Class-incremental Learning in CVPR’22

Gengwei Zhang

University of Technology Sydney

Background

Continual Learning / Incremental Learning / Life-long Learning

- Training sets are provided sequentially: $X_1, X_2, \ldots, X_t, \ldots$,
  each training set is typically called a task

- Model can only access to one training set X at step t
  (a few memory samples from previous tasks are acceptable),
  while being evaluated on test set of all tasks.

-> requires model to maintain performance on old tasks when learning on new tasks

Background

Class-incremental Learning

- Each training set corresponds to a label set Y_t, Y_t are incrementally increasing
- Typically, $Y_i \cap Y_j = \emptyset$ for any $i \neq j$
- Test set contains $Y = Y_1 \cup Y_2 \cup \cdots Y_t \cup \cdots$

Background

Three typical scenario of class-incremental learning

1. Training from scratch
   -> DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion

2. Training with a large initial phase
   -> Mimicking the Oracle: An Initial Phase Decorrelation Approach for Class Incremental Learning

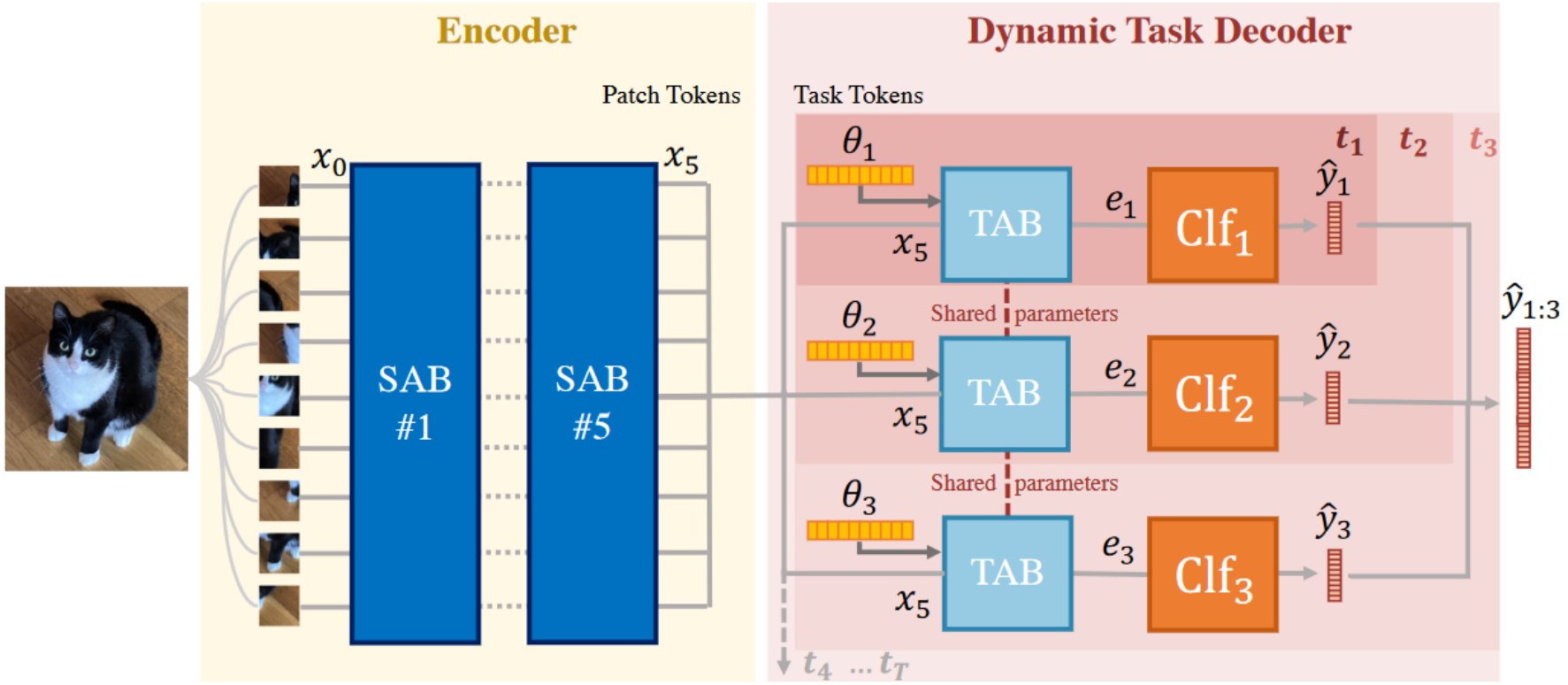
3. Training with no sample memory available
   -> Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning

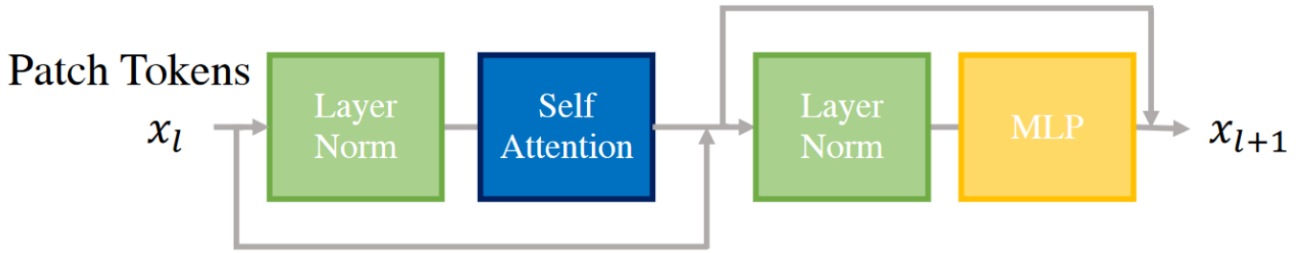
Dytox

DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion

Arthur Douillard[1,2], Alexandre Ramé[1], Guillaume Couairon[1,3], Matthieu Cord[1,4]
[1]Sorbonne Université, [2]Heuritech, [3]Meta AI, [4]valeo.ai
arthur.douillard@heuritech.com, {alexandre.rame, matthieu.cord}@sorbonne-universite.fr,
gcouairon@fb.com

Dytox





DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion

Dytox

Dynamic task token expansion

- For each task, learn a task token θ_t,
  the task token is first concatenate with the feature tokens

$$z_i = [\boldsymbol{\theta}_i, x_L] \in \mathbb{R}^{(\mathbb{N}+1)\times\mathbb{D}}$$

- Obtain task-specific feature via task-attention (cross attention):

$$\begin{aligned}
Q_i &= W_q \boldsymbol{\theta}_i \,, \\
K_i &= W_k z_i \,, \\
V_i &= W_v z_i \,, \\
A_i &= \mathrm{Softmax}\left(Q_i \cdot K_i^T / \sqrt{d/h}\right) \,, \\
O_i &= W_o A_i V_i + b_o \in \mathbb{R}^{1\times\mathbb{D}} \,,
\end{aligned}$$

- After obtain task-specific embeddings $E = \{e_1, e_2, \ldots, e_t\}$,
  classification score for each task is obtained by

$$\hat{y}_i = \mathrm{Clf}_i(e_i) = \sigma(W_i \,\mathrm{Norm}_i \, e_i + b_i) \,,$$

DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion

Dytox

Others:

- Objectives

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_{\text{clf}} + \alpha\mathcal{L}_{\text{kd}} + \lambda\mathcal{L}_{\text{div}}$$

- Classification scores are obtained separately
  -> requires classification scores to be well calibrated
  -> BCE rather than CE
  -> post-hoc fine-tuning

Dytox

Results

Methods	ImageNet100 10 steps					ImageNet1000 10 steps				
	#P	top-1		top-5		#P	top-1		top-5	
		Avg	Last	Avg	Last		Avg	Last	Avg	Last
ResNet18 joint	11.22	-	-	-	95.10	11.68	-	-	-	89.27
Transf. joint	11.00	-	79.12	-	93.48	11.35	-	73.58	-	90.60
E2E [5]	11.22	-	-	89.92	80.29	11.68	-	-	72.09	52.29
Simple-DER [48]	-	-	-	-	-	28.00	66.63	59.24	85.62	80.76
iCaRL [59]	11.22	-	-	83.60	63.80	11.68	38.40	22.70	63.70	44.00
BiC [32]	11.22	-	-	90.60	84.40	11.68	-	-	84.00	73.20
WA [80]	11.22	-	-	91.00	84.10	11.68	65.67	55.60	86.60	81.10
RPSNet [56]		-	-	87.90	74.00	-	-	-	-	-
DER w/o P [75]	112.27	**77.18**	66.70	**93.23**	87.52	116.89	68.84	60.16	88.17	82.86
DER† [75]	-	76.12	66.06	92.79	88.38	-	66.73	58.62	87.08	81.89
DyTox	11.01	**77.15**	**69.10**	92.04	**87.98**	11.36	**71.29**	**63.34**	**88.59**	**84.49**

DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion

Dytox

Ablations

		Knowledge Distillation	Finetuning	Token Expansion	Divergence Classifier	Indendepent Classifiers	Avg	Last
DyTox	Transformer						60.69	38.87
		✓					61.62	39.35
		✓	✓				63.42	42.21
	Dynamic	✓	✓	✓			67.30	47.57
		✓	✓	✓	✓		68.28	49.45
		✓	✓	✓	✓	✓	70.20	52.34

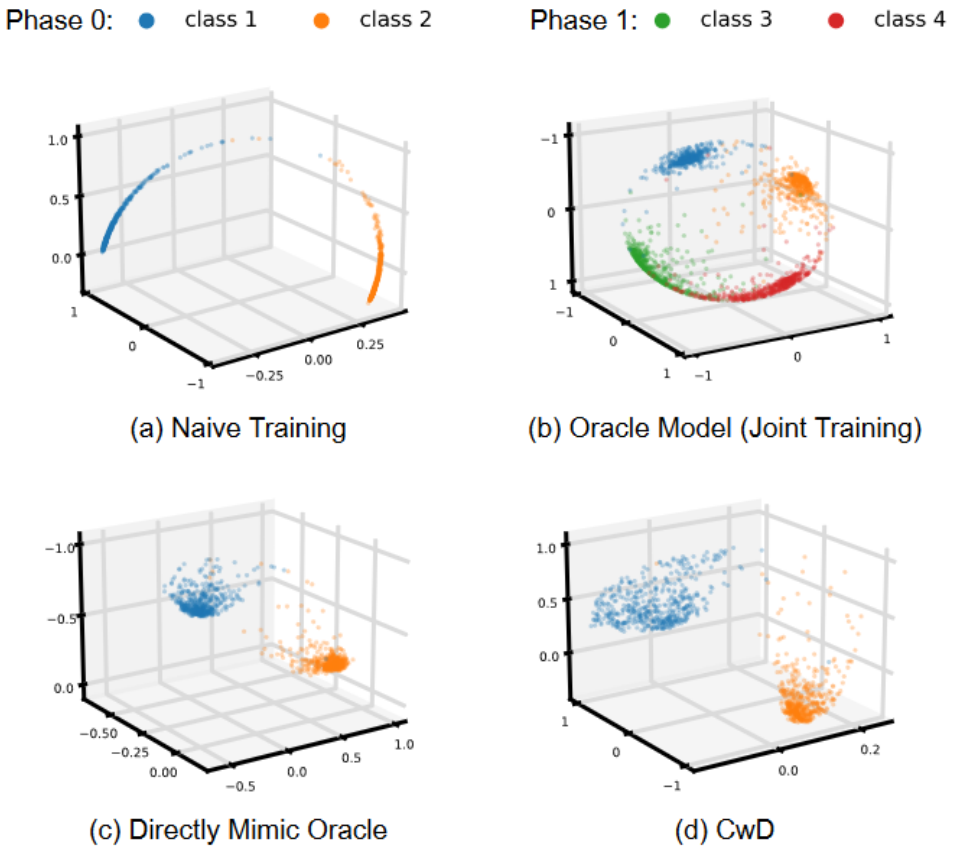
DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion

