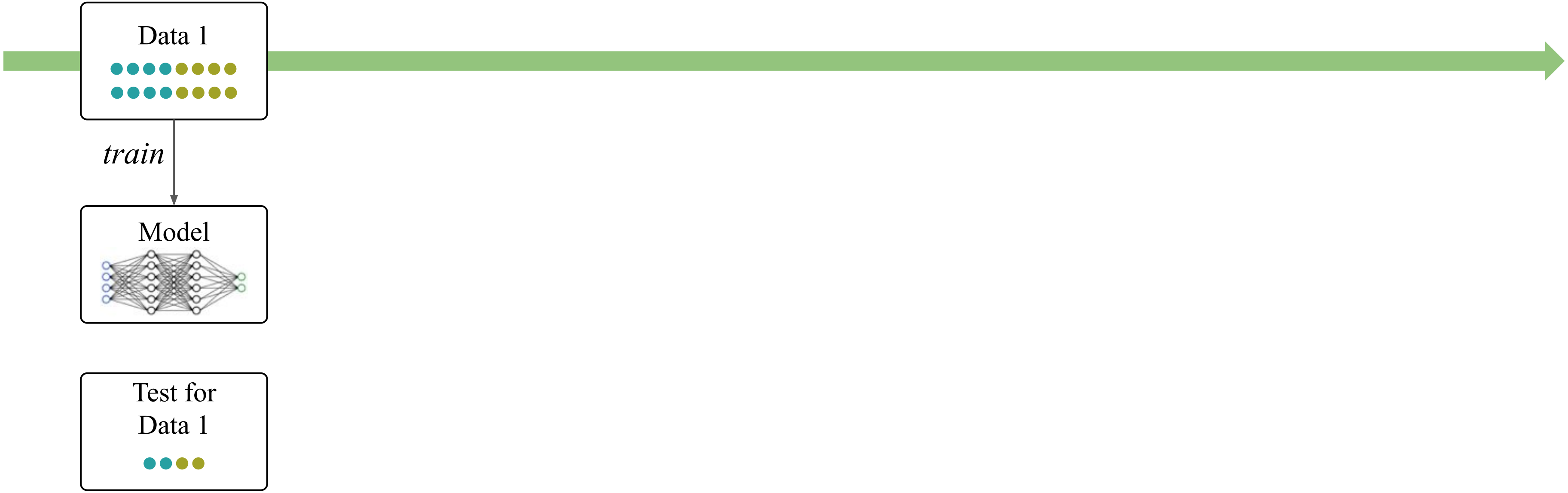
Online Hyperparameter Optimization for Class-Incremental Learning

Yaoyao Liu[1] Yingying Li[2] Bernt Schiele[1] Qianru Sun[3]

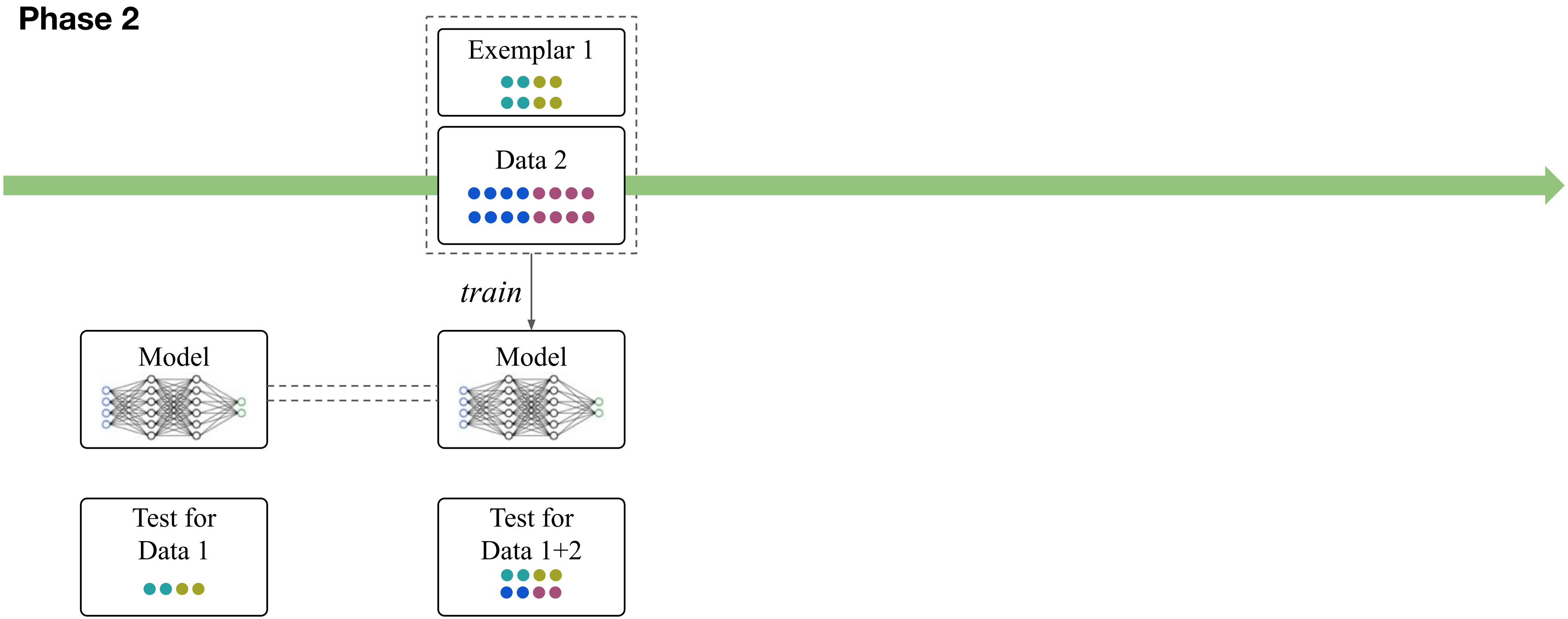
[1]Max Planck Institute for Informatics [2]California Institute of Technology [3]Singapore Management University

