



Topics in AI (CPSC 532S): Multimodal Learning with Vision, Language and Sound



A decorative horizontal bar at the bottom of the slide, composed of five colored segments: light green, medium green, cyan, light blue, and purple.

Lecture 23: Large Scale Visio-Lingual Models (cont.)

Today is our **last lecture** ...

... I hope you enjoyed the class!

(Please do fill out evaluation reports on **Canvas**)

Logistics

- **Assignment 3 & 4** grades (blank output)
- **Assignment 3** grade fixes (out of 100, not 135 — fixed)
- Be careful of looking at Average Grade on Canvas
- **Assignment 5** is due **today** (can hand in by Friday)
- **Research Paper Presentations** (all are in)
- **Reading Reviews** (some 3 & 4 outstanding)

Logistics

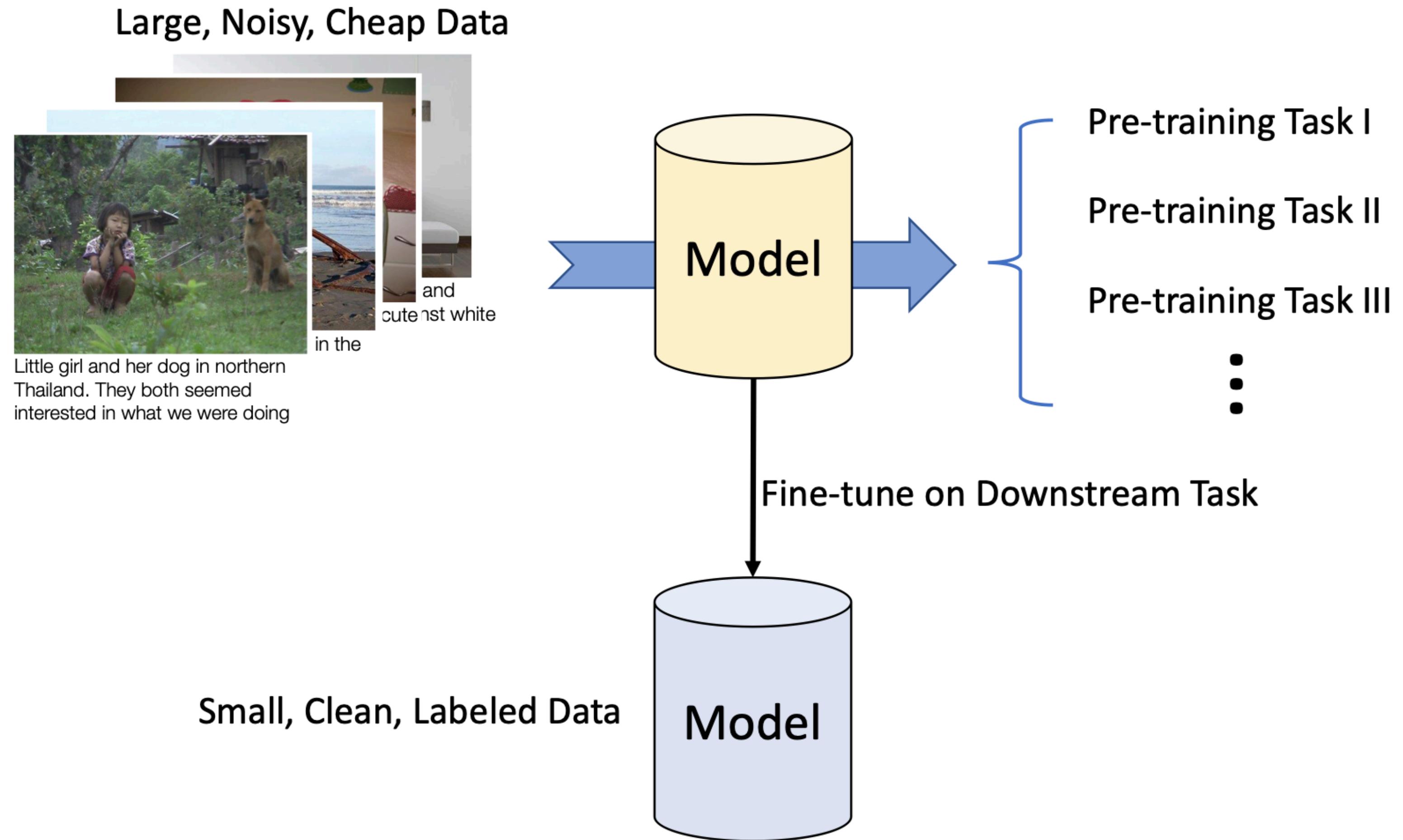
My todo's

- List of paper presentations
- Grades for **Paper Readings and Presentations**
- Grades for **Assignment 5**

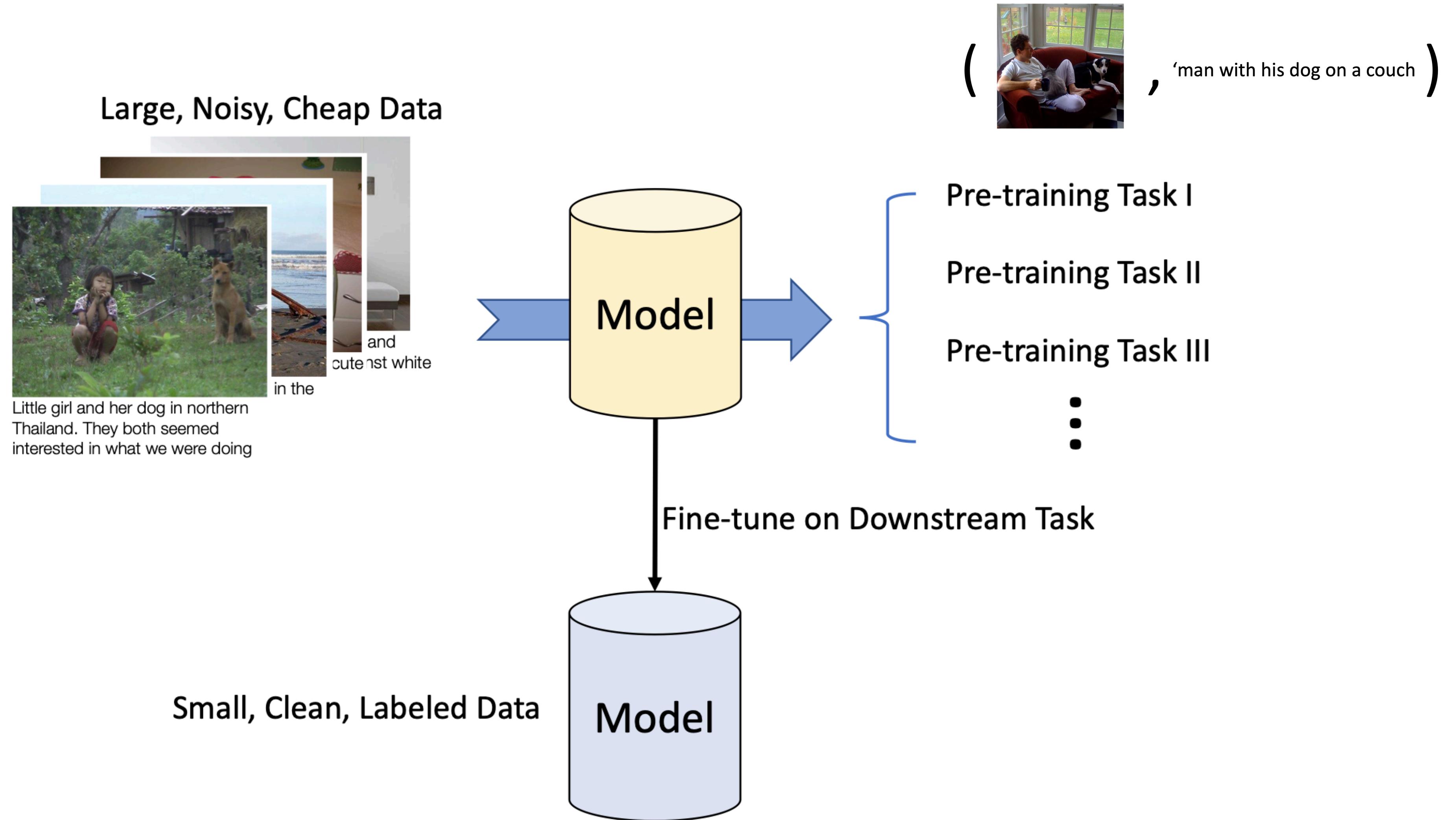
Your todo's

- Hand in **Assignment 5**
- Hand in **Paper Readings 3 & 4** (if you have not done this yet)
- Final Project Presentations on Tuesday next week 12-3pm

