Prospective Representation Learning for Non-Exemplar Class-Incremental Learning

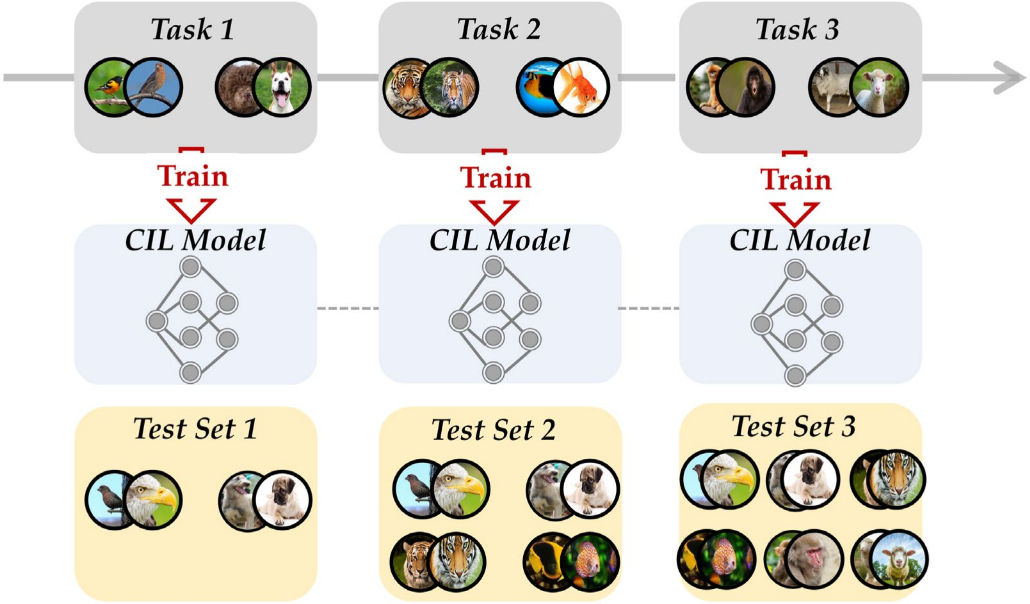
Wuxuan Shi[1] , Mang Ye[1,2*]

* Corresponding Author

[1]School of Computer Science, Wuhan University [2]Hubei Luojia Laboratory

Background

- **Class-Incremental Learning**



① Learns continuously from a sequential data stream in which new classes occur

② Perform multi-class classification for all classes observed so far

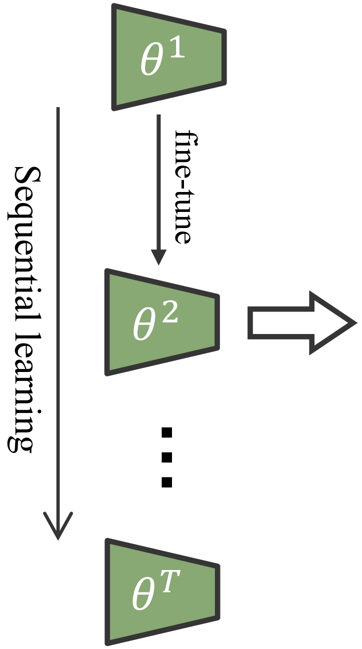
③ Computational requirements and memory footprint remain bounded

[1] Zhou D W, Wang Q W, Qi Z H, et al. Deep class-incremental learning: A survey[J]. arXiv preprint arXiv:2302.03648, 2023.

Background

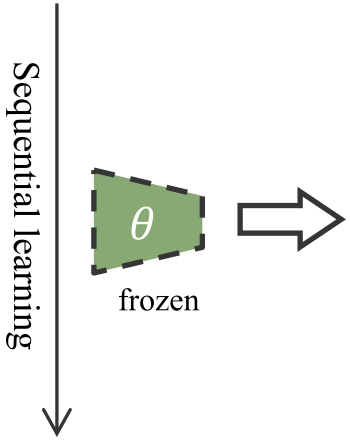
WUHAN UNIVERSITY

- **Stability-Plasticity Dilemma**



Stability Degradation

If we fine-tune the model on new data, the decision boundary of the unified classifier is greatly changed and biased towards the new classes