Research background: Class-Incremental Learning (CIL)

Phase 1

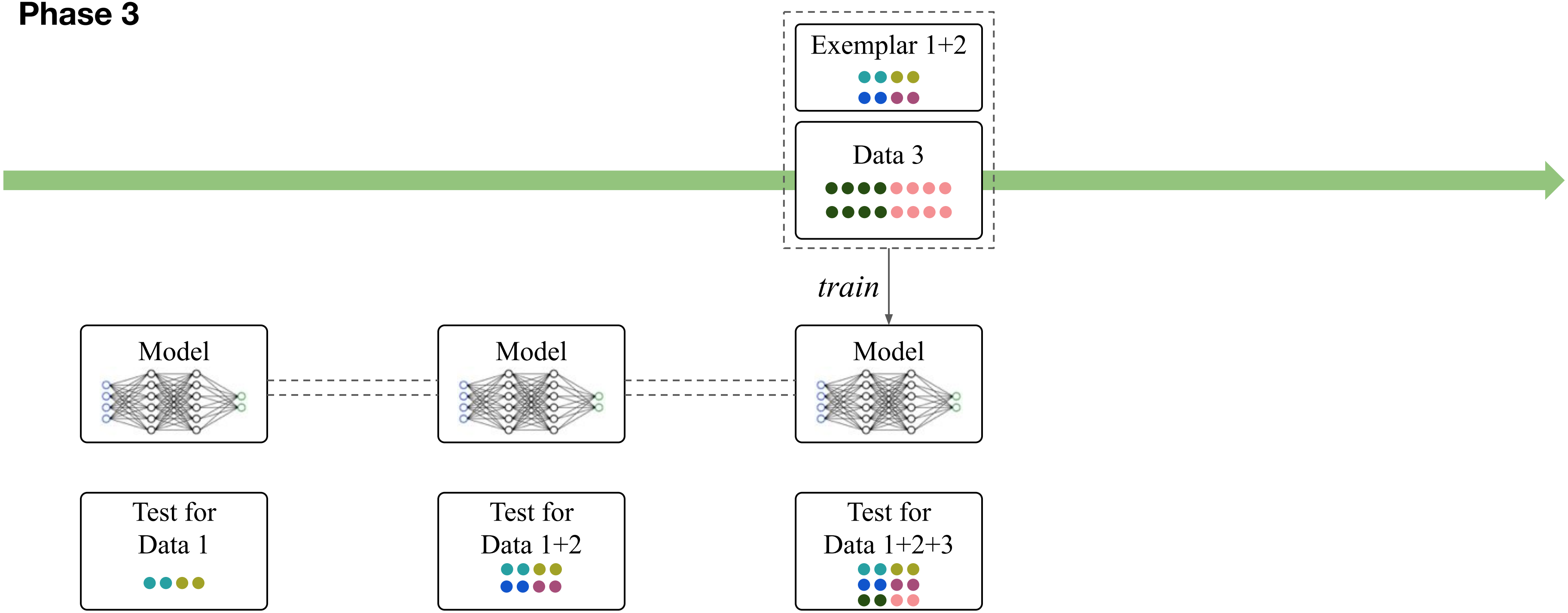


Research background: Class-Incremental Learning (CIL)



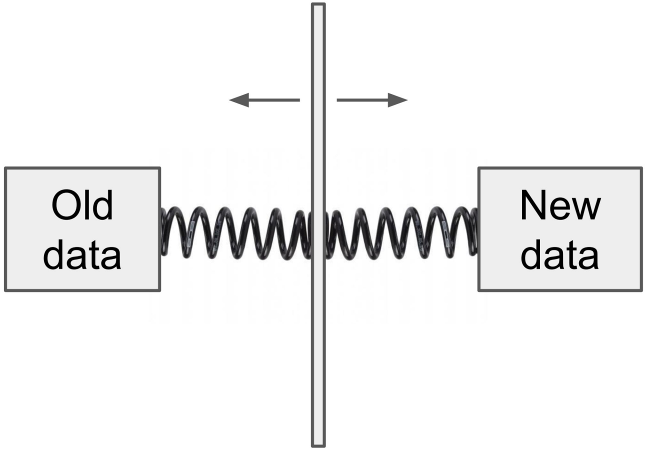
Research background: Class-Incremental Learning (CIL)

