[12/13截止] 期末教學意見調查 - 填問卷、抽iPad
https://if163.aca.ntu.edu.tw/eportfolio/ or
https://investea.aca.ntu.edu.tw/opinion/login.asp

# Deep Learning for Computer Vision

113-1/Fall 2024

https://cool.ntu.edu.tw/courses/41702 (NTU COOL)

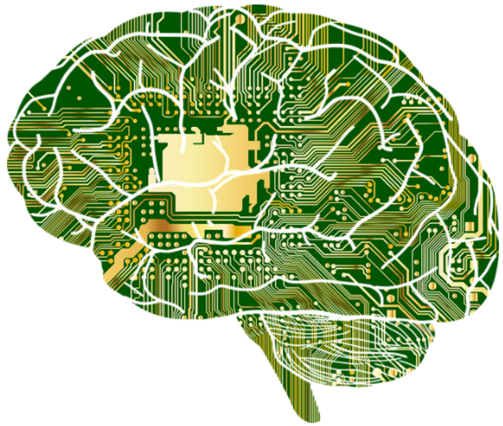
http://vllab.ee.ntu.edu.tw/dlcv.html (Public website)

Yu-Chiang Frank Wang 王鈺強, Professor

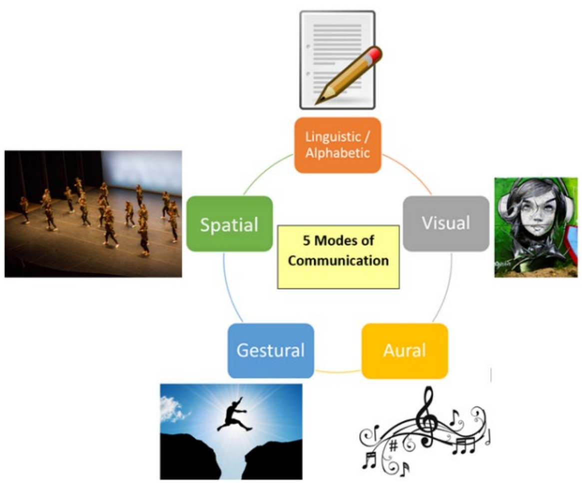
Dept. Electrical Engineering, National Taiwan University

2024/12/10

# What to Be Covered Today…

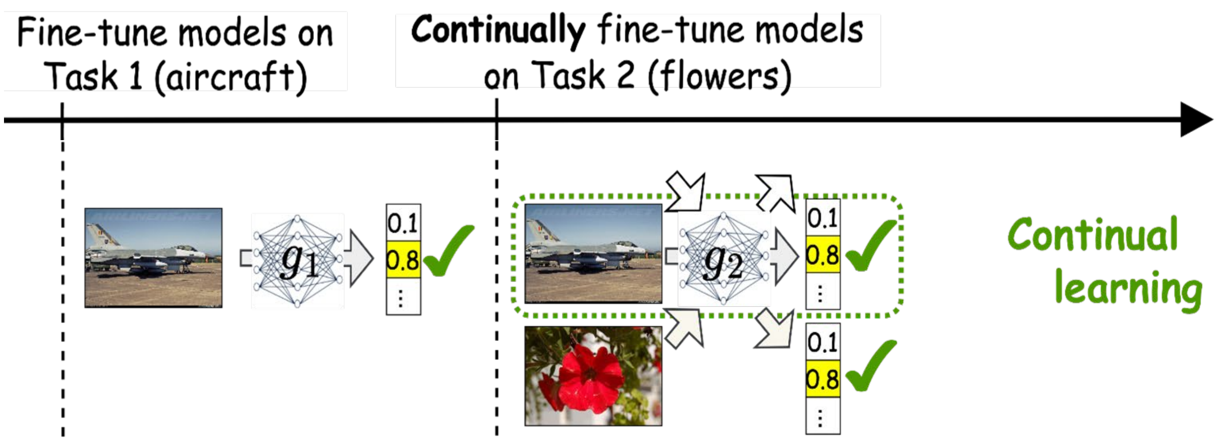


- **Additional Topics in DLCV**
  - Continual Learning
  - Meta Learning
  - Domain Generalization
  - Federated Learning
- **Experience Sharing**
  - Tim Chou (MS, GICE, NTU 2023), AI SW Engineer, NVIDIA




# Continual Learning (aka Incremental Learning)

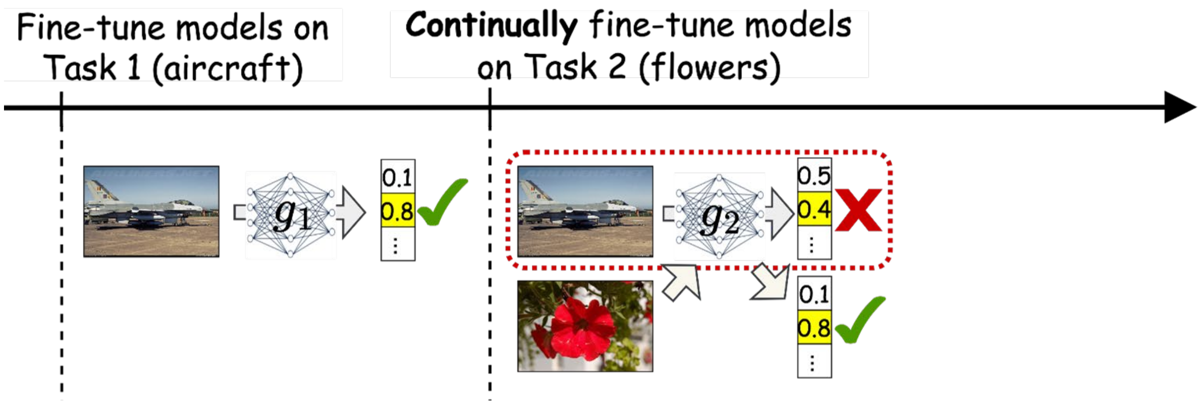
- **Motivation**
  - Always new dataset, knowledge, etc, to finetune the LLM/VLM
    - No practical to re-train foundation models from scratch
  - It is a naive learning way, since **human is a continual learner**.
  - Goal: learn downstream tasks/datasets in a sequential (or incremental) way, while not forgetting what models have learned before.



# Continual Learning (cont'd)

- **Task Definition**
  - Learning a list of datasets in a sequential manner **without forgetting** previous knowledge.

- **The most straight forward strategy**
  - Directly fine-tune a pre-trained model on a new dataset…any concern?
  - **Challenge:** Suffer from the well-known **catastrophic forgetting issue**,
    as the model weights can be totally distorted toward the new task only



# Previous works on Continual Learning

- **Rehearsal-based methods**
  - iCaRL (CVPR’17)
- **Regularization-based methods**
  - EWC (PNAS’17)
- **Continual Learning for open-vocab. Vision-Language Models**
  - ZSCL (ICCV’23)
  - Select and Distill (ECCV’24)

# iCaRL: Incremental Classifier and Representation Learning, Oxford, CVPR'17

- **Rehearsal-based method**
- **Idea:**
  - Maintain a subset of previous data in a class exemplar sets $P = (P_1, \cdots, P_{s-1})$ where {1, 2, …, k-1} are the learned classes
  - Joint training with the current data $X^s, \ldots, X^t$ with classes {s, …, t}
- **Method**
  - For data in *P*, enforce the learned model θ output as that of $\theta_{old}$.
    - Can be viewed as **Knowledge Distillation**
  - For newly observed data, training with the standard cross entropy loss.

$$Y_{\text{old}} = \{f_{\theta_{\text{old}}}(x) | \forall x \in P\}$$

$$\mathcal{L}(\theta) = \sum_{(x,y)\in D} \left[ \sum_{y=s}^{t} \mathcal{L}(Y_{\text{new}}, \hat{Y}) + \sum_{y=1}^{s-1} \mathcal{L}(Y_{\text{old}}, \hat{Y}) \right]$$

  - Any concern?



iCaRL: Incremental Classifier and Representation Learning

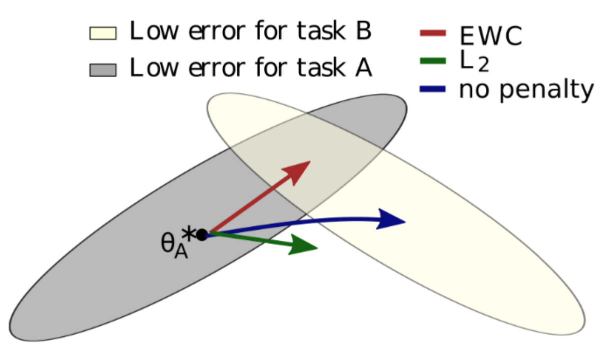
# EWC: Overcoming catastrophic forgetting in neural networks, DeepMind, PNAS'17

- **Regularization-based method**
- **Idea:**
  - Weight Consolidation:
    restrict the learned weights not to be too distinct from the original model ones

$$\mathcal{L}_{\mathrm{WC}} = \sum_i (\theta_i - \bar{\theta}_i)^2$$

  - Elastic Weight Consolidation:
    each parameter should be treated differently (w/ different weights)
    - i: the index of the model parameters.

$$\mathcal{L}_{\mathrm{EWC}} = \sum w_i \cdot (\theta_i - \bar{\theta}_i)^2$$




Overcoming catastrophic forgetting in neural networks

# EWC, DeepMind, PNAS’17 (cont’d)

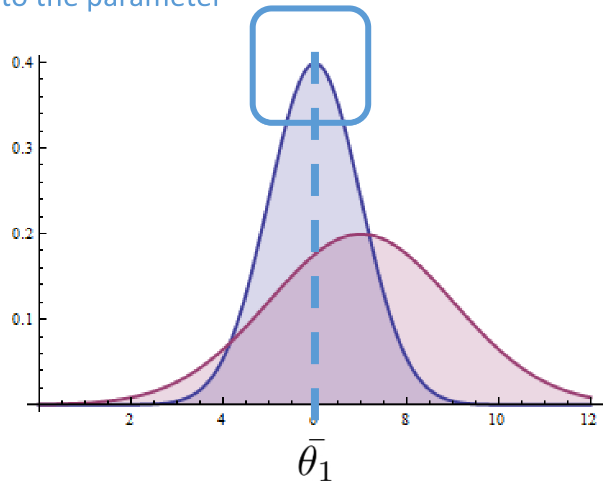
- **Method (cont’d)**
  - Using Fisher Information (F) to determine
    the importance of a parameter to the previous task.
    - Fisher information: the expectation of second derivative of negative log-likelihood at $\bar{\theta}$

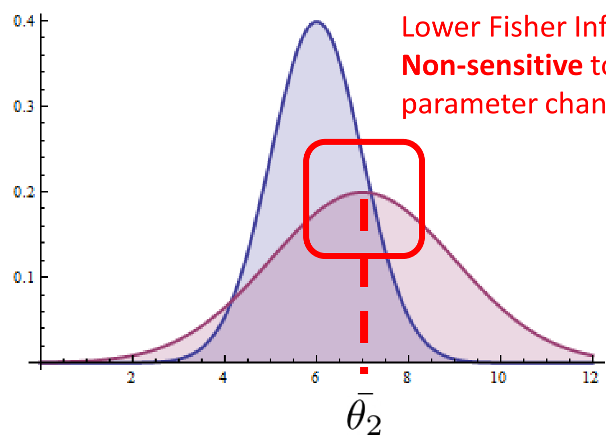
$$\mathcal{L}_{\mathrm{EWC}} = \sum_i \frac{\lambda}{2} F_i (\theta_i - \bar{\theta}_i)^2$$

    - $\lambda$: a hyper-parameter to determine the overall importance of previous tasks.

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2$$

Higher Fisher Information. **Sensitive** to the parameter change



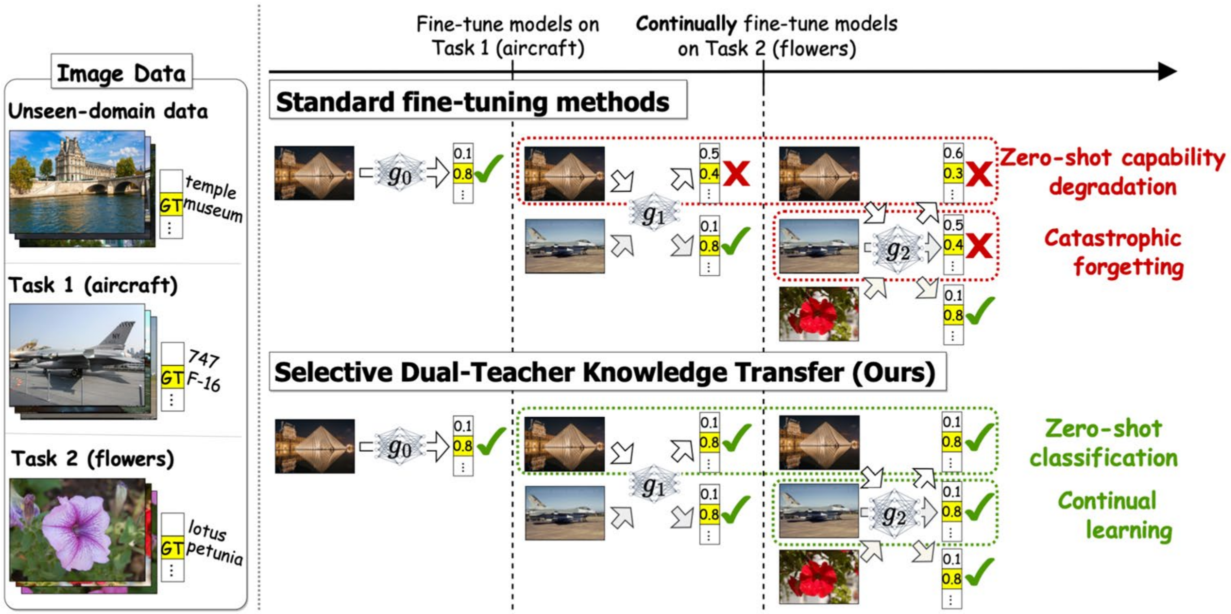




Lower Fisher Information. **Non-sensitive** to the parameter change.

[Overcoming catastrophic forgetting in neural networks]

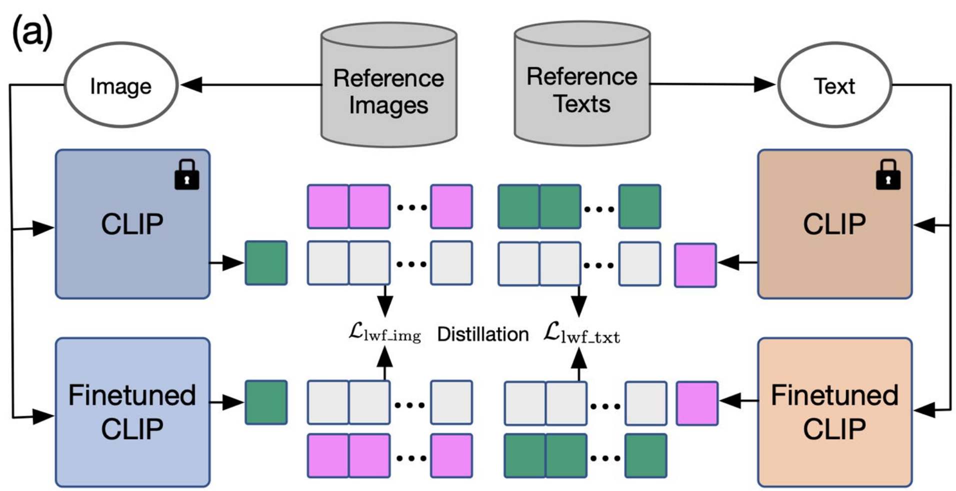
# Continual Learning for Vision-Language Models

- **Motivation**
  - With the prevalence of large-scale Vision-Language Models (VLMs), Continual Learning for VLMs has emerged as a potential research trends.
- **Goal**
  - Sequentially learning from new datasets
  - Preserve the original zero-shot ability for unseen data
  - Maintain knowledge learned from previous stages (as existing CL methods do)



# ZSCL: Preventing Zero-Shot transfer degradation in Continual Learning of vision-language models, NUS, ICCV’23

- **Method**
  - Utilize an auxiliary reference dataset (e.g., ImageNet), and perform **Knowledge Distillation** from **the original CLIP model.**
    - (1) Distill knowledge on **both visual and textual sides**.




Teacher (Original CLIP) output

$$\mathcal{L}_{\text{dist\_img}} = \mathrm{CE}(\boldsymbol{p}, \overline{\boldsymbol{p}}) = -\sum_{j=1}^{m} \boxed{\boldsymbol{p}_j} \cdot \log \boxed{\overline{\boldsymbol{p}}_j}$$

Student output

Preventing Zero-Shot transfer degradation in Continual Learning of vision-language models

# ZSCL, ICCV'23 (cont'd)

- **Method (cont'd)**
  - (2) WE: Weight space Ensemble to regularize the weights
    - The updated model weights would not be too different from the weights learned from the previous stage.

$$\hat{\theta}_t = \begin{cases} \theta_0 & t = 0 \\ \frac{1}{t+1}\theta_t + \frac{t}{t+1} \cdot \hat{\theta}_{t-1} & \text{every I iterations} \end{cases}.$$

    - Same form as EMA (exponetial moving average)

  - Training strategy: (1) -> (2) -> (1) -> (2) -> …

(1) $$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda \cdot (\mathcal{L}_{\text{lwf\_img}} + \mathcal{L}_{\text{lwf\_txt}})$$

Data from novel task + auxiliary ref dataset

(2) $$\hat{\theta}_t = \begin{cases} \theta_0 & t = 0 \\ \frac{1}{t+1}\theta_t + \frac{t}{t+1} \cdot \hat{\theta}_{t-1} & \text{every I iterations} \end{cases}.$$



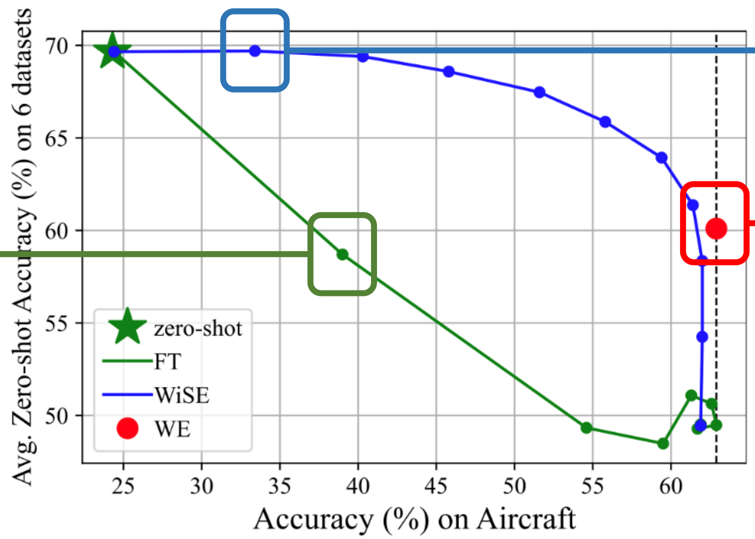
Preventing Zero-Shot transfer degradation in Continual Learning of vision-language models

# ZSCL, ICCV’23 (cont’d)

- **Comparisons**
  - Zero-shot accuracy vs. accuracy on novel task

Ensemble the original and fine-tuned model weights with different ratios. **Sensitive to the choose of the ratio**

Sample every 100 iterations. As training progresses, the model's zero-shot capability deteriorates.