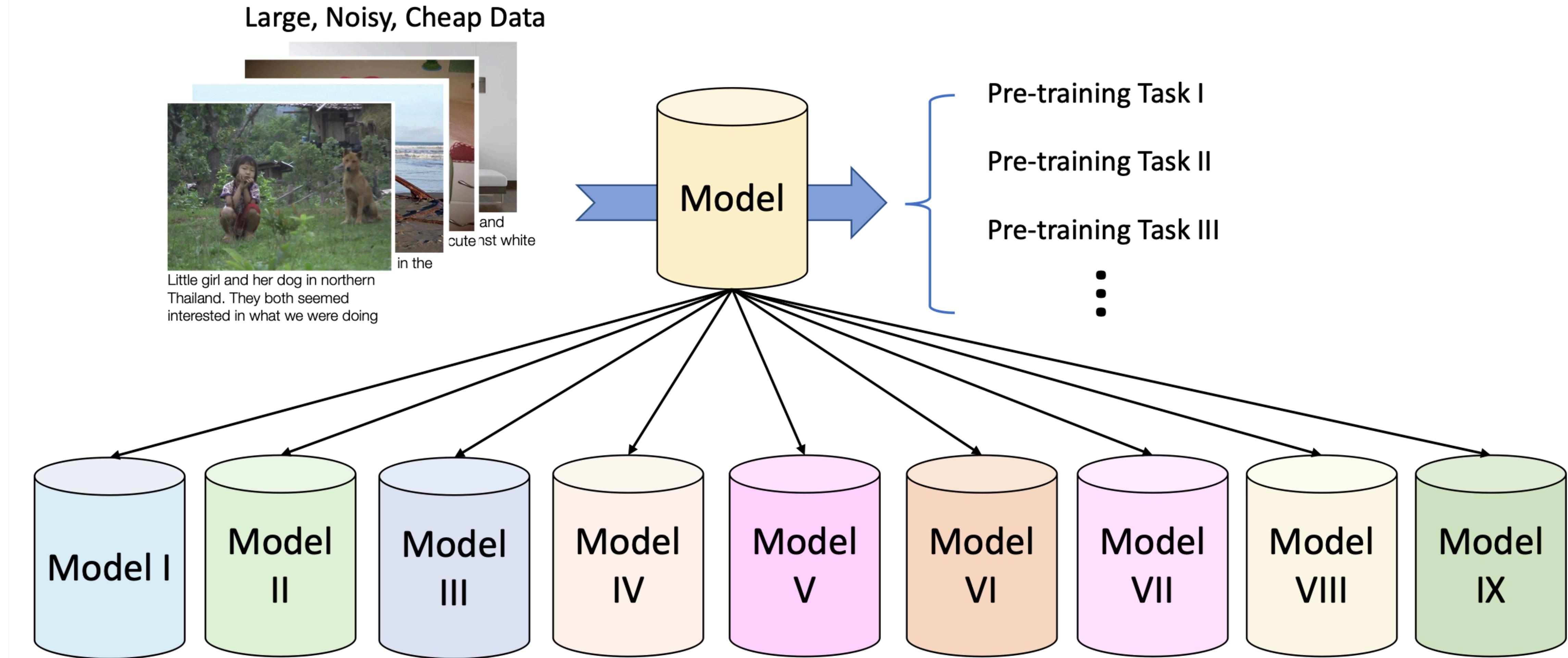
Pre-training and Foundational Models



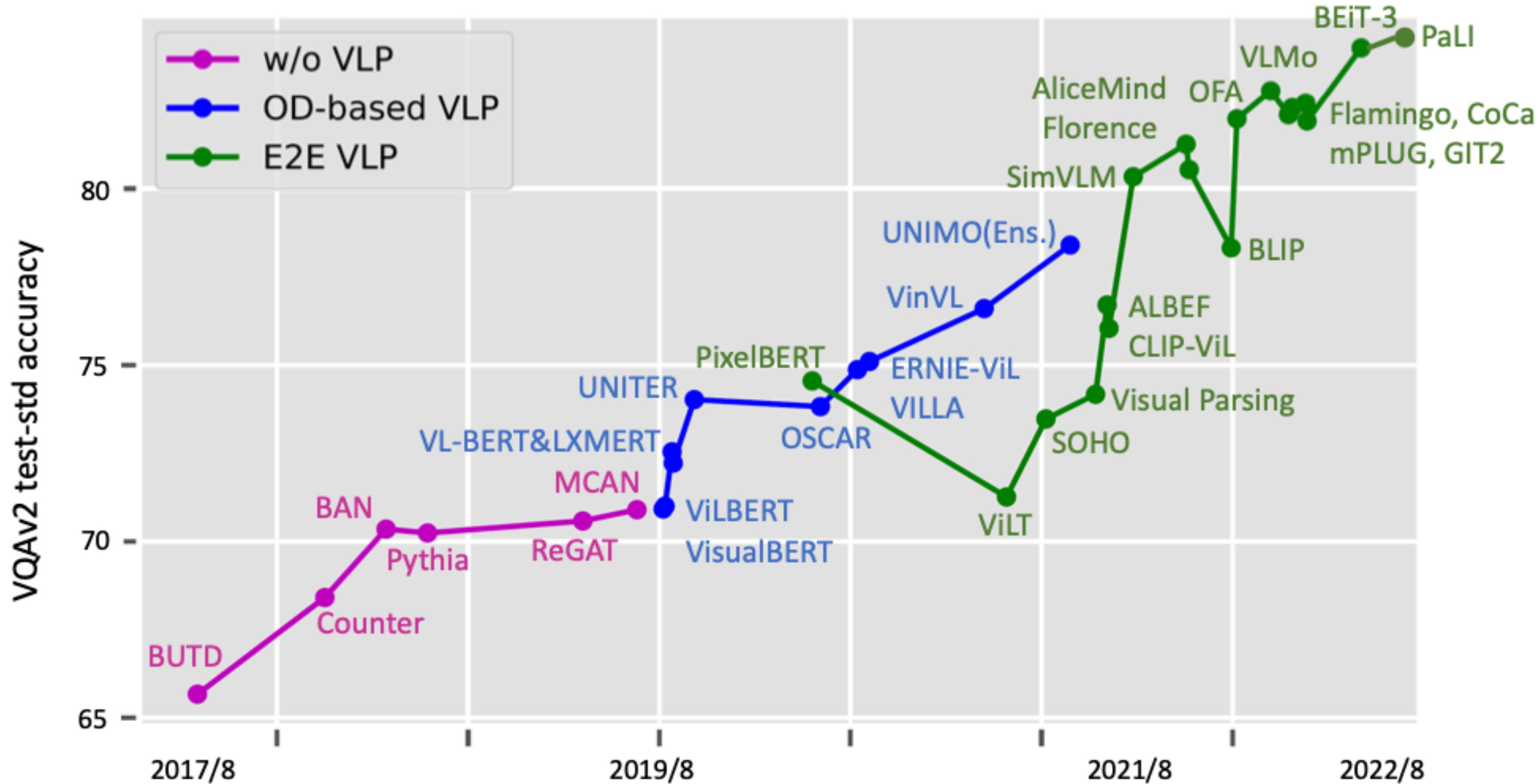
Pre-training and Foundational Models



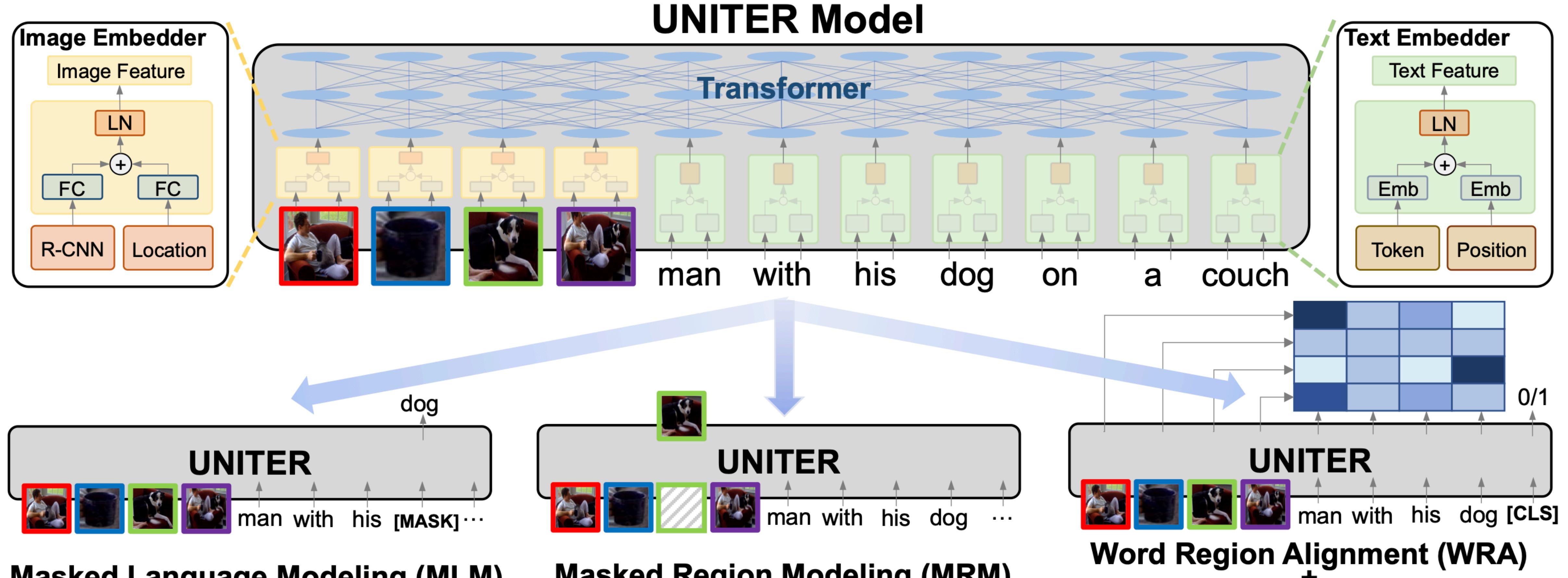
Pre-training and Foundational Models



Recent History of Visio-Lingual Models



UNITER: UNiversal Image-TExt Representation Learning



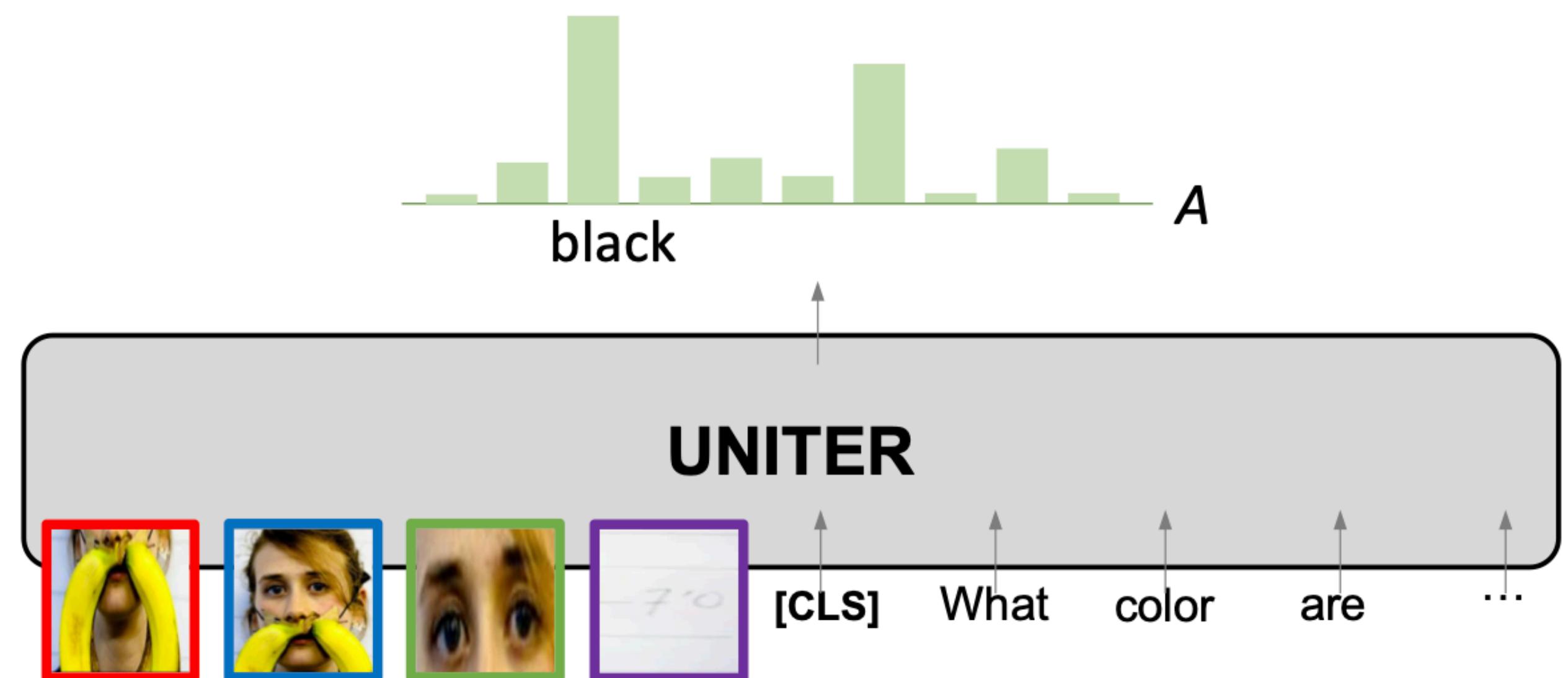
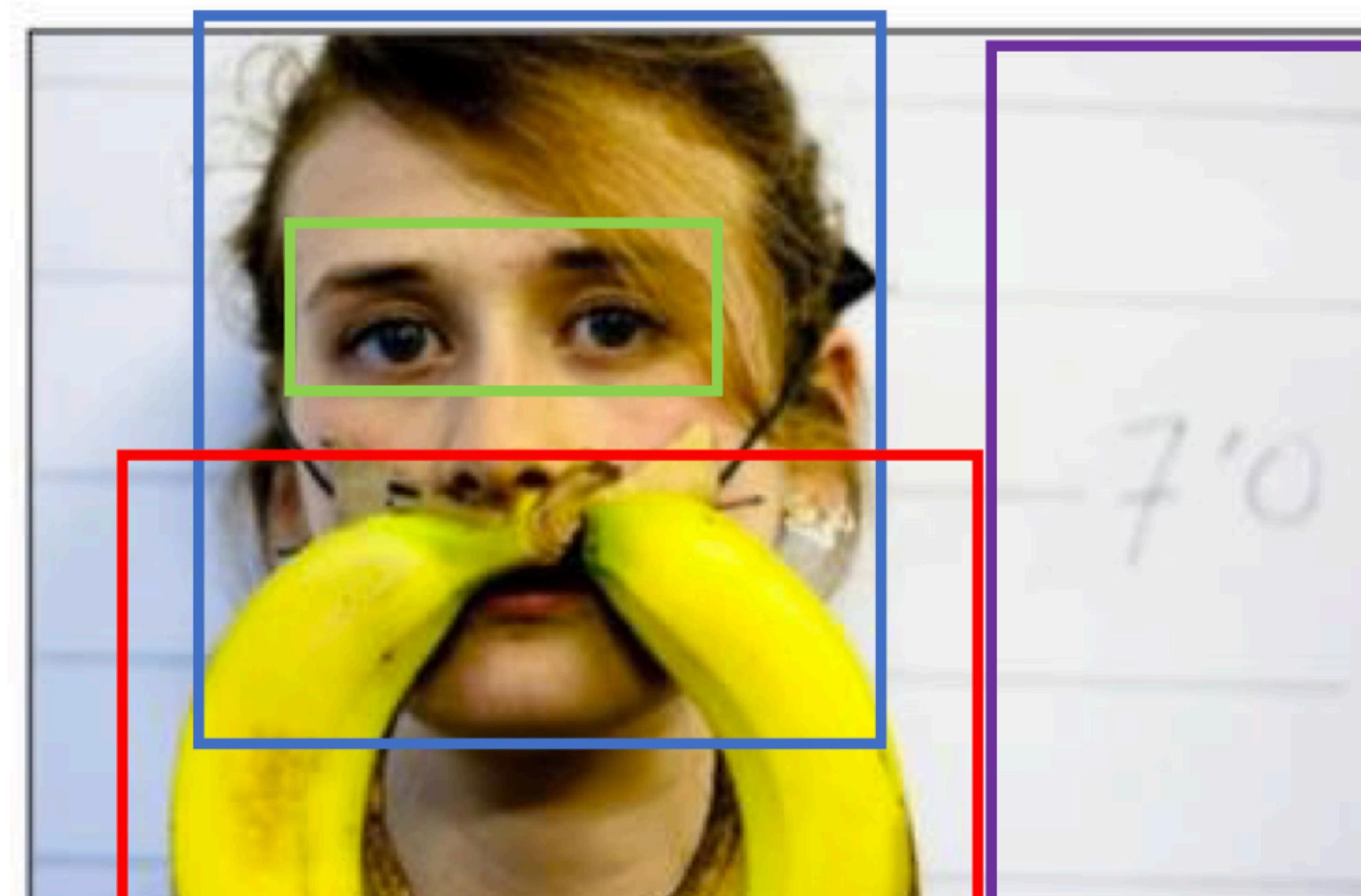
Masked Language Modeling (MLM)

Masked Region Modeling (MRM)

- Masked Region Feature Regression (MRFR)
- Masked Region Classification (MRC)
- Masked Region Classification with KL-Divergence (MRC-kl)

Word Region Alignment (WRA) + Image-Text Matching (ITM)

Downstream Task 1: Visual Question Answering



Downstream Task 2: Visual Entailment



Premise

+

- Two woman are holding packages.
- The sisters are hugging goodbye while holding to go packages after just eating lunch.
- The men are fighting outside a deli.

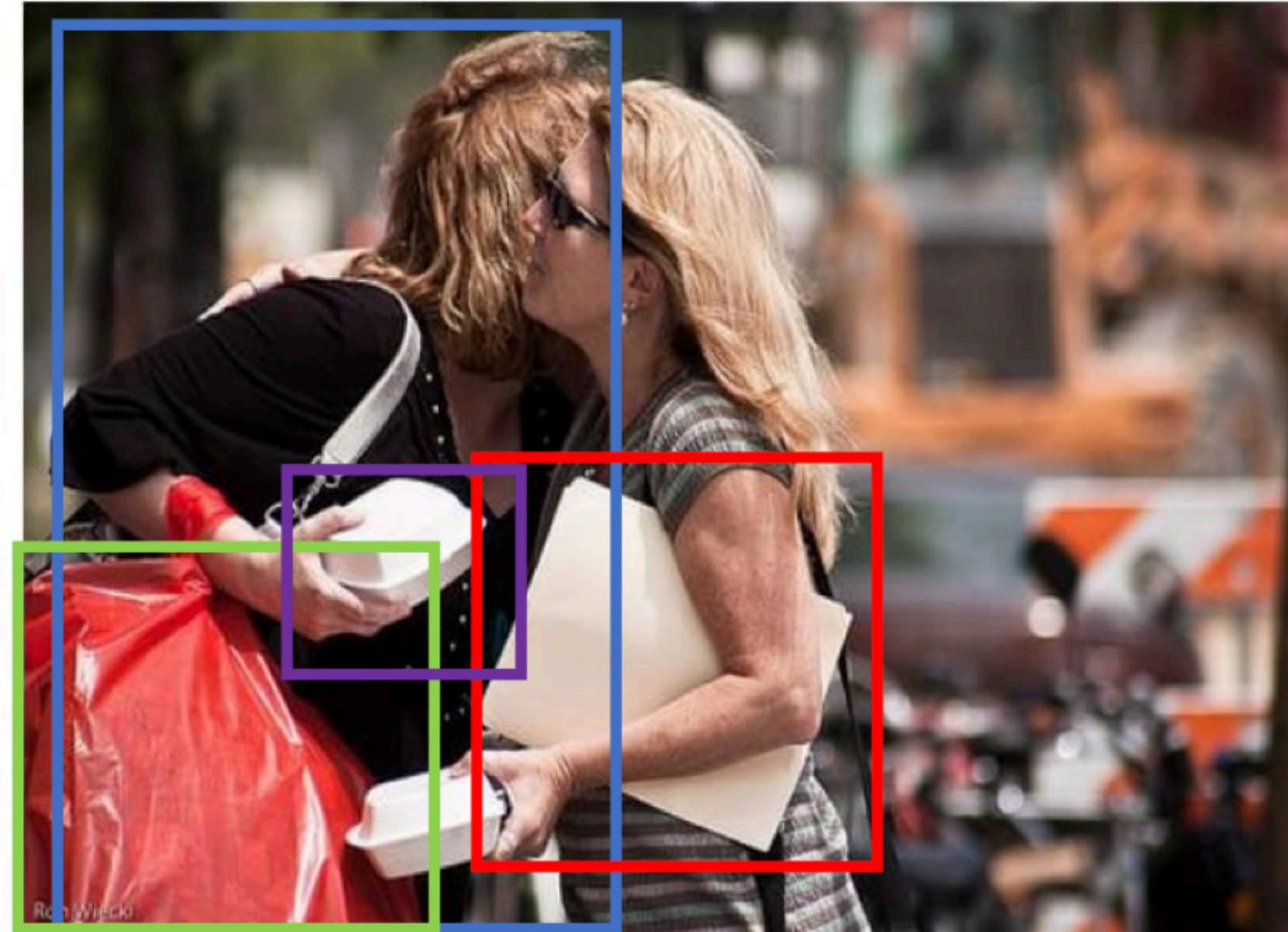
=

Hypothesis

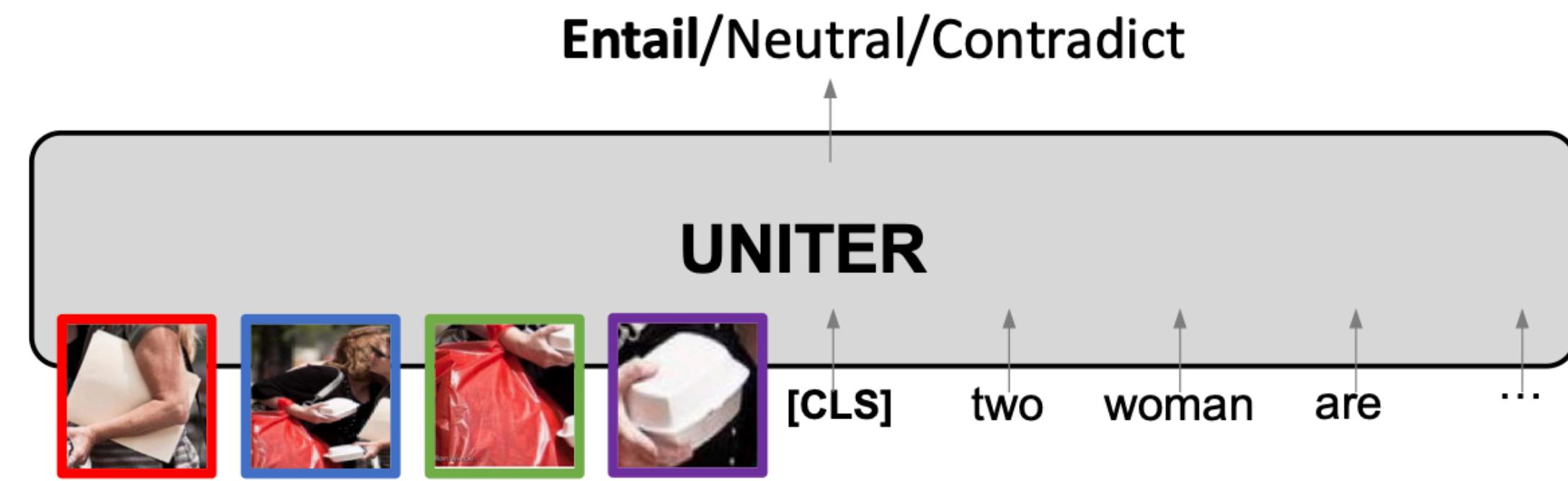
- Entailment
- Neutral
- Contradiction

Answer

Downstream Task 2: Visual Entailment



Two woman are holding packages.



Downstream Task 3: Natural Language for **Visual Reasoning**



The left image contains twice the number of dogs as the right image, and at least two dogs in total are standing.

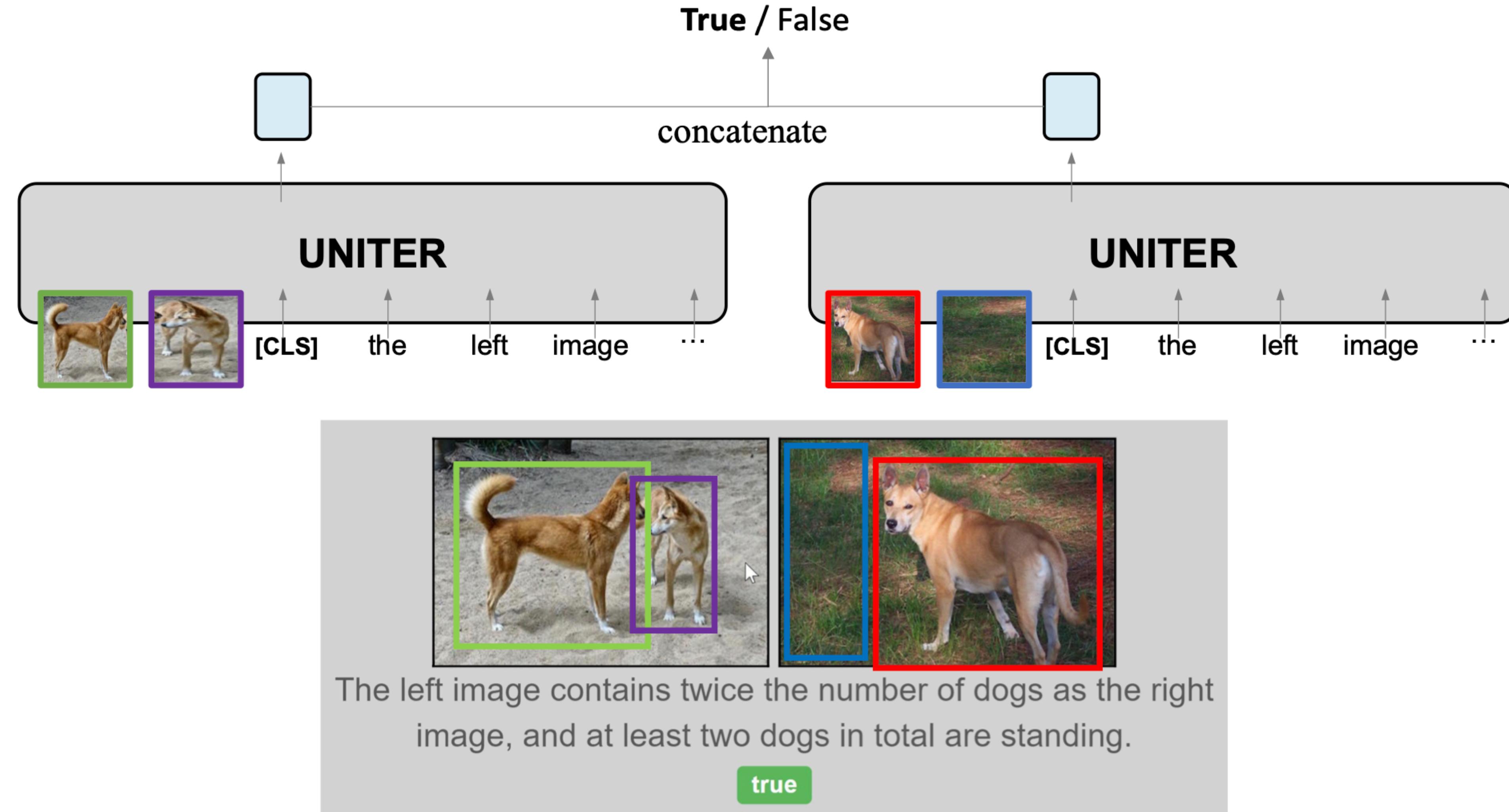
true



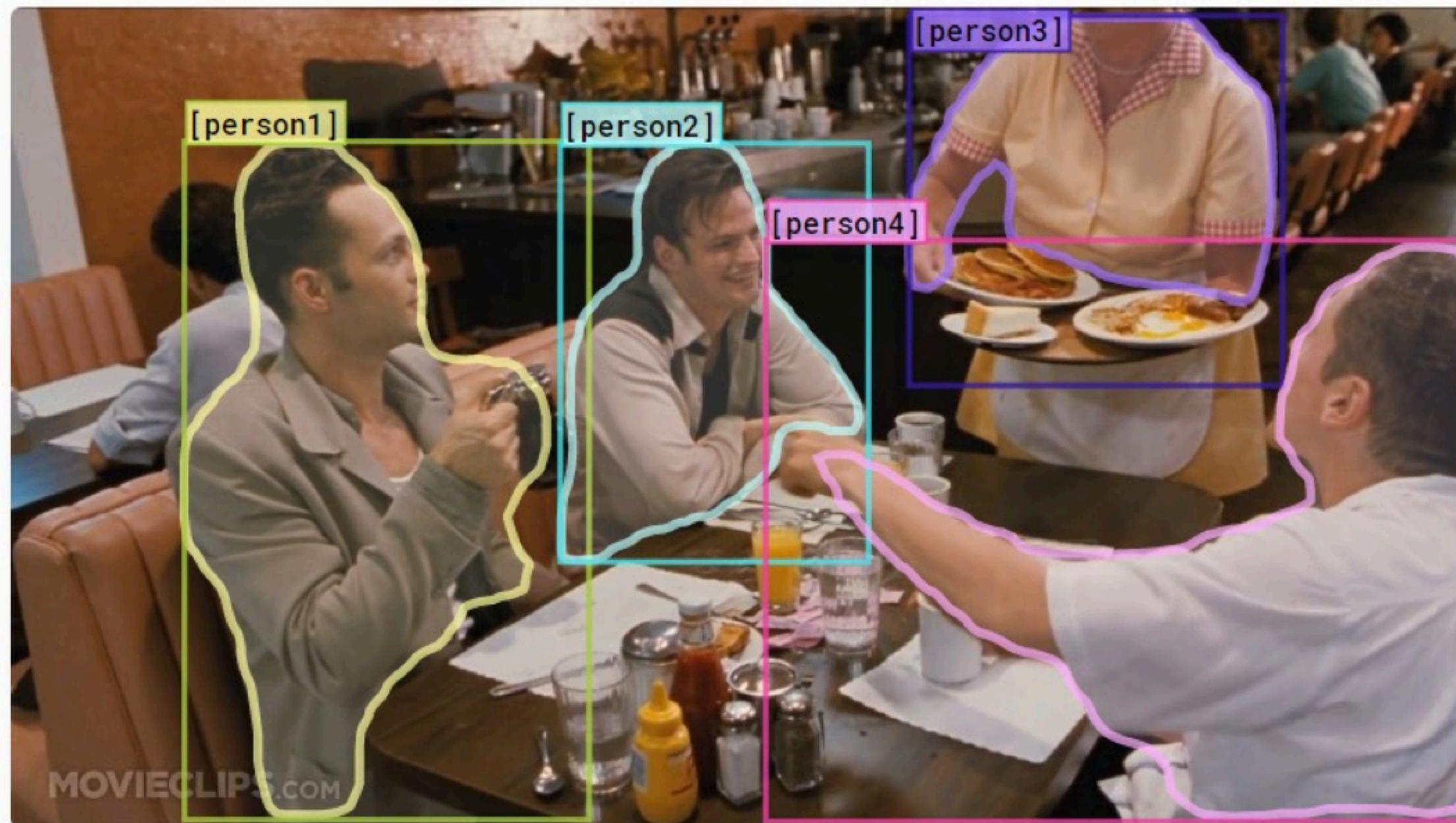
One image shows exactly two brown acorns in back-to-back caps on green foliage.

false

Downstream Task 3: Natural Language for **Visual Reasoning**



Downstream Task 4: Visual Commonsense Reasoning



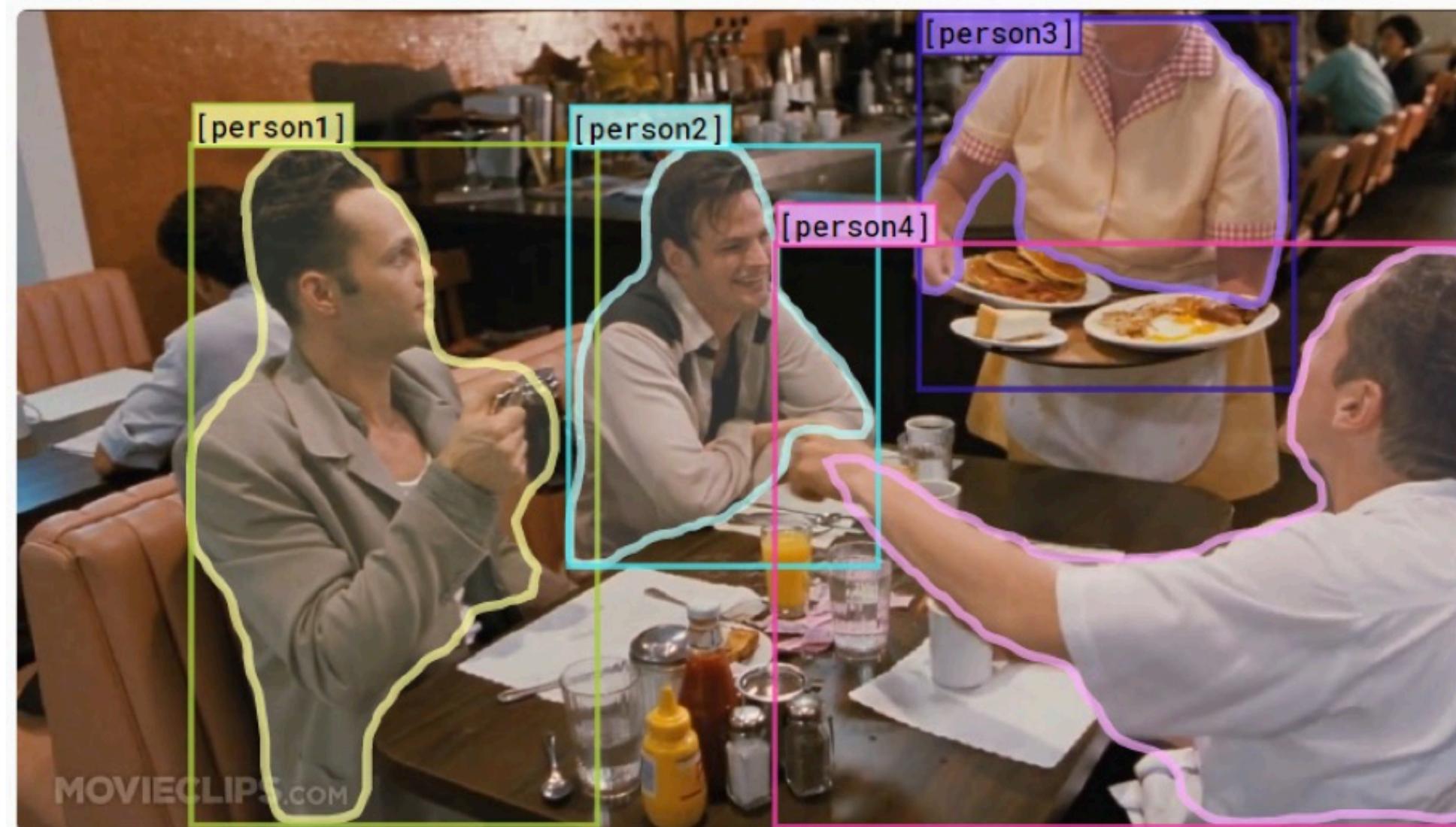
Why is [person4] pointing at [person1]?