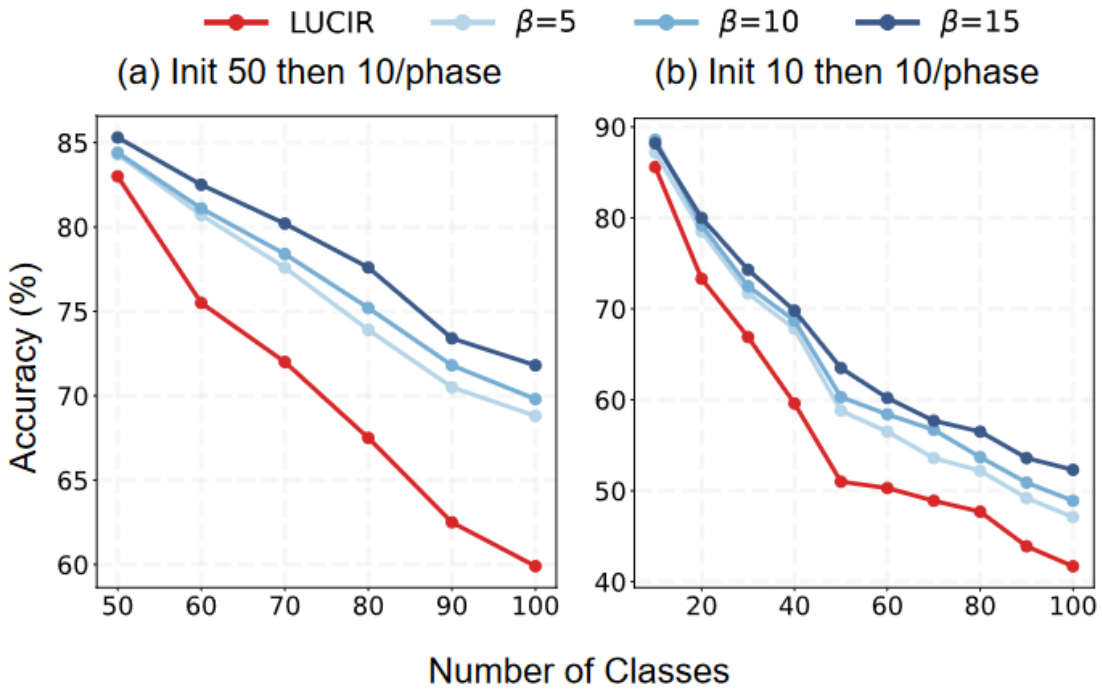
CwD

Mimicking the Oracle: An Initial Phase Decorrelation Approach for Class Incremental Learning

Yujun Shi[1] * Kuangqi Zhou[1] Jian Liang[3] Zihang Jiang[1]
Jiashi Feng[2] Philip Torr[4] Song Bai[2] Vincent Y. F. Tan[1]
[1]National University of Singapore [2] ByteDance Inc.
[3] Institute of Automation, Chinese Academy of Sciences (CAS) [4] University of Oxford
shi.yujun@u.nus.edu vtan@nus.edu.sg

CwD





Mimicking the Oracle: An Initial Phase Decorrelation Approach for Class Incremental Learning