Phase 3



max planck institut
informatik

Main Challenge: the stability-plasticity trade-off



Higher stability
weakens the model from learning the data of new classes

Higher plasticity
results in the forgetting of old classes (i.e., “**catastrophic forgetting**”)

Image from: https://www.slideshare.net/xavigiro/lifelong-incremental-learning-d3l2-2017-upc-deep-learning-for-computer-vision

Different data-receiving settings require different stability/plasticity

Reference
[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[2] Castro, Francisco M., et al. "End-to-end incremental learning." ECCV 2018.

Different data-receiving settings require different stability/plasticity

E.g., CIFAR-100 5-phase

The "training-from-half" (TFH)[1] setting requires higher stability

50 classes	10 classes	10 classes	10 classes	10 classes	10 classes
Phase 0	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5

Reference
[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[2] Castro, Francisco M., et al. "End-to-end incremental learning." ECCV 2018.

Different data-receiving settings require different stability/plasticity

E.g., CIFAR-100 5-phase

The "training-from-half" (TFH)[1] setting requires higher stability

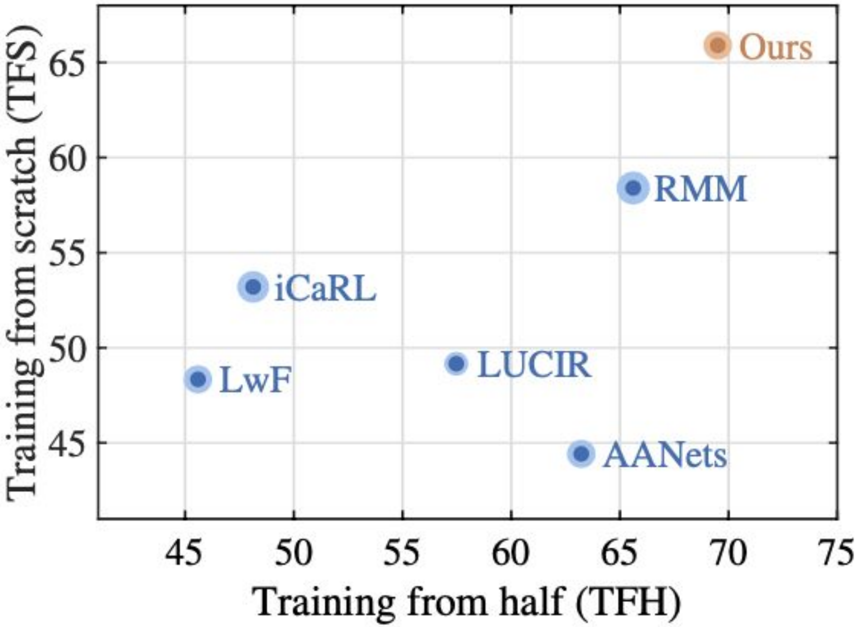
50 classes	10 classes	10 classes	10 classes	10 classes	10 classes
Phase 0	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5

The "training-from-scratch" (TFS)[2] setting requires higher plasticity

20 classes	20 classes	20 classes	20 classes	20 classes
Phase 1	Phase 2	Phase 3	Phase 4	Phase 5

Reference
[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[2] Castro, Francisco M., et al. "End-to-end incremental learning." ECCV 2018.

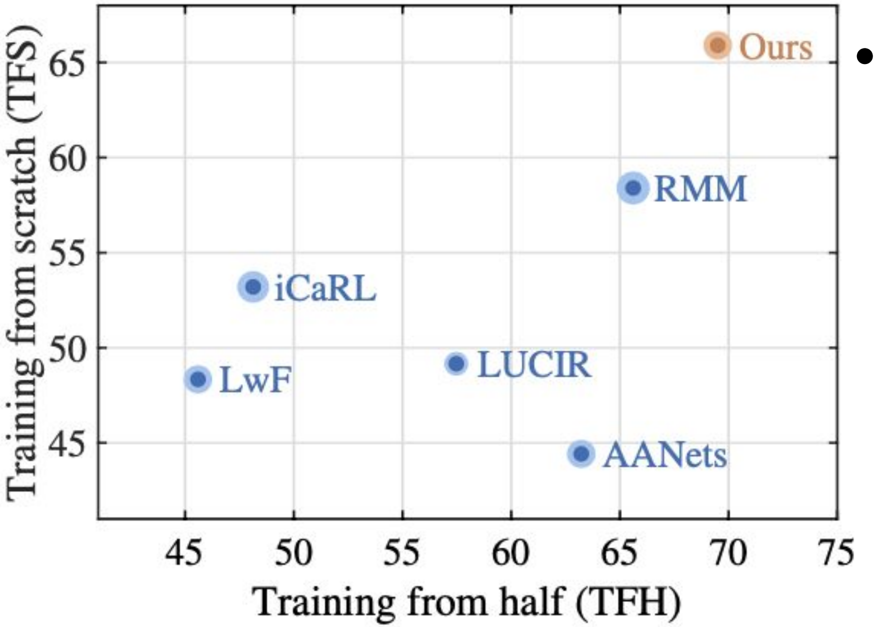
Existing CIL methods pre-fix the tradeoff balancing methods



Reference

[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[3] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "Adaptive aggregation networks for class-incremental learning." CVPR 2021.
[4] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "RMM: Reinforced memory management for class-incremental learning." NeurIPS 2021.
[5] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017.
[6] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." TPAMI 2017.

