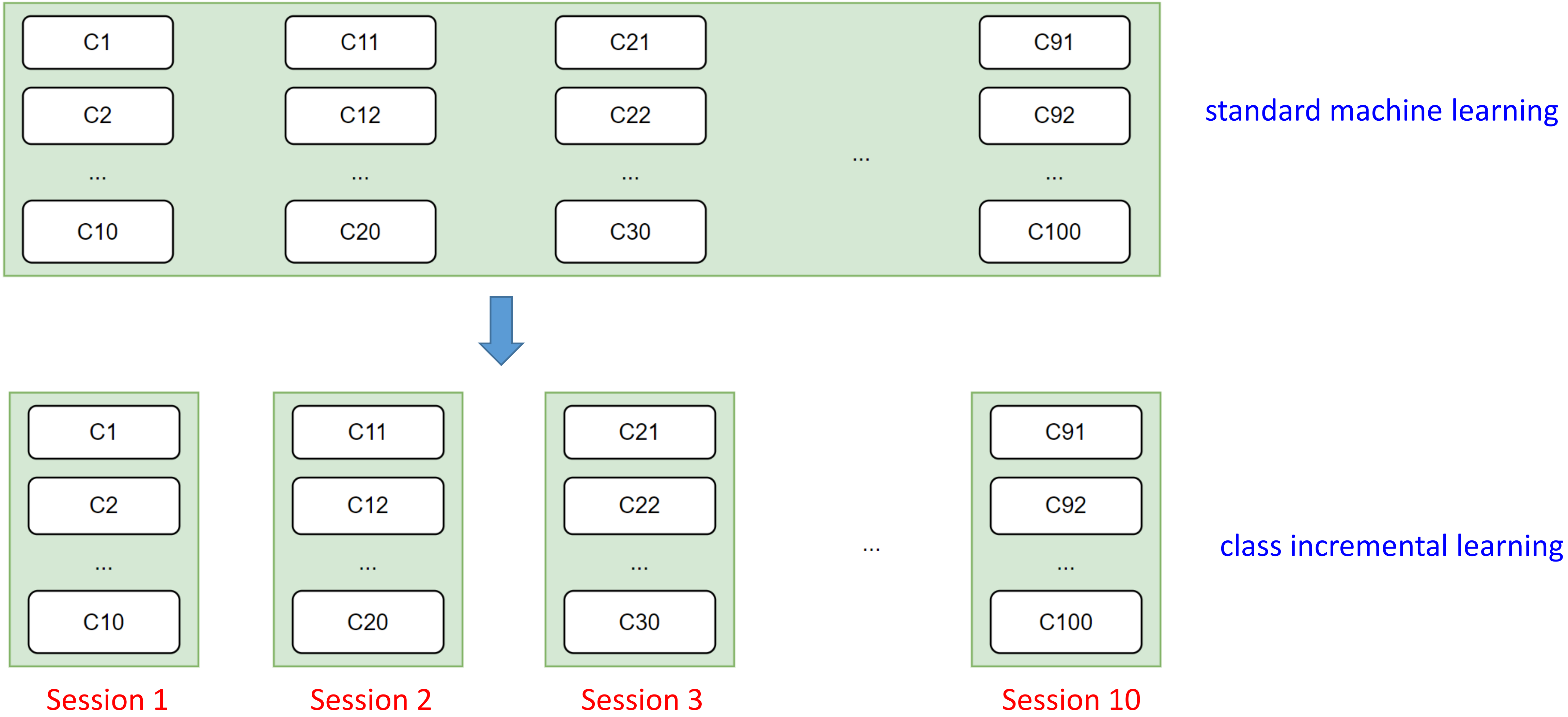
An example of class incremental learning (CIL)



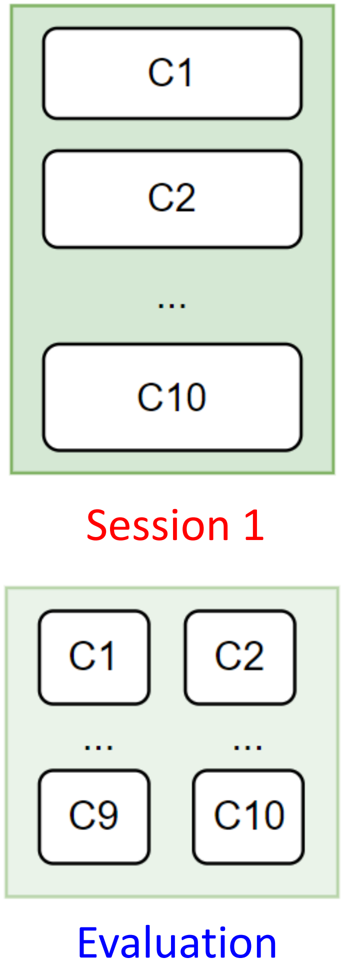
CIFAR100 dataset

- 100 classes
- 500 examples per class

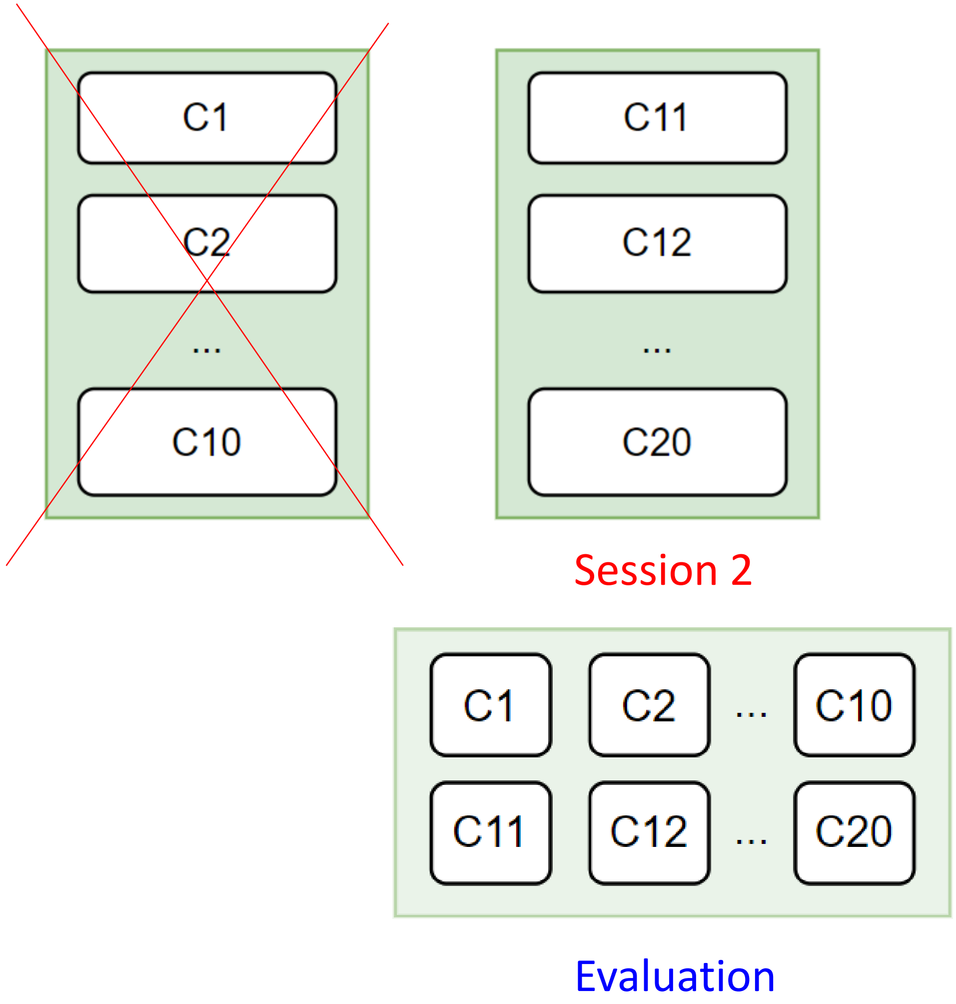
An example of class incremental learning (CIL)



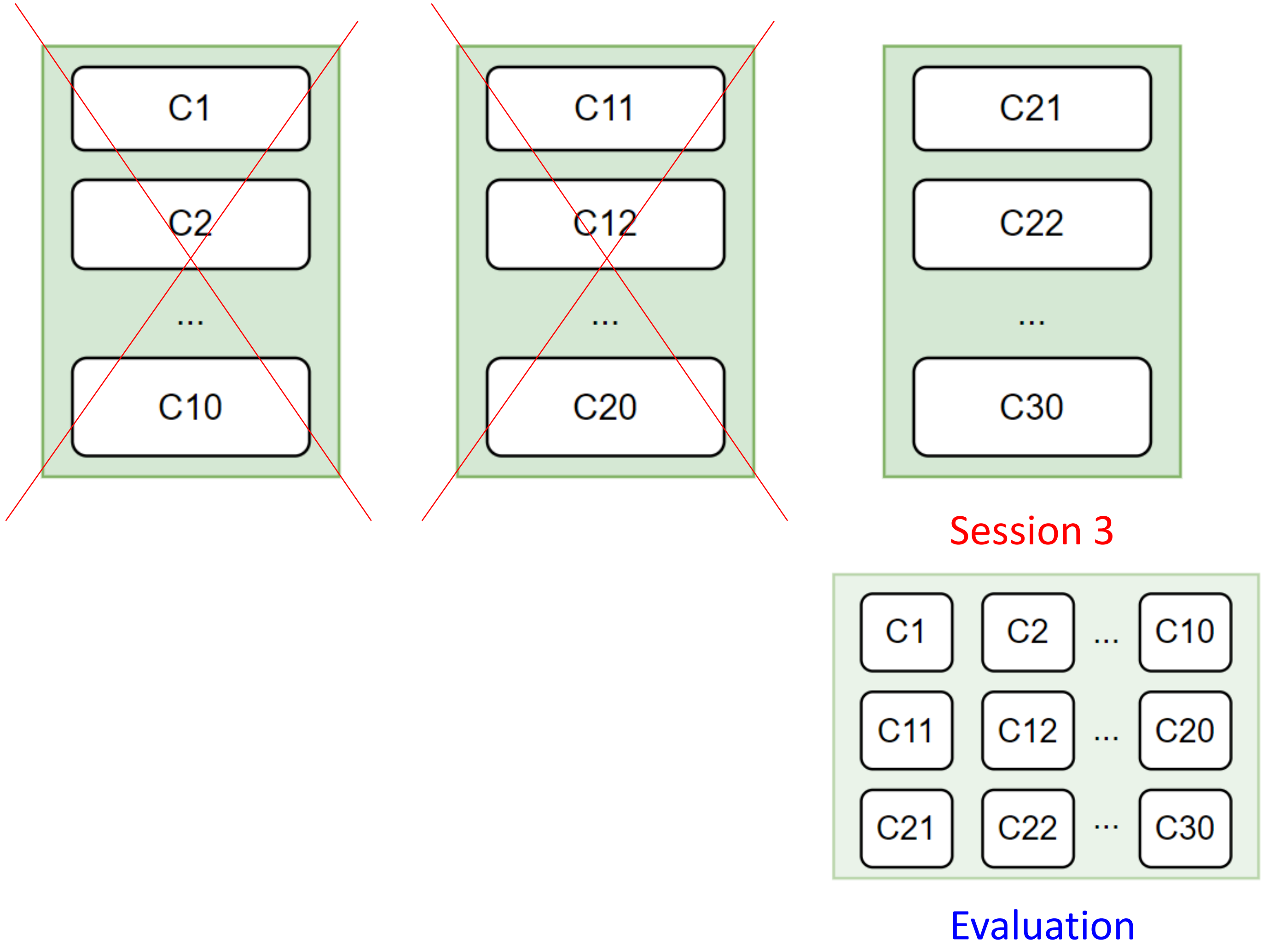
An example of class incremental learning (CIL)



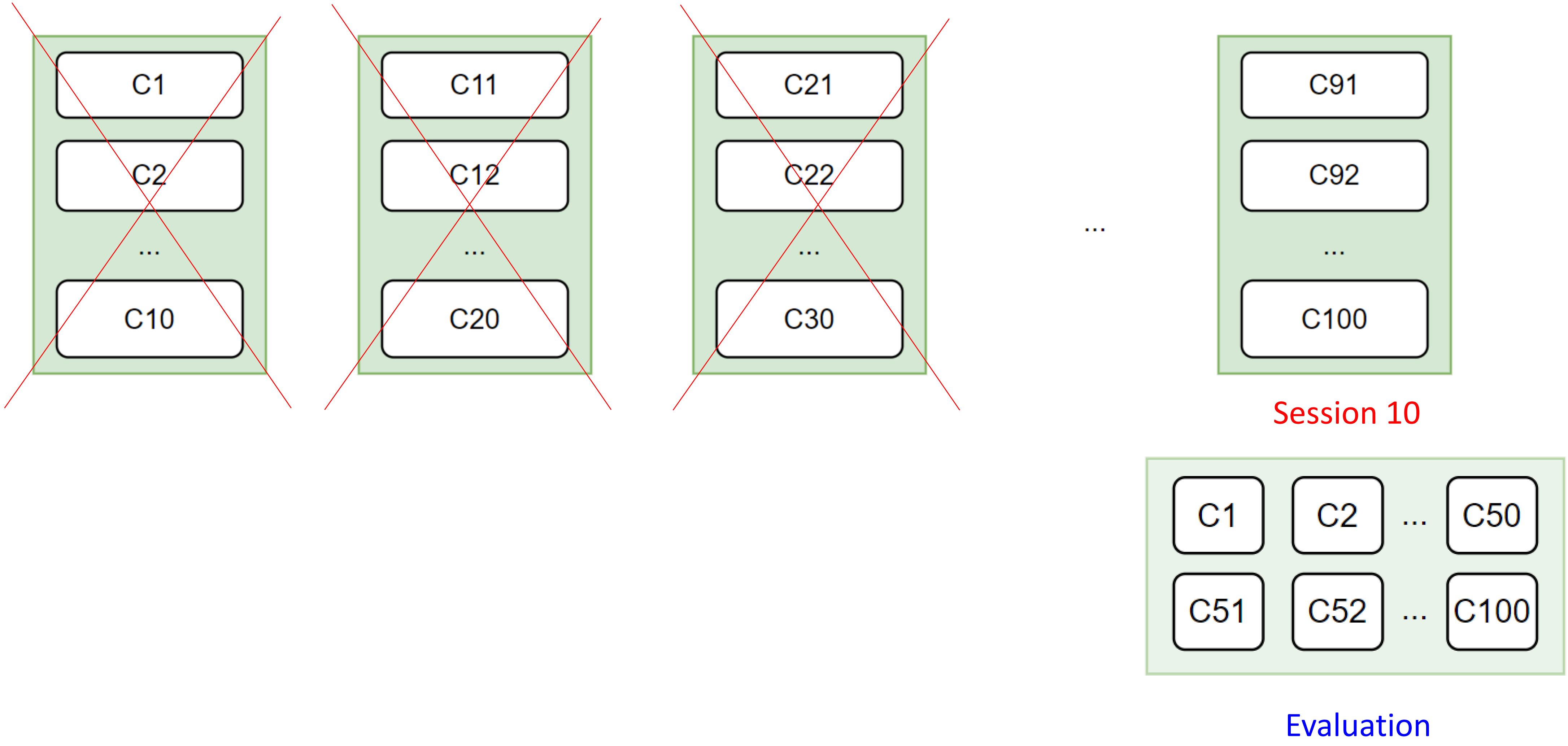
An example of class incremental learning (CIL)



An example of class incremental learning (CIL)



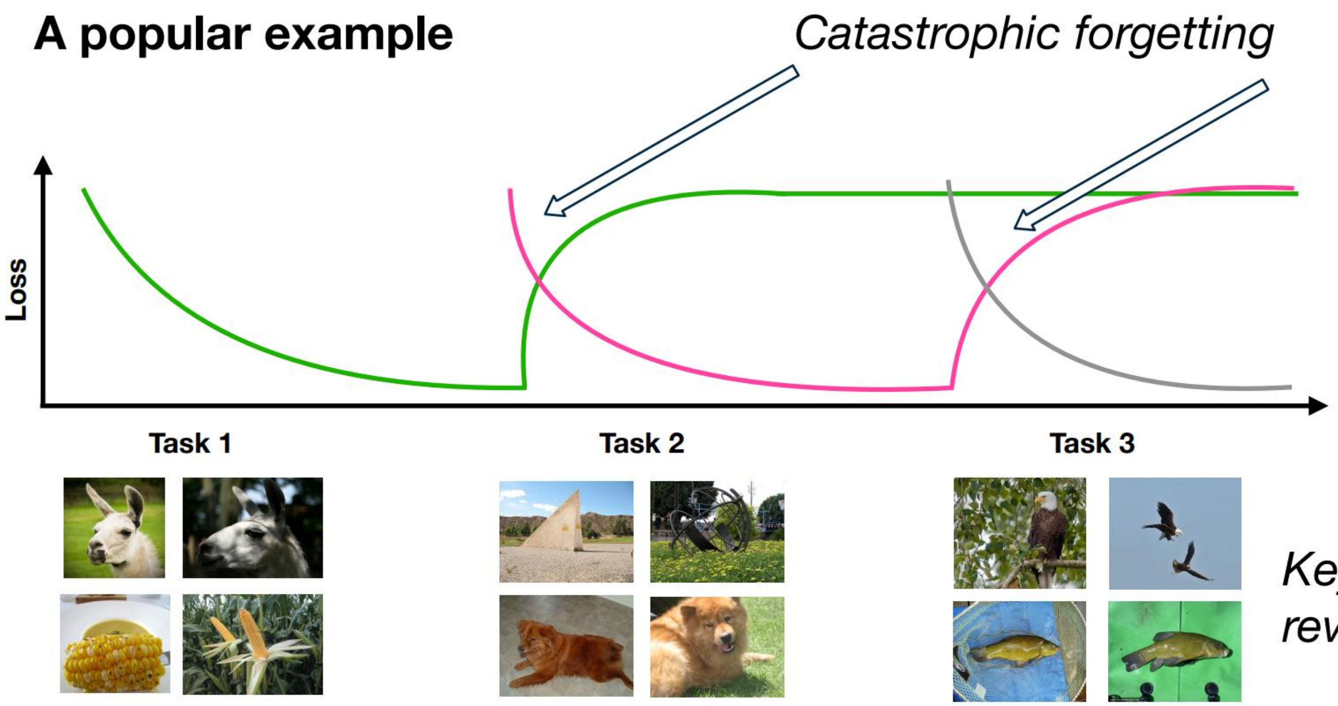
An example of class incremental learning (CIL)



Can we apply NNs directly to class incremental learning ?