- a) He is telling [person3] that [person1] ordered the pancakes.
- b) He just told a joke.
- c) He is feeling accusatory towards [person1].
- d) He is giving [person1] directions.

I choose (a) because:

- a) [person1] has the pancakes in front of him.
- b) [person4] is taking everyone's order and asked for clarification.
- c) [person3] is looking at the pancakes and both she and [person2] are smiling slightly.
- d) [person3] is delivering food to the table, and she might not know whose order is whose.

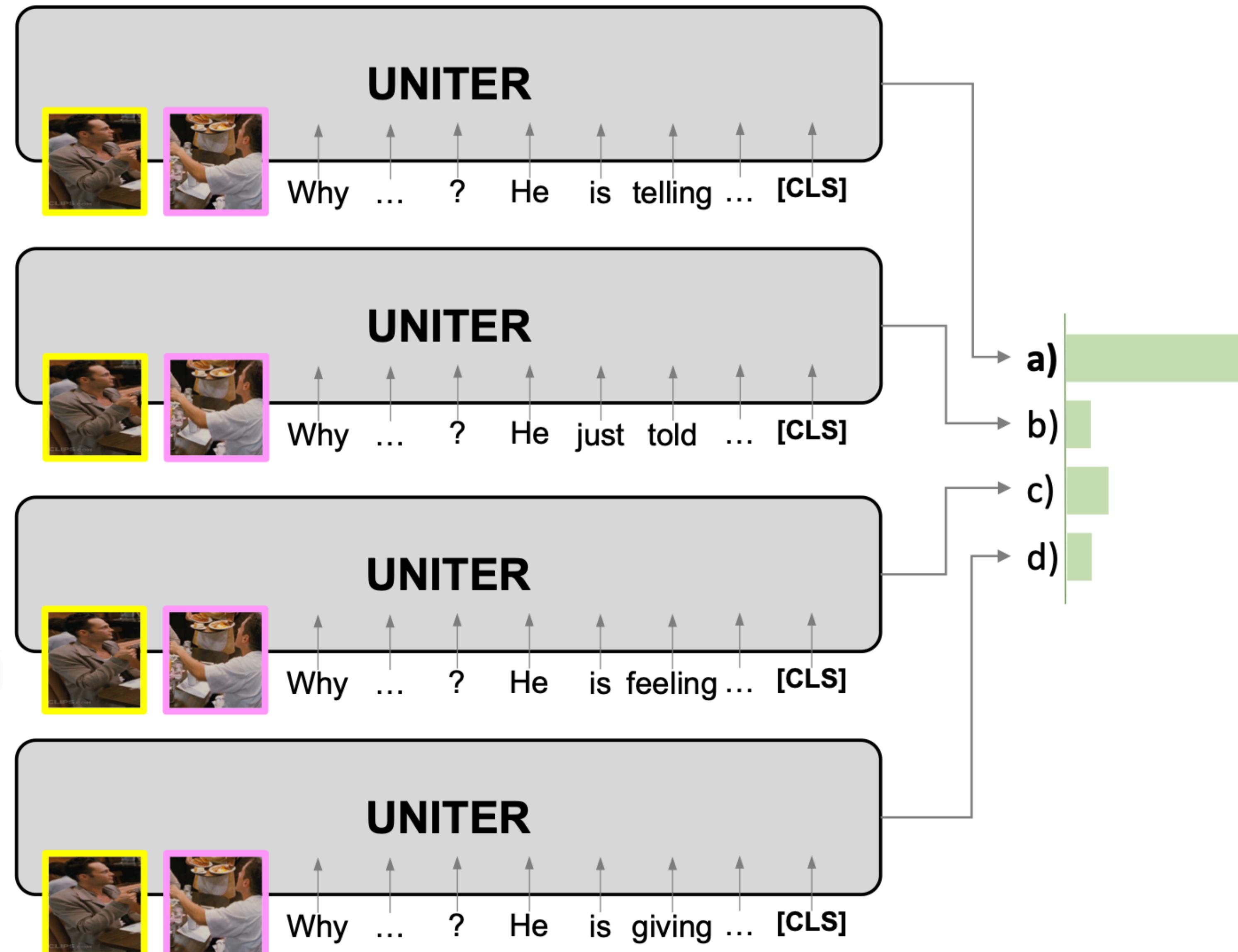
Downstream Task 4: Visual Commonsense Reasoning



Why is [person4] pointing at [person1]?

- a) He is telling [person3] that [person1] ordered the pancakes.
- b) He just told a joke.
- c) He is feeling accusatory towards [person1].
- d) He is giving [person1] directions.

Slide from Zhe Gan

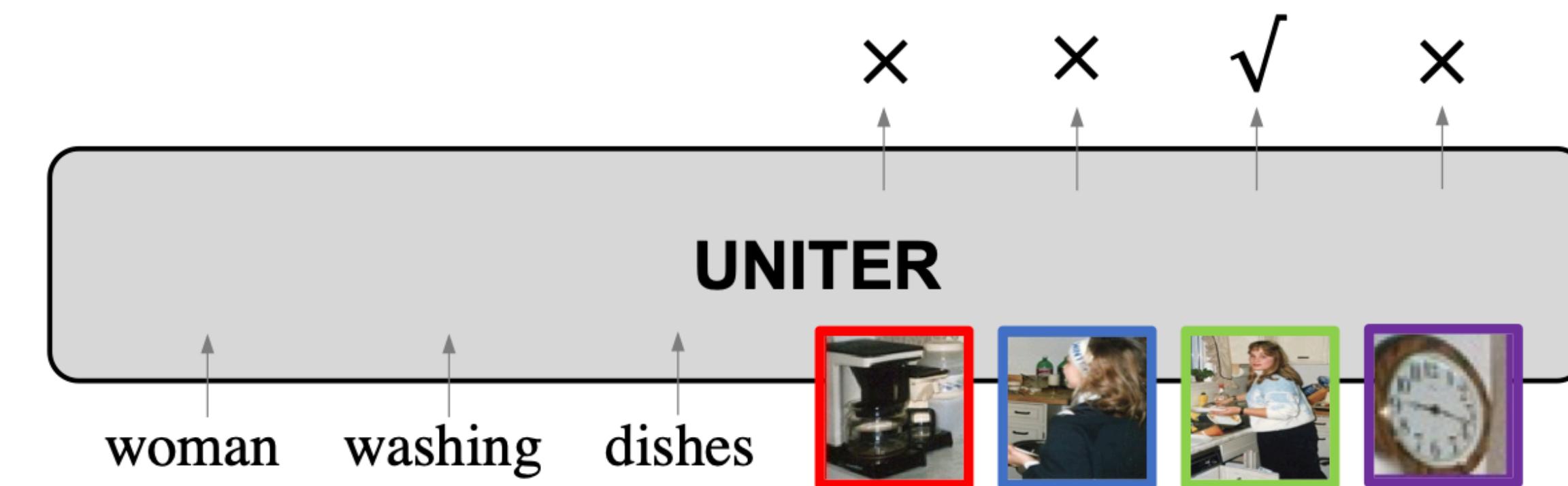
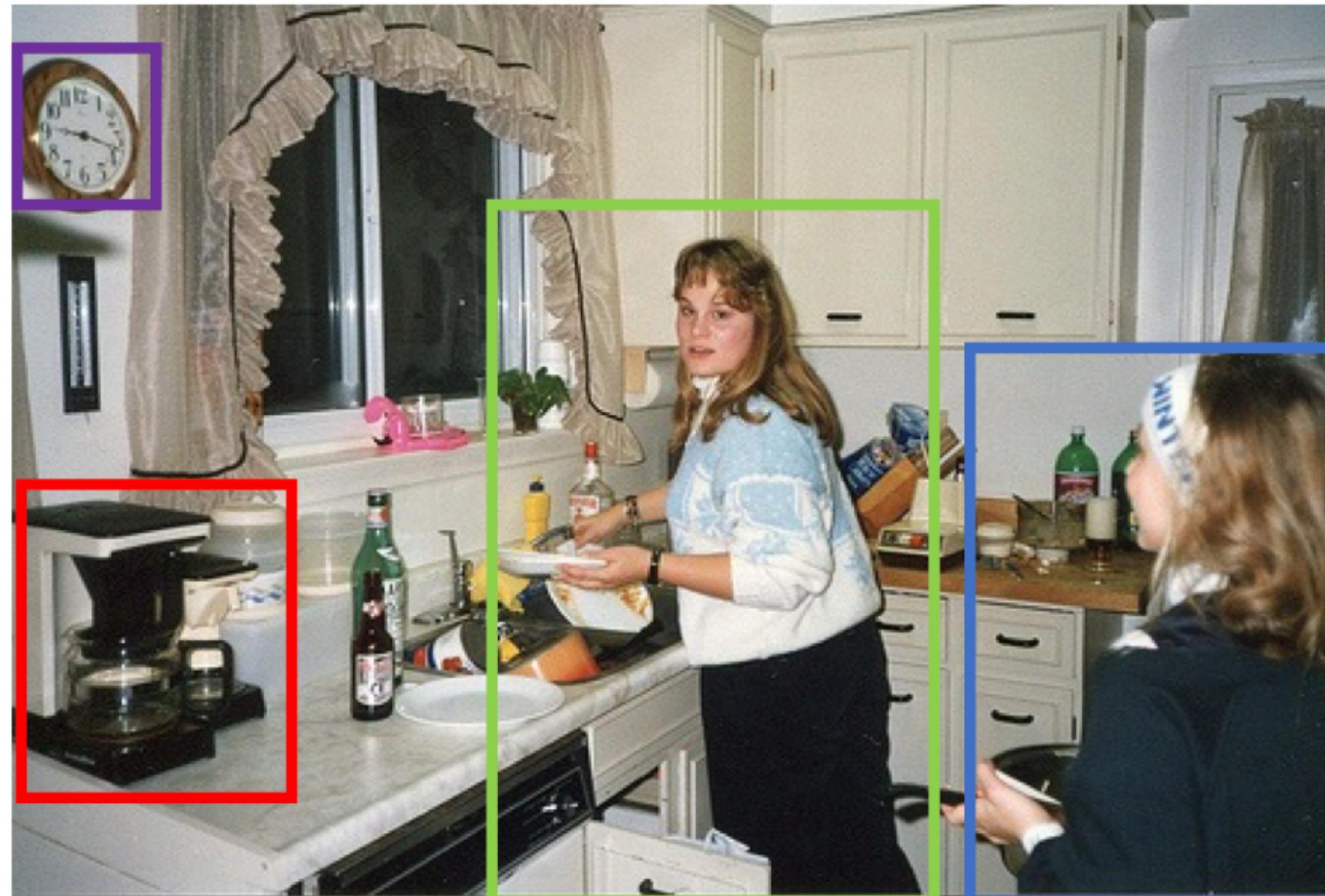


Downstream Task 5: Referring Expression Comprehension (Grounding)



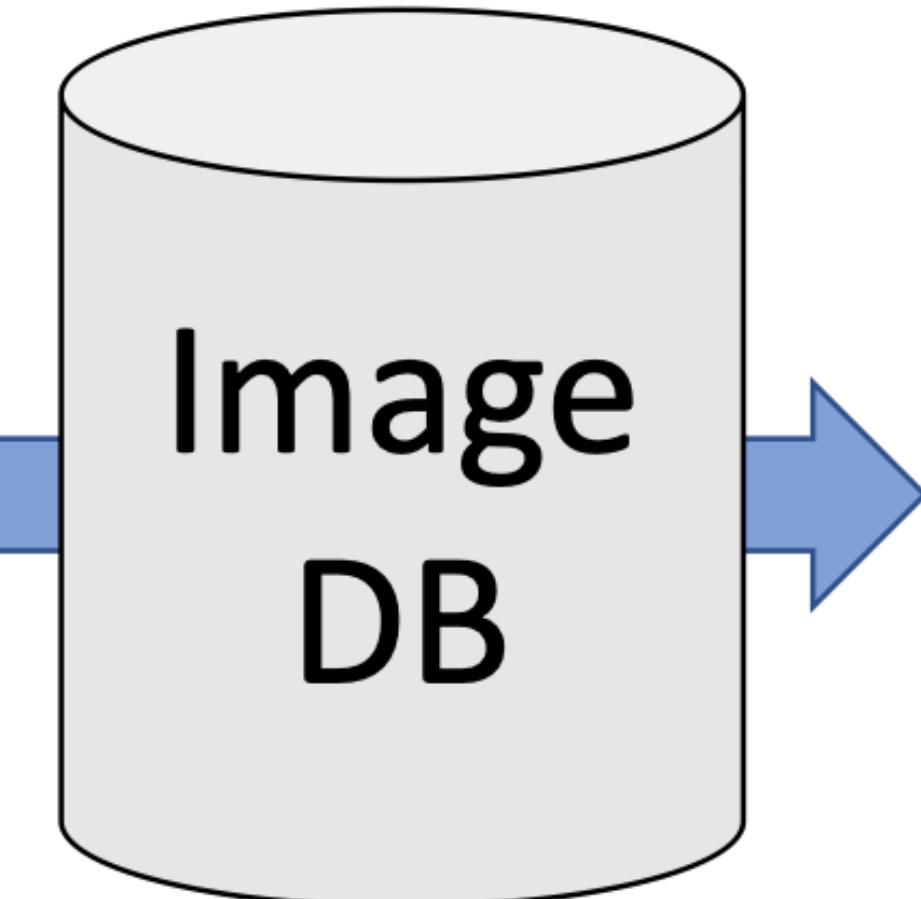
woman washing dishes

Downstream Task 5: Referring Expression Comprehension (Grounding)



Downstream Task 6: Image-Text Retrieval

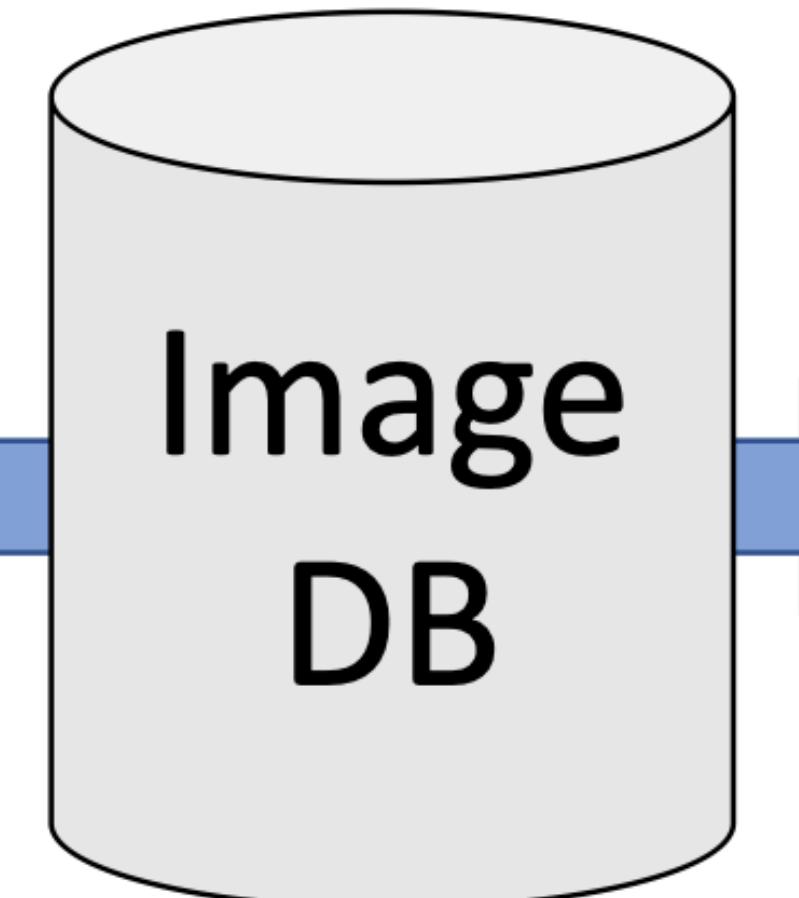
“a girl with a cat on grass”



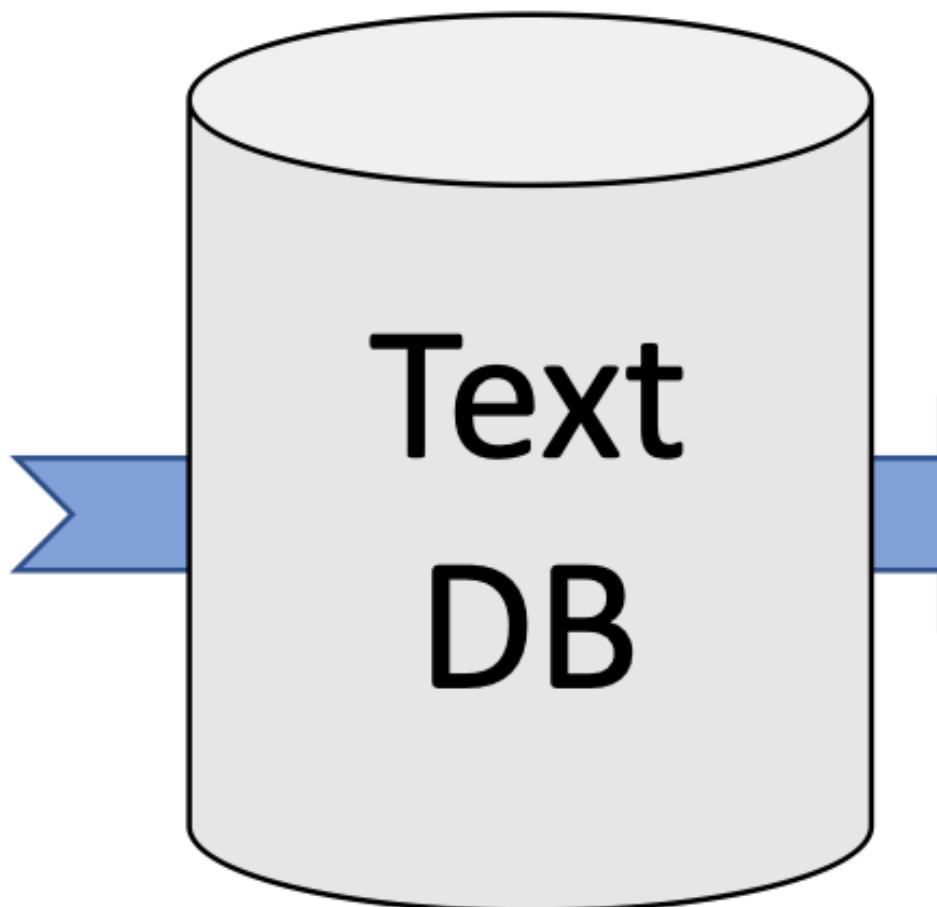
...