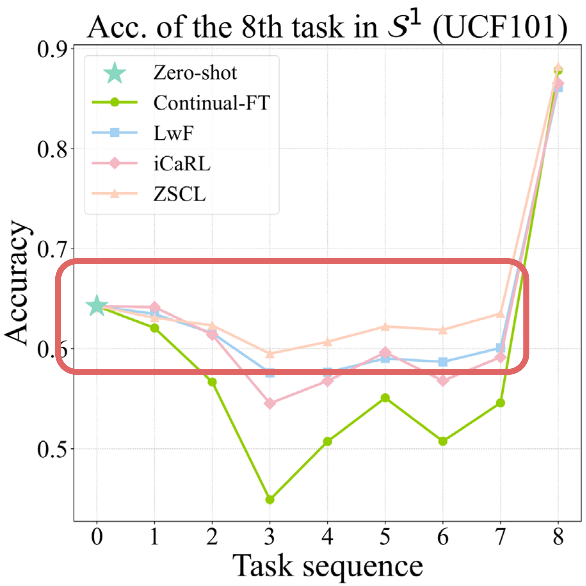

Iterative weight ensemble. Improved fine-tuned accuracy, with an acceptable decrease in zero-shot performance



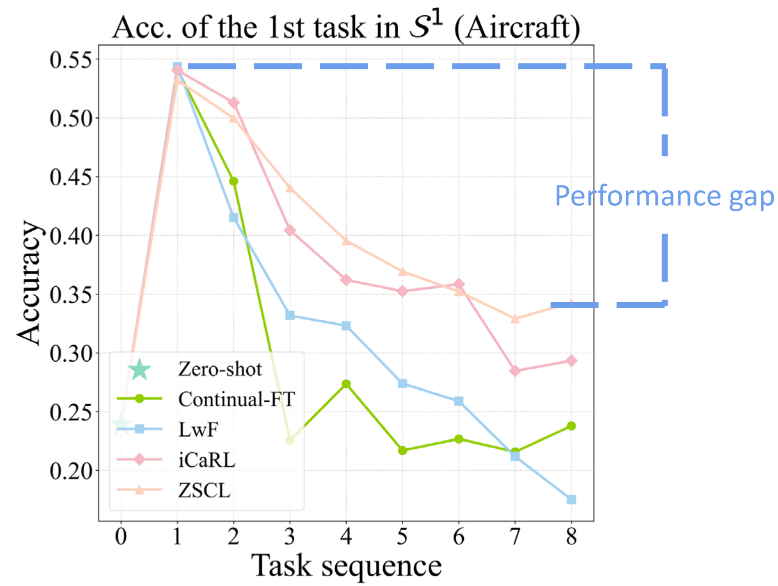
Preventing Zero-Shot transfer degradation in Continual Learning of vision-language models

# ZSCL, ICCV'23 (cont'd)

- **Limitation**
  - ZSCL still significantly suffers from catastrophic forgetting for previous tasks.




ZSCL can preserve zero-shot ability for unseen data



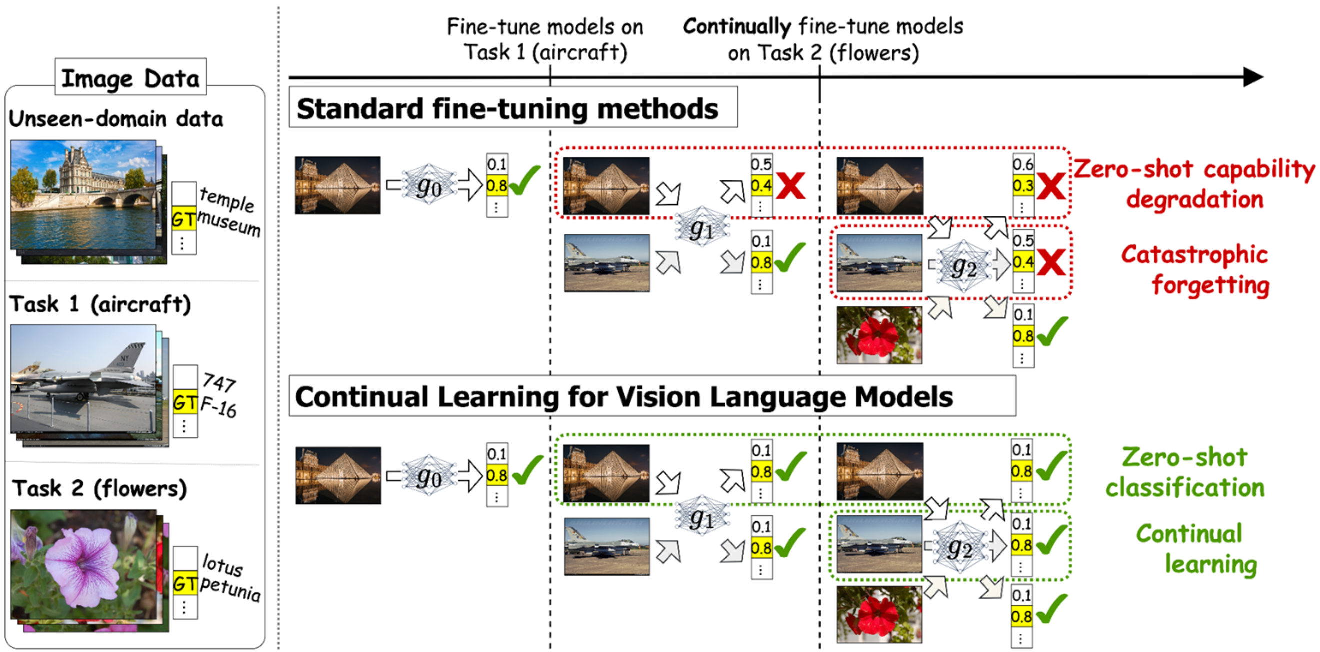

There is still a gap for previous task after training on multiple datasets



Preventing Zero-Shot transfer degradation in Continual Learning of vision-language models

# Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models, NTU, ECCV'24

- **Goal**
  - Same as ZSCK, adapt to new datasets sequentially while:
    - preserving the original pre-trained zero-shot ability
    - maintaining the knowledge learned from previous stages

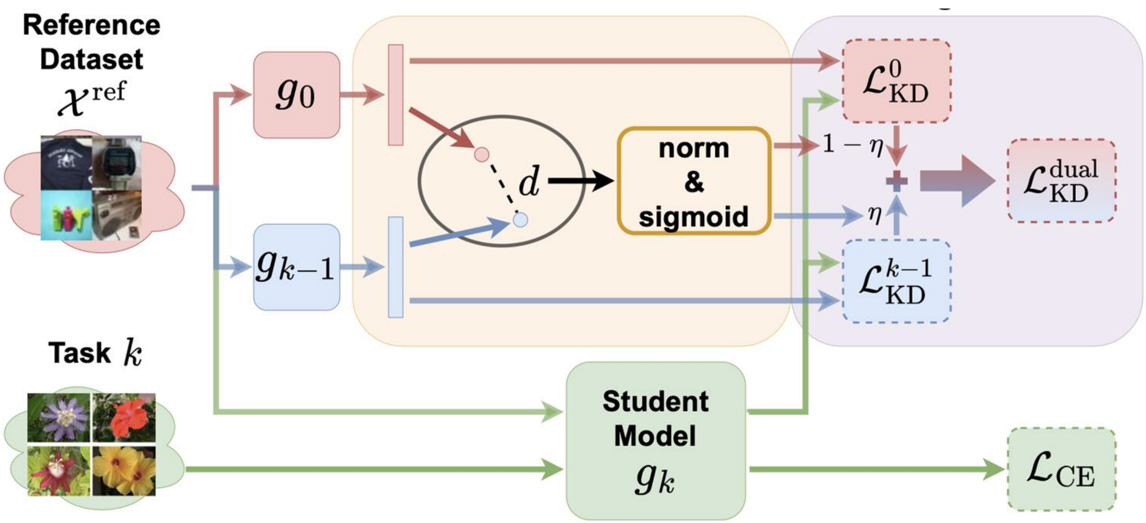





Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Idea**
  - Follow ZSCL, utilize a reference dataset for knowledge distillation
  - Dual-Teacher Knowledge Distillation (original VLM vs. recently tuned VLM)
    - Distill from $\boxed{\phantom{xxxxxxxxxxxx}}$ to preserve **zero-shot ability**.
    - Distill from $\boxed{\phantom{xxxxxxxxxxxxxx}}$ to preserve **prior knowledge**.
- **Key**
  - For any data point in the reference dataset,
    we need to **select a proper model** and distill its knowledge.





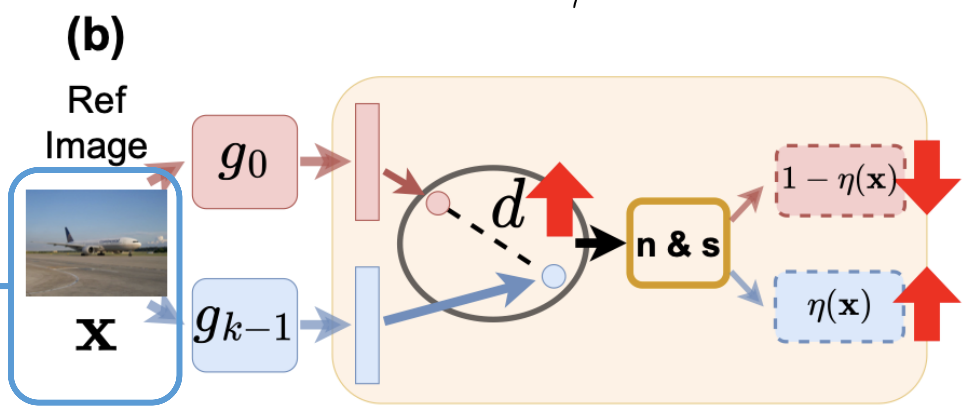

Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Observation**
  - If a data point is a previously learned data.
    - It must be seen by $g_{k-1}$, but never been seen by $g_0$
      thus, the **feature distance d** between $g_0$ and $g_{k-1}$ would be large or small?
    - Select $g_{k-1}$ as the teacher model to maintain previous knowledge
  - $\eta(\mathbf{x})$ : A normalized distance between 0~1, determine how much should we distill from $g_{k-1}$

$$\eta(\mathbf{x}) = \sigma(\frac{d(g_{k-1}(\mathbf{x}), g_0(\mathbf{x})) - \delta}{\gamma}),$$

Previously learned data



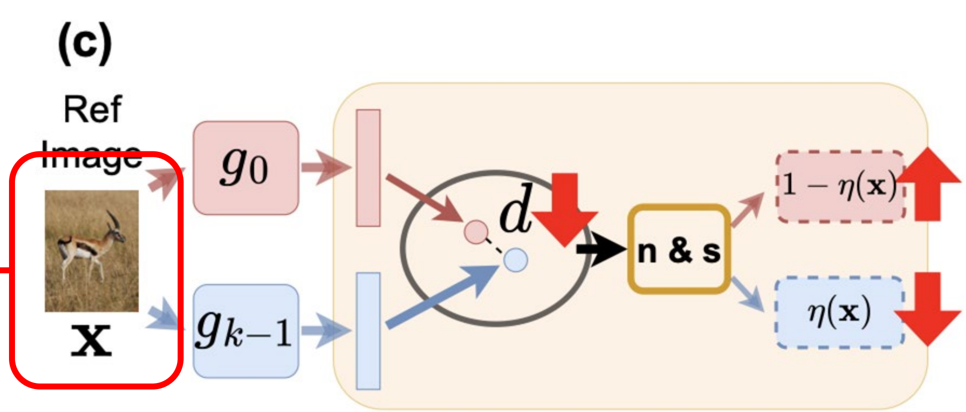



Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Observation**
  - If a data point has never been seen by both $g_{k-1}$ and $g_0$ (i.e., unseen data)
    - The **feature distance d** between $g_0$ and $g_{k-1}$ can be relatively **small/large?**
    - In this case, we should select $g_0$ as the teacher model to preserve the original zero-shot ability.

Unseen data





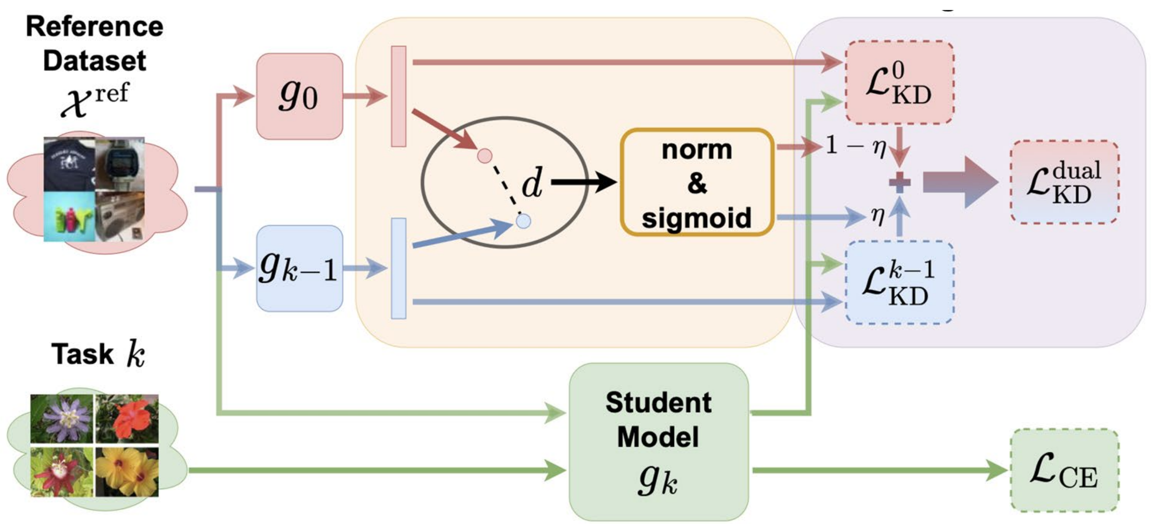

Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Objective**

$$\mathcal{L}_{\mathrm{KD}}^{k-1} = d(g_{k-1}(\mathbf{x}), g_k(\mathbf{x})), \; \mathcal{L}_{\mathrm{KD}}^{0} = d(g_0(\mathbf{x}), g_k(\mathbf{x}))$$

$$\mathcal{L}_{\mathrm{KD}}^{\mathrm{dual}} = \sum_{\mathbf{x} \sim \mathcal{X}^{\mathrm{ref}}} \eta(\mathbf{x}) \cdot \mathcal{L}_{\mathrm{KD}}^{k-1} + (1 - \eta(\mathbf{x})) \cdot \mathcal{L}_{\mathrm{KD}}^{0}$$

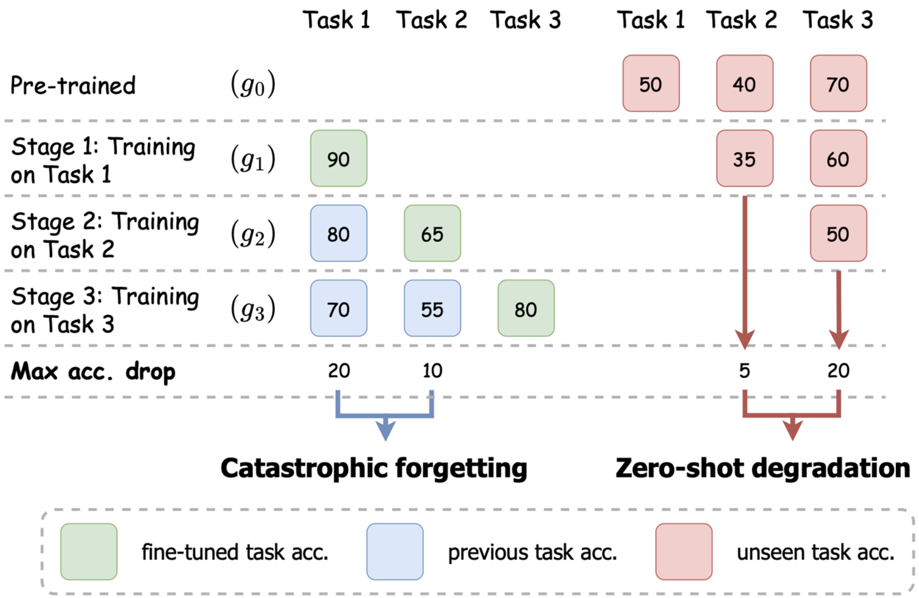





Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Metrics**
  - Average Accuracy
    - Average of the last performance on each dataset
  - Catastrophic forgetting
    - Max. performance gap after the task has been fine-tuned
  - Zero-shot degradation
    - Max. performance gap before the task has been fine-tuned

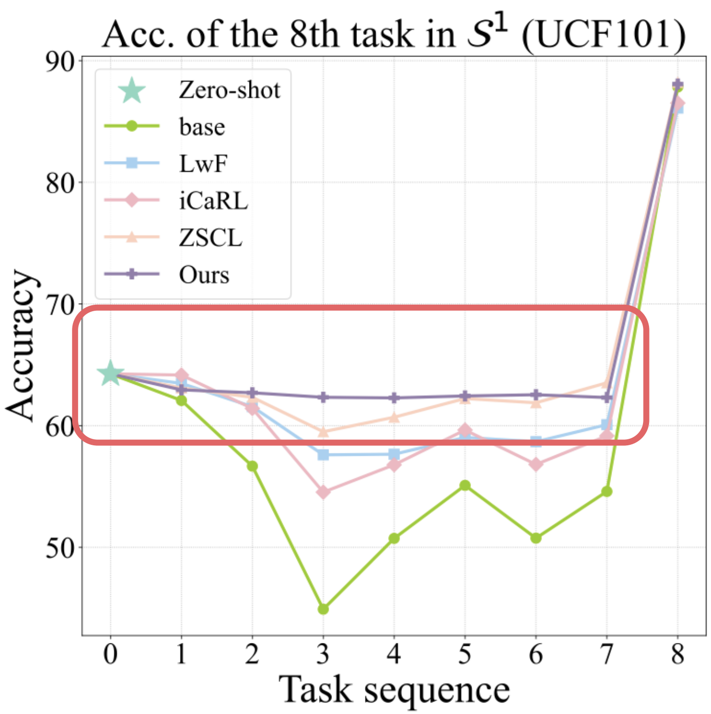




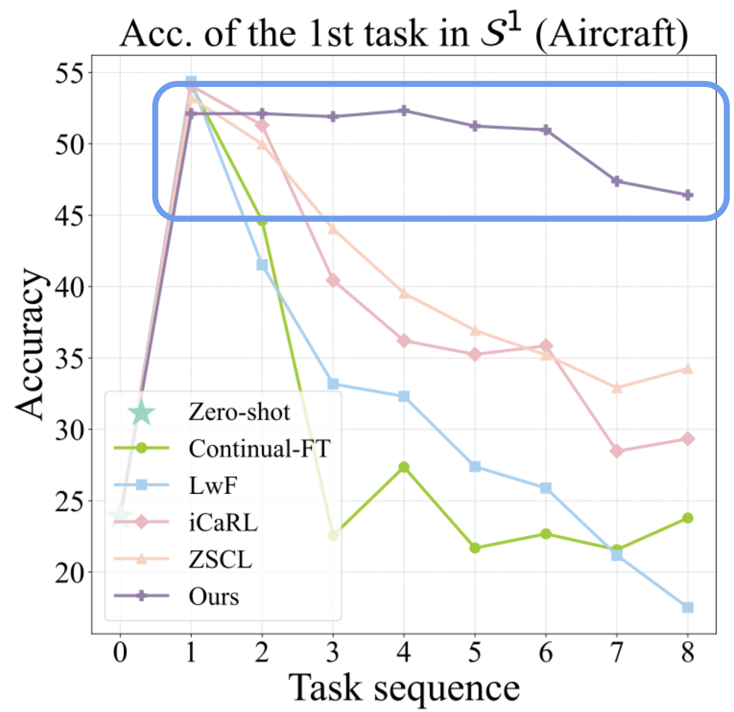

Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Results**
  - Successfully preserve the zero-shot ability for unseen data
  - Mitigate the catastrophic forgetting of previously learned data




Successfully preserve zero-shot ability for unseen data




Largely mitigate the performance gap

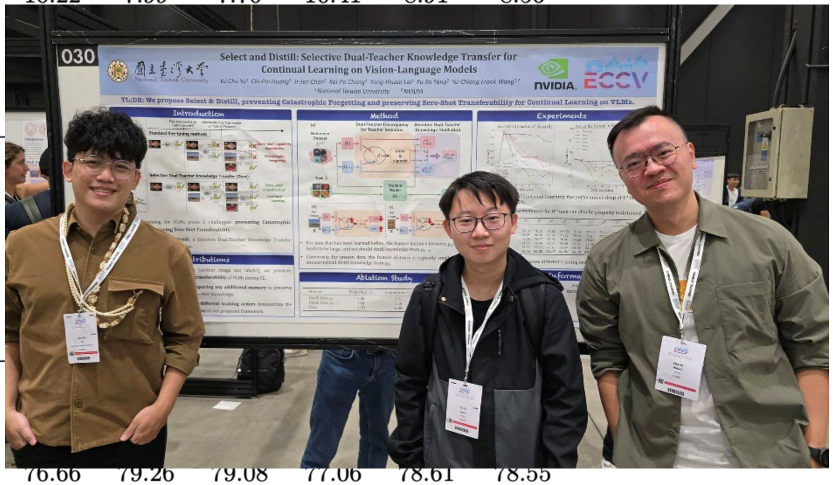


[Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models]