Key Challenge: Catastrophic Forgetting

- Forget previously learned information upon learning new information
- Mostly due to Gradient Descent. Assume the knowledge stored in NN's parameters



Plasticity – Stability dilemma

- Plasticity: Learn new knowledge
- Stability: Not forget old knowledge

Key assumption: no access to/ revisiting of prior "task" data!

Key Challenge: Catastrophic Forgetting

- Forget previously learned information upon learning new information
- Mostly due to Gradient Descent. Assume the knowledge stored in NN's parameters

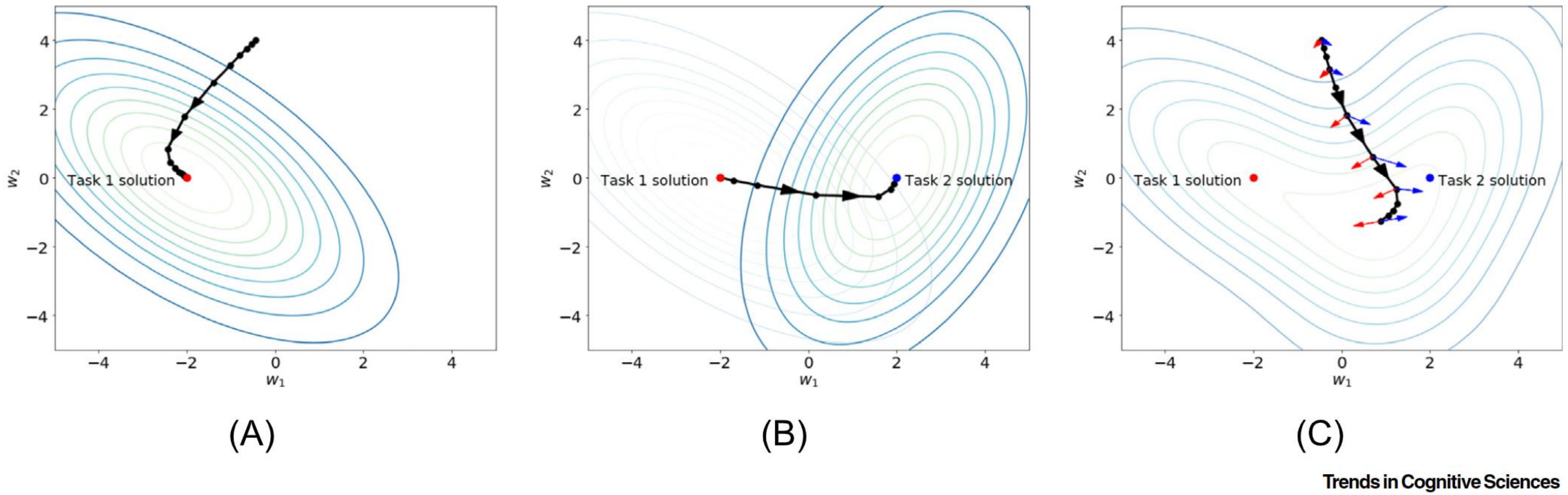


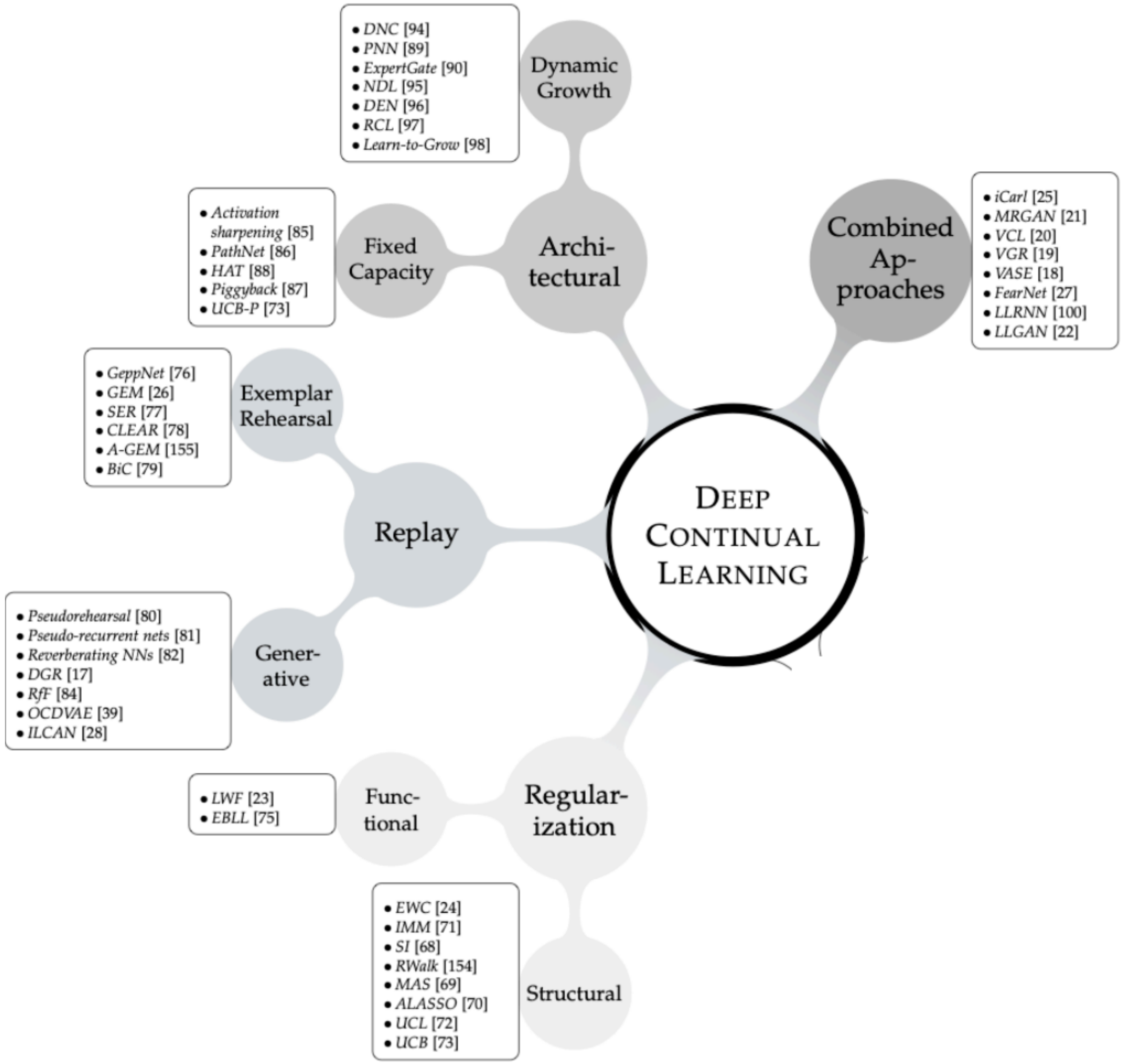
Figure 3. Illustrations of Gradient Descent Optimization for Different Tasks. (A) The trajectory taken by gradient descent optimization when minimizing a loss corresponding to a single task. (B) The optimization trajectory when subsequently training the same model on a second task. (C) The trajectory taken when using the total loss from both tasks (black) and the gradients from each individual task at multiple points during optimization (red and blue). See Box 2 for more detailed discussion.

Embracing Change: Continual Learning in Deep Neural Networks [Hadsell et al, Trends in Cognitive Sciences 2020]

How can we alleviate Catastrophic Forgetting ?

Common strategies on Lifelong Learning

1. Regularize important parameters
2. Rehearsal (Reply Buffer)
3. Modular architecture (Dynamic Architecture)



A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. [Mundt, M.A. *arXiv preprint 2020]*

How to address Catastrophic Forgetting

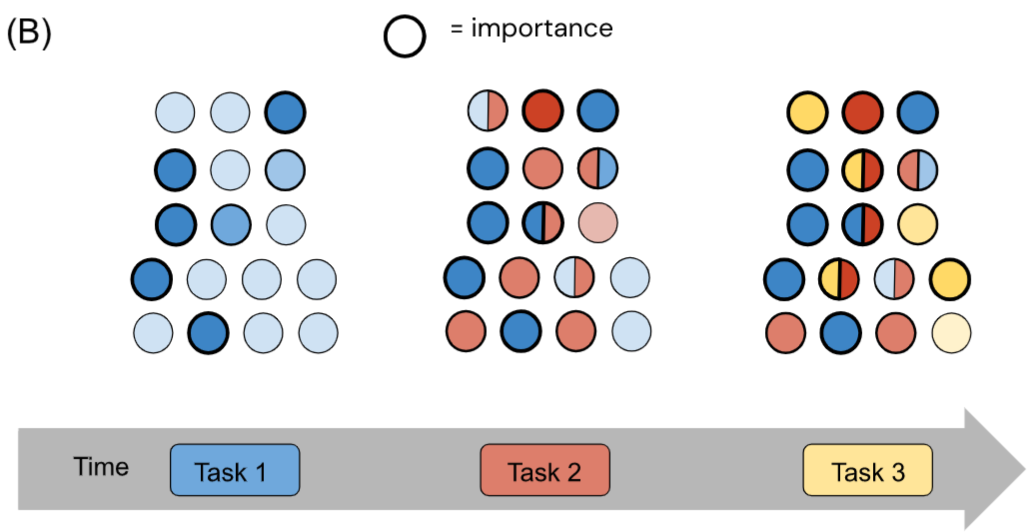
Regularize important parameters

Assumption

- Task-related knowledge stored in NN's parameter
- Less change on parameters, less forgetting issues

Solution

- Explicitly identify relevant parameters for old tasks and restrict its changes on new task



Embracing Change: Continual Learning in Deep Neural Networks [Hadsell et al, Trends in Cognitive Sciences 2020]

How to address Catastrophic Forgetting

Rehearsal (Replay buffer)

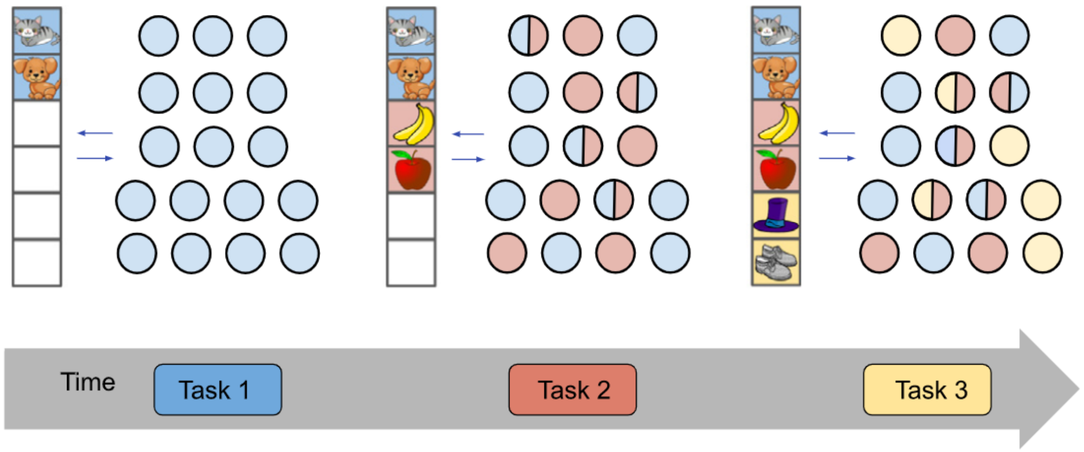
Assumption

- Task-related knowledge stored in a subset of informative data examples

Solution

- Identify informative old data and store them in a replay buffer
- Train a generative model on old task and then generate old task data in future tasks

(D)



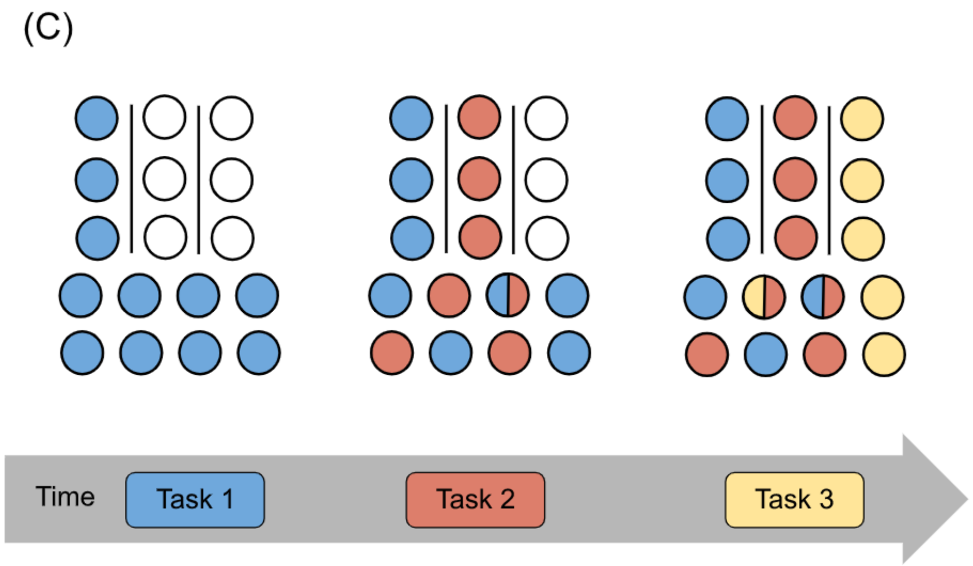
Embracing Change: Continual Learning in Deep Neural Networks [Hadsell et al, Trends in Cognitive Sciences 2020]

How to address Catastrophic Forgetting

Modular architecture (Dynamic Architecture)

Assumption

- The overlap of distributed representations leads to forgetting issue[1].
- Different tasks should have their own set of isolated parameters



[1] Using Semi-Distributed Representations to Overcome Catastrophic Forgetting in Connectionist Networks [French, R. M. AAAI 1993]

Embracing Change: Continual Learning in Deep Neural Networks [Hadsell et al, Trends in Cognitive Sciences 2020]

How to address Catastrophic Forgetting

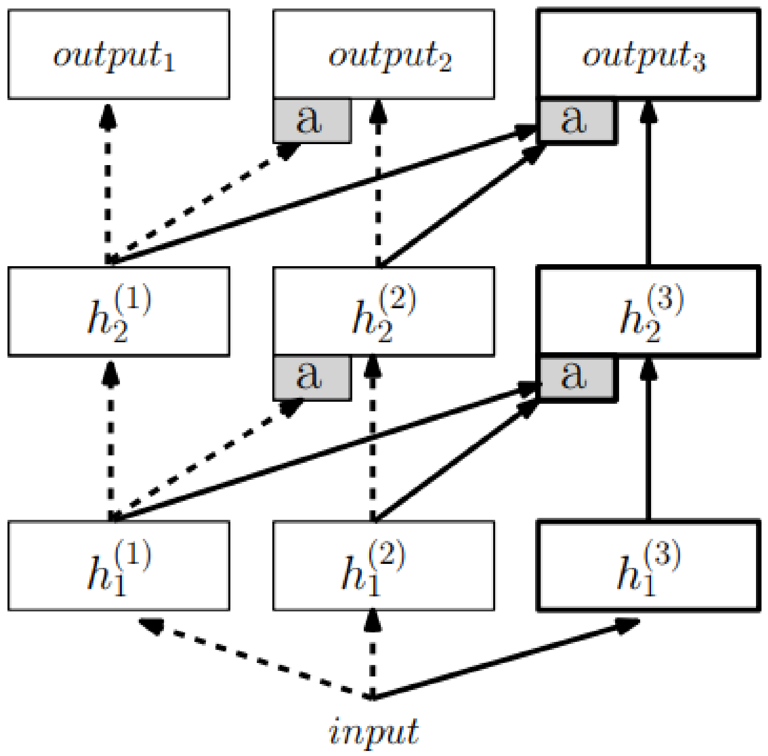
Modular architecture (Dynamic Architecture)

Assumption

- The overlap of distributed representations leads to forgetting issue.
- Different tasks should have their own set of isolated parameters

