JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Class-Incremental Exemplar Compression for Class-Incremental Learning

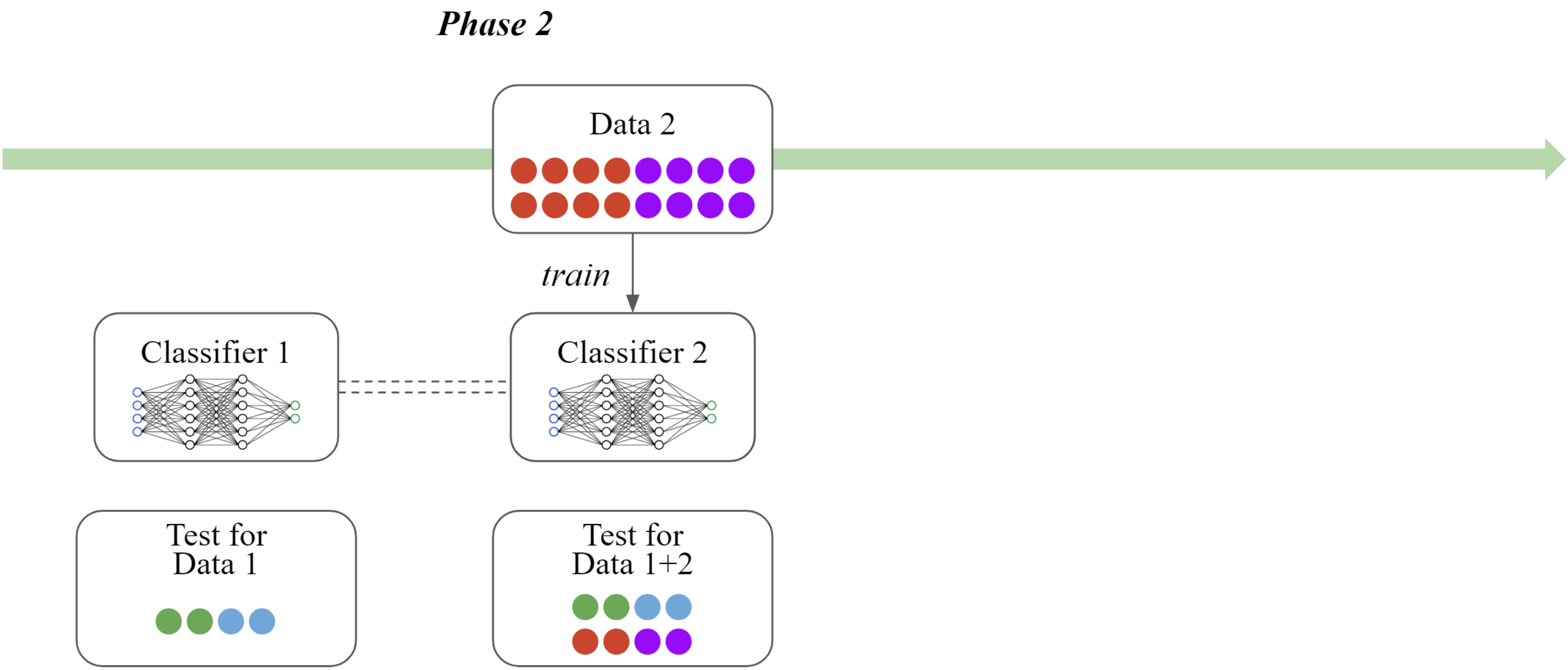
Zilin Luo[1], Yaoyao Liu[2], Bernt Schiele[2], Qianru Sun[1]