CwD

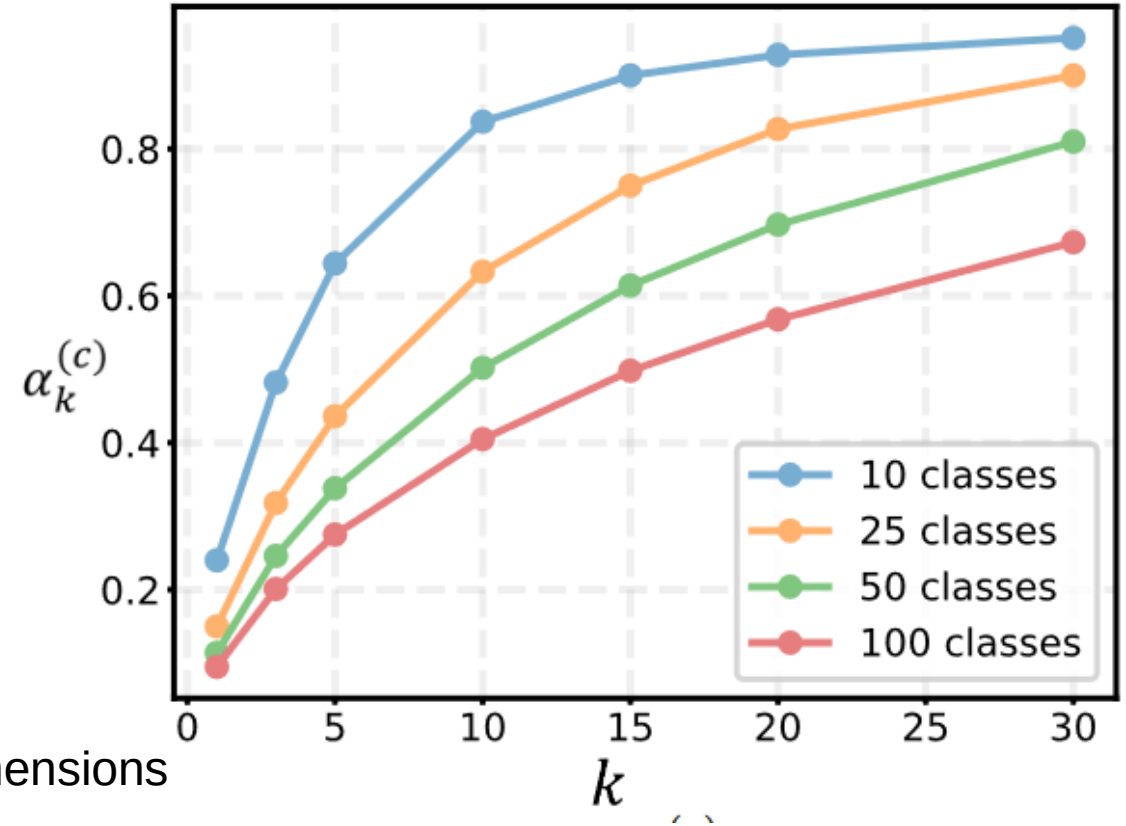
Quantitative Analysis

First obtain feature covariance matrix of class c

$$K^{(c)} = \frac{1}{n-1}\sum_{i=1}^{n}(Z_i^{(c)} - \bar{Z}^{(c)})(Z_i^{(c)} - \bar{Z}^{(c)})^\top$$

Then obtain the eigen value λ, and calculate

$$\alpha_k^{(c)} := \frac{\sum_{i=1}^{k}\lambda_i^{(c)}}{\sum_{i=1}^{d}\lambda_i^{(c)}} \in [0,1]$$



α represents the importance of the top-k feature dimensions

If α_k^c is close to 1 even when k is small, then the top eigenvalues dominate.

CwD

Class-wise Decorrelation (CwD) objective

Make eigen value distribute uniformly by minimizing:

$$L_{\text{shape}}^{(c)} = \frac{1}{d}\sum_{i=1}^{d}\left(\lambda_i^{(c)} - \frac{1}{d}\sum_{j=1}^{d}\lambda_j^{(c)}\right)^2,$$

It can be approve that

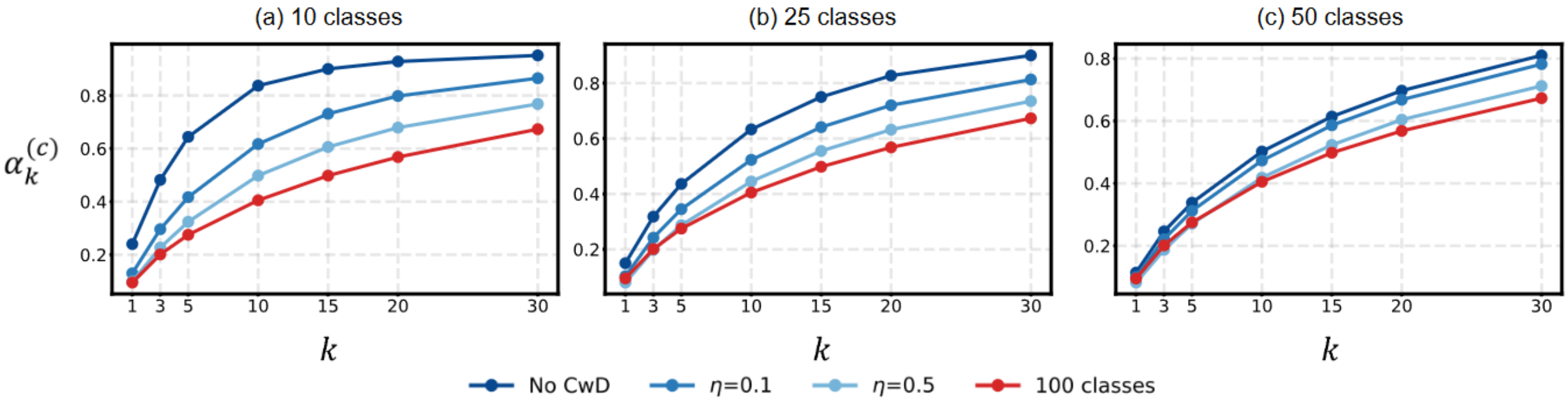
$$\sum_{i=1}^{d}\left(\lambda_i - \frac{1}{d}\sum_{j=1}^{d}\lambda_j\right)^2 = \|K\|_{\mathrm{F}}^2 - d.$$

Therefore, we have

$$L_{\mathrm{CwD}}(\theta) = \frac{1}{C \cdot d^2}\sum_{c=1}^{C}\|K^{(c)}\|_{\mathrm{F}}^2,$$

CwD

Effect


(a) 10 classes
(b) 25 classes
(c) 50 classes
$\alpha_k^{(c)}$
0.8
0.6
0.4
0.2
1 3 5 10 15 20 30
k
No CwD
η=0.1
η=0.5
100 classes