Solution 1

- Grow or expand the architecture



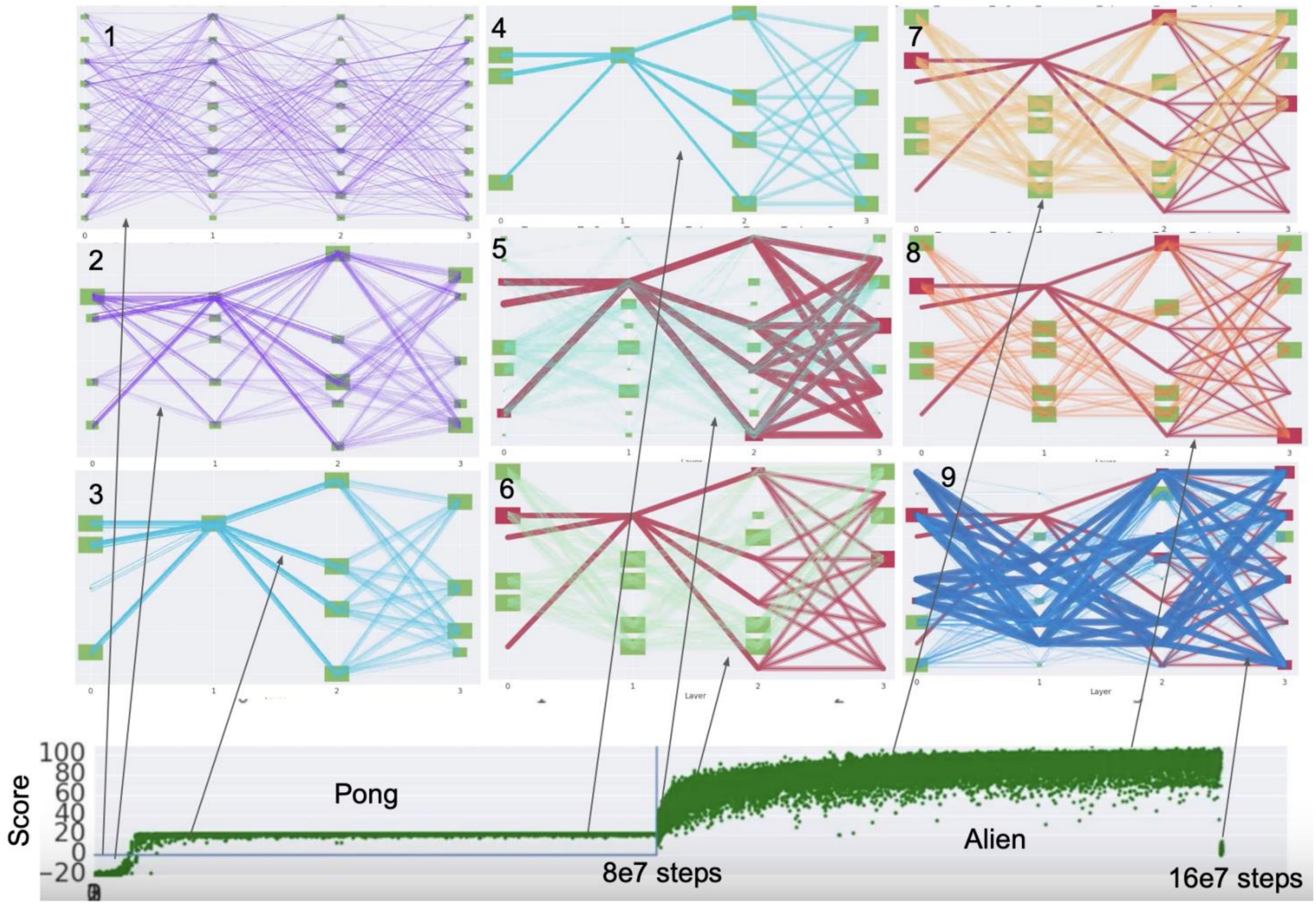
Progressive neural networks. [Rusu, A. A. *arXiv 2016]*

How to address Catastrophic Forgetting

Modular architecture (Dynamic Architecture)

Solution 2

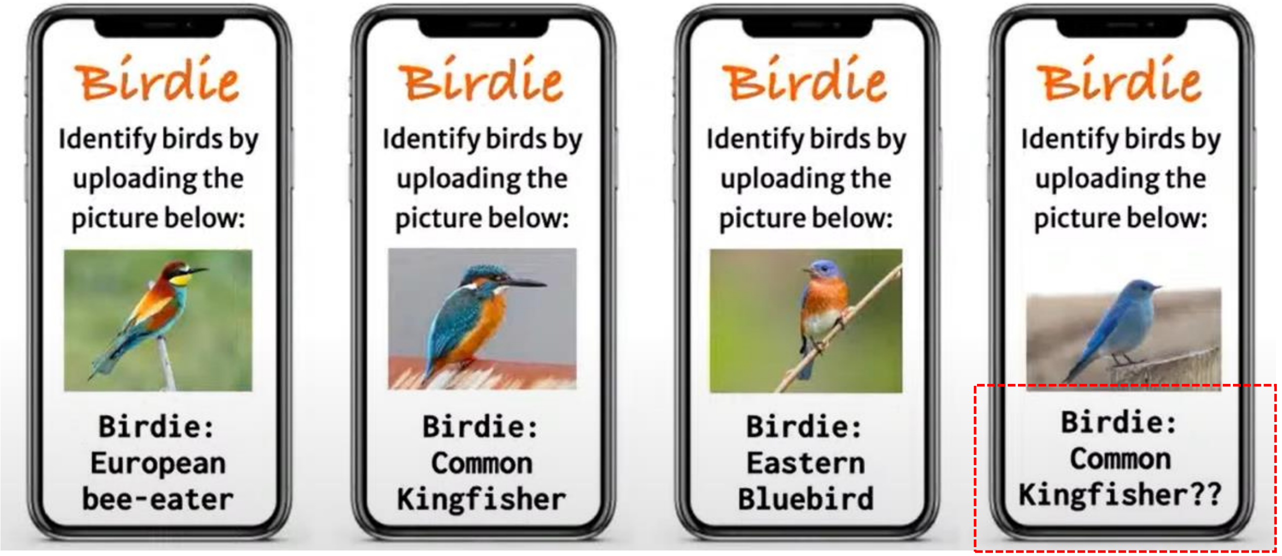
- The NN is over-parameterized
- Each task activates a small non-overlapped subset of parameters sequentially



PathNet: Evolution Channels Gradient Descent in Super Neural Networks. [Fernando et al. arXiv 2017]

What is Few Shot Class Incremental Learning?

Few shot class incremental learning (FSCIL)



Birdie is an AI agent that can classify 100 birds

101th bird ??

Collect & Annotate

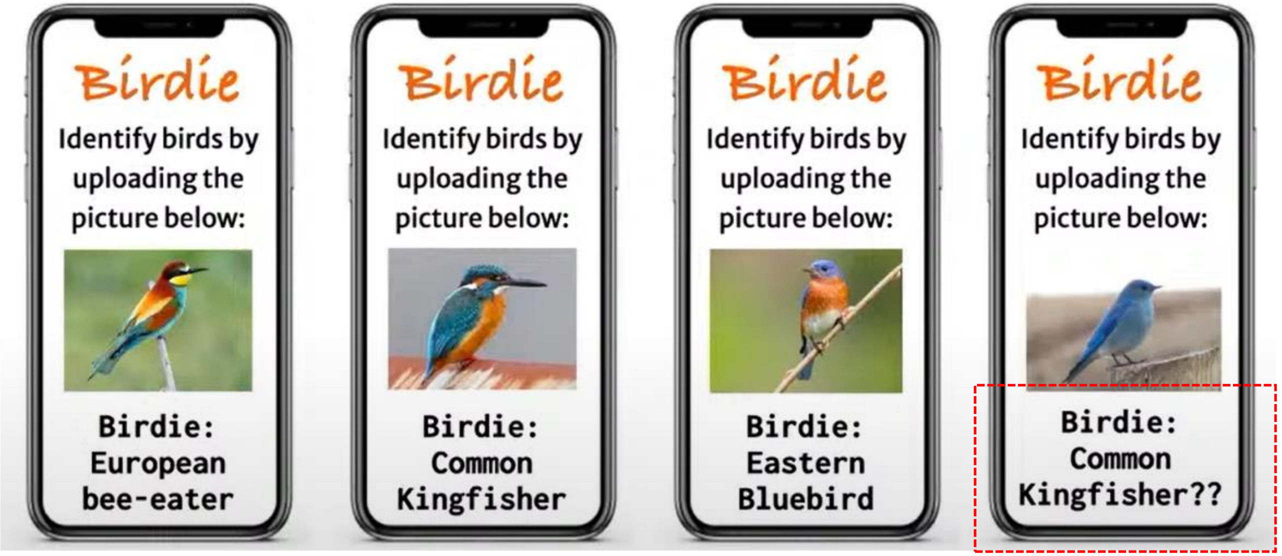
Few shot data

Subspace Regularizers for Few-Shot Class Incremental Learning. [Akyürek, A. ICLR 2022]

Few shot class incremental learning (FSCIL)

A more realistic scenario

- Imbalanced datasets across a sequence of tasks
- Few annotated data in each incremental session



Birdie is an AI agent that can classify 100 birds

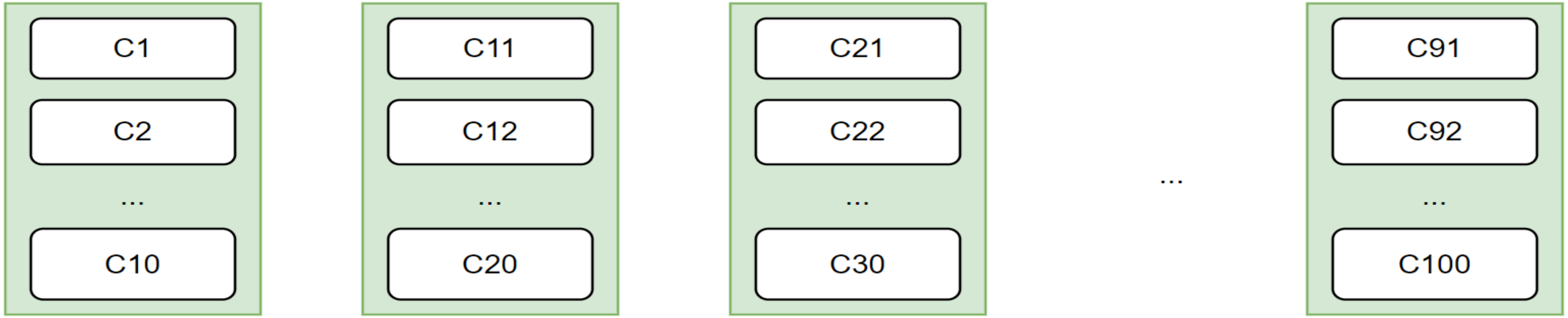
101th bird ??

Collect & Annotate

Few shot data

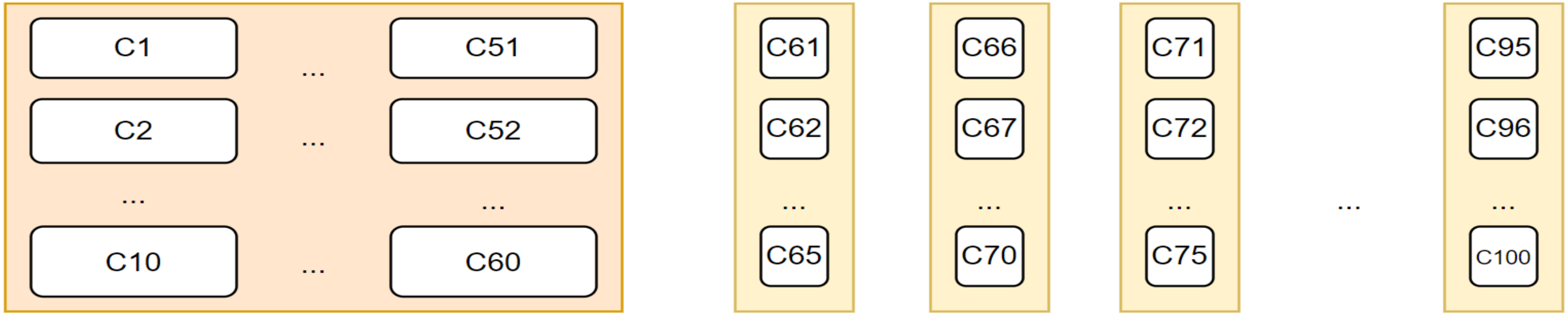
Subspace Regularizers for Few-Shot Class Incremental Learning. [Akyürek, A. ICLR 2022]

Few shot class incremental learning (FSCIL)



Problem

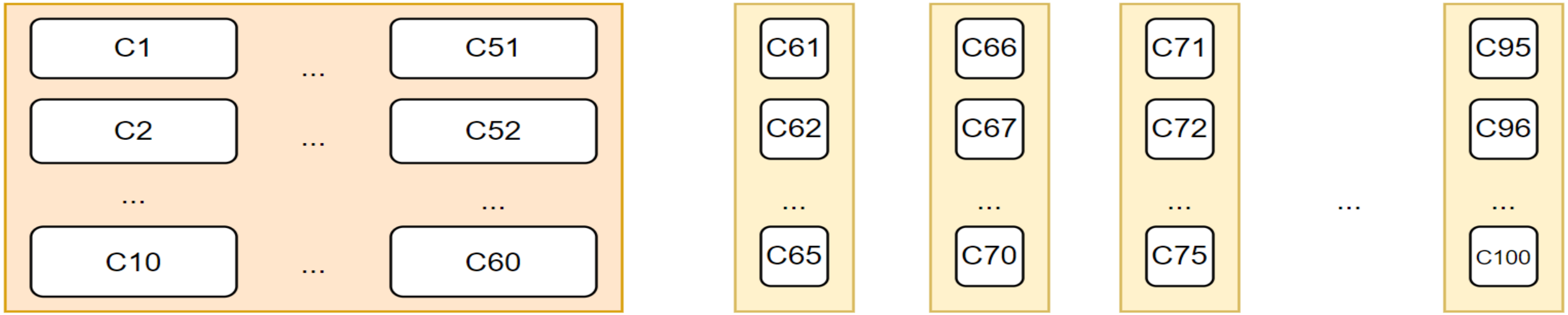
- Different number of classes between the base and incremental sessions
- Few-shot data in incremental sessions (N-way-K-shot)



Base session

8 incremental sessions

Few shot class incremental learning (FSCIL)



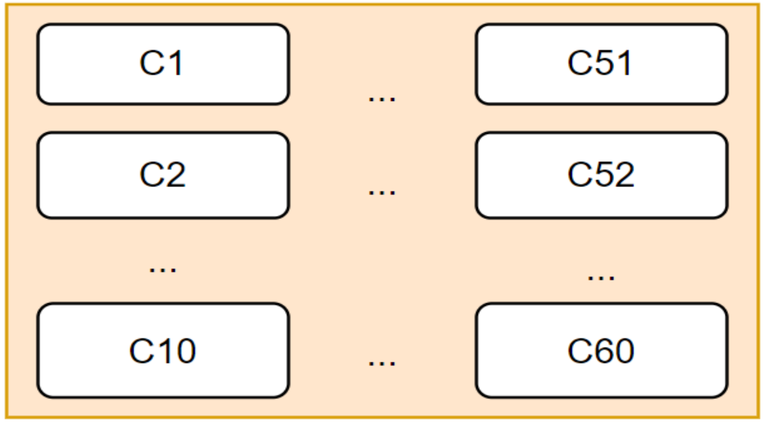
Challenges

- imbalanced datasets → Difficult to make a balance between plasticity & stability
- Few shot data → Overfitting in incremental sessions

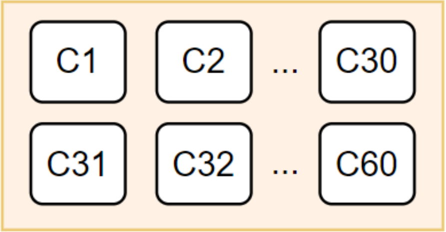
Three stages in FSCIL

- Base session training
- Incremental session learning
- Evaluation after each session