Existing CIL methods pre-fix the tradeoff balancing methods

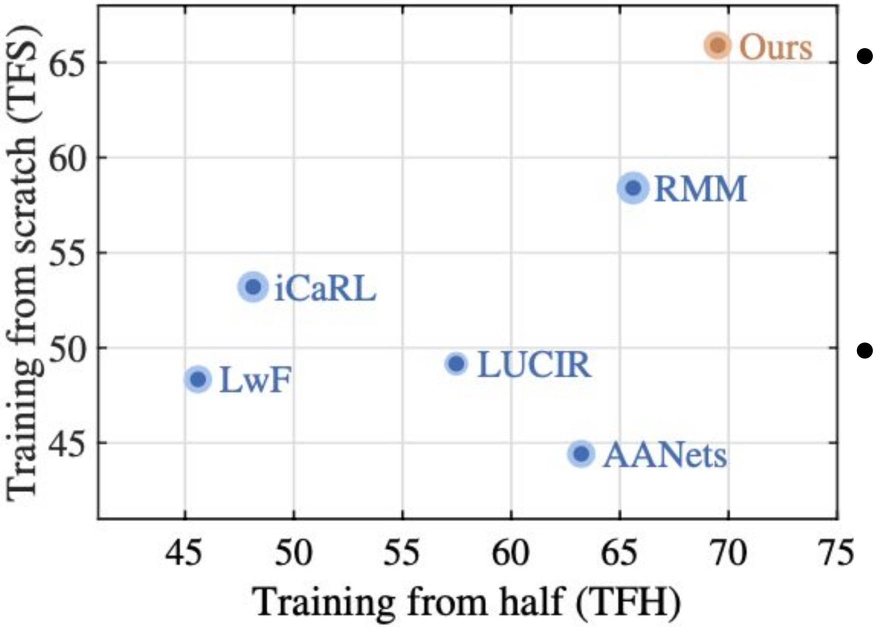


- **LUCIR[1], AANets[3], and RMM[4] are more suited for TFH**
  Reason: using strong (feature) knowledge distillation (KD) → high stability

Reference
[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[3] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "Adaptive aggregation networks for class-incremental learning." CVPR 2021.
[4] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "RMM: Reinforced memory management for class-incremental learning." NeurIPS 2021.
[5] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017.
[6] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." TPAMI 2017.

max planck institut
informatik

Existing CIL methods pre-fix the tradeoff balancing methods



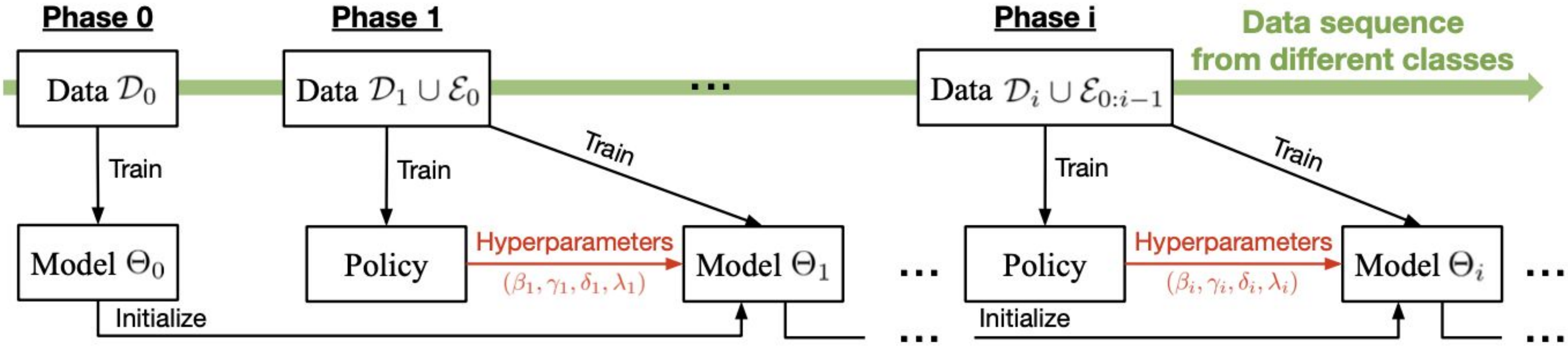
- **LUCIR[1], AANets[3], and RMM[4] are more suited for TFH**
  Reason: using strong (feature) knowledge distillation (KD)
  → high stability

- **iCaRL[5] and LwF[6] are more suited for TFS**
  Reason: using weak (logit) knowledge distillation (KD)
  → high plasticity

Reference
[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[3] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "Adaptive aggregation networks for class-incremental learning." CVPR 2021.
[4] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "RMM: Reinforced memory management for class-incremental learning." NeurIPS 2021.
[5] Rebuffi, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning." CVPR 2017.
[6] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." TPAMI 2017.

