

Deep Learning for Computer Vision

Fall 2022

<https://cool.ntu.edu.tw/courses/189345> (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

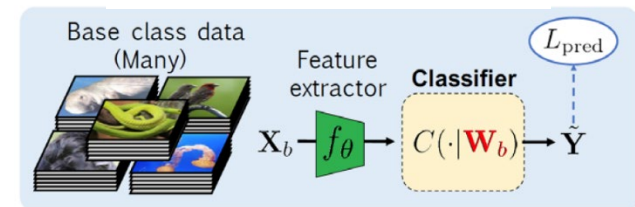
What to Cover Today...

- Recap on Transformer
- Vision & Language
 - Image Captioning
 - Text-to-Image Synthesis
- Meta-Learning
 - Meta-Learning for Few-Shot Learning (FSL)
 - Advanced Issues in Learning from Small Data
 - Few-Shot Segmentation & Detection (if time permits)

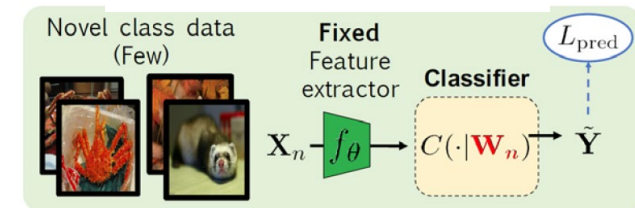


“a corgi wearing a bow tie and a birthday hat”

Meta-Training Stage

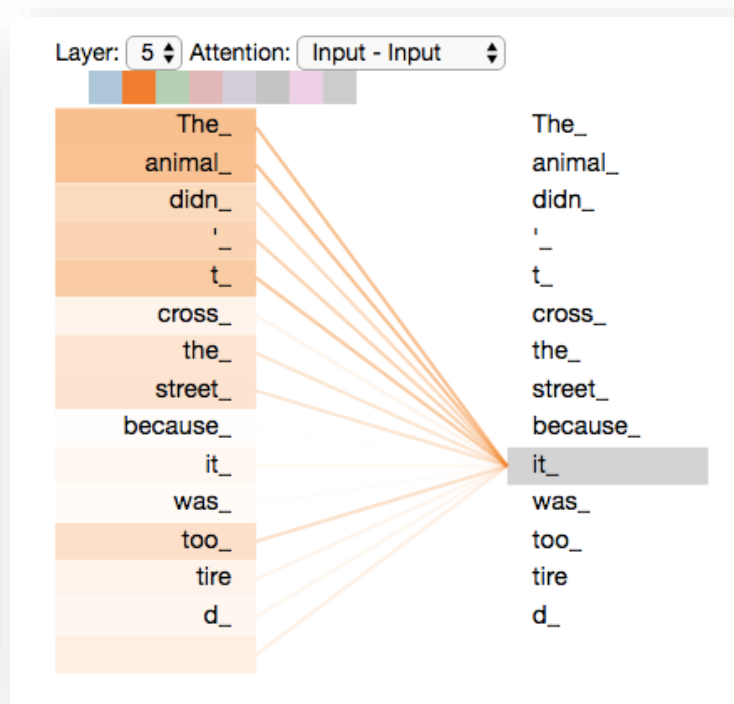
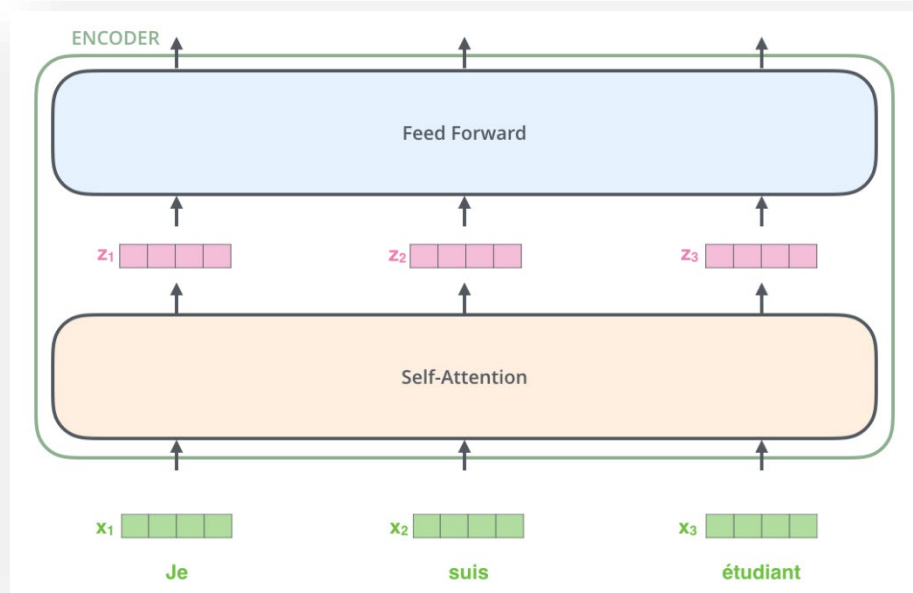
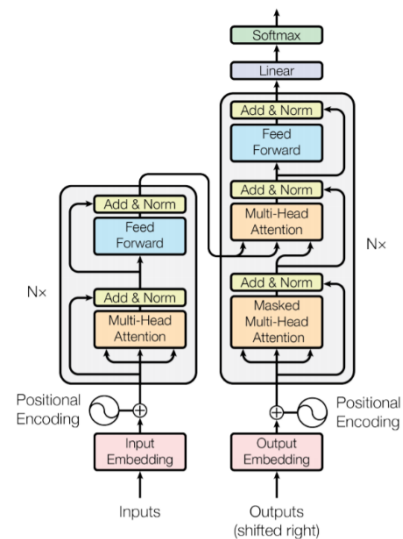


Meta-Testing Stage



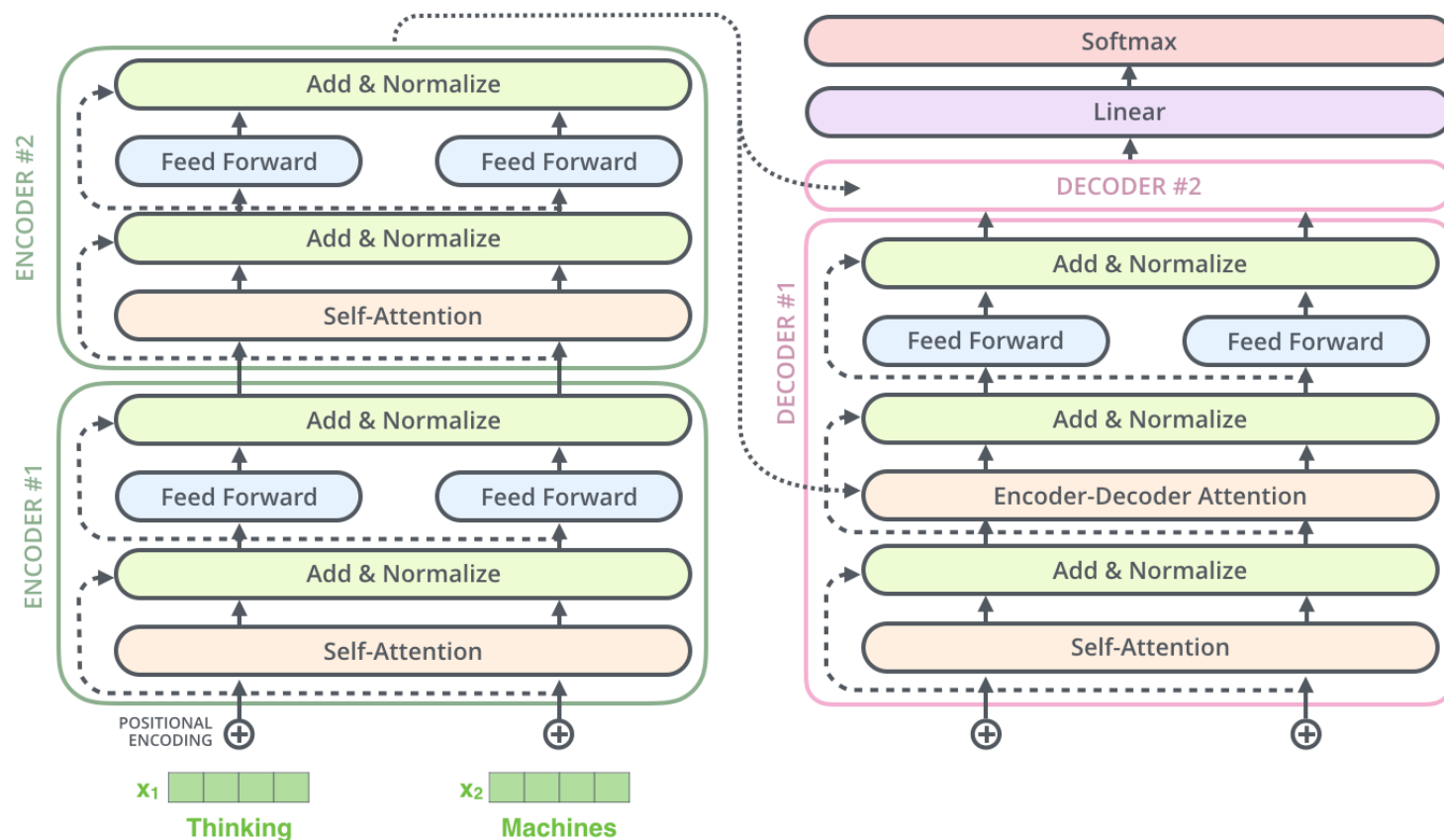
Transformer

- “Attention is all you need”, NeurIPS 2017
- Self-attention for text translation



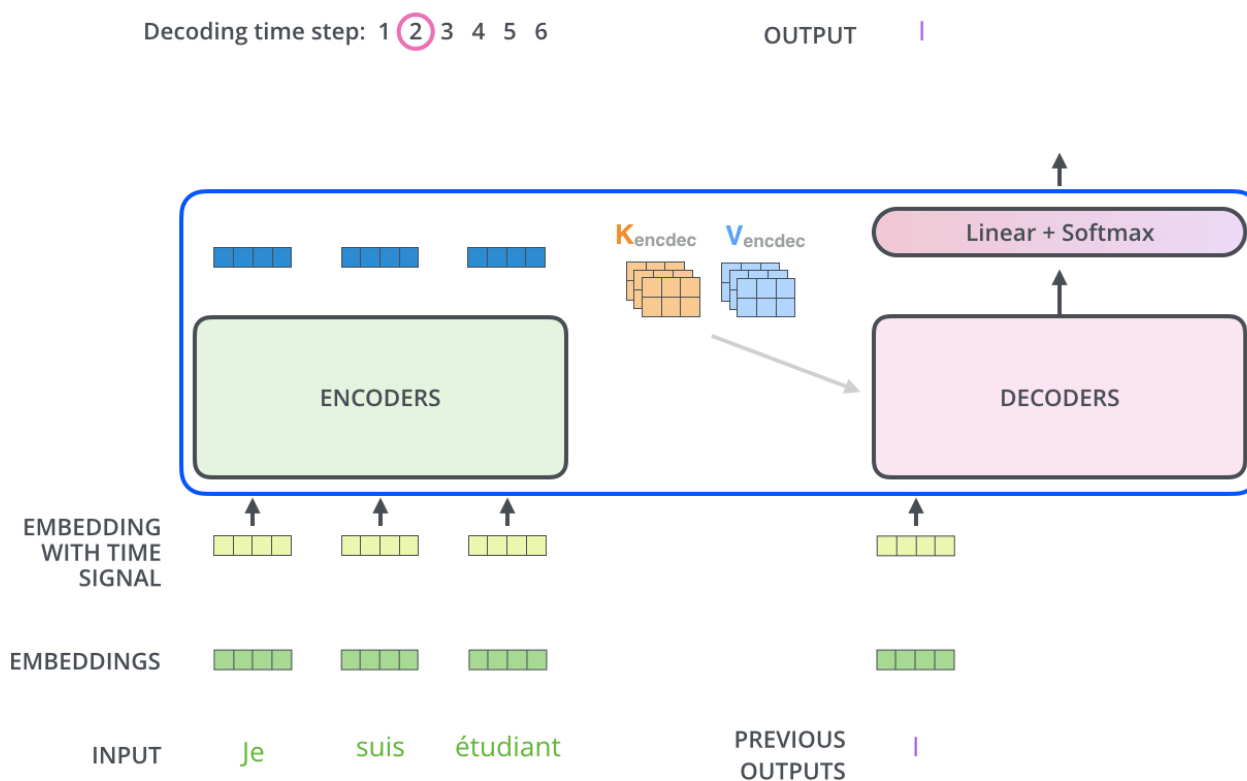
The Decoder in Transformer

- Design similar to that of encoder, except the decoder #1 takes additional inputs (of GT/predicted word embeddings).



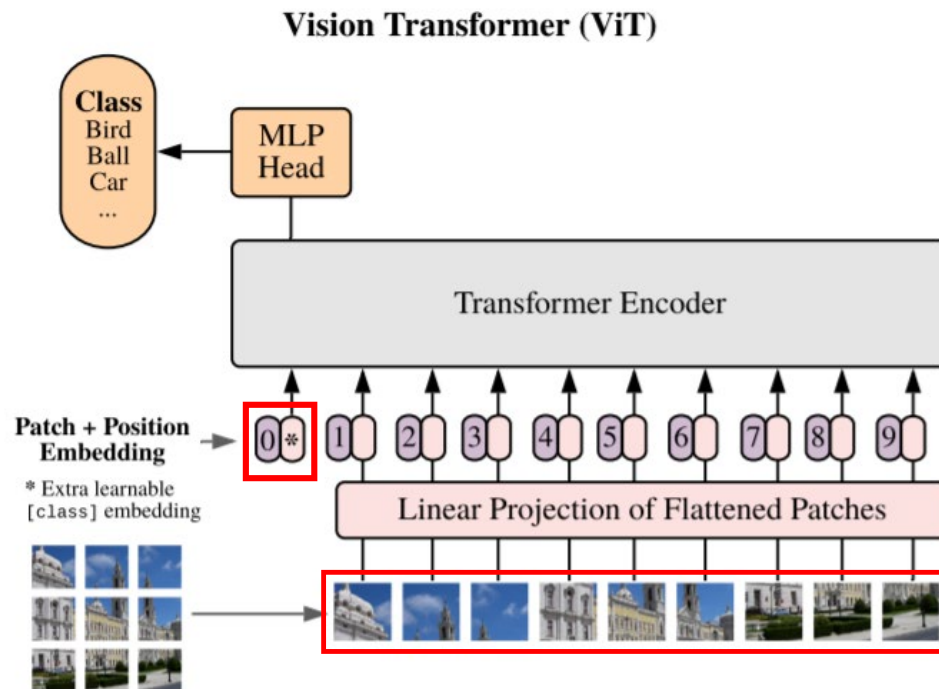
The Decoder in Transformer

- Design similar to that of encoder, except the 1st decoder takes additional inputs (of predicted word embeddings).



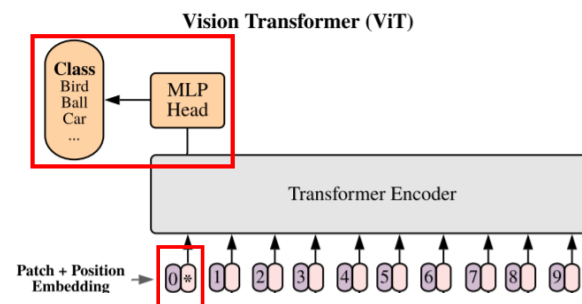
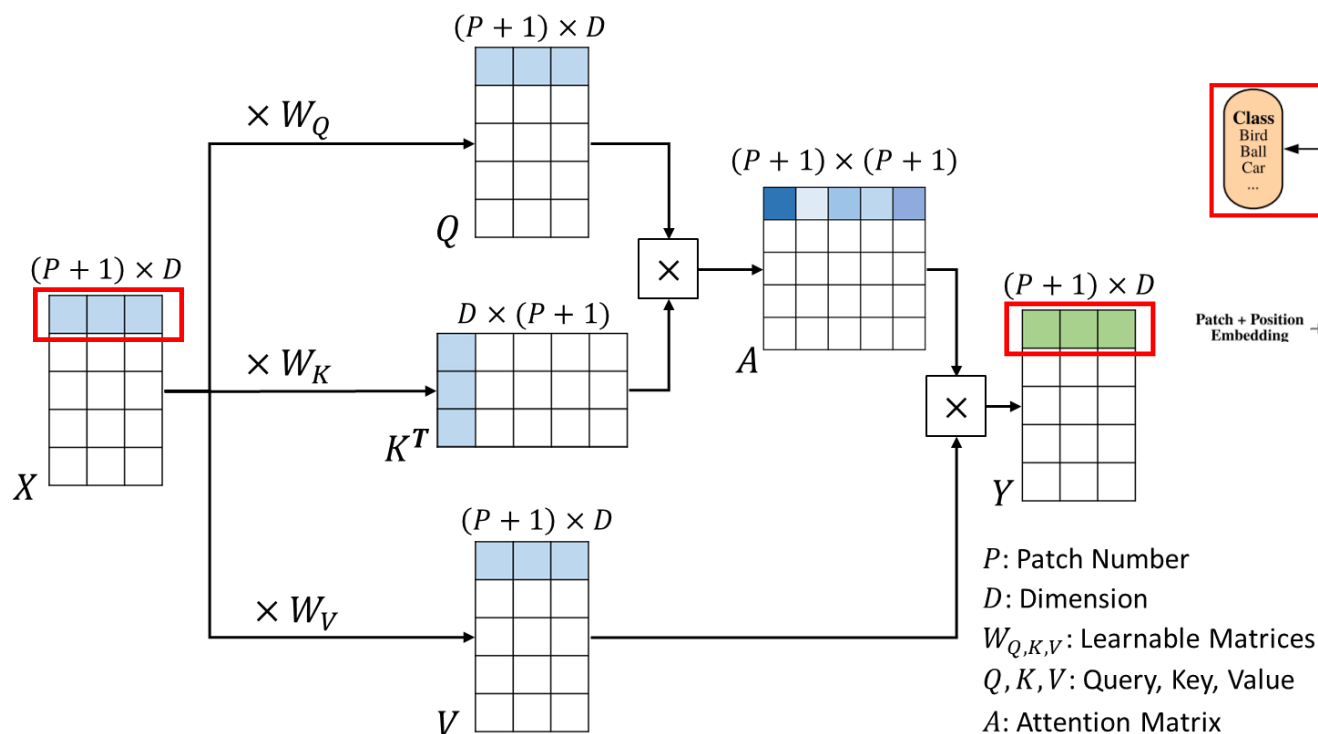
Vision Transformer

- “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR, 2021. (Google Research)
- Partition the input image into a **patch sequence**
- An additional **token** (*) is appended to perform attention on patches
- Both the “*” token and positional embeddings (denoted by 0, 1, 2 ...) are **trainable vectors**



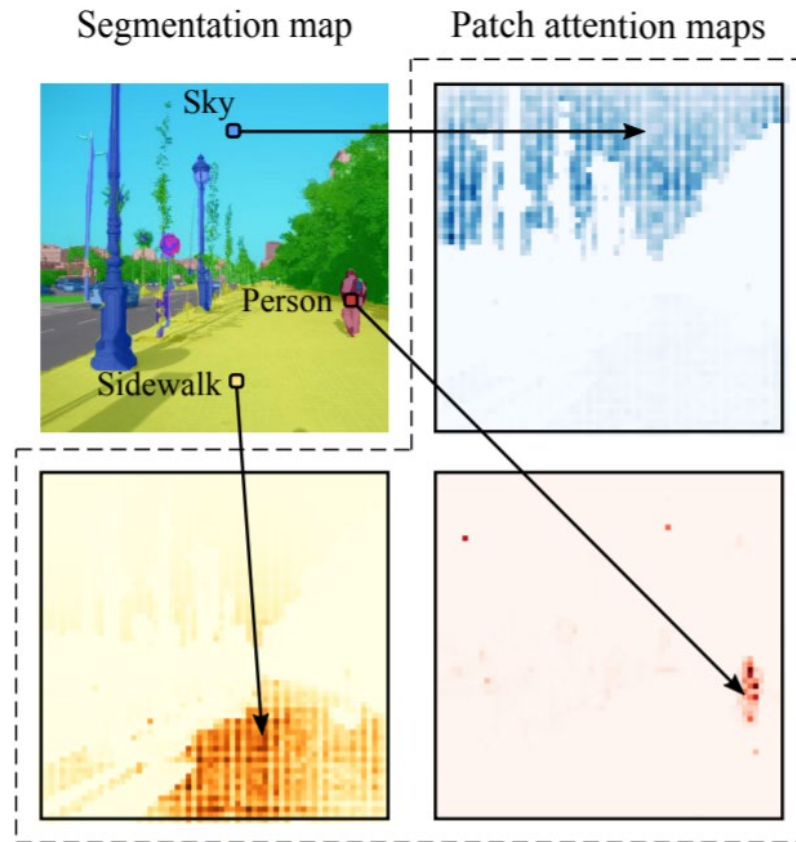
Query-Key-Value Attention in ViT

- In standard vision transformer, we only take the **first output token** of the output sequence (the **first row** of Y) for classification purposes
- This corresponds to the output when **token “0”** serves as query



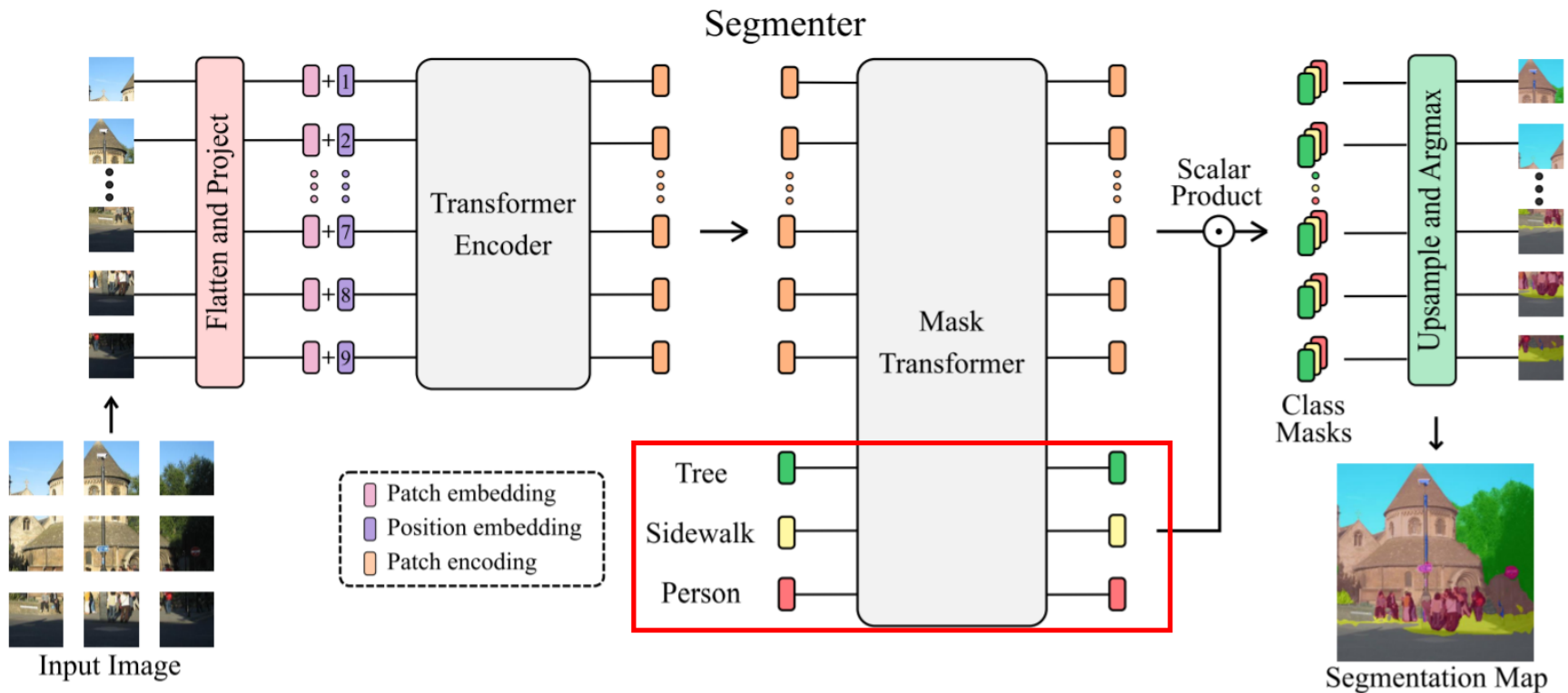
Transformer for Semantic Segmentation

- Segmentation via attention



Transformer for Semantic Segmentation

- Using different class tokens (“Tree”, “Sidewalk”, “Person”, ...) as queries



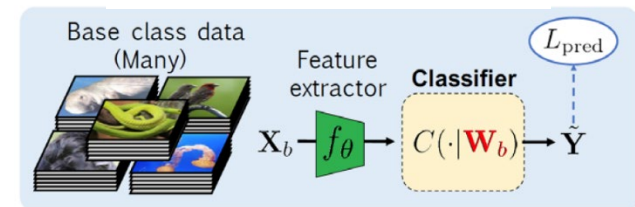
What to Cover Today...