[1]Singapore Management University [2]Max Planck Institute for Informatics

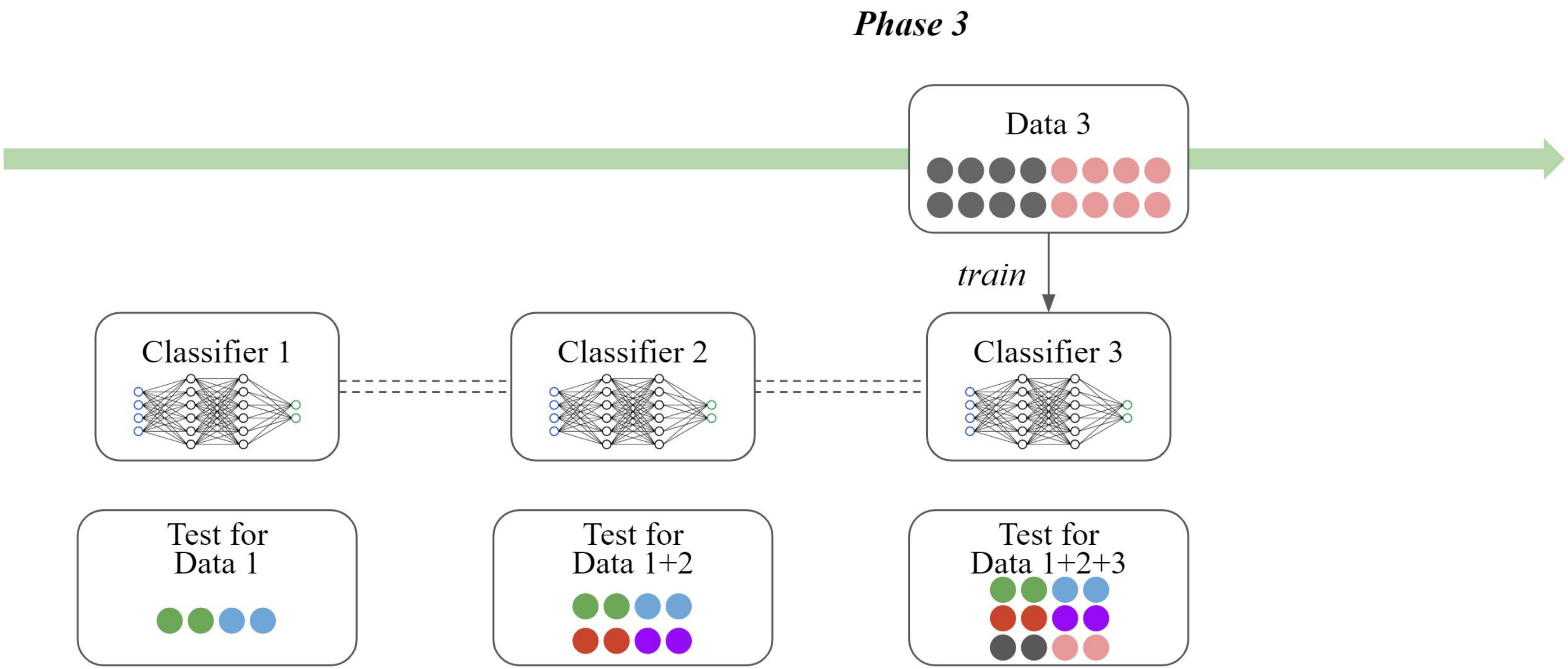
WED-AM-299

Research Background: Class-Incremental Learning (CIL)

