

# Incremental Learning

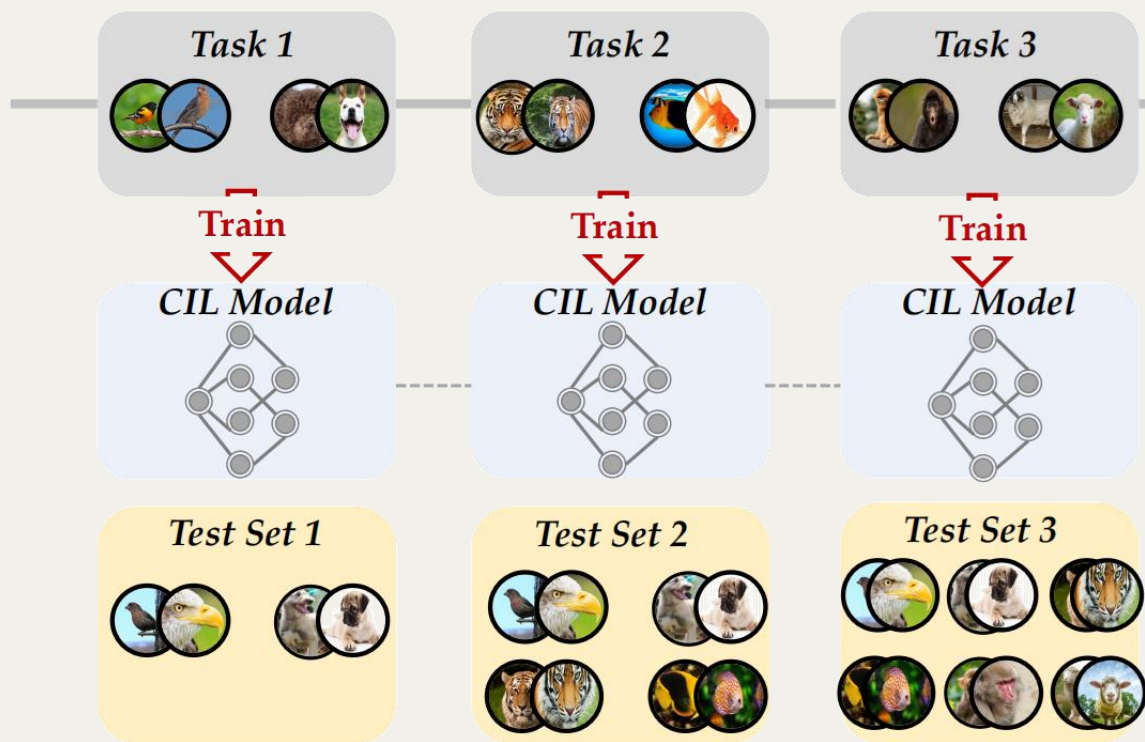


马鸣霄

2025.09.12

# Incremental learning

增量学习 ( Incremental Learning, 持续学习(Continual Learning) )



**动机：**随着时间的推移，更多的**新数据逐渐可用**，同时**旧数据**可能由于存储限制或隐私保护等原因**逐渐不可用**。

**能力：**不断地处理现实世界中连续的信息流，在吸收新知识的同时保留甚至整合、优化旧知识的能力。<sup>[1]</sup>

**操作：**通过对新的复杂多变环境下的数据进行**持续学习**，**而不是重头训练整个模型**。

[1] Parisi G I, Kemker R, Part J L, et al. Continual lifelong learning with neural networks: A review[J]. Neural networks, 2019, 113: 54-71.

[2] Zhou D W, Wang Q W, Qi Z H, et al. Class-incremental learning: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

# Incremental learning

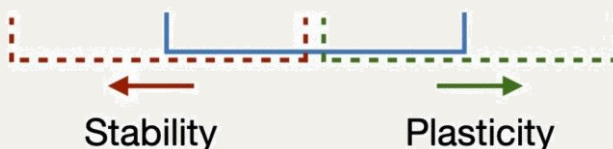
**关键点：**灾难性遗忘（catastrophic forgetting），此长彼消

**原因：**原有的固定数据→连续的数据流，模型由平稳→非平稳，Stability-Plasticity dilemma

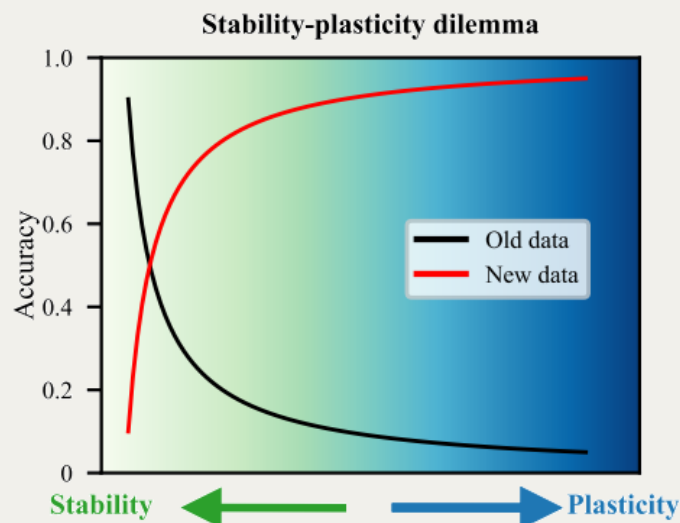
## 稳定性（memory stability）

防止新输入对已有知识的显著干扰

→参数不变



VS



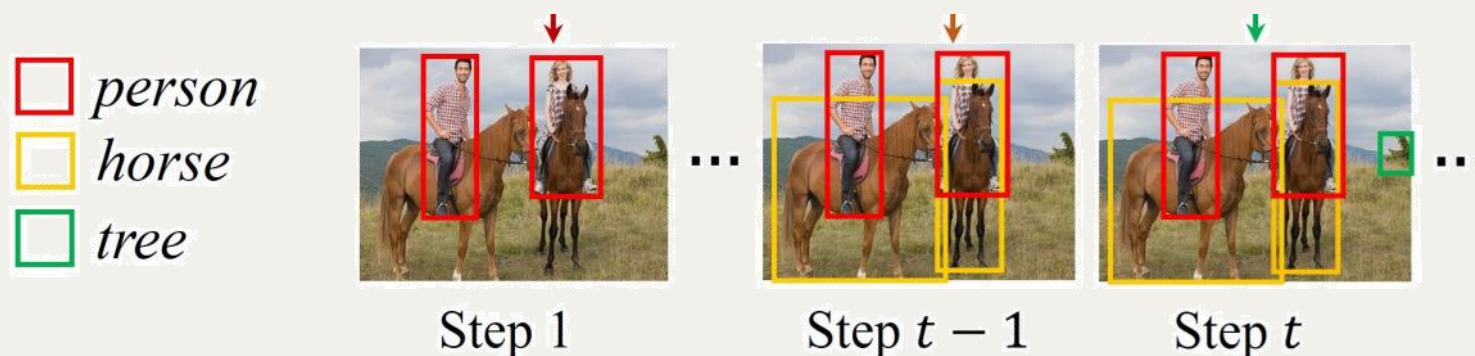
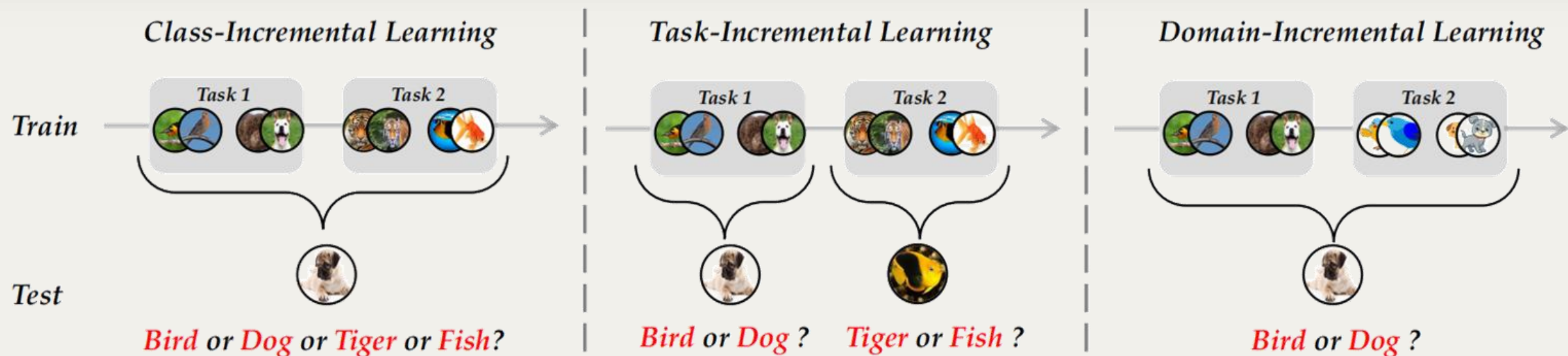
## 可塑性（learning plasticity）