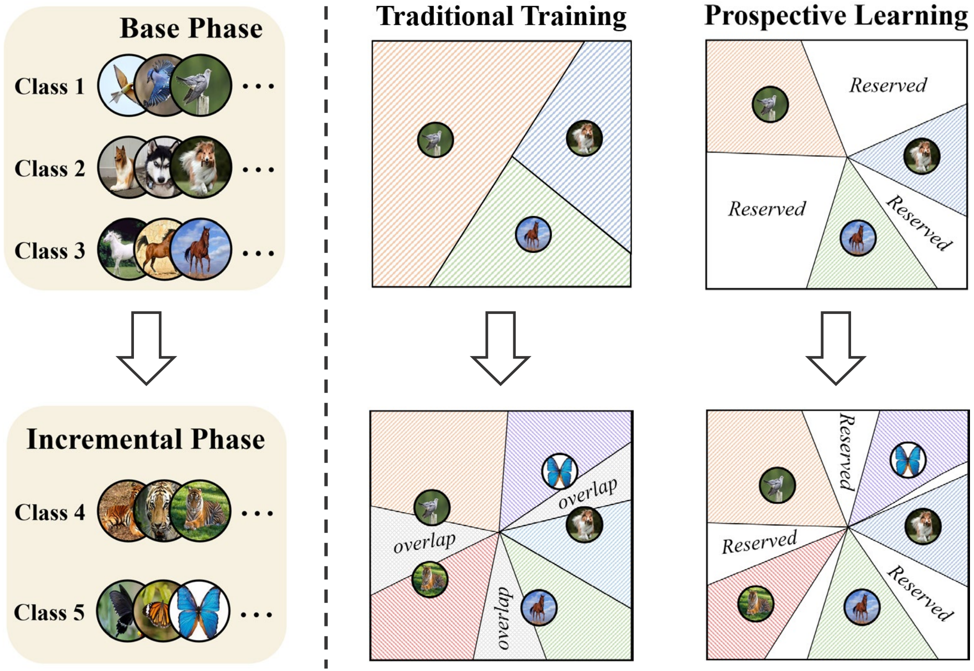
Plasticity Degradation

If we fix the feature embedding space of a trained model, its generalization ability suffers thus performing poorly on new tasks

Motivation

WUHAN UNIVERSITY

- **Conflicts between Old and New classes**



Traditional Training:

① Divide up all the embedding in the base phase

② Deal with the conflicts after a new task comes in

Prospective Learning:

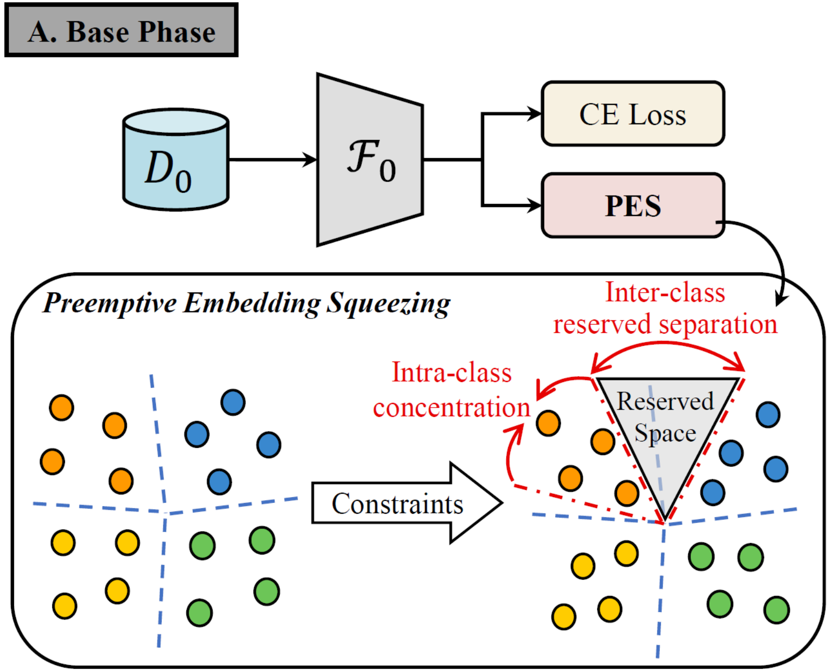
① Reserve space for unknown classes

② Make the newly coming class embedded in the reserved space

Method

WUHAN UNIVERSITY

- **Preemptive Embedding Squeezing (PES)**



During the base phase ($t = 0$):

➢ Reinforce **intra-class concentration** and **inter-class reserved separation**:

$$s = \sum_{\substack{\forall x^i, x^j \in B \\ i \neq j \\ y_i = y_j}} \langle \mathcal{F}_{\theta_t}(x^i), \mathcal{F}_{\theta_t}(x^j) \rangle,$$

$$d = \sum_{\substack{\forall x^i, x^k \in B \\ y_i \neq y_k}} \langle \mathcal{F}_{\theta_t}(x^i), \mathcal{F}_{\theta_t}(x^k) \rangle,$$