CwD

Results

Method	CIFAR100 (B=50)			ImageNet100 (B=50)			ImageNet (B=100)	
	S=10	5	2	10	5	2	100	50
LwF [18]	53.59±0.51	48.66±0.58	45.56±0.28	53.62†	47.64†	44.32†	40.86±0.13	27.72±0.12
iCaRL [26]	60.82±0.03	53.74±0.25	47.86±0.41	65.44†	59.88†	52.97†	49.56±0.09	42.61±0.15
BiC [33]	51.58±0.16	48.07±0.02	43.10±0.37	70.07†	64.96†	57.73†	43.23±0.13	38.83±0.12
LUCIR [12]	66.27±0.28	60.80±0.29	52.96±0.25	70.60±0.43	67.76±0.40	62.76±0.22	56.40±0.10	52.75±0.18
+CwD (ours)	67.26±0.16	62.89±0.09	56.81±0.21	71.94±0.11	69.34±0.31	65.10±0.59	57.42±0.11	53.37±0.22
PODNet [8]	66.98±0.13	63.76±0.48	61.00±0.18	75.71±0.37	72.80±0.35	65.57±0.41	57.01±0.12	54.06±0.09
+CwD (ours)	67.44±0.35	64.64±0.38	62.24±0.32	76.91±0.10	74.34±0.02	67.42±0.07	58.18±0.20	56.01±0.14
AANet [19]	69.79±0.21	67.97±0.26	64.92±0.30	71.96±0.12	70.05±0.63	67.28±0.34	51.76*±0.14	46.86*±0.13
+CwD (ours)	70.30±0.37	68.62±0.17	66.17±0.13	72.92±0.29	71.10±0.16	68.18±0.27	52.30*±0.08	47.61*±0.20

CwD

Ablations

S	B	LUCIR	+CwD (ours)	↑
10	10	$57.01_{\pm 0.14}$	$57.90_{\pm 0.07}$	+0.89
	20	$61.21_{\pm 0.35}$	$62.49_{\pm 0.36}$	+1.28
	30	$64.82_{\pm 0.38}$	$66.54_{\pm 0.35}$	+1.72
	40	$67.68_{\pm 0.37}$	$69.70_{\pm 0.10}$	+2.02
	50	$70.60_{\pm 0.43}$	$71.94_{\pm 0.11}$	+1.33
5	10	$50.47_{\pm 0.31}$	$51.92_{\pm 0.10}$	+1.45
	20	$56.41_{\pm 0.37}$	$58.14_{\pm 0.13}$	+1.73
	30	$61.00_{\pm 0.09}$	$63.18_{\pm 0.14}$	+2.18
	40	$63.73_{\pm 0.23}$	$66.25_{\pm 0.16}$	+2.52
	50	$67.76_{\pm 0.40}$	$69.34_{\pm 0.31}$	+1.58

Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning

Kai Zhu[1] Wei Zhai[1] Yang Cao[1,3,†] Jiebo Luo[2] Zheng-Jun Zha[1]

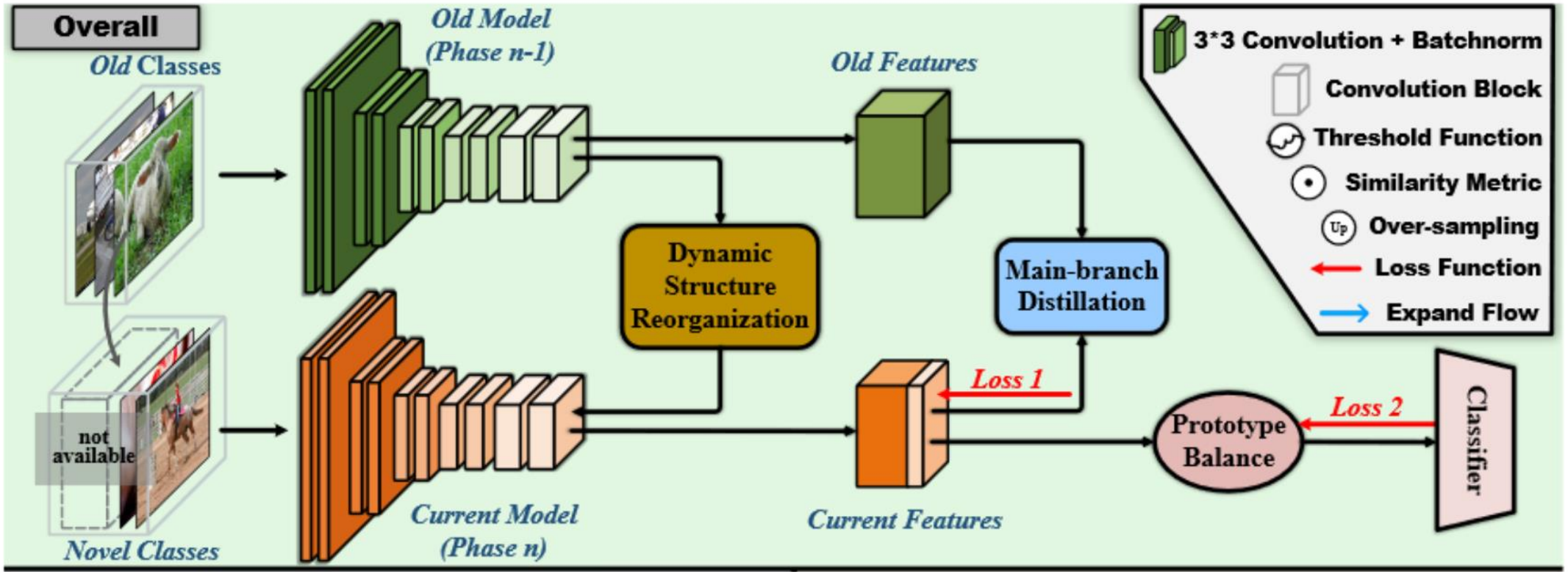
[1] University of Science and Technology of China [2] University of Rochester