An example of FSCIL

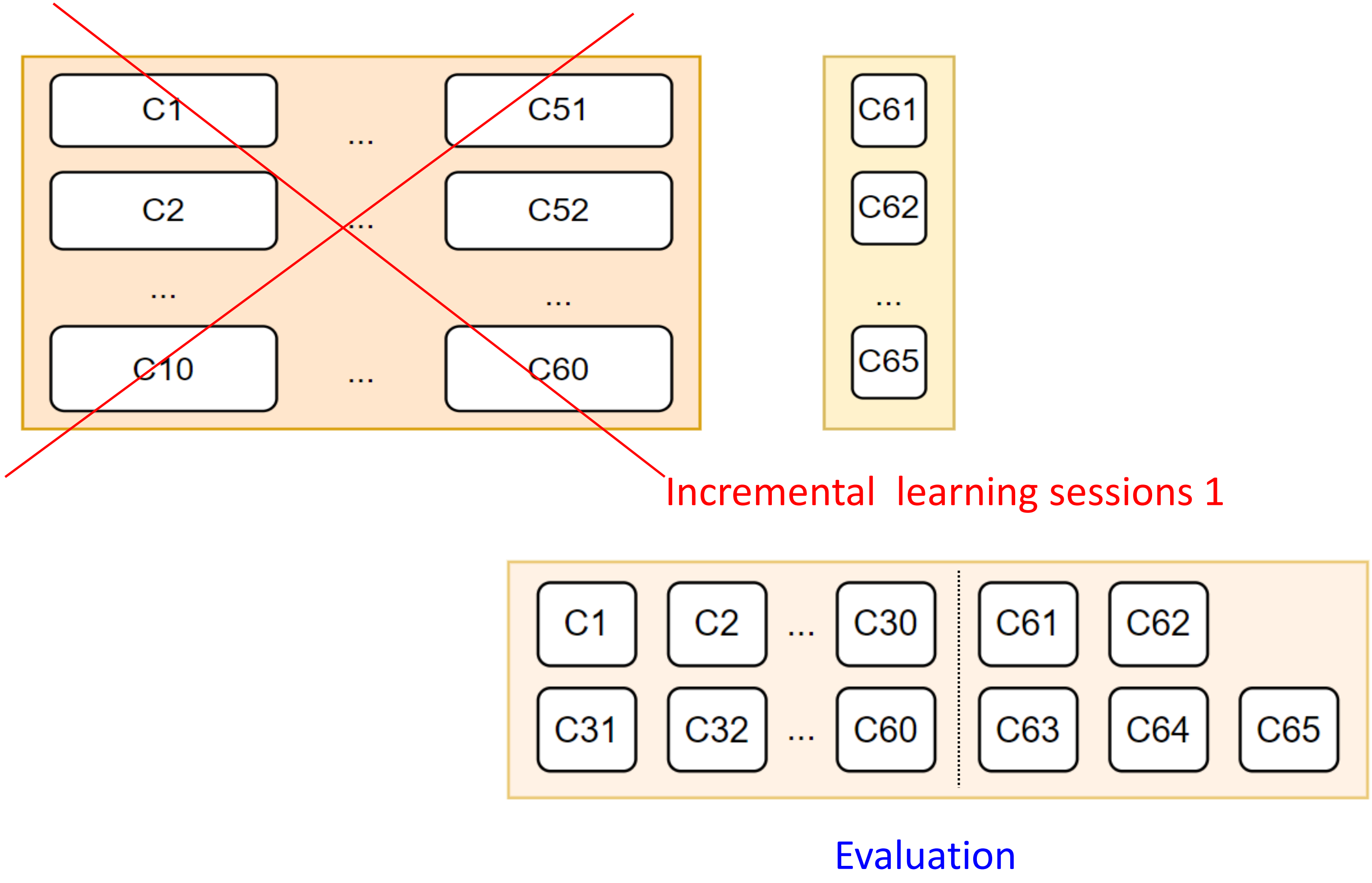


Base session training



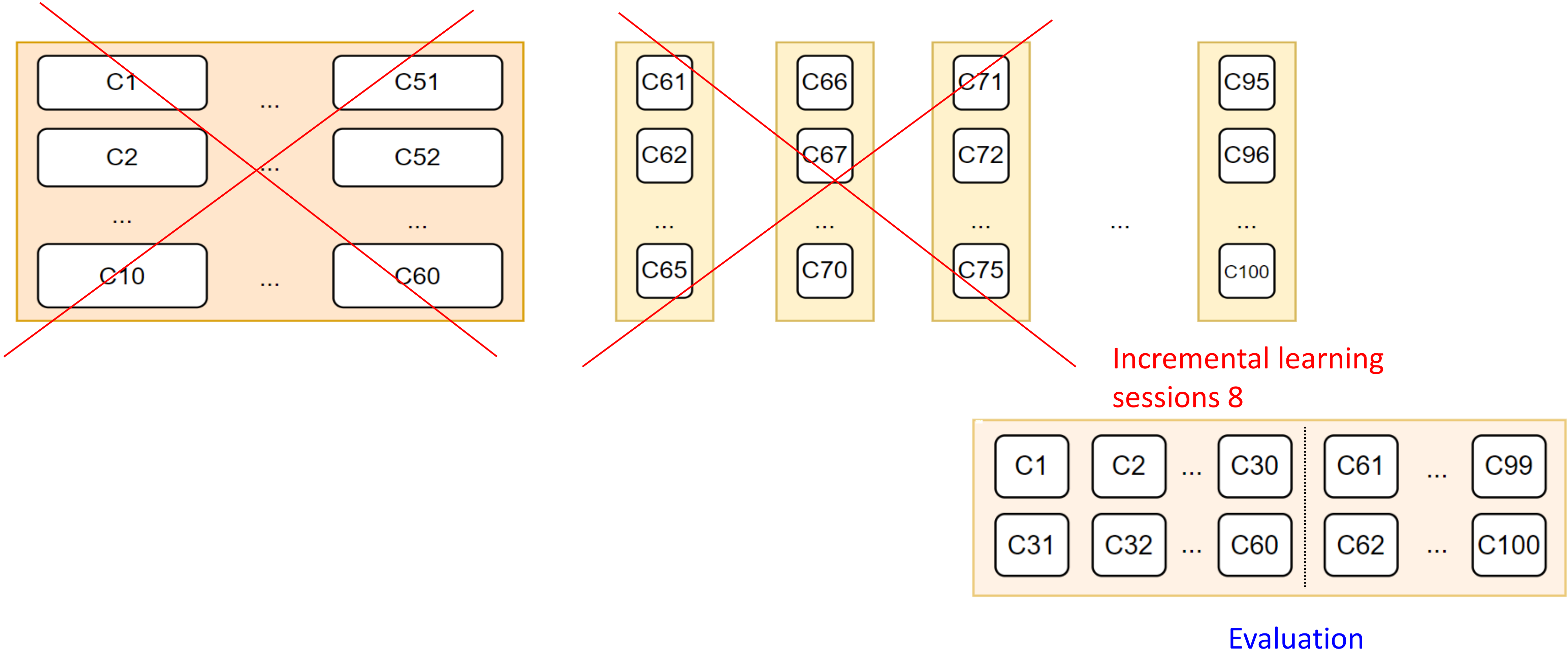
Evaluation

Few shot class incremental learning (FSCIL)

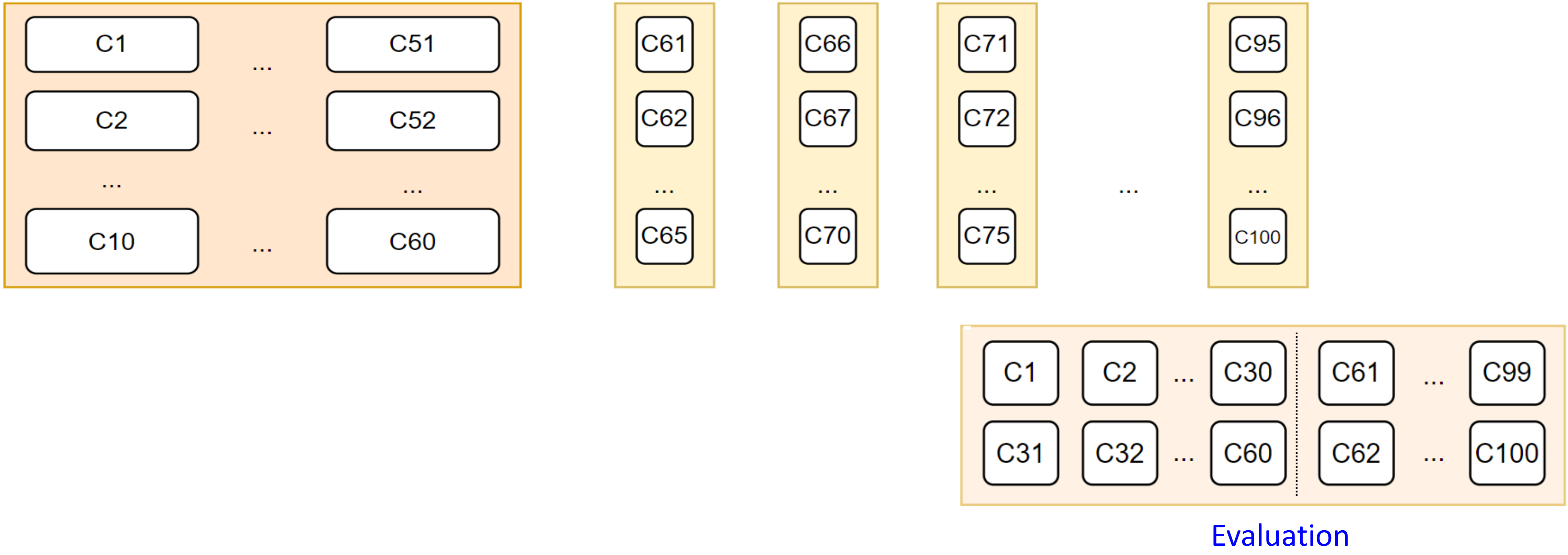


Classes across each session during evaluation phase are balanced

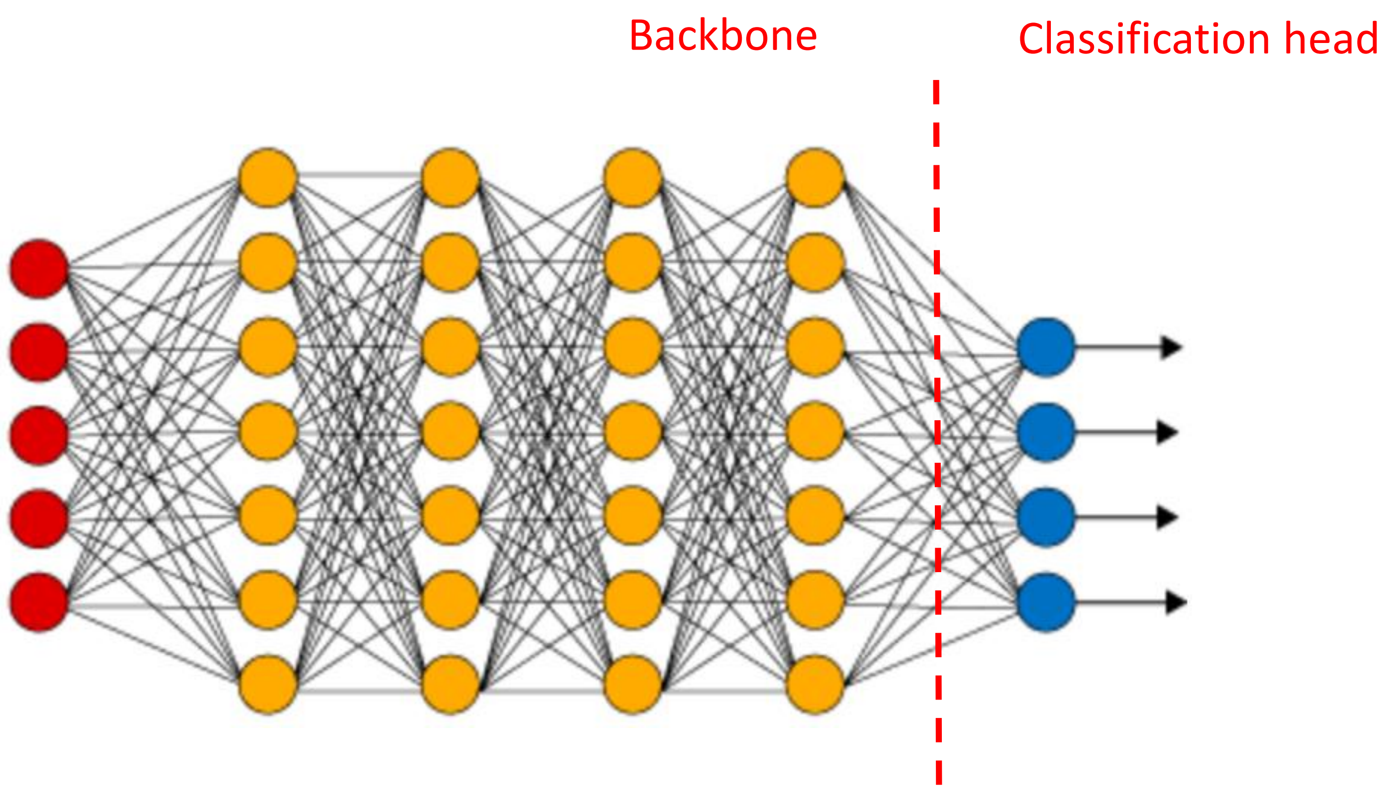
Few shot class incremental learning (FSCIL)



Few shot class incremental learning (FSCIL)



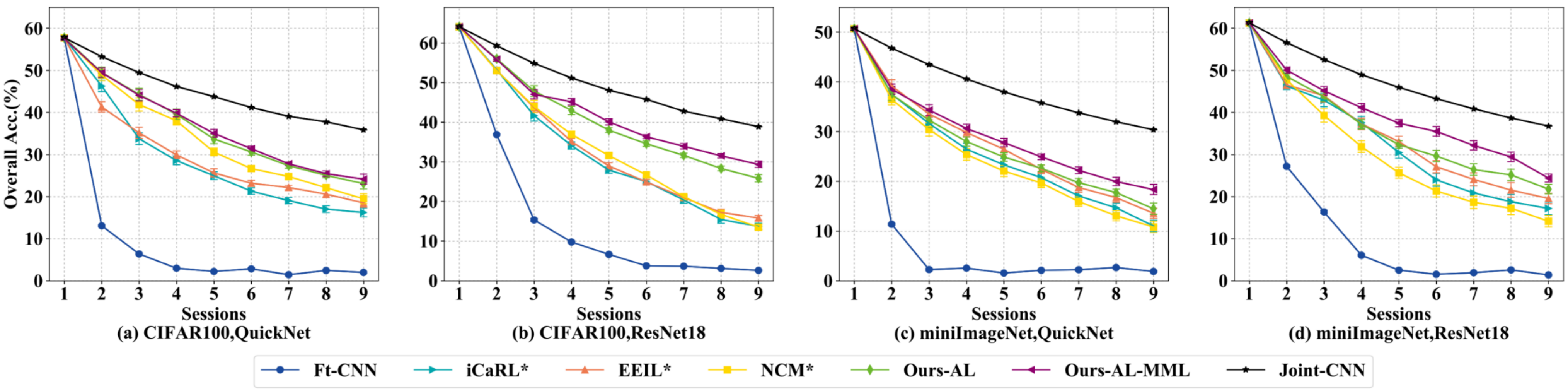
Methods on Few Shot Class Incremental Learning



Baseline (Fine-tune)

- Pre-trained backbone and classification head on the base session
- Update backbone on new data
- Train a classification head on new data, and concatenate with the old head

Few shot class incremental learning (FSCIL)

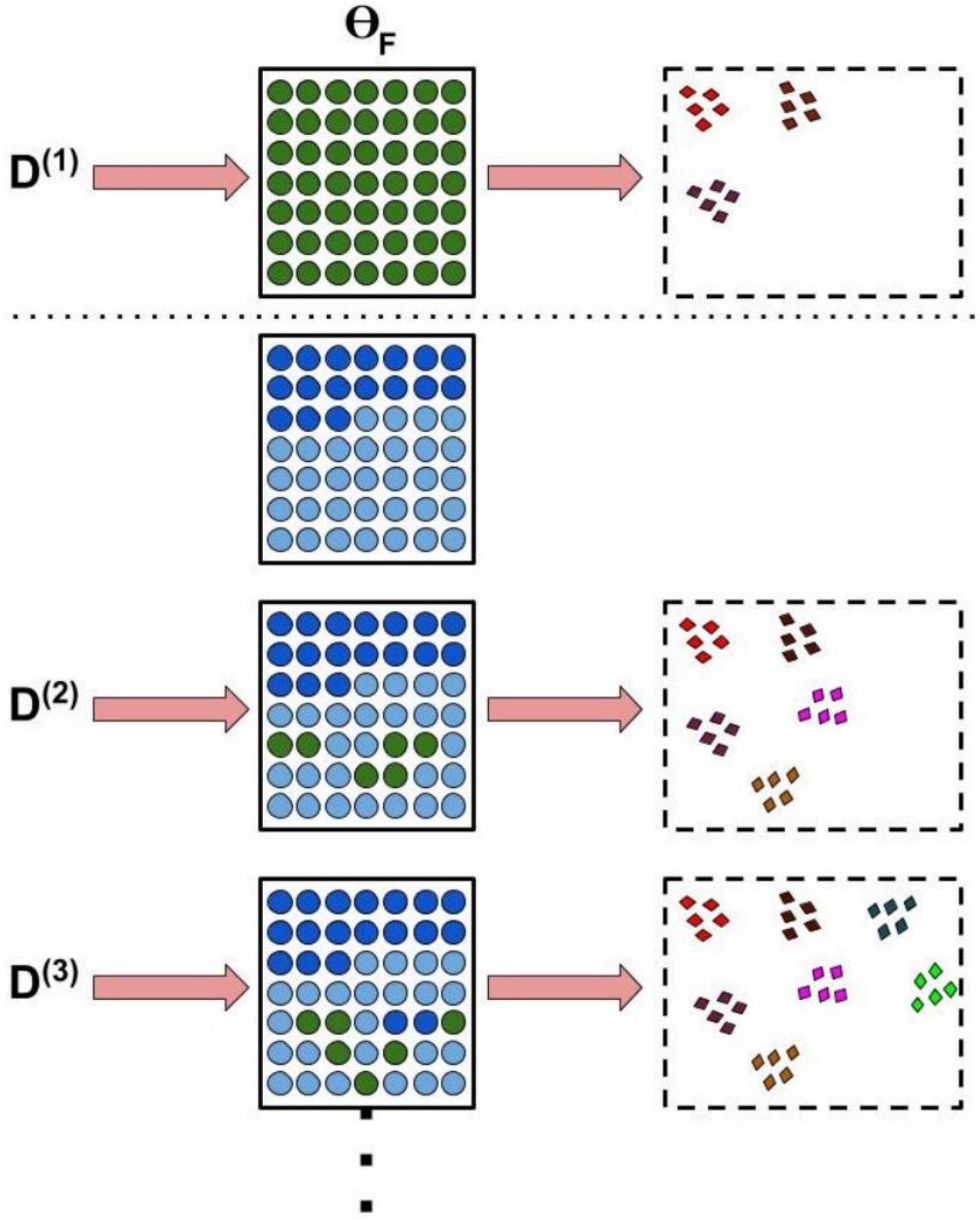


Method	sessions											our relative
	1	2	3	4	5	6	7	8	9	10	11	improvements
Ft-CNN	68.68	44.81	32.26	25.83	25.62	25.22	20.84	16.77	18.82	18.25	17.18	**+9.10**
Joint-CNN	68.68	62.43	57.23	52.80	49.50	46.10	42.80	40.10	38.70	37.10	35.60	upper bound
iCaRL* [32]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	**+5.12**
EEIL* [2]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	**+4.17**
NCM* [13]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	**+6.41**
Ours-AL	**68.68**	**61.01**	**55.35**	**50.01**	**42.42**	**39.07**	**35.47**	**32.87**	**30.04**	**25.91**	**24.85**	+1.43
Ours-AL-MML	**68.68**	**62.49**	**54.81**	**49.99**	**45.25**	**41.40**	**38.35**	**35.36**	**32.22**	**28.31**	**26.28**	

Few-Shot Class-Incremental Learning. [Tao, Xiaoyu, et al. CVPR 2020]

Idea

- Pretrain NN on base session
- Identify and freeze “important” backbone parameters to minimize catastrophic forgetting