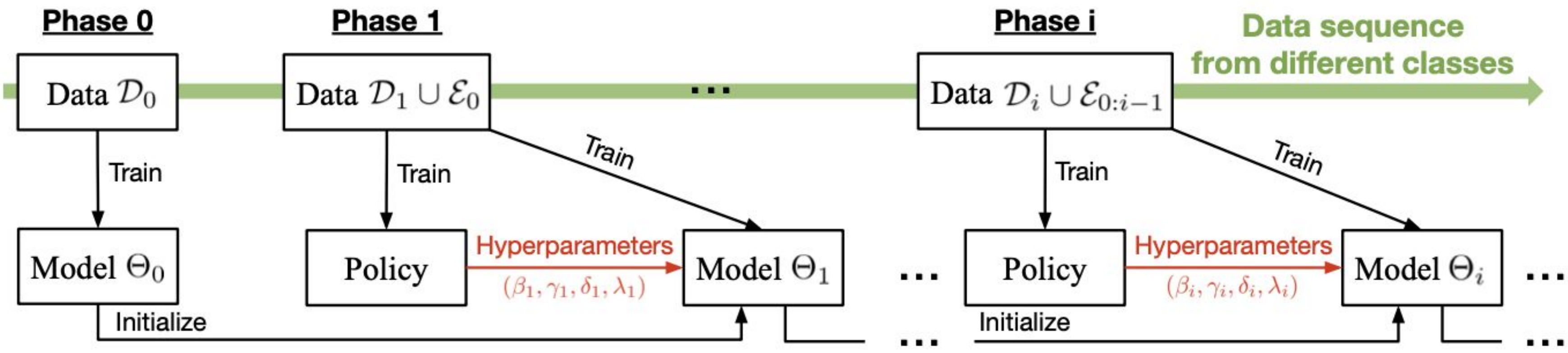
Question: how to design an adaptive trade-off balancing method?

Our solution: formulating the CIL task as an online Markov decision process (MDP), and learning a policy to produce the hyperparameters.



Question: how to design an adaptive trade-off balancing method?

Our solution: formulating the CIL task as an online Markov decision process (MDP), and learning a policy to produce the hyperparameters.



Stage: each phase

Action: hyperparameters

State: CIL model

Reward: validation accuracy

Our objective: maximizing the cumulative reward, i.e., the average accuracy

How to solve the online MDP?

A common solution for an **online MDP**: approximating it as an **online learning problem**, and solve it using **online learning algorithms.**[7]

Phase *j* reward, i.e., Phase *j* validation accuracy

$$R = \sum_{j=1}^{N} r_{\mathcal{H}_j}(\Theta_j, \mathbf{a}_j)$$

Cumulative reward

Reference

[7] Even-Dar, Eyal, Sham M. Kakade, and Yishay Mansour. "Online Markov decision processes." Mathematics of Operations Research 34.3 (2009): 726-736.

How to solve the online MDP?

A common solution for an **online MDP**: approximating it as an **online learning problem**, and solve it using **online learning algorithms.**[7]

$$R = \sum_{j=1}^{N} r_{\mathcal{H}_j}(\Theta_j, \mathbf{a}_j)$$

Phase i

[7]: The approximation error (regret) is sublinear in N

$$\hat{R} = \sum_{j=1}^{i-1} r_{\mathcal{H}_j}(\Theta_j, \mathbf{a}_j) + \sum_{j=i}^{N} r_{\mathcal{H}_i}(\Theta_j, \mathbf{a}_j)$$

Part 2: the long-term reward of a time-invariant local MDP

Part 1: the historical rewards

Reference

[7] Even-Dar, Eyal, Sham M. Kakade, and Yishay Mansour. "Online Markov decision processes." Mathematics of Operations Research 34.3 (2009): 726-736.

How to solve the online MDP?