- Recap on Transformer
- **Vision & Language**
 - Image Captioning
 - Text-to-Image Synthesis
- **Meta-Learning**
 - Parametric vs. Non-Parametric Approaches
 - Meta-Learning for Few-Shot Learning
 - Few-Shot Image Segmentation
 - Few-Shot Object Detection

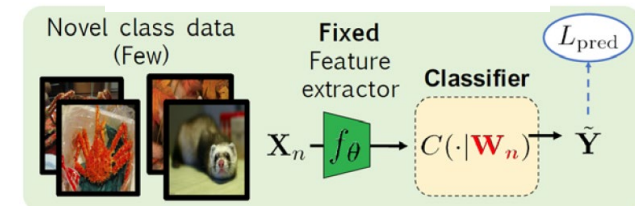


“a corgi wearing a bow tie and a birthday hat”

Meta-Training Stage



Meta-Testing Stage



A picture is worth a thousand words...
Is it that simple?



- Thing
- Airplane
- Flying airplane in blue sky
- A Lufthansa MD-11 cargo plane in blue sky flying over mountainous terrain

Vision + Language → ?

- Image Captioning
- Image Manipulation/Completion
- Composed Image Retrieval
- Visual Question Answering (VQA)
and many more...

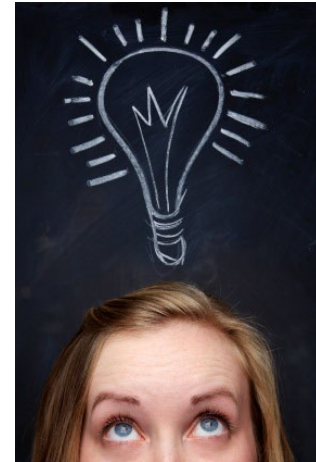
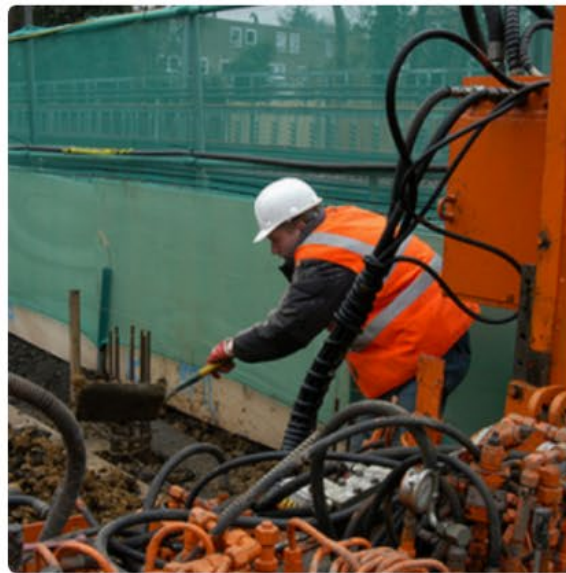


Image Captioning



Applications: semantics understanding, image-text retrieval, medical AI, etc.

Image Captioning (cont'd)

- Training a captioning model requires strong supervision
 - A large amount of image-caption data pairs
- Image captioning in the wild:
 - Describing images with novel content during inference
 - For example, COCO dataset has 80 object categories.
How to generalize captioning models to Open Image (w/ 600 classes)?

COCO (80 classes)



Two pug **dogs** sitting on a **bench** at the beach.



A **child** is sitting on a **couch** and holding an **umbrella**.

Open Images (600 classes)



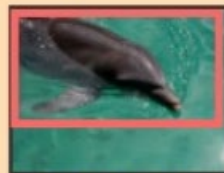
goat



artichoke



accordion



dolphin



waffle



balloon

Image Captioning *in the Wild*

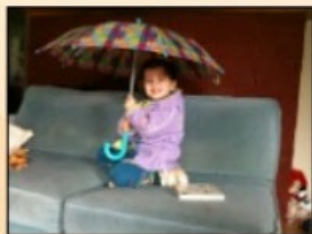
- **Novel Object Captioning (NOC)**

- Training with **captioned** and **uncaptioned** data
 - captioned** data: labeled image data with captions (e.g., COCO)
 - uncaptioned** data: only labels of novel classes available (e.g., Open Images)
- Will come back to this task later

COCO (80 classes)



Two pug **dogs** sitting on a **bench** at the beach.



A **child** is sitting on a **couch** and holding an **umbrella**.

We have captioning data

Open Images (600 classes)



goat



artichoke



accordion



dolphin



waffle

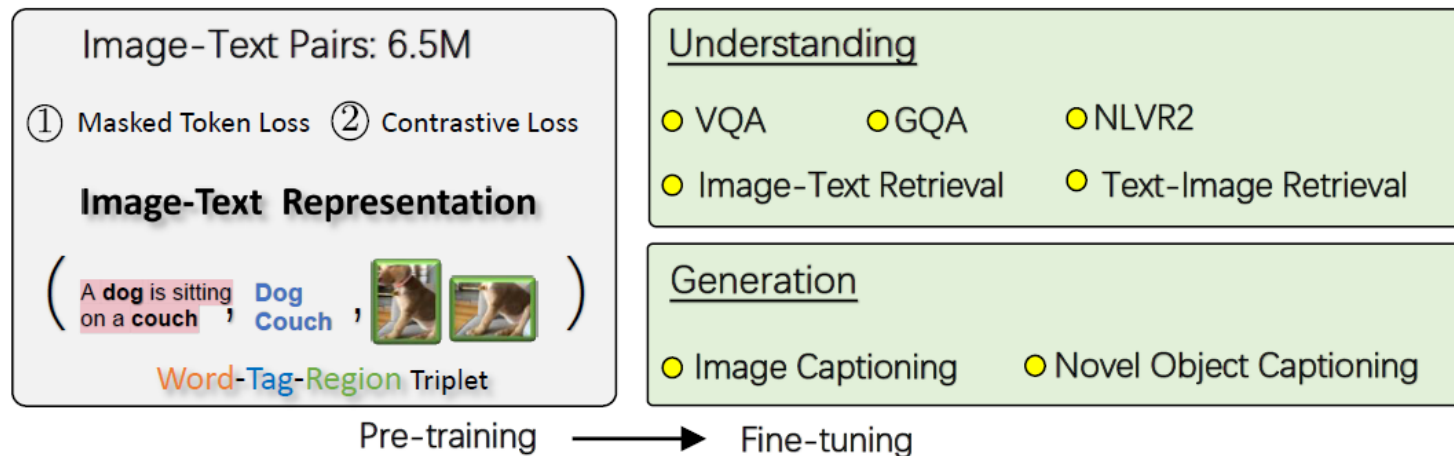


balloon

Data with labels for novel objects but w/o captions

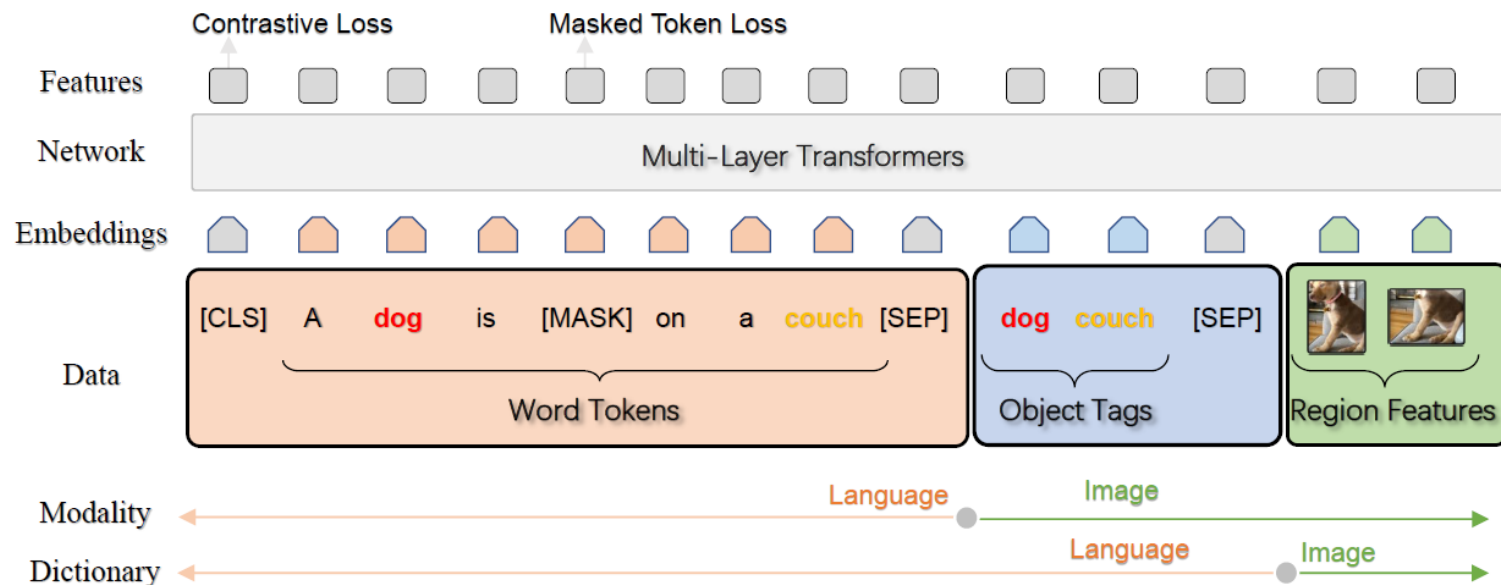
Beyond Image Captioning: Unified Vision & Language Model

- **Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks (ECCV'20)**
 - Training data:
triplets of **caption-tag-region**
 - Objectives:
 1. Masked token loss for **words** & **tags**
 2. Contrastive loss **tags** and others
 - Fine-tuning:
5 vision & language tasks (VQA, image-text retrieval, image captioning, NOC, etc.)



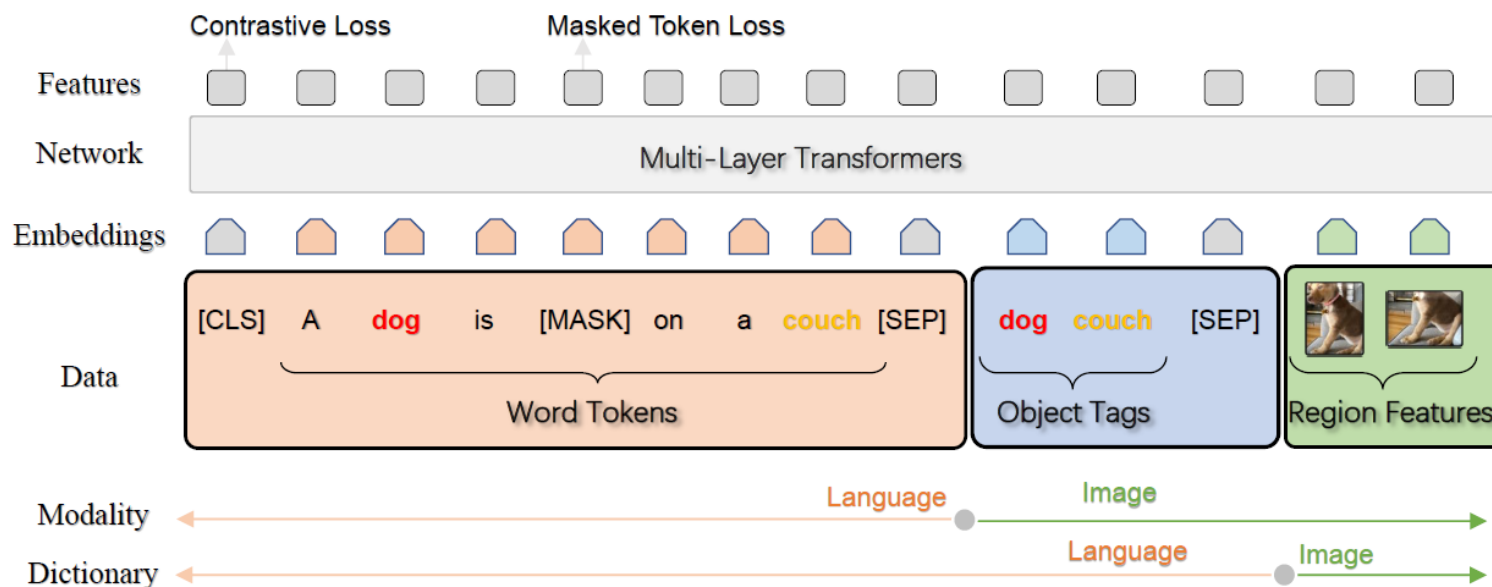
Semantics-Aligned Pre-training for V+L Tasks

- **Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks (ECCV'20)**
 - Training:
 - Inputs: triplets of **caption-tag-region**
 - Objectives: Masked token loss for **words** & **tags** + Contrastive loss **tags** and others
 - Fine-tuning:
5 vision & language tasks (image captioning, NOC, VQA, image-text retrieval, etc.)



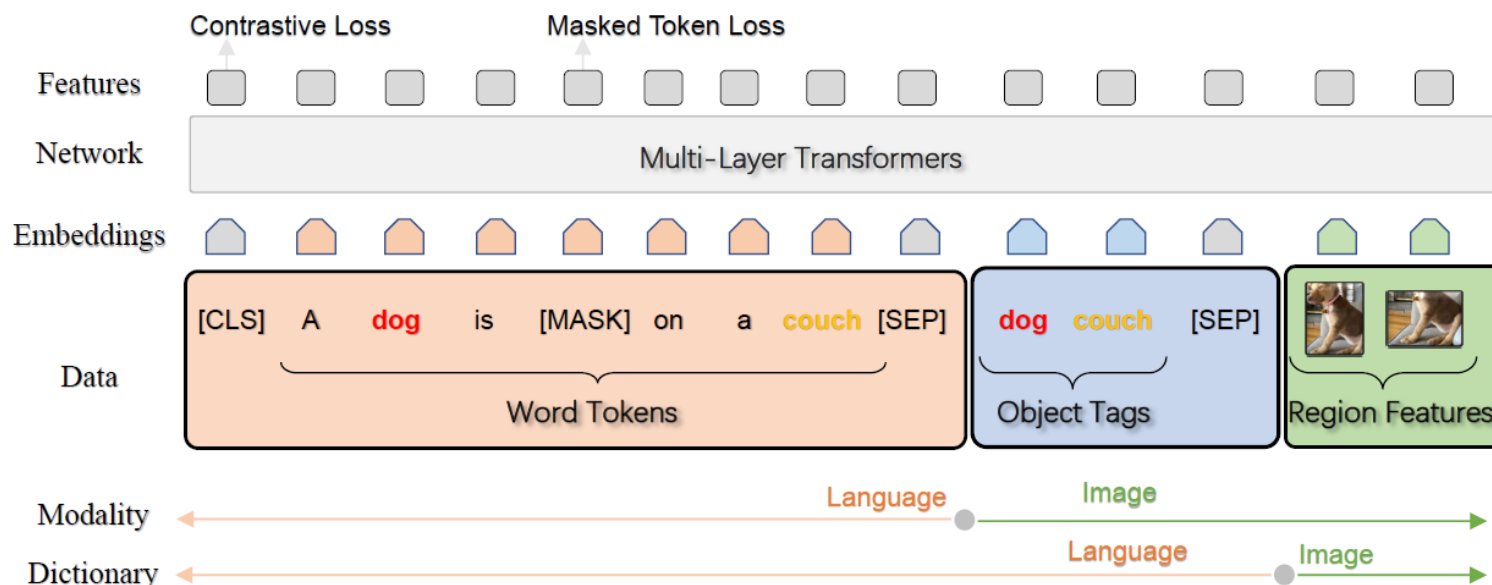
Semantics-Aligned Pre-training for V+L Tasks (cont'd)

- **Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks (ECCV'20)**
 - Training:
 - Inputs: triplets of word-tag-region
 - Objectives: Masked token loss for words & tags + Contrastive loss tags and others
 - Fine-tuning:
 - 5 vision & language tasks (image captioning, NOC, VQA, image-text retrieval, etc.)

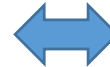


Semantics-Aligned Pre-training for V+L Tasks (cont'd)

- **Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks (ECCV'20)**
 - Fine-tuning:
5 vision & language tasks (image captioning, NOC, VQA, image-text retrieval, etc.)
 - Take **image captioning** as an example
 - Training: triplets of **image regions features** + **object tags** + **captions** as inputs;
caption tokens with full attention on image regions/tags but not the other way around
 - Inference: **image regions**, **tags** and **[CLS]** as inputs,
with **[MASK]** tokens sequentially added/predicted



Holding an apple



or



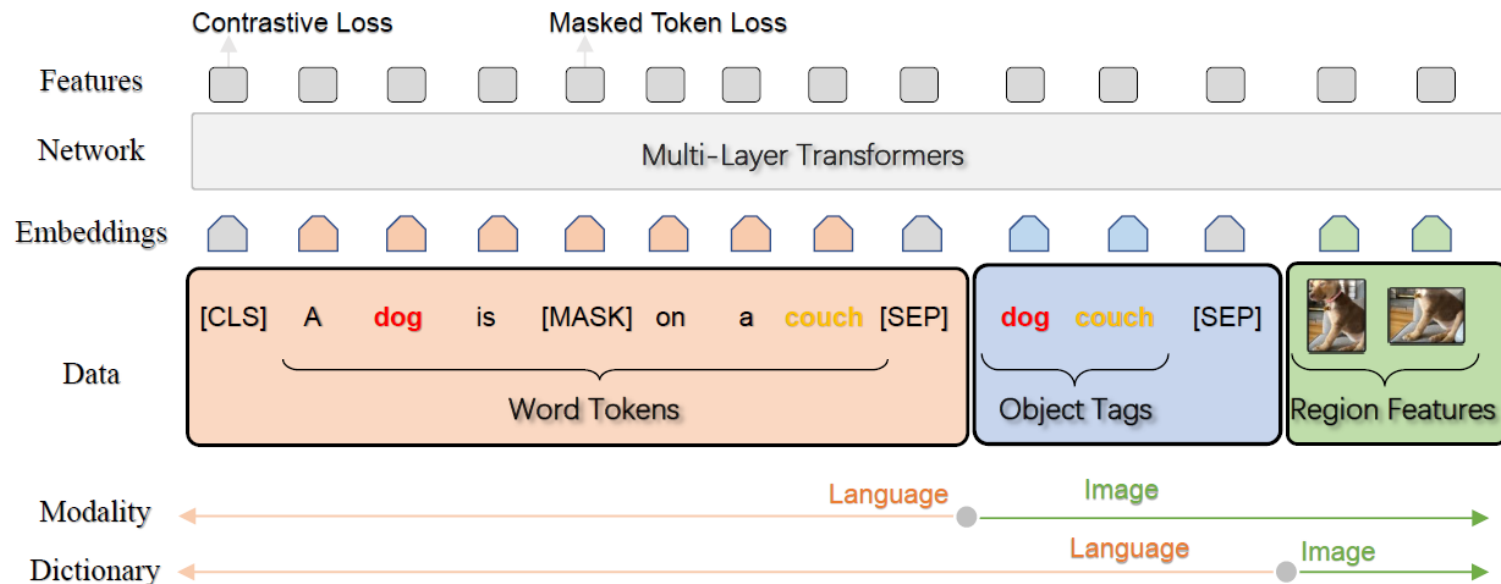
- **Oscar (cont'd)**

- Fine-tuning:

- 5 vision & language tasks (image captioning, NOC, VQA, image-text retrieval, etc.)

- Take **image-text retrieval** as an example

- Training: aligned/mis-aligned **image-text** pairs as positive/negative input pairs, with **[CLS]** for binary classification (1/0)
 - Inference: for either image or text retrieval, calculate classification score of **[CLS]** for the input query

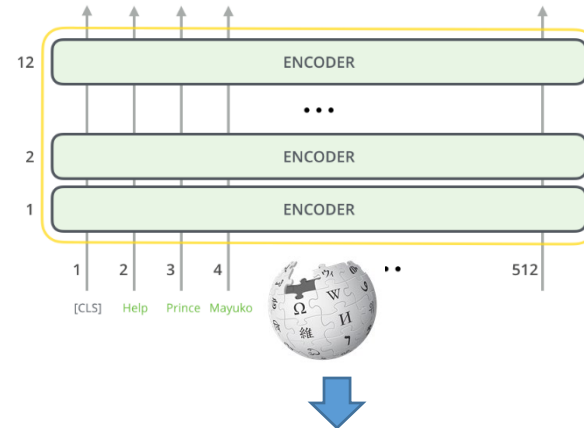


Novel Object Captioning

- **VIVO: Visual Vocabulary Pre-Training for Novel Object Caption Captioning (AAAI'21)**
 - Pre-training a **cross-modality Transformer** for vision & language tasks
 - **Pre-training** really matters, since it's been observed in
 - Computer Vision (e.g., models pre-trained on ImageNet)
 - Natural Language Processing (e.g., BERT pre-trained on Wikipedia)



Object detection,
semantic segmentation, etc.



Question answering,
Sentence classification, etc.