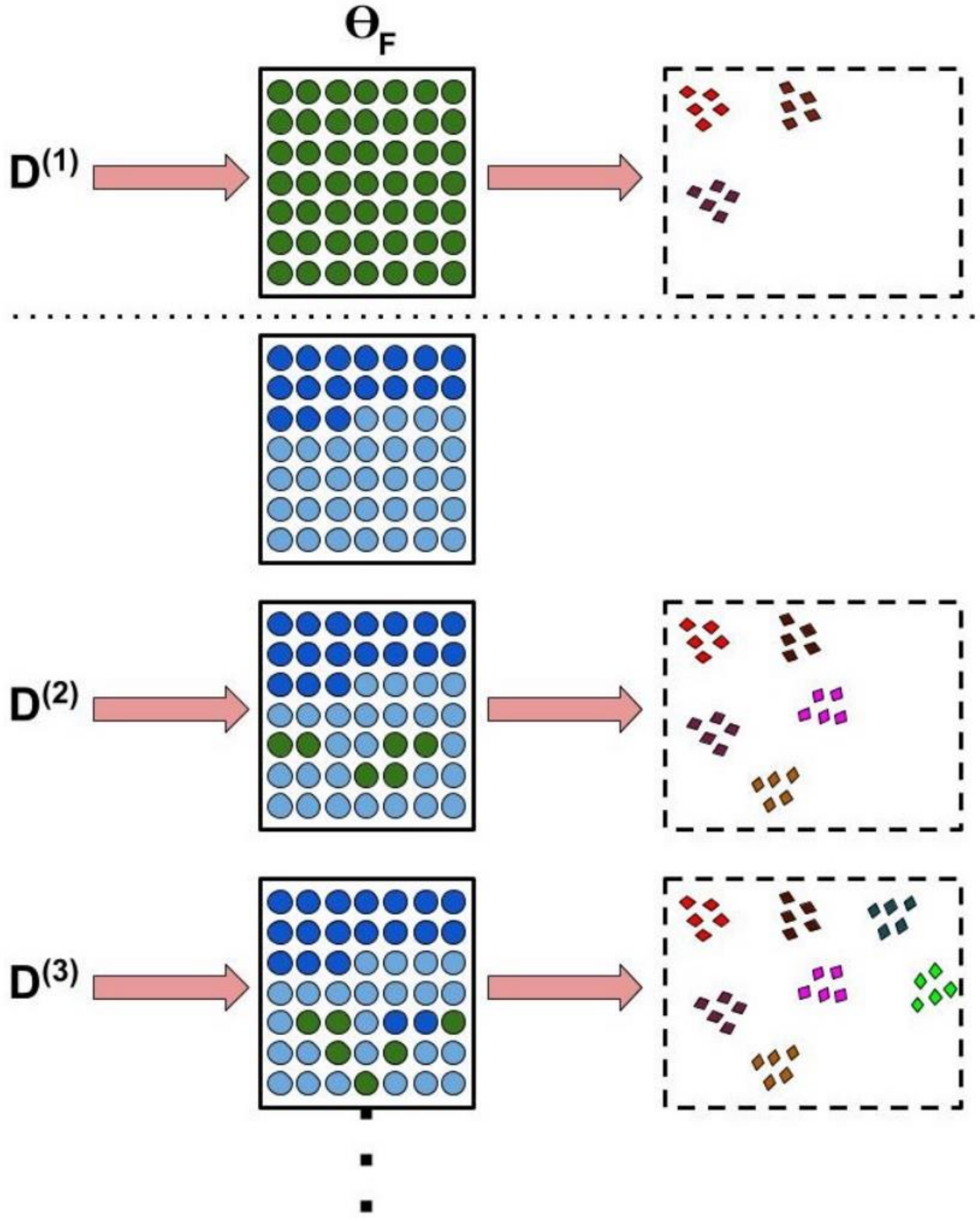
Few-Shot Lifelong Learning [Mazumder et,al, AAAI 2021]

Idea

- Pretrain NN on base session
- Identify and freeze "important" backbone parameters to minimize catastrophic forgetting

How

- Heuristically identify "unimportant" parameters
- Criteria: 10% parameters with lowest magnitude in each layer



Idea

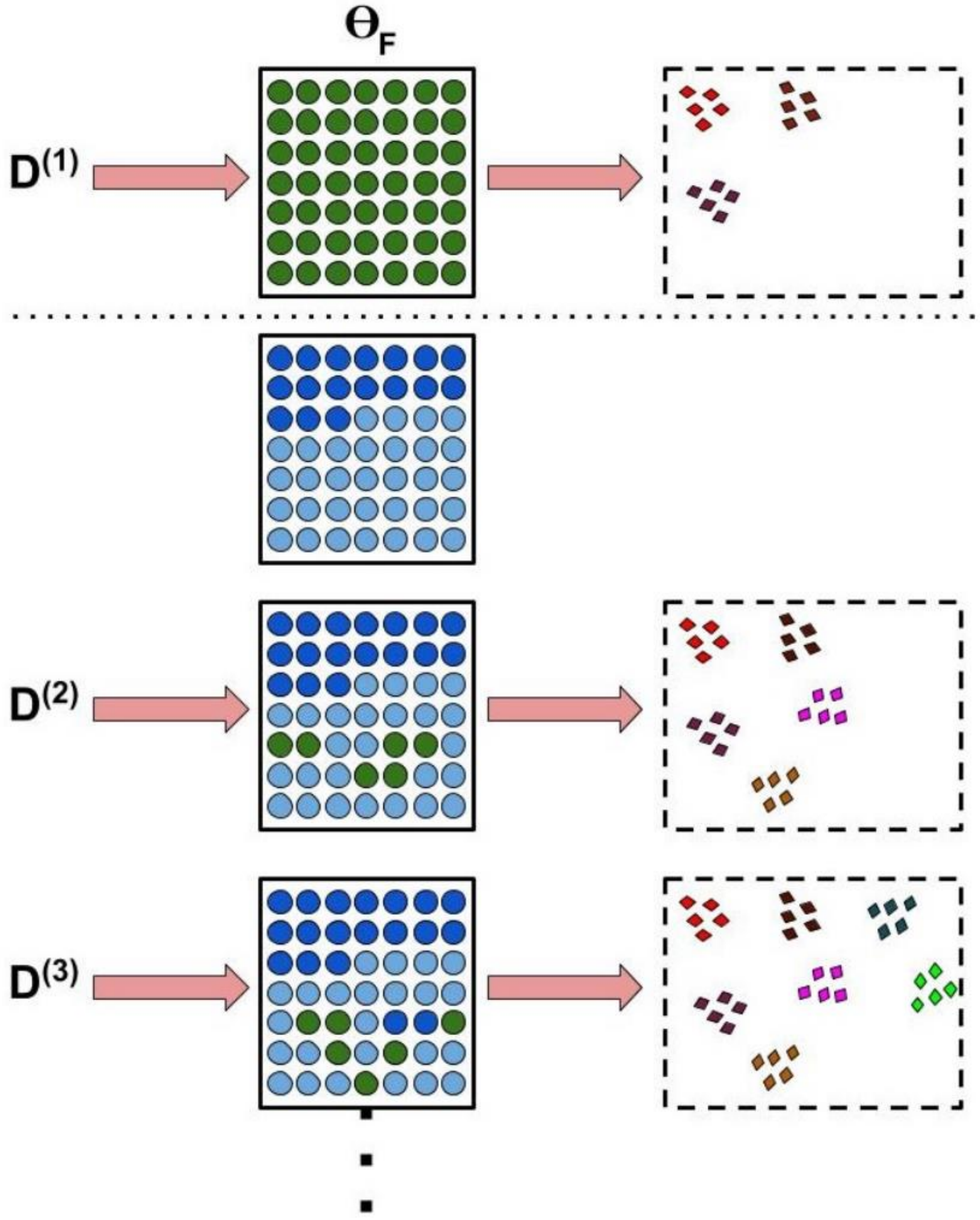
- Pretrain NN on base session
- Identify and freeze "important" backbone parameters to minimize catastrophic forgetting

How

- Heuristically identify "unimportant" parameters
- Criteria: 10% parameters with lowest magnitude in each layer

Weakness

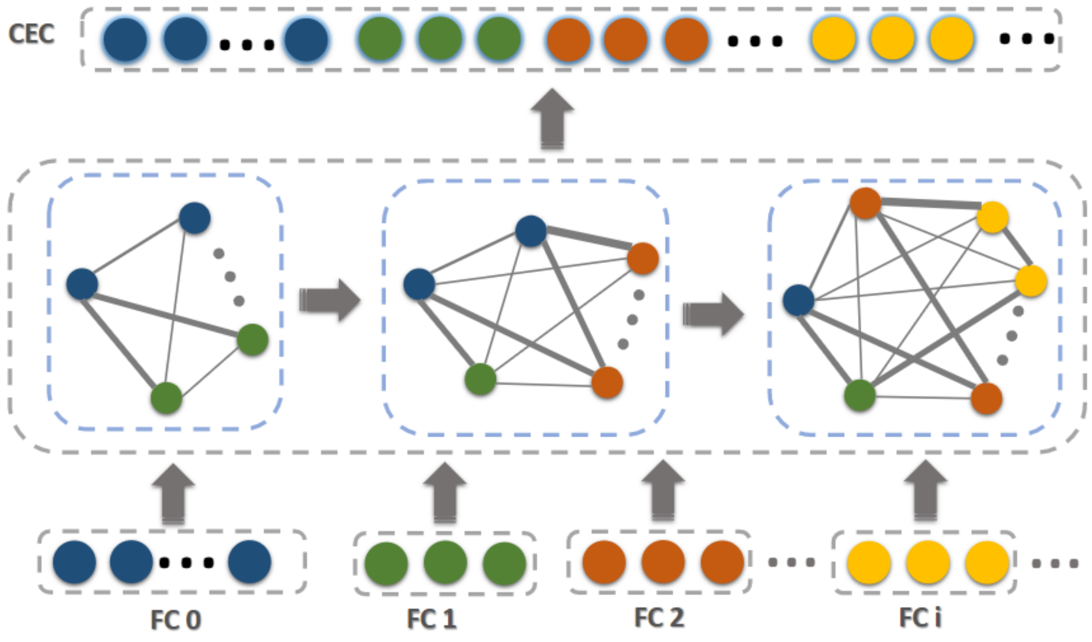
- Hand-engineered method requires extra hyper-parameters tuning
- Not consider any incremental scenarios during training



Few-Shot Lifelong Learning [Mazumder et,al, AAAI 2021]

Idea

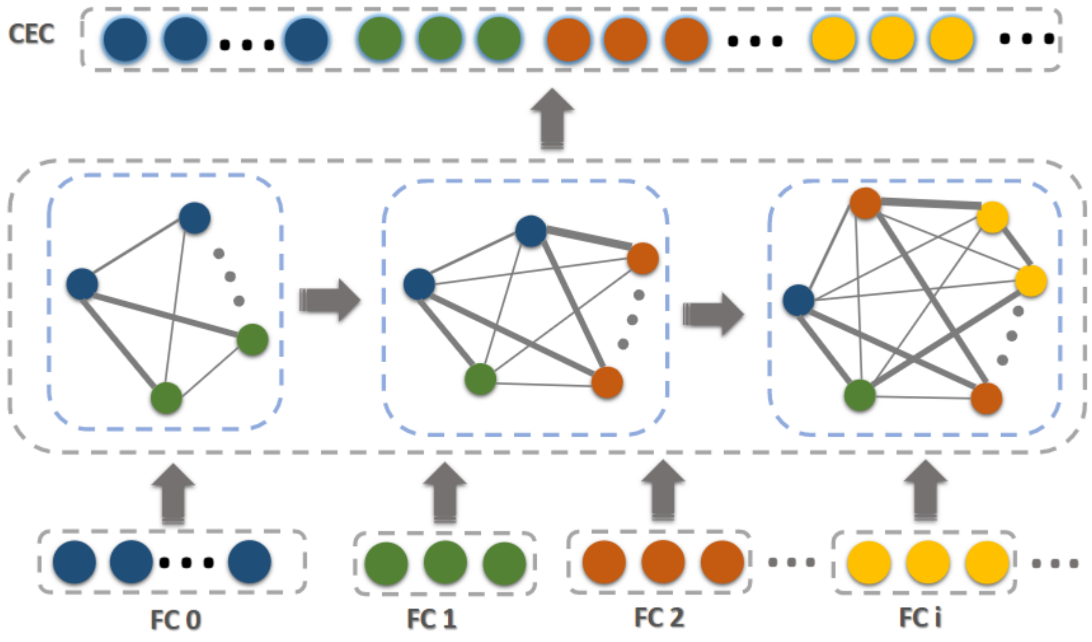
- Freeze backbone to avoid catastrophic forgetting
- Combine old and current classifier heads via GNN to encourage knowledge transfer

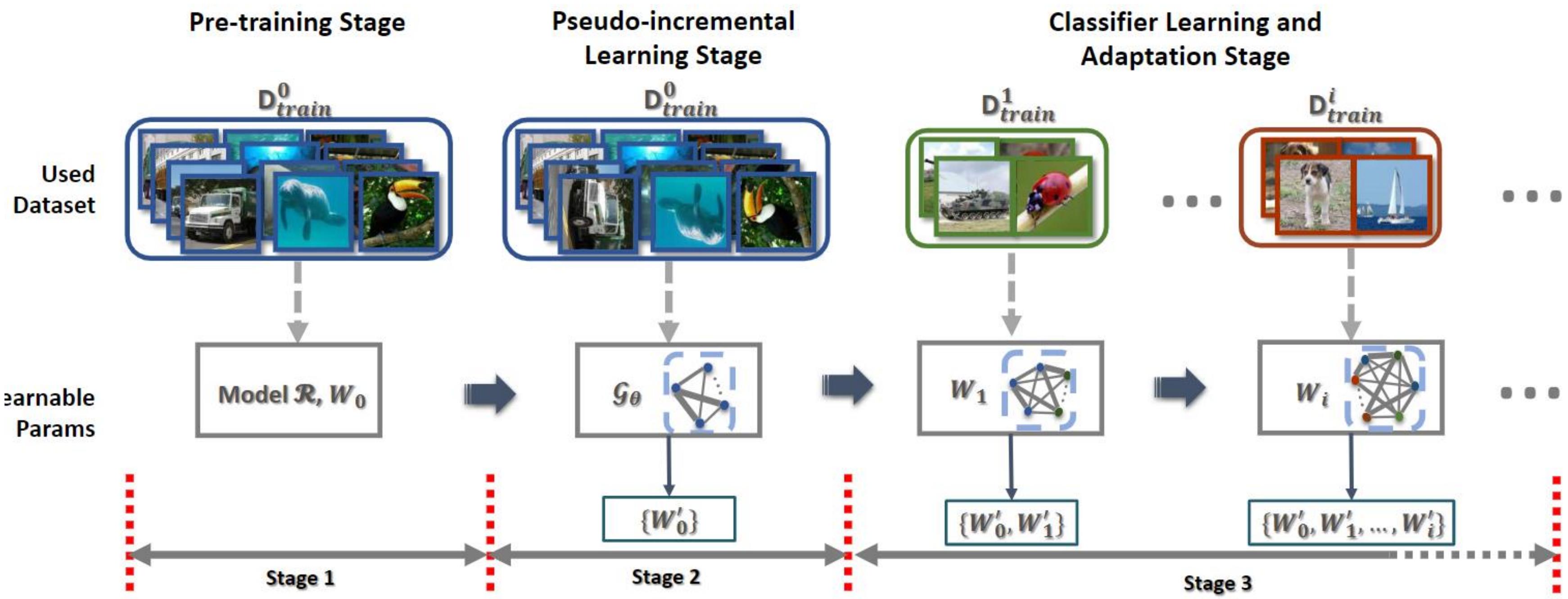


Few-Shot Incremental Learning with Continually Evolved Classifiers [Zhang et,al, CVPR 2021]

Idea

- Freeze backbone to avoid catastrophic forgetting
- Combine old and current classifier heads via GNN to encourage knowledge transfer
- Simulate incremental scenarios in base session training to meta learn GNN via episodic learning





Few-Shot Incremental Learning with Continually Evolved Classifiers [Zhang et,al, CVPR 2021]

Meta learning: Learn to Learn

What is Meta Learning?

- A novel machine learning paradigm: learn the process of learning (learning algorithm)
- Representation engineering → Representation learning (deep learning) → Algorithm or Meta-representation learning (meta learning)

Meta-Learning in Neural Networks: A Survey. [T. Hospedales. TPAMI 2021]

Meta learning: Learn to Learn

What is Meta Learning?

- A novel machine learning paradigm: learn the process of learning (learning algorithm)
- Representation engineering → Representation learning (deep learning) → Algorithm or Meta-representation learning (meta learning)

What is Meta Representation?

- Some aspects of learning algorithm
  - How to design network architecture
  - How to initialize network parameters
  - How to optimize the model (learning rate, regularization, optimizers, full/partial network)
- Meta Representation + A specific Task = Representation

Meta-Learning in Neural Networks: A Survey. [T. Hospedales. TPAMI 2021]

Meta learning: Learn to Learn

What is Meta Learning?

- A novel machine learning paradigm: learn the process of learning (learning algorithm)
- Representation engineering → Representation learning (deep learning) → Algorithm or Meta-representation learning (meta learning)

How to learn Meta representation?