# Select and Distill, NTU, ECCV'24 (cont'd)

- **Robustness**
  - We shuffle the training orders, producing 8 different training sequences.
  - Our methods showing state-of-the-art performance on all metrics, and the results are stable across all training sequences.

| Method / Sequence | $\mathcal{S}^1$ | $\mathcal{S}^2$ | $\mathcal{S}^3$ | $\mathcal{S}^4$ | $\mathcal{S}^5$ | $\mathcal{S}^6$ | $\mathcal{S}^7$ | $\mathcal{S}^8$ | **Mean** |
|---|---|---|---|---|---|---|---|---|---|
| **Catastrophic forgetting (↓)** | | | | | | | | | |
| Continual FT | 10.98 | 10.60 | 8.80 | 19.17 | 10.11 | 11.95 | 15.19 | 9.48 | 12.04 |
| LwF [24] | 10.38 | 6.52 | 6.37 | 10.22 | 7.99 | 7.70 | 10.41 | 8.91 | 8.56 |
| iCaRL [35] | 8.42 | 7.00 | 6.45 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| ZSCL [50] | 4.67 | 2.35 | 2.13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| MoE-Adapters [48] | 2.74 | 4.71 | 4.28 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Ours | **1.70** | **1.16** | **0.89** | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| **Zero-shot degradation (↓)** | | | | | | | | | |
| Continual FT | 24.81 | 23.58 | 19.54 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| LwF [24] | 10.75 | 10.23 | 8.63 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| iCaRL [35] | 13.77 | 12.68 | 11.28 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| ZSCL [50] | 3.44 | 3.94 | 4.02 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| MoE-Adapters [48] | 1.62 | 2.58 | **1.04** | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Ours | **1.55** | **2.04** | 1.21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| **Average accuracy (↑)** | | | | | | | | | |
| Continual FT | 76.16 | 76.24 | 78.03 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| LwF [24] | 76.78 | 80.45 | 80.65 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| iCaRL [35] | 77.99 | 79.77 | 79.93 | 76.66 | 79.26 | 79.08 | 77.06 | 78.61 | 78.55 |
| ZSCL [50] | 81.89 | 83.98 | 84.30 | 83.49 | 83.41 | 82.38 | 81.92 | 81.97 | 82.92 |
| MoE-Adapters [48] | 82.71 | 80.74 | 81.15 | 83.97 | 83.68 | 83.68 | 82.73 | 79.68 | 82.29 |
| Ours | **84.48** | **84.92** | **84.97** | **84.89** | **85.50** | **85.07** | **85.02** | **84.52** | **84.92** |

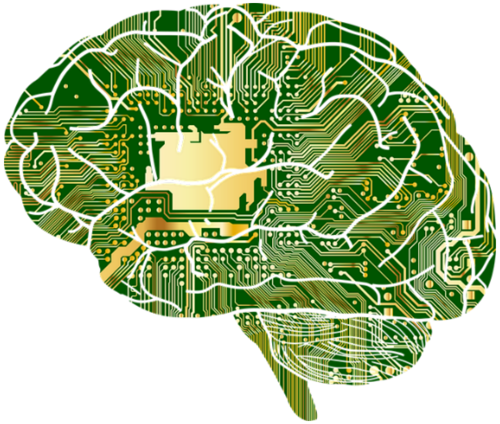


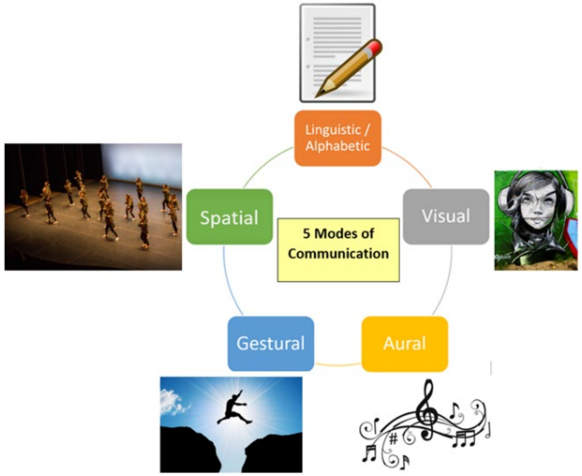



[Select and Distill: Selective Dual-Teacher Knowledge Transfer for Continual Learning on Vision-Language Models]

# What to Be Covered Today…

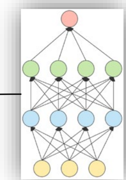
- **Additional Topics in DLCV**
  - Continual Learning
  - Meta Learning
  - Domain Generalization
  - Federated Learning
- **Experience Sharing**
  - Tim Chou (MS, GICE, NTU 2023), AI SW Engineer, NVIDIA





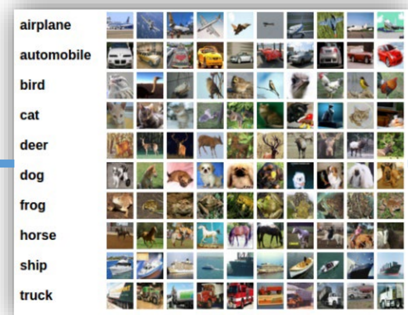

# Meta Learning 元學習

- Meta Learning ⊆ Supervised Learning
- For Supervised Learning,
  - Given training data **D** = {**X**, **Y**}, learn function/model ***f*** so that $f(x_i) = y_i$

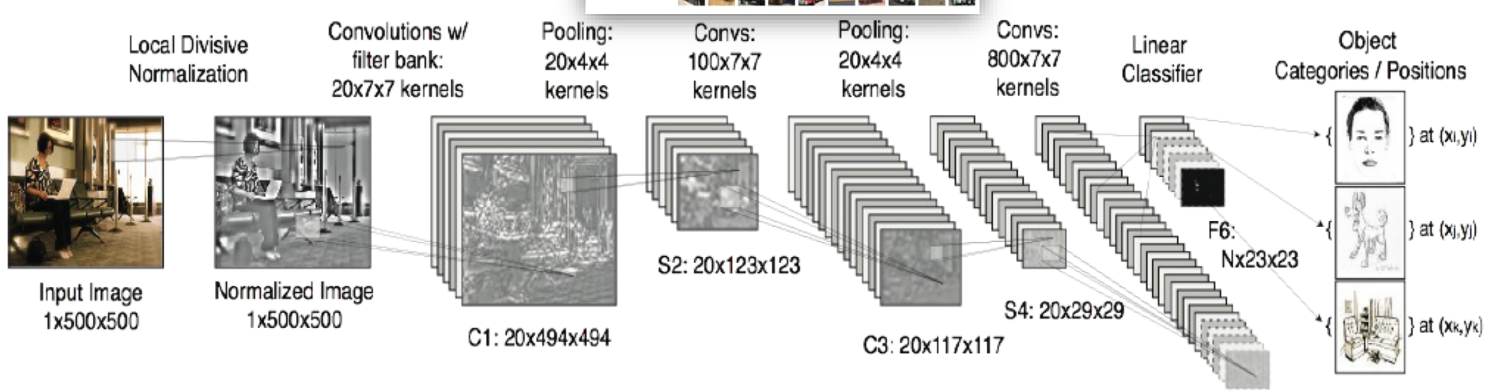


$f($  $)$ = "Cat"

Training data **X**

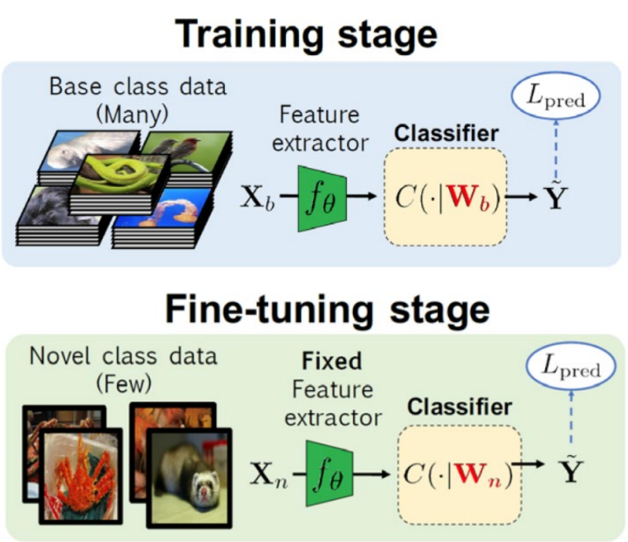


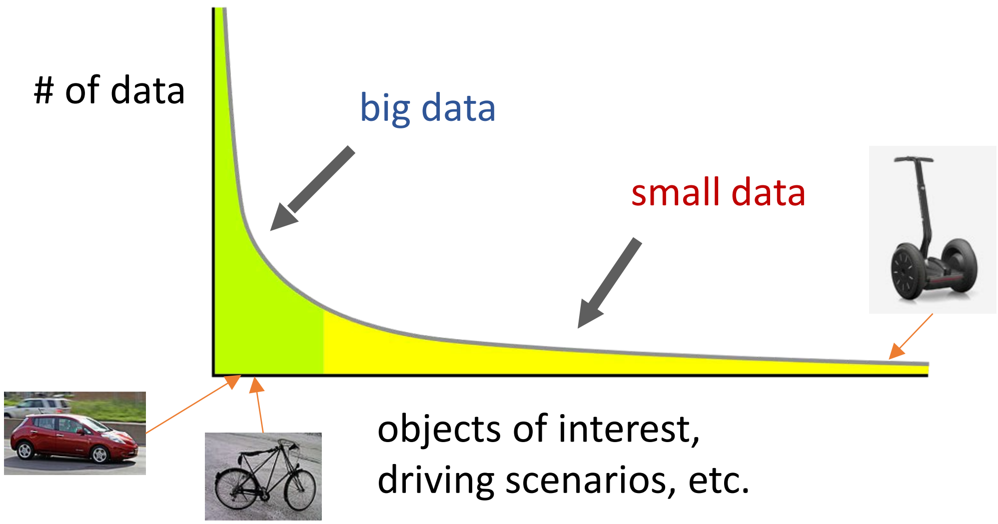

Ground truth labels **Y**



# What If Only Limited Amount of Data Available?

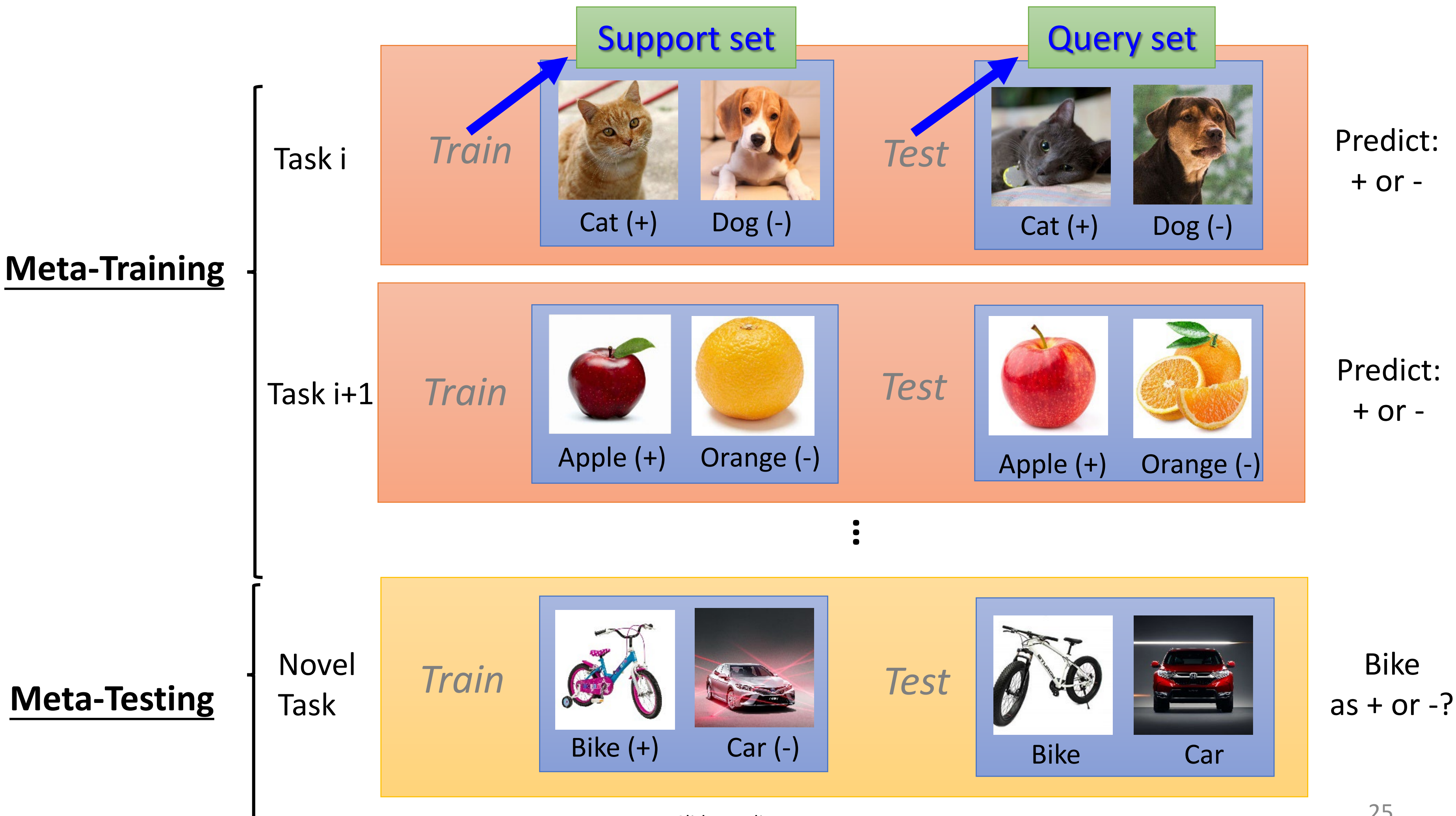
- **Naive transfer?**
  - **Model finetuning**:
    - Train a learning model (e.g., CNN) on large-size data (base classes), followed by finetuning on small-size data (novel classes).
    - That is, freeze feature backbone (learned from base classes) and learn/update classifier weights for novel classes.
  - Question: What would be the concern/limitation?



# Meta Learning = Learning to Learn

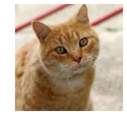
- Let’s consider the following “2-way 1-shot” learning scheme:




Slide credit: H.-Y. Lee

# Some ML Backgrounds (if time permits…)

- (Standard) Supervised Learning



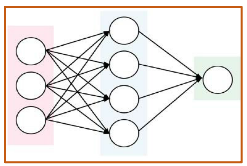
“cat”

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

input (e.g., image)

label

$$\arg\max_{\phi} \log p(\phi|\mathcal{D})$$



model parameters

training data

$$= \arg\max_{\phi} \log p(\mathcal{D}|\phi) + \log p(\phi)$$

data likelihood

regularizer (e.g., weight decay)

$$= \arg\max_{\phi} \sum_{i} \log p(y_i|x_i, \phi) + \log p(\phi)$$

- We know the biggest problem is that…
  - Can’t always collect a large amount of labeled data **D** in advance.

- Now, for the *Meta Learning* scheme…

supervised learning:

$$\arg\max_{\phi} \log p(\phi|\mathcal{D})$$

Few-shot data domain of interest

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$


➡ can we incorporate *additional* data?

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$$

➡ $\arg\max_{\phi} \log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}})$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\}$$

$\mathcal{D}_{\text{meta-train}}$

$\mathcal{D}_1$

$\mathcal{D}_2$

$\mathcal{D}$

# What Meta Learning Solves:

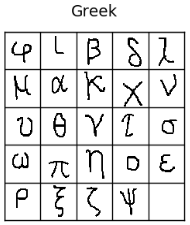
Object label: “cat”

Object ID: “person”

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$$

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_k, y_k)\}$$




- what if we don’t want to keep $\mathcal{D}_{\text{meta-train}}$ around forever?
- learn *meta-parameters* $\theta$: $p(\theta|\mathcal{D}_{\text{meta-train}})$

whatever we need to know about $\mathcal{D}_{\text{meta-train}}$ to solve new tasks

- $$\log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}}) = \log \int_{\Theta} p(\phi|\mathcal{D}, \theta) p(\theta|\mathcal{D}_{\text{meta-train}}) d\theta$$
$$\approx \log p(\phi|\mathcal{D}, \theta^\star) + \log p(\theta^\star|\mathcal{D}_{\text{meta-train}})$$

# What Meta Learning Solves:

$$\arg\max_{\phi} \log p(\phi | \mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

Object label: “cat”

Object ID: “person”

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$$

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

Greek

$$\log p(\phi | \mathcal{D}, \mathcal{D}_{\text{meta-train}}) = \log \int_{\Theta} p(\phi | \mathcal{D}, \theta) p(\theta | \mathcal{D}_{\text{meta-train}}) d\theta$$

$$\approx \log p(\phi | \mathcal{D}, \theta^{\star}) + \log p(\theta^{\star} | \mathcal{D}_{\text{meta-train}})$$

---

$$\arg\max_{\phi} \log p(\phi | \mathcal{D}, \mathcal{D}_{\text{meta-train}}) \approx \arg\max_{\phi} \log p(\phi | \mathcal{D}, \theta^{\star})$$

- What meta learning cares is the learning of Φ from D (and implicitly from $D_{\text{meta-train}}$)
- What makes meta learning challenging is the learning of optimal Θ* from $D_{\text{meta-train}}$:

$$\boxed{\theta^{\star} = \arg\max_{\theta} \log p(\theta | \mathcal{D}_{\text{meta-train}})}$$

# A Quick Review:

Person ID:
"Brad Pitt"

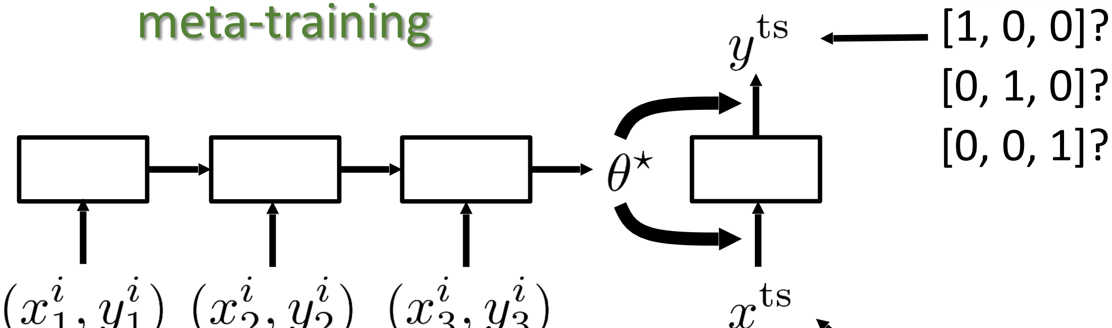
- **Meta training:** $\theta^\star = \arg\max_\theta \log p(\theta|\mathcal{D}_{\text{meta-train}})$
- Meta testing: $\phi^\star = \arg\max_\phi \log p(\phi|\mathcal{D}, \theta^\star)$

$$\boxed{\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}}$$

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_k, y_k)\}$$

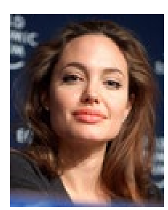
$$\mathcal{D}_i = \{(x_1^i, y_1^i), \dots, (x_k^i, y_k^i)\}$$

meta-training




$$\underbrace{(x_1^i, y_1^i)\ (x_2^i, y_2^i)\ (x_3^i, y_3^i)}_{\mathcal{D}_i}$$





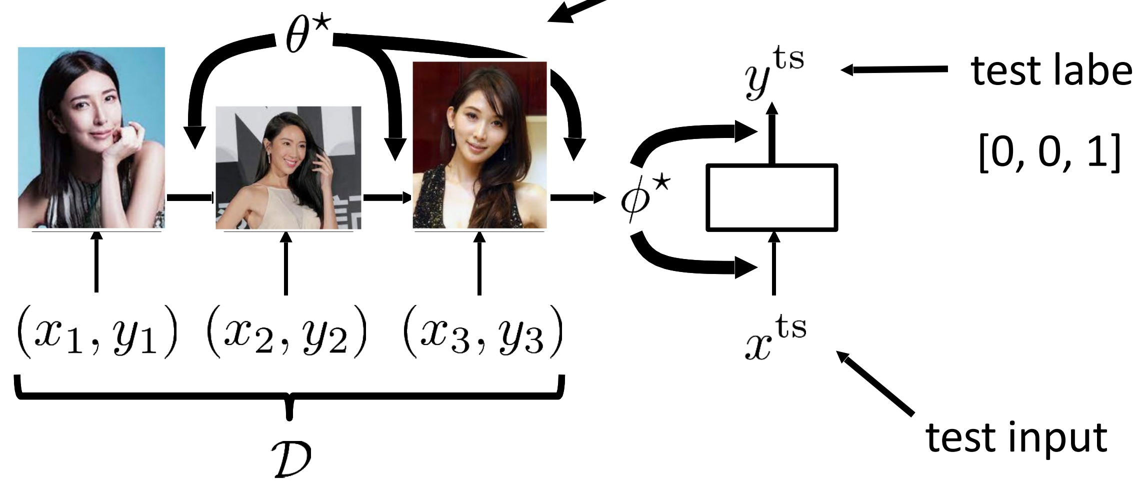

# A Quick Review (cont'd):

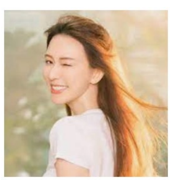
- Meta training: $\theta^\star = \arg\max_\theta \log p(\theta|\mathcal{D}_{\text{meta-train}})$
- $\boxed{\text{Meta testing:}}$ $\phi^\star = \arg\max_\phi \log p(\phi|\mathcal{D}, \theta^\star)$