从新数据中整合新知识和提炼已有知识的能力

→参数可变



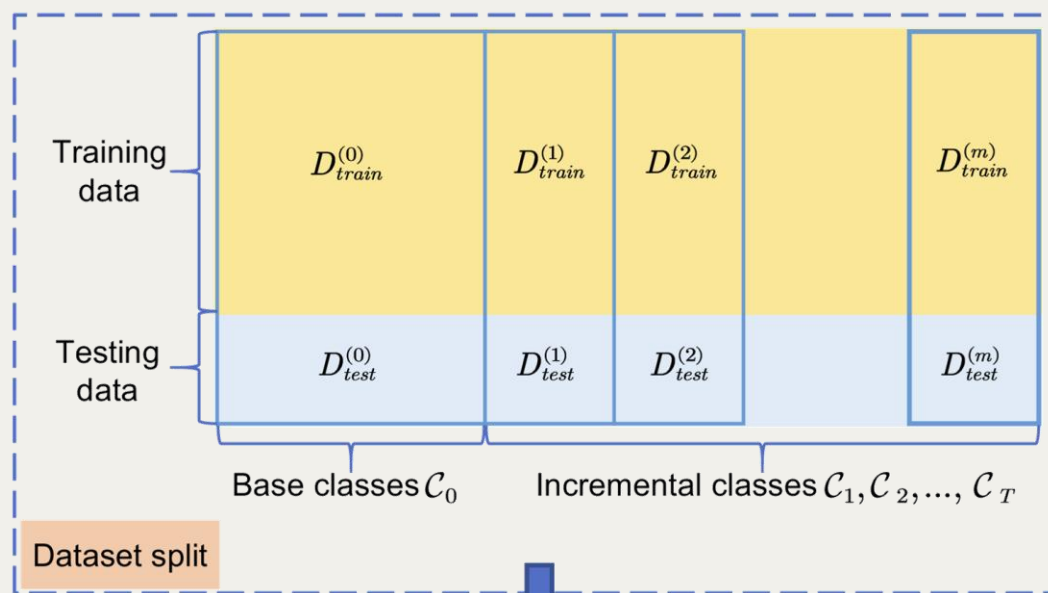
# Incremental learning



[1] Zhou D W, Wang Q W, Qi Z H, et al. Class-incremental learning: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

# Incremental learning

## Class Incremental Learning:



**Definition 1. Class-Incremental Learning** aims to learn from an evolutive stream with new classes. Assume there is a sequence of  $B$  training tasks<sup>1</sup>  $\{\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^B\}$  without overlapping classes, where  $\mathcal{D}^b = \{(\mathbf{x}_i^b, y_i^b)\}_{i=1}^{n_b}$  is the  $b$ -th incremental step with  $n_b$  training instances.  $\mathbf{x}_i^b \in \mathbb{R}^D$  is an instance of class  $y_i^b \in Y_b$ ,  $Y_b$  is the label space of task  $b$ , where  $Y_b \cap Y_{b'} = \emptyset$  for  $b \neq b'$ . We can only access data from  $\mathcal{D}^b$  when training task  $b$ . The ultimate goal of CIL is to continually build a classification model for all classes. In other words, the model should not only acquire the knowledge from the current task  $\mathcal{D}^b$  but also preserve the knowledge from former tasks. After each task, the trained model is evaluated over all seen classes  $\mathcal{Y}_b = Y_1 \cup \dots \cup Y_b$ . Formally, CIL aims to fit a

## External Knowledge Injection for CLIP-Based Class-Incremental Learning

Da-Wei Zhou<sup>1,2</sup>, Kai-Wen Li<sup>1,2</sup>, Jingyi Ning<sup>2</sup>, Han-Jia Ye<sup>1,2</sup>(✉), Lijun Zhang<sup>1,2</sup>, De-Chuan Zhan<sup>1,2</sup>

<sup>1</sup> School of Artificial Intelligence, Nanjing University

<sup>2</sup> National Key Laboratory for Novel Software Technology, Nanjing University