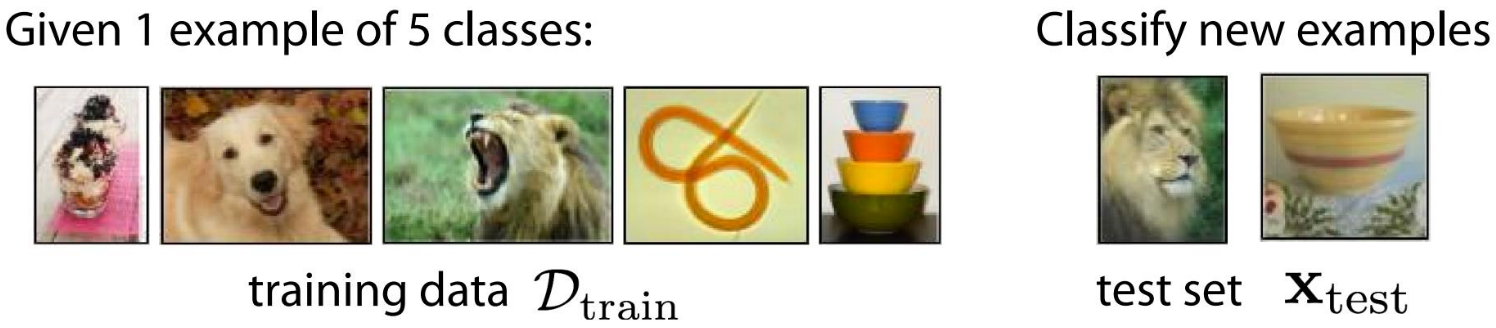
- Key assumption
  - Tasks samples from a task distribution
  - They share same meta-representation (meta-knowledge)
- Two phases
  - Meta-Training: Learn meta-representation across a large-scale of tasks
  - Meta-Testing: Verify whether generalize to a unseen specific task

Meta-Learning in Neural Networks: A Survey. [T. Hospedales. TPAMI 2021]

How does meta-learning work? An example

Few Shot Learning (FSL)

- How to learn new concepts with a few examples
- N-way-K-shot data: N classes, K samples per class



Slide from Stanford CS 330: Deep Multi-Task and Meta Learning, Fall 2021

How does meta-learning work? An example

Simulated episodes

One episode

Under meta-learning framework

- Task = learn to generalize with a few examples
- Task distribution = Simulate a large sale of similar tasks

How does meta-learning work? An example

Simulated episodes

One episode

Support set

Query set

Under meta-learning framework

- Task = learn to generalize with a few examples
- Task distribution = Simulate a large sale of similar tasks
- Goal = How to adapt with support set and perform well on query set

Episodic Learning

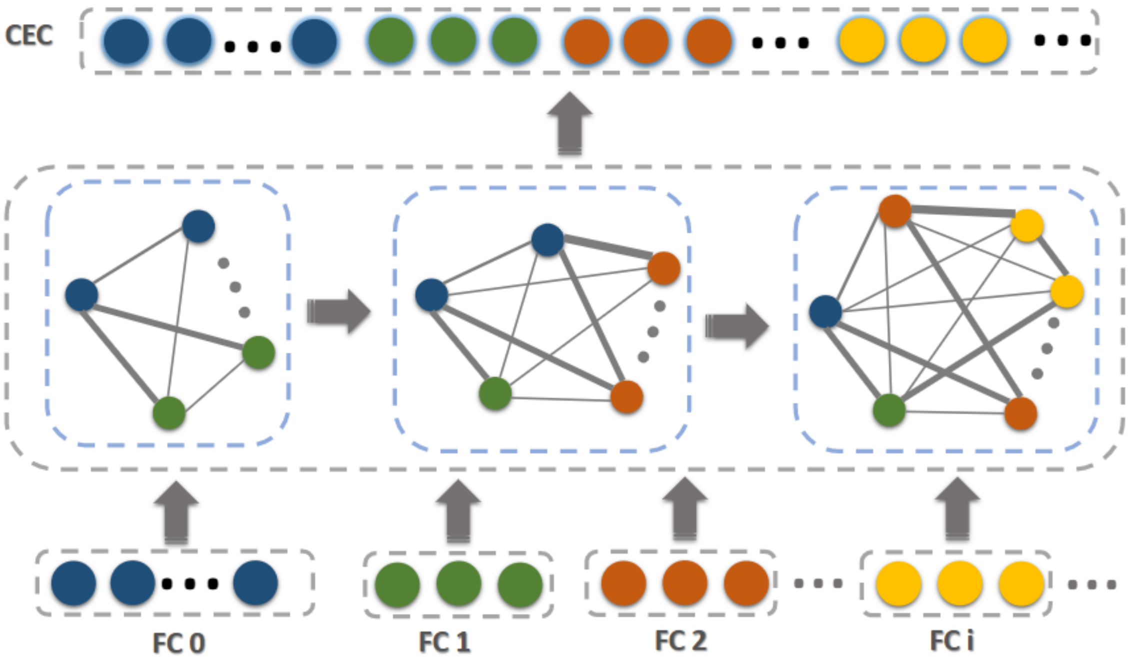
FSL= Learn to generalize to unseen classes with a few examples

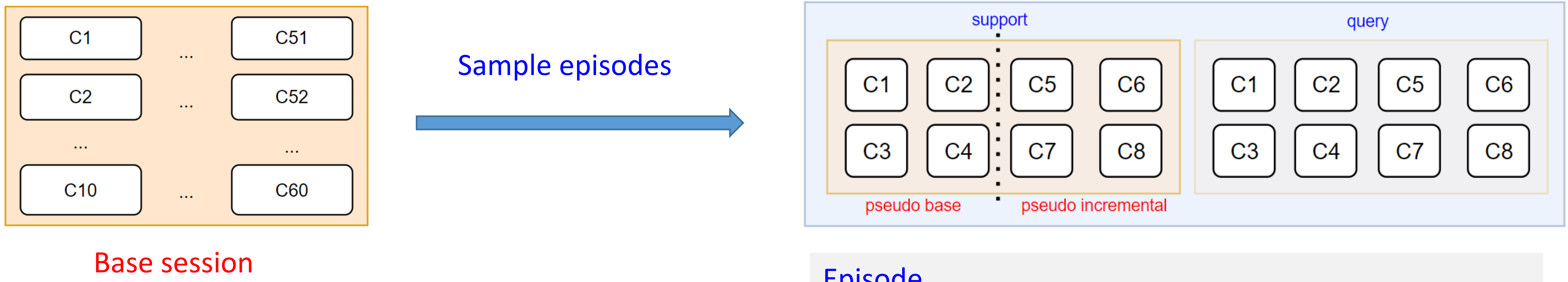
Also use meta learning

FSCIL = Learn to incrementally learn with a few examples

Idea

- Freeze backbone to avoid catastrophic forgetting
- Combine old and current classifier heads via GNN to encourage knowledge transfer
- Simulate incremental scenarios in base session training to meta learn GNN via episodic learning





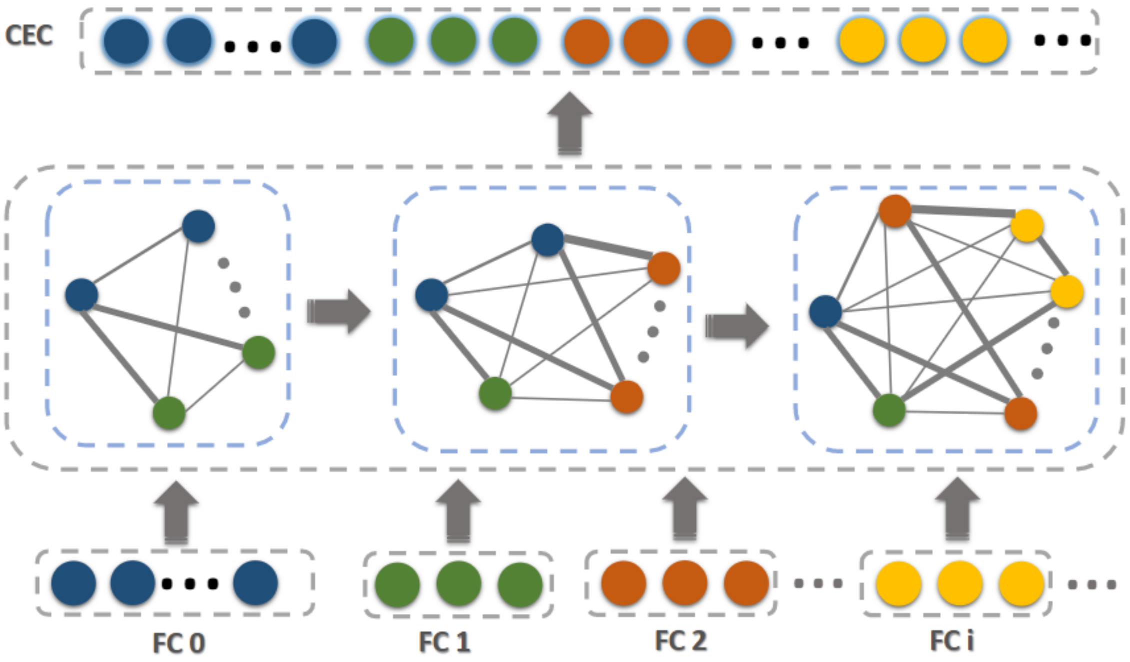
Episode

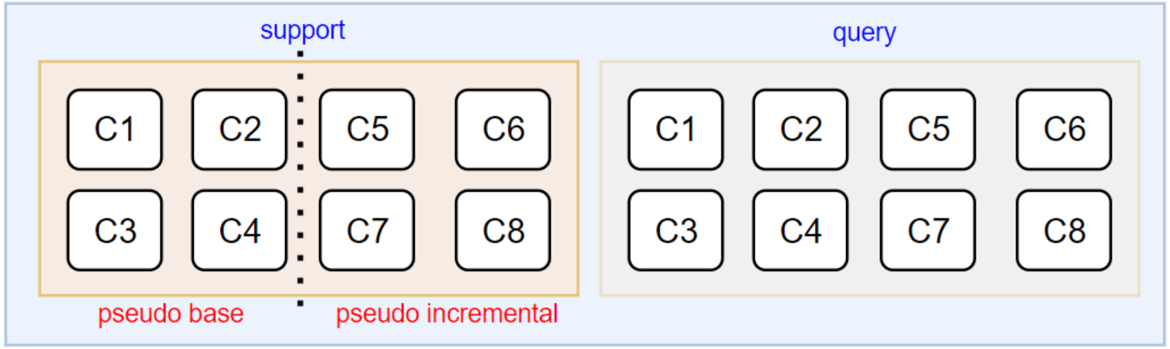
- One pseudo base session + One pseudo incremental session
- The number of classes = 1: 1

Few-Shot Incremental Learning with Continually Evolved Classifiers [Zhang et,al, CVPR 2021]

Strengths

- End-to-End learning without hand-engineered modules
- Design a module to encourage knowledge transfer
- Consider incremental scenarios during training





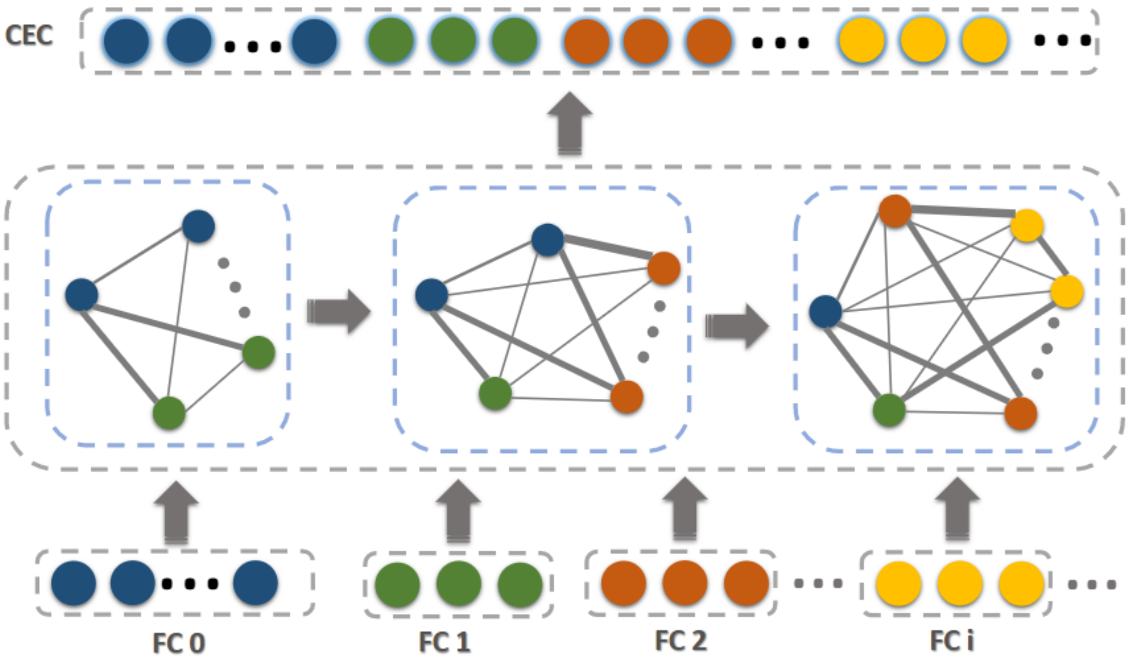
Few-Shot Incremental Learning with Continually Evolved Classifiers [Zhang et,al, CVPR 2021]

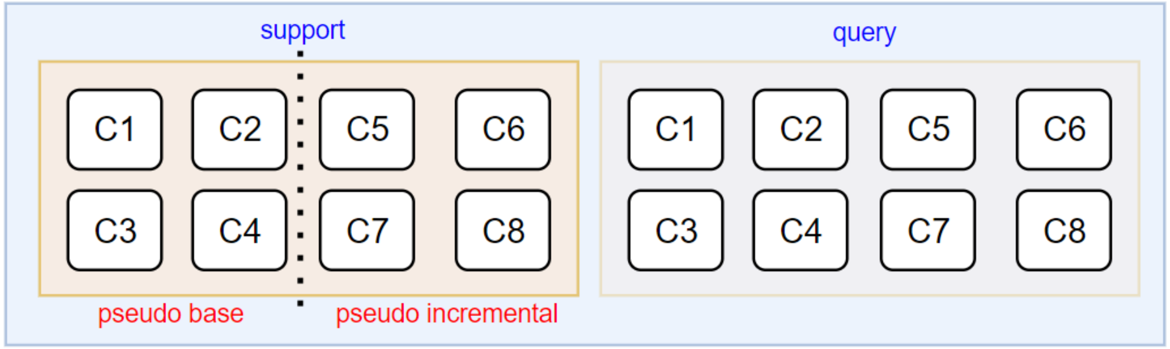
Strengths

- End-to-End learning without hand-engineered modules
- Design a module to encourage knowledge transfer
- Consider incremental scenarios during training

Weakness

- Misalignment between training and evaluation phases due to the episode construction
- Not learn to incrementally learn in a longer horizon
- Fixed backbone limits generalization on new classes





Meta-training: two pseudo sessions
Meta-testing (Test-time)

- multiple incremental sessions
- Imbalance classes between base and incremental sessions

Few-Shot Incremental Learning with Continually Evolved Classifiers [Zhang et,al, CVPR 2021]

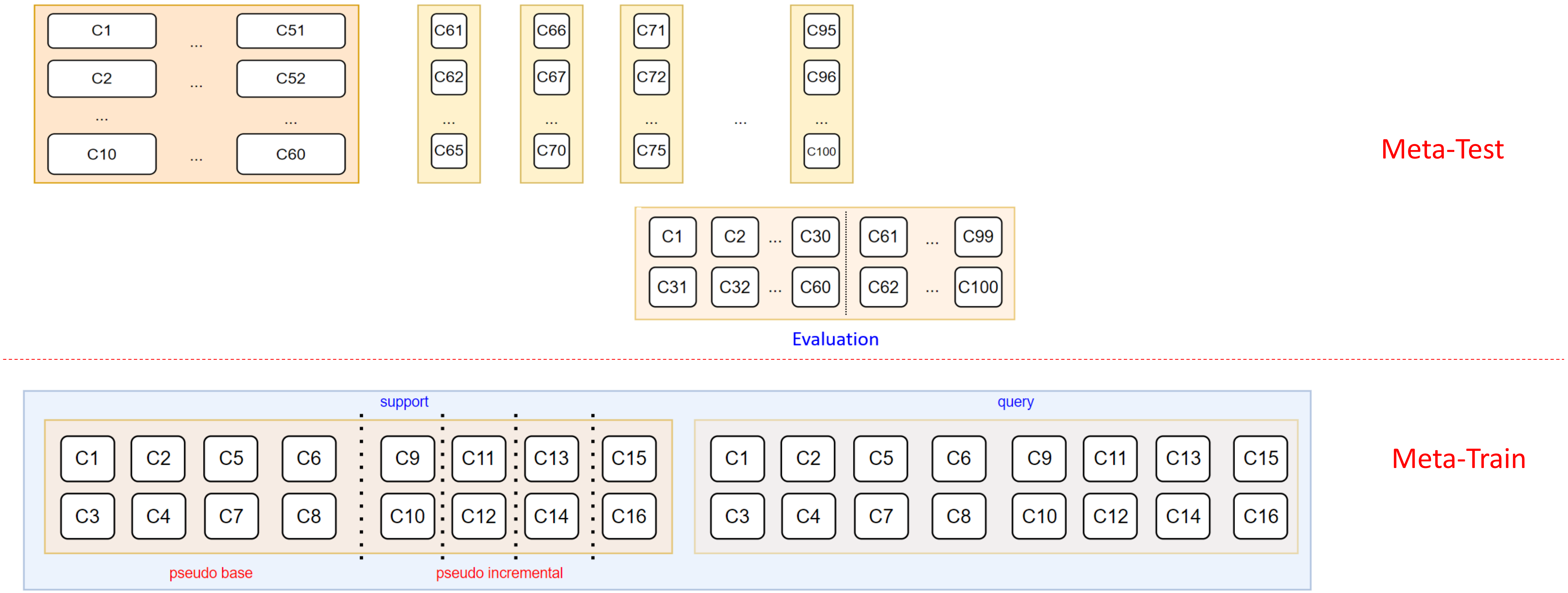
MetaFSCIL: A Meta-Learning Approach for Few-Shot Class Incremental Learning