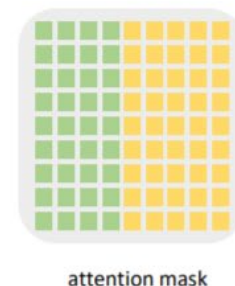


Recent Work on Novel Object Captioning

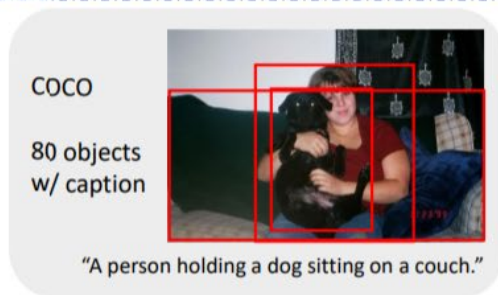
- **VIVO: Visual Vocabulary Pre-Training for Novel Object Caption Captioning**
 - Pre-training: **uncaptioned image data** containing **novel class labels**
 - Fine-tuning: (a limited amount of) **image data** with **class labels** & **descriptions**



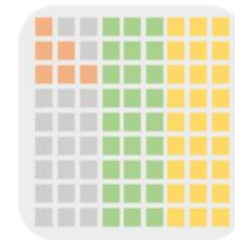
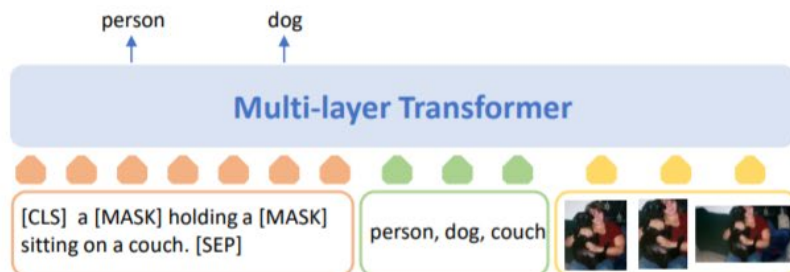
(a) Pre-training: learn visual vocabulary



attention mask



(b) Fine-tuning: learn sentence description



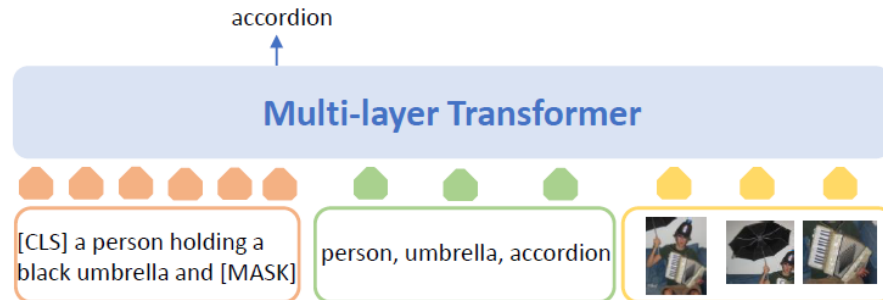
attention mask

Novel Object Captioning (cont'd)

- **VIVO: Visual Vocabulary Pre-Training for Novel Object Caption Captioning**
 - Pre-training: uncaptioned image data containing novel class labels
 - Fine-tuning: (a limited amount of) image data with class labels & descriptions
 - Inference:
 - Inputs: image (with region features & tags) & [CLS]
 - Output: caption



(c) Inference: novel object captioning



A person holding a black umbrella and **accordion**.

Novel Object Captioning (cont'd)

- VIVO: Visual Vocabulary Pre-Training for Novel Object Caption Captioning
 - Properly aligned image and text data for captioning

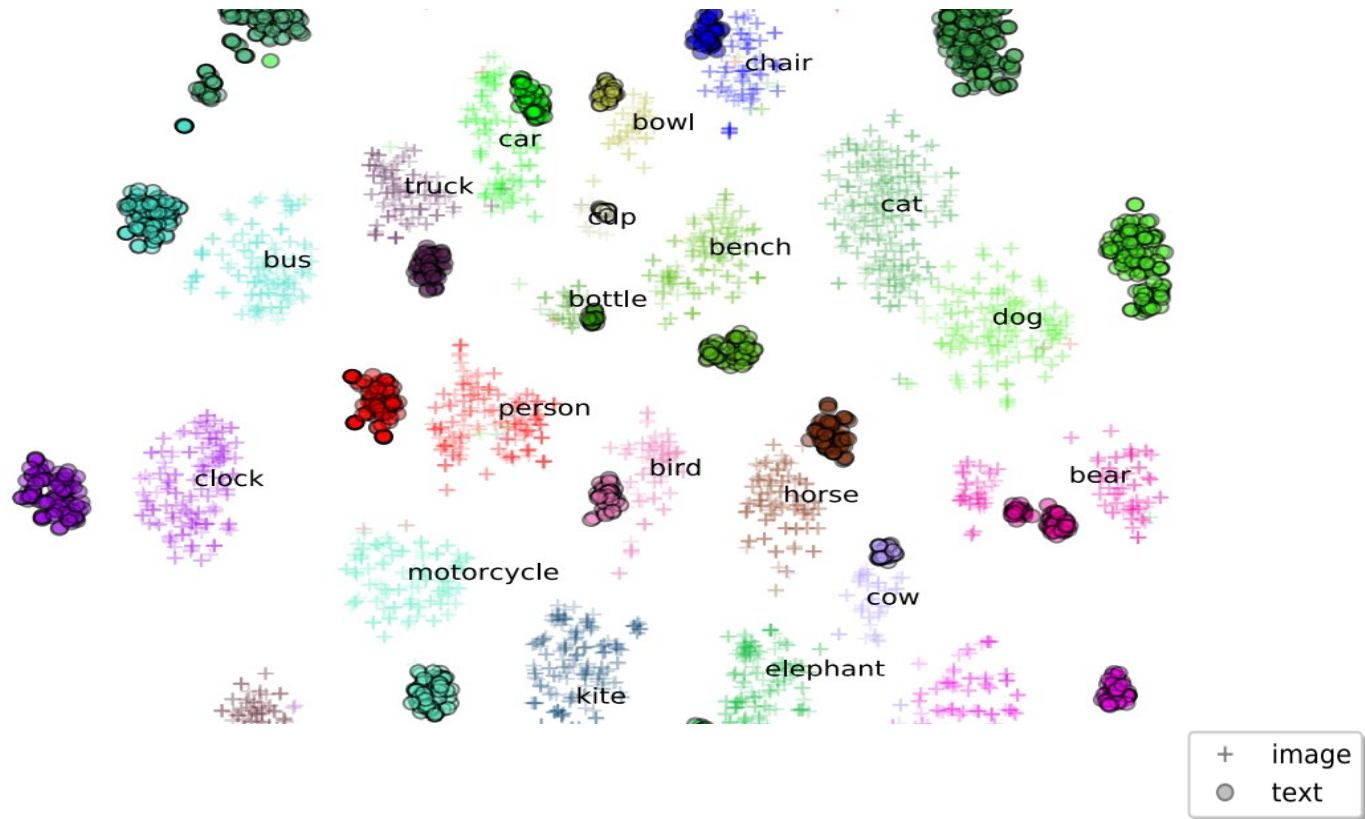


Image Change Captioning

- **Goal: Caption the difference(s) between input images**
 - Inputs: images with difference(s) + ground truth caption for the difference(s)
 - For image pair with one change



- For image pair with multiple changes (Yue et al., ICCV'21)

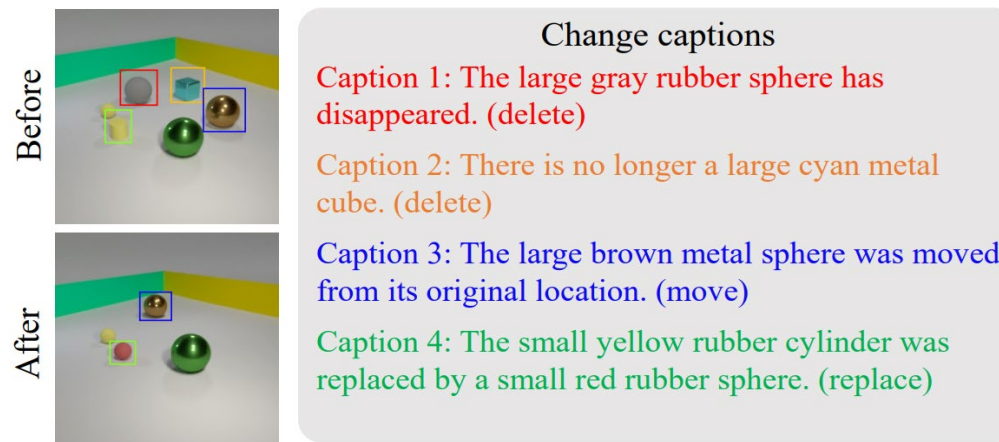
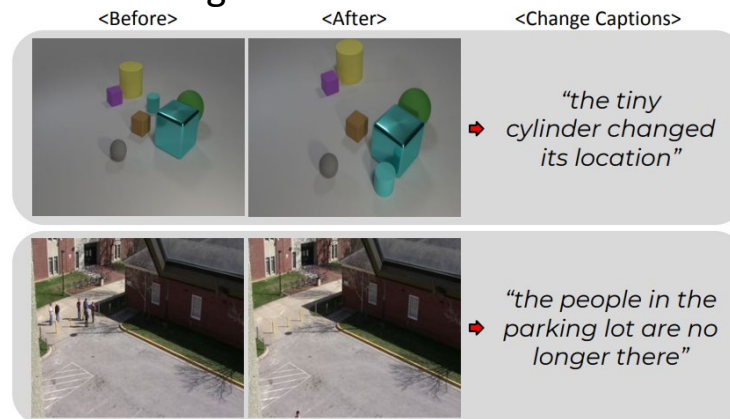


Image Change Captioning

- **Goal: Caption the difference(s) between input images**
 - Inputs: images with difference(s) + ground truth caption for the difference(s)
 - For image pair with one change



- E.g., Robust Image Change Captioning, Dong et al., ICCV'19

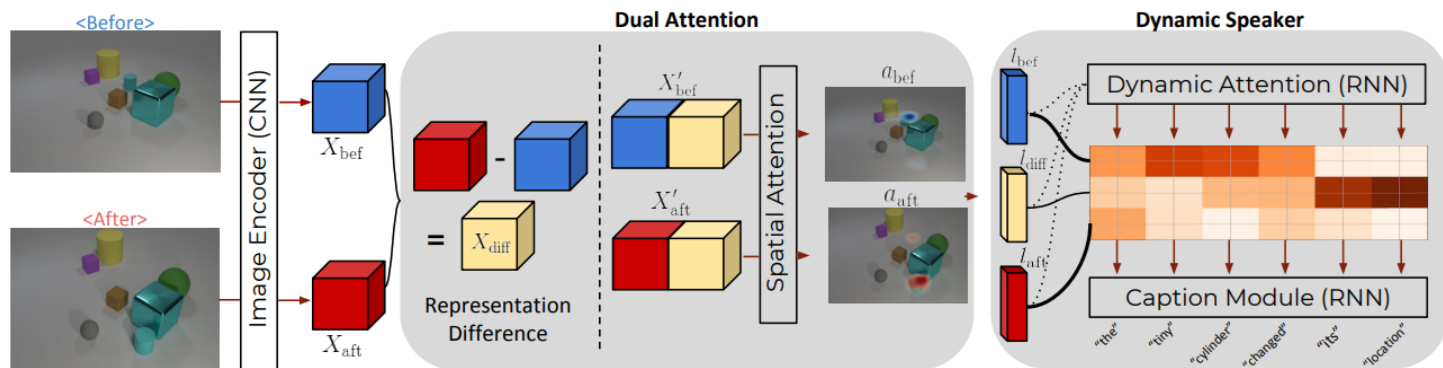
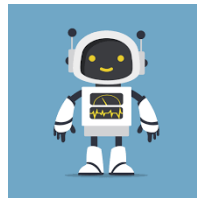


Image Manipulation

- Text-to-Image Synthesis & Manipulation
 - Task #1: Text-to-image generation
 - Produce images based on their descriptions
 - Training: image-caption pairs
 - Recent works: Show & Tell (CVPR'15), StackGAN (ICCV'17), DALL-E (OpenAI)
 - Example:

*Teddy bears shopping for groceries
in the style of ukiyo-e*



DALL-E



- Text-to-Image Synthesis & Manipulation (cont'd)
 - Text-to-image generation
 - Task #2: Image manipulation by text instruction
 - Allow users to edit an image with complex instructions (e.g., **add**, **remove**, etc.)
 - Training: reference image & instruction as inputs; target image as output
 - E.g., GeNeVa-GAN (ICCV'19), TIM-GAN (MM'21)
 - Task #3: Text/caption-guided image manipulation
 - Edit image regions to match **image descriptions**
 - Training: image-caption pairs
 - E.g., GLIDE (OpenAI'21), Tedi-GAN (CVPR'21), ManiTrans (CVPR'22)

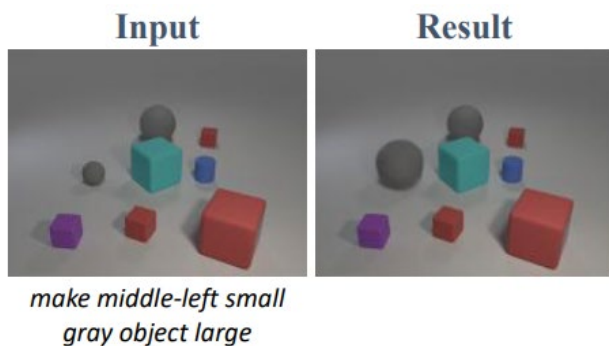


Fig. 1 Example of image manipulation by text instruction

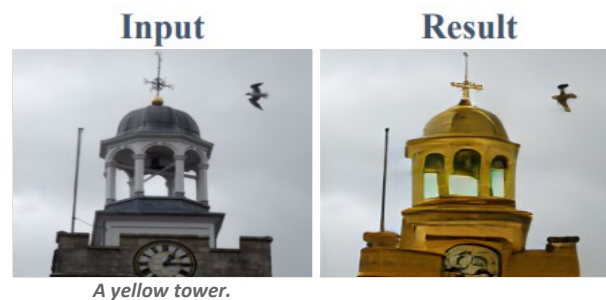


Fig. 2 Example of text (caption)-guided image manipulation

Challenges in Text-Guided Image Manipulation

- **Localization**

- Needs to **identify objects** in an image, **locate the target location** or **objects** of interest
- Requires image understanding (with both semantics & spatial info)

- **Manipulation**

- Needs to **understand the input caption/instruction** for manipulating images
- **Preserves object interaction and style** to alleviate possible mismatch after manipulation

Input



Localization



Manipulation



a fire in the background

Text-Guided Image Manipulation (cont'd)

- Remarks & Opportunities
 - Not easy to collect training data with **full supervision**
 - Large-scale **V&L pre-training models** available (e.g., CLIP)
 - **Task #2** (manipulate by instruction) vs. **Task #3** (manipulate by text guidance)

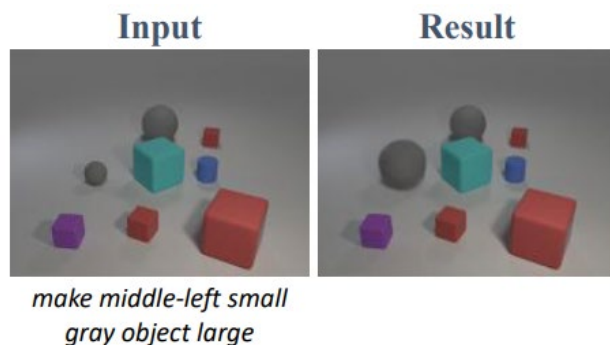


Fig. 1 Example of image manipulation by text instruction

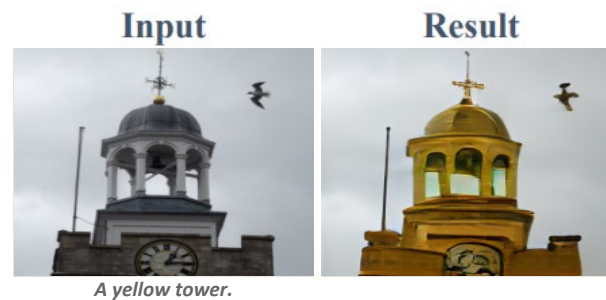


Fig. 2 Example of text (caption)-guided image manipulation

- Can scale up to industrial level with paired training data available

Selected Work on Text-Guided Image Manipulation

- GLIDE

- Developed by OpenAI in 2021
- Training:
 - Image-caption pairs and randomly generated masks
 - Learns to recover the missing part based on the caption
- Testing: image, caption, and mask annotated by user
- Later extended by a recent CVPR'22 work (DiffusionCLIP) for semantics improvements



“a corgi wearing a bow tie and a birthday hat”



“only one cloud in the sky today”

Composed Image Retrieval

- Goal
 - Given a **reference image** and its **modification text** (i.e., a cross-modal query), retrieve the **target image** from the database
 - Very different from image-text or text-image retrieval!



+

I want to change it to **longer sleeves and yellow in color.**



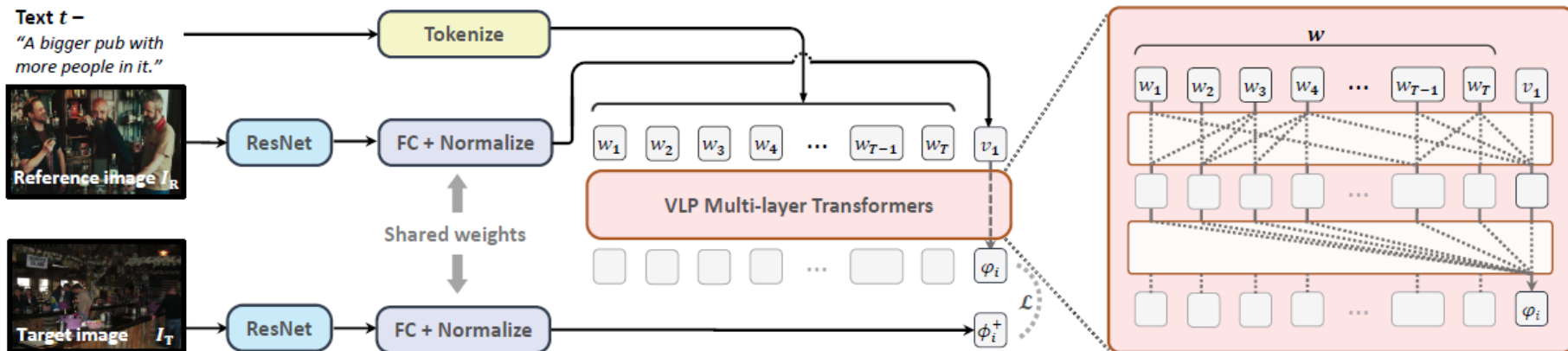
Reference Image

Modification Text

Target Image

Composed Image Retrieval with Pre-trained V&L Models

- Composed Image Retrieval using Pretrained LAnguage Transformers (CIRPLANT)
 - Extract image features by a pre-trained ResNet
 - Aggregate information from **modification text** and **reference image** by a pre-trained **OSCAR**
 - Instead of use of output token [CLS], the derived **output image feature ϕ** is used for retrieval

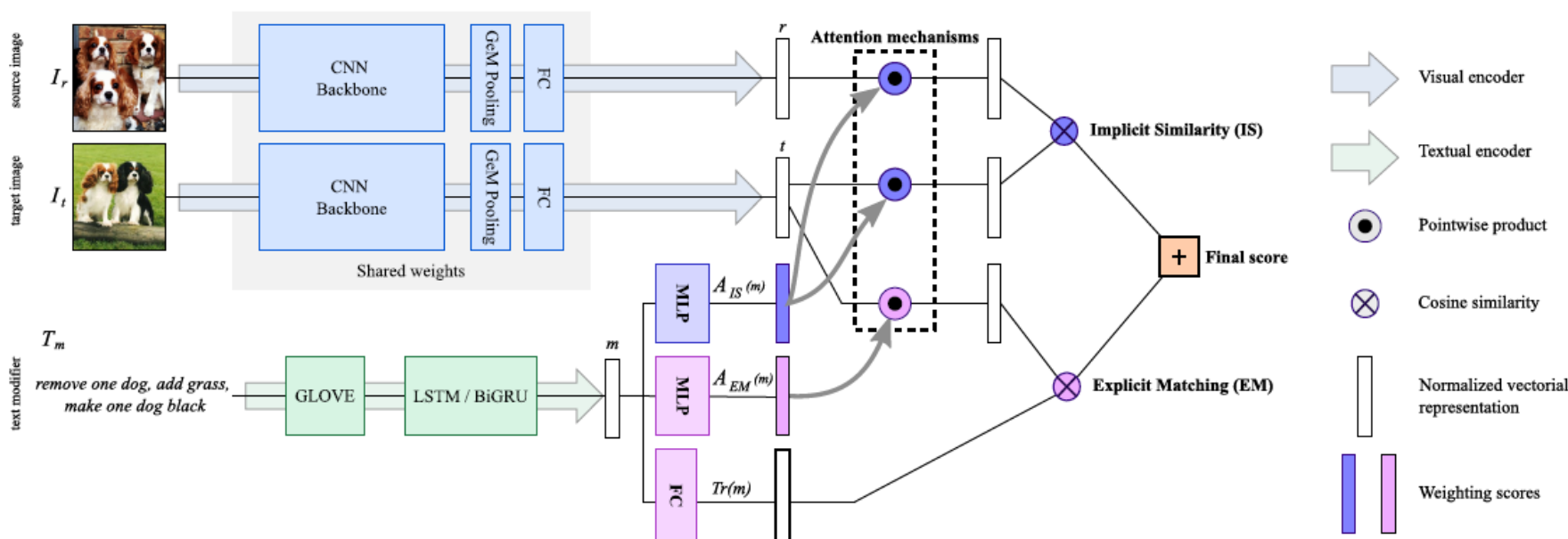


Retrieval with Text-Explicit Matching & Implicit Similarity

- **Attention-based Retrieval with Text-Explicit Matching and Implicit Similarity (ARTEMIS)**
 - Image search with free-form text modifier
 - Cross-modal learning and visual retrieval
 - **Text-guided attention** is introduced ARTEMIS

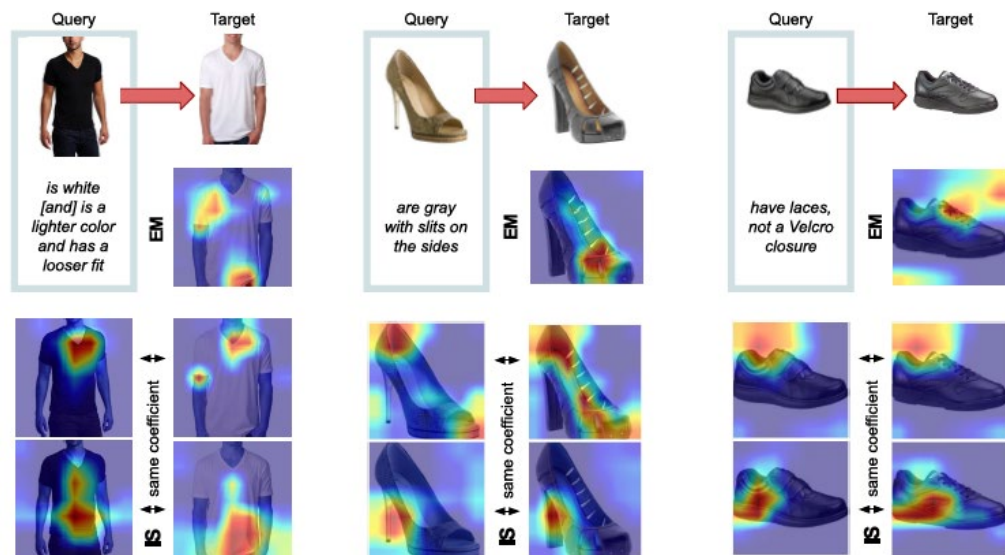


- **Attention-based Retrieval with Text-Explicit Matching and Implicit Similarity (ARTEMIS)** (cont'd)
 - **Implicit Similarity (IS):**
attention mechanism focusing on what's not mentioned by text and should be preserved
 - **Explicit Matching (EM):**
attention mechanism focusing on what's mentioned by text and should be changed.