{zhoudw, likw, yehj, zhanglj, zhandc}@lamda.nju.edu.cn, ningjy@nju.edu.cn

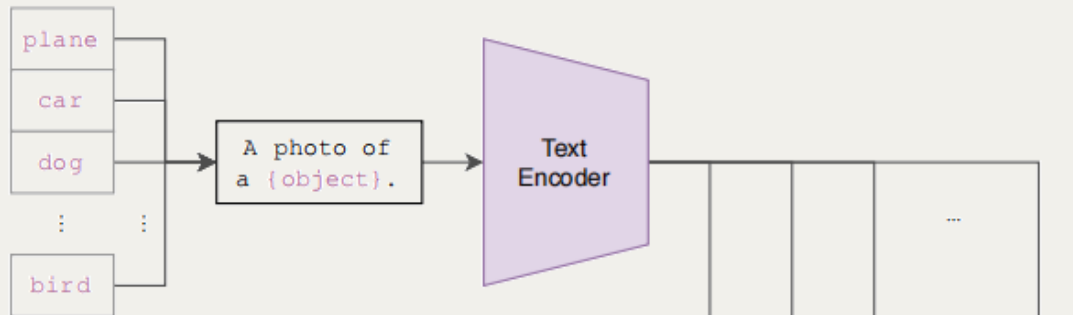


# External Knowledge Injection for CLIP-Based Class-Incremental Learning

## Task & Setting: Class Incremental Learning, CLIP

### Motivation: Why clip?

(2) Create dataset classifier from label text



(3) Use for zero-shot prediction

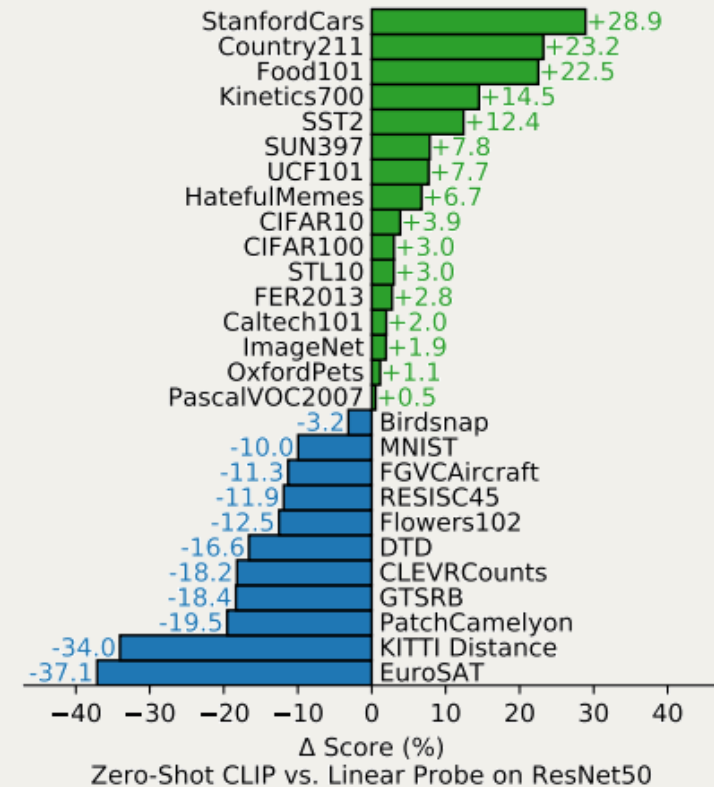
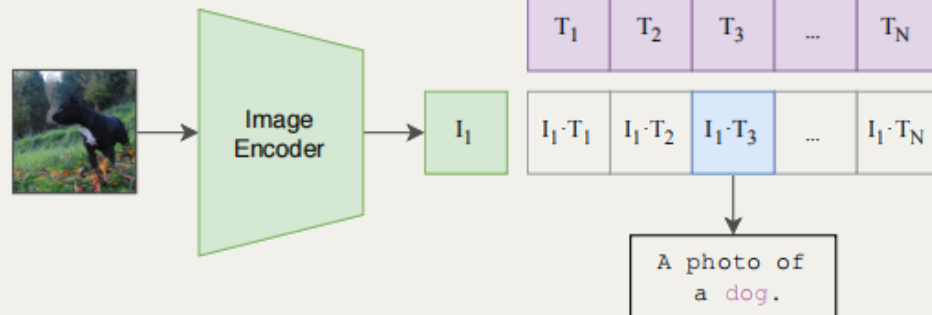
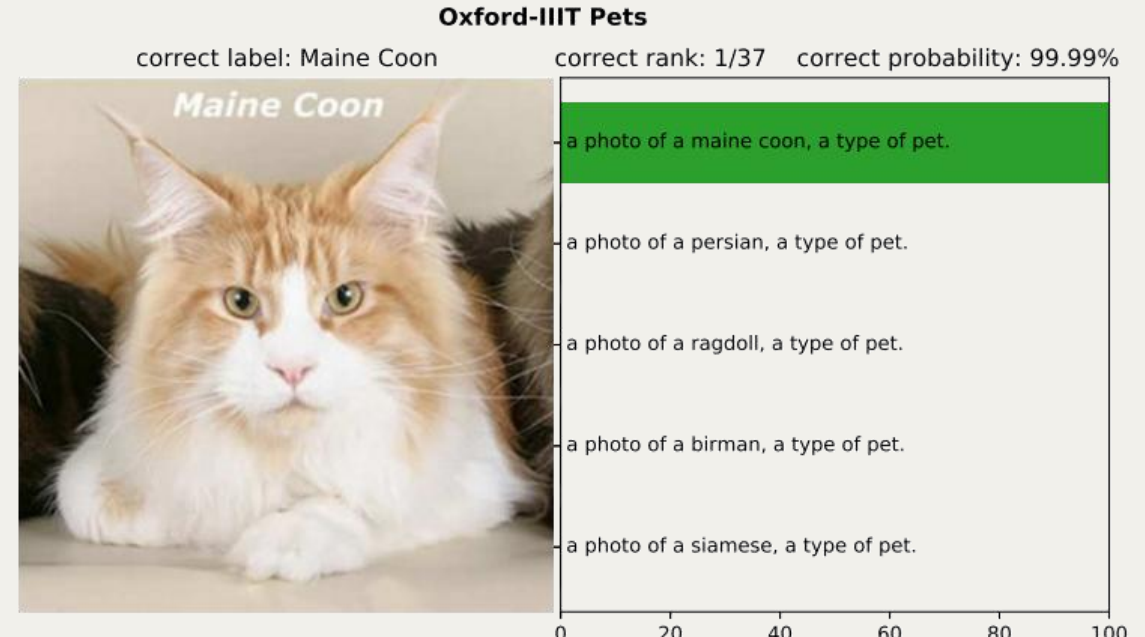
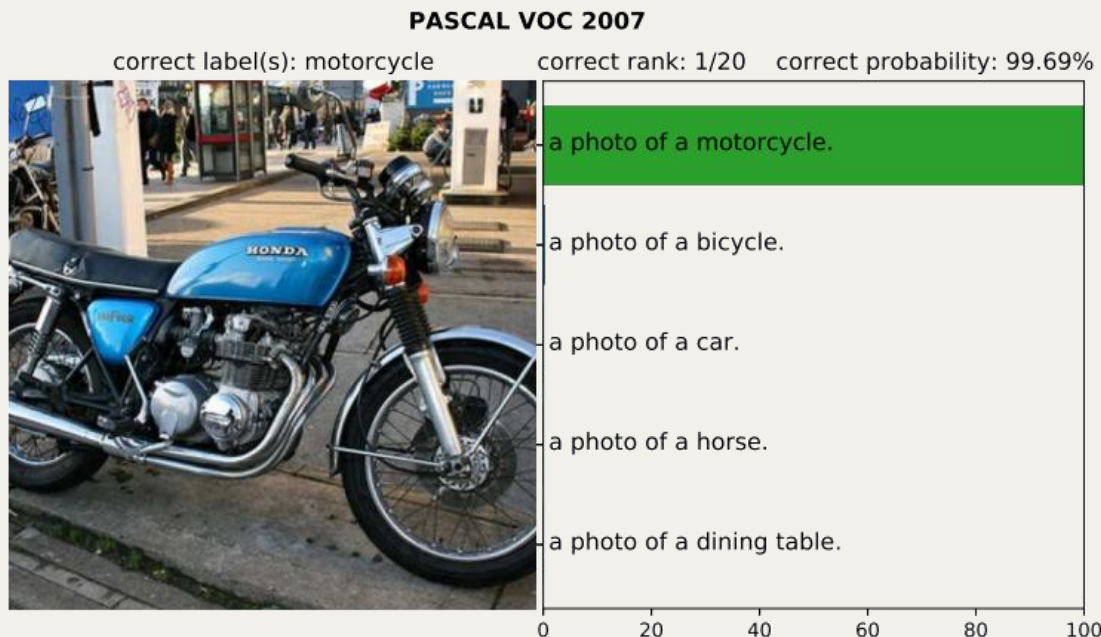


Figure 5. Zero-shot CLIP is competitive with a fully supervised baseline. Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

# External Knowledge Injection for CLIP-Based Class-Incremental Learning

Task & Setting: Class Incremental Learning,

Motivation: Visual features often contain fine-grained information, these detailed descriptors are neglected when using CLIP class names “A photo of a {label}” as matching targets.



[1] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PmLR, 2021: 8748-8763.



# External Knowledge Injection for CLIP-Based Class-Incremental Learning

## Motivation:







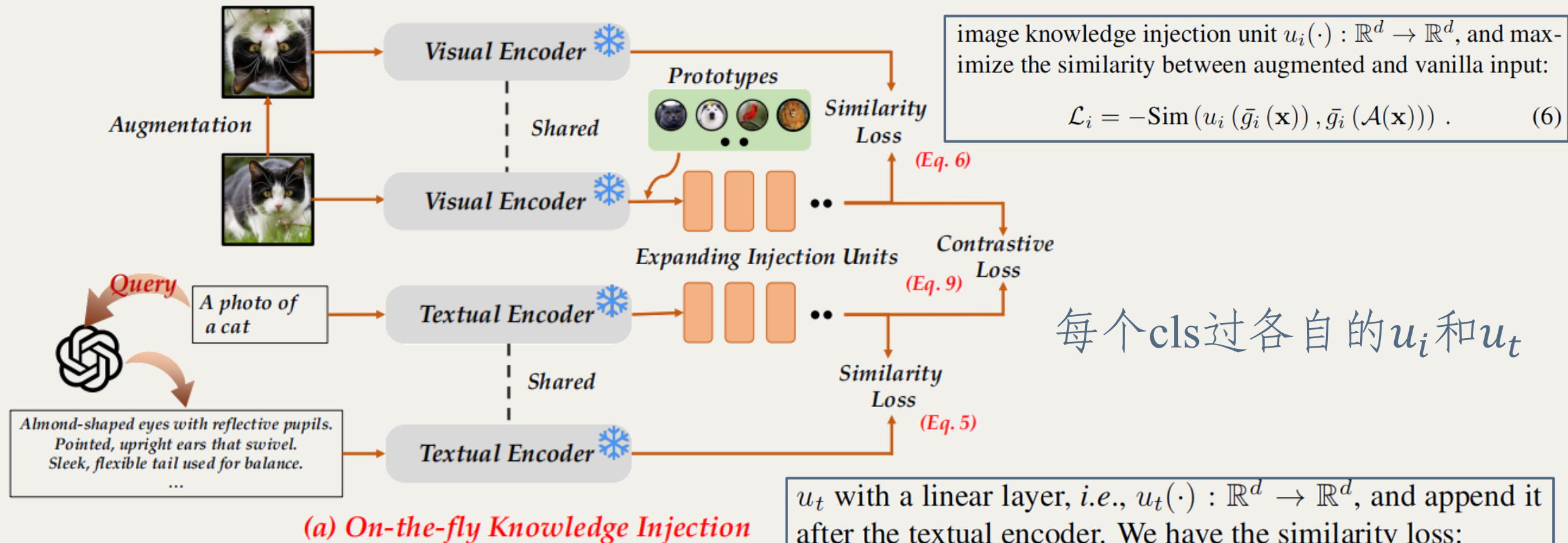
	Caltech101	Prompt	Accuracy
		a [CLASS].	80.77
		a photo of [CLASS].	78.99
		a photo of a [CLASS].	84.42
		[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	92.00
(a)			
	Flowers102	Prompt	Accuracy
		a photo of a [CLASS].	56.68
		a flower photo of a [CLASS].	61.23
		a photo of a [CLASS], a type of flower.	62.32
		[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	93.22
(b)			
	Describable Textures (DTD)	Prompt	Accuracy
		a photo of a [CLASS].	38.24
		a photo of a [CLASS] texture.	37.71
		[CLASS] texture.	40.72
		[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	62.55
(c)			
	EuroSAT	Prompt	Accuracy
		a photo of a [CLASS].	22.30
		a satellite photo of [CLASS].	31.12
		a centered satellite photo of [CLASS].	31.53
		[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	81.60
(d)			

Figure 1: **Prompt engineering vs. context optimization (CoOp).** The latter uses only 16 shots for learning in these examples.

