Advancing Cross-domain Discriminability in Continual Learning of Vision-Language Models

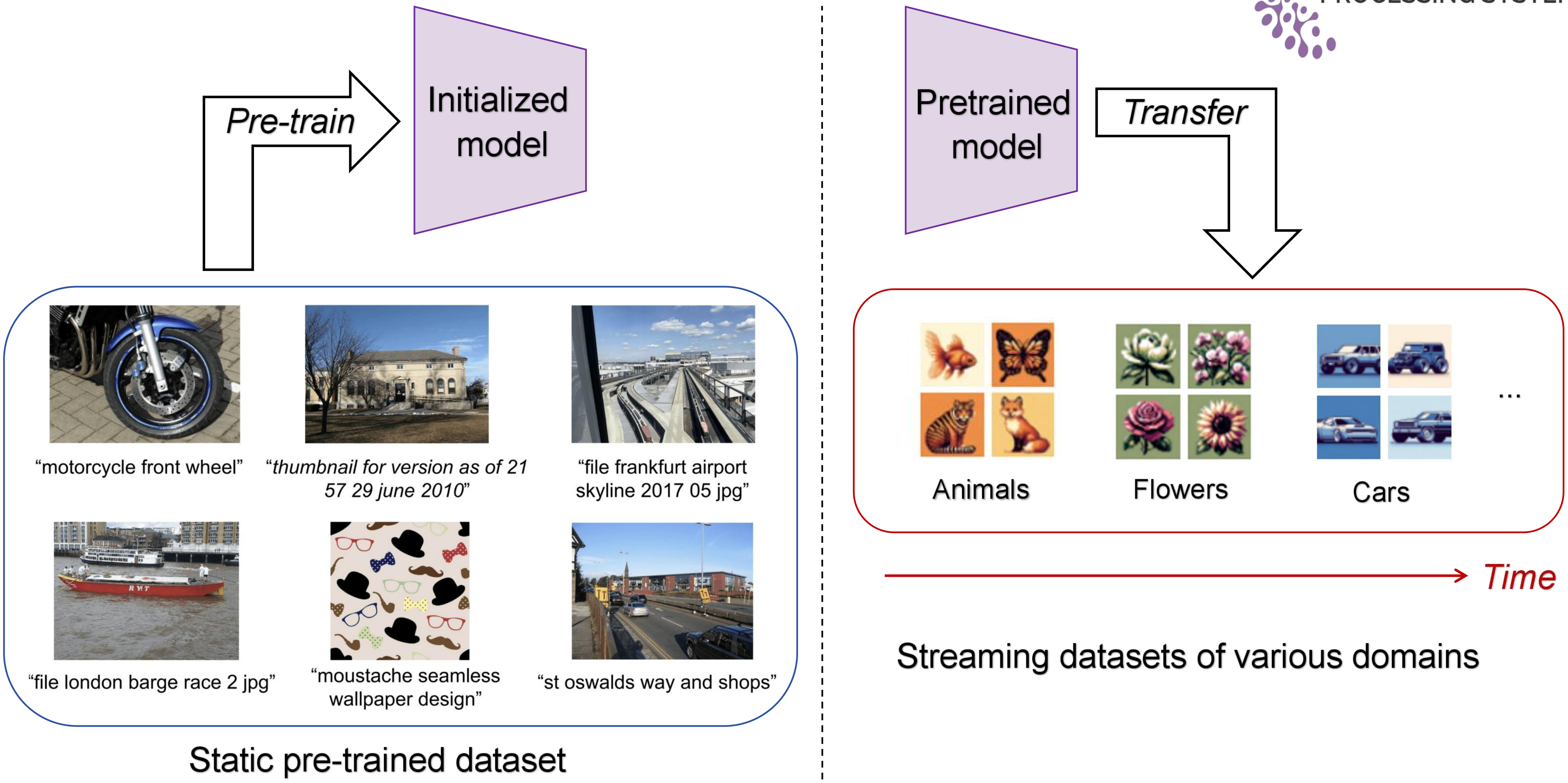
NEURAL INFORMATION PROCESSING SYSTEMS

Yicheng Xu, Yuxin Chen, Jiahao Nie, Yusong Wang, Huiping Zhuang, Manabu Okumura

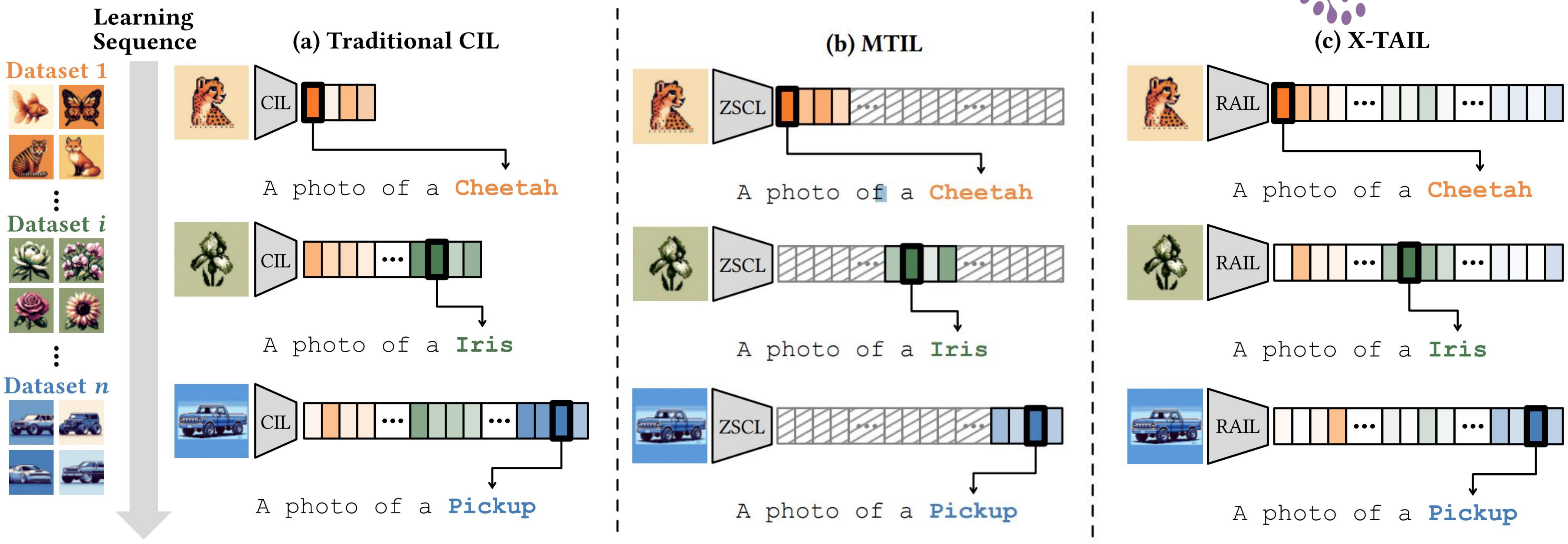
Institute of Science Tokyo
(Tokyo Institute of Technology)

Pre-training & Continual Learning

NEURAL INFORMATION PROCESSING SYSTEMS



Continual Learning of Vision-Language Model



Class-Incremental Learning:
models classify images within only previously encountered classes.

Multi-Task Incremental Learning:
models classify images from both seen and unseen domains based on the given domain-identities.

Cross-domain Task-Agnostic Incremental Learning:
models classify images from both seen and unseen domains without any domain-identity hint.