- **Attention-based Retrieval with Text-Explicit Matching and Implicit Similarity (ARTEMIS)** (cont'd)

- Example Results & Extension



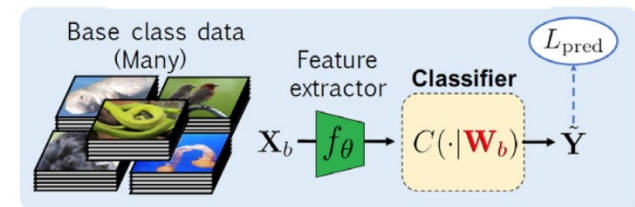
What to Cover Today...

- Recap on Transformer
- Vision & Language
 - Image Captioning
 - Text-to-Image Synthesis
- **Meta-Learning**
 - Parametric vs. Non-Parametric Approaches
 - Meta-Learning for Few-Shot Learning
 - Few-Shot Image Segmentation
 - Few-Shot Object Detection

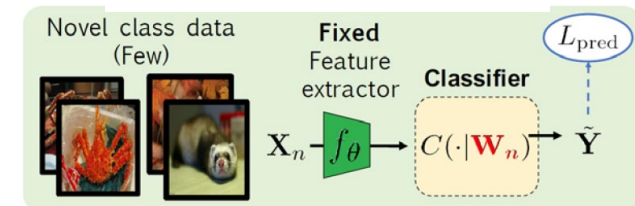


“a corgi wearing a bow tie and a birthday hat”

Meta-Training Stage

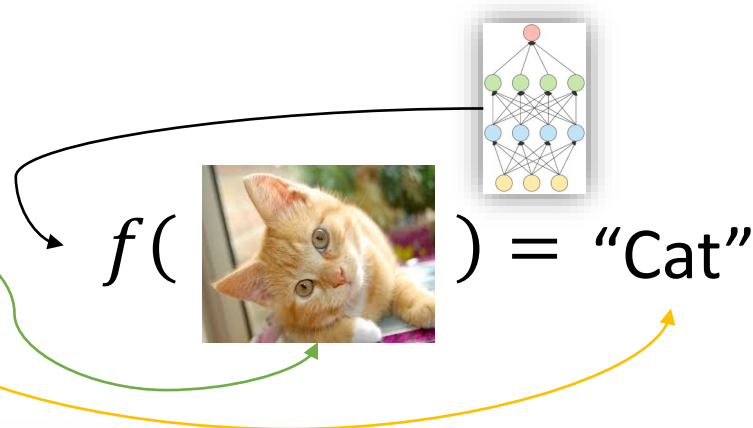


Meta-Testing Stage



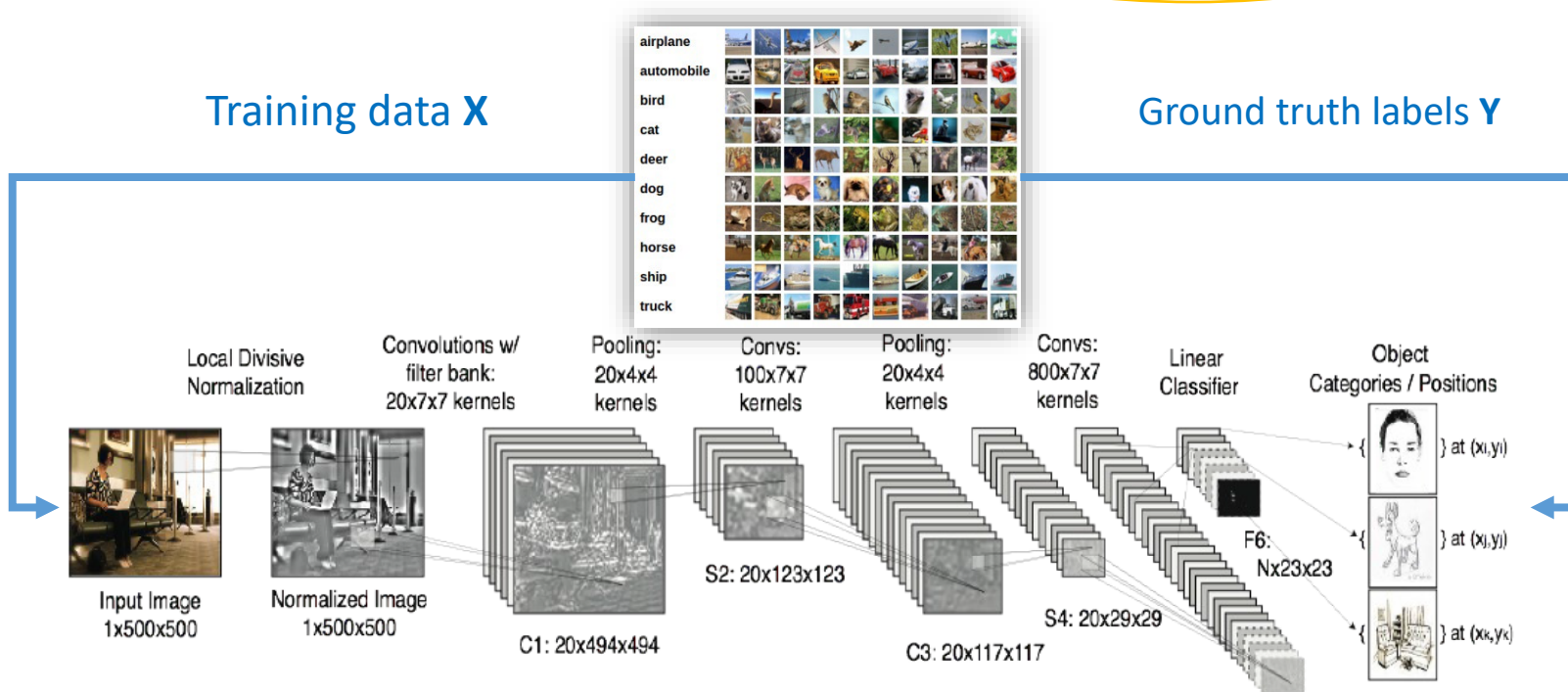
Meta Learning 元學習

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



Training data X

Ground truth labels Y



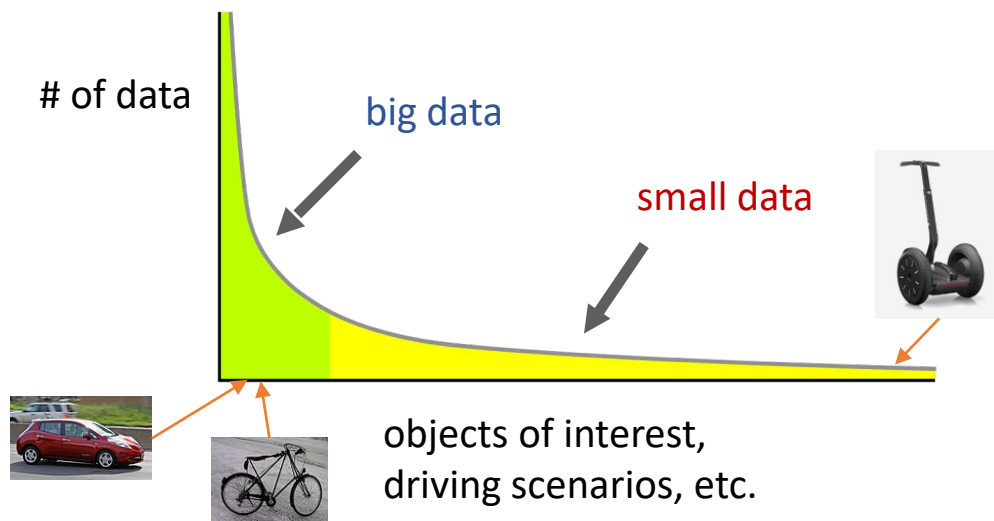
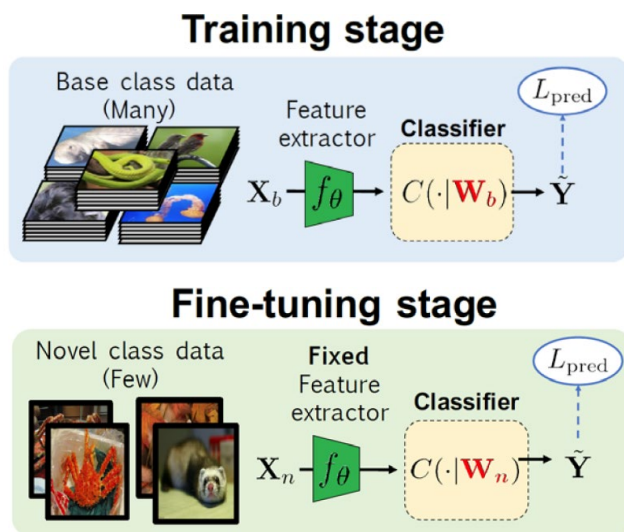
What If Only Limited Amount of Data Available?

- **Naive transfer?**

- **Model finetuning:**

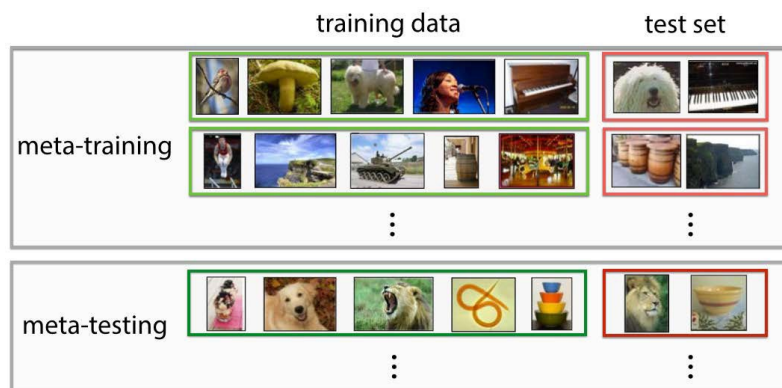
- Train a learning model (e.g., CNN) on **large-size** data (**base classes**), followed by finetuning on **small-size** data (**novel classes**).
 - That is, **freeze** feature backbone (learned from base classes) and learn/update **classifier weights** for novel classes.

- **Question: What would be the concern/limitation?**



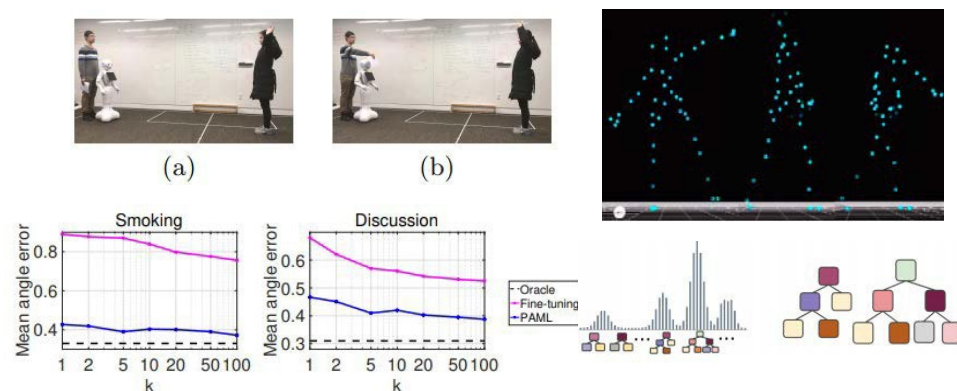
Selected Applications of Few-Shot Learning in Computer Vision

- Few-Shot Image Classification



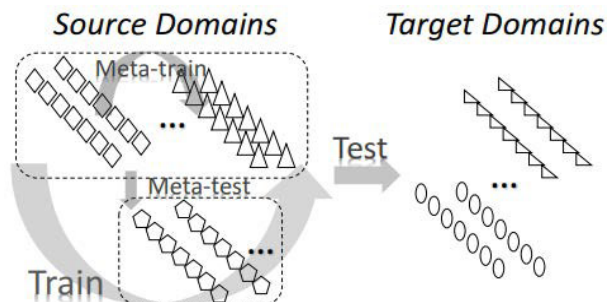
Vinyals et al., NIPS 2016

- Human Pose/Motion Prediction



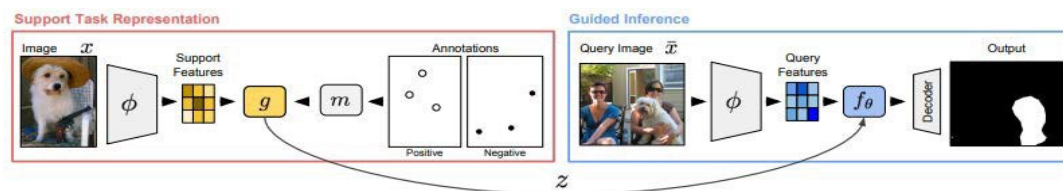
Gui et al., ECCV 2018

- Domain Transfer/Generalization



Li et al., AAAI 2018

- Few-Shot Image Segmentation



Wang et al., ICCV 2019

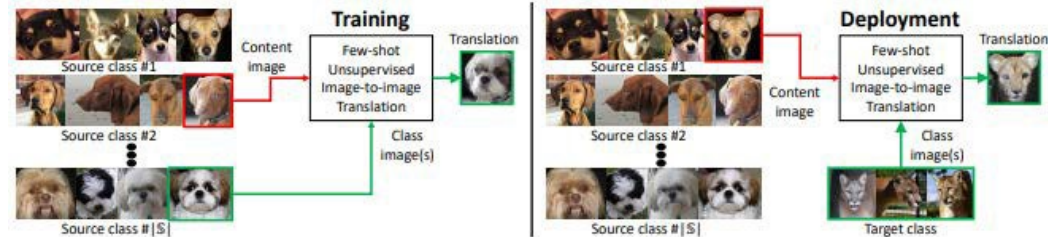
Selected Applications of Few-Shot Learning in Computer Vision

- Few-Shot Image Generation



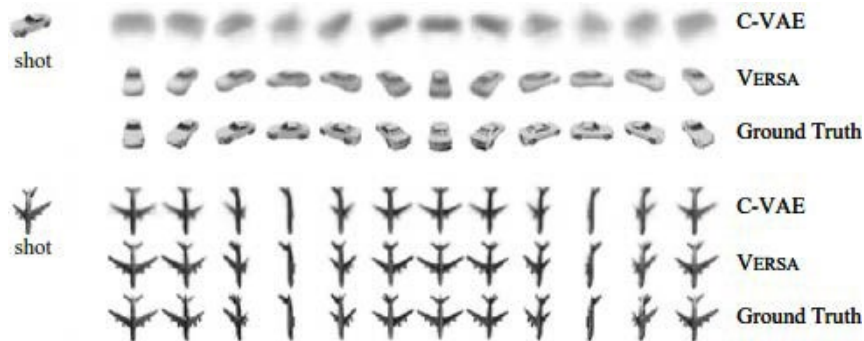
Reed et al., ICLR 2018

- Few-Shot Image-to-Image Translation



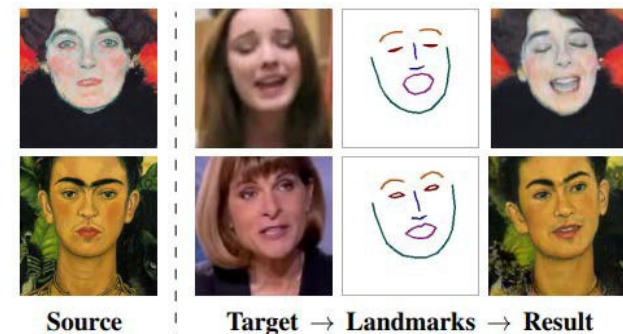
Liu et al., ICCV 2019

- Generation of Novel Viewpoints



Gordon et al., NIPS Workshop 2018

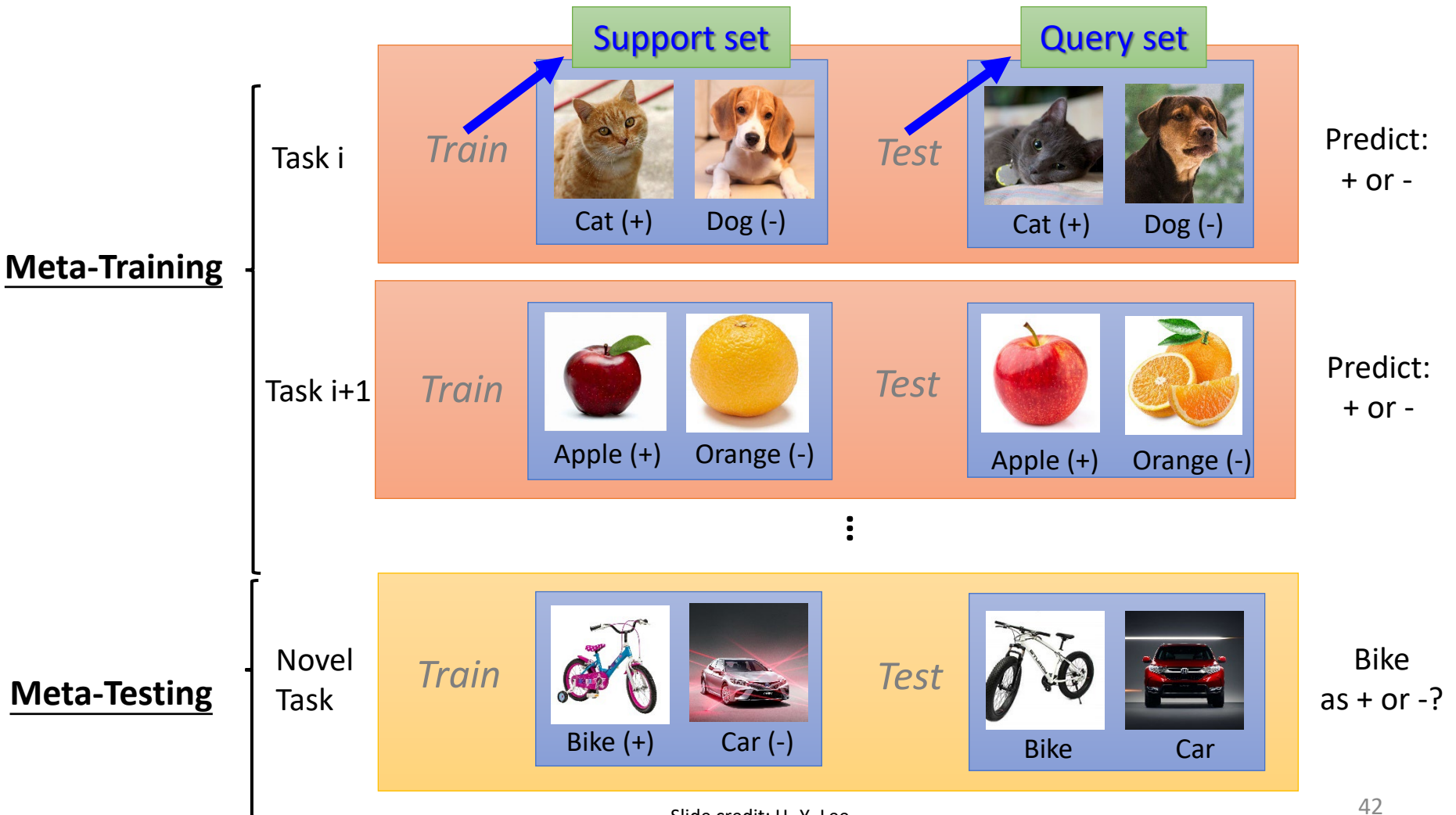
- Generating Talking Heads from Images



Zakharov et al., ICCV 2019

Meta Learning = Learning to Learn

- Let's consider the following "2-way 1-shot" learning scheme:



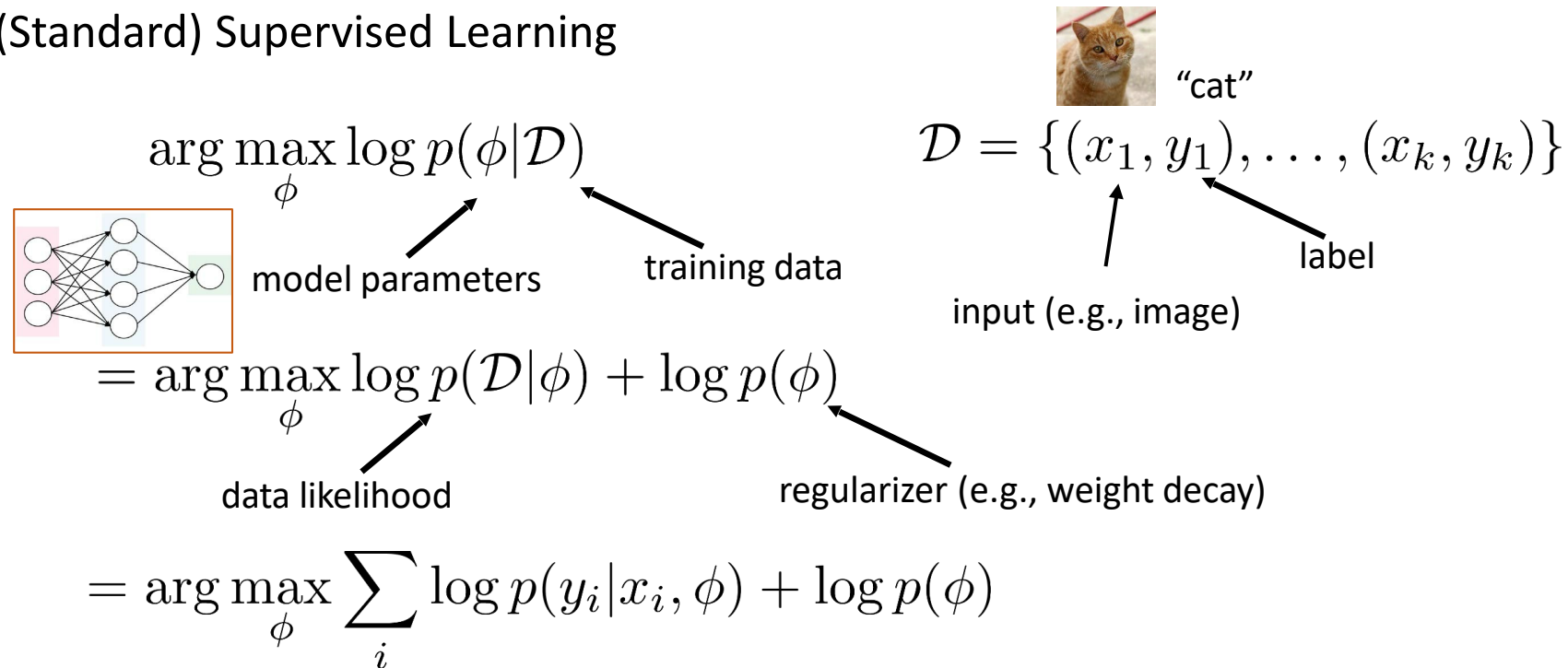
Question:

- 換句話說，元學習Meta Learning本質上是?
 - (A) 唐國師占星解盤
 - (B) 東施效顰
 - (C) 舉一反三



Some ML Backgrounds (if time permits...)