[1] Zhou K, Yang J, Loy C C, et al. Learning to prompt for vision-language models[J]. International Journal of Computer Vision, 2022, 130(9): 2337-2348,

# External Knowledge Injection for CLIP-Based Class-Incremental Learning

## Method:



we seek help from GPT-4 [1] to provide *discriminative visual features*:

**Q:** What are unique visual features of [CLASS]<sub>i</sub> in a photo? Focus on the key visual features.

**A:** 1. Long, thin tail that aids in balance. 2. ...

# External Knowledge Injection for CLIP-Based Class-Incremental Learning

## Method:

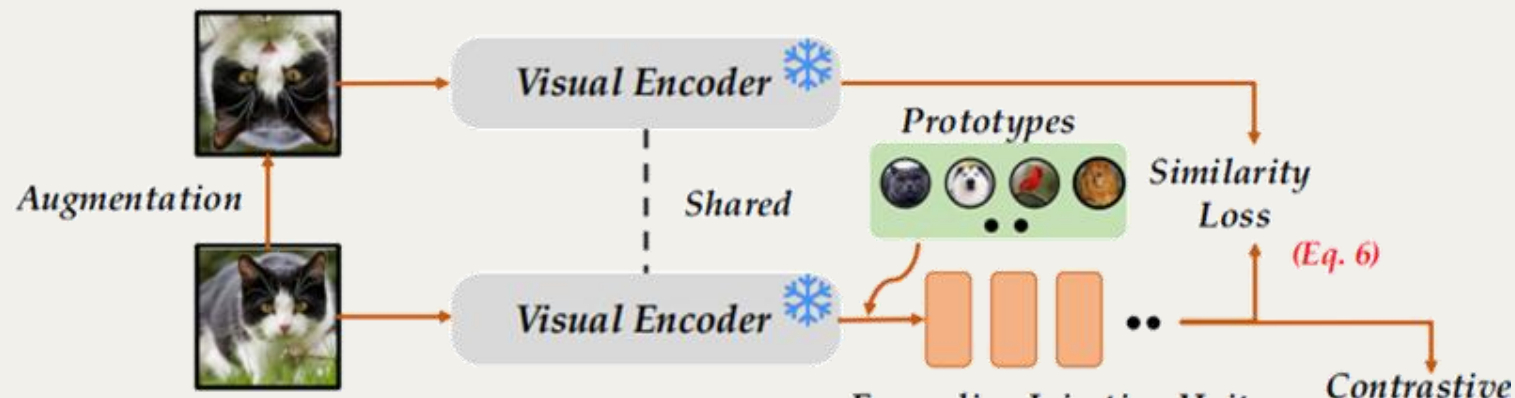


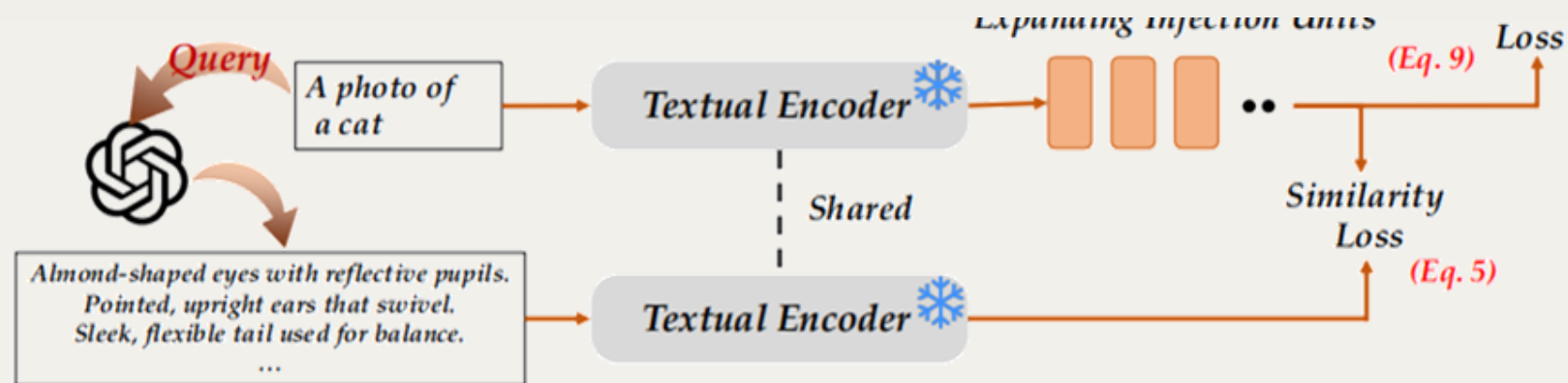
image knowledge injection unit  $u_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ , and maximize the similarity between augmented and vanilla input:

$$\mathcal{L}_i = -\text{Sim}(u_i(\bar{g}_i(\mathbf{x})), \bar{g}_i(\mathcal{A}(\mathbf{x}))) . \quad (6)$$

每个cls过各自的 $u_i$ 和 $u_t$

# External Knowledge Injection for CLIP-Based Class-Incremental Learning

## Method:



(a) On-the-fly Knowledge Injection

we seek help from GPT-4 [1] to provide *discriminative visual features*:

**Q:** What are unique visual features of [CLASS]<sub>i</sub> in a photo? Focus on the key visual features.

**A:** 1. Long, thin tail that aids in balance. 2. ...

$u_t$  with a linear layer, i.e.,  $u_t(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ , and append it after the textual encoder. We have the similarity loss:

$$\mathcal{L}_t = -\text{Sim}(u_t(\bar{g}_t(\mathbf{t}_i)), \bar{g}_t(\mathbf{d}_i)), \quad (5)$$

每个cls过各自的 $u_i$ 和 $u_t$



# External Knowledge Injection for CLIP-Based Class-Incremental Learning

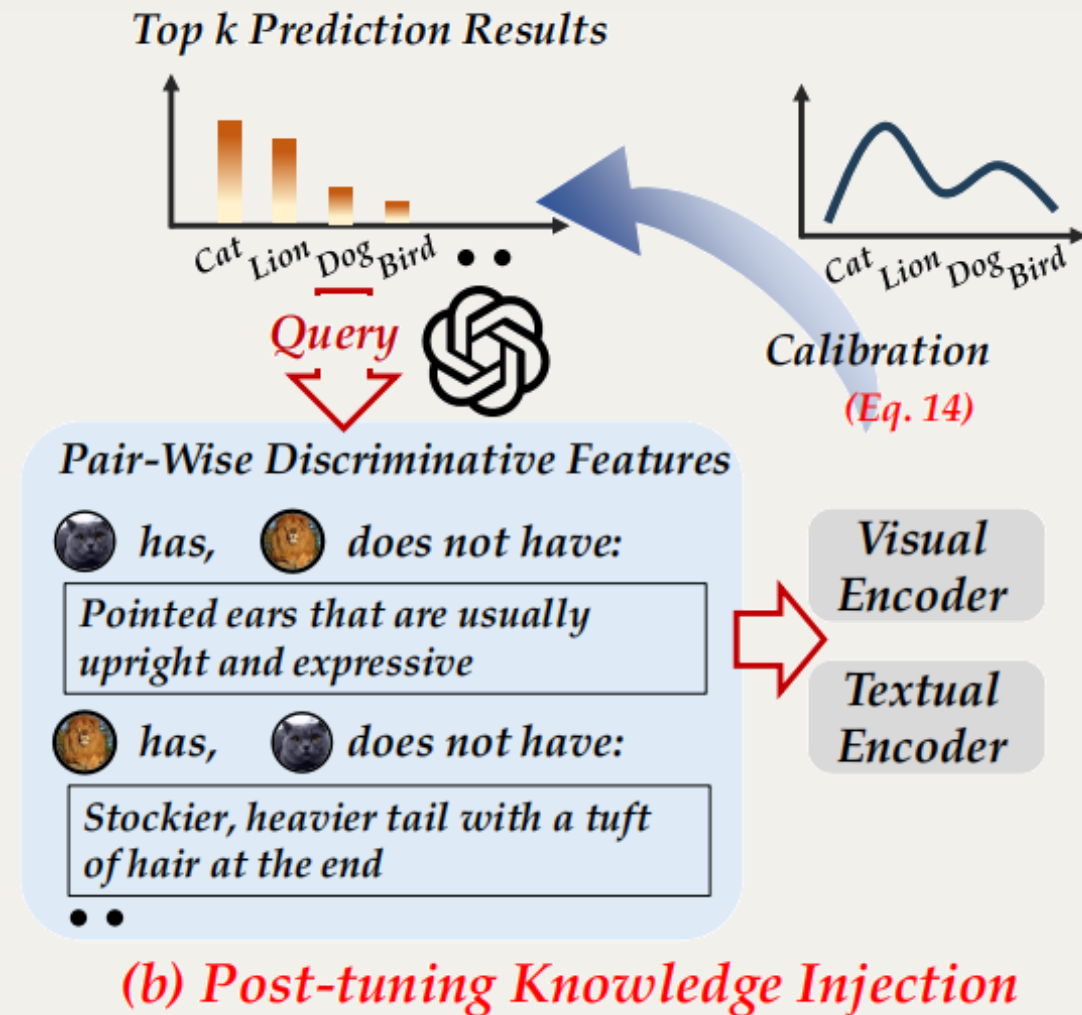
## Method:



among this label set. We again utilize GPT-4 to generate *pair-wise* discriminative features:

**Q:** What are unique visual features of  $[\text{CLASS}]_i$  compared to  $[\text{CLASS}]_j$  in a photo? Focus on their key visual differences.