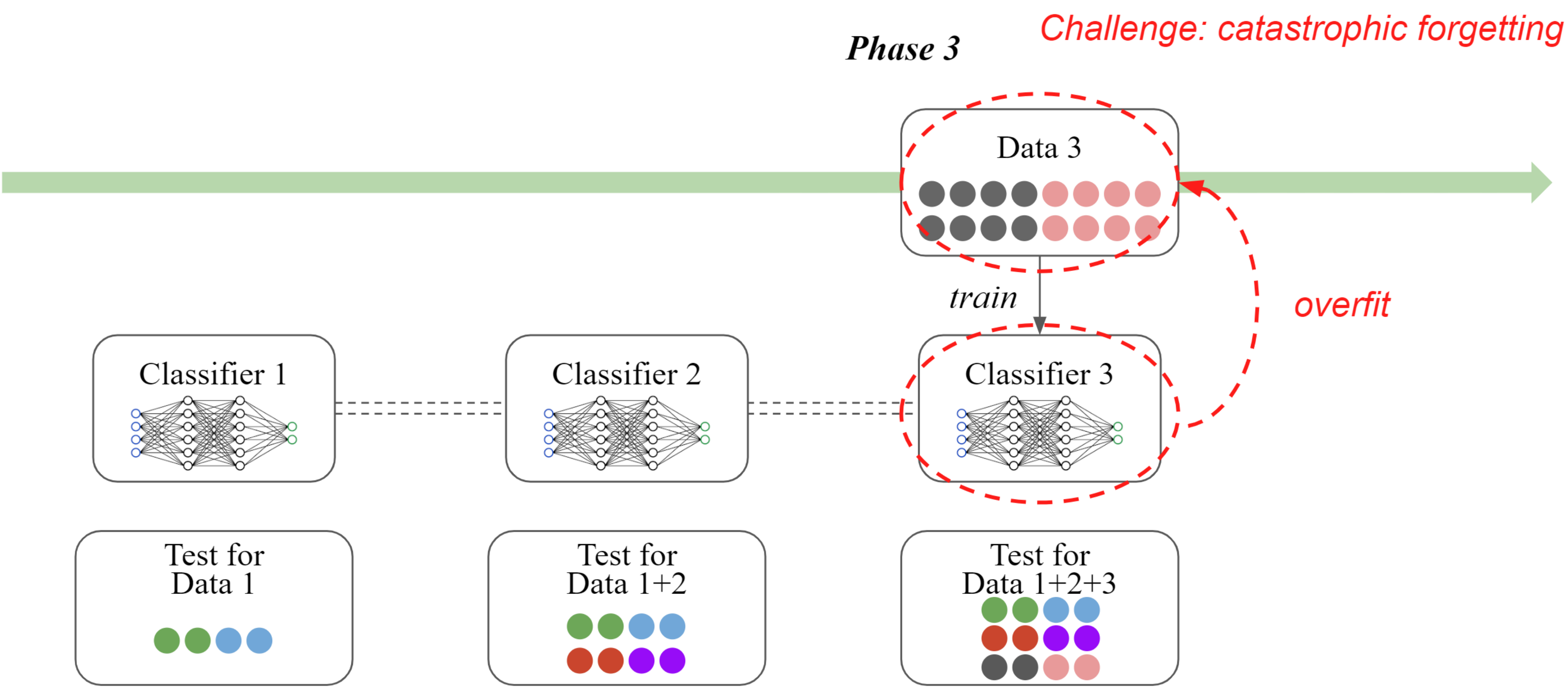
Research Background: Class-Incremental Learning (CIL)



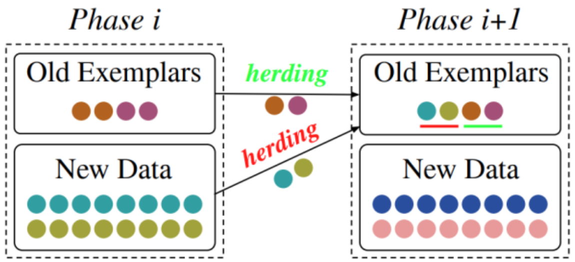
Research Background: Class-Incremental Learning (CIL)



Research Background: Class-Incremental Learning (CIL)



Research Background: Exemplar-based CIL



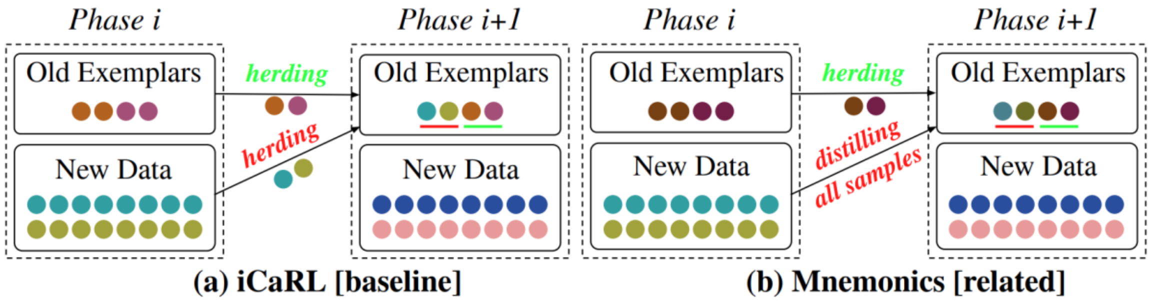
(a) iCaRL [baseline]

(a) iCaRL [related]
- maintains a memory with limited capacity
- selects exemplars using herding technique

Reference

[1] Rebuffi, Sylvestre-Alvise, et al. “iCaRL: Incremental classifier and representation learning.” In CVPR 2017.
[2] Liu, Yaoyao, et al. “Mnemonics Training: Multi-Class Incremental Learning without Forgetting.” In CVPR 2020.
[3] Wang, Liyuan, et al. “Memory Replay with Data Compression for Continual Learning.” In ICLR 2022.