- (Standard) Supervised Learning



- 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

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

Greek

ϕ	λ	β	δ	λ
μ	α	κ	χ	ν
υ	θ	γ	τ	σ
ω	π	η	ο	ε
ρ	ξ	ζ	ψ	

➡ can we incorporate *additional* data?

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

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

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

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

\mathcal{D}_1



\mathcal{D}_2



⋮

⋮

\mathcal{D}



What Meta Learning Solves:

Object label:
"cat"



Object ID:
"person"



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

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

Greek

ϕ	λ	β	δ	λ
κ	α	κ	χ	ν
υ	θ	γ	ι	σ
ω	π	η	ο	ε
ρ	ξ	ζ	ψ	

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

➡ what if we don't want to keep $\mathcal{D}_{\text{meta-train}}$ around forever?


➡ learn *meta-parameters* θ : $p(\theta | \mathcal{D}_{\text{meta-train}})$

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


$$\begin{aligned} \text{➡ } \log p(\phi | \mathcal{D}, \mathcal{D}_{\text{meta-train}}) &= \log \int_{\Theta} p(\phi | \mathcal{D}, \theta) p(\theta | \mathcal{D}_{\text{meta-train}}) d\theta \\ &\approx \log p(\phi | \mathcal{D}, \theta^*) + \log p(\theta^* | \mathcal{D}_{\text{meta-train}}) \end{aligned}$$

What Meta Learning Solves:

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



Object label: "cat"



Object ID: "person"

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

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

Greek					
ϕ	λ	β	δ	λ	
μ	α	κ	χ	ν	
υ	θ	γ	ι	σ	
ω	π	η	ο	ε	
ρ	ξ	ζ	ψ		

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

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

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

➔ What meta learning cares is the **learning of Φ from \mathcal{D}** (and implicitly from $\mathcal{D}_{\text{meta-train}}$)

➔ What makes meta learning challenging is the **learning of optimal Θ^* from $\mathcal{D}_{\text{meta-train}}$** :

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

A Quick Example

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

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



Person ID:
"Brad Pitt"

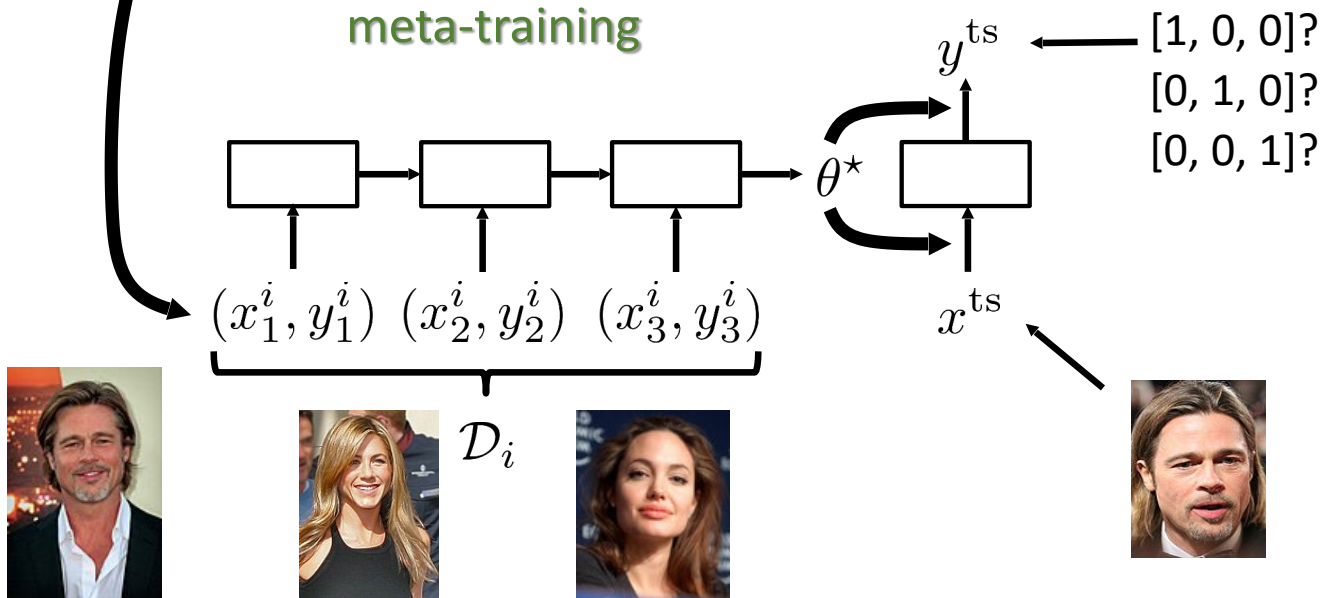
$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$

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



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

meta-training



A Quick Example (cont'd)

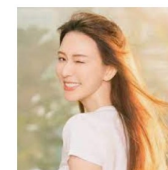
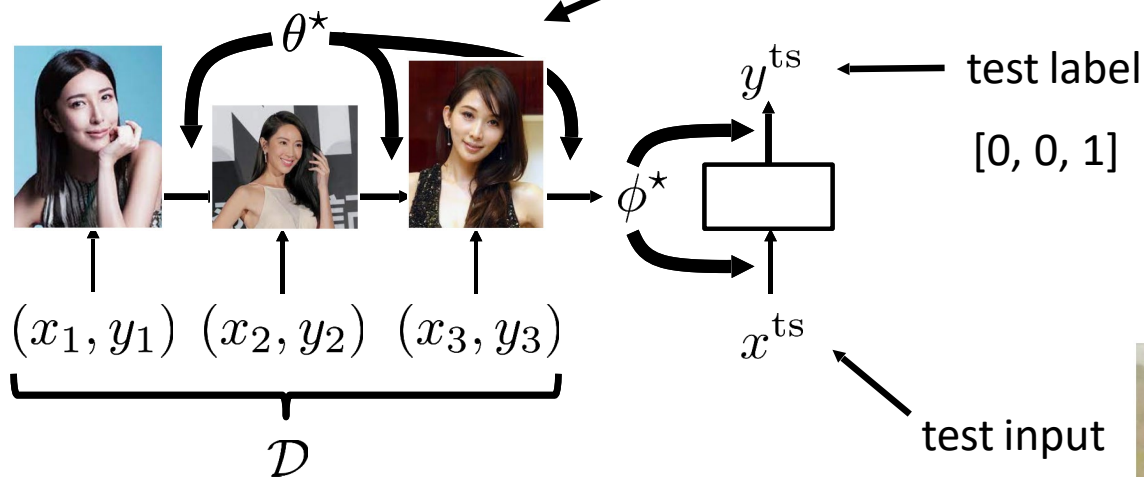
→ Meta training: $\theta^* = \arg \max_{\theta} \log p(\theta | \mathcal{D}_{\text{meta-train}})$

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

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

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

meta-testing

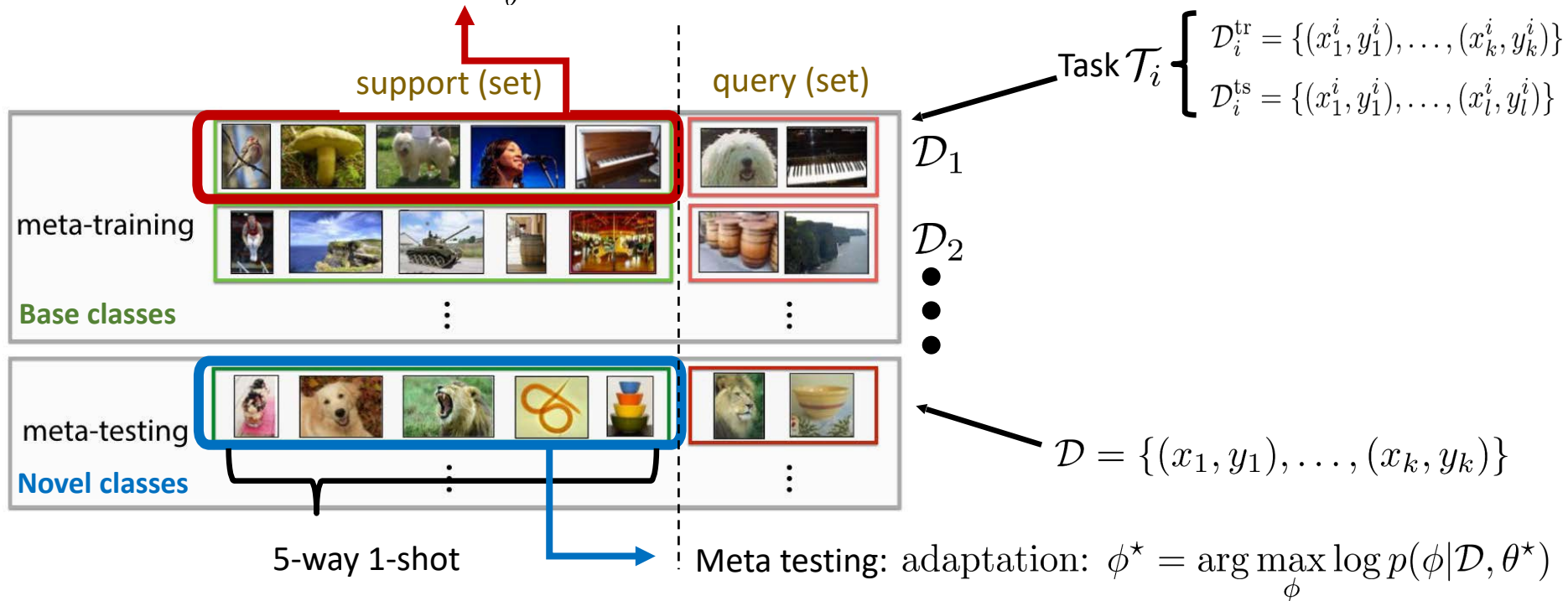


✓ Key Idea:

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 Terminology & Comments

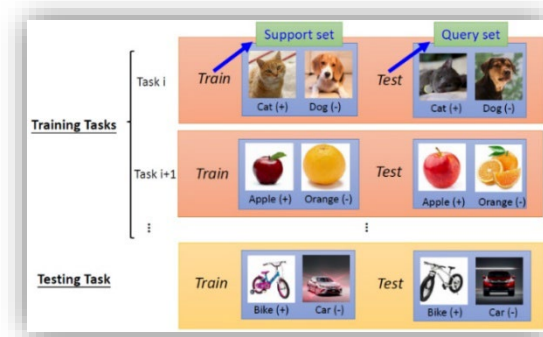
meta-learning: $\theta^* = \arg \max_{\theta} \log p(\theta | \mathcal{D}_{\text{meta-train}})$



✓ 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?

A Closely Related Yet Different Task: Multi-Task Learning



- Meta Learning

➡ Meta training: $\theta^* = \arg \max_{\theta} \log p(\theta | \mathcal{D}_{\text{meta-train}})$ $\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$

➡ Meta testing: $\phi^* = \arg \max_{\phi} \log p(\phi | \mathcal{D}, \theta^*)$ $\mathcal{D} = \{(x_1, y_1), \dots, (x_k, y_k)\}$

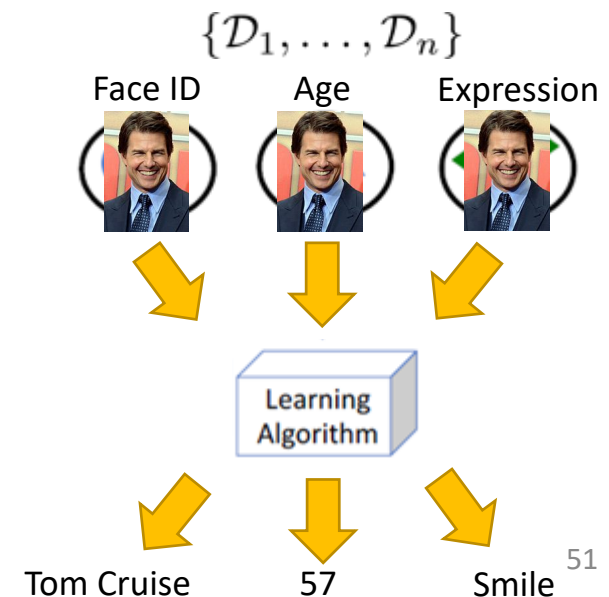
- Multi-Task Learning

- Learn model with parameter Θ^* that simultaneously solves multiple tasks

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n \log p(\theta | \mathcal{D}_i)$$

- Can be viewed as a special case where

$$\phi_i = \theta \text{ (i.e., } f_{\theta}(\mathcal{D}_i) = \theta \text{)}$$



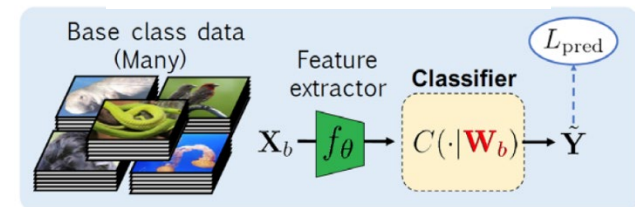
What to Cover Today...

- Recap on Transformer
- Vision & Language
 - Image Captioning
 - Text-to-Image Synthesis
- **Meta-Learning**
 - Meta-Learning for Few-Shot Learning
 - Parametric vs. Non-Parametric Approaches
 - Few-Shot Image Segmentation
 - Few-Shot Object Detection

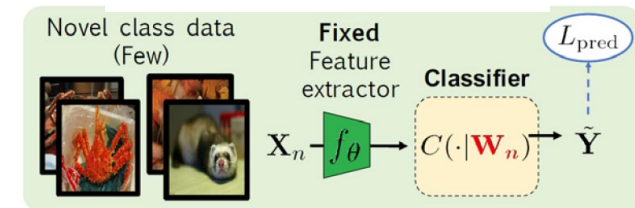


“a corgi wearing a bow tie and a birthday hat”

Meta-Training Stage



Meta-Testing Stage



Approaches

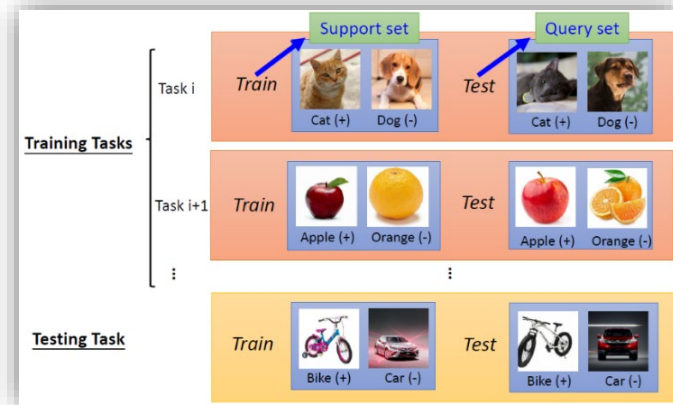
- Two Ways to View Meta Learning

- Probabilistic View (e.g., optimization-based)**

- Extract **prior** info from a set of (**meta training**) tasks, allowing efficient learning of a new task (i.e., **meta-testing**)
 - Learning a new task uses this prior and (small) training set to infer most likely **posterior model parameters**
- Easy to **understand** meta learning algorithms

- Mechanistic View (e.g., metric-learning based)**

- Meta training: A learning model (e.g., DNN) reads in a meta-dataset which consists of many datasets, each for a different task
 - Meta-testing: the model observes new data points (for a novel task) and make prediction accordingly
- Easy to implement meta learning algorithms



Approach #1: Optimization-Based Approach



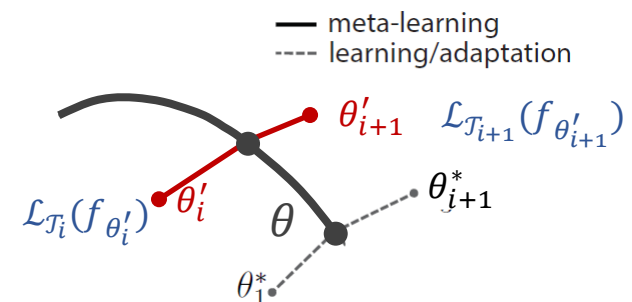
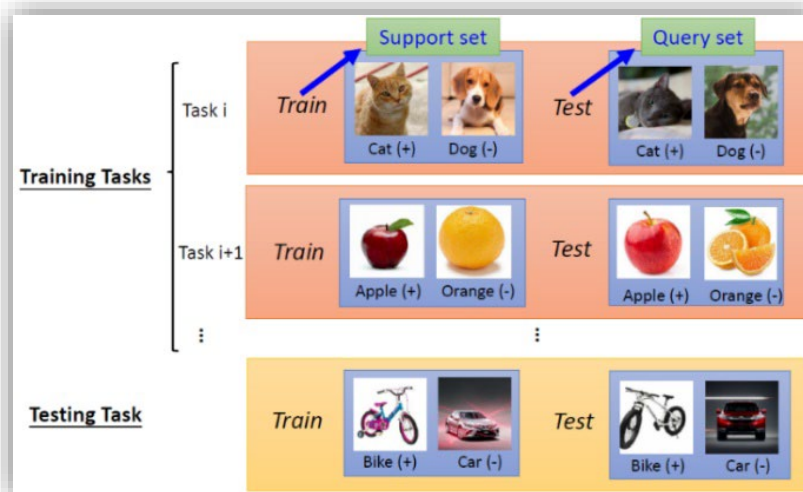
- **Model-Agnostic Meta-Learning (MAML)***

- **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 Θ_0 of model that transfers/generalizes to novel tasks well.
- That is, learn model Θ_0 which can be **fine-tuned by novel tasks efficiently/effectively**.



optimize model parameter θ so that it can quickly adapt to new tasks

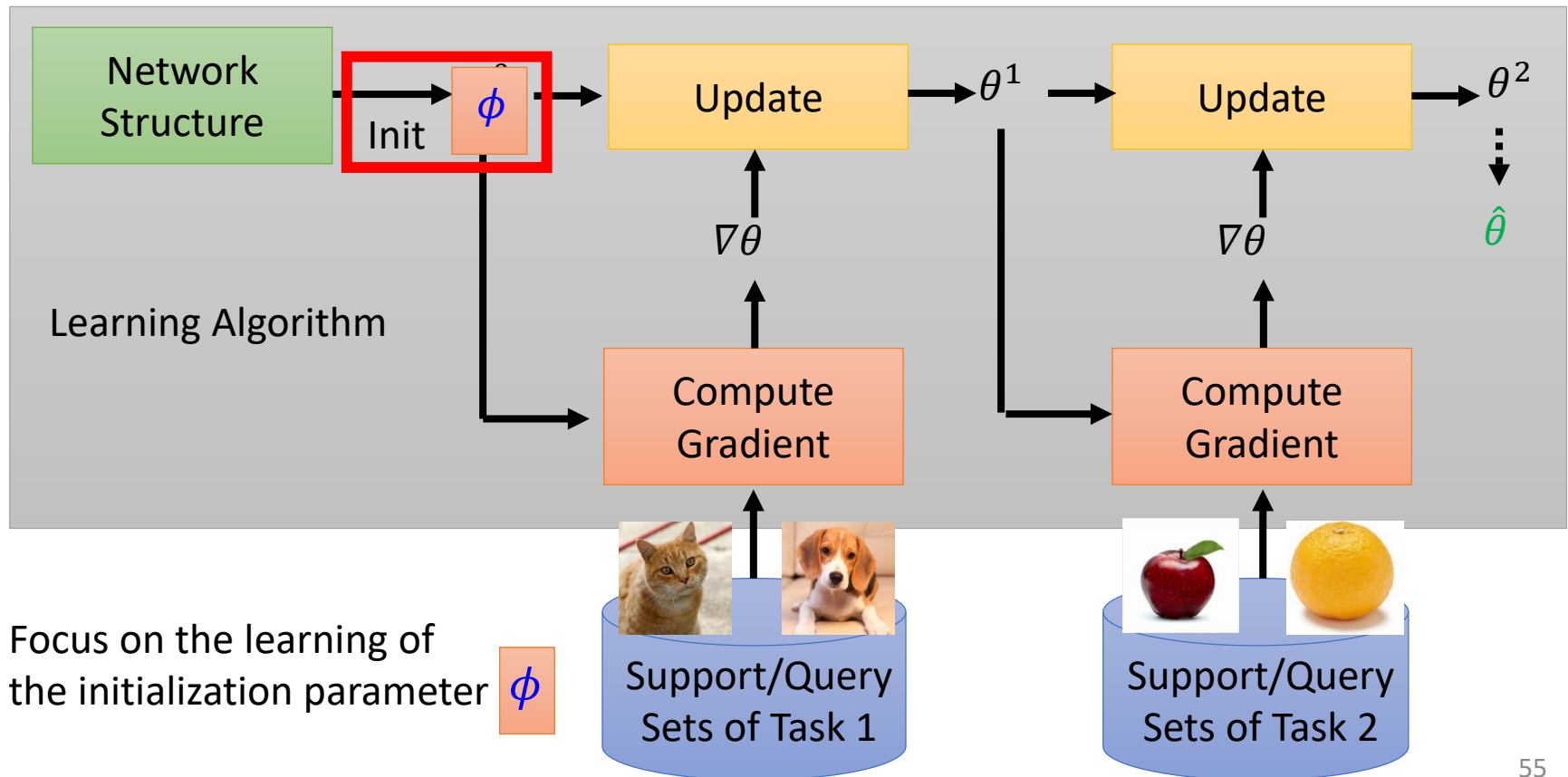
MAML

Loss Function:

$$L(\phi) = \sum_{n=1}^N l^n(\hat{\theta}^n)$$

$l^n(\hat{\theta}^n)$: loss of task n on the **query set** of task n

$\hat{\theta}^n$: model learned from the **support set** of task n

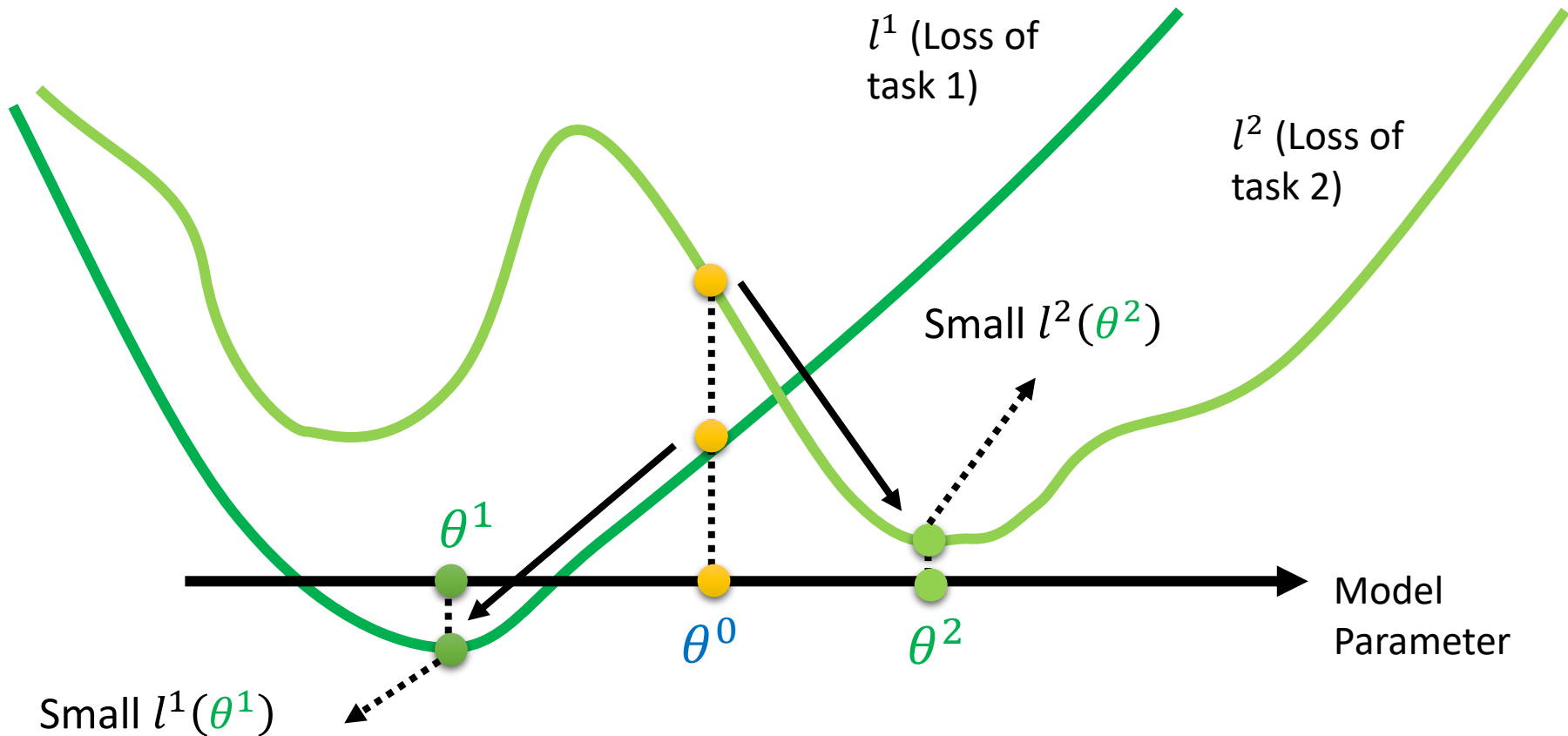


- Illustration of MAML

$$L(\theta^0) = \sum_{n=1}^N l^n(\theta^n)$$

MAML doesn't care how model θ^0 performs on each task.