**A:**  $[\text{CLASS}]_i$ : **1.** Long, thin tail that aids in balance  $\dots$   $[\text{CLASS}]_j$ : **1.** Stockier, heavier tail with a tuft of hair at the end  $\dots$



# External Knowledge Injection for CLIP-Based Class-Incremental Learning

**Experiment:** **Dataset split:** Following [52, 66], we use ‘B- $m$  Inc- $n$ ’ to split the classes in CIL.  $m$  indicates the number of classes in the first stage, and  $n$  represents that of every following stage. We follow [52] to randomly shuffle the class order with random seed 1993 for all compared methods, and keep this same for every method.

Table 1. Average and last performance comparison of different methods. The best performance is shown in bold. **All methods are initialized with the same pre-trained CLIP without exemplars for a fair comparison.**

Method	Aircraft				CIFAR100				Cars			
	B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	3.16	0.96	1.72	1.05	7.84	4.44	5.30	2.46	3.14	1.10	1.54	1.13
CoOp [89]	14.54	7.14	13.05	7.77	47.00	24.24	41.23	24.12	36.46	21.65	37.40	20.87
SimpleCIL [85]	59.24	48.09	53.05	48.09	84.15	76.63	80.20	76.63	92.04	86.85	88.96	86.85
ZS-CLIP [50]	26.66	17.22	21.70	17.22	81.81	71.38	76.49	71.38	82.60	76.37	78.32	76.37
L2P [66]	47.19	28.29	44.07	32.13	82.74	73.03	81.14	73.61	76.63	61.82	76.37	65.64
DualPrompt [65]	44.30	25.83	46.07	33.57	81.63	72.44	80.12	72.57	76.26	62.94	76.88	67.55
CODA-Prompt [54]	45.98	27.69	45.14	32.28	82.43	73.43	78.69	71.58	80.21	66.47	75.06	64.19
RAPF [26]	50.38	23.61	40.47	25.44	86.14	78.04	82.17	77.93	82.89	62.85	75.87	63.19
ENGINE	<b>69.69</b>	<b>58.69</b>	<b>64.38</b>	<b>59.02</b>	<b>86.92</b>	<b>79.22</b>	<b>83.15</b>	<b>79.47</b>	<b>94.14</b>	<b>90.08</b>	<b>91.61</b>	<b>90.03</b>



# External Knowledge Injection for CLIP-Based Class-Incremental Learning

## Experiment:

Method	ImageNet-R				CUB				UCF			
	B0 Inc20		B100 Inc20		B0 Inc20		B100 Inc20		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	1.37	0.43	1.01	0.88	2.06	0.64	0.56	0.47	4.51	1.59	1.21	0.80
CoOp [89]	60.73	37.52	54.20	39.77	27.61	8.57	24.03	10.14	47.85	33.46	42.02	24.74
SimpleCIL [85]	81.06	74.48	76.84	74.48	83.81	77.52	79.75	77.52	90.44	85.68	88.12	85.68
ZS-CLIP [50]	83.37	77.17	79.57	77.17	74.38	63.06	67.96	63.06	75.50	67.64	71.44	67.64
L2P [66]	75.97	66.52	72.82	66.77	70.87	57.93	75.64	66.12	86.34	76.43	83.95	76.62
DualPrompt [65]	76.21	66.65	73.22	67.58	69.89	57.46	74.40	64.84	85.21	75.82	84.31	76.35
CODA-Prompt [54]	77.69	68.95	73.71	68.05	73.12	62.98	73.95	62.21	87.76	80.14	83.04	75.03
RAPF [26]	81.26	70.48	76.10	70.23	79.09	62.77	72.82	62.93	92.28	80.33	90.31	81.55
ENGINE	<b>86.22</b>	<b>80.37</b>	<b>83.63</b>	<b>80.98</b>	<b>86.65</b>	<b>80.20</b>	<b>82.59</b>	<b>79.30</b>	<b>94.35</b>	<b>90.03</b>	<b>92.51</b>	<b>89.58</b>

Method	SUN				Food				ObjectNet			
	B0 Inc30		B150 Inc30		B0 Inc10		B50 Inc10		B0 Inc20		B100 Inc20	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	4.51	1.59	0.78	0.72	3.49	1.71	2.14	1.52	1.34	0.47	0.69	0.54
CoOp [89]	45.93	23.11	39.33	24.89	36.01	14.18	33.13	18.67	21.24	6.29	16.21	6.82
SimpleCIL [85]	82.13	75.58	78.62	75.58	87.89	81.65	84.73	81.65	52.06	40.13	45.11	40.13
ZS-CLIP [50]	79.42	72.11	74.95	72.11	87.86	81.92	84.75	81.92	38.43	26.43	31.12	26.43
L2P [66]	82.82	74.54	79.57	73.10	85.66	77.33	80.42	73.13	51.40	39.39	48.91	42.83
DualPrompt [65]	82.46	74.40	79.37	73.02	84.92	77.29	80.00	72.75	52.62	40.72	49.08	42.92
CODA-Prompt [54]	83.34	75.71	80.38	74.17	86.18	78.78	80.98	74.13	46.49	34.13	40.57	34.13
RAPF [26]	82.13	72.47	78.04	73.10	88.57	81.15	85.53	81.17	48.67	27.43	39.28	28.73
ENGINE	<b>85.04</b>	<b>78.54</b>	<b>81.57</b>	<b>78.45</b>	<b>89.81</b>	<b>83.89</b>	<b>86.89</b>	<b>83.94</b>	<b>59.11</b>	<b>45.19</b>	<b>51.32</b>	<b>44.99</b>

# External Knowledge Injection for CLIP-Based Class-Incremental Learning

Visualization:

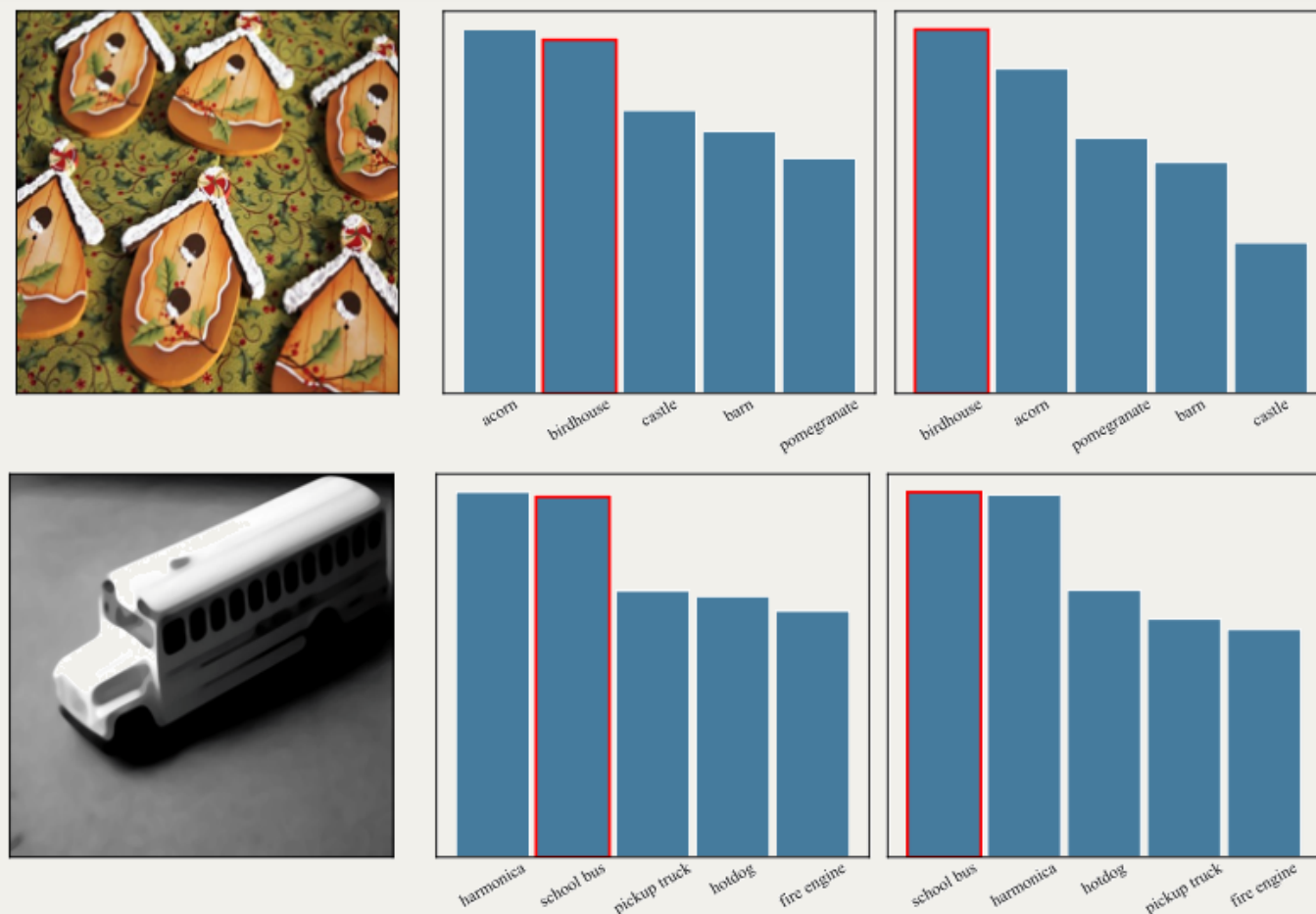


Figure 6. **Left:** Input. **Middle:** Top-5 predictions before post-tuning knowledge injection. **Right:** Top-5 predictions after post-tuning. More cases are shown in the supplementary.

# Integrating Task-Specific and Universal Adapters for Pre-Trained Model-based Class-Incremental Learning

Yan Wang, Da-Wei Zhou<sup>(✉)</sup>, Han-Jia Ye

School of Artificial Intelligence, Nanjing University

National Key Laboratory for Novel Software Technology, Nanjing University

`{wangy, zhoudw, yehj}@lamda.nju.edu.cn`

Task & Setting: Class Incremental Learning, Pre-Trained Model

Motivation: Concentrate on the acquisition of task-specific knowledge and ignore the general knowledge shared between different tasks

## Method:

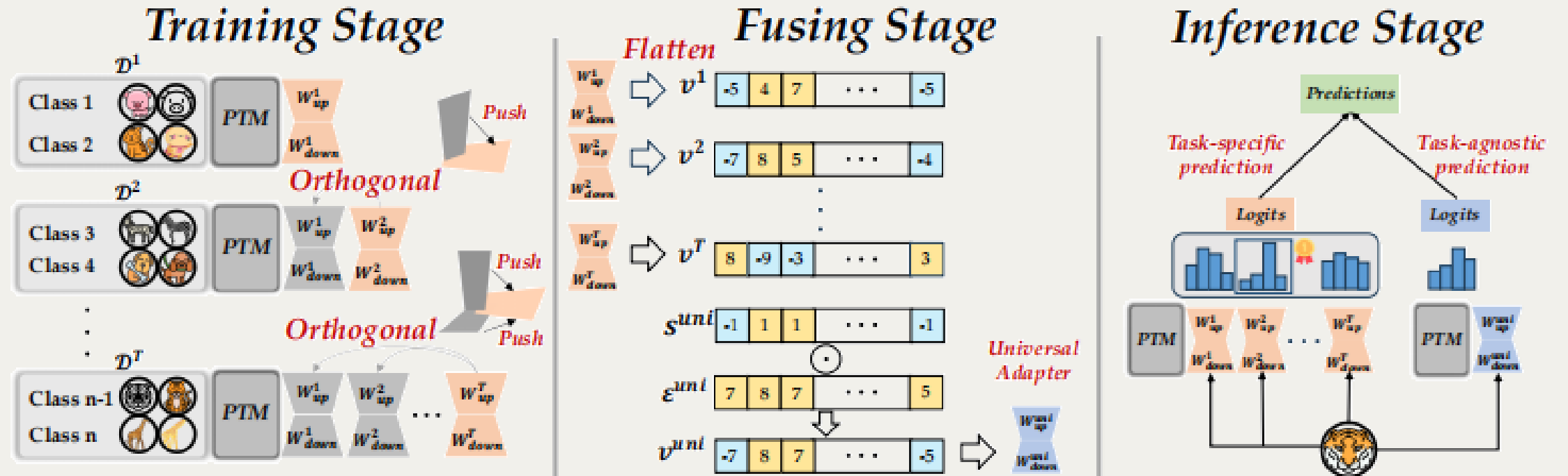
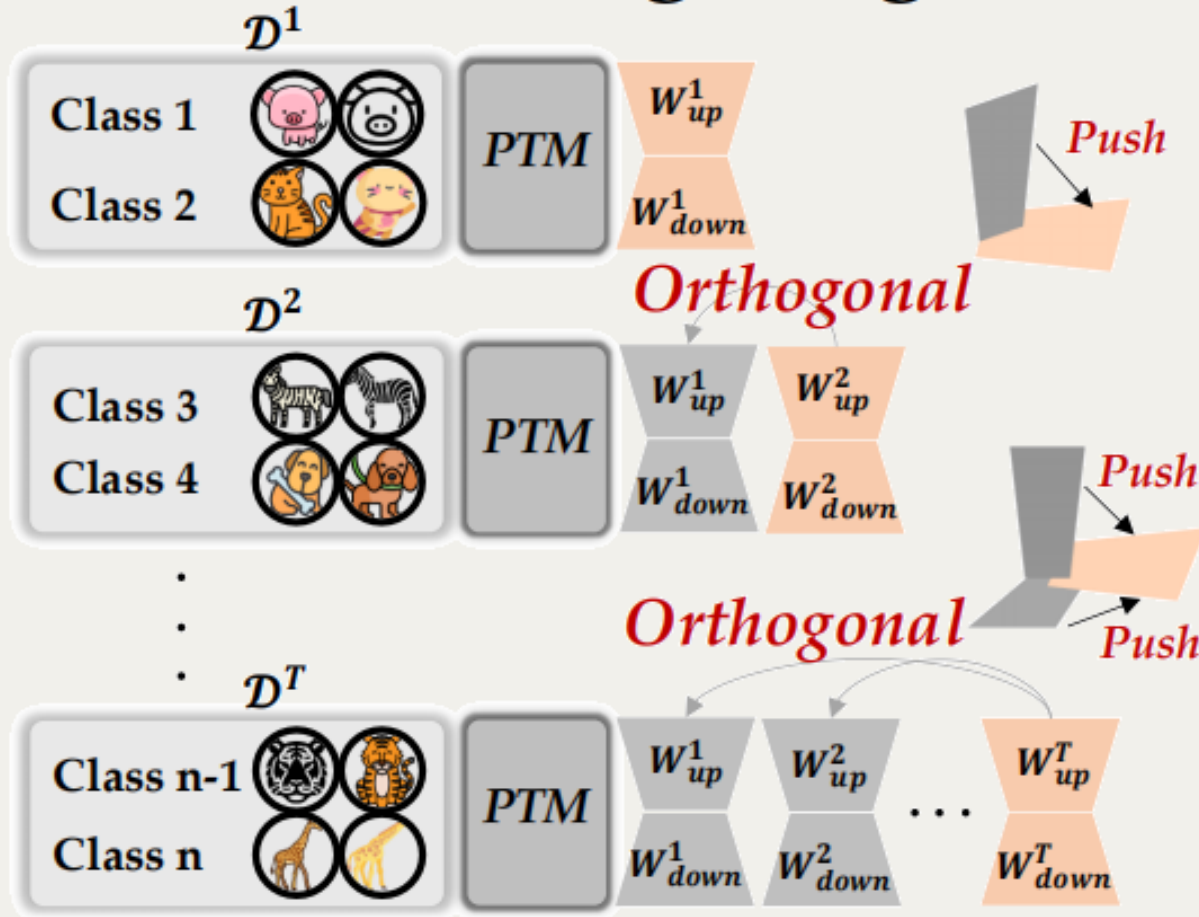


Figure 1. Illustration of TUNA. **Left:** The training protocol of TUNA. We use orthogonal loss to train task-specific adapters. **Middle:** The fusing process. We construct an aggregated sign vector and a magnitude vector, which are combined to form the universal task vector. **Right:** During the inference phase, we select the most appropriate task-specific adapter based on entropy, and then combine the outputs from both the task-specific and universal adapters.