Downstream Task 6: Image-Text Retrieval

“a girl with a cat on grass” ➔



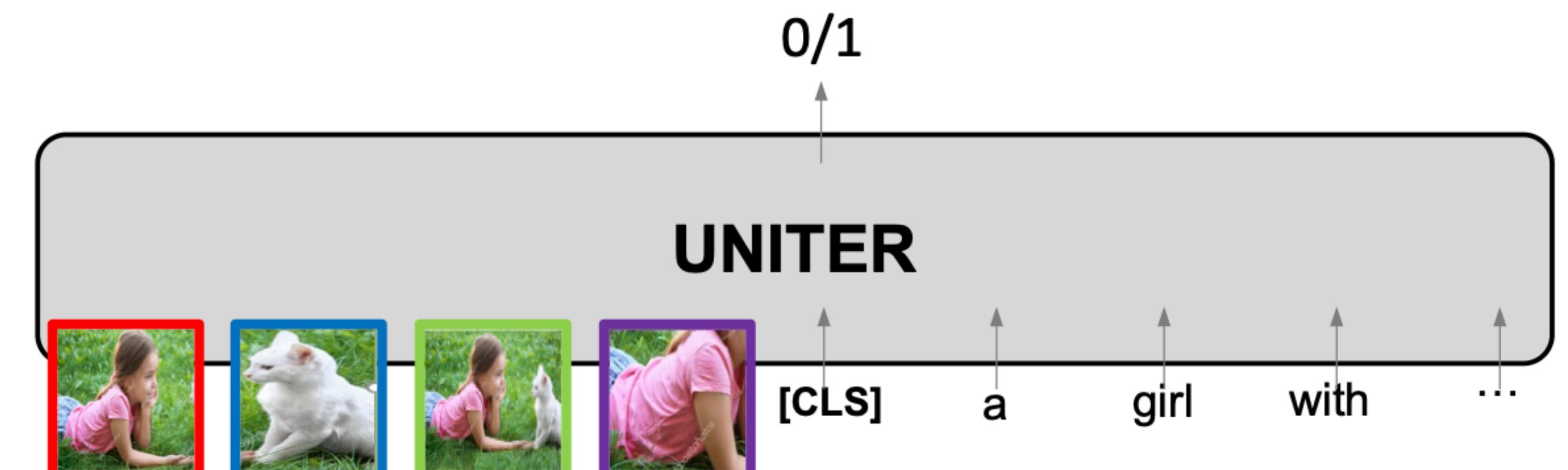
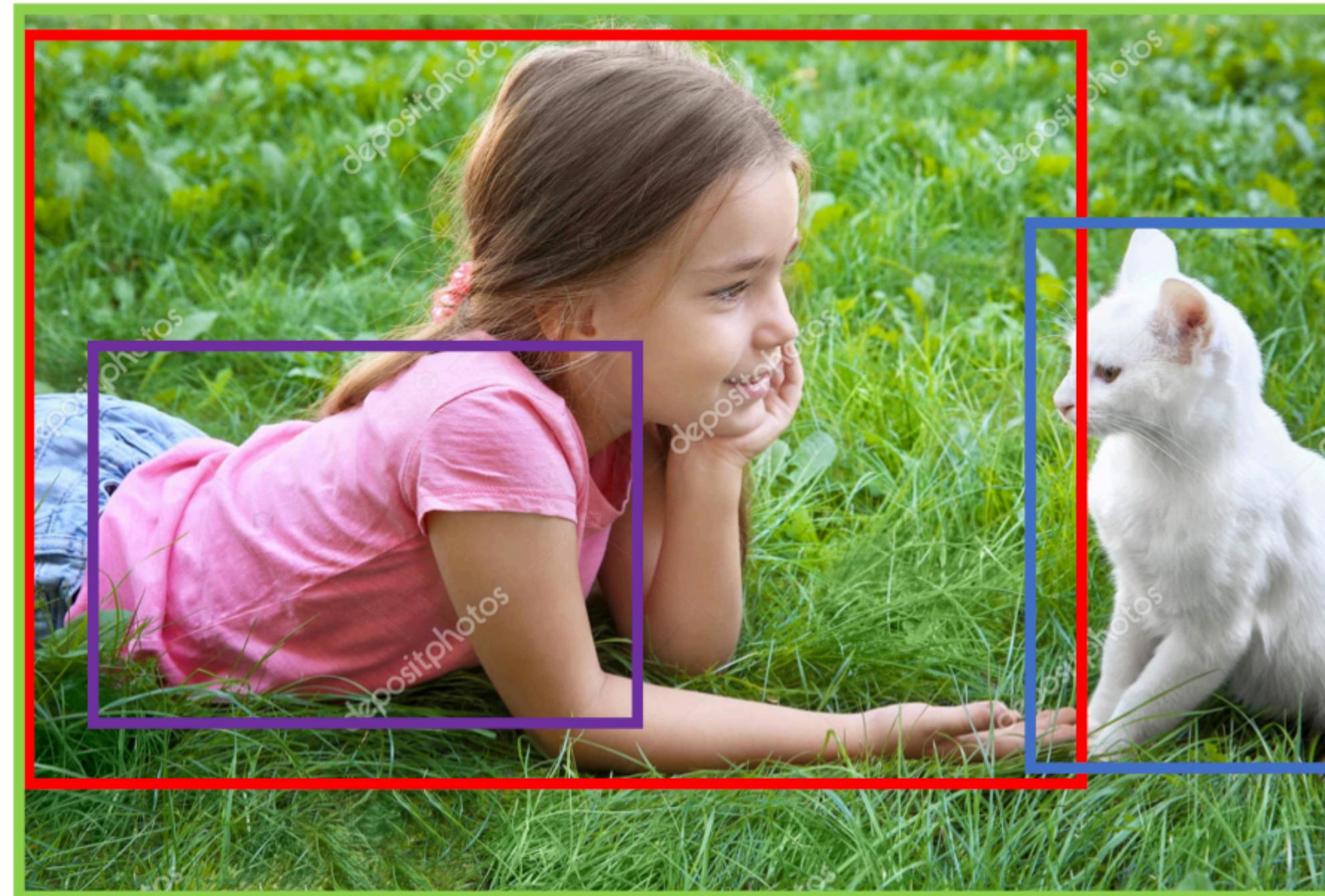
...



“four people with ski poles in their hands in the snow”
“four skiers hold on to their poles in a snowy forest”
“a group of young men riding skis”
“skiers pose for a picture while outside in the woods”
“a group of people cross country skiing in the woods”

⋮

Downstream Task 6: Image-Text Retrieval



VILLA: Vision-and-Language Large-scale Adversarial Training

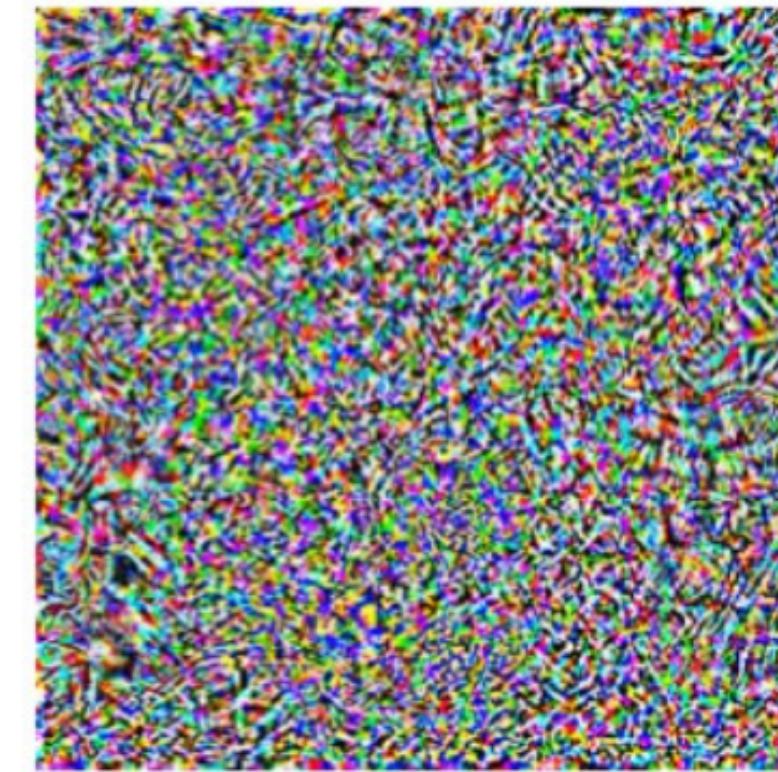
Preliminary: Adversarial Attacks

- Neural Networks are prone to label-preserving adversarial examples

Computer Vision:



$$+ 0.005 \times$$



=



Natural Language
Processing:

Original: What is the oncorhynchus also called? **A:** chum salmon

Changed: **What's** the oncorhynchus also called? **A:** **keta**

(b) Example for (*WP is* → *WP's*)

Original: How long is the Rhine?
A: 1,230 km

Changed: **How long is the Rhine??**
A: **more than 1,050,000**

(c) Example for (? → ??)

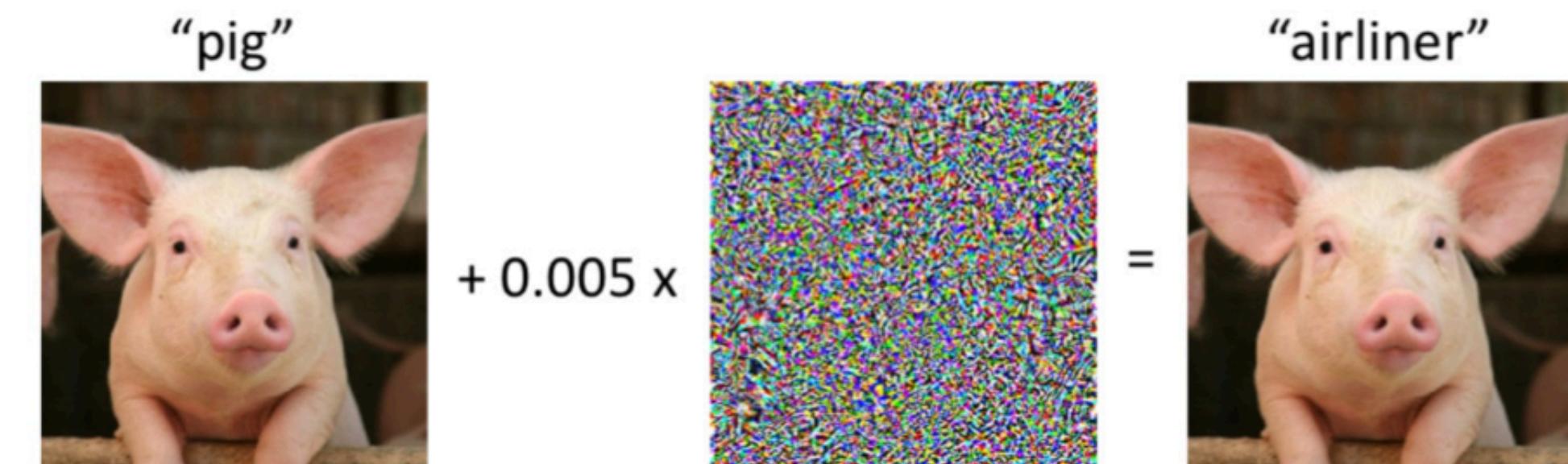
[1] Explaining and harnessing adversarial examples. *arXiv:1412.6572*

[2] Semantically equivalent adversarial rules for debuaaina nlp models. *ACL (2018)*

Preliminary: Adversarial Training

- A min-max game to harness adversarial examples

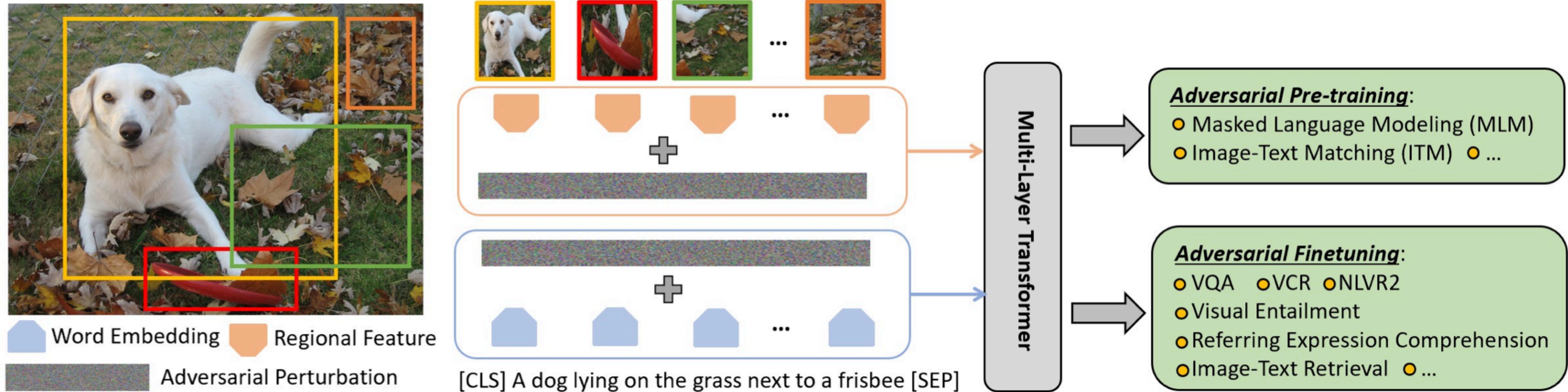
$$\min_{\theta} \mathbb{E}_{(x,y) \sim \hat{\mathcal{D}}} \left[\max_{\delta \in S} \mathcal{L}(x + \delta, y; \theta) \right]$$



- Use adversarial examples as additional training samples
 - On one hand, we try to find perturbations that maximize the empirical risk
 - On the other hand, the model tries to make correct predictions on adversarial examples
- *What doesn't kill you makes you stronger!*

VILLA: Vision-and-Language Large-scale Adversarial Training

- Ingredient #1: Adversarial pre-training + finetuning
- Ingredient #2: Perturbations in the embedding space
- Ingredient #3: Enhanced adversarial training algorithm



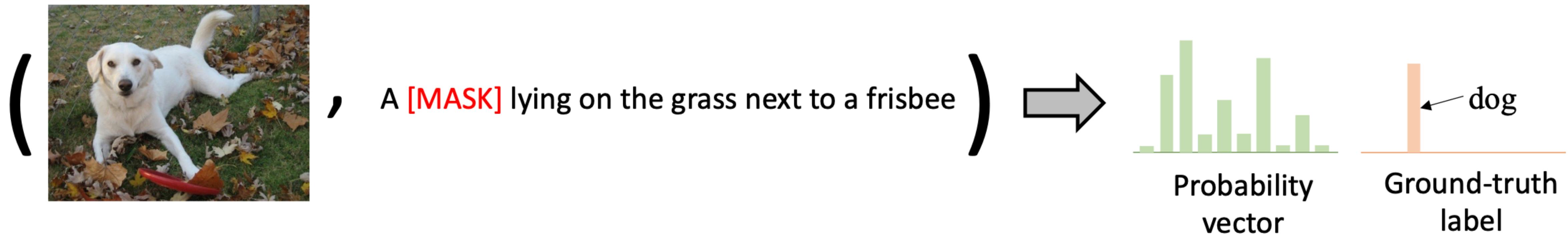
VILLA: Vision-and-Language Large-scale Adversarial Training

- Training objective:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x}_{img}, \mathbf{x}_{txt}, \mathbf{y}) \sim \mathcal{D}} [\mathcal{L}_{std}(\theta) + \mathcal{R}_{at}(\theta) + \alpha \cdot \mathcal{R}_{kl}(\theta)]$$

- Cross-entropy loss on clean data:

$$\mathcal{L}_{std}(\theta) = L(f_{\theta}(\mathbf{x}_{img}, \mathbf{x}_{txt}), \mathbf{y})$$



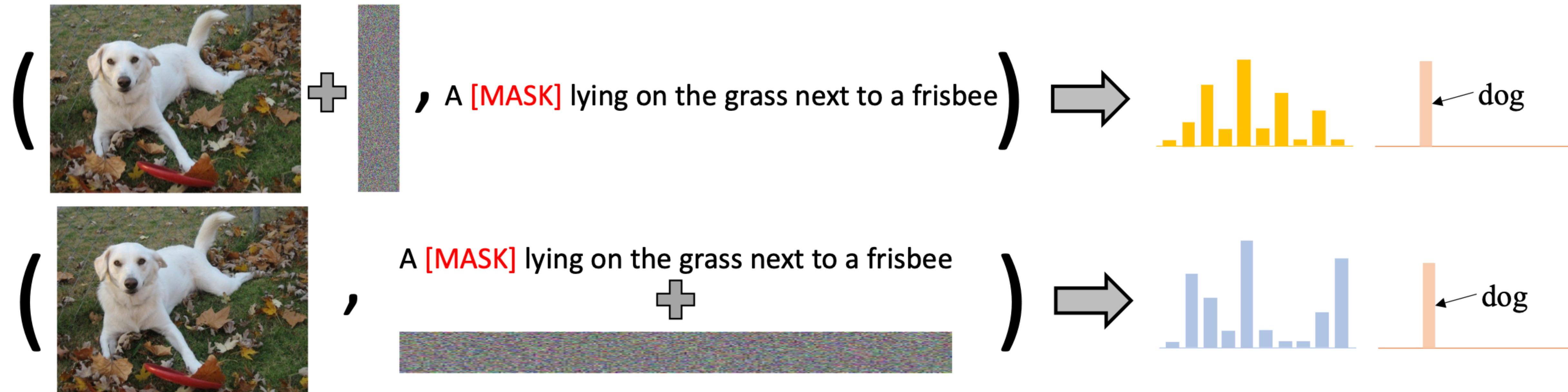
VILLA: Vision-and-Language Large-scale Adversarial Training

- Training objective:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x}_{img}, \mathbf{x}_{txt}, \mathbf{y}) \sim \mathcal{D}} \left[\mathcal{L}_{std}(\theta) + \mathcal{R}_{at}(\theta) + \alpha \cdot \mathcal{R}_{kl}(\theta) \right]$$

- Cross-entropy loss on adversarial embeddings:

$$\mathcal{R}_{at}(\theta) = \max_{\|\delta_{img}\| \leq \epsilon} L(f_{\theta}(\mathbf{x}_{img} + \delta_{img}, \mathbf{x}_{txt}), \mathbf{y}) + \max_{\|\delta_{txt}\| \leq \epsilon} L(f_{\theta}(\mathbf{x}_{img}, \mathbf{x}_{txt} + \delta_{txt}), \mathbf{y})$$



VILLA: Vision-and-Language Large-scale Adversarial Training

- Training objective:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x}_{img}, \mathbf{x}_{txt}, \mathbf{y}) \sim \mathcal{D}} \left[\mathcal{L}_{std}(\boldsymbol{\theta}) + \mathcal{R}_{at}(\boldsymbol{\theta}) + \alpha \cdot \mathcal{R}_{kl}(\boldsymbol{\theta}) \right]$$

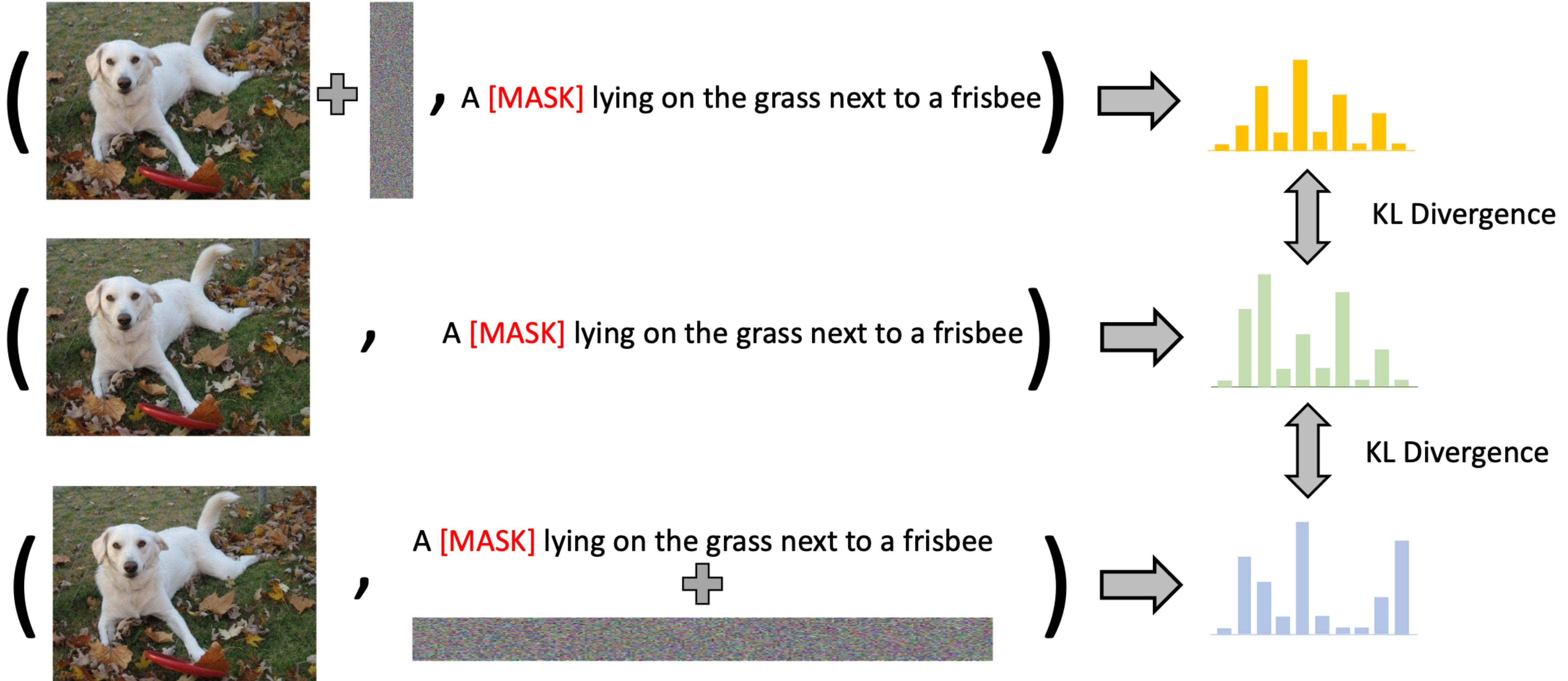
- KL-divergence loss for fine-grained adversarial regularization

$$\begin{aligned} \mathcal{R}_{kl}(\boldsymbol{\theta}) &= \max_{\|\boldsymbol{\delta}_{img}\| \leq \epsilon} L_{kl}(f_{\boldsymbol{\theta}}(\mathbf{x}_{img} + \boldsymbol{\delta}_{img}, \mathbf{x}_{txt}), f_{\boldsymbol{\theta}}(\mathbf{x}_{img}, \mathbf{x}_{txt})) \\ &\quad + \max_{\|\boldsymbol{\delta}_{txt}\| \leq \epsilon} L_{kl}(f_{\boldsymbol{\theta}}(\mathbf{x}_{img}, \mathbf{x}_{txt} + \boldsymbol{\delta}_{txt}), f_{\boldsymbol{\theta}}(\mathbf{x}_{img}, \mathbf{x}_{txt})), \end{aligned}$$

where $L_{kl}(p, q) = \text{KL}(p||q) + \text{KL}(q||p)$.