[3] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

{zkzy@mail., wzhai056@mail., forrest@}ustc.edu.cn jluo@cs.rochester.edu zhazj@ustc.edu.cn

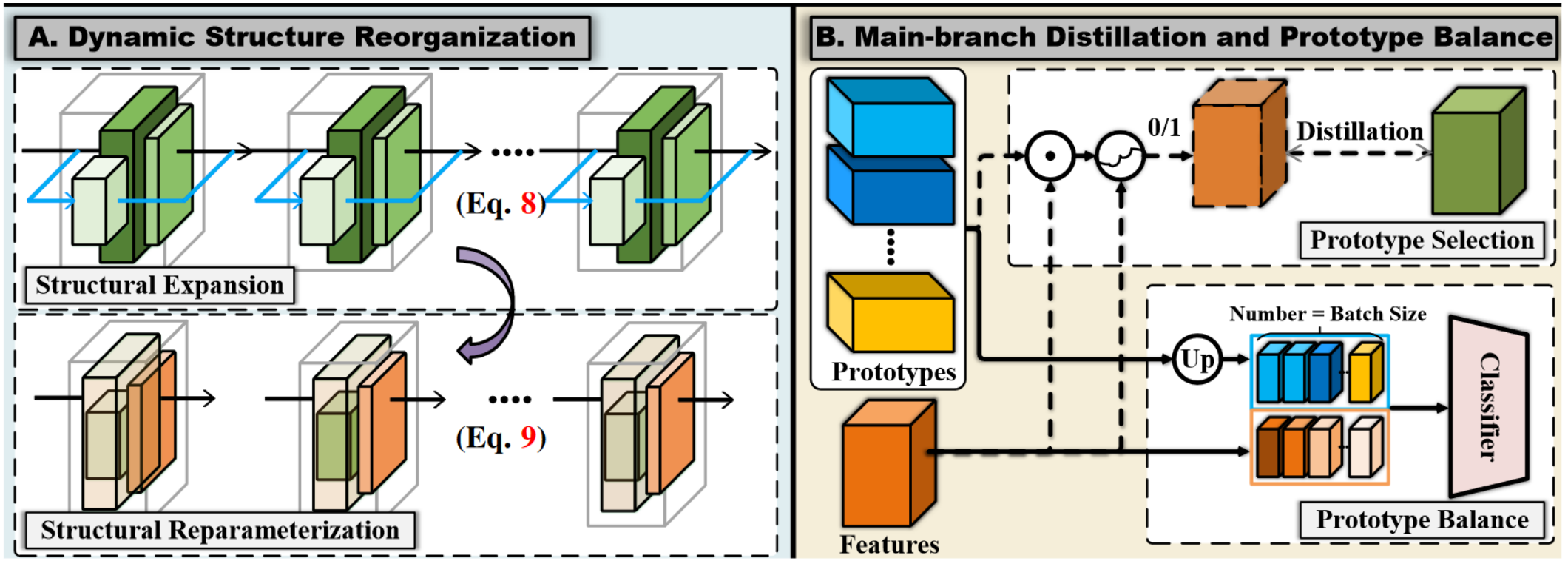
DSR

Overview



DSR

Details


A. Dynamic Structure Reorganization
(Eq. 8)
Structural Expansion
(Eq. 9)
Structural Reparameterization
B. Main-branch Distillation and Prototype Balance
Prototypes
Features
0/1
Distillation
Prototype Selection
Number = Batch Size
Up
Classifier
Prototype Balance