## Method:

### Training Stage



via residual connections. An adapter comprises a down-projection layer  $W_{down} \in \mathbb{R}^{d \times r}$ , a non-linear activation function ReLU and an up-projection layer  $W_{up} \in \mathbb{R}^{r \times d}$ . The output formula of the MLP layer is formulated as follows:

$$\mathbf{x}_o = \text{MLP}(\mathbf{x}_i) + \text{ReLU}(\mathbf{x}_i W_{down}) W_{up}, \quad (4)$$

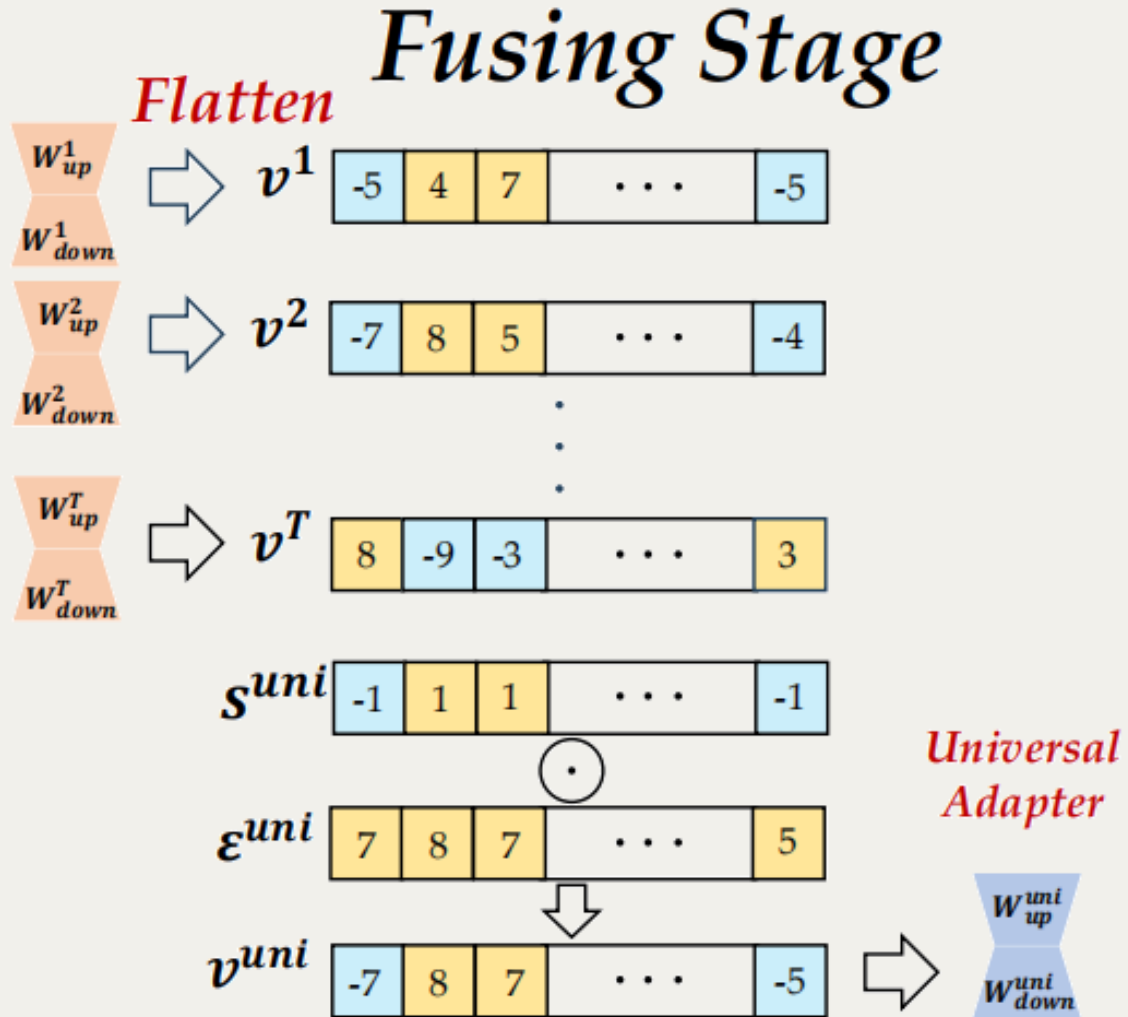
where  $\mathbf{x}_i$  and  $\mathbf{x}_o$  are the input and output of the MLP, re-

$$\mathcal{L}_{orth} = \sum_{i=1}^{t-1} \left\| W_{up}^t \cdot W_{up}^i{}^\top \right\|_1, \quad (6)$$

where  $\|\cdot\|_1$  represents the  $L_1$  norm. The up-projection



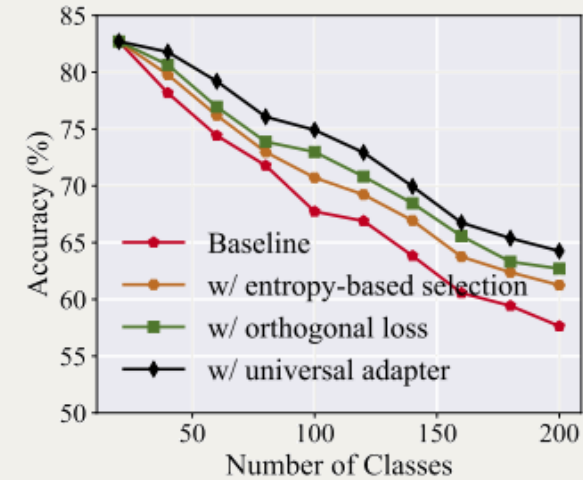
Method:



across all task-specific vectors. This is done by taking the sign of the sum of the corresponding parameters:

$$\mathbf{s}^{\text{uni}} = \text{sgn} \left( \sum_{i=1}^t \mathbf{v}^i \right), \quad (8)$$

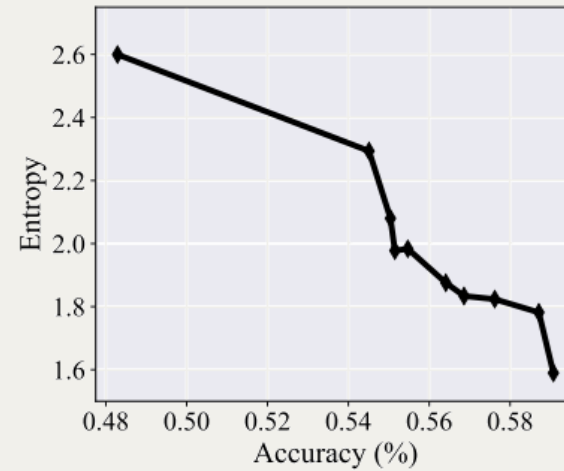
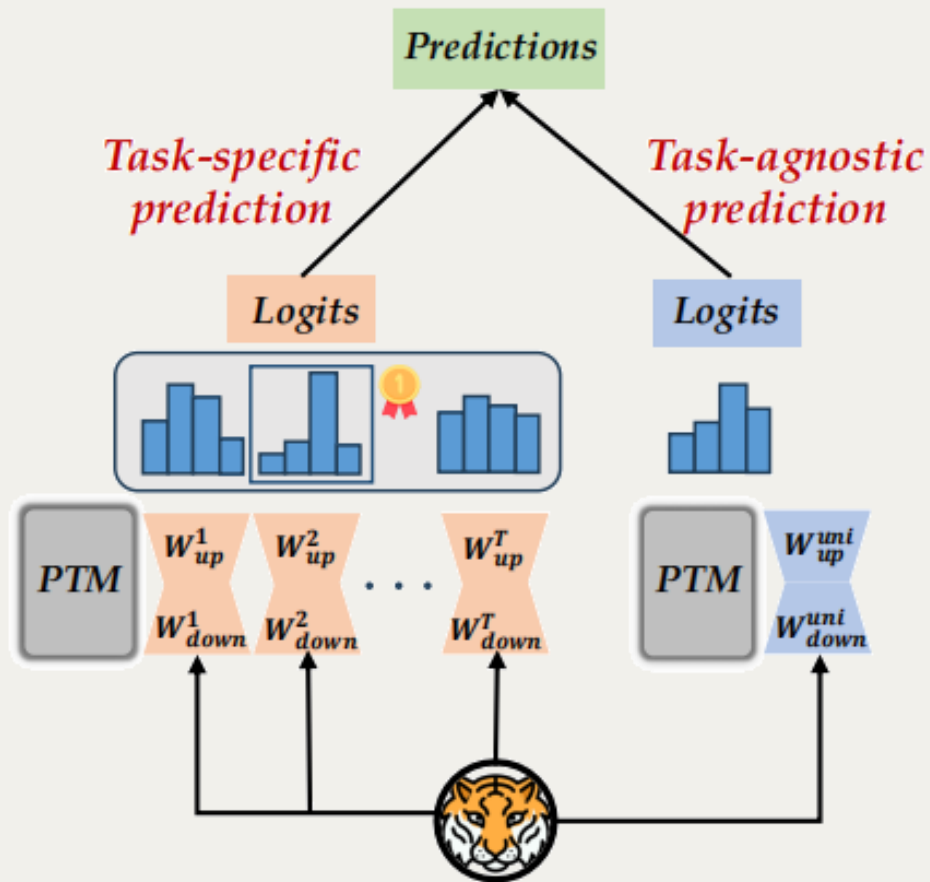
where  $\text{sgn}(\cdot)$  denotes the sign function. For each parame-