- Not only label-preserving, but the confidence level of the prediction between clean data and adversarial examples should also be close

VILLA: Vision-and-Language Large-scale Adversarial Training



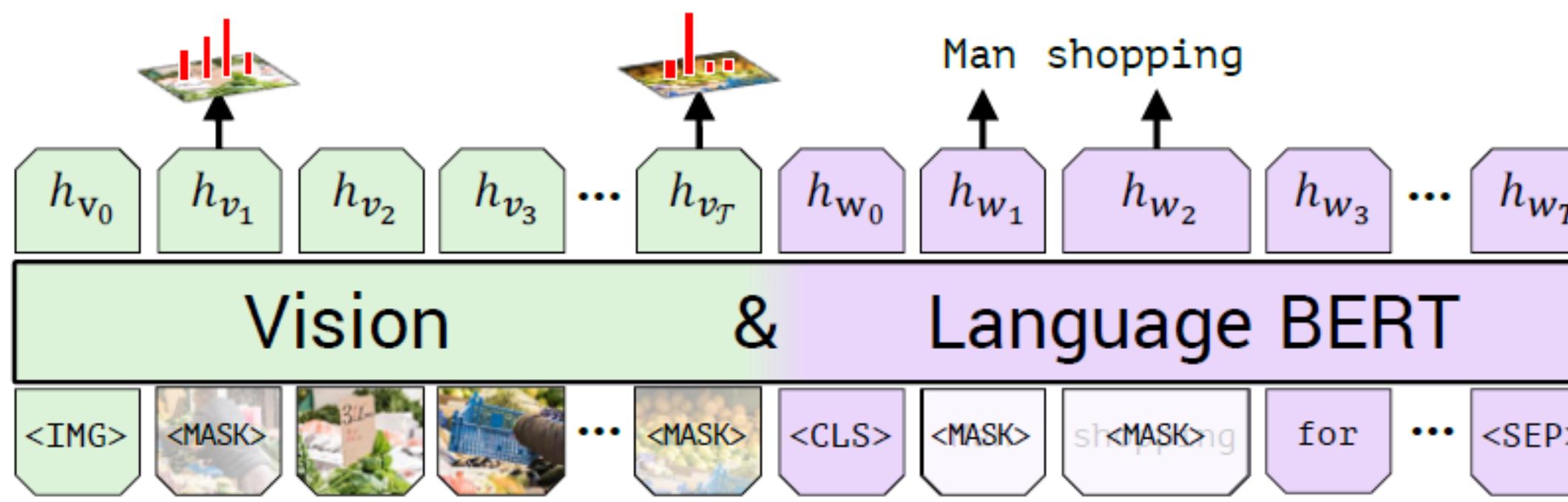
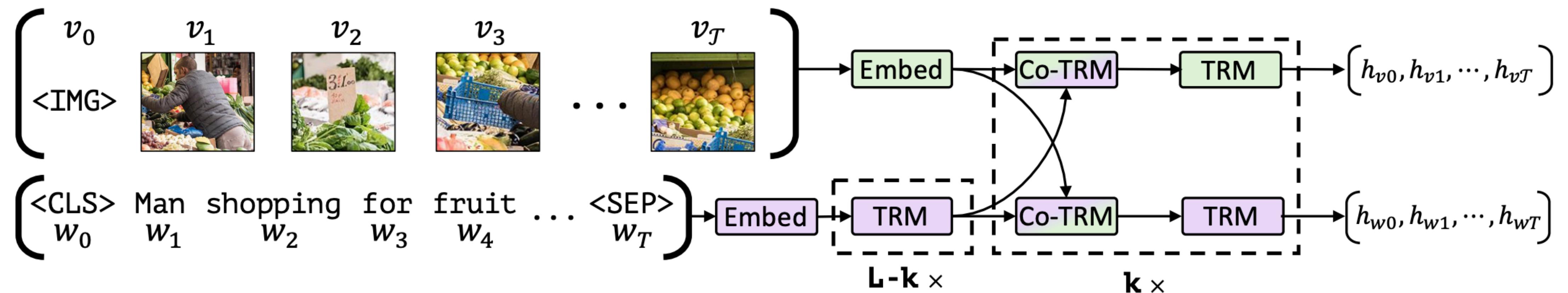
VILLA: Vision-and-Language Large-scale Adversarial Training

- Established new state of the art on all the tasks considered
- Gain: +0.85 on VQA, +2.9 on VCR, +1.49 on NLVR2, +0.64 on SNLI-VE

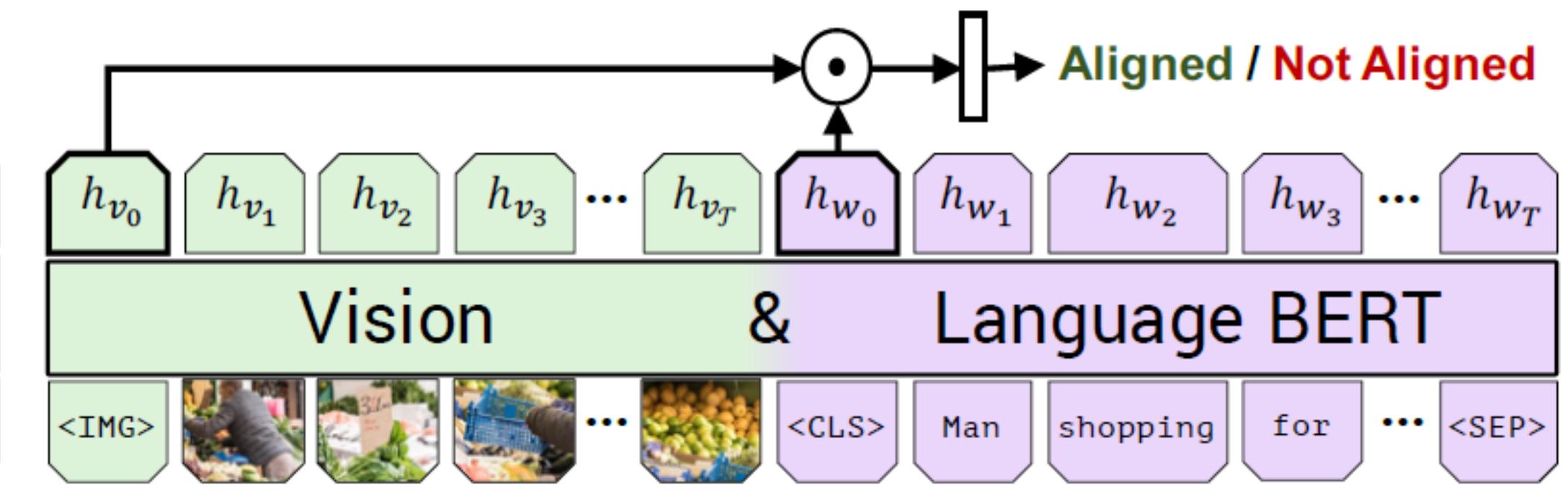
Method	VQA		VCR			NLVR ²		SNLI-VE	
	test-dev	test-std	Q→A	QA→R	Q→AR	dev	test-P	val	test
ViLBERT	70.55	70.92	72.42 (73.3)	74.47 (74.6)	54.04 (54.8)	-	-	-	-
VisualBERT	70.80	71.00	70.8 (71.6)	73.2 (73.2)	52.2 (52.4)	67.4	67.0	-	-
LXMERT	72.42	72.54	-	-	-	74.90	74.50	-	-
Unicoder-VL	-	-	72.6 (73.4)	74.5 (74.4)	54.4 (54.9)	-	-	-	-
12-in-1	73.15	-	-	-	-	-	78.87	-	76.95
VL-BERT _{BASE}	71.16	-	73.8 (-)	74.4 (-)	55.2 (-)	-	-	-	-
Oscar _{BASE}	73.16	73.44	-	-	-	78.07	78.36	-	-
UNITER _{BASE}	72.70	72.91	74.56 (75.0)	77.03 (77.2)	57.76 (58.2)	77.18	77.85	78.59	78.28
VILLA _{BASE}	73.59	73.67	75.54 (76.4)	78.78 (79.1)	59.75 (60.6)	78.39	79.30	79.47	79.03
VL-BERT _{LARGE}	71.79	72.22	75.5 (75.8)	77.9 (78.4)	58.9 (59.7)	-	-	-	-
Oscar _{LARGE}	73.61	73.82	-	-	-	79.12	80.37	-	-
UNITER _{LARGE}	73.82	74.02	77.22 (77.3)	80.49 (80.8)	62.59 (62.8)	79.12	79.98	79.39	79.38
VILLA _{LARGE}	74.69	74.87	78.45 (78.9)	82.57 (82.8)	65.18 (65.7)	79.76	81.47	80.18	80.02

(a) Results on VQA, VCR, NLVR², and SNLI-VE.

Visual BERT (ViLBERT)



(a) Masked multi-modal learning

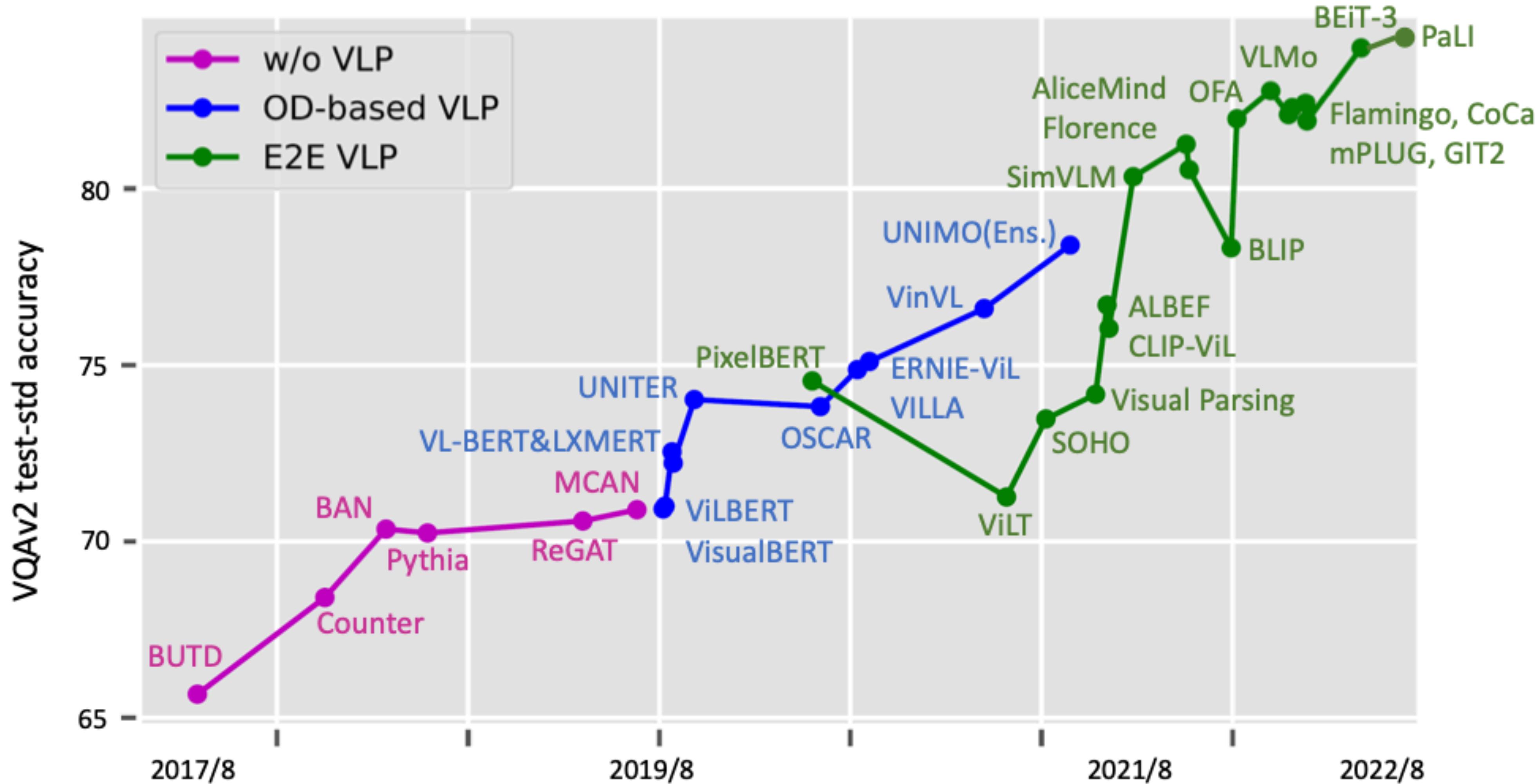


(b) Multi-modal alignment prediction

12-in-1: Multi-task Vision and Language Representation

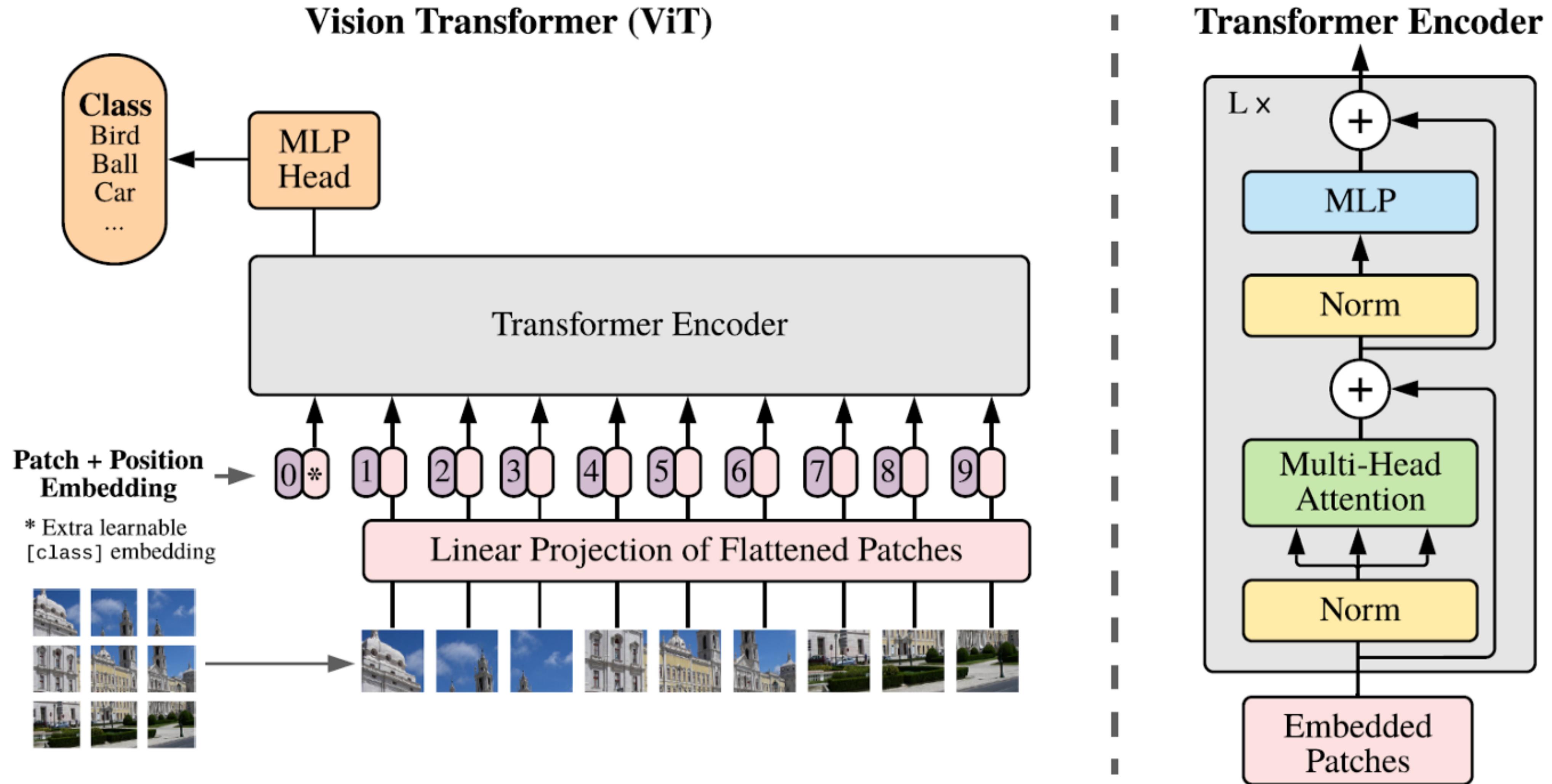
	<i>Clean</i>	Vocab-based VQA (G1)			Image Retrieval (G2)		Referring Expression (G3)					Verification (G4)		# params (# models)	All Tasks Average	
		VQAv2	GQA	VG QA	COCO	Flickr30k	COCO	COCO+	COCOg	V7W	GW	NLVR ²	SNLI-VE			
		test-dev	test-dev	val	test(R1)	test(R1)	test	test	test	test	test	testP	test			
1	Single-Task (ST)	71.82	58.19	34.38	65.28	61.14	78.63	71.11	72.24	80.51	62.81	74.25	76.72	3B (12)	67.25	
2	Single-Task (ST)	✓	71.24	59.09	34.10	64.80	61.46	78.17	69.47	72.21	80.51	62.53	74.25	76.53	3B (12)	67.03
3	Group-Tasks (GT)	✓	72.03	59.60	36.18	65.06	66.00	80.23	72.79	75.30	81.54	64.78	74.62	76.52	1B (4)	68.72
4	All-Tasks (AT)	✓	72.57	60.12	36.36	63.70	63.52	80.58	73.25	75.96	82.75	65.04	78.44	76.78	270M (1)	69.08
5	All-Tasks _{w/o G4}	✓	72.68	62.09	36.74	64.88	64.62	80.76	73.60	75.80	83.03	65.41	-	-	266M (1)	-
6	GT $\xrightarrow{\text{finetune}}$ ST	✓	72.61	59.96	35.81	66.26	66.98	79.94	72.12	75.18	81.57	64.56	74.47	76.34	3B (12)	68.81
7	AT $\xrightarrow{\text{finetune}}$ ST	✓	72.92	60.48	36.56	65.46	65.14	80.86	73.45	76.00	83.01	65.15	78.87	76.73	3B (12)	69.55
8	AT $\xrightarrow{\text{finetune}}$ ST		73.15	60.65	36.64	68.00	67.90	81.20	74.22	76.35	83.35	65.69	78.87	76.95	3B (12)	70.24

Recent History of Visio-Lingual Models

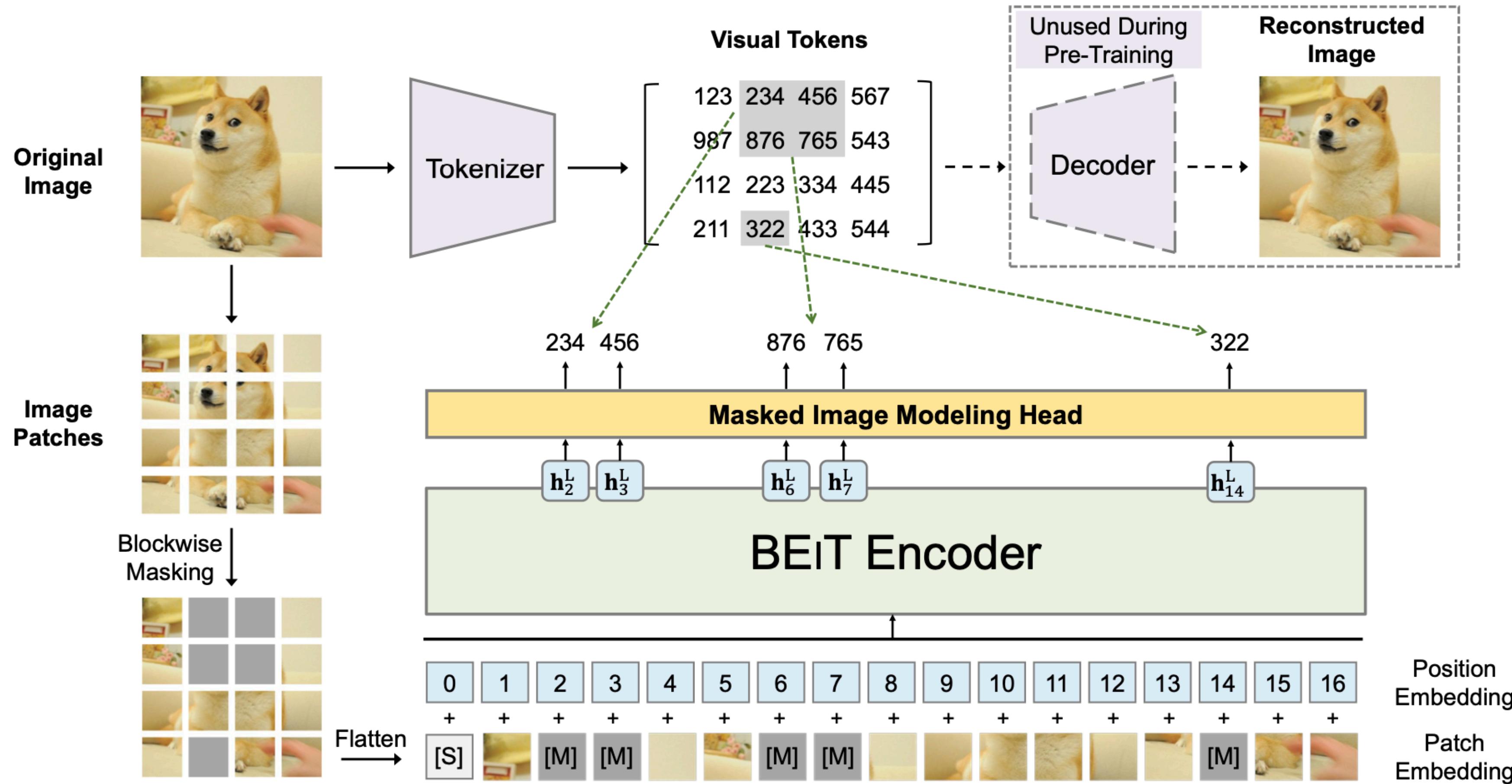


Vision Transformer

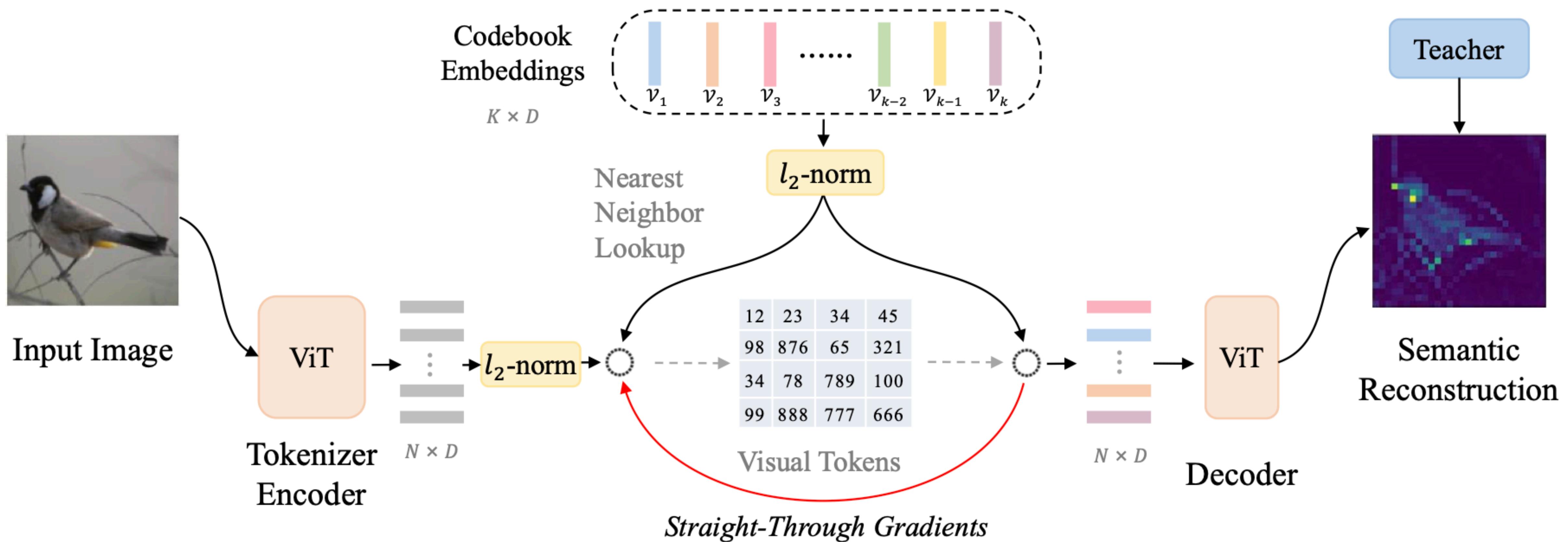
[Dosovitskiy et al., 2020]