Motivation

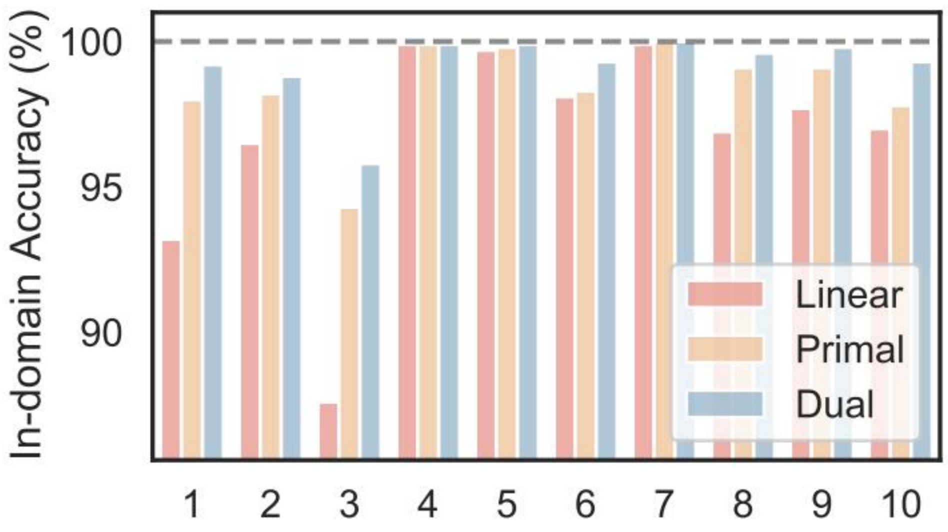
NEURAL INFORMATION PROCESSING SYSTEMS

Challenges

- *How to preserve the zero-shot ability of the pre-trained VLM?*
- *How to distinguish data from different newly learned domains?*
- *How to avoid forgetting on continually learned domains?*

Solutions

- Freeze the pre-trained VLM.
- Cooperate primal & dual regression methods with non-linear projections.
- Extend the closed-form solutions of regression methods to an continual learning manner.



Both primal & dual regression methods can classify images into their respective domains accurately without domain identity hint.

Non-forgetting Solutions

NEURAL INFORMATION PROCESSING SYSTEMS

Optimization target:

$$\underset{\mathbf{W}^{(n)}}{\arg\min} \left\| \mathbf{Y}^{(1:n)} - \mathbf{\Phi}^{(1:n)}\mathbf{W}^{(n)} \right\|_F^2 + \lambda \left\| \mathbf{W}^{(n)} \right\|_F^2$$

Standard solutions

Continual learning forms

Ridge Regression:

$$\mathbf{W} = \left(\mathbf{\Phi}^\top \mathbf{\Phi} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Phi}^\top \mathbf{Y}$$

Theorem 1 *The parameter calculated by*

$$\mathbf{W}^{(n)} = \begin{bmatrix} \mathbf{W}^{(n-1)} - \mathbf{M}_p^{(n)} \mathbf{\Phi}^{(n)\top} \mathbf{\Phi}^{(n)} \mathbf{W}^{(n-1)} & \mathbf{M}_p^{(n)} \mathbf{\Phi}^{(n)\top} \mathbf{Y}^{(n)} \end{bmatrix}$$

is an optimal solution to the optimization problem of joint training on all n domains in Eqn. 4, where $\mathbf{M}_p^{(n)}$ is obtained by

$$\mathbf{M}_p^{(n)} = \mathbf{M}_p^{(n-1)} - \mathbf{M}_p^{(n-1)} \mathbf{\Phi}^{(n)\top} \left(\mathbf{I} + \mathbf{\Phi}^{(n)} \mathbf{M}_p^{(n-1)} \mathbf{\Phi}^{(n)\top} \right)^{-1} \mathbf{\Phi}^{(n)} \mathbf{M}_p^{(n-1)}.$$

Dual Ridge Regression:

$$\boldsymbol{\alpha} = \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{Y}$$

Theorem 2 *The parameter calculated by*

$$\boldsymbol{\alpha}^{(n)} = \left(\mathbf{K}^{(n)} + \lambda \mathbf{I} \right)^{-1} \mathbf{C}^{(n)}$$

is an optimal solution to the optimization problem of joint training on all n domains in Eqn. 4, where

$$\mathbf{K}^{(n)} = \begin{bmatrix} \mathbf{K}^{(n-1)} & \mathcal{K}\left(\mathbf{X}_e^{(n)}, \mathbf{M}_d^{(n-1)}\right)^\top \\ \mathcal{K}\left(\mathbf{X}_e^{(n)}, \mathbf{M}_d^{(n-1)}\right) & \mathcal{K}\left(\mathbf{X}_e^{(n)}, \mathbf{X}_e^{(n)}\right) \end{bmatrix}, \quad \mathbf{C}^{(n)} = \begin{bmatrix} \mathbf{C}^{(n-1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}^{(n)} \end{bmatrix},$$

and the memory matrix is given by $\mathbf{M}_d^{(n)} = \begin{bmatrix} \mathbf{M}_d^{(n-1)\top} & \mathbf{X}_e^{(n)\top} \end{bmatrix}^\top$.