Research Background: Exemplar-based CIL



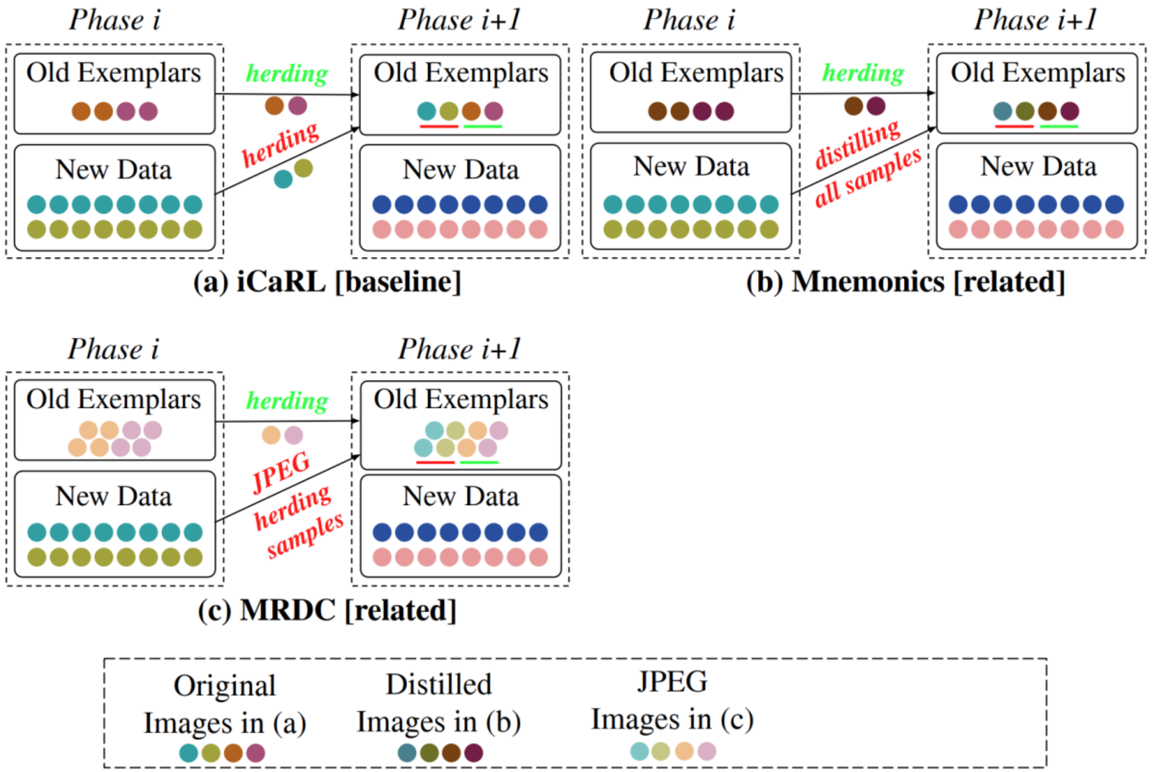
(b) Mnemonics [related]

- distills exemplars as optimizable parameters
- one exemplar carries more information
- number of saved exemplars is not changed

Reference

[1] Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." In CVPR 2017.
[2] Liu, Yaoyao, et al. "Mnemonics Training: Multi-Class Incremental Learning without Forgetting." In CVPR 2020.
[3] Wang, Liyuan, et al. "Memory Replay with Data Compression for Continual Learning." In ICLR 2022.