Zhixiang Chi[1], Li Gu[1], Huan Liu[1,2], Yang Wang[1,3], Yuanhao Yu[1], Jin Tang[1]
[1]Noah's Ark Lab, Huawei Technologies [2]McMaster University, Canada
[3]University of Manitoba, Canada

Idea 1: Aligned Episode construction

Idea

- Align the scenario (incremental learning process) between meta-training and meta-testing



Idea 1: Aligned Episode construction

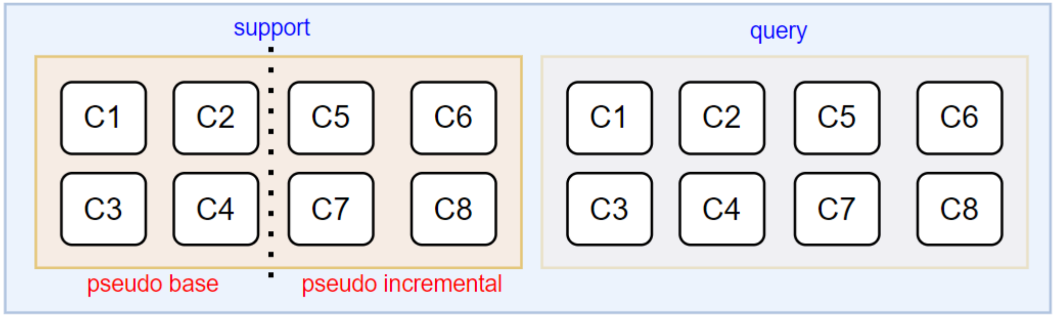
Idea

- Align the scenario (incremental learning process) between meta-training and meta-testing

How

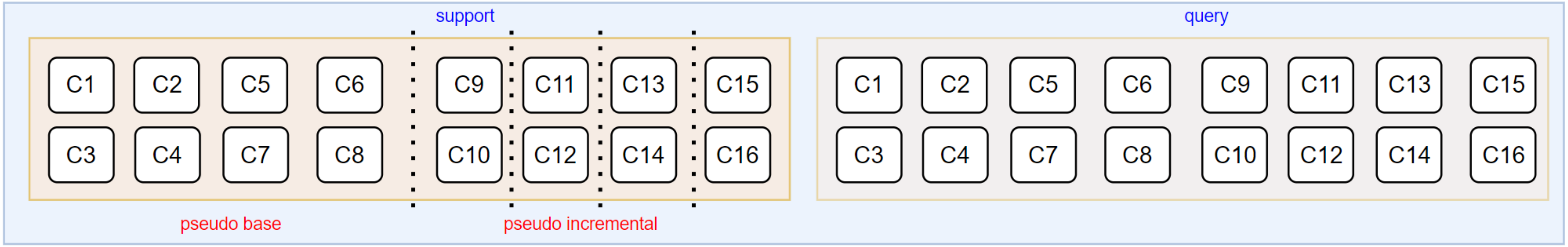
- Sample multiple pseudo incremental sessions
- Introduce class imbalance to simulate combining base and incremental sessions

CEC



CEC: Learn with two sessions
Ours: Learn with a sequence of sessions

Ours



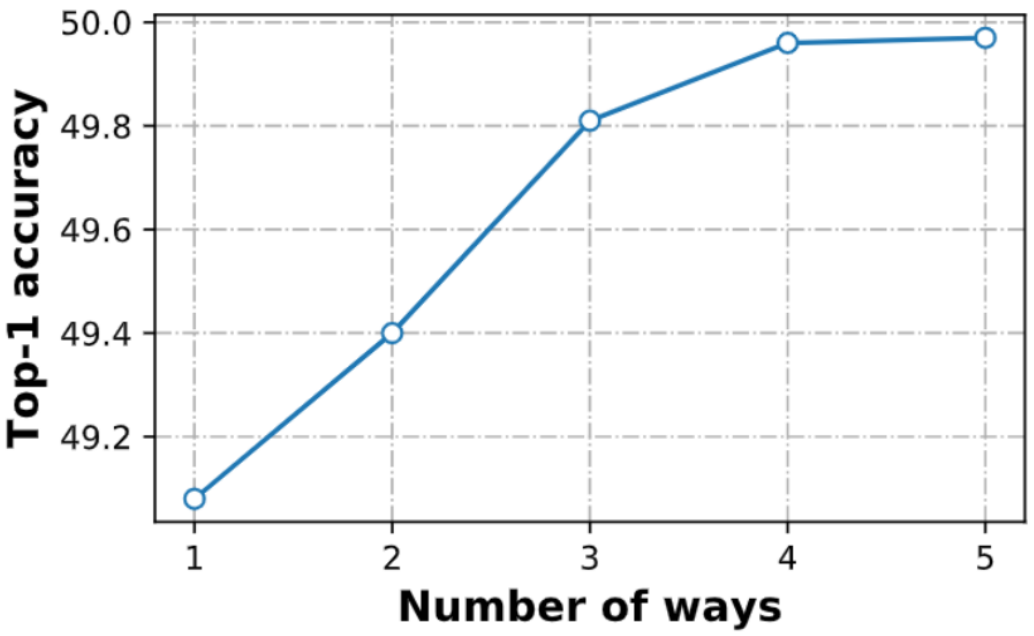
Idea 1: Aligned Episode construction

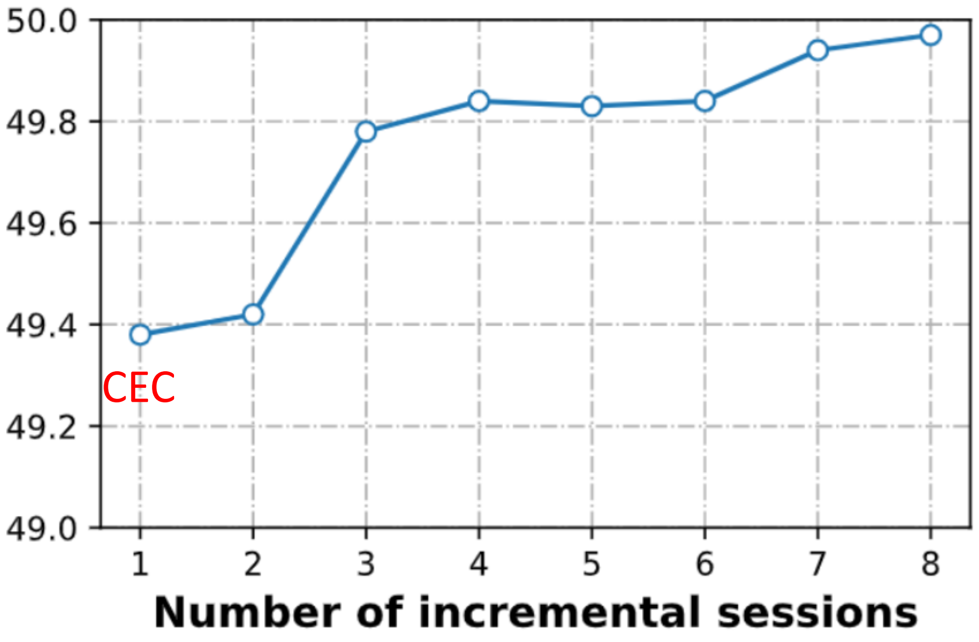
Idea

- Align the scenario (incremental learning process) between meta-training and meta-testing

How

- Sample multiple pseudo incremental sessions
- Introduce class imbalance to simulate combining base and incremental sessions





Idea 2: Meta learned backbone

Idea

- Not handle domain distribution between base and incremental sessions via frozen pretrained backbone (Rep.) in CEC
- Update backbone parameter updates to encourage quick adaptation

Idea 2: Meta learned backbone

Idea

- Not handle domain distribution between base and incremental sessions via frozen pretrained backbone (Rep.) in CEC
- Update backbone parameter updates to encourage quick adaptation

How

- Meta train a backbone initialization[1] (Init.), and update its parameters during incremental sessions at meta-test time

Methods	Sessions (CIFAR100 w/ ResNet20)								
	0	1	2	3	4	5	6	7	8
Baseline (Rep.)	74.33	67.23	63.18	59.24	56.03	53.05	50.66	48.69	46.47
Baseline (Init.)	74.33	66.78	62.30	57.18	54.33	51.68	48.73	46.67	43.80
+Meta-learning (Rep.)	74.45	70.03	65.75	61.69	58.68	55.81	53.68	51.68	49.30
+Meta-learning (Init.)	74.45	70.05	65.97	61.76	58.78	55.92	53.80	51.77	49.41

[1] Model-agnostic meta-learning for fast adaptation of deep networks. [Finn, C., ICML 2017]

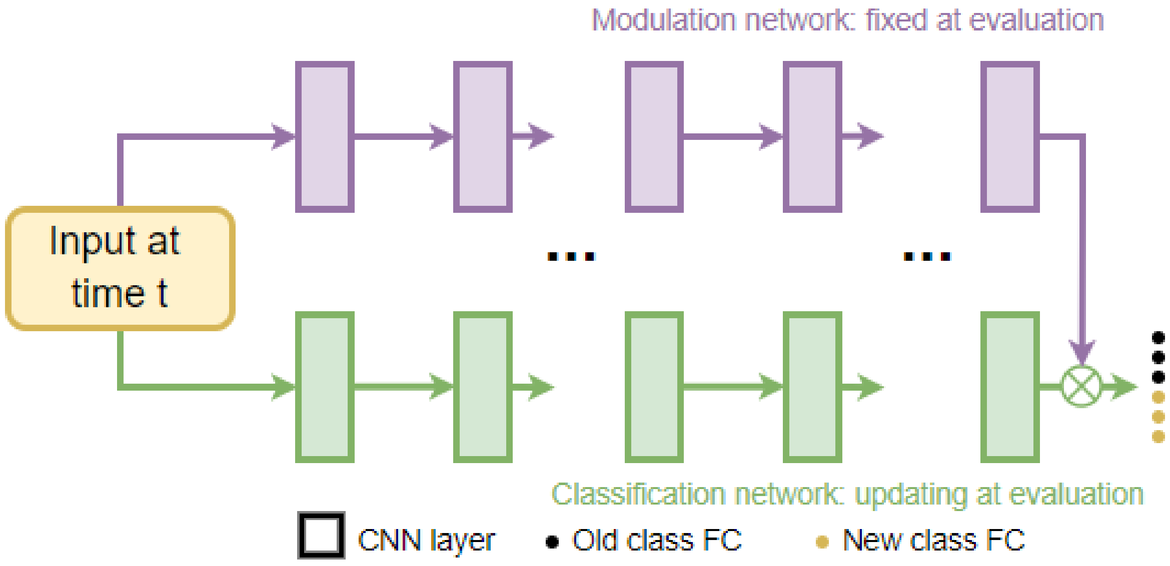
Architecture Design: Modulation Network

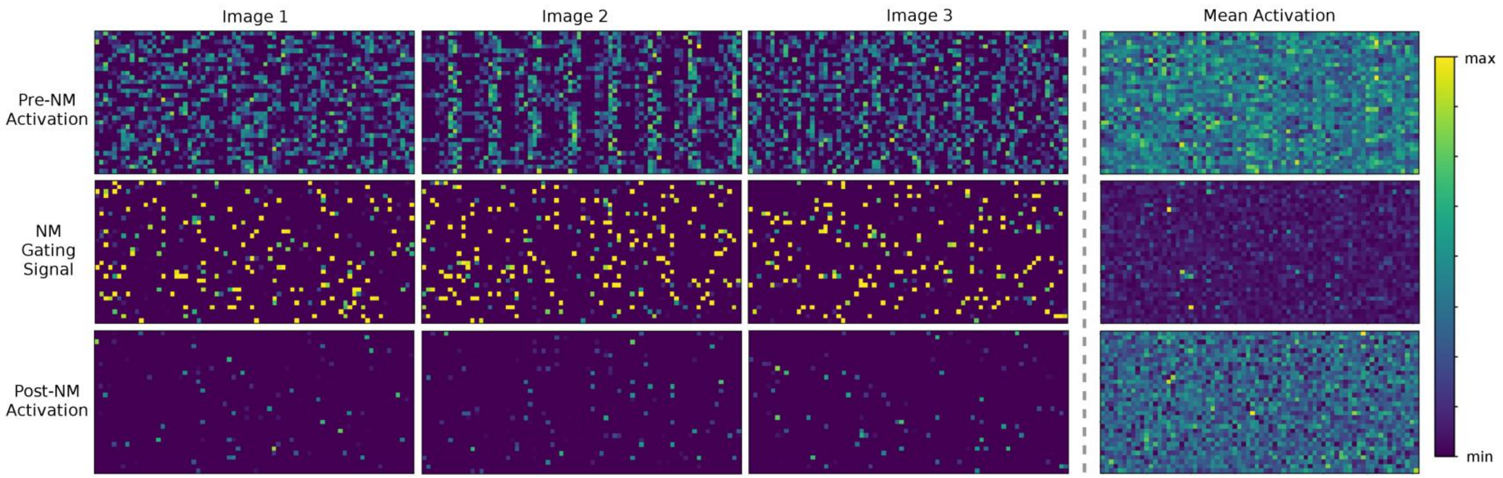
Idea

- Modular/Dynamic architecture
- Implicitly learn to sparsify non-overlapped activations for each task

How

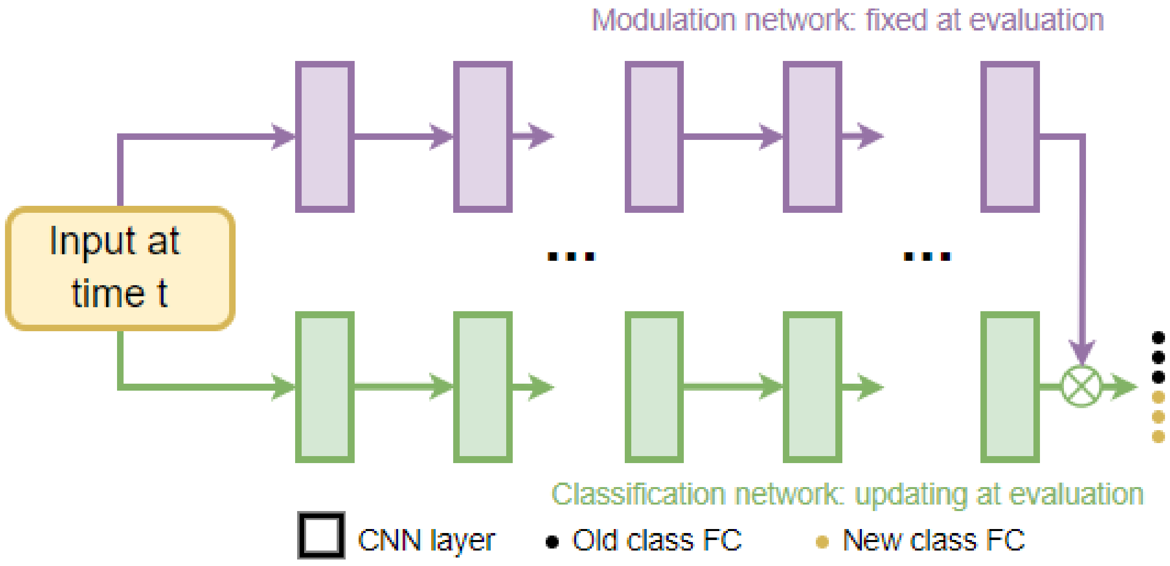
- Generate a mask on last layer's activation via Modulation Network





Learning to Continually Learn, Beaulieu et,al, ECAI 2020

Architecture Design: Modulation Network

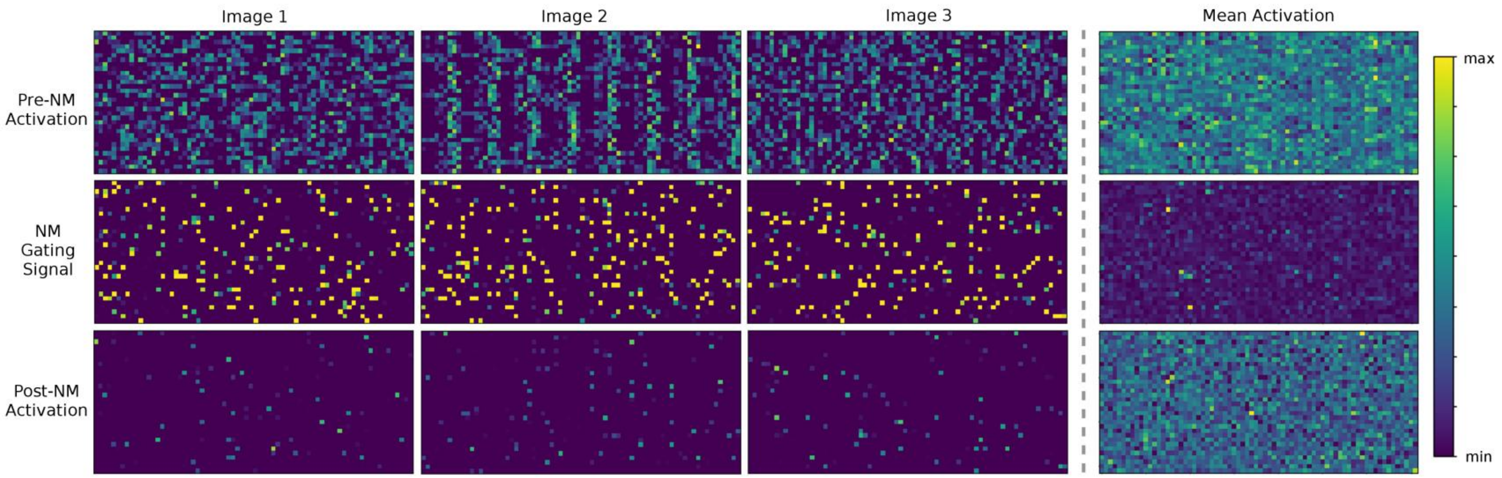


Idea

- Modular/Dynamic architecture
- Implicitly learn to sparsify non-overlapped activations for each task