A common solution for an **online MDP**: approximating it as an **online learning problem**, and solve it using **online learning algorithms.**[7]

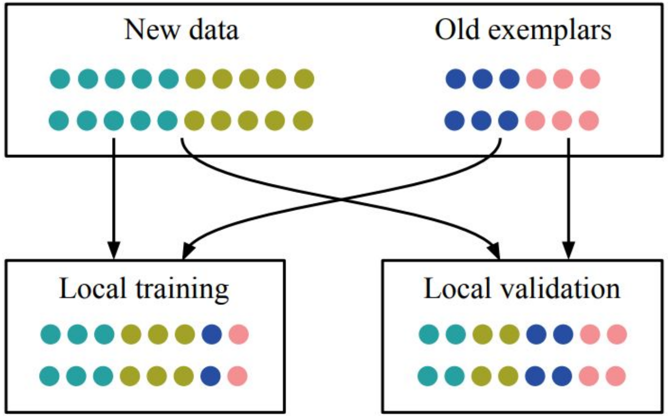
$$R = \sum_{j=1}^{N} r_{\mathcal{H}_j}(\Theta_j, \mathbf{a}_j)$$

[7]: The approximation error (regret) is sublinear in N

$$\hat{R} = \sum_{j=1}^{i-1} r_{\mathcal{H}_j}(\Theta_j, \mathbf{a}_j) + \sum_{j=i}^{N} r_{\mathcal{H}_i}(\Theta_j, \mathbf{a}_j)$$

Create a local environment

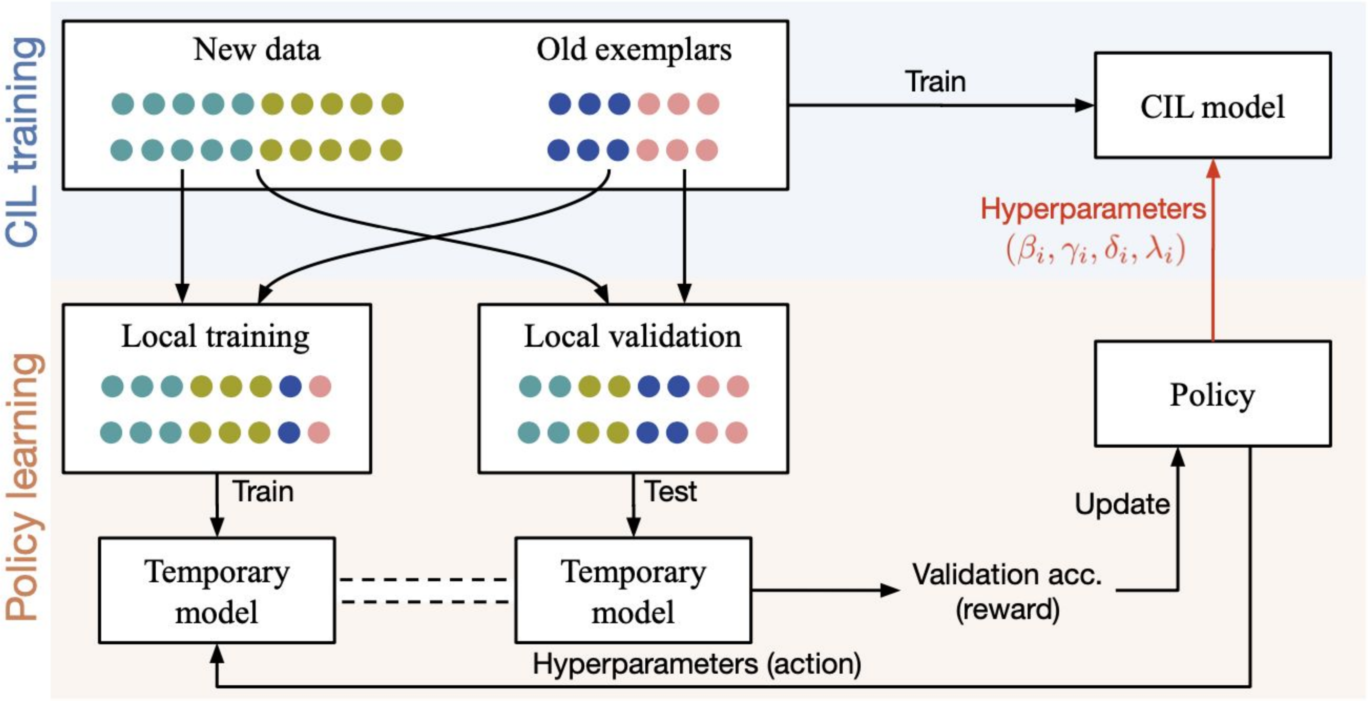
$$\hat{R}(\mathbf{a}_i, h_i) = \sum_{j=1}^{i-1} r_{\mathcal{H}_j}(\Theta_j, \mathbf{a}_j) + \sum_{j=i}^{i+n} r_{h_i}(\Theta_j, \mathbf{a}_i)$$



Reference

[7] Even-Dar, Eyal, Sham M. Kakade, and Yishay Mansour. "Online Markov decision processes." Mathematics of Operations Research 34.3 (2009): 726-736.

Policy training and deployment in Phase i



Our method can be used to optimize different hyperparameters

Ablation results (average accuracy, %) on CIFAR-100. Baseline: LUCIR[1]

No.	Optimizing			N=5		N=25	
	(β, γ)	δ	λ	TFH	TFS	TFH	TFS
1	Baseline			63.11	62.96	57.47	49.16
2	✓			63.20	63.60	58.27	50.91
3	✓	✓		63.23	64.08	58.20	51.94
4	✓	✓	✓	**63.88**	**64.92**	**59.27**	**52.44**
5	Cross-val fixed			63.33	64.02	57.50	51.64
6	Offline RL			63.42	63.88	58.12	51.53
7	Bilevel HO			63.20	63.02	57.56	49.42

(β, γ) : KD loss weights

δ : Classifier type (FC classifier vs. prototype classifier)

λ : Learning rates

Reference

[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.