$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$

$\boxed{\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}}$

meta-testing



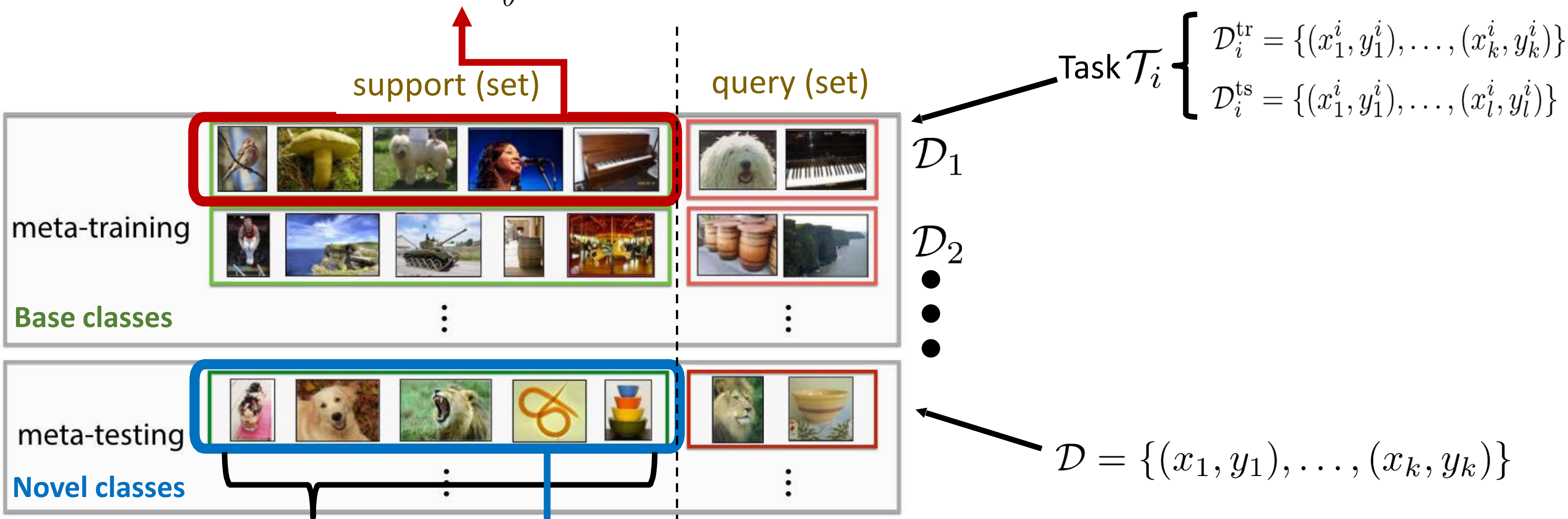



✓ **Key Idea:**

The condition/mechanism of meta-training and meta-testing must match.
In other words, meta learning is to learn the mechanism, ***not*** to fit the data/labels.

# Meta-Learning Terminologies & Additional Remarks

meta-learning: $\theta^\star = \arg\max_\theta \log p(\theta | \mathcal{D}_{\text{meta-train}})$



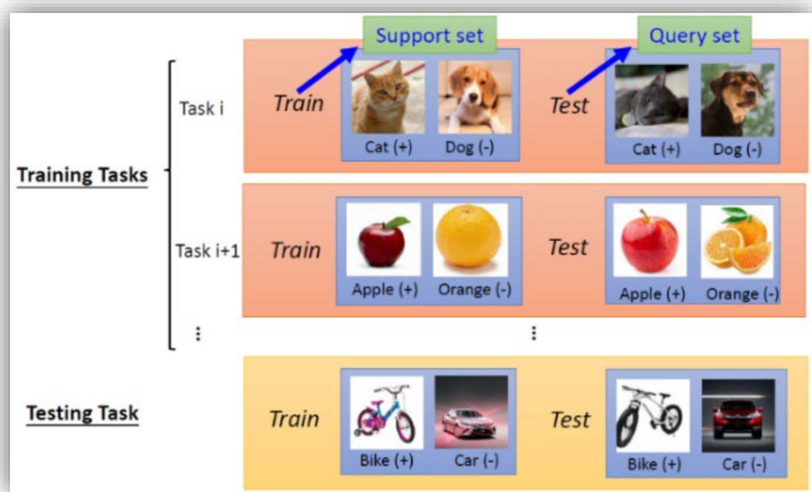

Meta testing: adaptation: $\phi^\star = \arg\max_\phi \log p(\phi | \mathcal{D}, \theta^\star)$

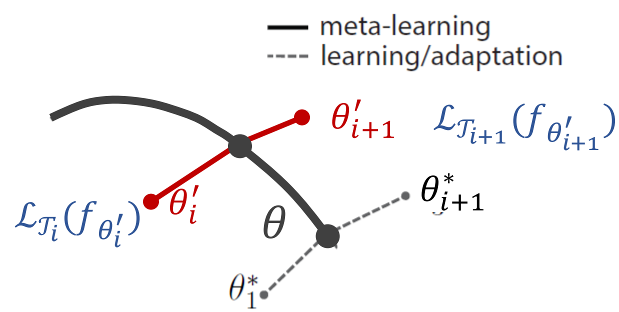
✓ **Remarks**

- Meta learning: learn a N-way K-shot learning mechanism, ***not*** fitting data/labels
- The conditions (i.e., N-way K-shot) of meta-training and meta-testing must match.
- Question: Remarks on N & K vs. performances?

# Approach #1: Optimization-Based Approach

- **M**odel-**A**gnostic **M**eta-**L**earning (MAML)*
  - Key idea:
    - Train over many tasks (with a small amount of data & few gradient steps), so that the learned model parameter would generalize to novel tasks
    - Learning to initialize/fine-tune
  - Meta-Learner $\Phi \rightarrow \Theta_0$:
    - Learn a parameter initialization $\Theta_0$ of model that transfers/generalizes to novel tasks well.
    - That is, learn model $\Theta_0$ which can be fine-tuned by novel tasks efficiently/effectively.



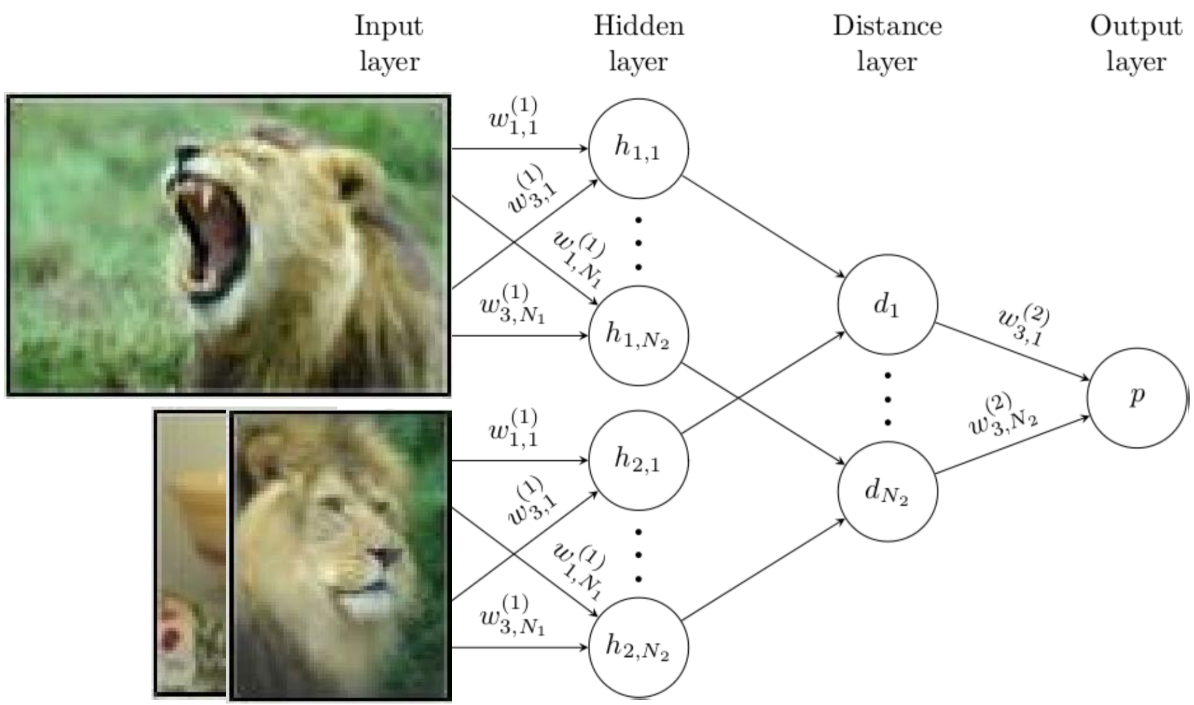




optimize model parameter $\theta$ so that it can quickly adapt to new tasks

*Finn, Abbeel, Levine, ICML 2017

# Approach #2: Non-Parametric Approach

- Can models learn to compare?
- E.g., Siamese Network
  - Learn a network to determine whether a pair of images are of the same category.



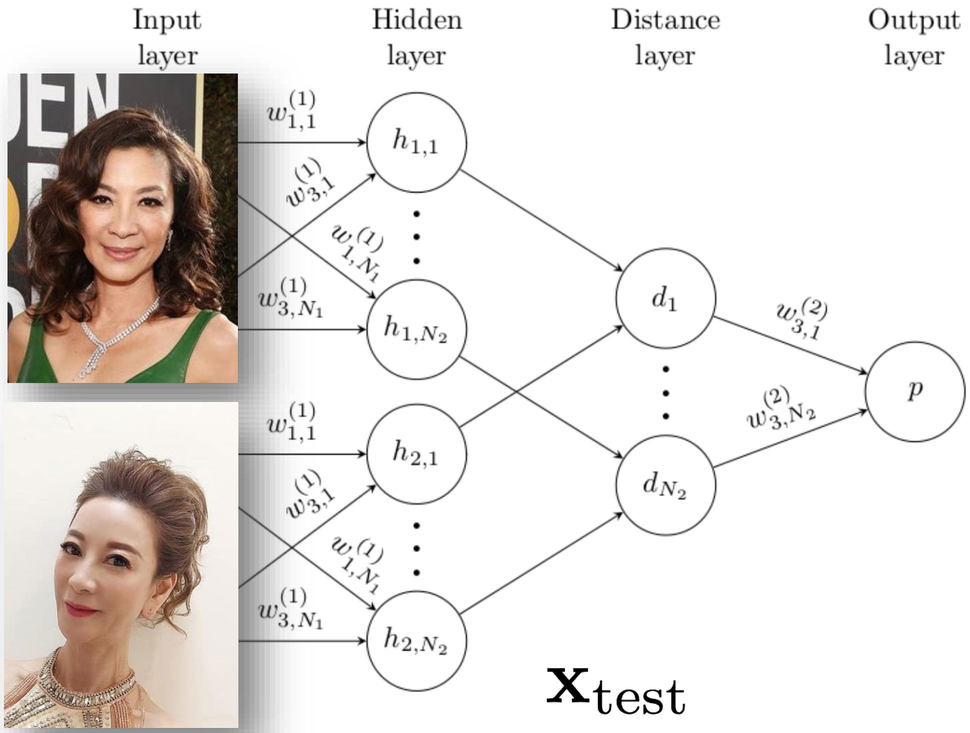

label

**1**

Koch et al., Siamese Neural Networks for One-Shot Image Recognition, ICML WS 2015

# Learn to Compare (cont'd)


- Siamese Network (cont'd)
  - Meta-training/testing: learn to match
    - Question: output label of the following example is **1** or **0**? (i.e., **same ID** or **not**)



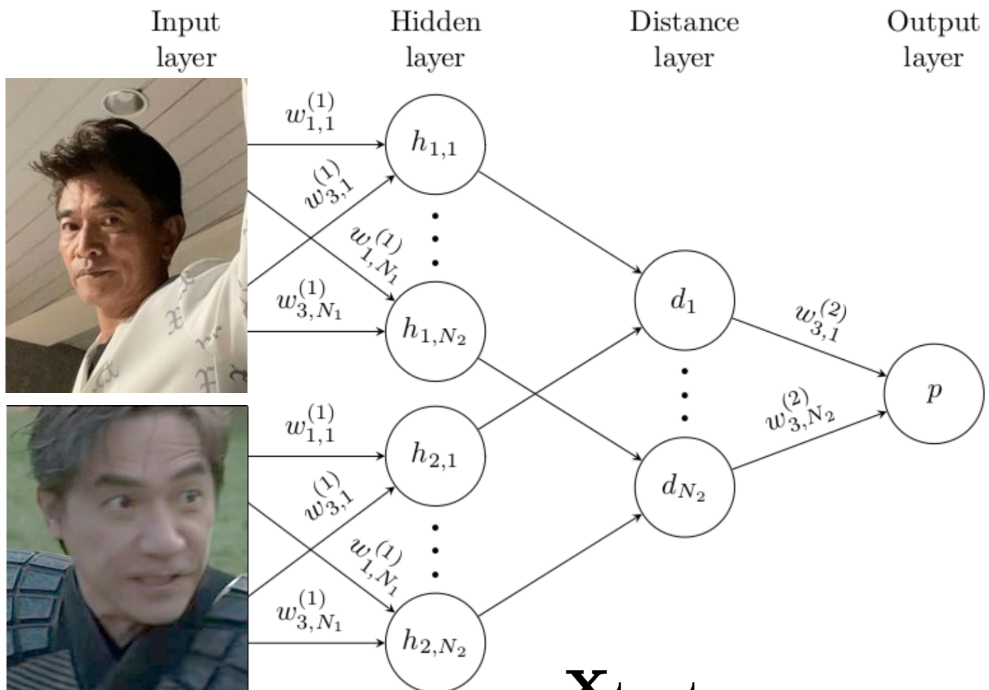

label

?

$\mathbf{x}_{\text{test}}$

# Learn to Compare (cont’d)

- Siamese Network (cont’d)
  - Meta-training/testing: learn to match
    - Question: output label of the following example is 1 or 0? (i.e., same ID or not)




$\mathbf{x}_{\text{test}}$

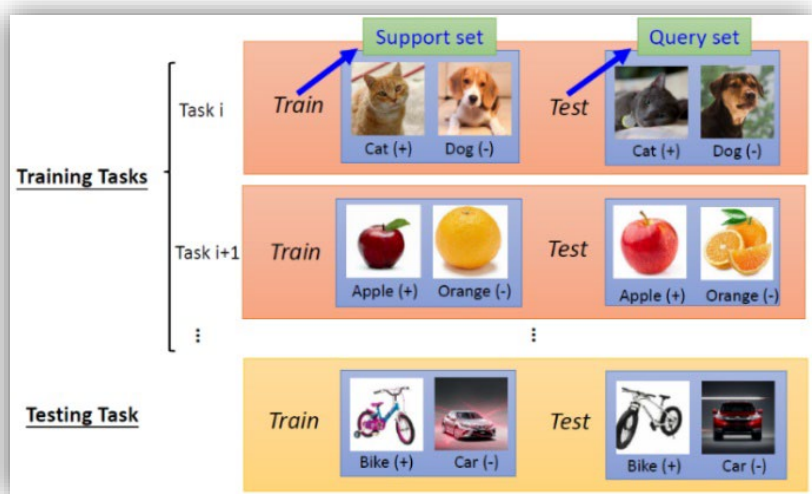
label

?

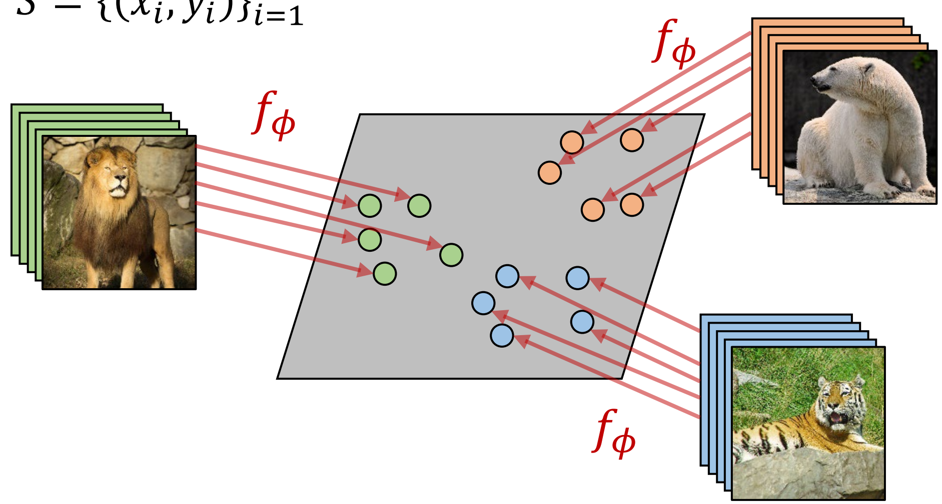

- What did we learn from these examples?
- And, can we perform multi-way classification (beyond matching)?

# Learn to Compare…with the Representative Ones!

- **Prototypical Networks**
  - Learn a model which properly describes data in terms of intra/inter-class info.
  - Learn a prototype for each class, with data similarity/separation guarantees.