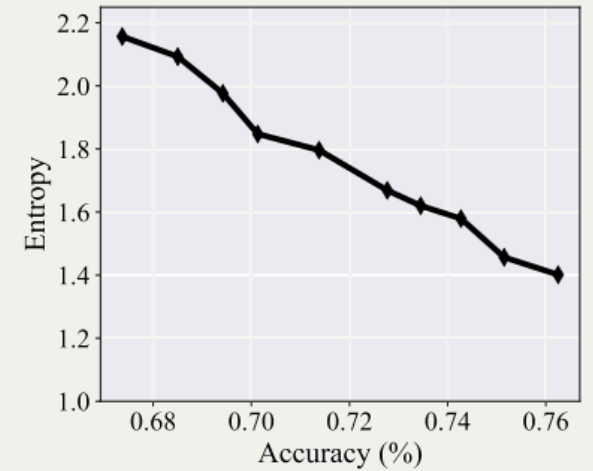
(a) Ablation study

Method:

## Inference Stage



(a) ImageNet-A B0 inc20



(b) ImageNet-R B0 inc20

Figure 2. Relationship between accuracy and entropy.

熵: 
$$H(X) = -\sum_{i=1}^n p(x_i) \log p(x_i)$$

$$y^* = \arg \max_y (f_y(\mathbf{x}; \mathcal{A}^*) + f_y(\mathbf{x}; \mathcal{A}_{\text{uni}}))$$

## Experiment: ViT-B/16-IN1K and ViT-B/16-IN21K

Table 1. Average and last performance comparison on four datasets with **ViT-B/16-IN21K** as the backbone. We report all compared methods with their source code. The best performance is highlighted in bold. None of the methods utilize exemplars in their implementation.

Method	CIFAR B0 Inc5		ImageNet-R B0 Inc20		ImageNet-A B0 Inc20		ObjectNet B0 Inc20	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
L2P [50]	85.94	79.93	75.46	69.77	49.39	41.71	63.78	52.19
DualPrompt [49]	87.87	81.15	73.10	67.18	53.71	41.67	59.27	49.33
CODA-Prompt [43]	89.11	81.96	77.97	72.27	53.54	42.73	66.07	53.29
SLCA [58]	92.49	88.55	81.17	77.00	68.66	58.74	72.55	61.30
SSIAT [45]	93.52	90.07	83.20	78.85	70.83	62.23	73.65	62.45
MOS [44]	93.30	89.25	82.96	77.93	67.08	56.22	74.69	63.62
SimpleCIL [64]	87.57	81.26	61.26	54.55	59.77	48.91	65.45	53.59
APER + Adapter [64]	90.65	85.15	75.82	67.95	60.47	49.37	67.18	55.24
RanPAC [37]	94.00	90.62	82.98	77.94	69.32	61.82	72.76	62.02
EASE [65]	91.51	85.80	81.74	76.17	65.34	55.04	70.84	57.86
TUNA (Ours)	<b>94.44</b>	<b>90.74</b>	<b>84.22</b>	<b>79.42</b>	<b>73.78</b>	<b>64.78</b>	<b>76.46</b>	<b>66.32</b>

## Experiment: ViT-B/16-IN1K and ViT-B/16-IN21K

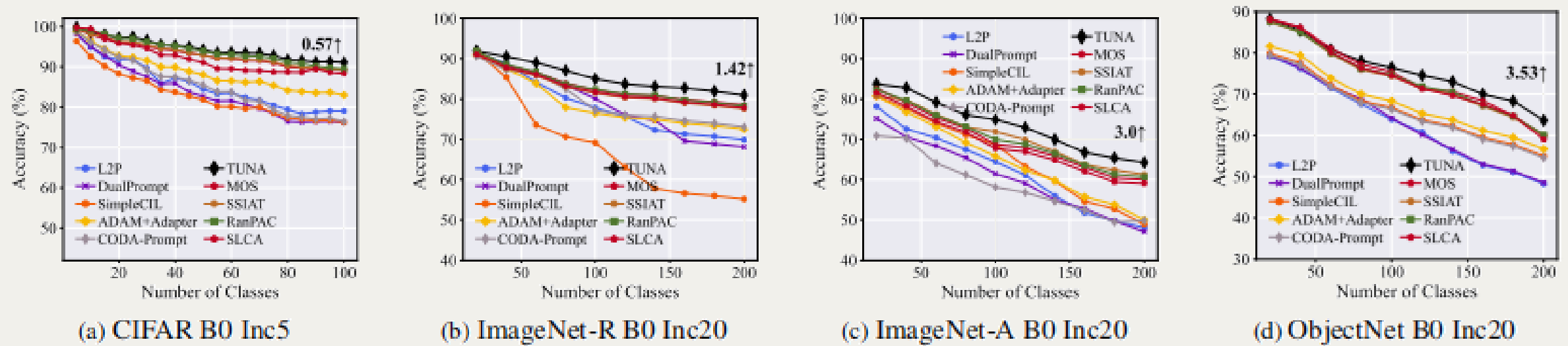


Figure 3. Performance curve of different methods under different settings. All methods are initialized with **ViT-B/16-IN1K**. The relative improvement over the second-best method is annotated with numerical values above the curves at the final incremental stage.

## Visualization:

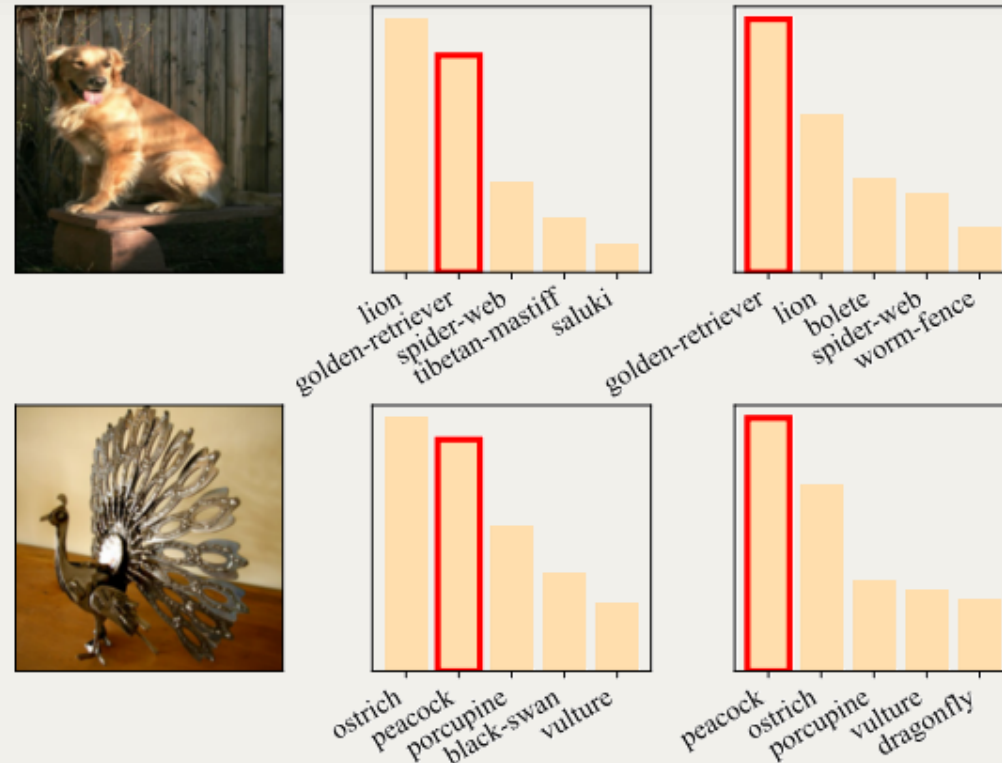


Figure 7. Visualizations of the predictions on ImageNet-R. The original images are depicted in the first column, followed by the top-5 prediction probability produced by task-specific adapter, and the probabilities generated by the universal adapter in the last column. The ground-truth class is highlighted with red boxes.

# Incremental Learning



马鸣霄

2025.09.12