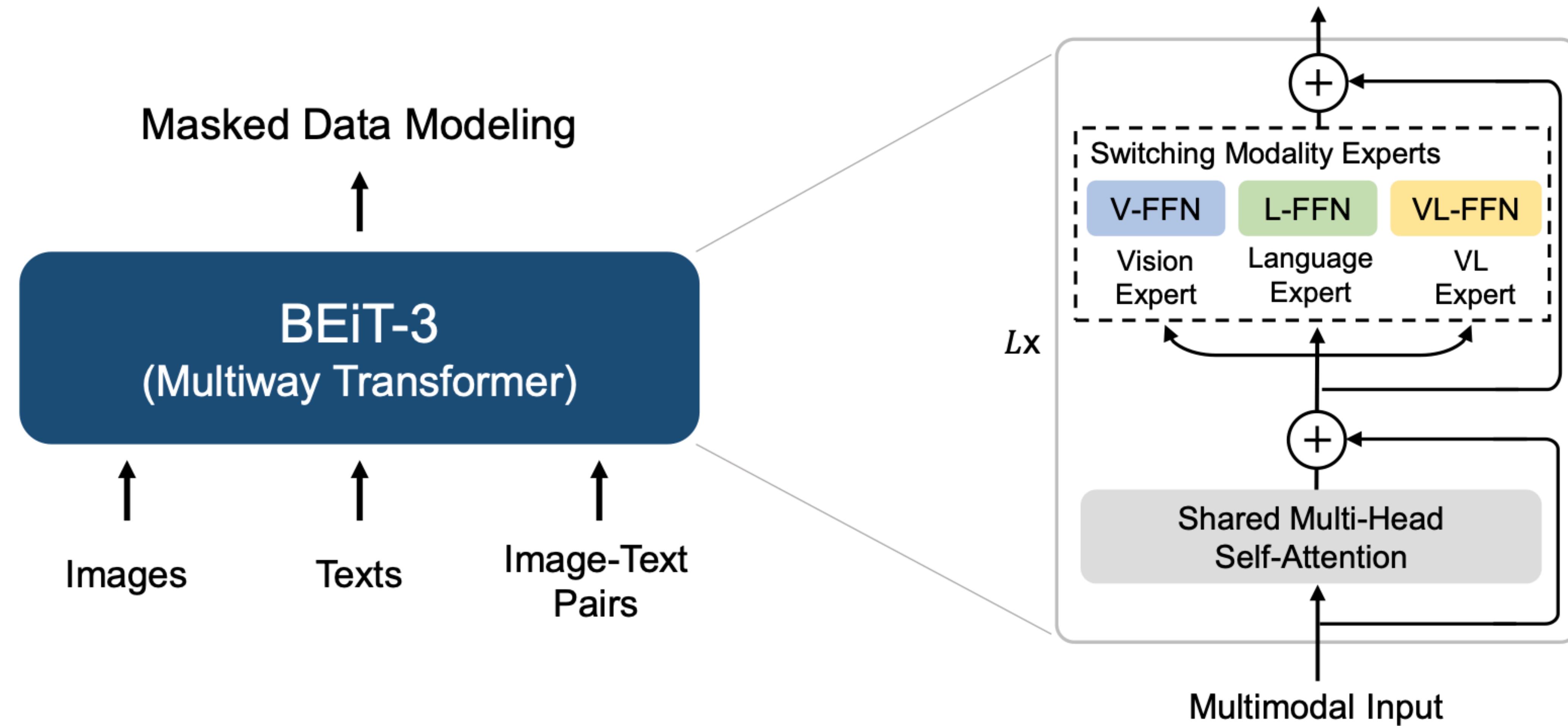
BEiT: BERT Pre-Training of Image Transformers



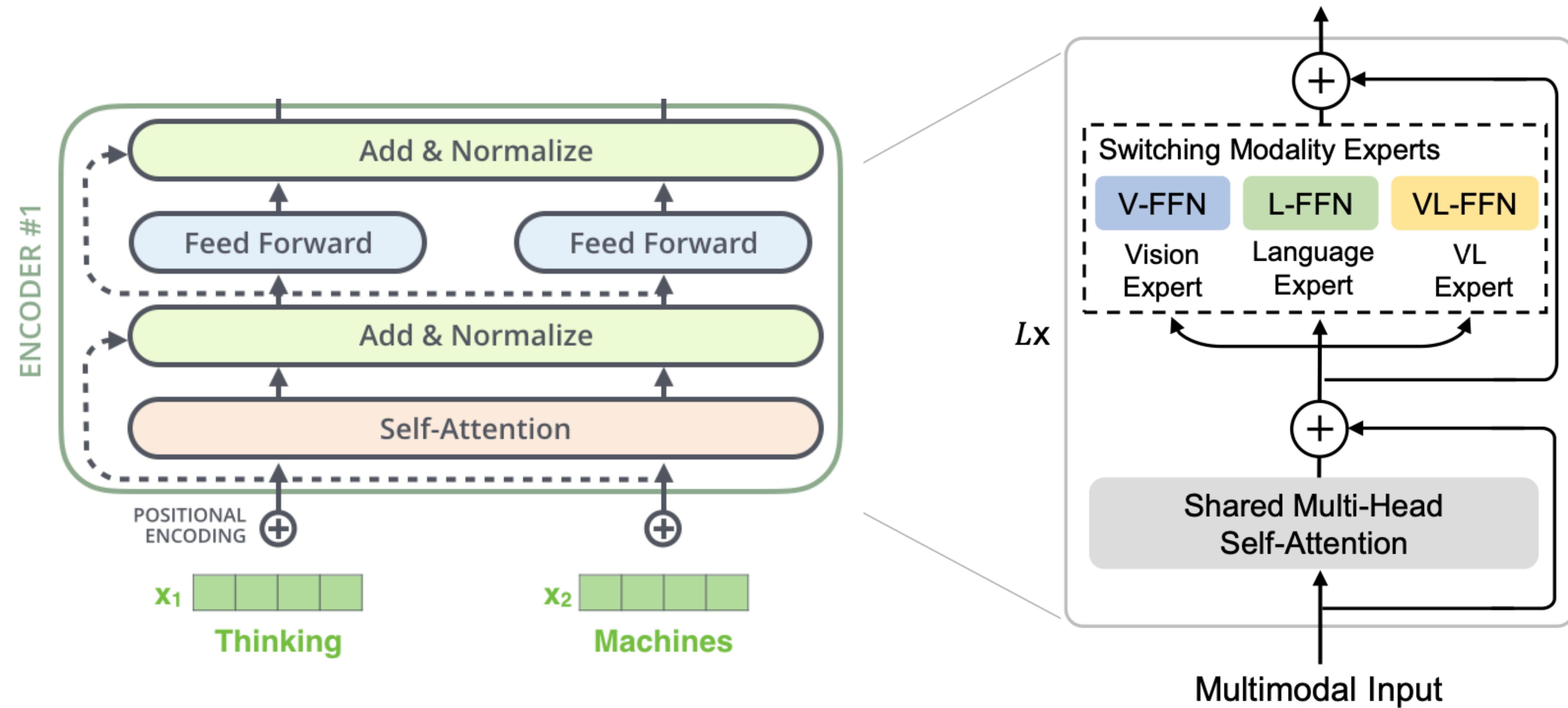
BEiT-v2



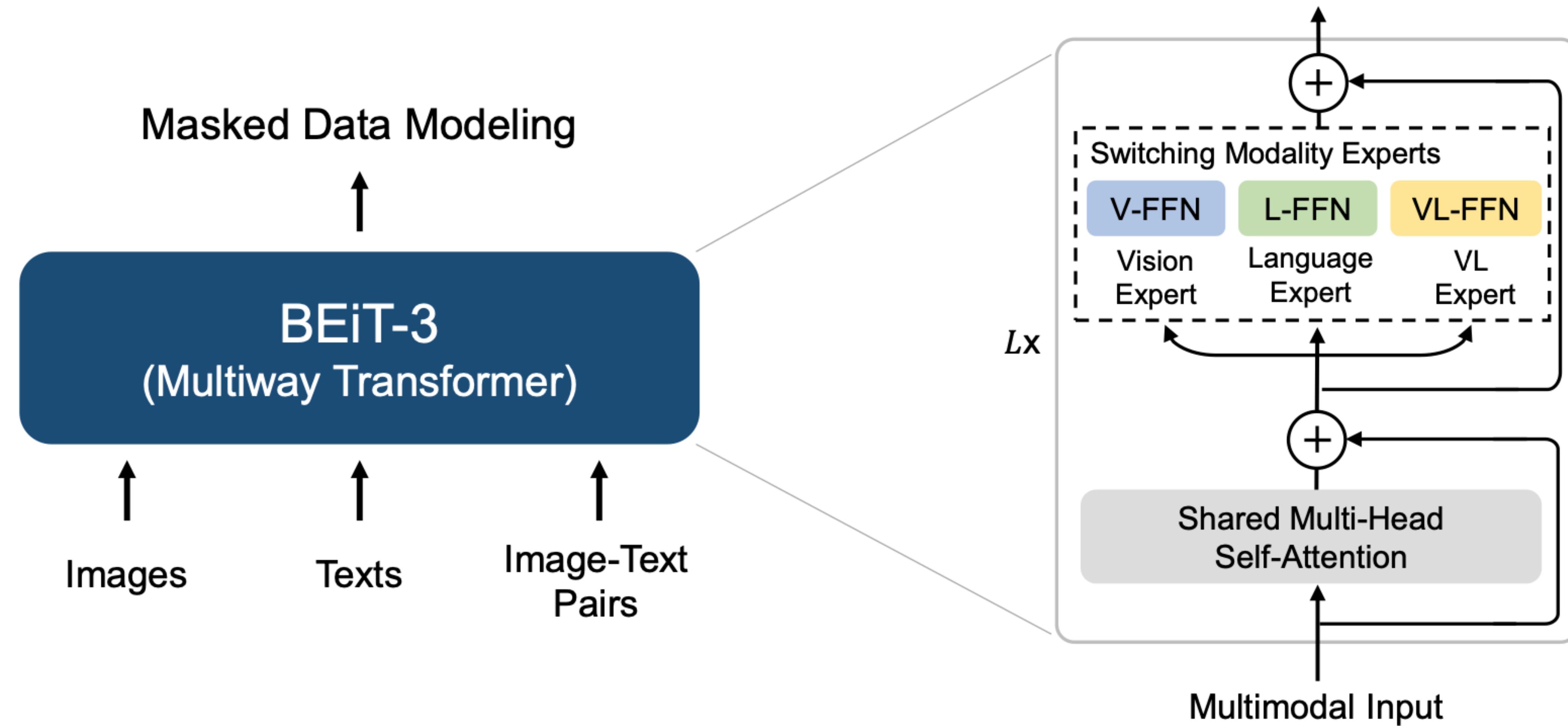
BEiT-v3: Image as a Foreign Language



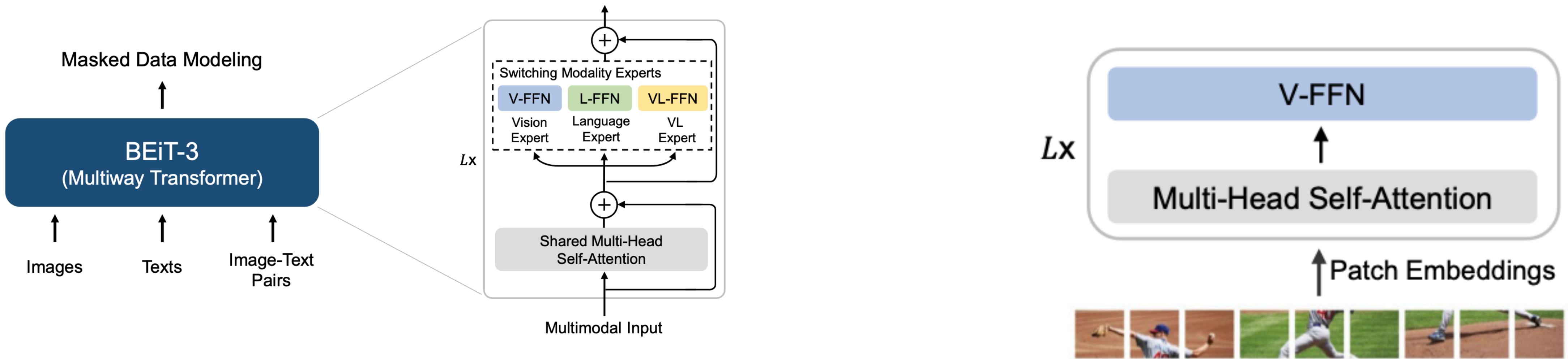
BEiT-v3: Image as a Foreign Language



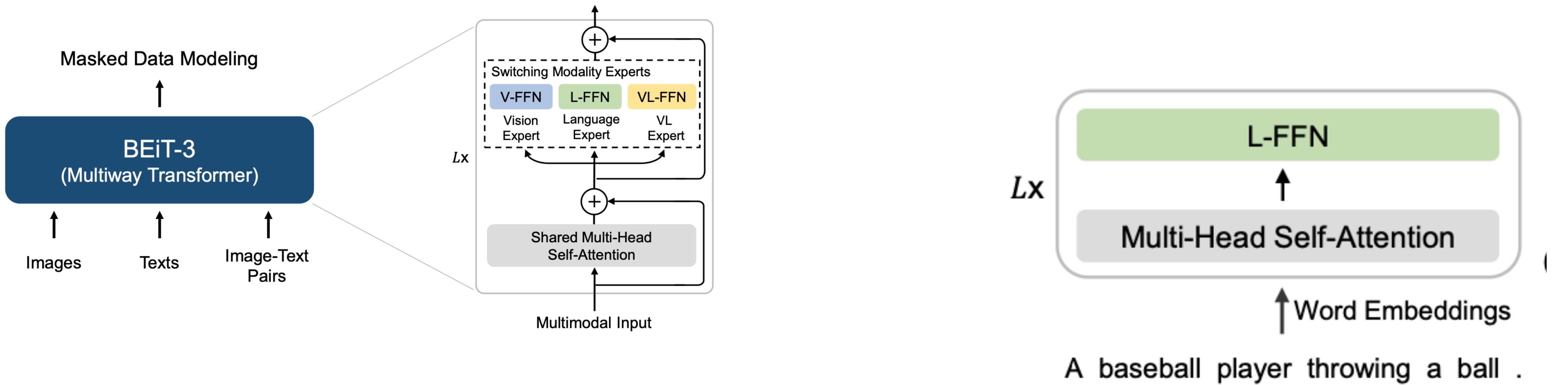
BEiT-v3: Image as a Foreign Language



BEiT-v3: Image as a Foreign Language

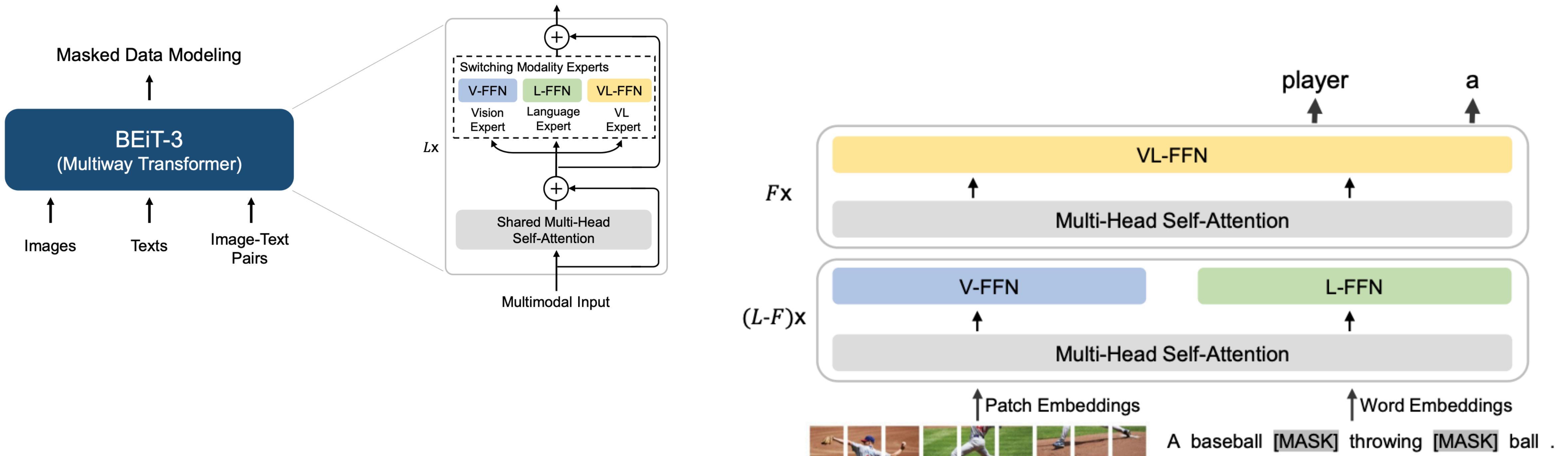


BEiT-v3: Image as a Foreign Language



(b) Language Encoder
Masked Language Modeling

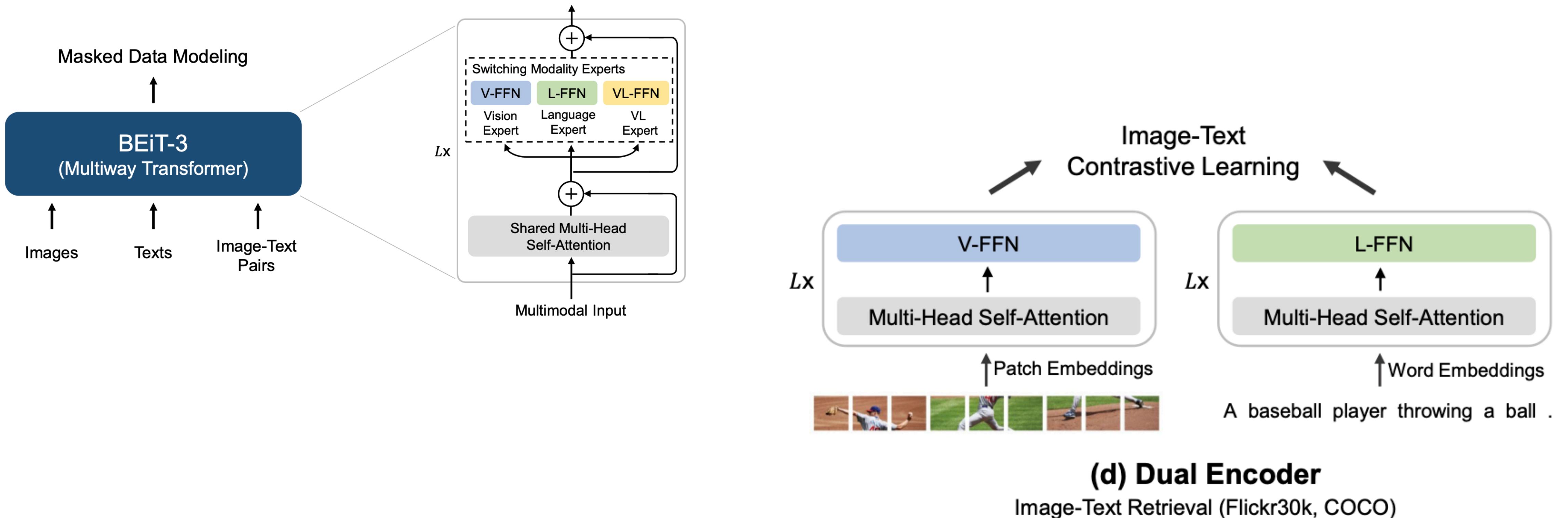
BEiT-v3: Image as a Foreign Language



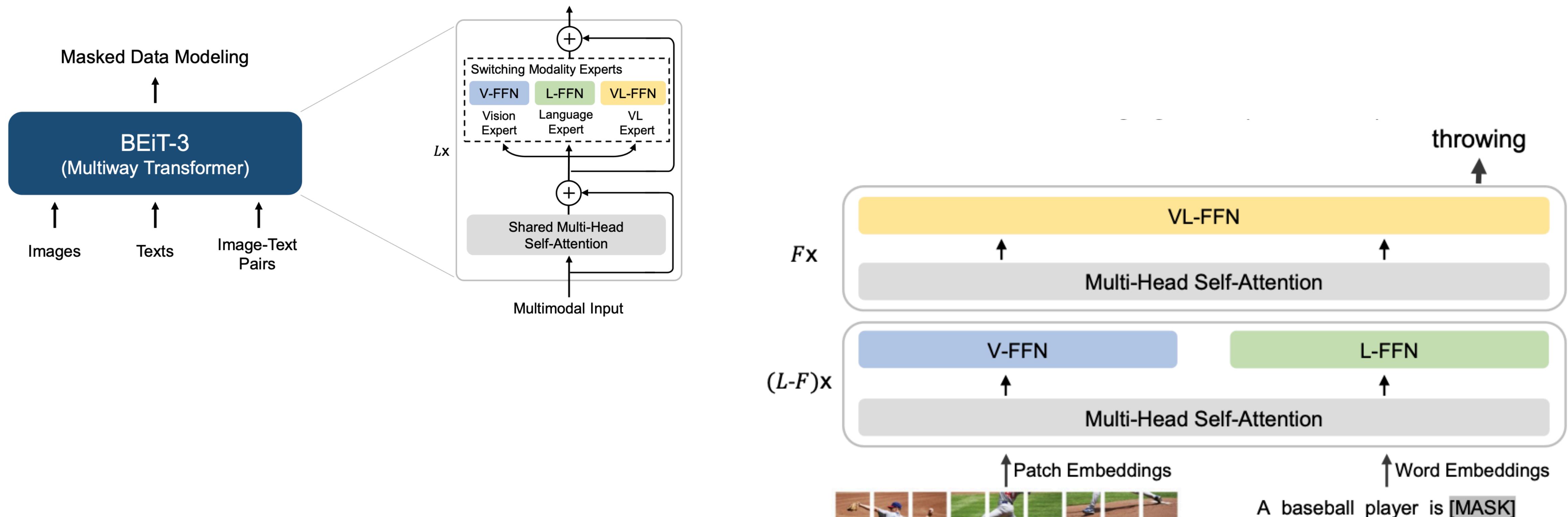
(c) Fusion Encoder

Masked Vision-Language Modeling
Vision-Language Tasks (VQA, NLVR2)