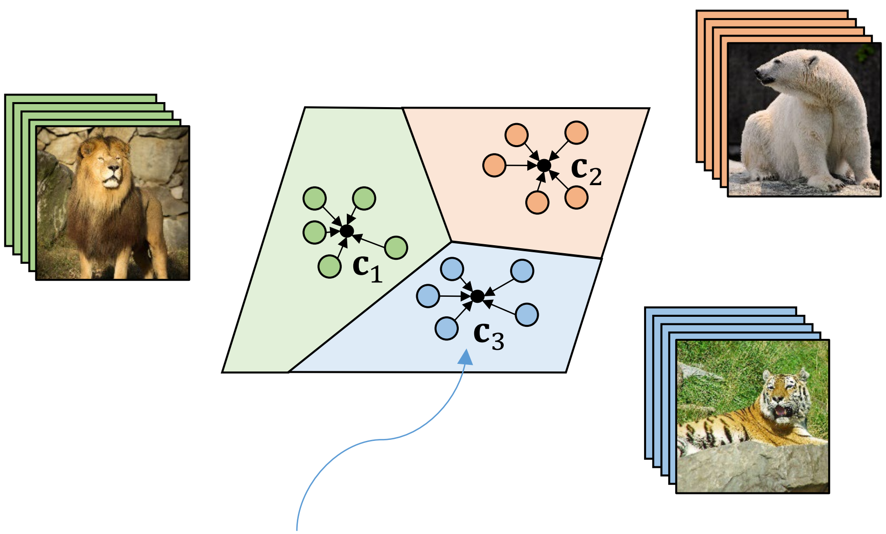


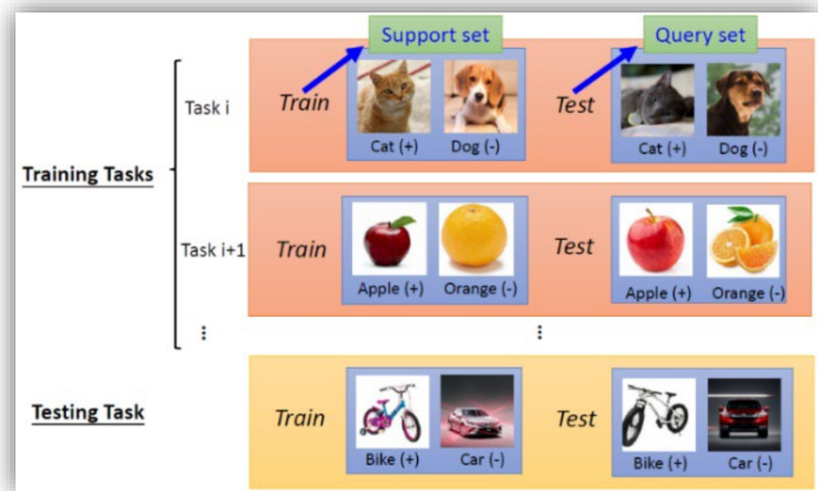

support set
$S = \{(x_i, y_i)\}_{i=1}^{k}$




Snell et al., Prototypical Networks for Few-Shot Learning, NIPS 2017

- **Prototypical Networks** (cont'd)
  - Learn a model which properly describes data in terms of intra/inter-class info.
  - It learns a prototype for each class, with data similarity/separation guarantees.
  - For DL version, the above embedding space is derived by a non-linear mapping $f_\phi$ and the representatives (or anchors) of each class is the **mean feature vector** $\mathbf{c}_k$.




$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$$, where $S_k \subset S$ is the subset of support set $S$ with class $k$

Snell et al., Prototypical Networks for Few-Shot Learning, NIPS 2017

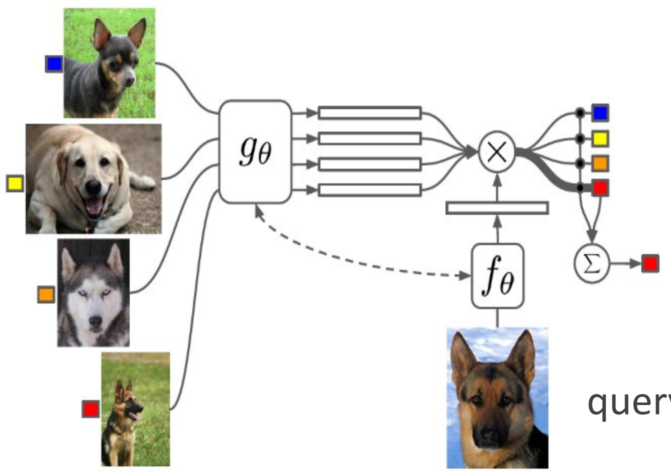
# Learn to Compare

- **Matching Networks**
  - Inspired by the **attention** mechanism,
    access an augmented memory containing useful info to solve the task of interest
  - The authors proposed a weighted nearest-neighbor classifier,
    with attention over a learned embedding from the support set $S = \{(x_i, y_i)\}_{i=1}^{k}$,
    so that the label of the query $\hat{x}$ can be predicted.

$c(.,.)$: cosine similarity

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i \quad \text{with} \quad a(\hat{x}, x_i) = e^{\boxed{c(f(\hat{x}), g(x_i))}} / \sum_{j=1}^{k} e^{\boxed{c(f(\hat{x}), g(x_j))}}$$

support set $S = \{(x_i, y_i)\}_{i=1}^{k}$

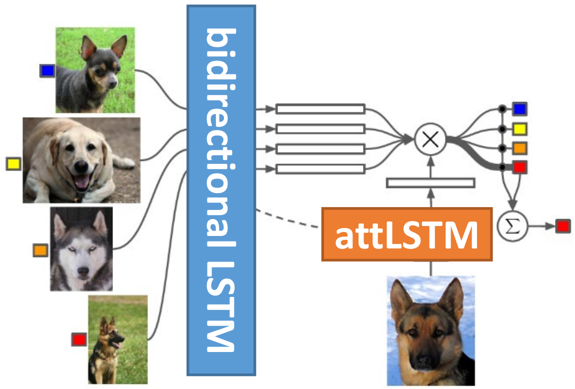



query example $\hat{x}$

Vinyals et al., "Matching Networks for One Shot Learning," NIPS, 2016

- **Matching Networks** (cont'd)
  - Full context embedding (FCE)
  - Each element in $S$ should not be embedded independently of other elements
    - $g(x_i)$ → $g(S)$ as a **bidirectional LSTM** by considering the whole $S$ as a **sequence**
  - Also, $S$ should be able to modify the way we embed $\hat{x}$
    - $f(\hat{x})$ → $f(\hat{x}, S)$ as an **LSTM** with **read-attention** over $g(S)$: $\text{attLSTM}(f'(\hat{x}), g(S), K)$, where $f'(\hat{x})$ is the (fixed) CNN feature, and $K$ is the number of unrolling steps
  - Experiment results on *mini*ImageNet

support set
$S = \{(x_i, y_i)\}_{i=1}^{k}$




query example $\hat{x}$

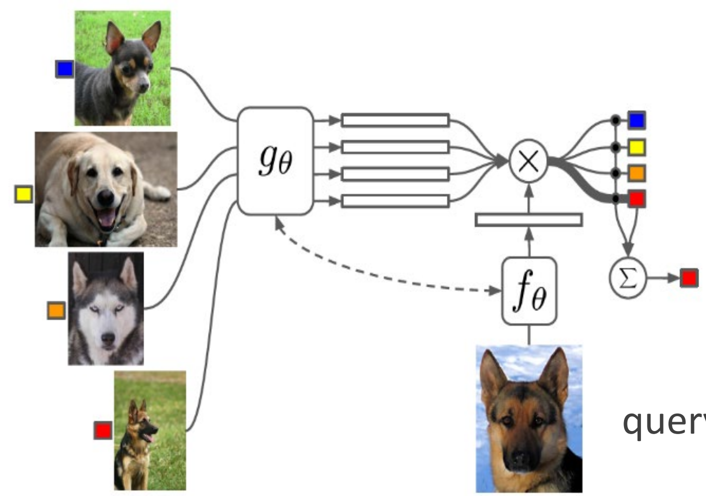
| Model | Matching Fn | Fine Tune | 5-way Acc 1-shot | 5-way Acc 5-shot |
|---|---|---|---|---|
| Pixels | Cosine | N | 23.0% | 26.6% |
| Baseline Classifier | Cosine | N | 36.6% | 46.0% |
| Baseline Classifier | Cosine | Y | 36.2% | 52.2% |
| Baseline Classifier | Softmax | Y | 38.4% | 51.2% |
| Matching Nets (Ours) | Cosine | N | 41.2% | 56.2% |
| Matching Nets (Ours) | Cosine | Y | 42.4% | 58.0% |
| Matching Nets (Ours) | Cosine (FCE) | N | 44.2% | 57.0% |
| Matching Nets (Ours) | Cosine (FCE) | Y | **46.6%** | **60.0%** |

Vinyals et al., "Matching Networks for One Shot Learning," NIPS, 2016

# Learn to Compare

- **Matching Networks** (cont'd)
  - If we have $g = f$,
    the model turns into a Siamese network like architecture
  - Also similar to prototypical network for **one**-shot learning

support set $S = \{(x_i, y_i)\}_{i=1}^{k}$

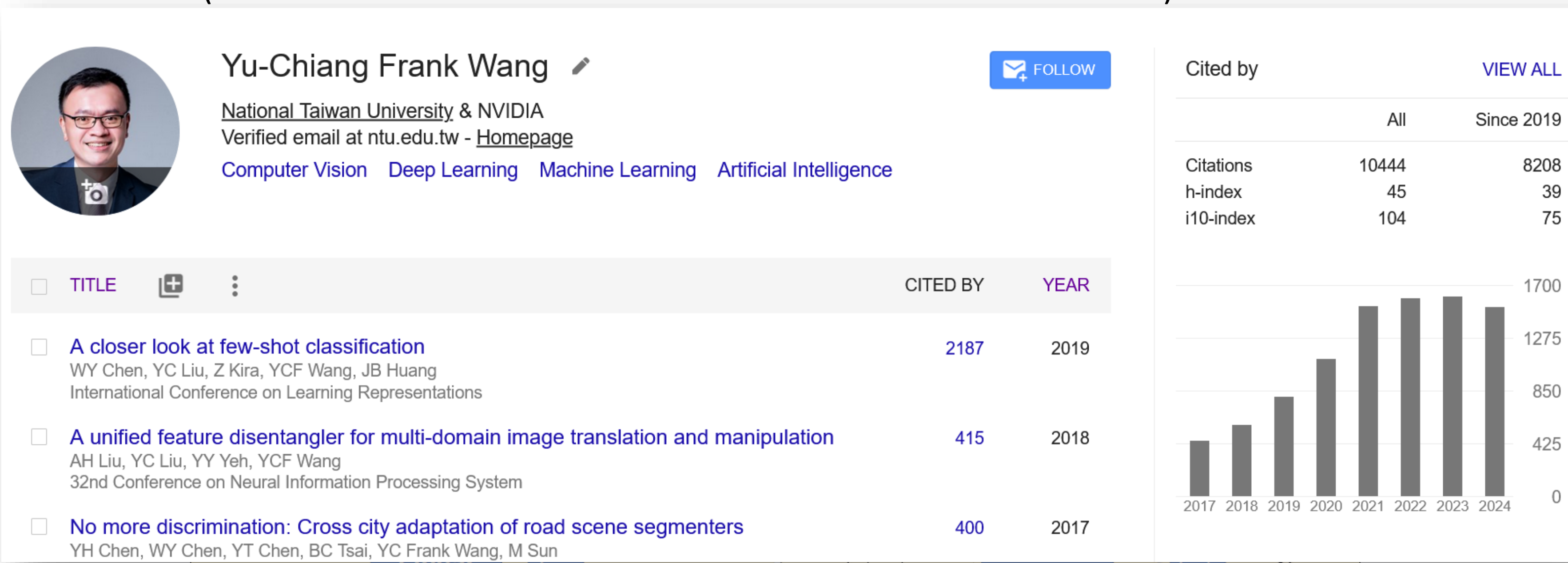


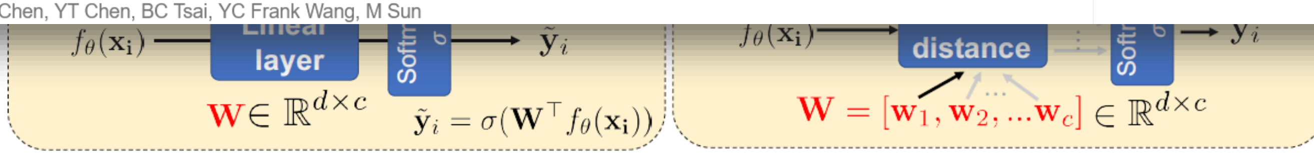

query example $\hat{x}$

Vinyals et al., "Matching Networks for One Shot Learning," NIPS, 2016

# Further Remarks:
# A Closer Look at FSL (1/3)

- Idea
  - **Deeper backbones** significantly reduce the gap across existing FSL methods. (with decreased domain shifts between base and novel classes)



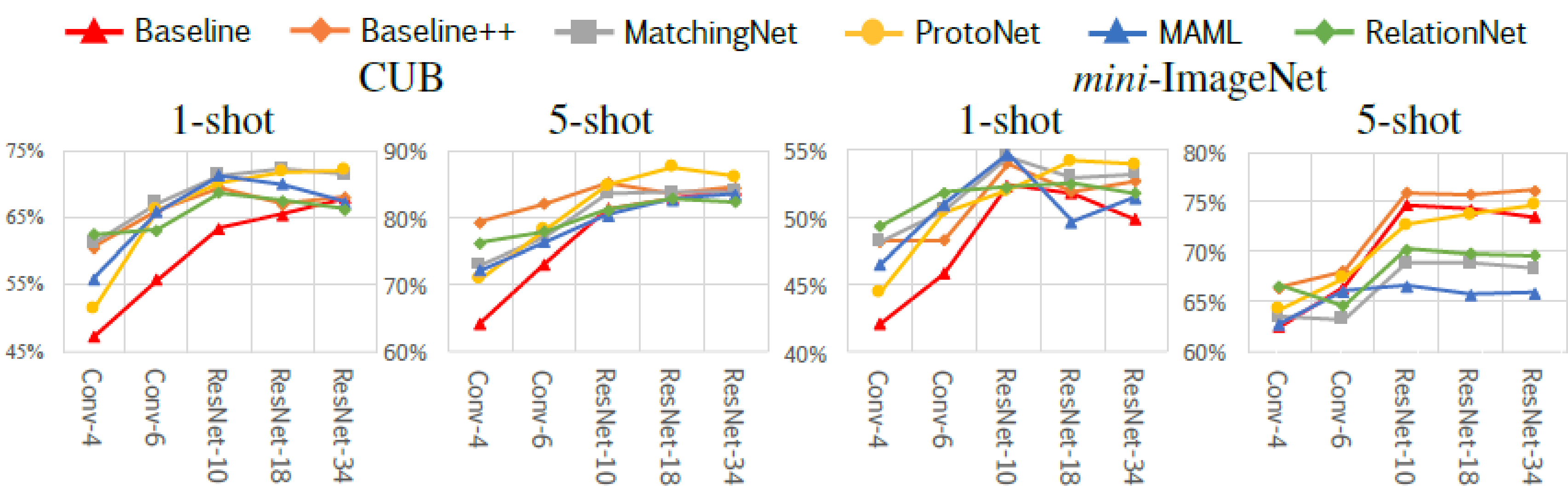

use **cosine distances** between the input feature and the weight vector for each class to reduce intra-class variations



Chen et al., A Closer Look at Few-shot Classification, *ICLR*, 2019

# A Closer Look at FSL (2/3)

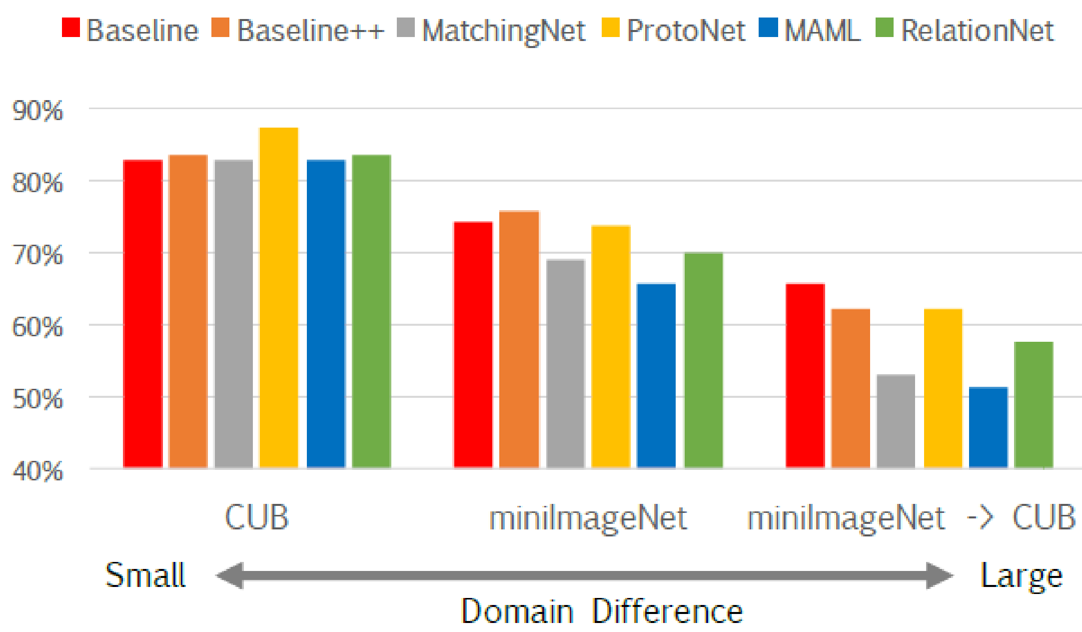
- Performance with deeper backbones
  - For CUB, gaps among different methods diminish as the backbone gets deeper.
  - For mini-ImageNet, some meta-learning methods are even beaten by baselines with a deeper backbone.




Chen et al., A Closer Look at Few-shot Classification, *ICLR*, 2019

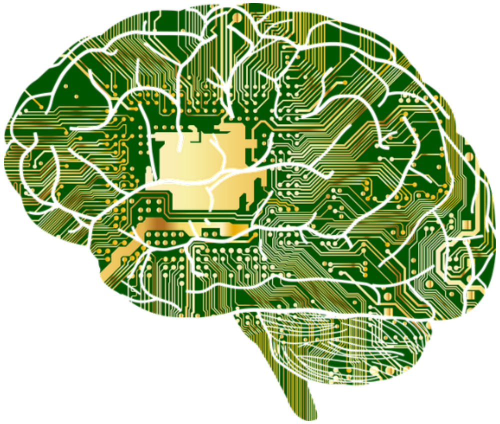
# A Closer Look at FSL (3/3)

- Performance with domain shifts (using ResNet-18)
  - Existing FSL methods fail to address large domain shifts (e.g., mini-ImageNet → CUB) and are inferior to the baseline methods.
  - This highlights the importance of learning to adapt to domain differences in FSL.