Our method performs better than other hyperparameter optimization methods

Ablation results (average accuracy, %) on CIFAR-100. Baseline: LUCIR[1]

No.	Optimizing			N=5		N=25	
	(β, γ)	δ	λ	TFH	TFS	TFH	TFS
1	Baseline			63.11	62.96	57.47	49.16
2	✓			63.20	63.60	58.27	50.91
3	✓	✓		63.23	64.08	58.20	51.94
4	✓	✓	✓	**63.88**	**64.92**	**59.27**	**52.44**
5	Cross-val fixed			63.33	64.02	57.50	51.64
6	Offline RL			63.42	63.88	58.12	51.53
7	Bilevel HO			63.20	63.02	57.56	49.42

Cross-val fixed: using cross-validation to find a set of fixed hyperparameters
Offline RL: using the policy pre-trained in an offline manner as [4].
Bilevel HO: using a bilevel hyperparameter optimization method as [8].

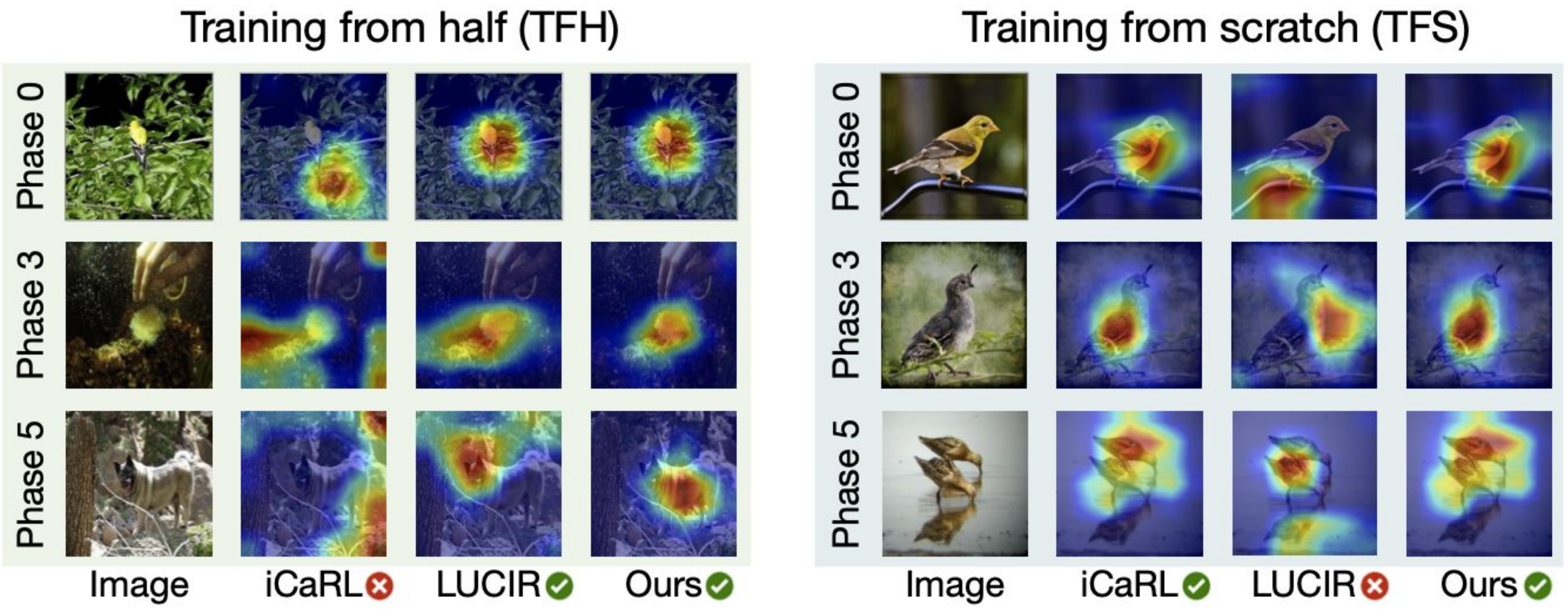
Reference
[1] Hou, Saihui, et al. "Learning a unified classifier incrementally via rebalancing." CVPR 2019.
[4] Liu, Yaoyao, Bernt Schiele, and Qianru Sun. "RMM: Reinforced memory management for class-incremental learning." NeurIPS 2021.
[8] Franceschi, Luca, et al. "Bilevel programming for hyperparameter optimization and meta-learning." ICML 2018.