BEiT-v3: Image as a Foreign Language



BEiT-v3: Image as a Foreign Language



(e) Image-to-Text Generation

Image Captioning (COCO)

BEiT-v3: Image as a Foreign Language

Model	MSCOCO (5K test set)							Flickr30K (1K test set)						
	Image → Text			Text → Image			Image → Text			Text → Image				
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10		
<i>Fusion-encoder models</i>														
UNITER [CLY ⁺ 20]	65.7	88.6	93.8	52.9	79.9	88.0	87.3	98.0	99.2	75.6	94.1	96.8		
VILLA [GCL ⁺ 20]	-	-	-	-	-	-	87.9	97.5	98.8	76.3	94.2	96.8		
Oscar [LYL ⁺ 20]	73.5	92.2	96.0	57.5	82.8	89.8	-	-	-	-	-	-		
VinVL [ZLH ⁺ 21]	75.4	92.9	96.2	58.8	83.5	90.3	-	-	-	-	-	-		
<i>Dual encoder + Fusion encoder reranking</i>														
ALBEF [LSG ⁺ 21]	77.6	94.3	97.2	60.7	84.3	90.5	95.9	99.8	100.0	85.6	97.5	98.9		
BLIP [LLXH22]	82.4	95.4	97.9	65.1	86.3	91.8	97.4	99.8	99.9	87.6	97.7	99.0		
<i>Dual-encoder models</i>														
ALIGN [JYX ⁺ 21]	77.0	93.5	96.9	59.9	83.3	89.8	95.3	99.8	100.0	84.9	97.4	98.6		
FILIP [YHH ⁺ 21]	78.9	94.4	97.4	61.2	84.3	90.6	96.6	100.0	100.0	87.1	97.7	99.1		
Florence [YCC ⁺ 21]	81.8	95.2	-	63.2	85.7	-	97.2	99.9	-	87.9	98.1	-		
BEiT-3	84.8	96.5	98.3	67.2	87.7	92.8	98.0	100.0	100.0	90.3	98.7	99.5		

BEiT-v3: Image as a Foreign Language

Model	Extra OD Data	Maximum Image Size	COCO test-dev	
			AP ^{box}	AP ^{mask}
ViT-Adapter [CDW ⁺ 22]	-	1600	60.1	52.1
DyHead [DCX ⁺ 21]	ImageNet-Pseudo Labels	2000	60.6	-
Soft Teacher [XZH ⁺ 21]	Object365	-	61.3	53.0
GLIP [LZZ ⁺ 21]	FourODs	-	61.5	-
GLIPv2 [ZZH ⁺ 22]	FourODs	-	62.4	-
Florence [YCC ⁺ 21]	FLOD-9M	2500	62.4	-
SwinV2-G [LHL ⁺ 21]	Object365	1536	63.1	54.4
Mask DINO [LZX ⁺ 22]	Object365	1280	-	54.7
DINO [ZLL ⁺ 22]	Object365	2000	63.3	-
BEiT-3	Object365	1280	63.7	54.8



THE UNIVERSITY OF BRITISH COLUMBIA

Topics in AI (CPSC 532S): Multimodal Learning with Vision, Language and Sound



A decorative horizontal bar at the bottom of the slide, composed of five colored segments: light green, medium green, cyan, light blue, and light purple.

Lecture 23: Meta-learning

Few-shot Learning

Given abundant training examples for the base classes, few-shot learning algorithms aim to learn to recognize novel classes with a limited amount of labeled examples

Few-shot Learning

Given abundant training examples for the base classes, few-shot learning algorithms aim to learn to recognize novel classes with a limited amount of labeled examples

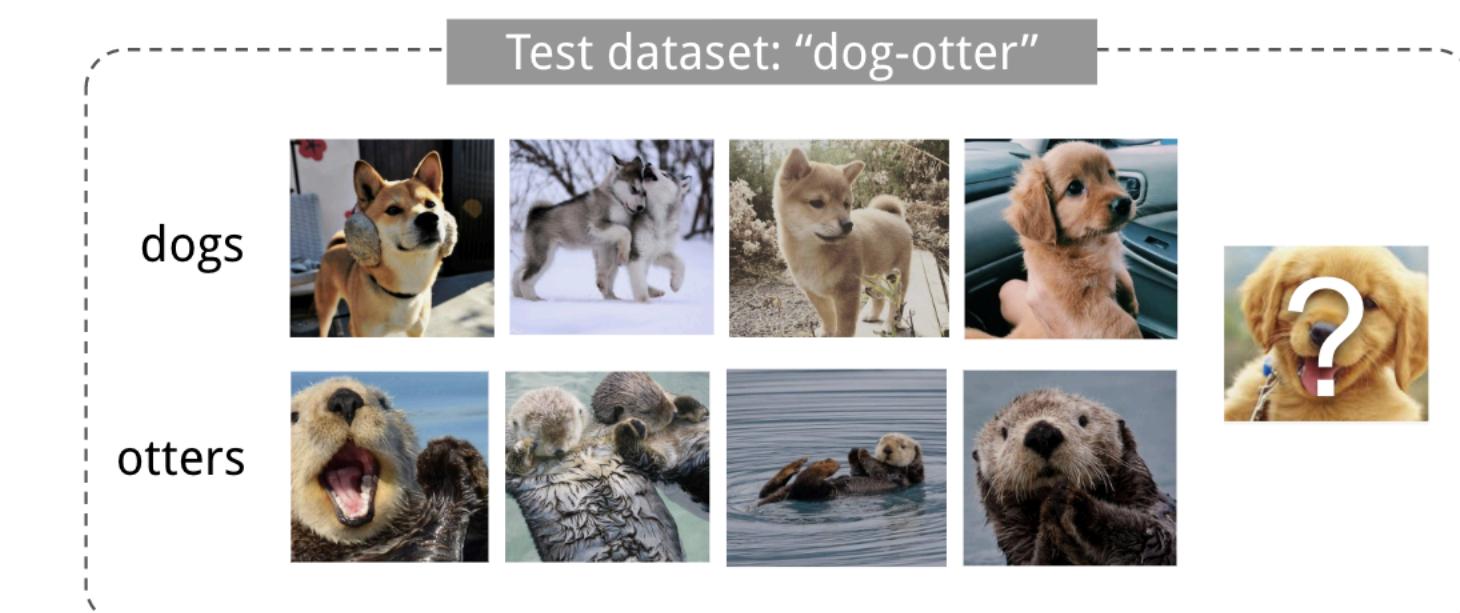


Few-shot Learning

Given abundant training examples for the base classes, few-shot learning algorithms aim to learn to recognize novel classes with a limited amount of labeled examples

Task: classify test (a.k.a. query) set images with “novel labels” (labels not present in base data but available in support set)

Given: limited novel-labelled Support Set with K images from each of N novel classes



Few-shot Learning

Given abundant training examples for the base classes, few-shot learning algorithms aim to learn to recognize novel classes with a limited amount of labeled examples

Task: classify test (a.k.a. query) set images with “novel labels” (labels not present in base data but available in support set)

Given: limited novel-labelled Support Set with K images from each of N novel classes

At test time: n-way k-shot tasks



2-way 4-shot

Few-shot Learning

Transfer learning baselines

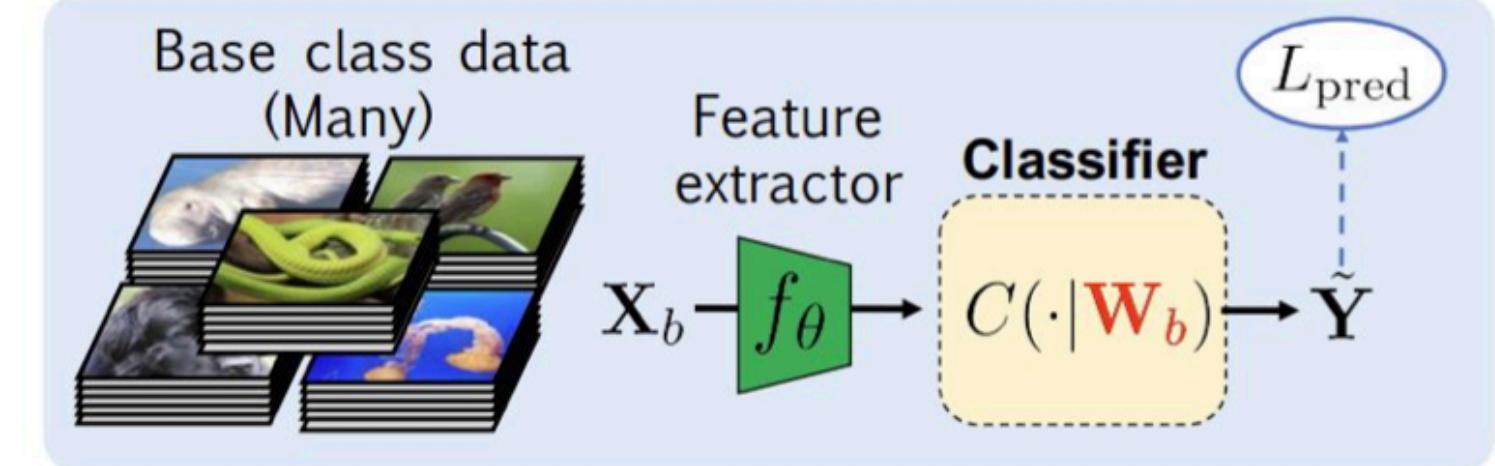
1. Pre-training: supervised learning of model on base data
2. Fine-tuning: supervised learning of (parts of or whole) model on labelled support data
3. Testing on test / query dataset

Few-shot Learning

Transfer learning baselines

1. Pre-training: supervised learning of model on base data
2. Fine-tuning: supervised learning of (parts of or whole) model on labelled support data
3. Testing on test / query dataset

Pre-Training stage

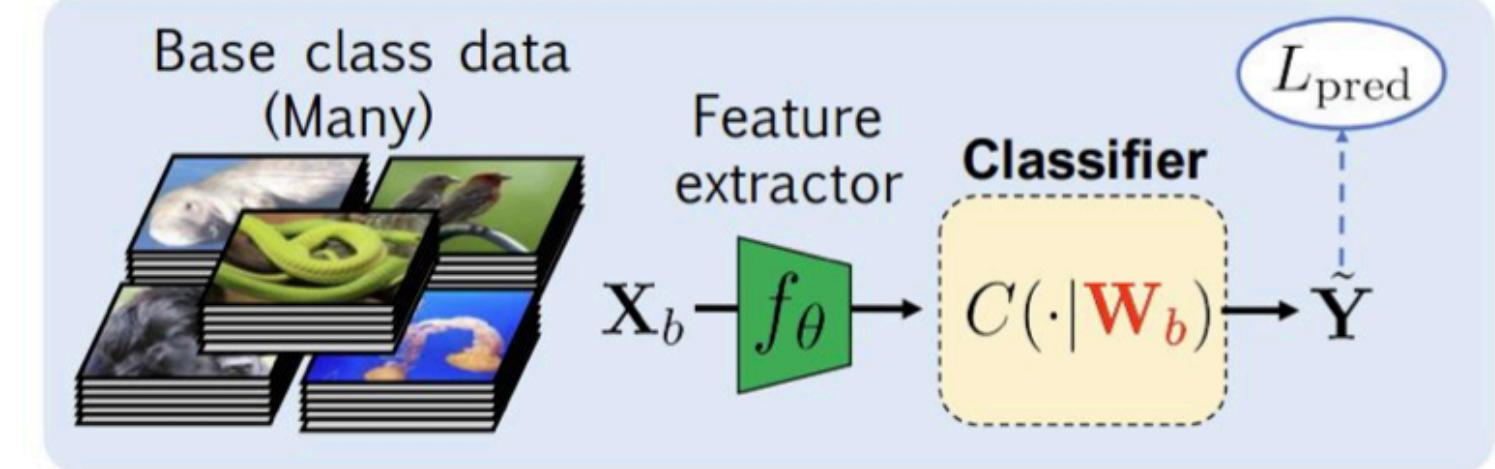


Few-shot Learning

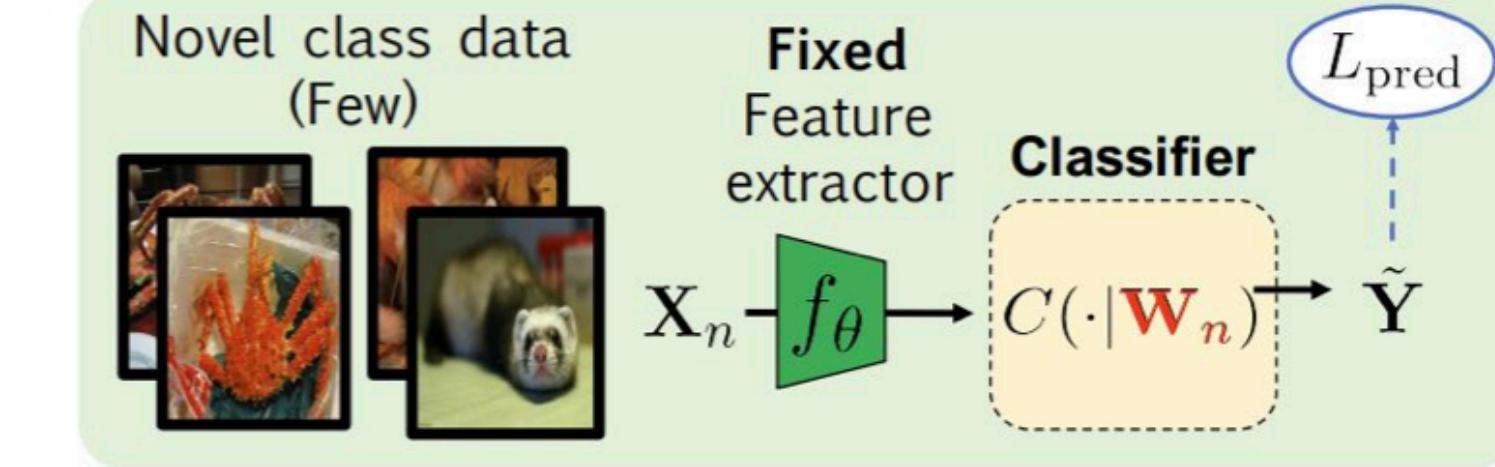
Transfer learning baselines

1. Pre-training: supervised learning of model on base data
2. Fine-tuning: supervised learning of (parts of or whole) model on labelled support data
3. Testing on test / query dataset

Pre-Training stage



Fine-tuning stage



Retrain only classifier

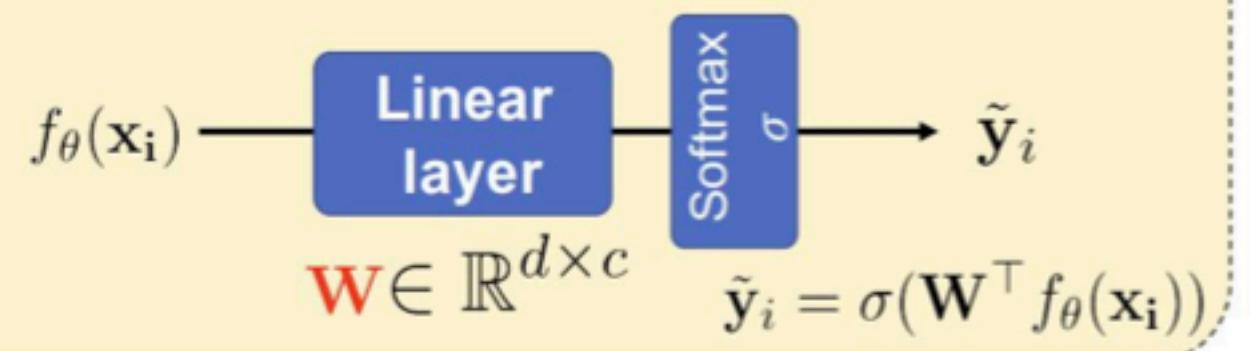
Few-shot Learning

Transfer learning baselines

1. Pre-training: supervised learning of model on base data
2. Fine-tuning: supervised learning of (parts of or whole) model on labelled support data
3. Testing on test / query dataset

Choice of Classifier

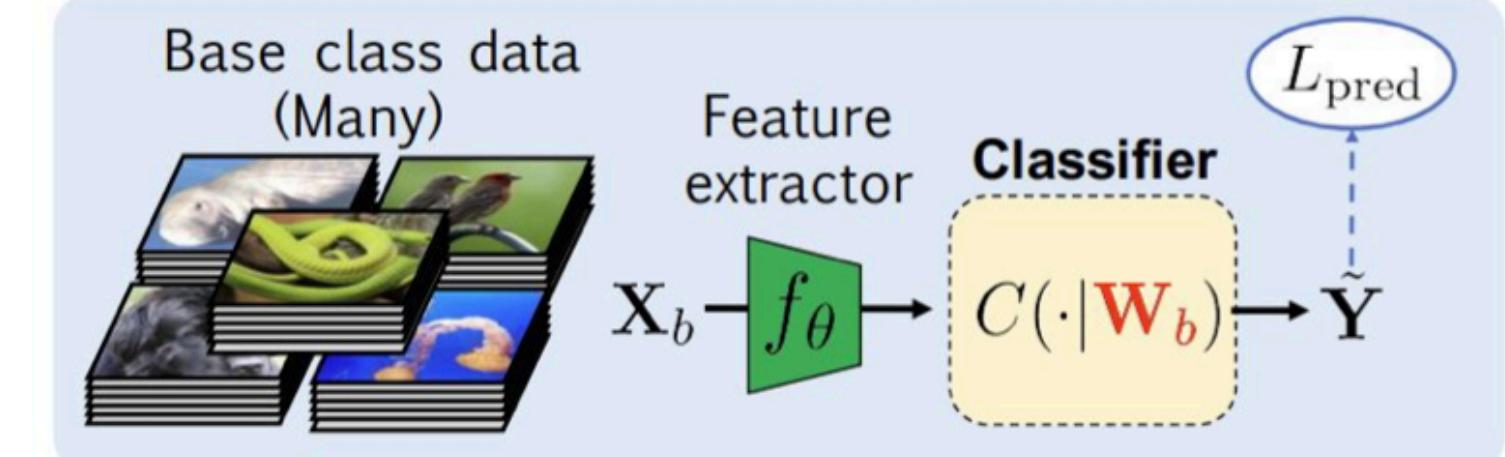
Baseline



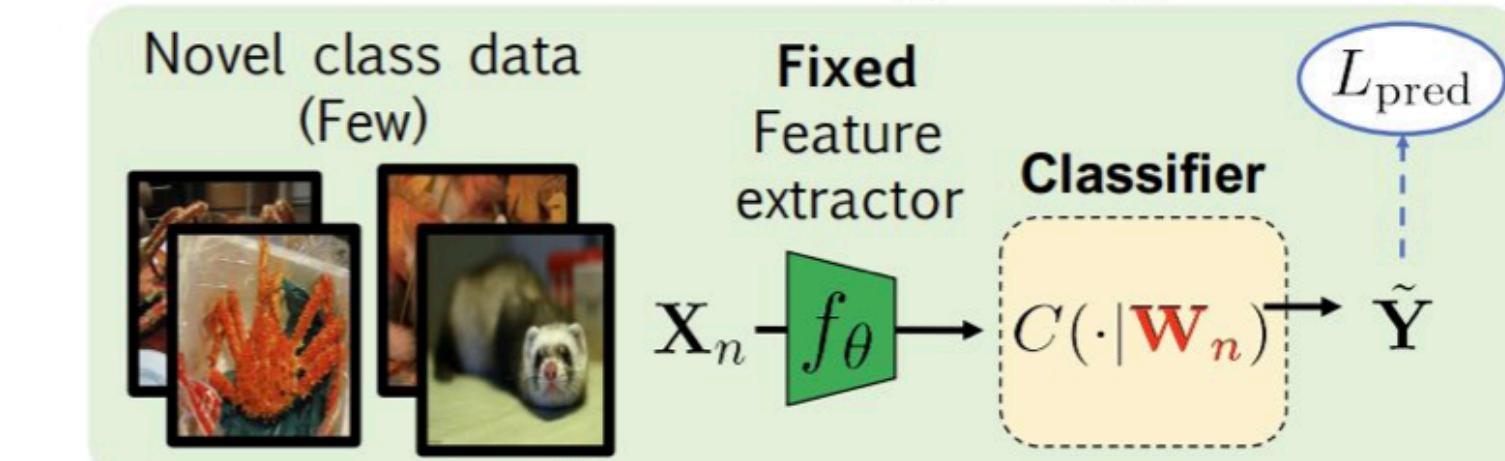
(Standard procedure)

Commonly seen last layer (a.k.a. logits) in a deep neural network classifying image into one of classes by min loss = f(predicted label probability vector, true one-hot encoded label).