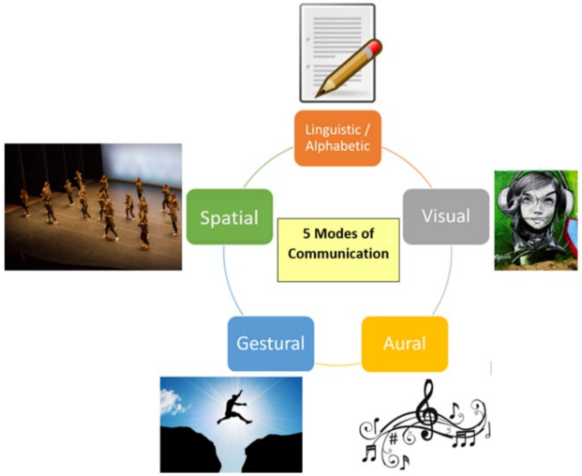



Chen et al., A Closer Look at Few-shot Classification, *ICLR*, 2019

# What to Be Covered Today…

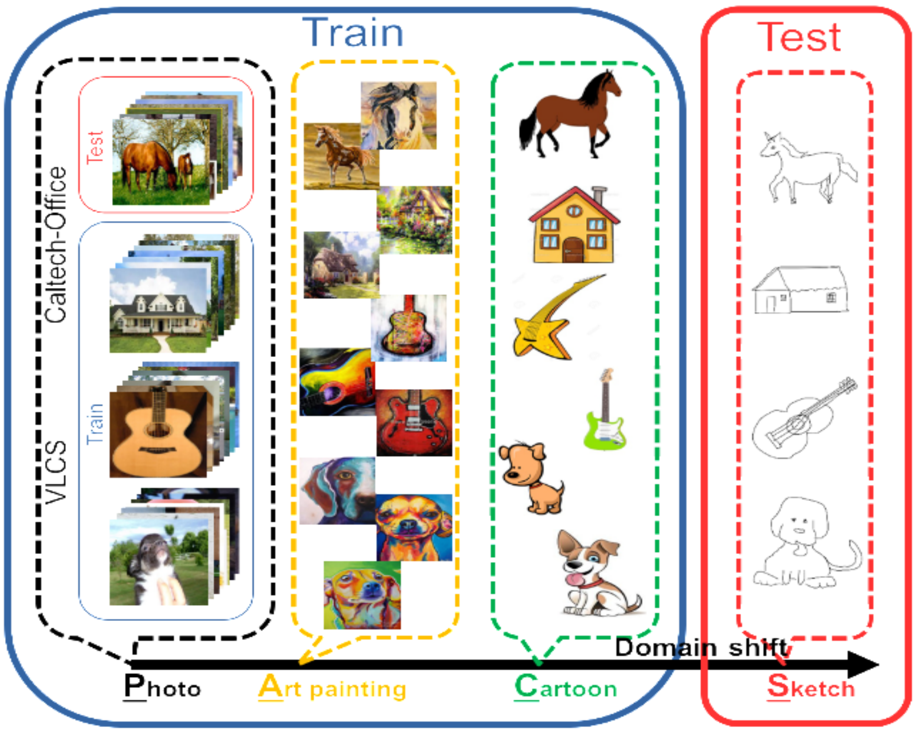


- **Additional Topics in DLCV**
  - Continual Learning
  - Meta Learning
  - Domain Generalization
  - Federated Learning
- **Experience Sharing**
  - Tim Chou (MS, GICE, NTU 2023), AI SW Engineer, NVIDIA




# Domain Generalization

- Input: Images and labels from multiple source domains
- Output: A well-generalized model for unseen target domains

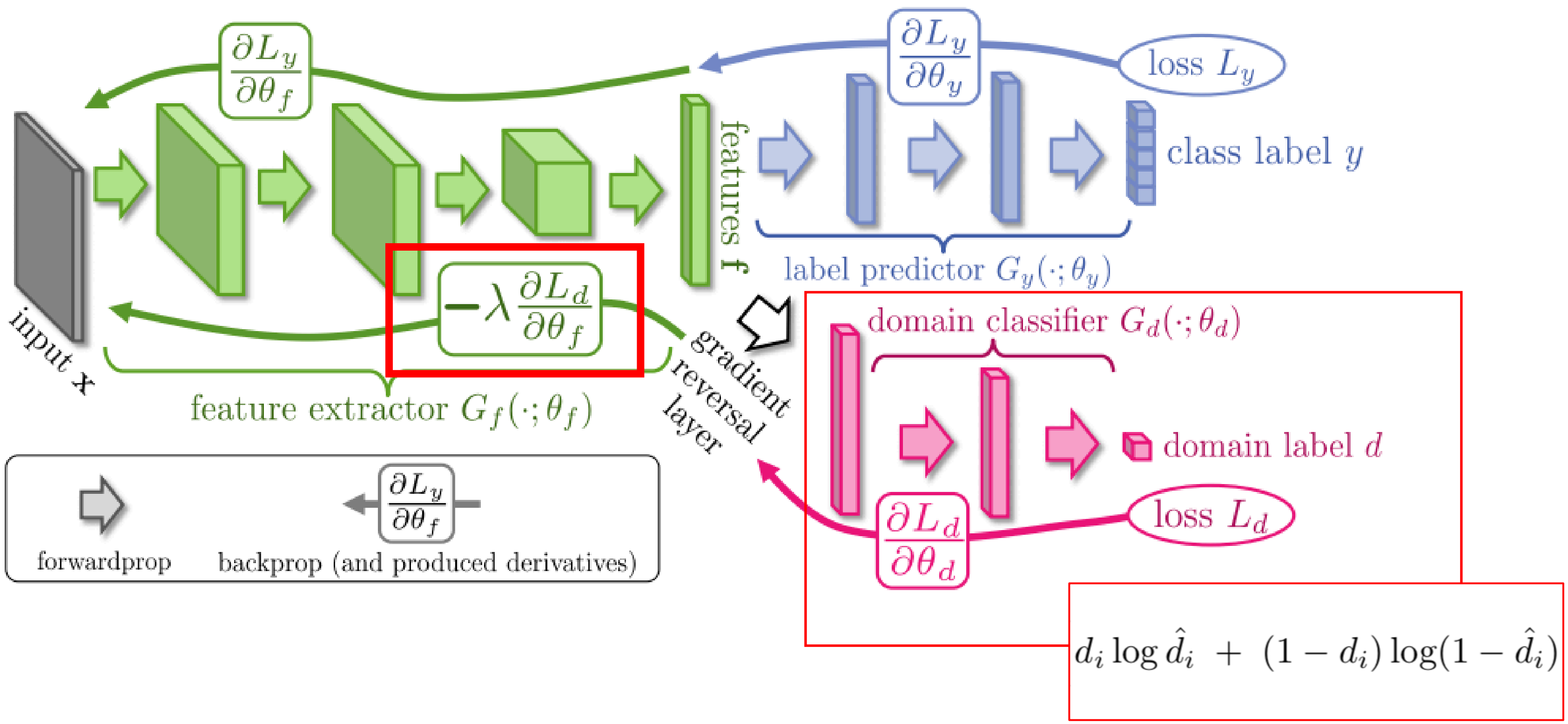



$D_S$ = {Photo, Painting, Cartoon}

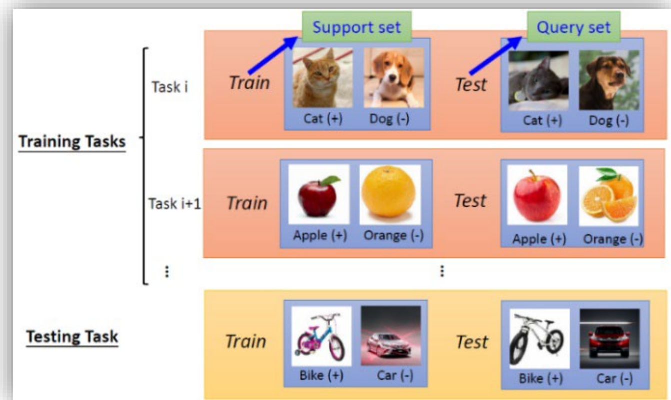
$D_T$ = {Sketch}

# Recap: Domain Adaptation

- Domain-Adversarial Training of Neural Networks (DANN)
  - Y. Ganin et al., ICML 2015
  - Maximize domain confusion = maximize domain classification loss
  - Minimize source-domain data classification loss
  - The derived feature f can be viewed as a disentangled & domain-invariant feature.

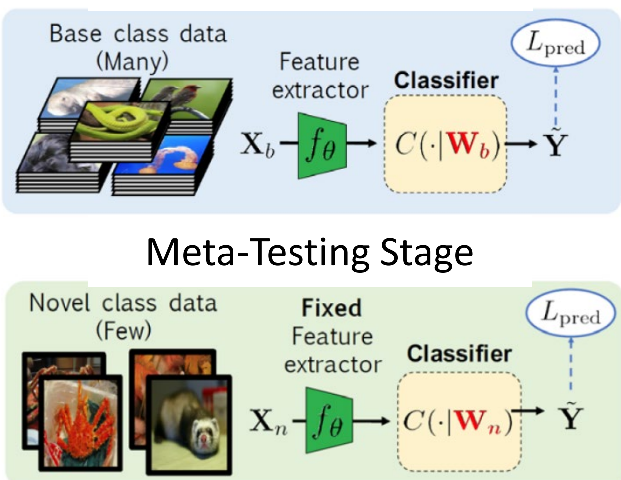


# Recap:
# Learn to Compare
# with the Representative Ones!



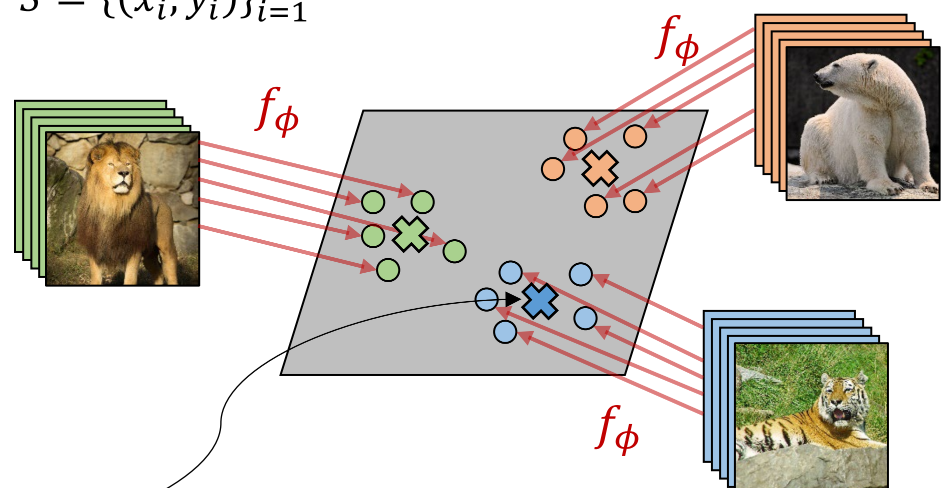

- Prototypical Networks
  - Learn a model which properly describes data in terms of intra/inter-class info.
  - It learns a prototype for each class, with data similarity/separation guarantees. For DL version, the learned feature space is derived by a non-linear mapping $f_\theta$ and the representatives (i.e., prototypes) of each class is the **mean feature vector $\mathbf{c}_k$**.

Meta-Training Stage




support set

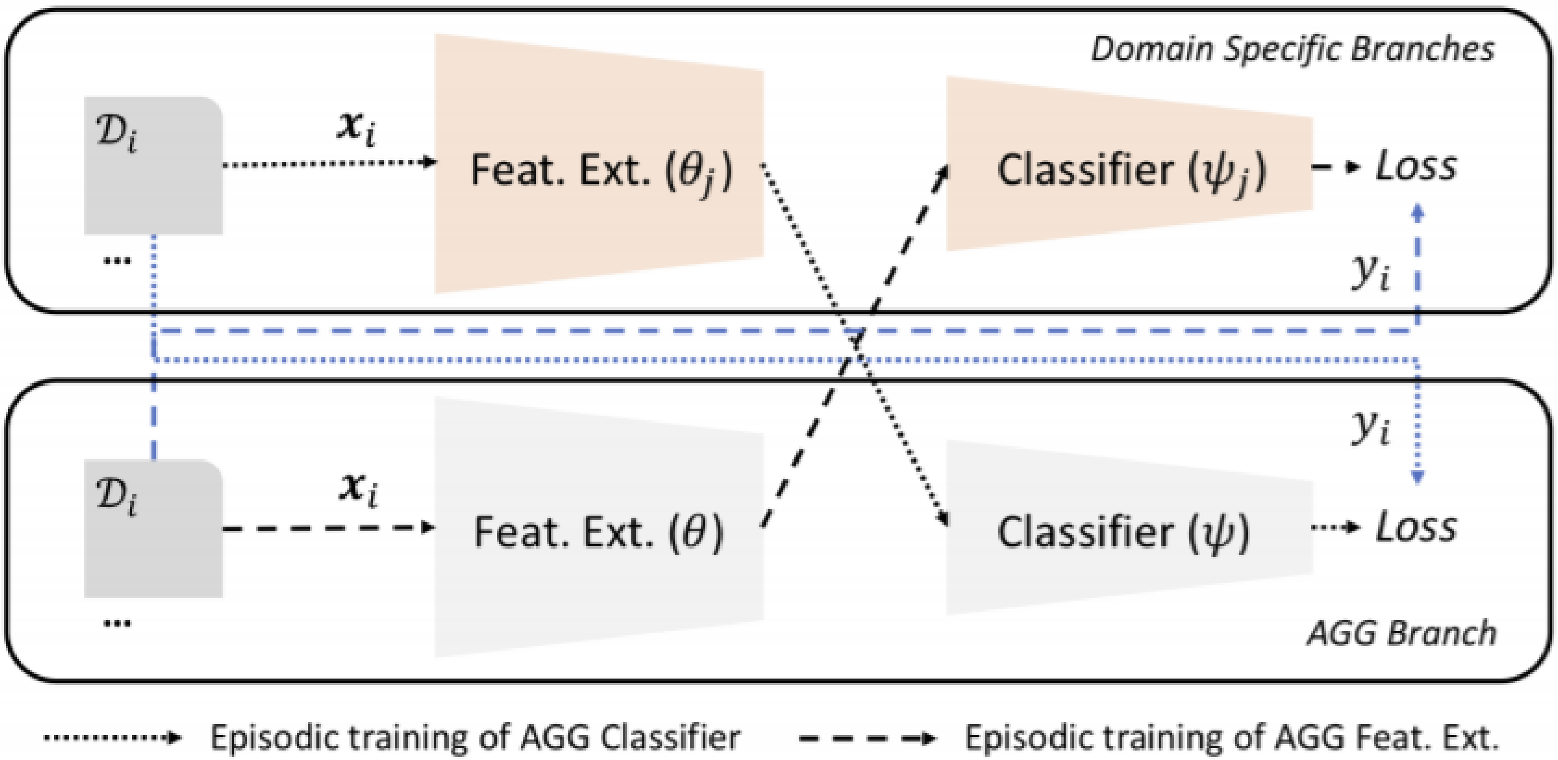
$S = \{(x_i, y_i)\}_{i=1}^{k}$




$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\theta(\mathbf{x}_i)$$, where $S_k \subset S$ indicates features of class $k$ from support set $S$

# Strategy of Episodic Training

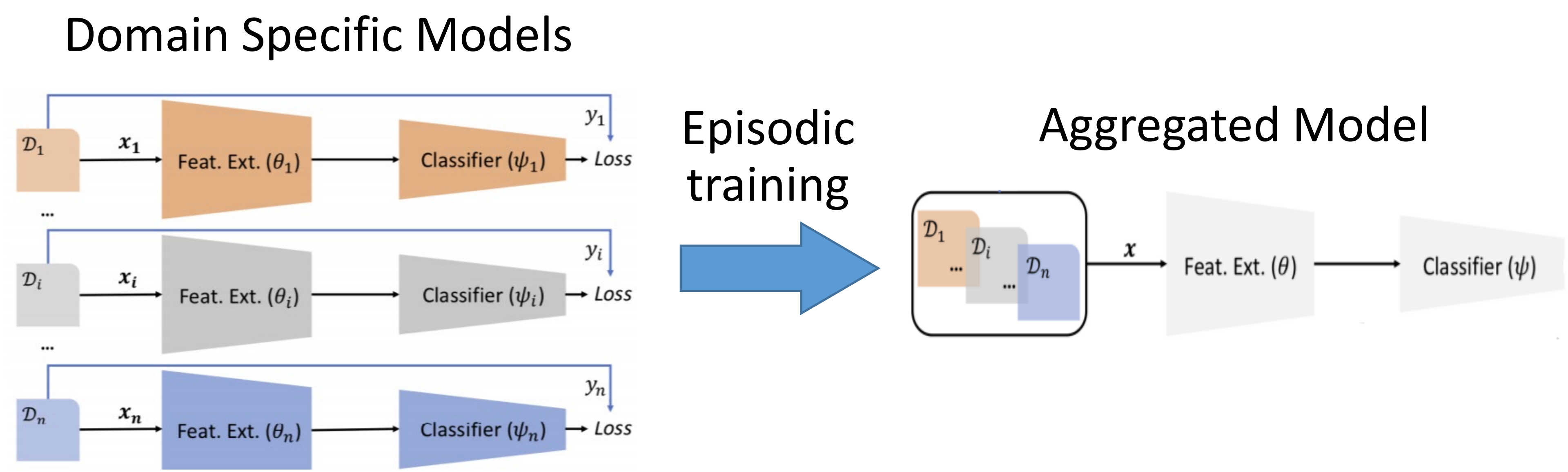
- **Episodic training for domain generalization** (ICCV'19)
- Generalize across domains via Meta-Learning




Zhang et al. : Episodic training for domain generalization. In ICCV (2019)

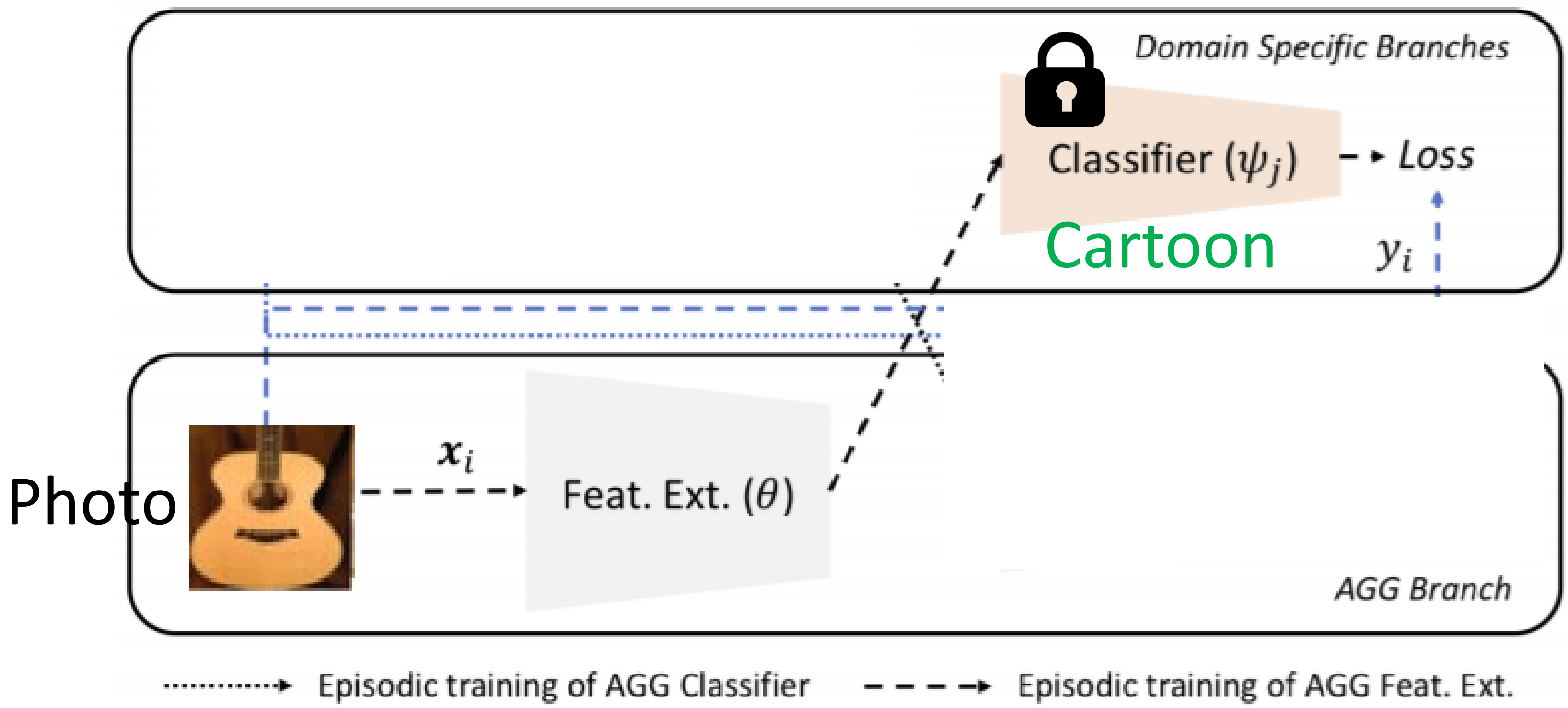
# Episodic Training (cont'd)