How

- Generate a mask on last layer's activation via Modulation Network to



Weakness

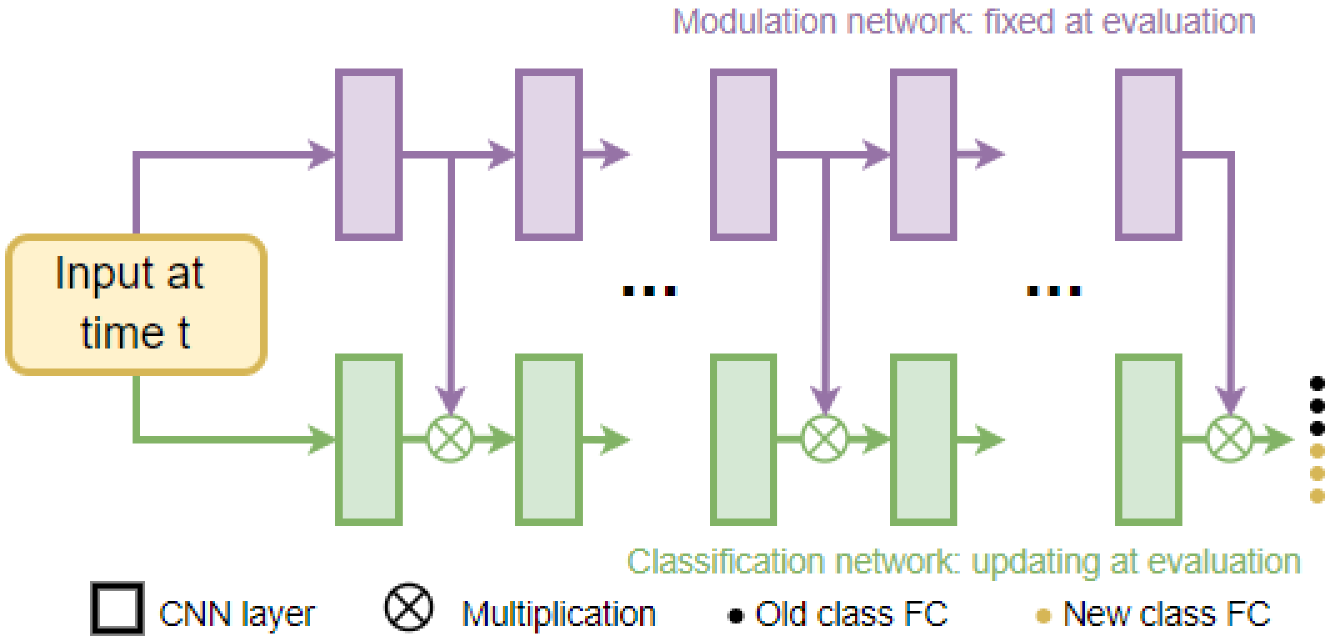
- Diminishing modulation effect with deeper networks
- Not old knowledge in Modulation

Learning to Continually Learn, Beaulieu et,al, ECAI 2020

Idea 3: Bi-directional Guided Modulation

Our improvement A

- Modulate both early and later layers in classification network



Methods	Sessions (CIFAR100 w/ ResNet20)								
	0	1	2	3	4	5	6	7	8
Baseline (Rep.)	74.33	67.23	63.18	59.24	56.03	53.05	50.66	48.69	46.47
Baseline (Init.)	74.33	66.78	62.30	57.18	54.33	51.68	48.73	46.67	43.80
+Meta-learning (Rep.)	74.45	70.03	65.75	61.69	58.68	55.81	53.68	51.68	49.30
+Meta-learning (Init.)	74.45	70.05	65.97	61.76	58.78	55.92	53.80	51.77	49.41
+Modulation (Last)	74.46	70.08	66.65	62.06	58.88	55.58	53.28	51.12	48.34
+Modulation (Uniform)	74.49	70.08	**67.00**	62.45	59.38	56.29	54.08	52.02	49.67

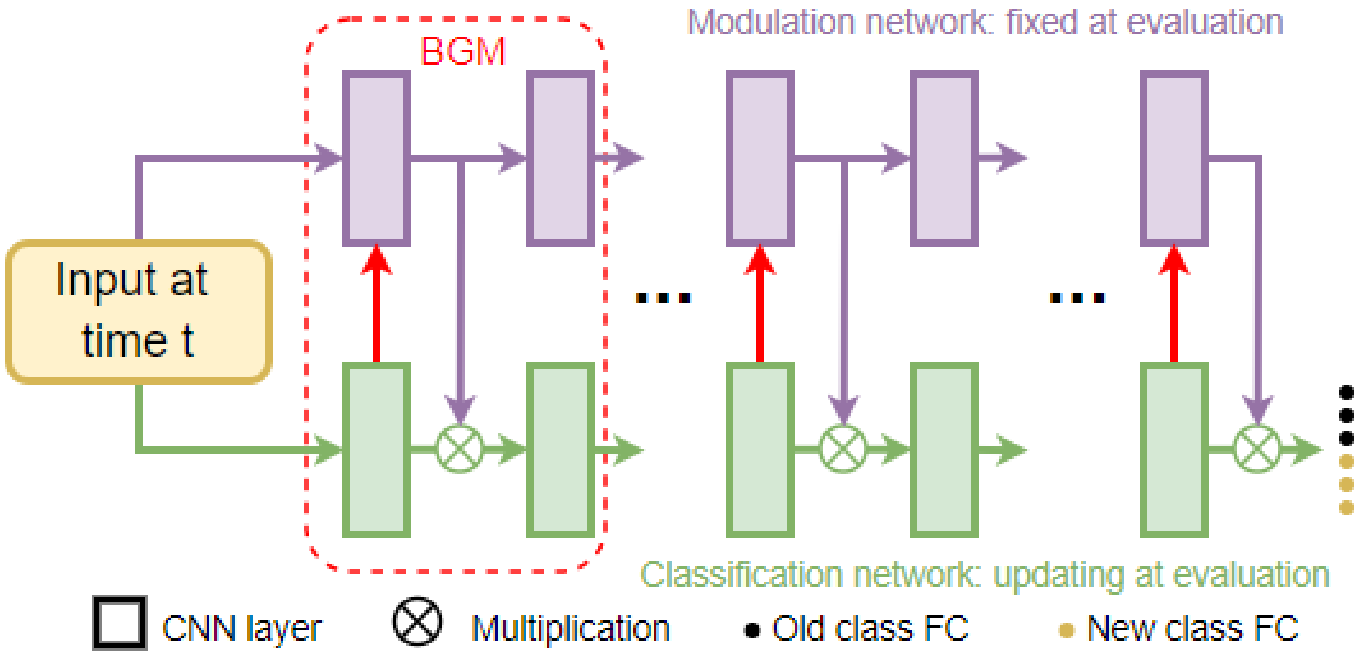
Idea 3: Bi-directional Guided Modulation

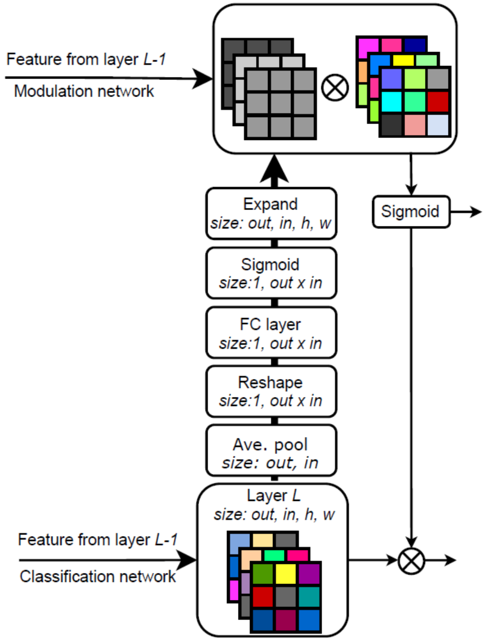
Our improvement A

- Modulate both early and later layers in classification network

Our improvement B

- Incorporate old knowledge
- Add extra connections to guide Modulation Network





Methods	Sessions (CIFAR100 w/ ResNet20)								
	0	1	2	3	4	5	6	7	8
Baseline (Rep.)	74.33	67.23	63.18	59.24	56.03	53.05	50.66	48.69	46.47
Baseline (Init.)	74.33	66.78	62.30	57.18	54.33	51.68	48.73	46.67	43.80
+Meta-learning (Rep.)	74.45	70.03	65.75	61.69	58.68	55.81	53.68	51.68	49.30
+Meta-learning (Init.)	74.45	70.05	65.97	61.76	58.78	55.92	53.80	51.77	49.41
+Modulation (Last)	74.46	70.08	66.65	62.06	58.88	55.58	53.28	51.12	48.34
+Modulation (Uniform)	74.49	70.08	**67.00**	62.45	59.38	56.29	54.08	52.02	49.67
+BGM (Full model)	**74.50**	**70.10**	66.84	**62.77**	**59.48**	**56.52**	**54.36**	**52.56**	**49.97**

Experiment Result: State-of-the-art

Methods	Venue	Sessions (MiniImageNet w/ ResNet18)									Average Acc	Final Impro.
		0	1	2	3	4	5	6	7	8		
TOPIC [26]	CVPR2020	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	+24.77
Zhu *et.al* [32]	CVPR2021	61.45	63.80	59.53	55.53	52.50	49.60	46.69	43.79	41.92	52.75	+7.27
Cheraghian *et.al* [4]	ICCV2021	61.40	59.80	54.20	51.69	49.45	48.00	45.20	43.80	42.1	50.63	+7.09
CEC [31]	CVPR2021	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	+1.56
MetaFSCIL (ours)	-	**72.04**	**67.94**	**63.77**	**60.29**	**57.58**	**55.16**	**52.9**	**50.79**	**49.19**	**58.85**	

Methods	Venue	Sessions (CIFAR100) w/ ResNet20									Average Acc	Final Impro.
		0	1	2	3	4	5	6	7	8		
TOPIC [26]	CVPR2020	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	+20.6
Zhu *et.al* [32]	CVPR2021	64.10	65.86	61.36	57.34	53.69	50.75	48.58	45.66	43.25	54.51	+6.72
Cheraghian *et.al* [4]	ICCV2021	62.00	57.00	56.7	52.00	50.60	48.8	45.00	44.00	41.64	50.86	+8.33
CEC [31]	CVPR2021	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	+0.83
MetaFSCIL (ours)	-	**74.50**	**70.10**	**66.84**	**62.77**	**59.48**	**56.52**	**54.36**	**52.56**	**49.97**	**60.79**	

Methods	Venue	Sessions (CUB200) w/ ResNet18											Average Acc	Final Impro.
		0	1	2	3	4	5	6	7	8	9	10		
TOPIC [26]	CVPR2020	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92	+26.36
Zhu *et.al* [32]	CVPR2021	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	+15.31
Cheraghian *et.al* [4]	ICCV2021	68.78	59.37	59.32	54.96	52.58	49.81	48.09	46.32	44.33	43.43	43.23	51.84	+9.41
CEC [31]	CVPR2021	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	+0.36
MetaFSCIL (ours)	-	**75.90**	**72.41**	**68.78**	**64.78**	**62.96**	**59.99**	**58.30**	**56.85**	**54.78**	**53.82**	**52.64**	**61.92**	

Experiment Result: Visualization

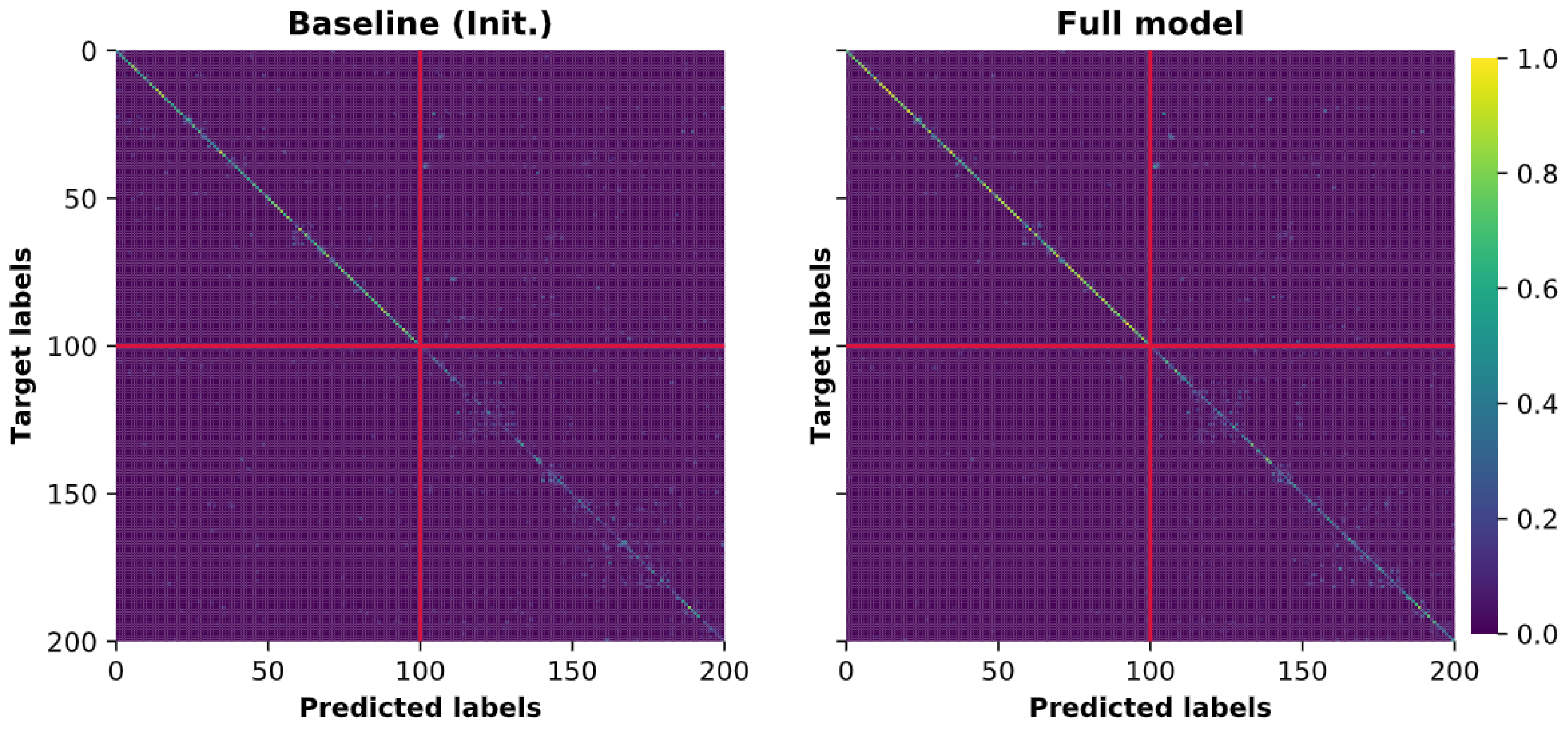


Figure 4. **Class-wise performance on CUB200 dataset.** The confusion matrices show that our method significantly improves the baseline for both *base* and *novel* classes (separated by red line).

Experiment Result: From Meta Learning Perspective

CEC

- Pre-trained backbone (representation) without updates during continual sessions
- Misaligned episode construction
- meta-learned classifier heads refinement (GNN)

Experiment Result: From Meta Learning Perspective

CEC

- Pre-trained backbone (representation) without updates during continual sessions
- Misaligned episode construction
- meta-learned classifier heads refinement (GNN)

Ours

- meta-learned backbone (initialization) + quick adaptation during continual sessions
- Aligned episode construction
- meta-learned Bi-directional Guided Modulation

Ours is a full meta learning based FSCIL method

Experiment Result: From Meta Learning Perspective

CEC

- Pre-trained backbone (representation) without updates during continual sessions
- Misaligned episode construction
- meta-learned classifier heads refinement (GNN)

Ours

- meta-learned backbone (initialization) + quick adaptation during continual sessions
- Aligned episode construction
- meta-learned Bi-directional Guided Modulation

Ours is a full meta learning based FSCIL method

Methods	MiniImageNet	CIFAR100	CUB200
CEC [31]	47.63	49.14	52.28
MetaFSCIL + CEC	48.95	49.71	52.64

Table 3. **Integration of our meta-learned backbone with CEC.**

Summary

Part 1: Introduction on Lifelong Learning

- Lifelong / continual learning can enable knowledge accumulation and quick adaptation in dynamically changing environments
- The main challenge in Lifelong learning is catastrophic forgetting (Plasticity – Stability dilemma)
- Three common strategies to address catastrophic forgetting
- Class incremental learning (CIL) is one category of Lifelong learning

Summary

Part 1: Introduction on Lifelong Learning

- Lifelong / continual learning can enable knowledge accumulation and quick adaptation in dynamically changing environments
- The main challenge in Lifelong learning is catastrophic forgetting (Plasticity – Stability dilemma)
- Three common strategies to address catastrophic forgetting
- Class incremental learning (CIL) is one category of Lifelong learning

Part 2: Introduction on Few shot class incremental learning

- Few shot class incremental learning (FSCIL) is more close to the realistic but more challenging
- Two recent works on FSCIL

Summary

Part 1: Introduction on Lifelong Learning

- Lifelong / continual learning can enable knowledge accumulation and quick adaptation in dynamically changing environments
- The main challenge in Lifelong learning is catastrophic forgetting (Plasticity – Stability dilemma)
- Three common strategies to address catastrophic forgetting
- Class incremental learning (CIL) is one category of Lifelong learning

Part 2: Introduction on Few shot class incremental learning

- Few shot class incremental learning (FSCIL) is more close to the realistic but more challenging
- Two recent works on FSCIL

Part 3: Our CVPR 2022 paper

- A full meta learning based method on FSCIL

Q & A

Methodologies on Few Shot Learning