Proposed Method: RAIL

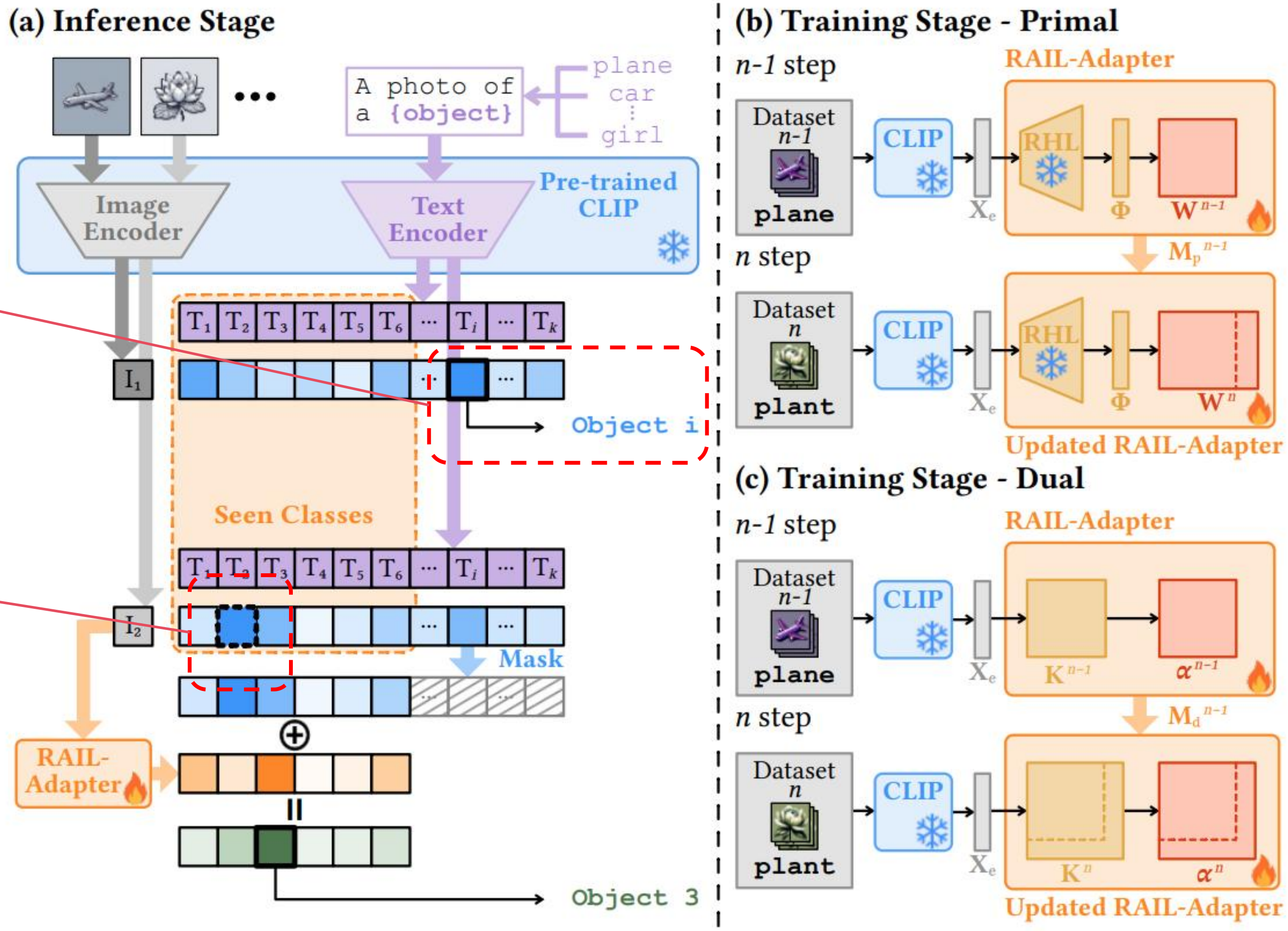
Regression-based Analytic Incremental Learning

i) Determine if the test image belongs to seen domains (ID) or unseen domains (OOD).