DSR

Results

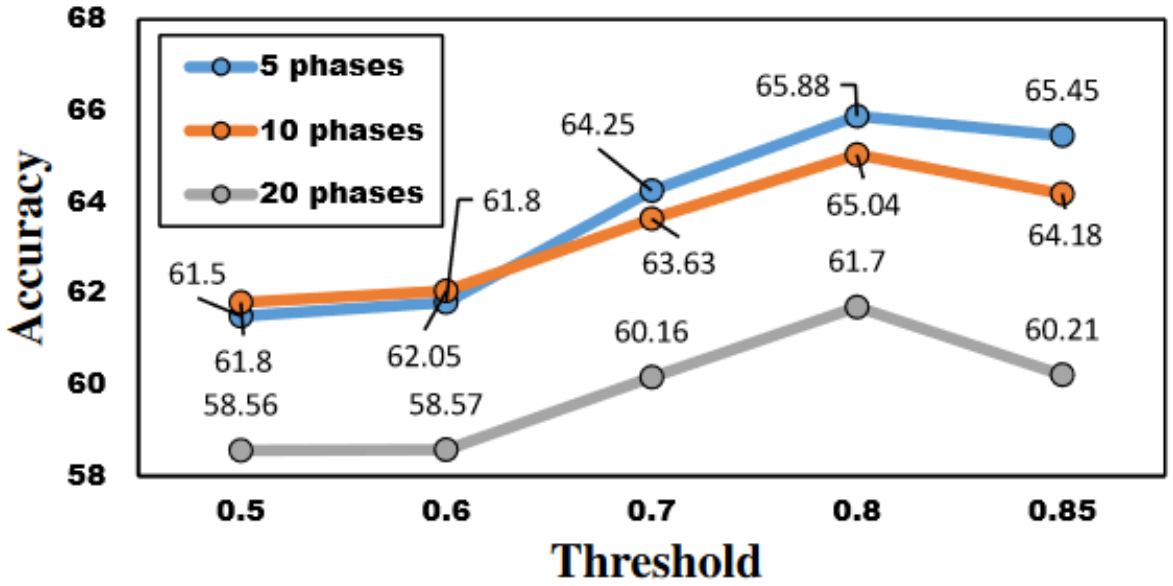
	Methods	CIFAR-100			TinyImageNet			ImageNet-Subset
		P=5	*P=10*	*P=20*	*P=5*	*P=10*	*P=20*	*P=10*
(1) E=20	iCaRL-CNN*	51.07	48.66	44.43	34.64	31.15	27.90	50.53
	iCaRL-NCM* [25]	58.56	54.19	50.51	45.86	43.29	38.04	60.79
	EEIL* [1]	60.37	56.05	52.34	47.12	45.01	40.50	63.34
	UCIR* [10]	63.78	62.39	59.07	49.15	48.52	42.83	66.16
(2) E=0	EWC* [14]	24.48	21.20	15.89	18.80	15.77	12.39	20.40
	LwF_MC* [25]	45.93	27.43	20.07	29.12	23.10	17.43	31.18
	MUC* [34]	49.42	30.19	21.27	32.58	26.61	21.95	35.07
	SDC [35]	56.77	57.00	58.90	-	-	-	61.12
	PASS [37]	63.47	61.84	58.09	49.55	47.29	42.07	61.80
	Ours	**65.88**+2.41	**65.04**+3.20	**61.70**+2.80	**50.39**+0.84	**48.93**+1.64	**48.17**+6.10	**67.69**+5.89

DSR

Ablations

DSR	MBD	PSM	CIFAR-100		
			5 phases	10 phases	20 phases
			61.11	57.08	51.04
√			64.86	63.25	54.09
	√		62.70	62.60	58.57
√	√		65.10	63.87	60.60
√	√	√	65.88	64.69	61.61

	CIFAR-100		
Method	5 phases	10 phases	20 phases
3×3 conv	64.28	63.47	60.81
1×1 conv + bn	65.88	64.84	60.72
1×1 conv	65.87	65.12	61.60



Conclusion

How to handle the compatibility between different tasks?

-> using task-specific design (DyTox, DSR)

-> increase the compatibility of previous representation (CwD)

-> selective optimization (DSR)