- Motivation



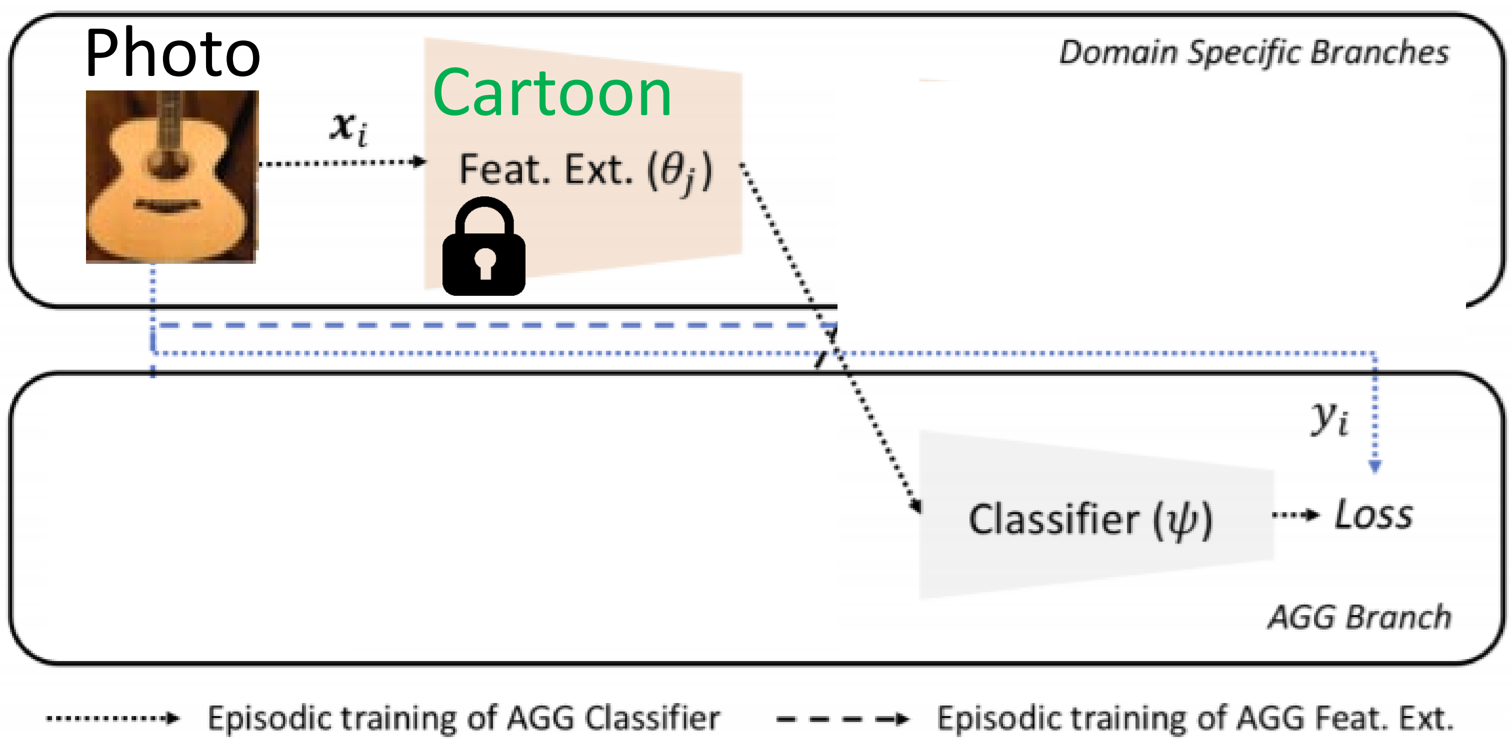
# Episodic Training (cont'd)

- Random sample two domains, e.g., Photo and Cartoon

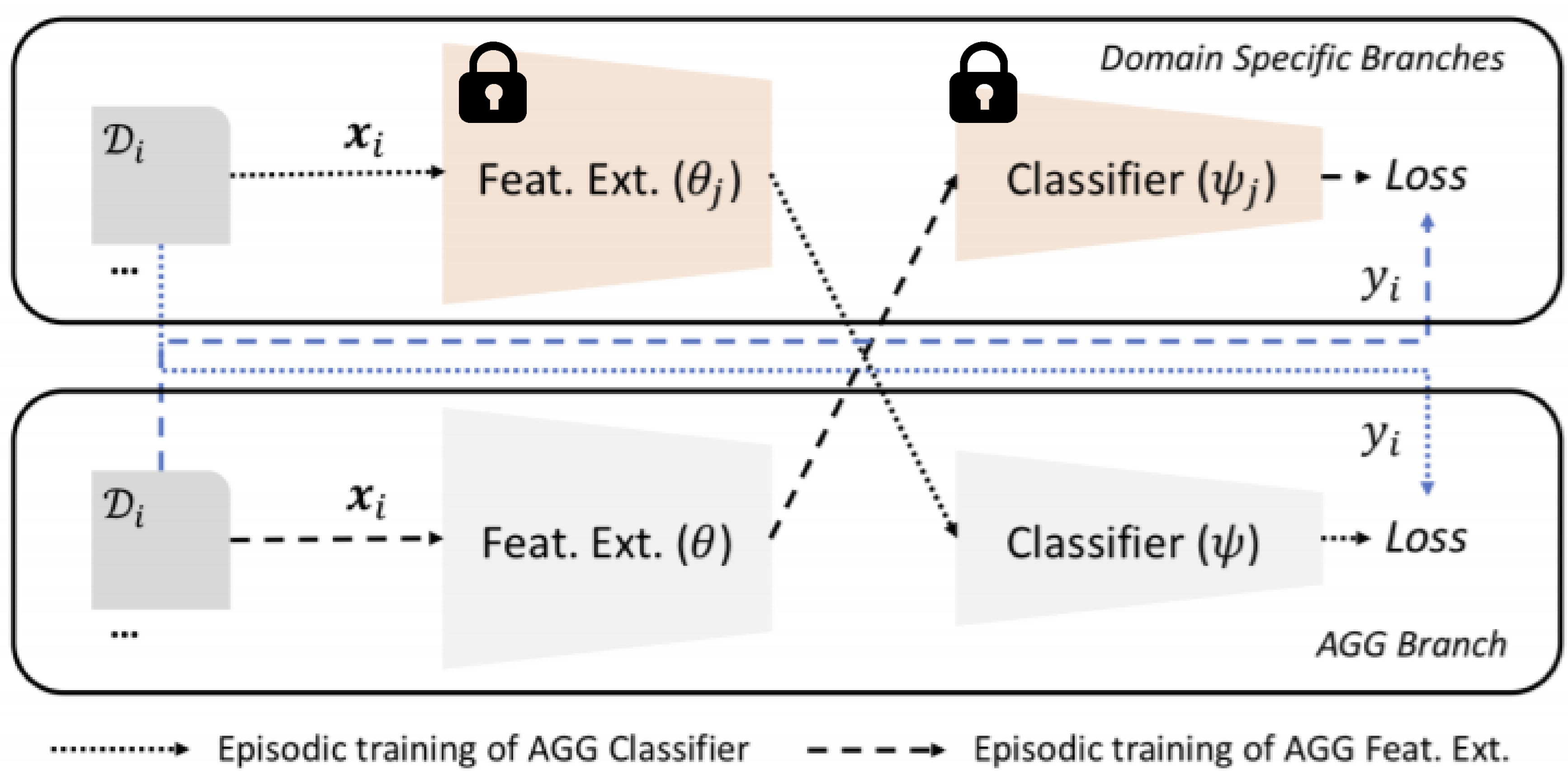


# Episodic Training (cont'd)

- Random sample two domains, e.g., Photo and Cartoon

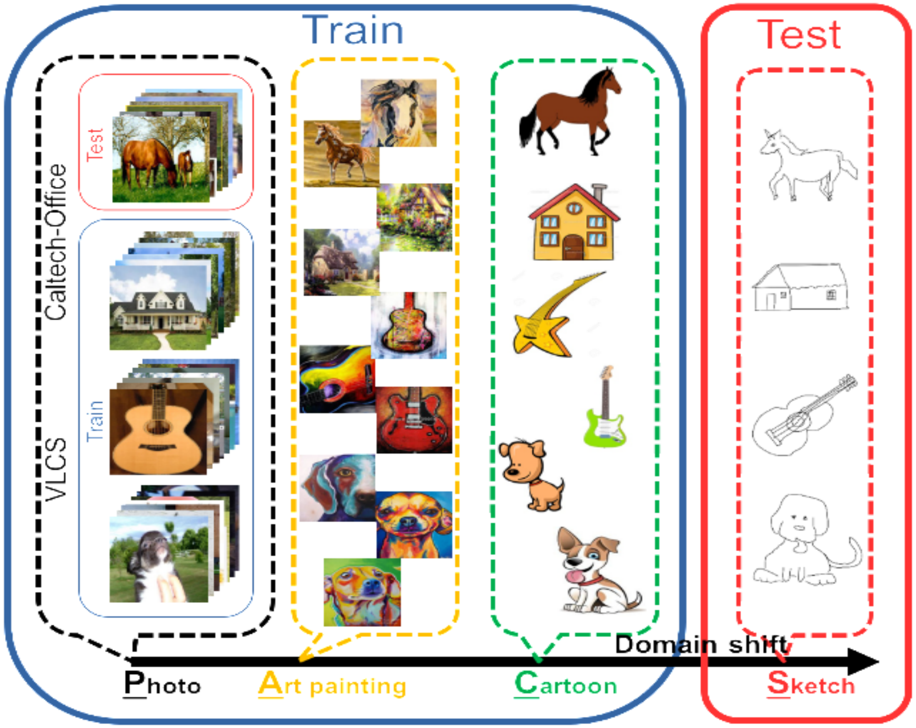


# Episodic Training (cont'd)



# Experiments

- Input: Images and labels from multiple source domains
- Output: A well-generalized model for unseen target domains




$D_S$ = {Photo, Painting, Cartoon}

$D_T$ = {Sketch}

# Experiments (cont'd)

- Domain Generalized Classification

| Source | Target | DICA [26] | LRE-SVM [38] | D-MTAE [12] | CCSA [25] | MMD-AAE [20] | DANN[11] | MLDG [18] | CrossGrad [32] | MetaReg [1] | AGG | Epi-FCR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0,1,2,3 | 4 | 61.5 | 75.8 | 78.0 | 75.8 | **79.1** | 75.0 | 70.7 | 71.6 | 74.2 | 73.1 | 76.9 |
| 0,1,2,4 | 3 | 72.5 | 86.9 | 92.3 | 92.3 | 94.5 | 94.1 | 93.6 | 93.8 | 94.0 | 94.2 | **94.8** |
| 0,1,3,4 | 2 | 74.7 | 84.5 | 91.2 | 94.5 | 95.6 | 97.3 | 97.5 | 95.7 | 96.9 | 95.7 | **99.0** |
| 0,2,3,4 | 1 | 67.0 | 83.4 | 90.1 | 91.2 | 93.4 | 95.4 | 95.4 | 94.2 | 97.0 | 95.7 | **98.0** |
| 1,2,3,4 | 0 | 71.4 | 92.3 | 93.4 | **96.7** | **96.7** | 95.7 | 93.6 | 94.0 | 94.7 | 94.4 | 96.3 |
| Ave. | | 69.4 | 84.6 | 87.0 | 90.1 | 91.9 | 91.5 | 90.2 | 89.9 | 91.4 | 90.6 | **93.0** |

Table 1: Cross-view action recognition results (accuracy. %) on IXMAS dataset. Best result in bold.

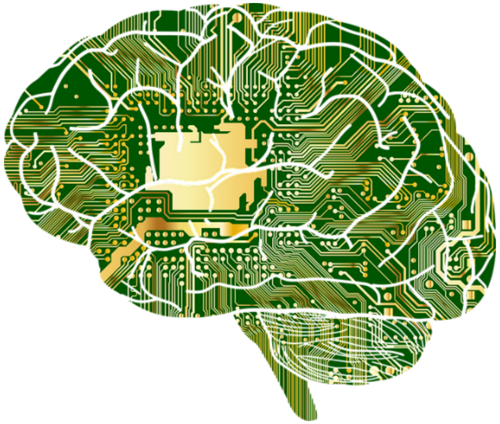
| Source | Target | DICA [26] | LRE-SVM [38] | D-MTAE [12] | CCSA [25] | MMD-AAE[20] | DANN [11] | MLDG [18] | CrossGrad [32] | MetaReg [1] | AGG | Epi-FCR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L,C,S | V | 63.7 | 60.6 | 63.9 | 67.1 | **67.7** | 66.4 | **67.7** | 65.5 | 65.0 | 65.4 | 67.1 |
| V,C,S | L | 58.2 | 59.7 | 60.1 | 62.1 | 62.6 | 64.0 | 61.3 | 60.0 | 60.2 | 60.6 | **64.3** |
| V,L,S | C | 79.7 | 88.1 | 89.1 | 92.3 | **94.4** | 92.6 | **94.4** | 92.0 | 92.3 | 93.1 | 94.1 |
| V,L,C | S | 61.0 | 54.9 | 61.3 | 59.1 | 64.4 | 63.6 | **65.9** | 64.7 | 64.2 | 65.8 | **65.9** |
| Ave. | | 65.7 | 65.8 | 68.6 | 70.2 | 72.3 | 71.7 | 72.3 | 70.5 | 70.4 | 71.2 | **72.9** |

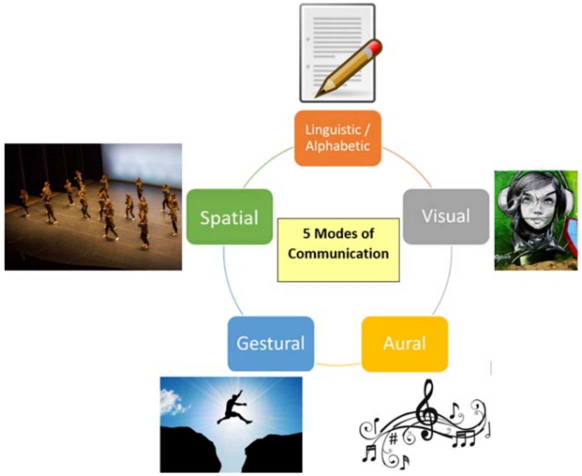
Table 2: Cross-dataset object recognition results (accuracy. %) on VLCS benchmark. Best in bold.

Zhang et al.: Episodic training for domain generalization. In ICCV (2019)

# What to Be Covered Today…

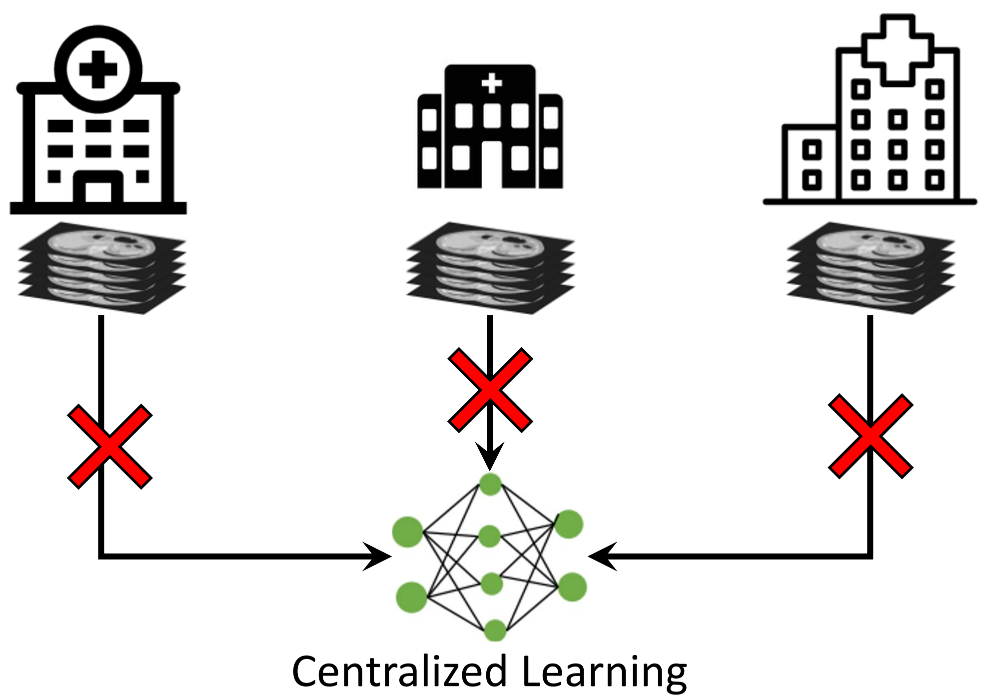
- **Additional Topics in DLCV**
  - Continual Learning
  - Meta Learning
  - Domain Generalization
  - Federated Learning
- **Experience Sharing**
  - Tim Chou (MS, GICE, NTU 2023), AI SW Engineer, NVIDIA





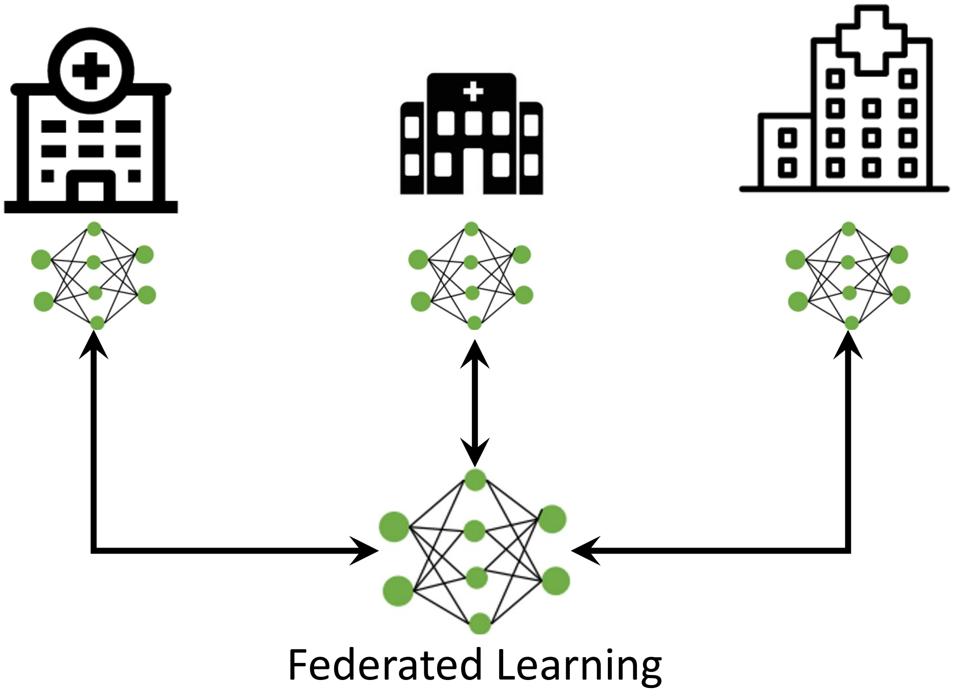

# Why Federated Learning?

- Data privacy issue becomes a growing concern in modern AI services
- Regulations like CCPA (California) or GDPR (Europe) restrict data transmission across different data sources



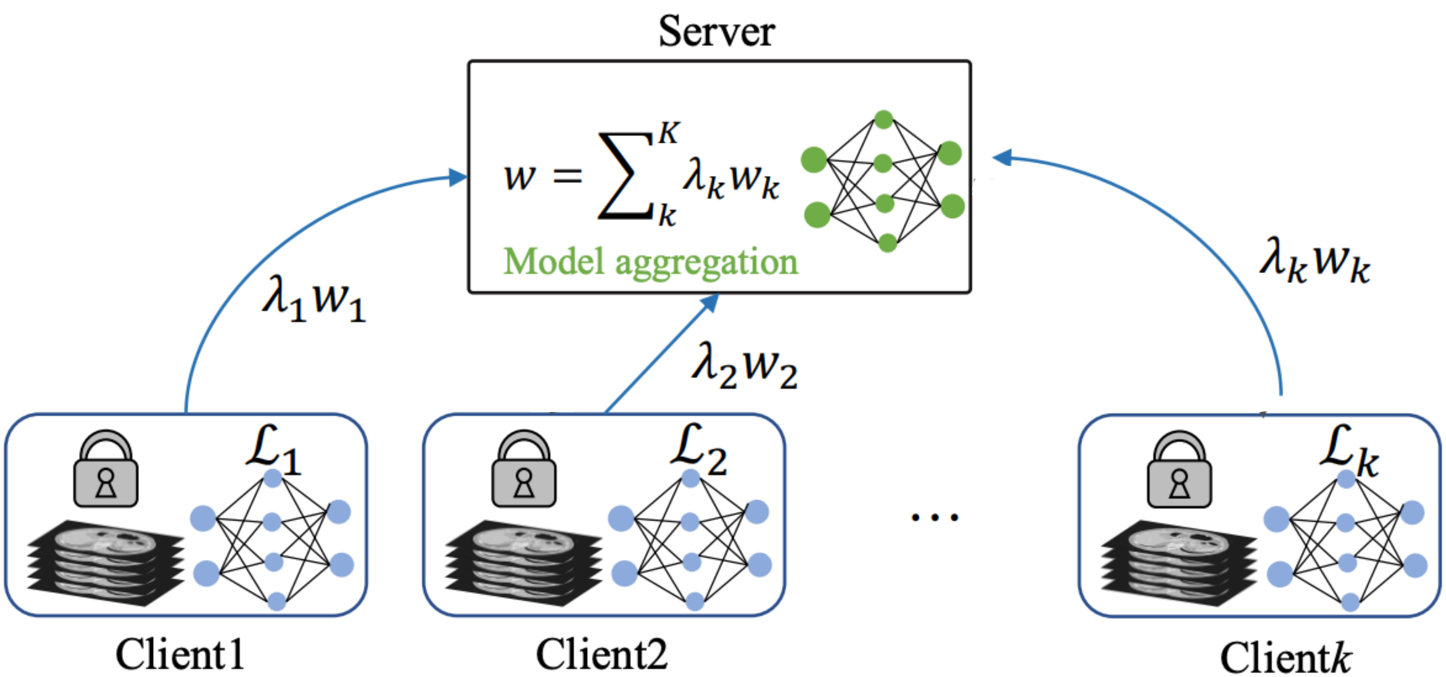
# Federated Learning

- **Collaborative learning** without centralizing data
- Share **model weights** instead of **raw data (or features)**!
- Model training occurs locally at each participant/client