Research Background: Exemplar-based CIL



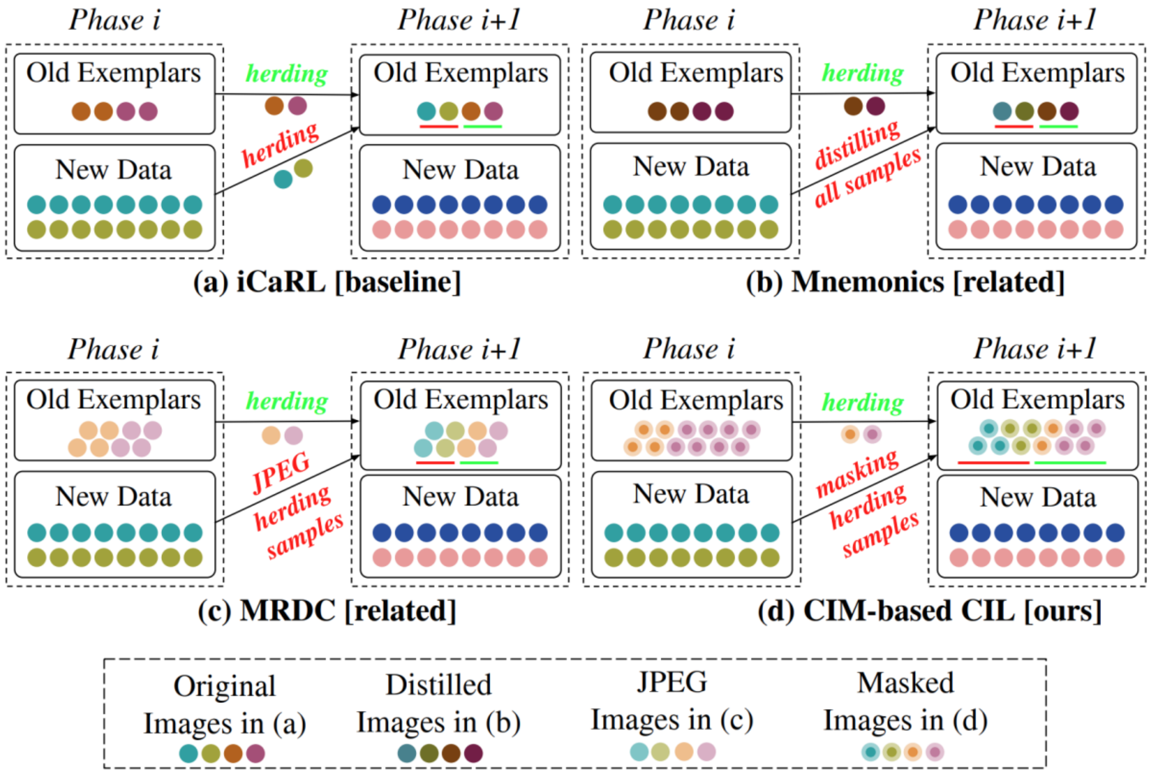
(c) MRDC [related]

- compresses exemplars with JPEG algorithm
- discriminativeness of each exemplar is weaken
- number of saved exemplars is increased
- aims to trade-off between quality and quantity

Reference

[1] Rebuffi, Sylvestre-Alvise, et al. "iCaRL: Incremental classifier and representation learning." In CVPR 2017.
[2] Liu, Yaoyao, et al. "Mnemonics Training: Multi-Class Incremental Learning without Forgetting." In CVPR 2020.
[3] Wang, Liyuan, et al. "Memory Replay with Data Compression for Continual Learning." In ICLR 2022.

Research Background: Exemplar-based CIL



(d) CIM-based CIL [ours]

- adopts pixel-selective compression strategy
- applies adaptive compression for dynamic CIL environments
- number of saved exemplars is increased
- little discriminativeness of exemplars is lost

Reference

[1] Rebuffi, Sylvestre-Alvise, et al. “iCaRL: Incremental classifier and representation learning.” In CVPR 2017.

[2] Liu, Yaoyao, et al. “Mnemonics Training: Multi-Class Incremental Learning without Forgetting.” In CVPR 2020.

[3] Wang, Liyuan, et al. “Memory Replay with Data Compression for Continual Learning.” In ICLR 2022.