It only cares how model θ^n performs for task n when starting from a properly learned θ^0 . In other words, a good θ^0 matters!

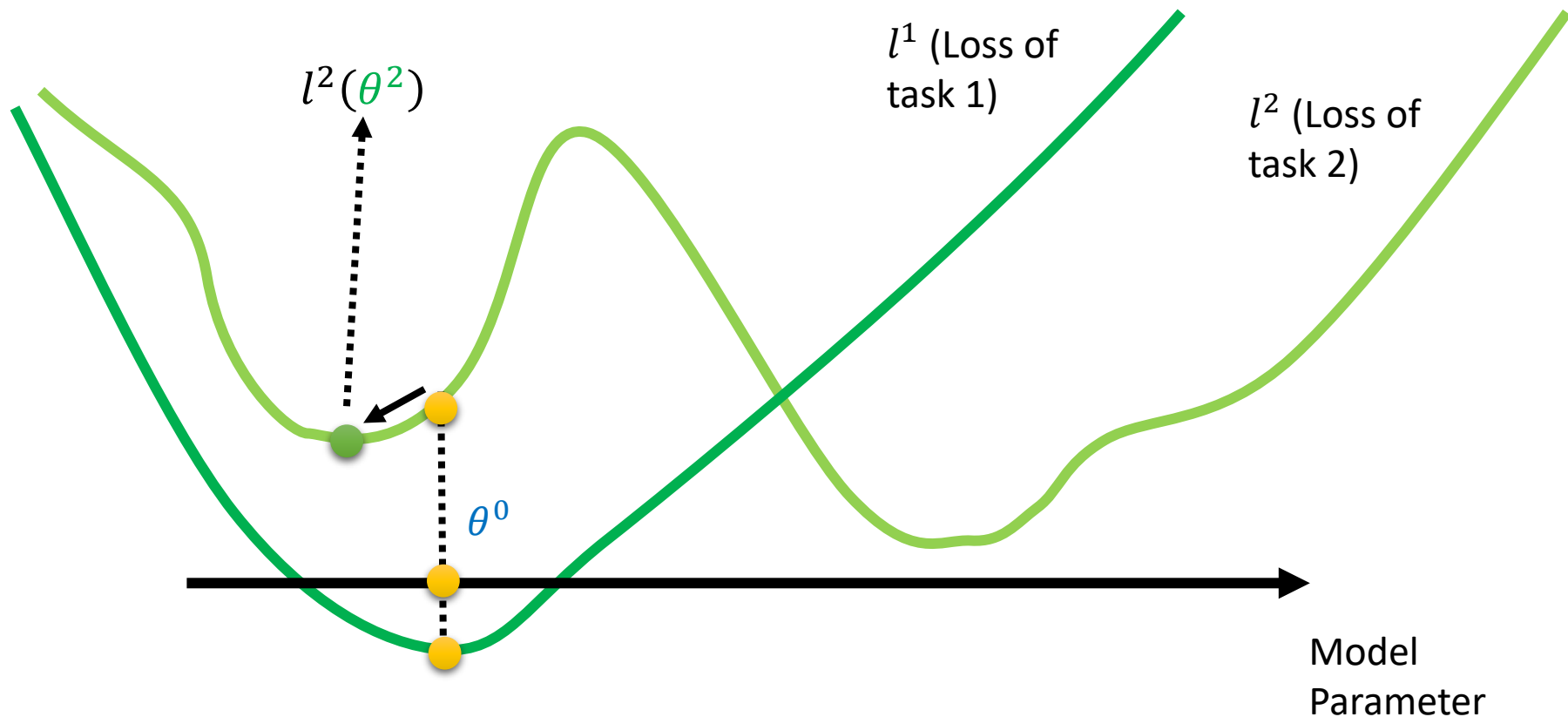


- Comparison:
Model Pre-Training or
Multi-Task Learning

$$L(\theta^0) = \sum_{n=1}^N l^n(\theta^0)$$

Determine the best θ^0 for all existing tasks

However, no guarantee that θ^0 is preferable
for learning good θ^n for task n .
Again, a good θ^0 really matters!



MAML

- Remarks

- Train a good initialized parameter set Φ (i.e., θ^0) for quick adaptation/generalization
- Meta-training:

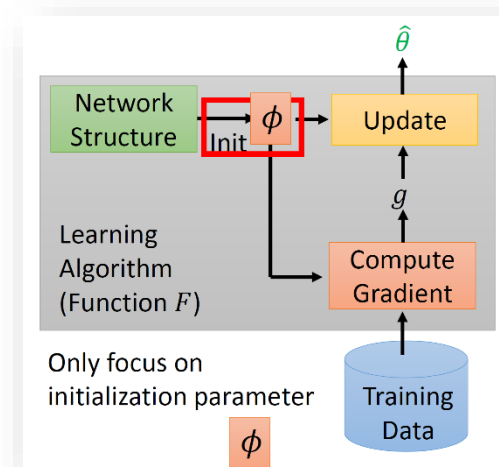
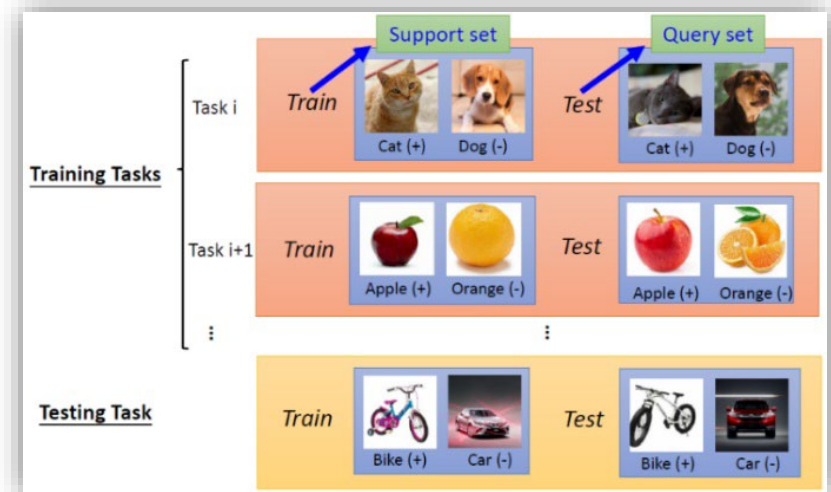
$$L(\phi) = \sum_{n=1}^N l^n(\hat{\theta}^n)$$

$$\phi \leftarrow \phi - \eta \nabla_{\phi} L(\phi)$$

- Meta-testing (for adaptation):

$$\hat{\theta} = \phi - \varepsilon \nabla_{\phi} l(\phi)$$

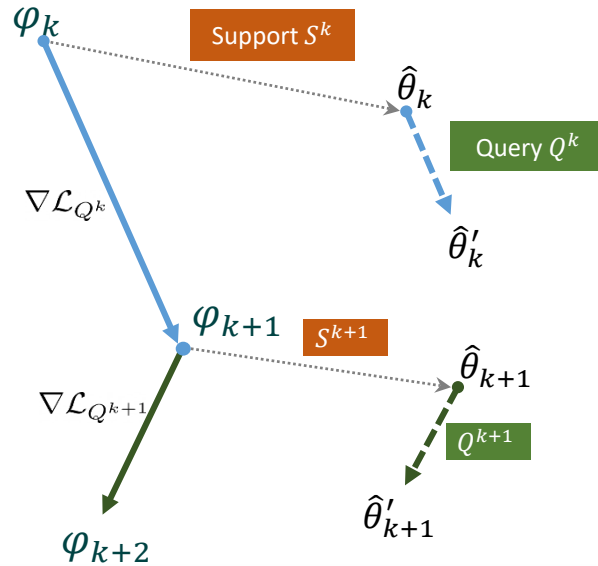
Note that one or multiple updates can be performed during meta-testing.



Meta-Training in MAML

φ : initial model parameters

$\hat{\theta}$: model parameters updated via the support set



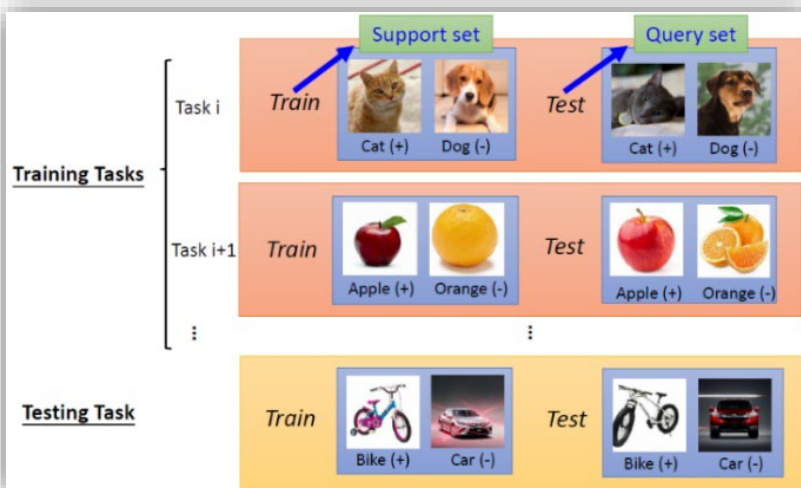
$$\varphi \leftarrow \varphi - \eta \cdot \nabla_{\varphi} L(\varphi) \quad (1)$$

$$L(\varphi) = \sum_{n=1}^N l^n(\hat{\theta}^n) \quad (2)$$

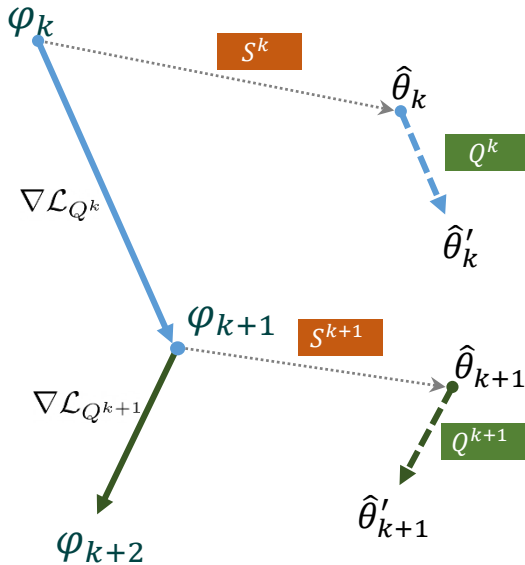
$$\hat{\theta} = \varphi - \varepsilon \cdot \nabla_{\varphi} l(\varphi) \quad (3)$$

$$\nabla_{\varphi} L(\varphi) = \sum_{n=1}^N \nabla_{\varphi} l^n(\hat{\theta}^n) \quad (4)$$

$$\nabla_{\varphi} l(\hat{\theta}) = \begin{bmatrix} \frac{\partial l(\hat{\theta})}{\partial \varphi_1} \\ \frac{\partial l(\hat{\theta})}{\partial \varphi_2} \\ \dots \\ \frac{\partial l(\hat{\theta})}{\partial \varphi_i} \end{bmatrix} \quad (5)$$



MAML



$$\hat{\theta} = \varphi - \varepsilon \cdot \nabla_{\varphi} l(\varphi) \quad (3)$$

$$\nabla_{\varphi} L(\varphi) = \sum_{n=1}^N \nabla_{\varphi} l^n(\hat{\theta}^n) \quad (4)$$

$$\nabla_{\varphi} l(\hat{\theta}) = \begin{bmatrix} \frac{\partial l(\hat{\theta})}{\partial \varphi_1} \\ \frac{\partial l(\hat{\theta})}{\partial \varphi_2} \\ \dots \\ \frac{\partial l(\hat{\theta})}{\partial \varphi_i} \end{bmatrix} \quad (5)$$

$$\frac{\partial l(\hat{\theta})}{\partial \varphi_i} = \sum \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_j} \frac{\partial \hat{\theta}_j}{\partial \varphi_i}$$

First-order approximation:

If $i \neq j$, then:

$$\hat{\theta}_j = \varphi_j - \varepsilon \cdot \frac{\partial l(\varphi)}{\partial \varphi_j} \quad \frac{\partial \hat{\theta}_j}{\partial \varphi_i} = -\varepsilon \cdot \frac{\partial l(\varphi)}{\partial \varphi_j \partial \varphi_i} \approx 0$$

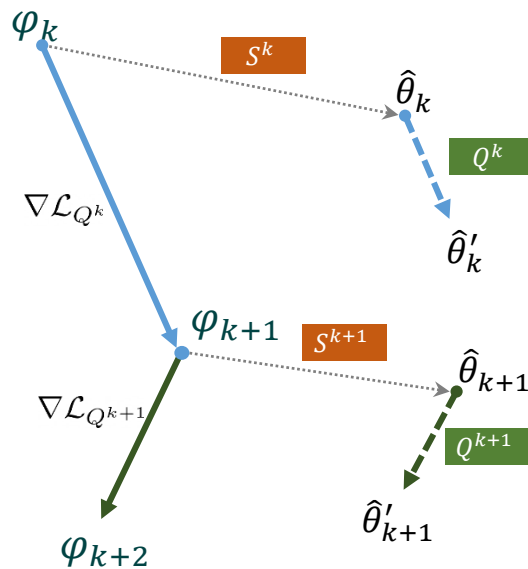
If $i = j$, then:

$$\frac{\partial \hat{\theta}_j}{\partial \varphi_i} = 1 - \varepsilon \cdot \frac{\partial l(\varphi)}{\partial \varphi_j \partial \varphi_i} \approx 1$$

φ : initial model parameters

$\hat{\theta}$: model parameters updated via the support set

MAML



$$\nabla_{\varphi} l(\hat{\theta}) = \begin{bmatrix} \frac{\partial l(\hat{\theta})}{\partial \varphi_1} \\ \frac{\partial l(\hat{\theta})}{\partial \varphi_2} \\ \dots \\ \frac{\partial l(\hat{\theta})}{\partial \varphi_i} \end{bmatrix} = \begin{bmatrix} \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_1} \\ \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_2} \\ \dots \\ \frac{\partial l(\hat{\theta})}{\partial \hat{\theta}_i} \end{bmatrix} = \nabla_{\hat{\theta}} l(\hat{\theta})$$

$$\nabla_{\varphi} L(\varphi) = \sum_{n=1}^N \nabla_{\varphi} l^n(\hat{\theta}^n) = \sum_{n=1}^N \nabla_{\hat{\theta}} l^n(\hat{\theta}^n)$$

$$\Rightarrow \varphi \leftarrow \varphi - \underbrace{\eta \cdot \nabla_{\varphi} L(\varphi)}_{\text{gradient w.r.t } \varphi} = \varphi - \underbrace{\eta \cdot \nabla_{\hat{\theta}} L(\hat{\theta})}_{\text{gradient w.r.t } \hat{\theta}}$$

Recap: MAML

- Remarks

- Train a good initialized parameter set Φ (i.e., θ^0) for quick adaptation/generalization
- Meta-training:

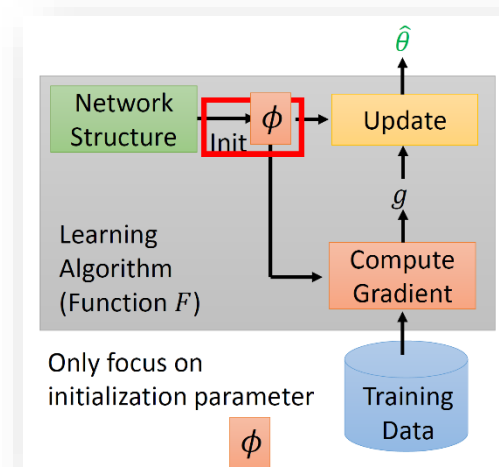
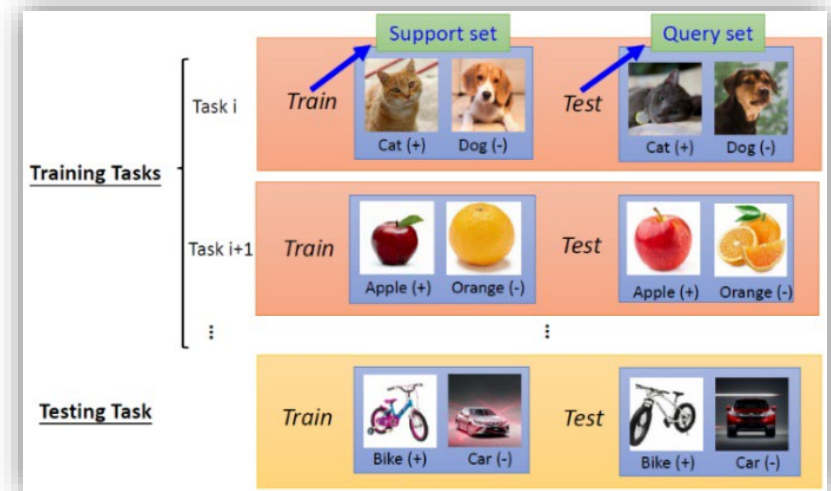
$$L(\phi) = \sum_{n=1}^N l^n(\hat{\theta}^n)$$

$$\phi \leftarrow \phi - \eta \nabla_{\phi} L(\phi)$$

- Meta-testing (for adaptation):

$$\hat{\theta} = \phi - \varepsilon \nabla_{\phi} l(\phi)$$

Note that one or multiple updates can be performed during meta-testing.



Approaches

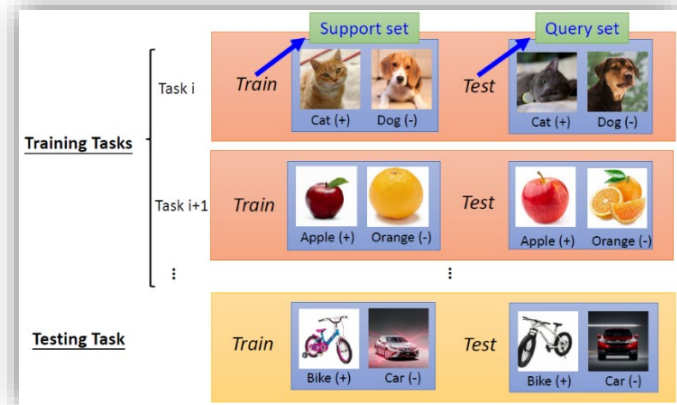
- Two Ways to View Meta Learning

- Probabilistic View (e.g., optimization-based)*

- Extract prior info from a set of (meta training) tasks, allowing efficient learning of a new task (i.e., meta-testing)
 - Learning a new task uses this prior and (small) training set to infer most likely posterior model parameters
- Easy to understand meta learning algorithms

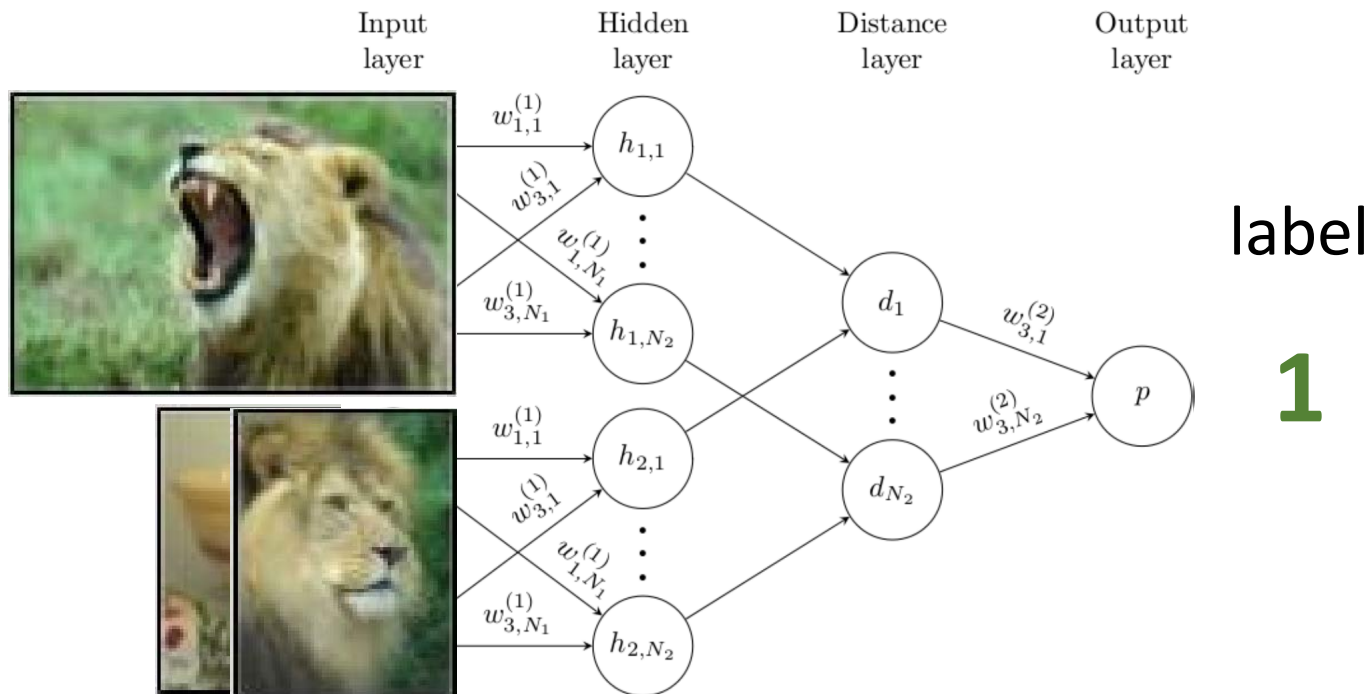
- Mechanistic View (e.g., metric-learning based)**