$$\mathcal{L}_{PES}(\theta_t; D_t) = (1 - s) + \lambda * (1 + d),$$

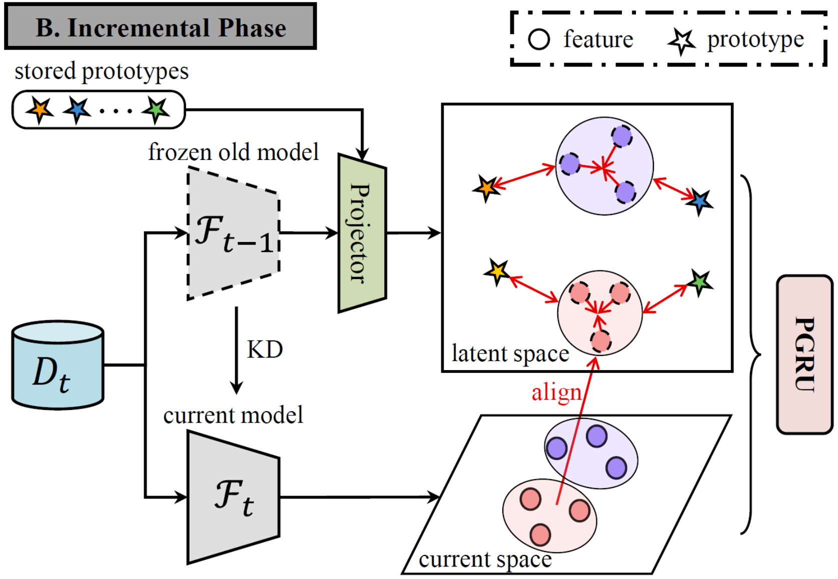
➢ Optimize the model with CE loss:

$$\mathcal{L}_t = \mathcal{L}_{ce}(\theta_t, \varphi_t; D_t) + \gamma * \mathcal{L}_{PES}(\theta_t; D_t).$$

Method

WUHAN UNIVERSITY

- **Prototype-Guided Representation Update (PGRU)**



During the incremental phase ($t > 0$):

➢ Employs prototypes as proxies for past classes to embed new classes into appropriate space:

$$\mathcal{L}_{ort} = \sum_{\substack{\forall x^i \in B \\ \forall \boldsymbol{p}^c \in \boldsymbol{P}_{0:t-1}}} |\langle \mathcal{P}_{\phi_t}(\mathcal{F}_{\theta_{t-1}}(x^i)), \mathcal{P}_{\phi_t}(\boldsymbol{p}^c)\rangle|$$

➢ Align the current embedding space with the latent space:

$$\mathcal{L}_{align} = \sum_{x \in X_i} \mathcal{L}_{MSE}(\mathcal{P}_{\phi_t}(\mathcal{F}_{\theta_{t-1}}(x^i)), \mathcal{F}_{\theta_t}(x^i))$$

➢ PGRU loss:

$$\mathcal{L}_{PGRU} = \mathcal{L}_{ort}(\phi_t; D_t, \boldsymbol{P}_{0:t-1}) + \mathcal{L}_{align}(\theta_t, \phi_t; D_t).$$

Experiment

WUHAN UNIVERSITY

- **Average Incremental Accuracy**

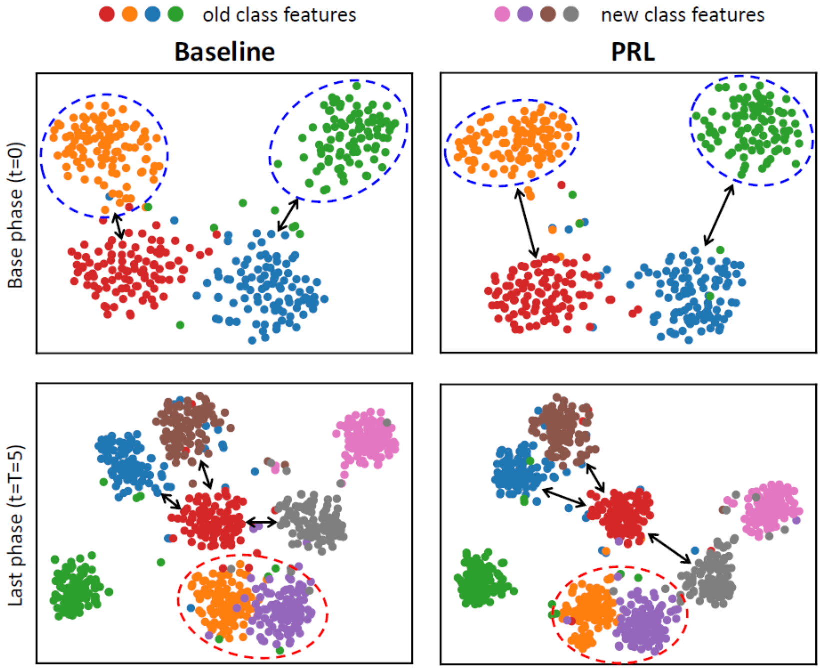
Table 1. Quantitative comparisons of the average incremental accuracy (%) with other methods on CIFAR-100, TinyImageNet, ImageNet-Subset, and ImageNet-1K. *P* represents the number of incremental phases. The best performance is shown in **bold**, and the sub-optimal performance is <u>underlined</u>. The relative improvement compared to the SOTA NECIL methods is shown by the red.