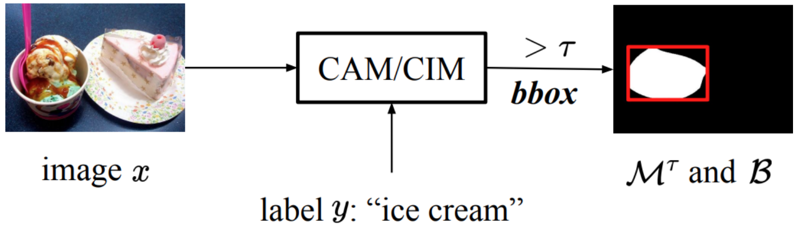
Overall Compression Framework

- Goal: reducing exemplar storage while losing little representativeness
- Core idea: downsampling only non-discriminative pixels

Reference
[4] Zhou, Bolei, et al. “Learning Deep Features for Discriminative Localization.” In CVPR 2016.

Overall Compression Framework

- Goal: reducing exemplar storage while losing little representativeness
- Core idea: downsampling only non-discriminative pixels



CAM: $$\mathcal{M}^{\text{CAM}} = \frac{A - \min(A)}{\max(A) - \min(A)}, \ A = \omega_{i,y}^{\top} F(x; \theta_i)$$

θ_i : parameters of i-phase feature extractor

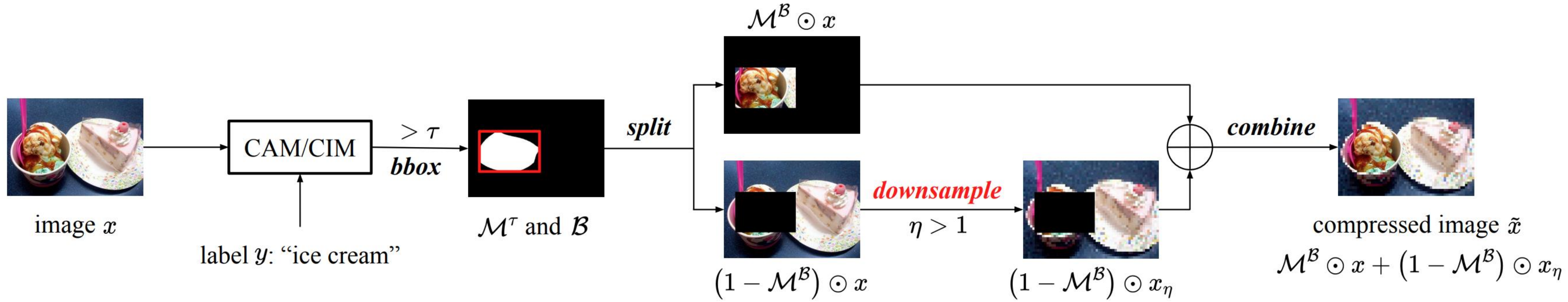
$\omega_{i,y}$: parameters of i-phase classifier corresponding to y-th class

Reference

[4] Zhou, Bolei, et al. "Learning Deep Features for Discriminative Localization." In CVPR 2016.

Overall Compression Framework

- Goal: reducing exemplar storage while losing little representativeness
- Core idea: downsampling only non-discriminative pixels



CAM: $\mathcal{M}^{\mathrm{CAM}} = \frac{A - \min(A)}{\max(A) - \min(A)}, \; A = \omega_{i,y}^{\top} F(x; \theta_i)$

θ_i : parameters of i-phase feature extractor
$\omega_{i,y}$: parameters of i-phase classifier corresponding to y-th class

storage: $m_{\tilde{x}} = \frac{H_{\mathcal{B}} W_{\mathcal{B}}}{HW} + \frac{1}{\eta}\left(1 - \frac{H_{\mathcal{B}} W_{\mathcal{B}}}{HW}\right)$
$= 1 - \left(1 - \frac{1}{\eta}\right) \cdot \left(1 - \frac{H_{\mathcal{B}} W_{\mathcal{B}}}{HW}\right)$