- Meta training:** A learning model (e.g., DNN) reads in a **meta-dataset** which consists of many datasets, each for a different task
 - Meta-testing:** the model observes new data points (for a **novel** task) and make prediction accordingly
- Easy to **implement** meta learning algorithms



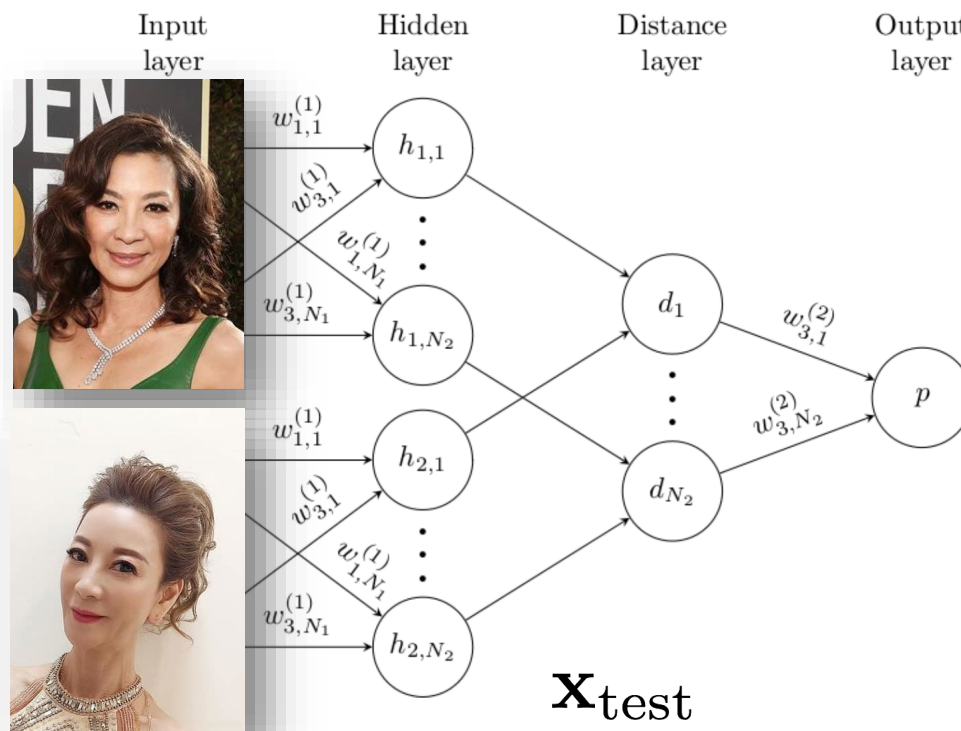
Approach #2: Non-Parametric Approach

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



Learn to Compare (cont'd)

- Siamese Network (cont'd)
 - Meta-training/testing: learn to match (i.e., 2-way image matching)
 - Question: output label of the following example is **1** or **0**? (i.e., **same ID** or **not**)



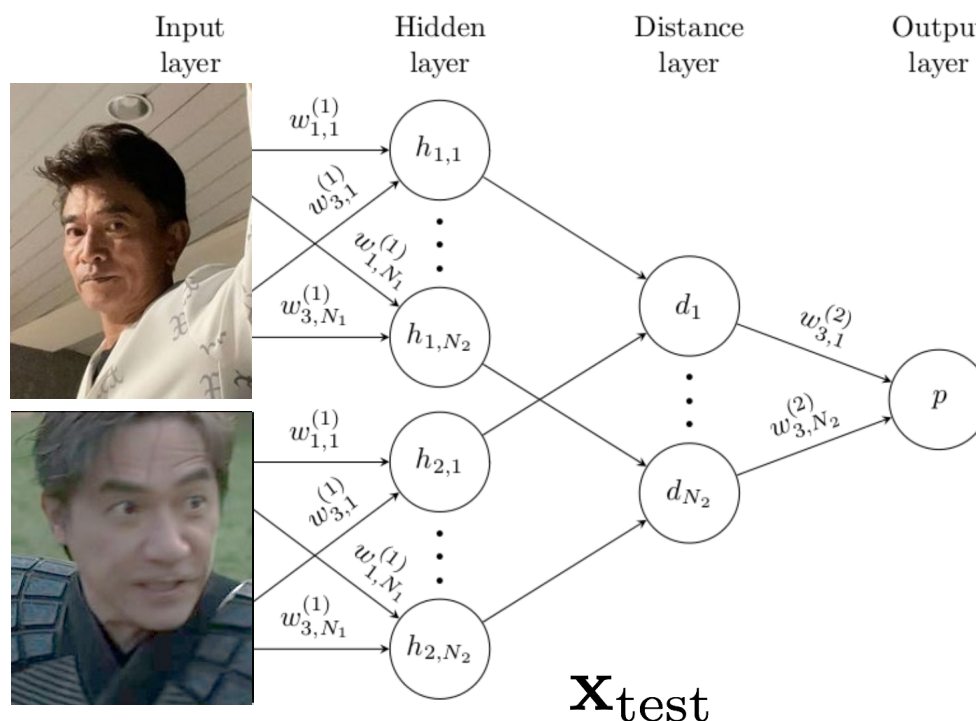
label

?



Learn to Compare (cont'd)

- Siamese Network (cont'd)
 - Meta-training/testing: learn to match (i.e., 2-way image matching)
 - Question: output label of the following example is 1 or 0?
(i.e., same ID or not)



label

?

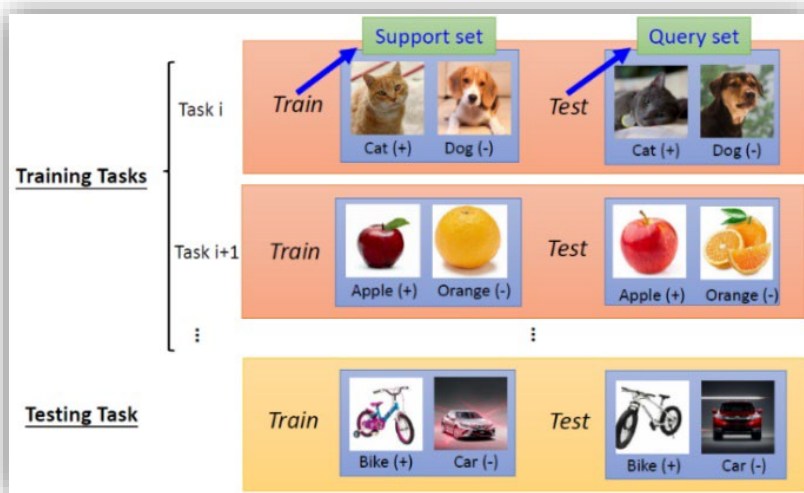


- What did we learn from these examples?
- And, can we perform multi-way classification (beyond matching)?

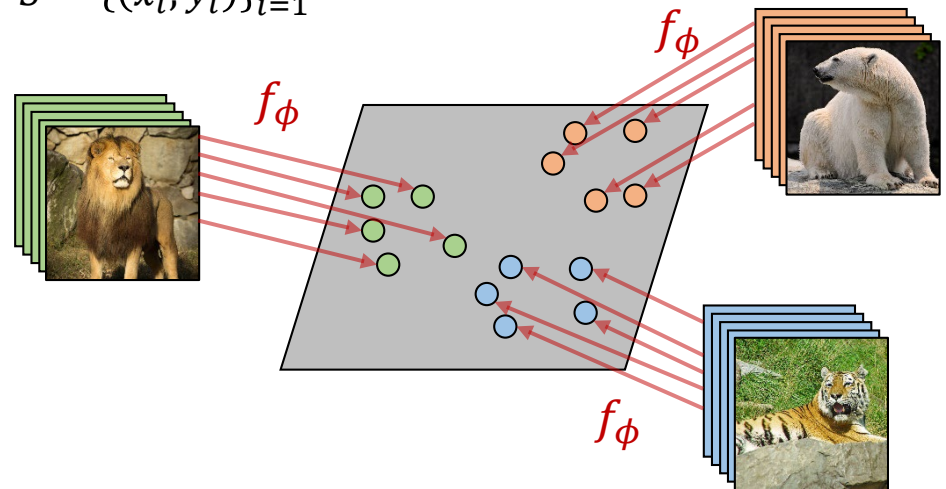
Learn to Compare...with the Representative Ones!

- **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.

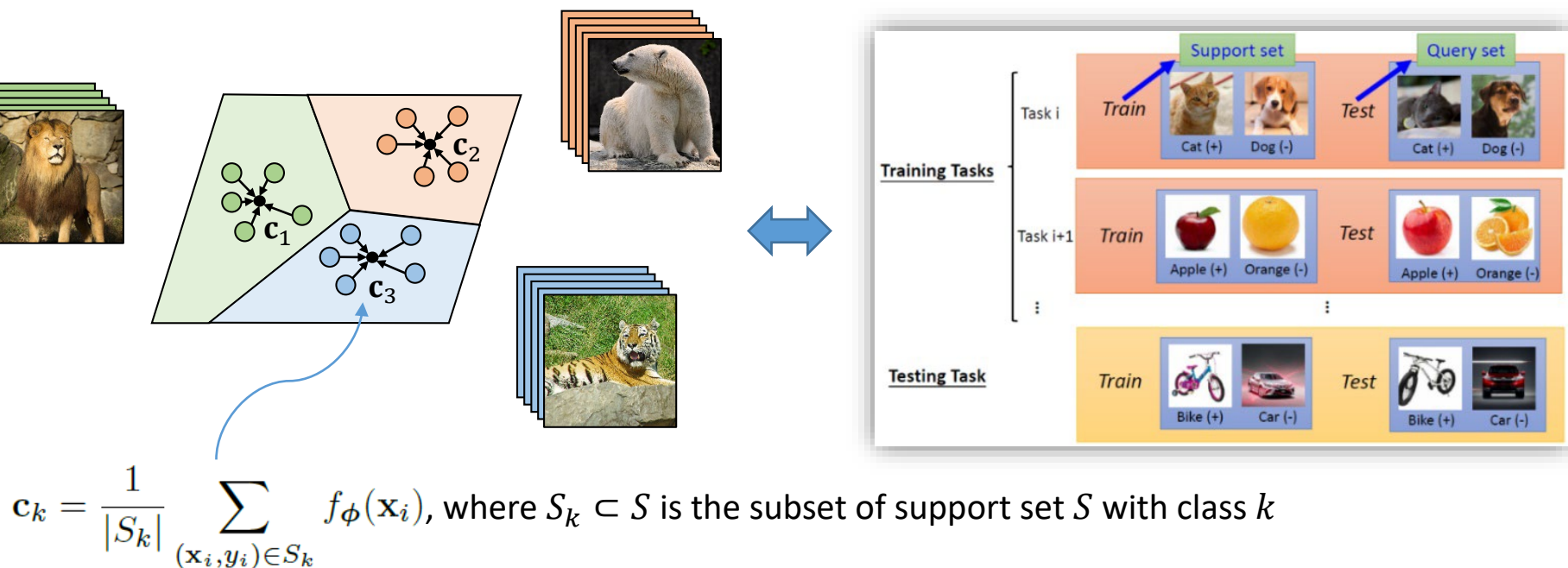


support set
 $S = \{(x_i, y_i)\}_{i=1}^k$



• 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_ϕ and the representatives (or anchors) of each class is the **mean feature vector** \mathbf{c}_k .



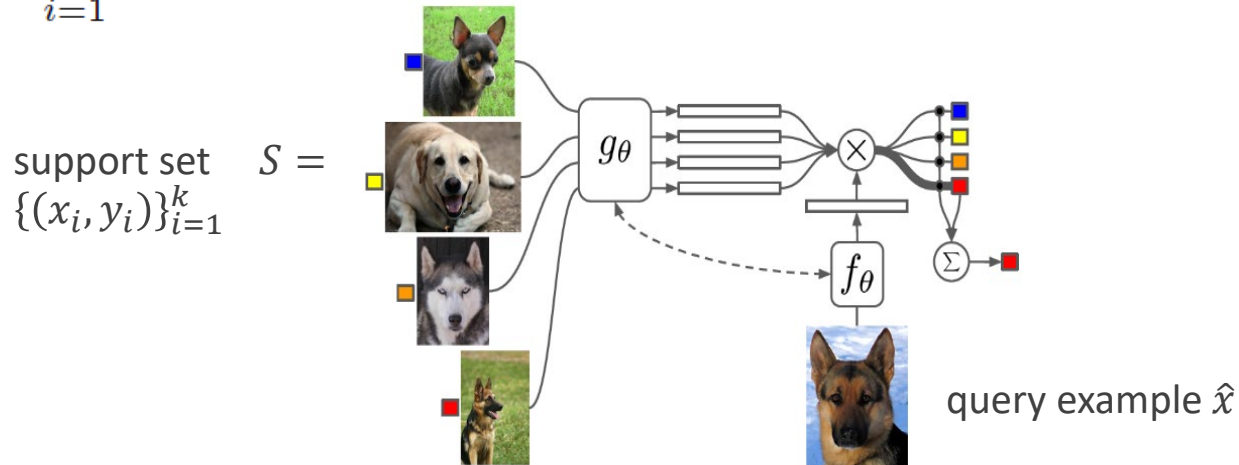
Learn to Compare

- **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.

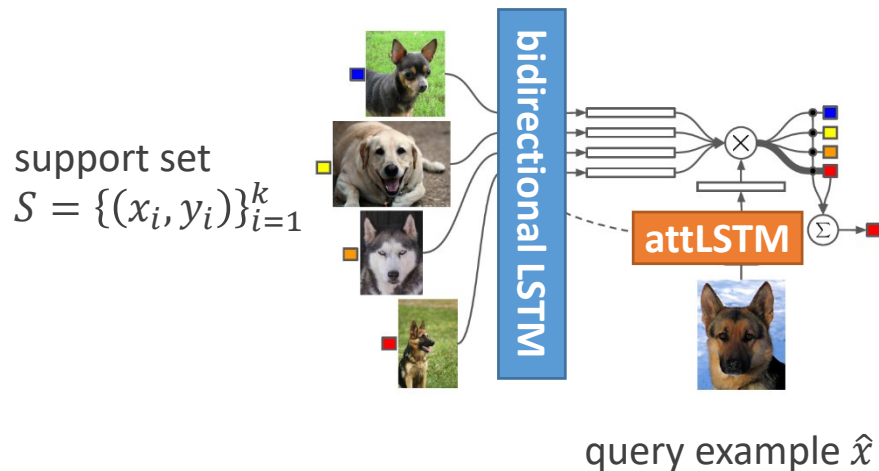
$$\hat{y} = \sum_{i=1}^k a(\hat{x}, x_i) y_i \quad \text{with} \quad a(\hat{x}, x_i) = e^{c(f(\hat{x}), g(x_i))} / \sum_{j=1}^k e^{c(f(\hat{x}), g(x_j))}$$

$c(.,.):$ cosine similarity



- **Matching Networks (cont'd)**

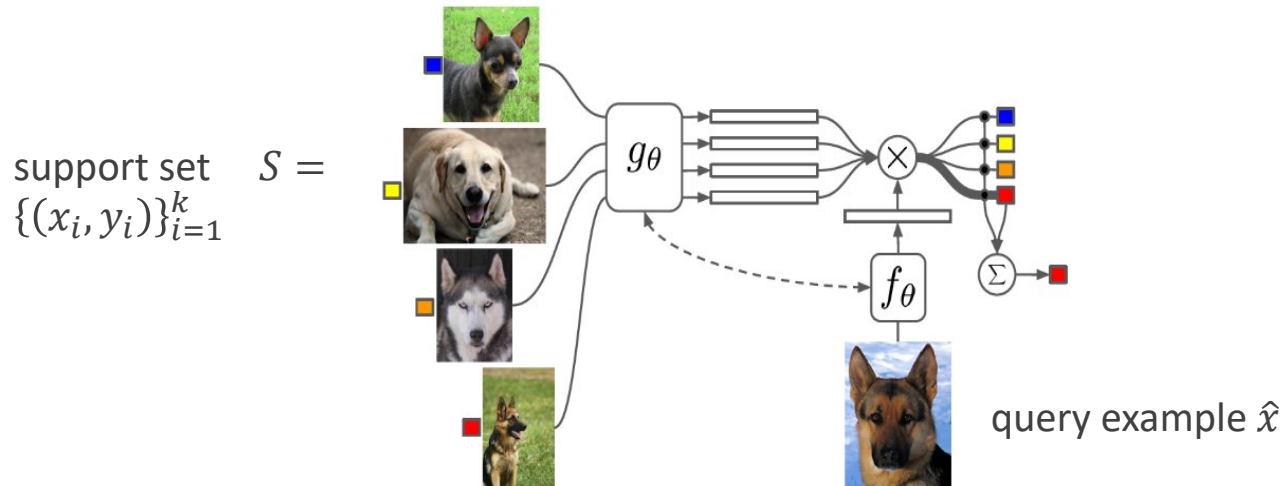
- Full context embedding (FCE)
- Each element in S should not be embedded independently of other elements
 - $g(x_i) \rightarrow g(S)$ as a **bidirectional LSTM** by considering the whole S as a **sequence**
- Also, S should be able to modify the way we embed \hat{x}
 - $f(\hat{x}) \rightarrow f(\hat{x}, S)$ 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 *minilmageNet*



Model	Matching Fn	Fine Tune	5-way Acc	
			1-shot	5-shot
PIXELS	Cosine	N	23.0%	26.6%
BASILINE CLASSIFIER	Cosine	N	36.6%	46.0%
BASILINE CLASSIFIER	Cosine	Y	36.2%	52.2%
BASILINE 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%

Learn to Compare

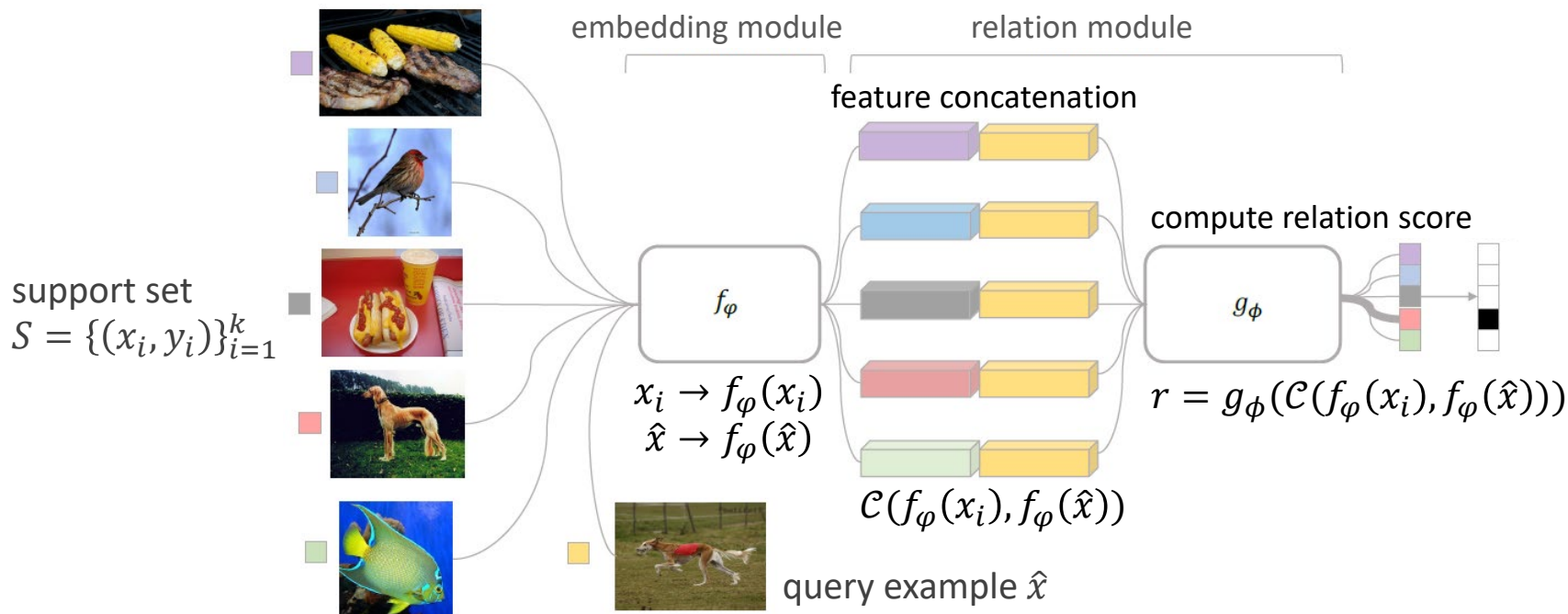
- **Matching Networks** (cont'd)
 - If we have $g = f$,
the model turns into a Siamese network like architecture
 - Also similar to prototypical network for **one**-shot learning



Learn to Compare...with Self-Learned Metrics!

- **Relation Network**

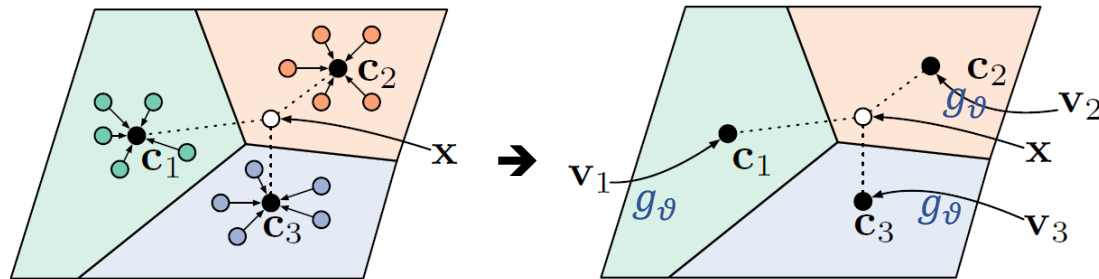
- Metric-learning approaches typically focus on learning an embedding function with a **fixed metric** (e.g., Euclidean distance, cosine similarity, ...)
- The authors proposed to train a **Relation Network** (RN) to explicitly learn a transferrable **deep distance metric** comparing the relation between images



Relation Networks (cont'd)

- Extension to **zero-shot learning** (if time permits):
 - Instead of few-shot images, the support set contains a **semantic embedding vector** \mathbf{v}_k (e.g., embedding of class label) for each training class.
 - One can use a **heterogeneous** embedding function g_ϑ to embed the semantic embedding vectors, which relates the image data $f_\phi(\mathbf{x}_i)$
 - E.g., **Prototypical Network** for zero-shot image classification:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i) \Rightarrow \mathbf{c}_k = g_\vartheta(\mathbf{v}_k)$$



Some Takeaways for Existing Meta-Learning Approaches

Parametric-based

- + handles **varying & large K** well
- + **structure lends well to out-of-distribution tasks**
- **second-order optimization**

Non-parametric based

- + **simple**
- + **entirely feedforward**
- + **computationally fast & easy to optimize**
- **harder to generalize to varying K**
- **hard to scale to very large K**
- **so far, limited to classification**

Generally, well-tuned versions of each perform **comparably** on existing FSL benchmarks.