Methods	CIFAR-100			TinyImageNet			ImageNet-Subset			ImageNet-1K
	P=5	P=10	P=20	P=5	P=10	P=20	P=5	P=10	P=20	P=10
Fine-tuning	23.15	12.96	7.93	18.64	10.68	5.75	23.43	13.12	7.96	11.32
Upper Bound	76.72	76.72	76.72	59.12	63.08	63.08	78.94	78.94	78.94	68.58
EWC [17]	24.48	21.20	15.89	18.80	15.77	12.39	—	20.40	—	—
LwF_MC [35]	45.93	27.43	20.07	29.12	23.10	17.43	—	31.18	—	—
MUC [24]	49.42	30.19	21.27	32.58	26.61	21.95	—	35.07	—	—
SDC [50]	56.77	57.00	58.90	—	—	—	—	61.12	—	—
PASS [57]	63.47	61.84	58.09	49.55	47.29	42.07	64.40	61.80	51.29	55.90
SSRE [58]	65.88	65.04	61.70	50.39	48.93	48.17	—	67.69	—	58.12
SOPE [59]	66.64	65.84	61.83	53.69	52.88	<u>51.94</u>	—	<u>69.22</u>	—	<u>60.20</u>
POLO [45]	68.95	68.02	65.71	<u>54.90</u>	<u>53.38</u>	49.93	<u>70.81</u>	69.11	—	—
PRAKA [40]	70.02	68.86	65.86	53.32	52.61	49.83	69.81	68.98	<u>63.95</u>	57.42
NAPA-VQ [28]	<u>70.44</u>	<u>69.04</u>	<u>67.42</u>	52.77	51.78	49.51	69.15	68.83	63.09	54.21
PRL(Ours)	**71.29**	**70.23**	**68.32**	**58.16**	**57.04**	**54.71**	**72.85**	**71.50**	**66.88**	**62.57**
Improvement	+0.85	+1.19	+0.90	+3.26	+3.66	+2.77	+2.04	+2.28	+2.93	+2.37

Experiment

WUHAN UNIVERSITY

- **Visualization**



Highlight inter-class separation

Highlight intra-class concentration

Highlight overlap between old and new classes

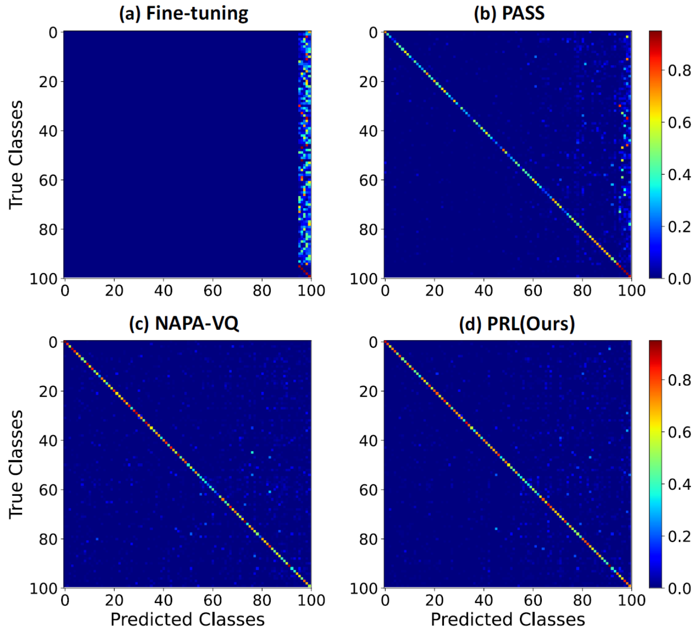
Model with PRL:

- More clustered intra-class distributions
- More dispersed inter-class distributions

Experiment

WUHAN UNIVERSITY

- ## Plasticity and stability analysis



Confusion matrix

The **diagonal entries** indicate correct classification
The **non-diagonal entries** indicate misclassification

Finetune: strong confusion on the last task

PASS: slightly biased toward recently learned tasks

NAPA-VQ: more accurate for the initial classes

PRL(Ours): higher average accuracy, more balanced performance on old and new classes

Thank you !

https://github.com/ShiWuxuan/NeurIPS2024-PRL