OOD

ID

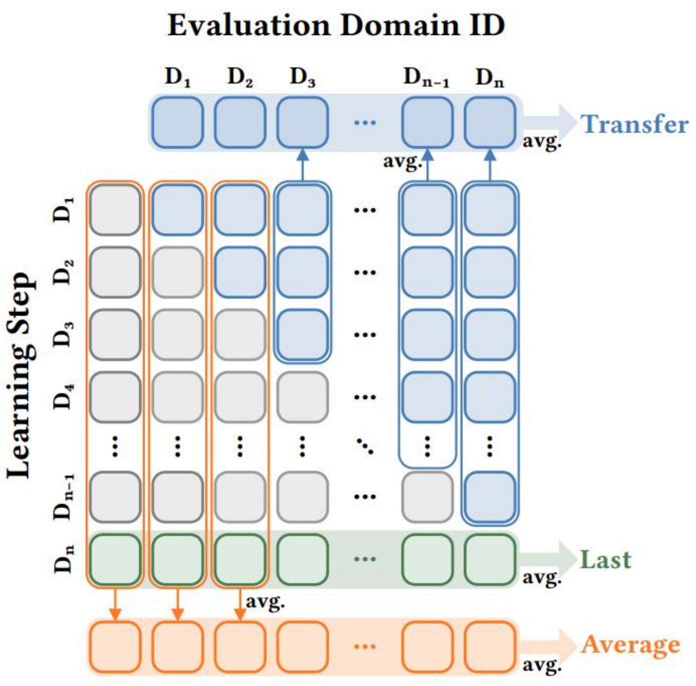
ii) Refine the prediction through RAIL adapter.



Train on streaming datasets based on aforementioned solutions.

Experiments

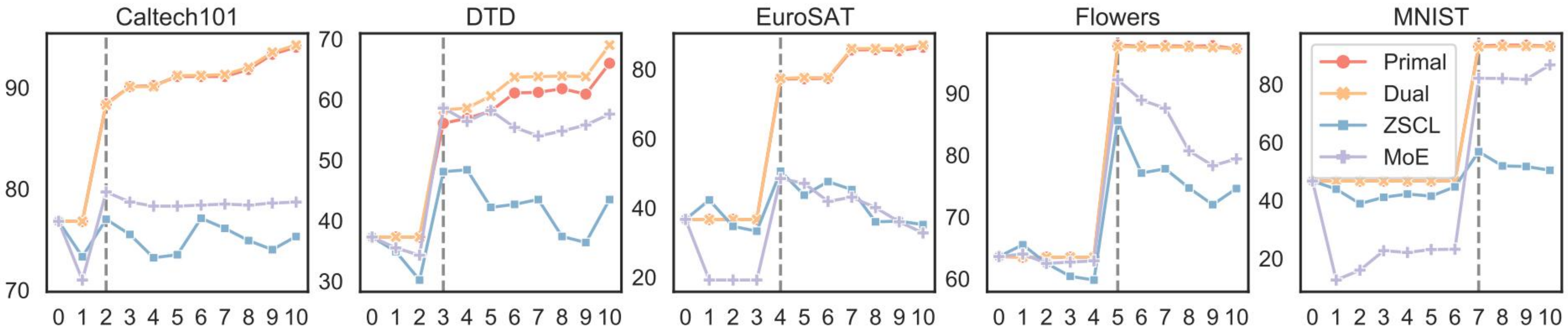
NEURAL INFORMATION PROCESSING SYSTEMS



- *Transfer: the extent to which the zero-shot ability is preserved.*
- *Last: the learner's adaptability to new domains.*
- *Average: the average accuracy of all learning steps across all domains.*