# Federated Learning (cont'd)

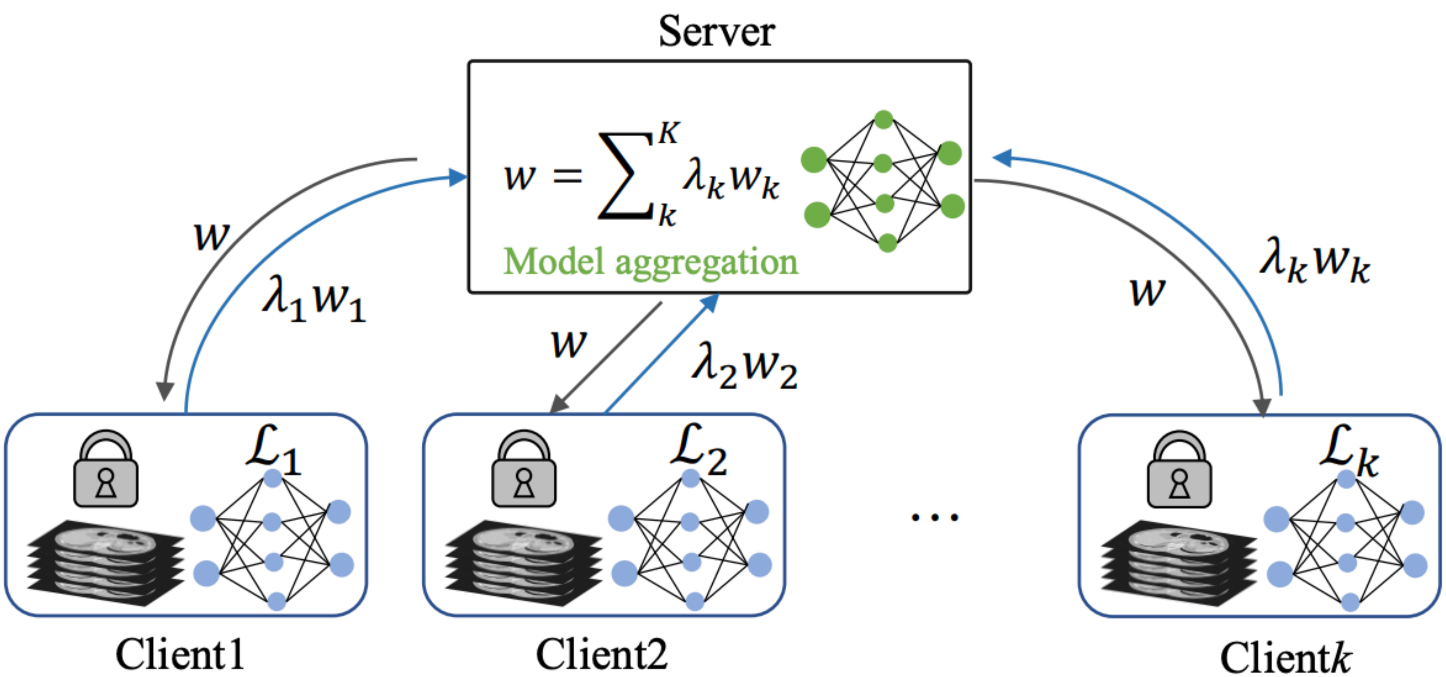
- Training models collaboratively without sharing the raw data
- FedAvg:
  - Local client training using private data --> Server aggregation (i.e., averaging)




Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS 2017

# Federated Learning (cont'd)

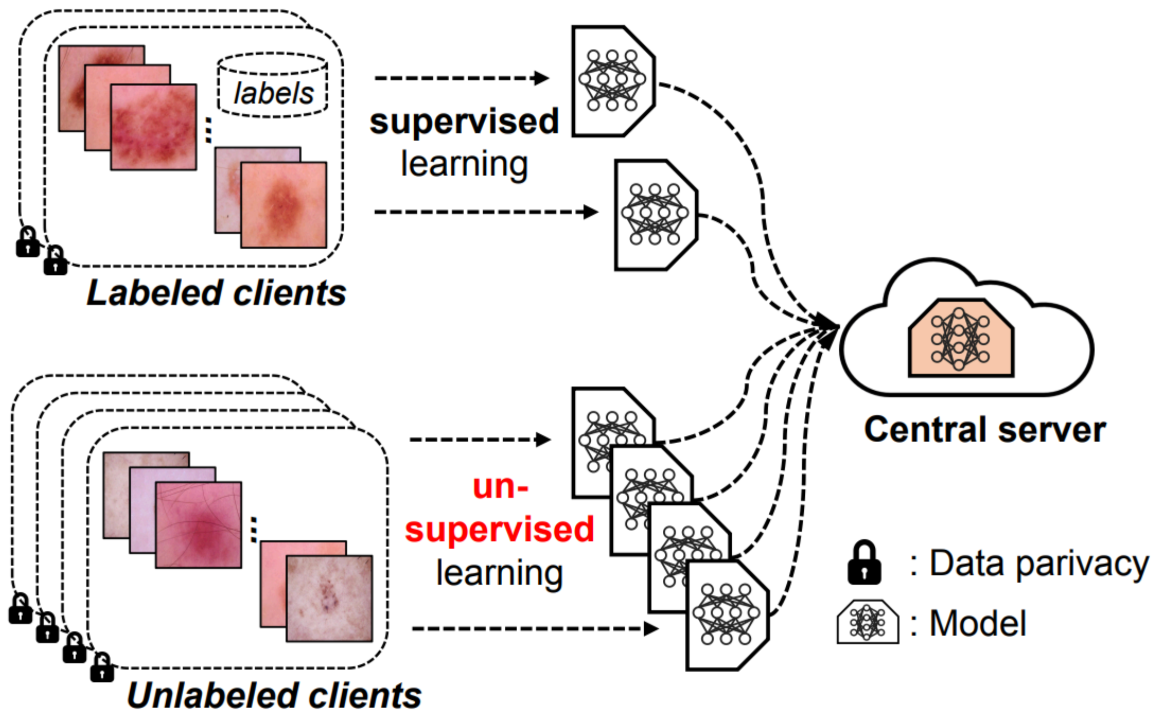
- Training models collaboratively without sharing the raw data
- FedAvg:
  - Local client training using private data --> Server aggregation (Averaging) --> Broadcast to clients (then iterate)




Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS 2017

# Extension of Federated Learning

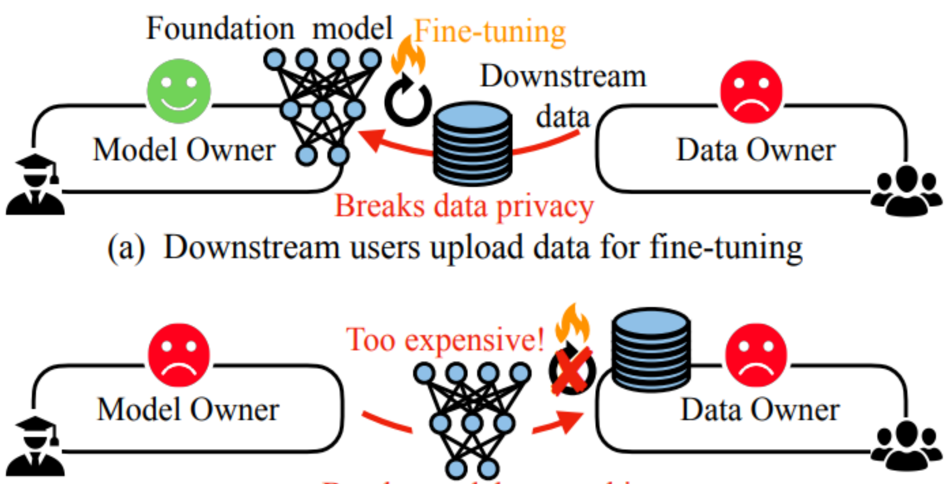
- Semi-Supervised FL
  - Some labeled clients, and other unlabeled clients




Federated Semi-supervised Medical Image Classification via Inter-client Relation Matching, MICCAI 2021

# Extension of Federated Learning (cont'd)

- Offsite-Tuning: Transfer Learning without Full Model (MIT, arxiv., 2023)
  - Sharing models across clients results in privacy concern
  - Model owners (Big Tech) don't want to share model weights
  - Users don't want to share data with personal or sensitive information
    ⟹ Cannot fine-tune to obtain full power of foundation model



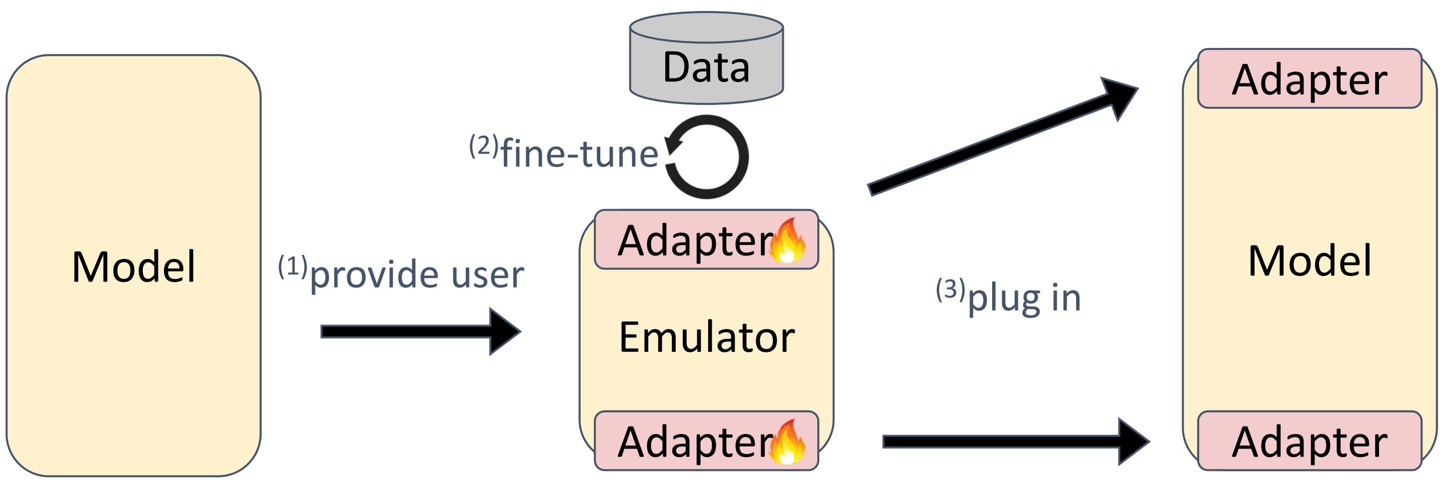

(a) Downstream users upload data for fine-tuning

(b) Model owner releases the model to downstream users

https://arxiv.org/abs/2302.04870

# Extension of Federated Learning (cont'd)

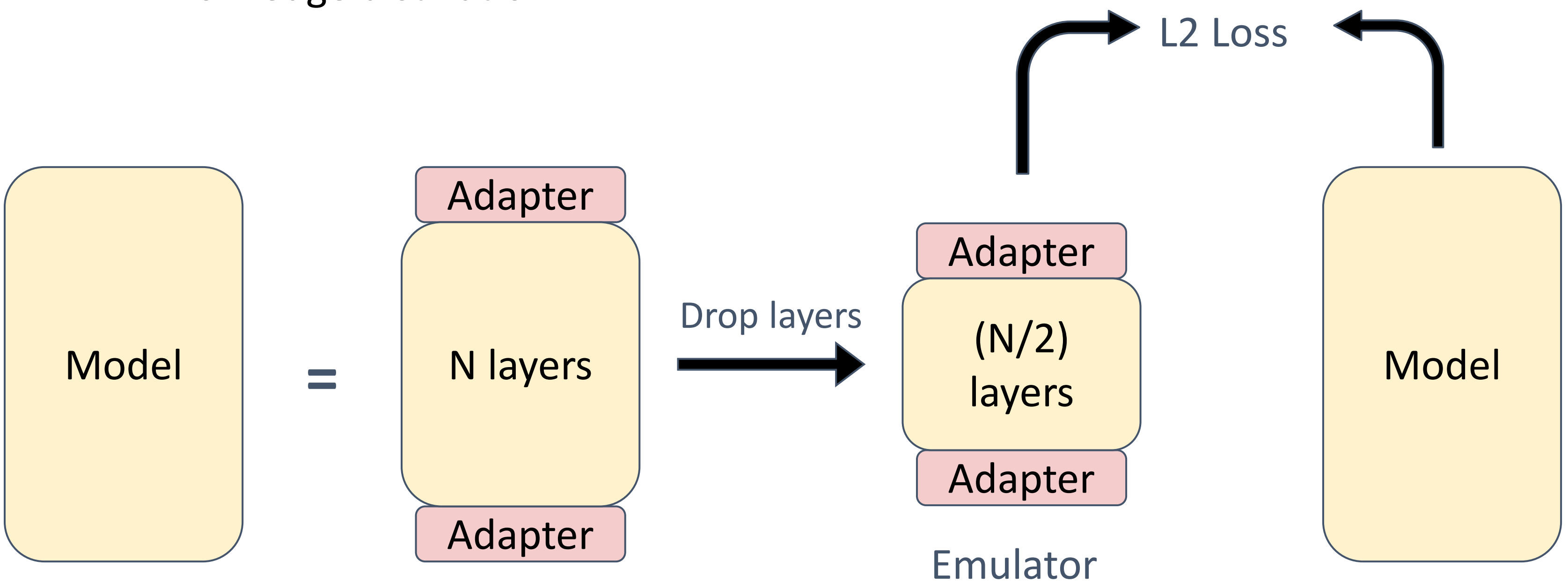
- Offsite-Tuning: Transfer Learning without Full Model (MIT, arxiv., 2023)
- Proposed idea
  - Smaller version of original model (efficiency for transfer and fine-tuning)
  - Less powerful (business consideration)
  - Trainable adapters that can transfer to model owner and "plug in" model




https://arxiv.org/abs/2302.04870

# Extension of Federated Learning (cont'd)

- Offsite-Tuning: Transfer Learning without Full Model (MIT, arxiv., 2023)
- How to construct emulators?
  - Keep the first 2 and last 2 layers of original model as adapters
  - Uniformly drop rest layers (e.g., every 2 layers)
  - Knowledge distillation




https://arxiv.org/abs/2302.04870

# Extension of Federated Learning (cont'd)

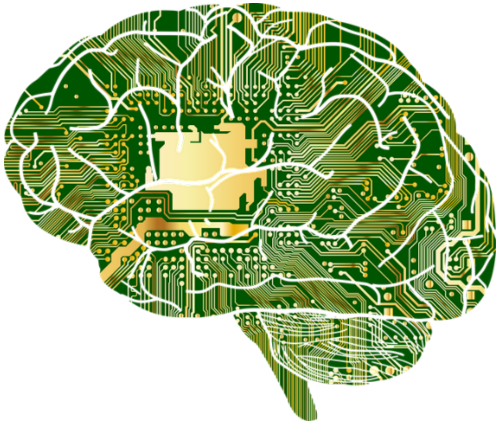
- Offsite-Tuning: Transfer Learning without Full Model (MIT, arxiv., 2023)
- Experiments
  - Accuracy of two LLMs on different QA benchmarks (higher is better)
  - ZS: zero shot, FT: full fine-tune,
    OT Emulator: adapters on emulator, OT Plug-in: adapters on original model

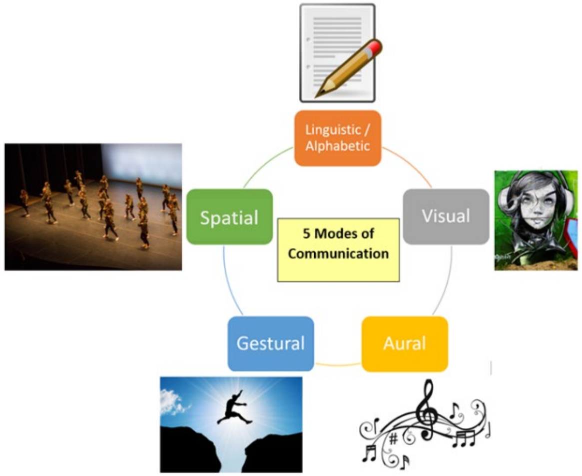
| *Setting* | OpenBookQA | PIQA | ARC-E | ARC-C | HellaSwag | SciQ | WebQs | RACE |
|---|---|---|---|---|---|---|---|---|
| | | | | GPT2-XL (2-16-2 Distill) | | | | |
| Full ZS | 23.0% | 70.9% | 58.2% | 25.1% | 40.0% | 83.2% | 1.5% | 33.0% |
| Emulator ZS | 18.8% | 67.7% | 53.2% | 20.8% | 33.5% | 77.0% | 0.2% | 30.0% |
| FT | 30.0% | 73.2% | 62.9% | 30.0% | 40.7% | 92.5% | 26.4% | 43.2% |
| OT Emulator | 24.0% | 70.3% | 58.2% | 23.9% | 35.8% | 92.7% | 18.9% | 39.4% |
| OT Plug-in | 28.2% | 73.6% | 61.4% | 28.5% | 41.6% | 93.2% | 19.9% | 39.9% |
| | | | | OPT-1.3B (2-8-2 Distill) | | | | |
| Full ZS | 23.4% | 71.6% | 56.9% | 23.5% | 41.5% | 84.4% | 4.6% | 34.2% |
| Emulator ZS | 19.4% | 68.7% | 53.9% | 21.5% | 35.1% | 80.9% | 1.3% | 33.0% |
| FT | 31.4% | 75.2% | 61.3% | 27.7% | 42.7% | 92.5% | 31.2% | 37.0% |
| OT Emulator | 24.8% | 71.6% | 58.1% | 26.1% | 37.0% | 92.2% | 24.3% | 38.6% |
| OT Plug-in | 29.0% | 74.5% | 59.4% | 27.8% | 43.3% | 92.9% | 26.2% | 38.9% |

https://arxiv.org/abs/2302.04870

# What to Be Covered Today…

- Additional Topics in DLCV
  - Continual Learning
  - Meta Learning
  - Domain Generalization
  - Federated Learning
- **Experience Sharing**
  - Tim Chou (MS, GICE, NTU 2023), AI SW Engineer, NVIDIA






# What We've Covered This Semester

- **MLP**: Linear to Non-linear Classification
- **CNN**: Classification, Segmentation, Detection, and SSL
- **Generative Model**: AE/VAE, GAN, Diffusion Model & Personalization
- **Transformer**: Learning from Sequential Data
- **Vision-Language Models**: Pre-training & Finetuning, PEFT
- **3D Vision**: Point Cloud, NeRF, 3DGS
- **More Topics**: Continual learning, Meta Learning,<br>Domain Generalization, Fed Learning
- **Guest Lectures**: 2 academic + 1 career planning talks/sharing
- **Your Feedback Is Appreciated! ☺**
    - 期末教學意見調查
    - https://if163.aca.ntu.edu.tw/eportfolio/

# Good Luck with the Final Project & All Your Finals!

See you all on Dec. 26th
(snack provided during final presentation)