up to compression ratio and bounding box

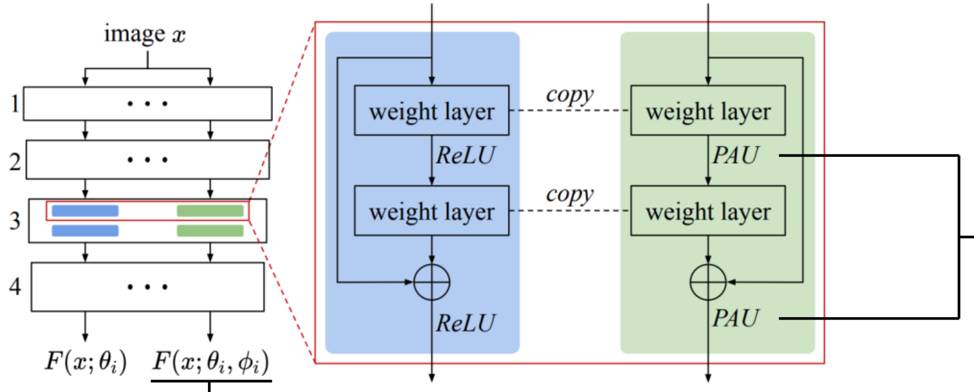
Reference

[4] Zhou, Bolei, et al. "Learning Deep Features for Discriminative Localization." In CVPR 2016.

Class-Incremental Masking

- Goal: applying adaptive compression for dynamic CIL environments
- Core idea: generating compression masks with the CIM module

CIM module: integrated as an "extension" on the original feature extractor



Pade Activation Units, parameterized by ϕ_i

uses features from the appended path to produce masks

Reference
[5] Molina, Alejandro, et al. "Pade Activation Units: End-to-end Learning of Flexible Activation Functions in Deep Networks." In ICLR 2019.

Class-Incremental Masking

We organize the optimization of the whole model into two levels: task-level and mask-level.

- Task-level Optimization $(\theta_i, \omega_i) \leftarrow (\theta_i, \omega_i) - \lambda \nabla_{(\theta,\omega)} \mathcal{L}_{\text{CIL}}(\tilde{\mathcal{E}}_{0:i-1} \cup \mathcal{D}_i; \theta_i, \omega_i)$

- Mask-level Optimization
$$\begin{cases} \text{inner step: } \theta_i^{+} \leftarrow \theta_i - \beta_1 \nabla_\theta \mathcal{L}_{\text{CIL}}(\tilde{\mathcal{E}}_{0:i-1} \cup \tilde{\mathcal{D}}_i(\phi_i); \theta_i, \omega_i) \\ \text{outer step } \phi_i \leftarrow \phi_i - \beta_2 \nabla_\phi [\mathcal{L}_{\text{CE}}(\mathcal{D}_i; \theta_i^{+}, \omega_i) + \mu \mathcal{R}(\phi_i) + \mu' \mathcal{L}_{\text{CE}}(\tilde{\mathcal{E}}_{0:i-1} \cup \mathcal{D}_i; \theta_i, \phi_i, \omega_i)] \end{cases}$$

validation loss

memory constraint

mask diversity loss

Algorithm 1: CIM-based CIL (i-th phase, $i \geq 1$)

Input: New data $\mathcal{D}_i$; old compressed exemplars $\tilde{\mathcal{E}}_{0:i-1}$ ($\tilde{\mathcal{E}}_0 = \varnothing$); last-phase CIL model $(\theta_{i-1}, \omega_{i-1})$ and CIM model ϕ_{i-1} (θ_0, ω_0 are random parameters, ϕ_0 is set to ReLU).

Output: New compressed exemplars $\tilde{\mathcal{E}}_i$; updated CIL model (θ_i, ω_i) and CIM model ϕ_i.

Initialize (θ_i, ω_i) with $(\theta_{i-1}, \omega_{i-1})$;
Initialize ϕ_i with ϕ_{i-1};
for *epochs* **do**
  Train (θ_i, ω_i) using $\tilde{\mathcal{E}}_{0:i-1} \cup \mathcal{D}_i$ by Eq. 5;
  Compress $\mathcal{D}_i$ into $\tilde{\mathcal{D}}_i^{\phi}$ using ϕ_i;
  Temporarily update θ_i to θ_i^{+} using $\tilde{\mathcal{E}}_{0:i-1} \cup \tilde{\mathcal{D}}_i^{\phi}$ by Eq. 7;
  Learn ϕ_i using $\mathcal{D}_i$ by Eq. 9;
end
Compress $\mathcal{D}_i$ into $\tilde{\mathcal{D}}_i$ using the learned ϕ_i by Eq. 3;
Select exemplars $\tilde{\mathcal{E}}_i$ from $\tilde{\mathcal{D}}_i$ by, e.g., herding [37].