Method	Aircraft	Caltech101	DTD	EuroSAT	Flowers	Food101	MNIST	Pets	Cars	Sun397	Average
Zero-shot	23.5	76.8	37.3	36.7	63.6	84.0	46.7	86.7	66.1	63.7	58.5
Fine-tune	39.6	93.3	68.2	89.2	95.4	85.5	95.1	84.4	77.4	72.4	80.1
Transfer											
LwF [6]	–	66.6	26.9	19.5	51.0	78.4	26.6	68.9	35.5	56.1	47.7
WiSE-FT [47]	–	70.1	31.9	25.3	56.3	79.8	29.9	74.9	45.6	56.8	52.3
iCaRL [7]	–	71.7	35.0	**43.0**	63.4	**86.9**	43.9	**87.8**	63.7	60.0	61.7
ZSCL [10]	–	73.3	32.6	36.8	62.1	83.8	42.1	83.6	56.5	60.2	59.0
MoE-Adapter [15]	–	71.0	34.9	19.2	63.0	86.6	20.0	87.2	63.7	58.6	56.0
Primal-RAIL	–	**76.8**	**37.3**	36.7	**63.6**	84.0	**46.7**	86.7	**66.1**	**63.7**	**62.4**
Dual-RAIL	–	**76.8**	**37.3**	36.7	**63.6**	84.0	**46.7**	86.7	**66.1**	**63.7**	**62.4**
Average											
LwF	24.7	79.7	38.3	36.9	63.9	81.0	36.5	71.9	42.7	56.7	53.2
WiSE-FT	27.1	76.5	40.9	31.3	68.7	81.6	31.4	74.7	51.7	58.4	54.2
iCaRL	25.4	72.1	37.5	51.6	65.1	**87.1**	59.1	88.0	63.7	60.1	61.0
ZSCL	36.0	75.0	40.7	40.5	71.0	85.3	46.3	83.3	60.7	61.5	60.0
MoE-Adapter	43.6	77.9	52.1	34.7	75.9	86.3	45.2	87.4	66.6	60.2	63.0
Primal-RAIL	42.4	89.8	55.7	68.5	**84.0**	83.3	**65.3**	85.8	67.9	64.5	70.7
Dual-RAIL	**45.3**	**89.9**	**57.6**	**68.7**	83.9	85.5	65.2	**88.4**	**69.4**	**65.0**	**71.9**
Last											
LwF	20.9	83.1	47.5	38.2	75.5	84.7	50.1	78.0	75.8	74.6	62.8
WiSE-FT	21.8	76.8	42.9	20.8	77.5	84.9	30.7	76.6	75.8	72.5	58.0
iCaRL	25.5	72.1	38.9	55.4	65.5	87.3	81.9	88.6	63.6	61.5	64.0
ZSCL	33.1	75.3	43.5	35.2	74.6	**87.4**	50.4	84.2	77.3	73.4	63.4
MoE-Adapter	43.2	78.7	57.6	32.8	79.4	86.0	86.7	87.8	78.2	74.2	70.5
Primal-RAIL	41.7	94.0	66.0	86.4	**97.2**	82.4	**93.1**	83.6	75.0	71.3	79.1
Dual-RAIL	**45.3**	**94.2**	**69.0**	**87.0**	**97.2**	87.2	93.0	**92.4**	**82.5**	**76.3**	**82.4**

Experiments



Accuracy (%) on five domains changes over all learning steps.

Speed Analysis

Model	Real time
ZSCL	514m 40.163s
Moe-Adapter	47m 2.319s
Primal-RAIL	4m 0.071s
Dual-RAIL	4m 13.200s

- ***No reference dataset.***
- ***Parameter efficiency.***
- ***Closed-form solutions -> require only one epoch!***

NEURAL INFORMATION PROCESSING SYSTEMS

Github

Paper

Institute of SCIENCE TOKYO

Thanks for your watching!