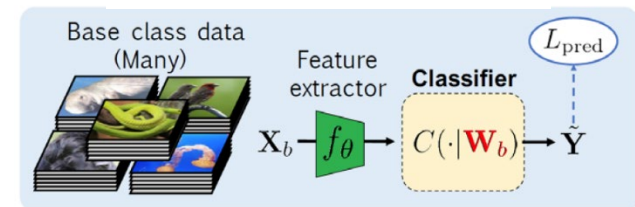
What to Cover Today...

- Recap on Transformer
- Vision & Language
- **Meta-Learning**
 - Meta-Learning for Few-Shot Learning
 - Parametric vs. Non-Parametric Approaches
 - Metric Learning vs. Data Hallucination
 - Advanced Issues in Learning from Small Data

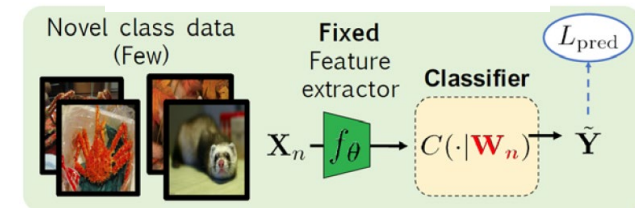


“a corgi wearing a bow tie and a birthday hat”

Meta-Training Stage



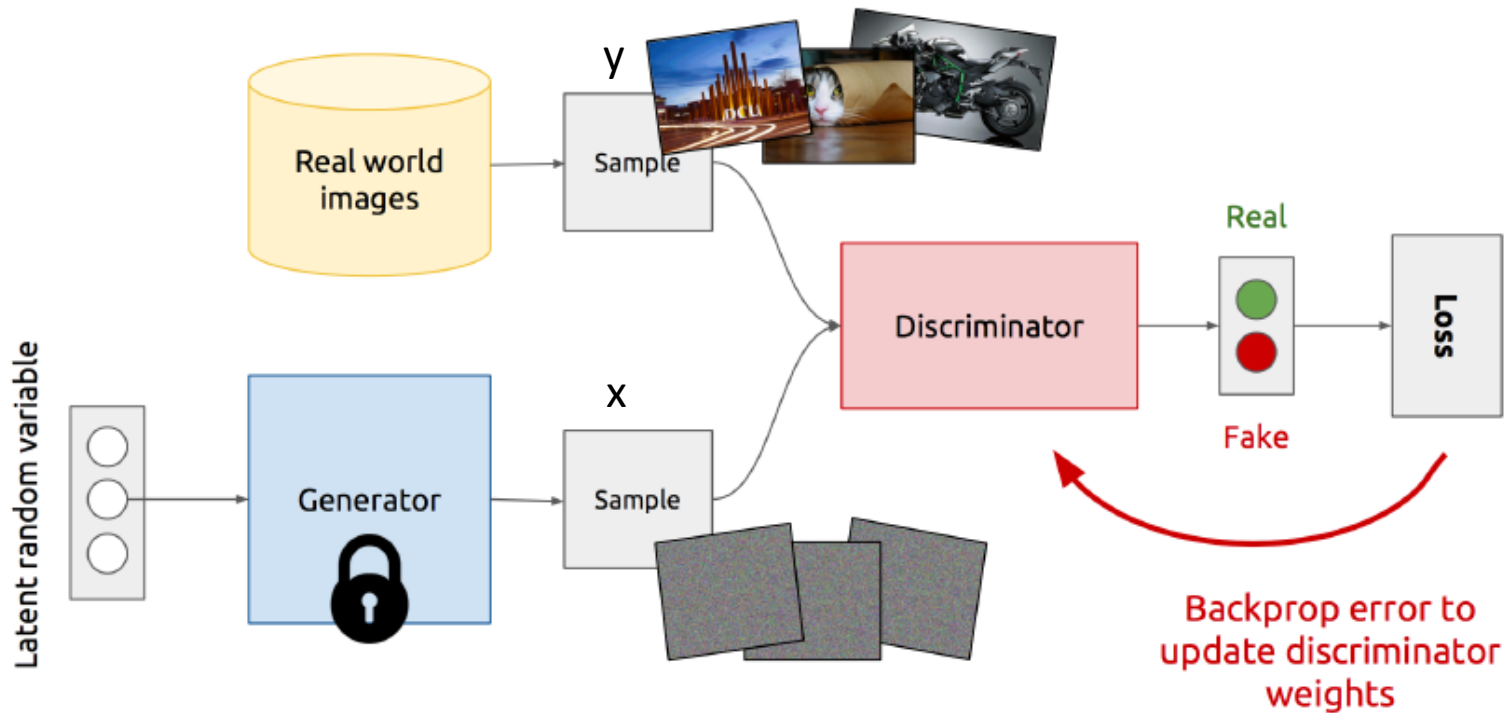
Meta-Testing Stage



A Super Brief Intro/Review for *Generative Adversarial Networks (GAN)*

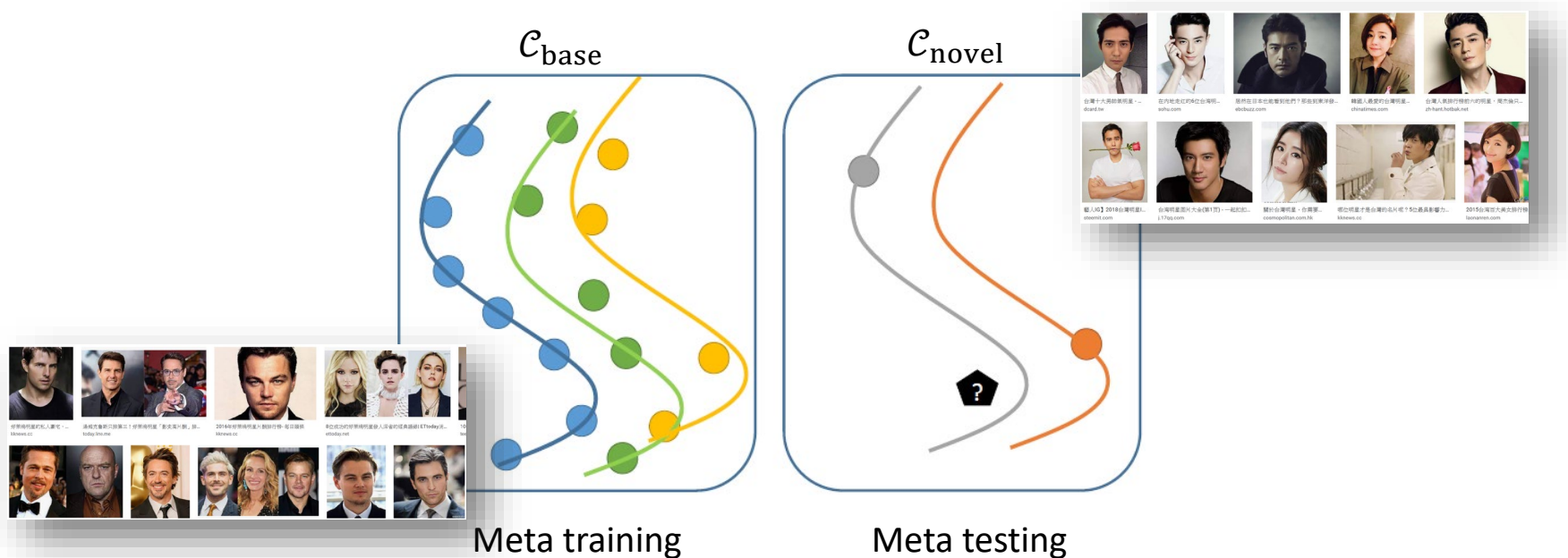
- Design of GAN

- Loss: $\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x)))] + \mathbb{E}[\log D(y)]$



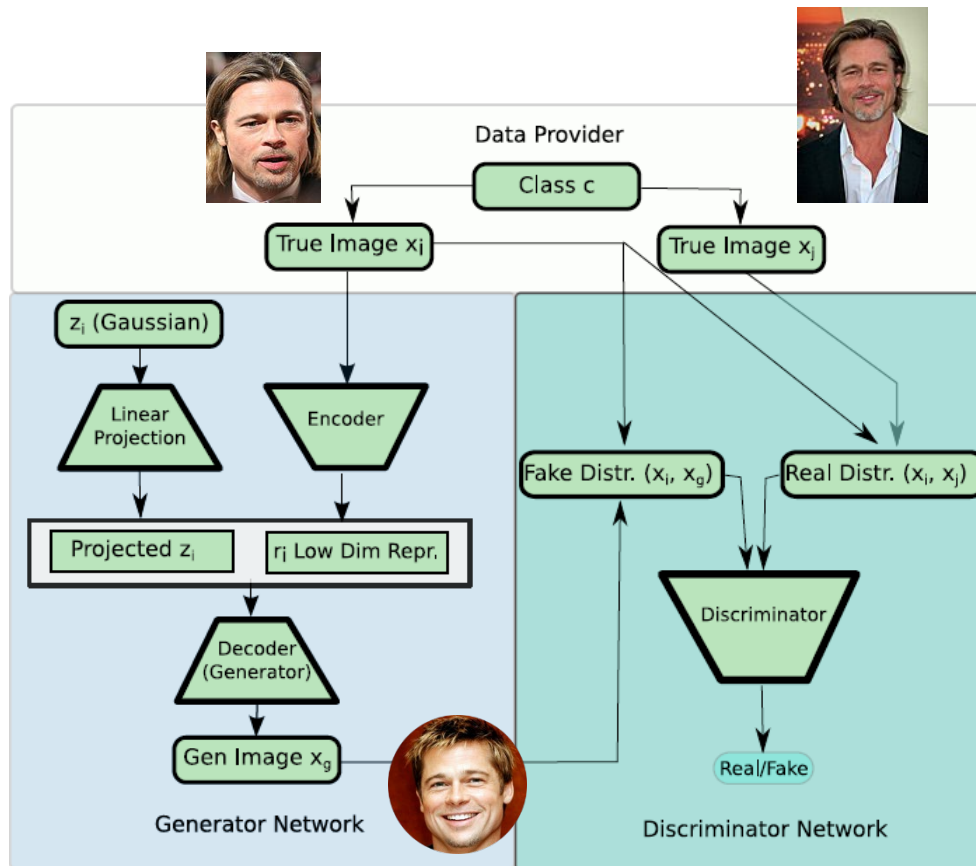
Learn to Augment...Data Hallucination for FSL (1/3)

- Data Hallucination by Conditional GAN
 - Can we learn a model resulting in a desirable **invariance space**, which can be derived by a conditional GAN in the **source domain** ($\mathcal{C}_{\text{base}}$), and apply it to the **target domain** ($\mathcal{C}_{\text{novel}}$)?



- Data Augmentation GAN

(Left) Generator
 $\mathbf{r}_i = \text{Enc}(\mathbf{x}_i)$
 $\mathbf{z}_i \sim N(\mathbf{0}, \mathbf{I})$
 $\mathbf{x}_g = \text{Dec}(\mathbf{z}_i, \mathbf{r}_i)$



Discriminator:

$D(\mathbf{x}_i, \mathbf{x}_j) \rightarrow$ Real pair

$D(\mathbf{x}_i, \mathbf{x}_g) \rightarrow$ Fake pair

Question:

Why not verify \mathbf{x}_j and \mathbf{x}_g ?
i.e., why conditioned on \mathbf{x}_i ?

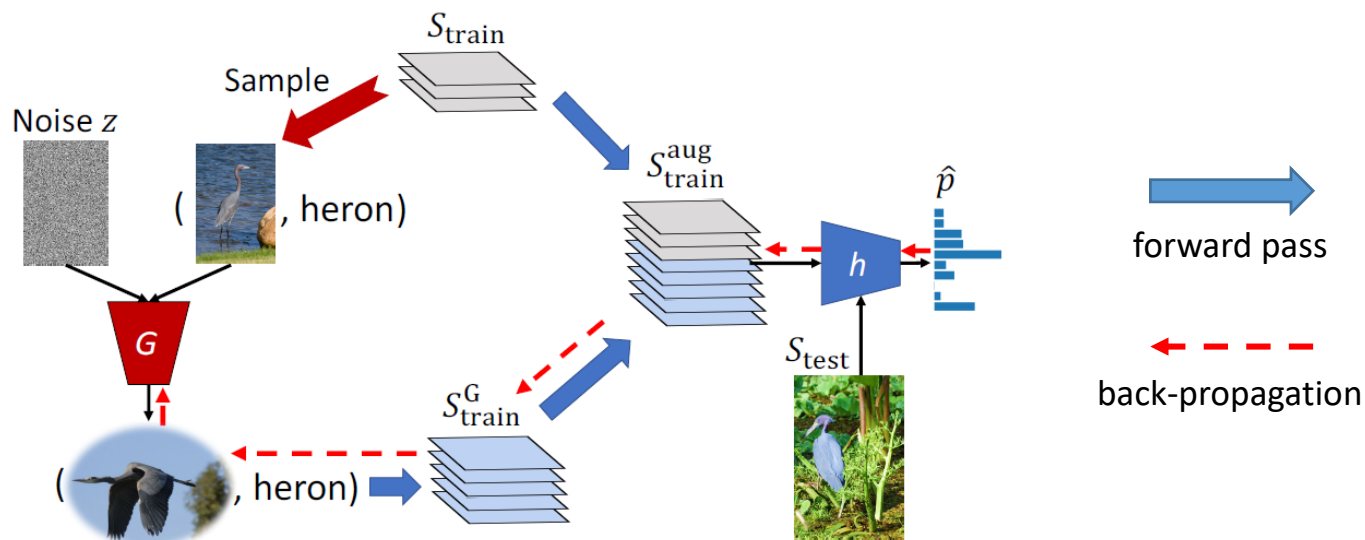
(1) prevent the generator from simply output the original image \mathbf{x}_i
(2) to improve diversity (aka. mode collapse)

\rightarrow (1) or (2) or...?

Learn to Augment...Data Hallucination for FSL (cont'd)

- Jointly Trained Hallucinator

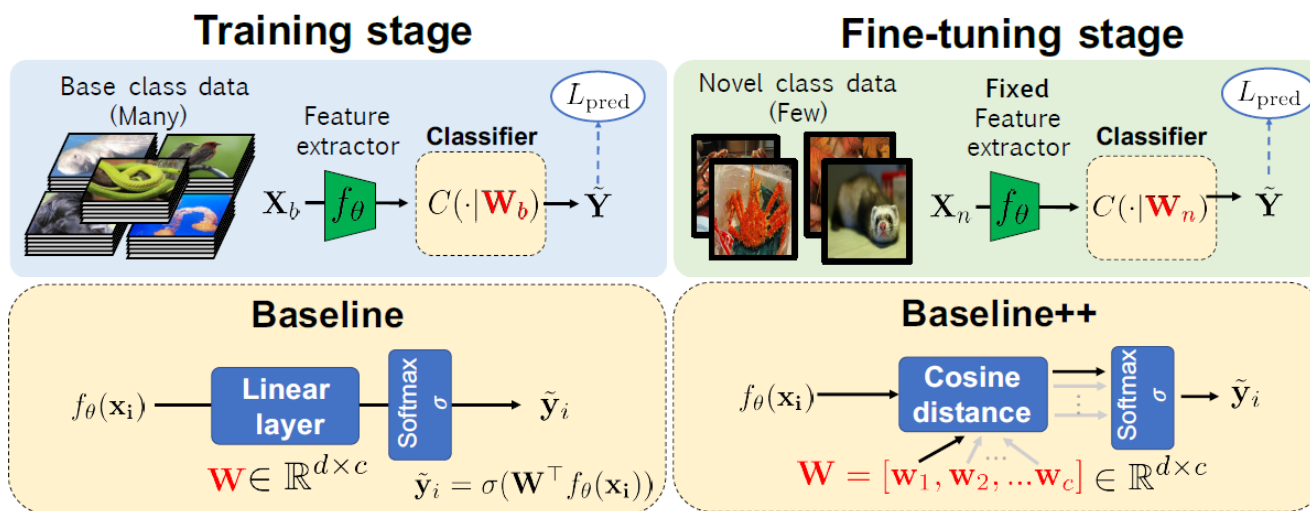
- The hallucinated examples should be **useful** for classification tasks, rather than just being **diverse** or **realistic** (that may fail to improve FSL performances).
- The authors proposed to train a **conditional-GAN-based** data hallucinator ($G(x, z)$) **jointly** with the meta-learning module (h) in an **end-to-end** manner.



Further Remarks:

A Closer Look at FSL (1/3)

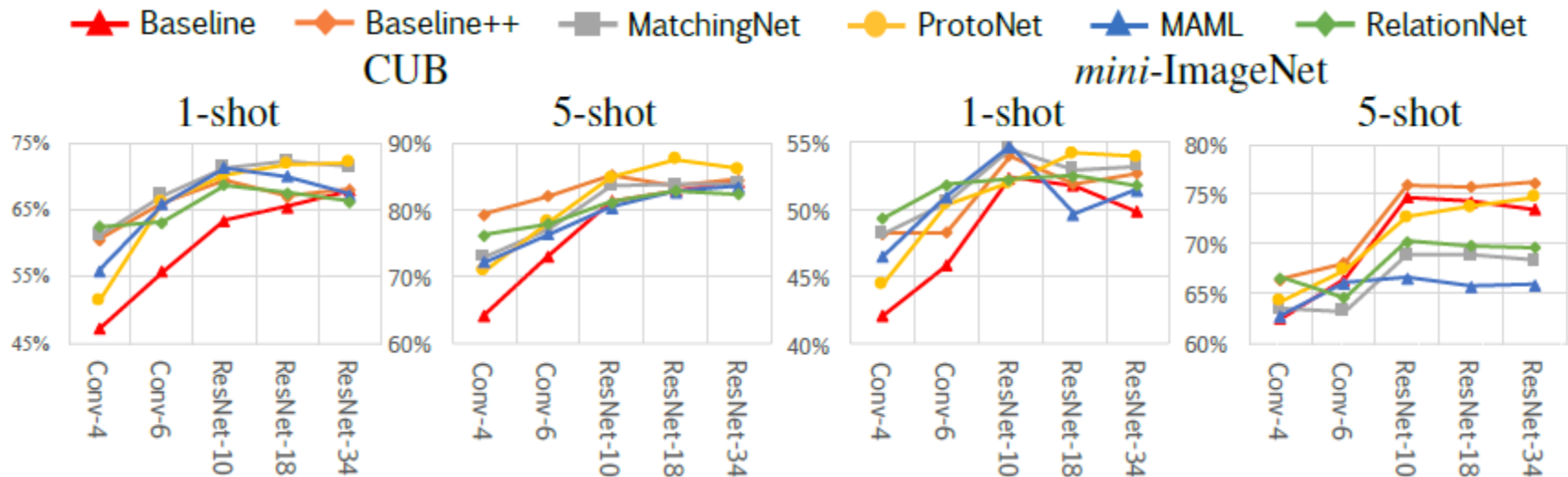
- Idea
 - **Deeper backbones** significantly reduce the gap across existing FSL methods. (with decreased **domain shifts** between base and novel classes)
 - A slightly modified baseline method (**baseline++**) surprisingly achieves competitive performance.
 - Simple baselines (**baseline** and **baseline++**: trained on base and fine-tuned on novel) outperform representative FSL methods when the **domain shift** grows larger.



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

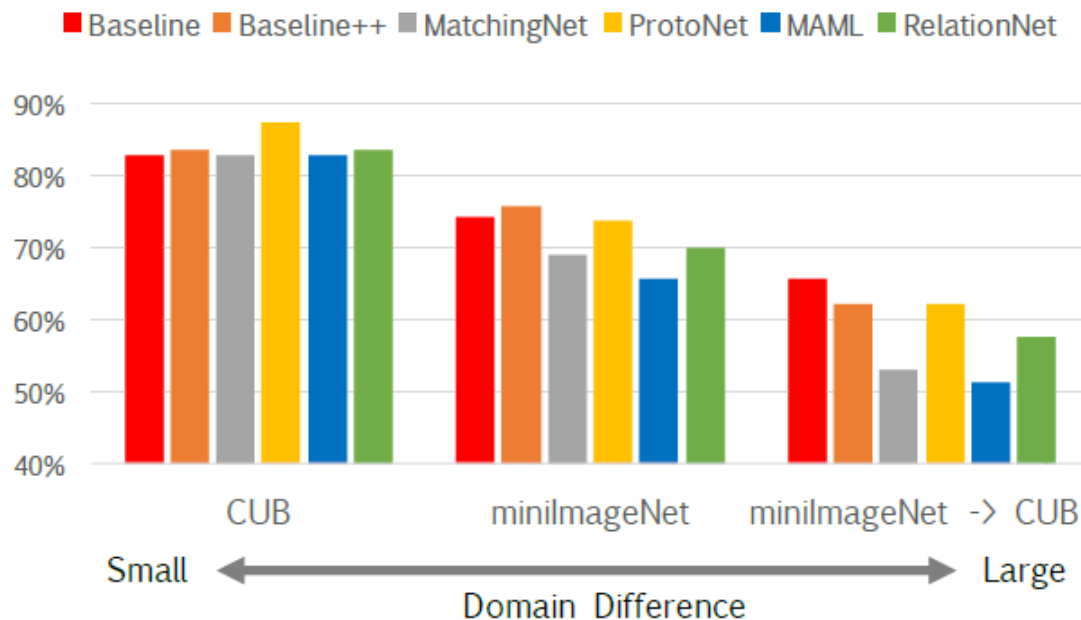
A Closer Look at FSL (2/3)

- 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.



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 \rightarrow CUB) and are inferior to the baseline methods.
 - This highlights the importance of learning to adapt to domain differences in FSL.



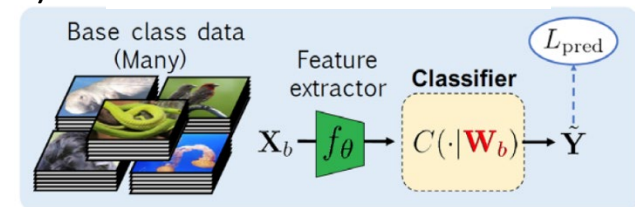
What to Cover Today...

- Recap on Transformer
- Vision & Language
- **Meta-Learning**
 - Meta-Learning for Few-Shot Learning
 - Parametric vs. Non-Parametric Approaches
 - Metric Learning vs. Data Hallucination
 - Advanced Issues in Learning from Small Data
 - Few-Shot Image Segmentation
 - Few-Shot Object Detection (next lecture)
 - Anomaly Detection (next lecture)

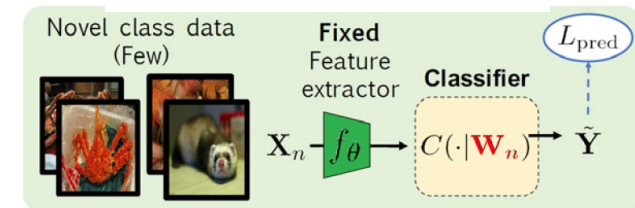


“a corgi wearing a bow tie and a birthday hat”

Meta-Training Stage



Meta-Testing Stage



Semantic Segmentation

- Goal
 - Assign a class label to each pixel in the input image
 - Don't differentiate instances, only care about pixels



Few-Shot Segmentation

- Images of base categories are with pixel-wise ground truth labels, while those of novel classes them are with limited amounts of GT pixel-wise labels.
- A **shared CNN backbone** produces feature maps for both **support** and **query** images.
- **Prototypes** for each class is obtained by **masked pooling** from support feature maps.
- Query feature maps are then compared with the prototypes in a **pixel-by-pixel** fashion.
- Typically, **cosine similarity** is adopted for pixel-wise feature comparison.