Reference
[5] Molina, Alejandro, et al. "Pade Activation Units: End-to-end Learning of Flexible Activation Functions in Deep Networks." In ICLR 2019.

Comparing with SOTA

Method	*Learning from Scratch (LFS)*						*Learning from Half (LFH)*					
	Food-101			*ImageNet-100*			*Food-101*			*ImageNet-100*		
	N=5	10	20	5	10	20	5	10	25	5	10	25
iCaRL [37]	69.66	62.18	56.70	73.90	67.06	62.36	60.13	53.42	46.87	62.53	59.88	52.97
WA [50]	70.94	63.69	58.45	74.64	68.62	63.20	63.55	57.60	52.48	65.75	63.71	58.34
PODNet [13]	68.03	61.24	47.38	72.14	63.96	53.69	75.37	70.01	65.32	75.54	74.33	68.33
AANets [26]	69.46	61.59	48.83	72.98	65.77	55.36	76.07	71.22	66.93	76.96	75.58	71.78
DER [48]	73.88	70.76	64.39	78.50	76.12	73.79	78.13	73.45	-	79.08	77.73	-
DER *w/* ours	75.63	73.09	69.17	79.63	77.57	**75.36**	79.25	75.76	-	80.30	**79.05**	-
FOSTER [44]	75.03	72.72	66.73	79.93†	76.55†	74.49	79.08	75.07	68.08	80.07†	77.54	72.40*
FOSTER *w/* ours	**76.44**	**74.85**	**70.20**	**80.58**	**77.94**	75.23	**79.76**	**76.86**	**70.50**	**80.93**	78.66	**75.74**

- serves as a plug-in module to baselines
- achieves consistent performance improvements in multiple settings and datasets

Reference
[6] Yan, Shipeng, et al. "DER: Dynamically Expandable Representation for Class Incremental Learning." In CVPR 2021.
[7] Wang Fuyun, et al. "FOSTER: Feature Boosting and Compression for Class-Incremental Learning." In ECCV 2022.

Comparing with SOTA

Memory Budget	Method	N=5 Avg.	N=5 Last	N=10 Avg.	N=10 Last
$M=20k$	iCaRL [37]	44.36	27.78	38.40	22.70
	WA [50]	58.37	50.62	54.10	45.66
	DER [48]	67.49	59.75	66.73	58.62
	FOSTER [44]	69.21	64.88	68.34	60.14
	FOSTER *w/* ours	**69.93**	**66.05**	**69.53**	**62.07**
$M=5k$	FOSTER	57.19	49.42	54.72	44.96
	FOSTER *w/* ours	**61.37**	**54.46**	**59.48**	**50.83**

On large-scale ImageNet-1000:

- brings larger performance improvement under stricter memory budget

Ablation Studies

Ablation Method	*Food-101*		*ImageNet-100*	
	N=10	20	10	20
1 Baseline	72.72	66.73	76.55	72.37
2 Artifact Aug.	71.38	66.03	75.63	71.45
3 Full Comp.	73.03	67.38	76.92	73.26
4 Random Acti.	73.10	67.54	76.88	73.54
5 Center Acti.	73.29	67.88	76.78	73.82
6 Class Acti.	73.76	68.65	77.21	74.67
7 Phase-wise τ	73.83	69.17	77.06	74.78
8 Joint Train	73.44	69.01	77.34	74.59
9 BOP (ours)	74.85	**70.20**	**77.94**	75.23
10 LastBlock Only	74.55	69.87	77.72	74.86
11 Fg Compressed	**75.02**	70.13	77.87	**75.46**

- Line 1-2: SOTA baselines.
- Line 3-6: different activation methods.
- Line 7-9: different optimization strategies.
- Line 10-11: two variants.

Performances on Different-Size Objects

Metric	Small	Middle	Large
Mean of #Exemplars	39.40	38.30	34.77
Last Acc. (%, baseline)	66.13	68.40	69.93
Last Acc. (%, ours)	70.00	71.10	72.26
Improvement (%)	+3.87	+3.65	+2.33

baseline: 20 exemplars/class.

- achieves larger performance boost for small objects

Thanks for listening!

Paper

Code