Our AANets achieve SOTA performance

Methods	CIFAR-100, N=5			CIFAR-100, N=25			ImageNet-Subset, N=5			ImageNet-Subset, N=25		
	TFH	TFS	Avg.	TFH	TFS	Avg.	TFH	TFS	Avg.	TFH	TFS	Avg.
iCaRL [30]	58.1	64.0	61.0	48.1	53.2	50.7	65.3	70.4	67.9	53.0	53.5	53.3
PODNet [8]	64.7	63.6	64.2	60.3	45.3	52.8	64.3	58.9	61.6	68.3	39.1	53.7
DER [40]	67.6	72.3	70.0	65.5	67.3	66.4	78.4	76.9	77.7	75.4	71.0	73.2
FOSTER [38]	70.4	72.5	71.5	63.8	**70.7**	67.3	80.2	78.3	79.3	69.3	72.9	71.1
LUCIR [14]	63.1$\pm$0.7	63.0$\pm$0.6	63.1$\pm$0.7	57.5$\pm$0.4	49.2$\pm$0.5	53.4$\pm$0.5	65.3$\pm$0.6	66.7$\pm$0.5	66.0$\pm$0.6	61.4$\pm$0.7	46.2$\pm$0.8	53.8$\pm$0.8
w/ ours	63.9$\pm$0.6	64.9$\pm$0.5	64.4$\pm$0.6	59.3$\pm$0.5	52.4$\pm$0.5	55.9$\pm$0.5	70.6$\pm$0.7	68.4$\pm$0.6	69.5$\pm$0.7	62.9$\pm$0.6	54.1$\pm$0.6	58.5$\pm$0.6
	$\uparrow$0.8	$\uparrow$1.9	$\uparrow$1.3	$\uparrow$1.8	$\uparrow$3.2	$\uparrow$2.5	$\uparrow$5.3	$\uparrow$1.7	$\uparrow$3.5	$\uparrow$1.5	$\uparrow$7.9	$\uparrow$4.7
AANets [22]	65.3$\pm$0.4	63.1$\pm$0.3	64.2$\pm$0.4	63.2$\pm$0.3	44.4$\pm$0.4	53.8$\pm$0.4	77.0$\pm$0.7	68.9$\pm$0.6	73.0$\pm$0.7	72.2$\pm$0.6	60.7$\pm$0.5	66.5$\pm$0.6
w/ ours	67.0$\pm$0.3	65.1$\pm$0.3	66.1$\pm$0.3	64.1$\pm$0.4	50.3$\pm$0.5	57.2$\pm$0.5	77.3$\pm$0.6	70.6$\pm$0.5	74.0$\pm$0.6	72.9$\pm$0.5	64.8$\pm$0.5	68.9$\pm$0.5
	$\uparrow$1.7	$\uparrow$2.0	$\uparrow$1.9	$\uparrow$0.9	$\uparrow$5.9	$\uparrow$3.4	$\uparrow$0.3	$\uparrow$1.7	$\uparrow$1.0	$\uparrow$0.7	$\uparrow$4.1	$\uparrow$2.4
RMM [23]	67.6$\pm$0.7	70.4$\pm$0.8	69.0$\pm$0.8	65.6$\pm$0.6	58.4$\pm$0.6	62.0$\pm$0.6	79.5$\pm$0.2	80.5$\pm$0.3	80.0$\pm$0.3	75.0$\pm$0.3	71.6$\pm$0.3	73.3$\pm$0.3
w/ ours	**70.8**$\pm$0.7	**72.7**$\pm$0.6	**71.8**$\pm$0.7	**69.5**$\pm$0.8	65.9$\pm$0.7	**67.7**$\pm$0.8	**81.0**$\pm$0.3	**82.2**$\pm$0.4	**81.6**$\pm$0.4	**76.1**$\pm$0.2	**73.2**$\pm$0.4	**74.7**$\pm$0.3
	$\uparrow$3.2	$\uparrow$2.3	$\uparrow$2.8	$\uparrow$3.9	$\uparrow$7.5	$\uparrow$5.7	$\uparrow$1.5	$\uparrow$1.7	$\uparrow$1.6	$\uparrow$1.1	$\uparrow$1.6	$\uparrow$1.4

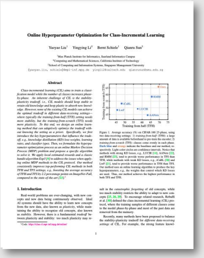
- Generic
- Boost the performance for THREE different baselines

The activation maps using Grad-CAM



The 5-th phase (the last phase) model on ImageNet-Subset 5-phase. Samples are selected from the classes coming in the zeroth, third, and fifth phases, respectively.

Thanks!


Online Hyperparameter Optimization for Class-Incremental Learning

Yaoyao Liu[1] Yingying Li[2] Bernt Schiele[1] Qianru Sun[3]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus
[2]Computing and Mathematical Sciences, California Institute of Technology
[3]School of Computing and Information Systems, Singapore Management University

Abstract

Online Hyperparameter Optimization for Class-Incremental Learning

Webpage: https://class-il.mpi-inf.mpg.de/online/

Code: https://class-il.mpi-inf.mpg.de/online/code/