Pre-Training stage



Fine-tuning stage



Retrain only classifier

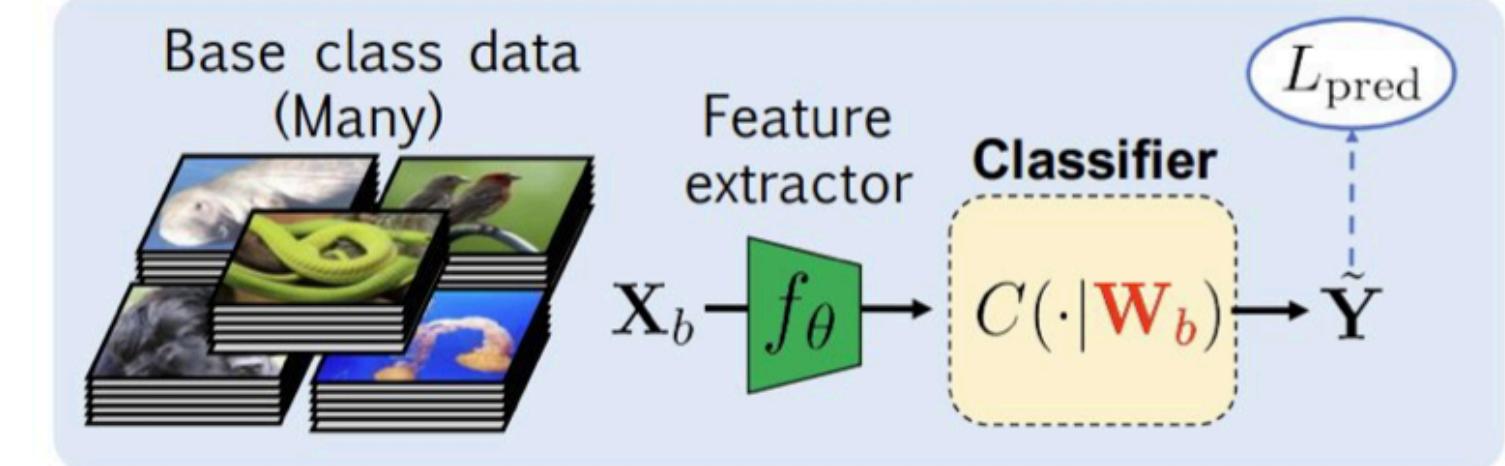
Few-shot Learning

Transfer learning baselines

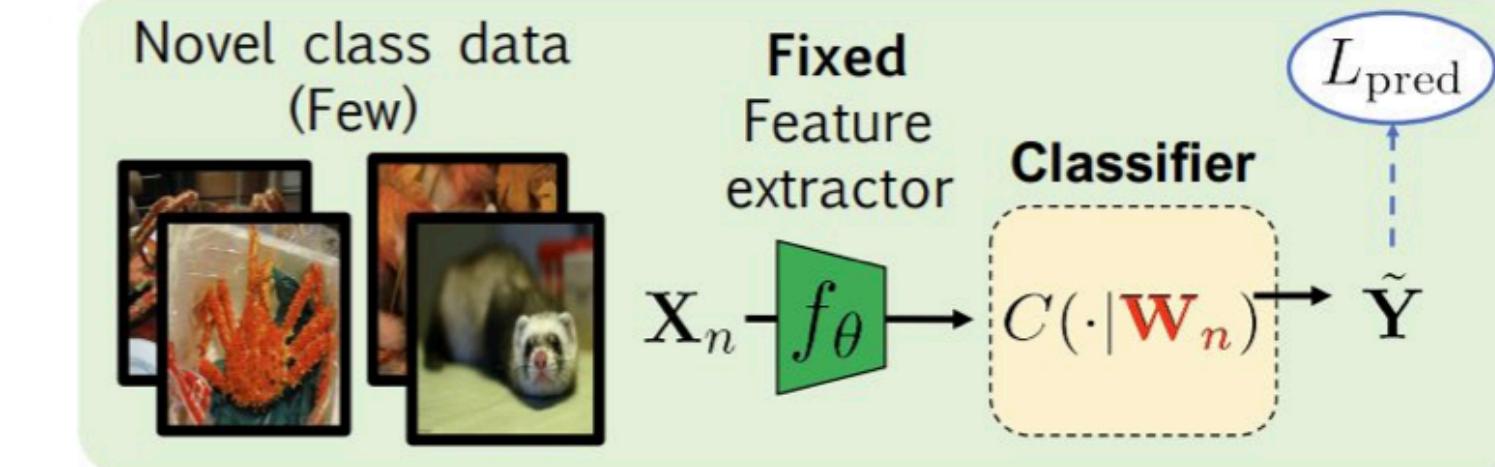
1. Pre-training: supervised learning of model on base data
2. Fine-tuning: supervised learning of (parts of or whole) model on labelled support data
3. Testing on test / query dataset

Expected to not perform well on test / query dataset without large support set (in fine-tuning stage)

Pre-Training stage



Fine-tuning stage



Retrain only classifier

Few-shot Learning

Transfer learning
baselines

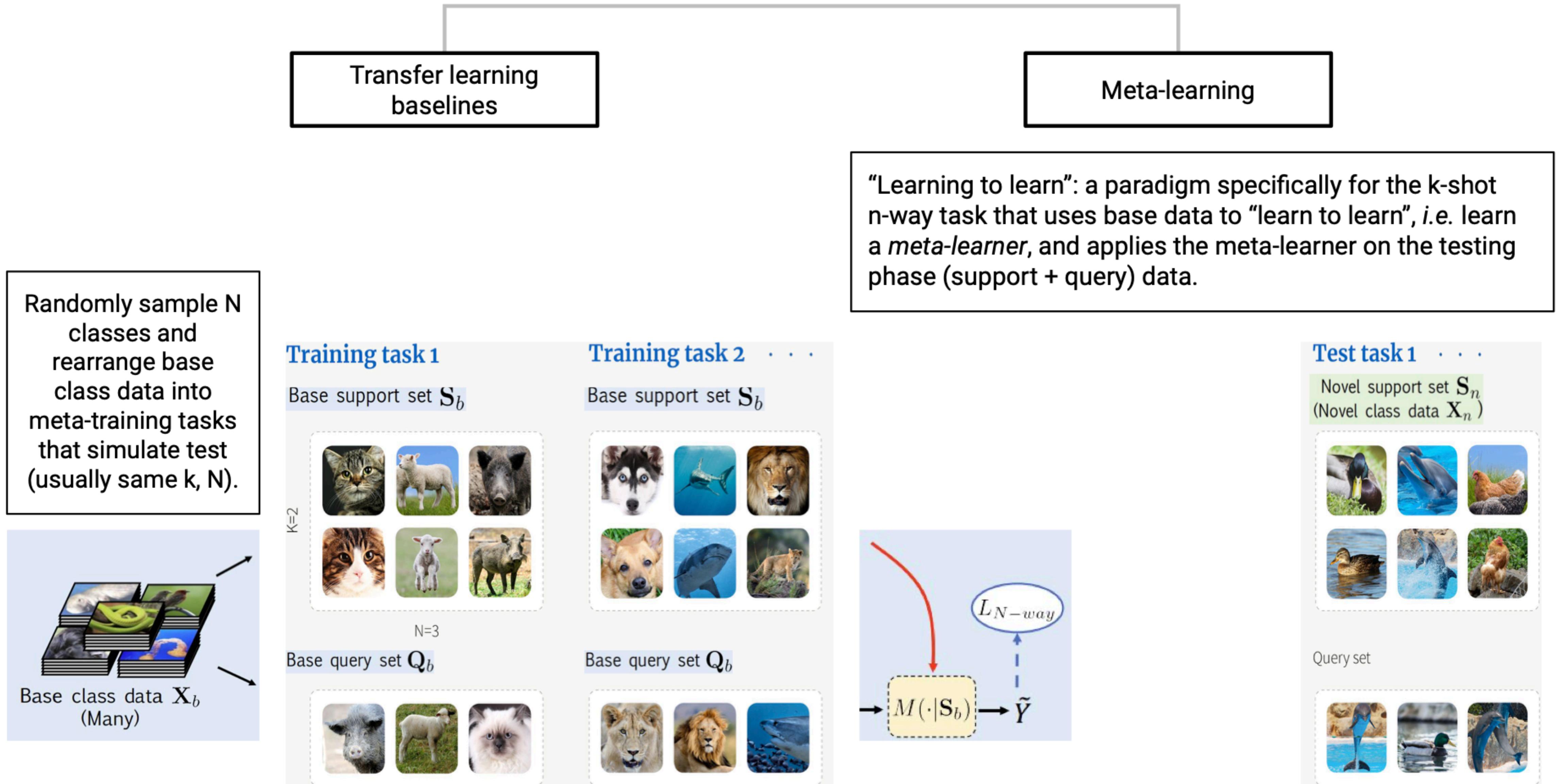
Meta-learning

1. Pre-training: supervised learning of model on base data
2. Fine-tuning: supervised learning of (parts of or whole) model on labelled support data
3. Testing on test / query dataset

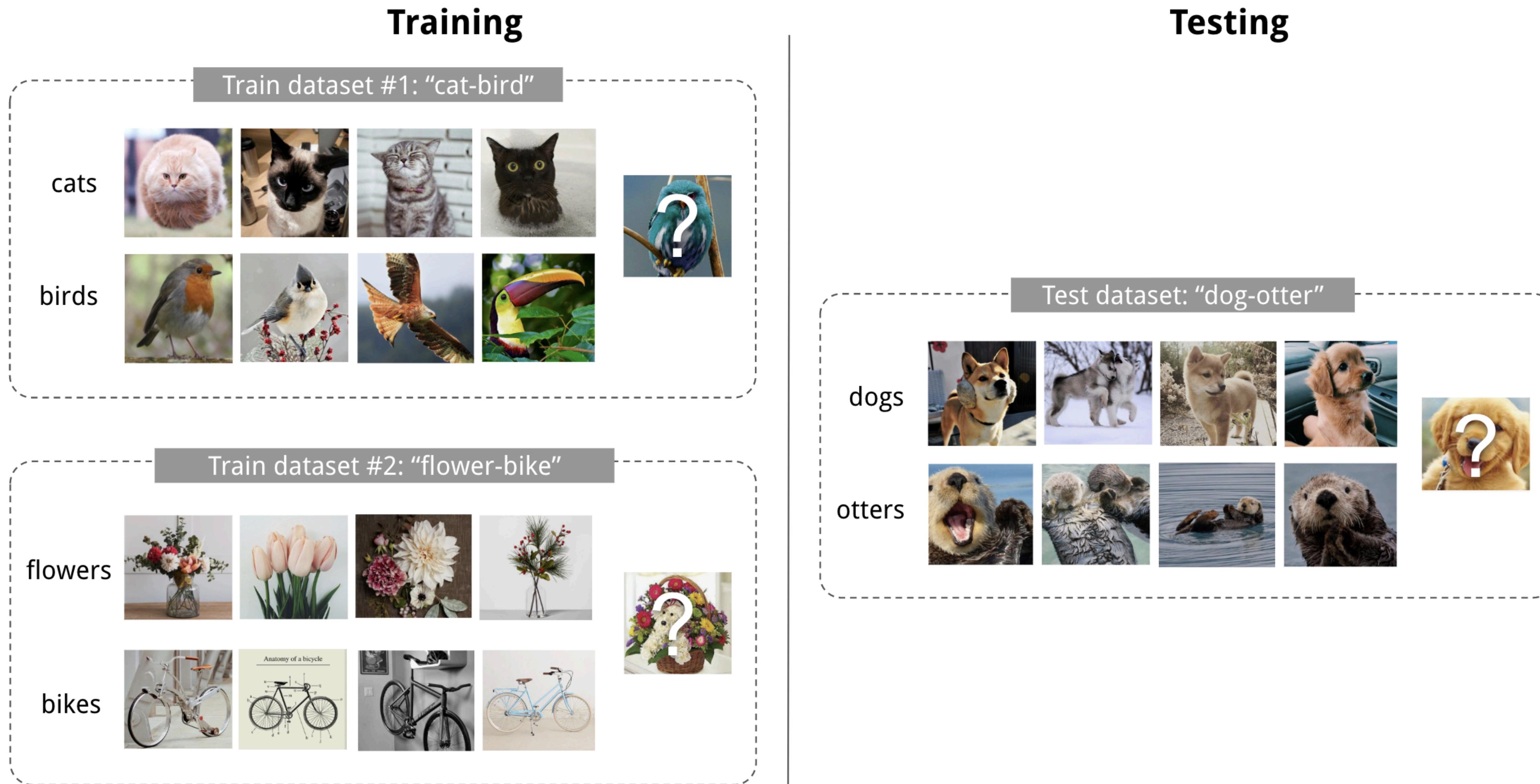
“Learning to learn”: a paradigm specifically for the k-shot n-way task that uses base data to “learn to learn”, i.e. learn a *meta-learner*, and applies the meta-learner on the testing phase (support + query) data.

Expected to not perform well on test / query dataset without large support set (in fine-tuning stage)

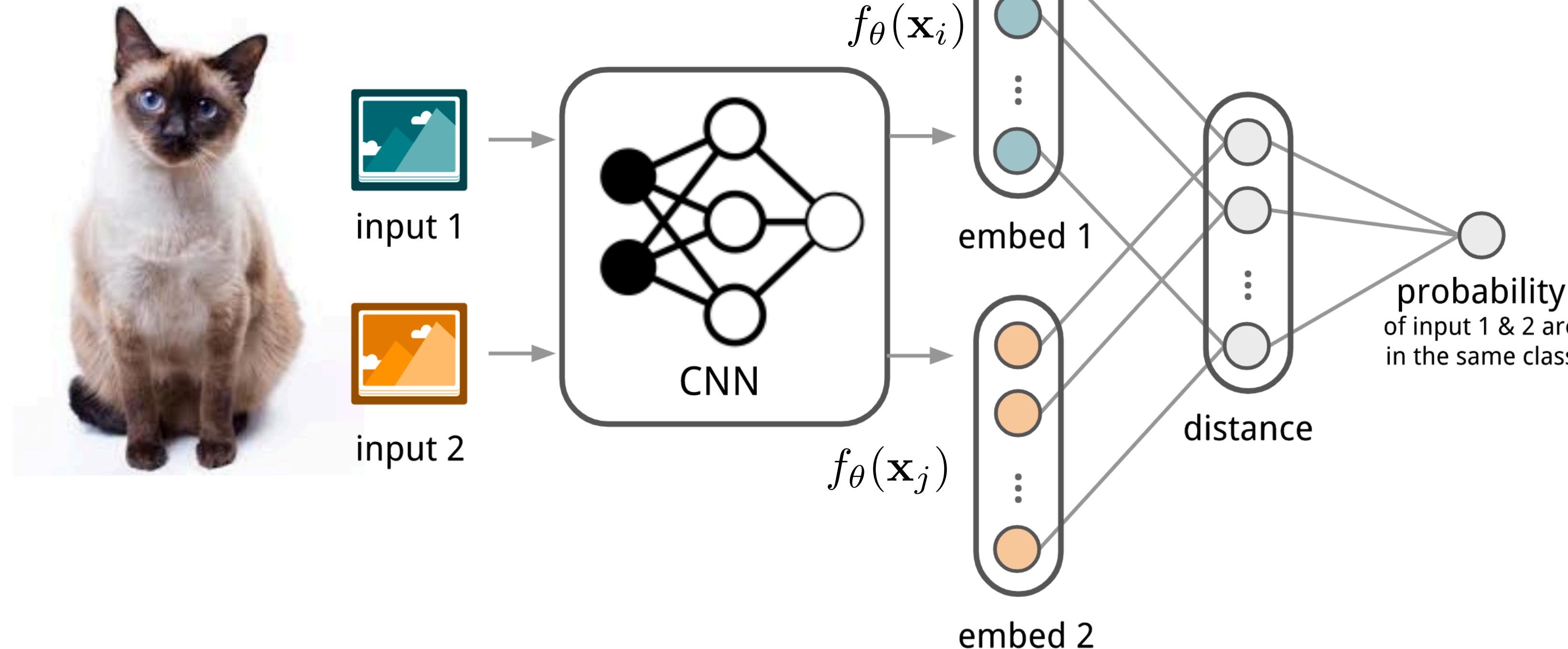
Few-shot Learning



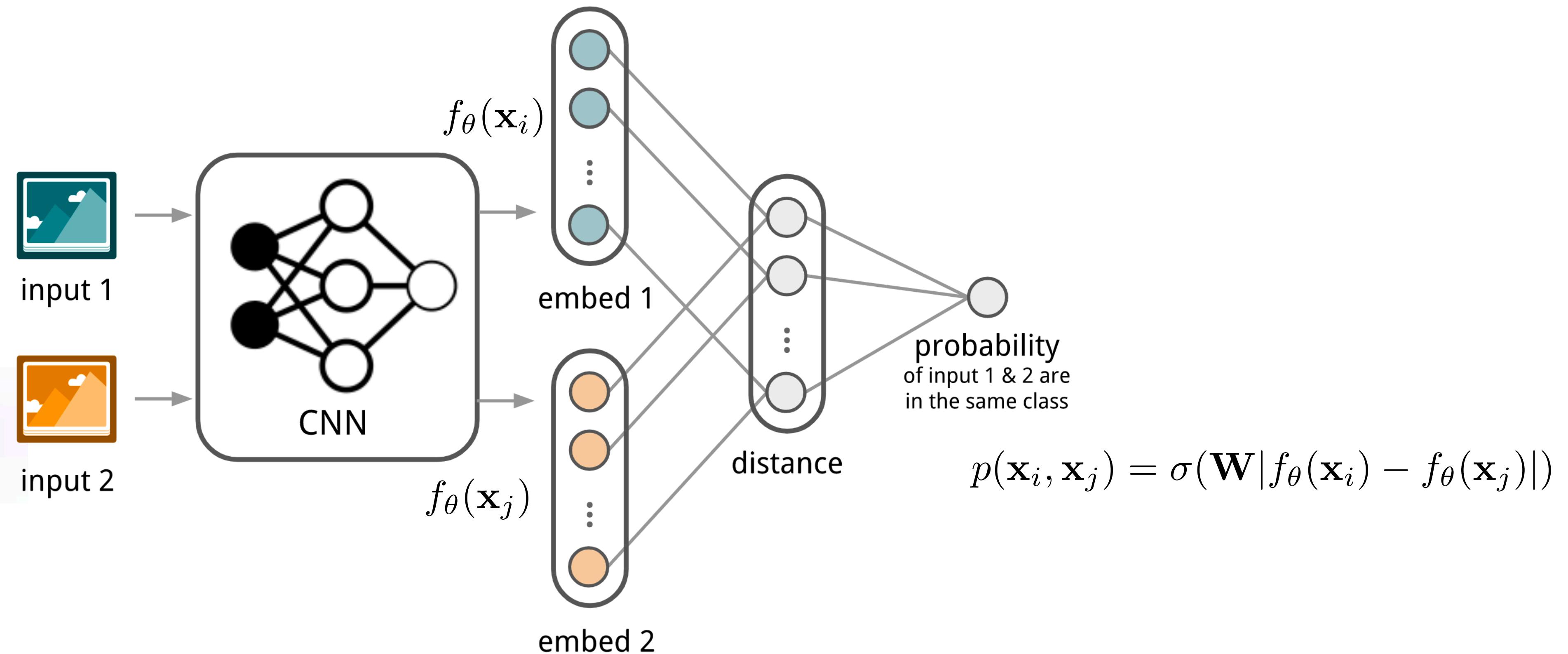
Meta-Learning



Metric-Based Meta-Learning – Siamese Neural Nets

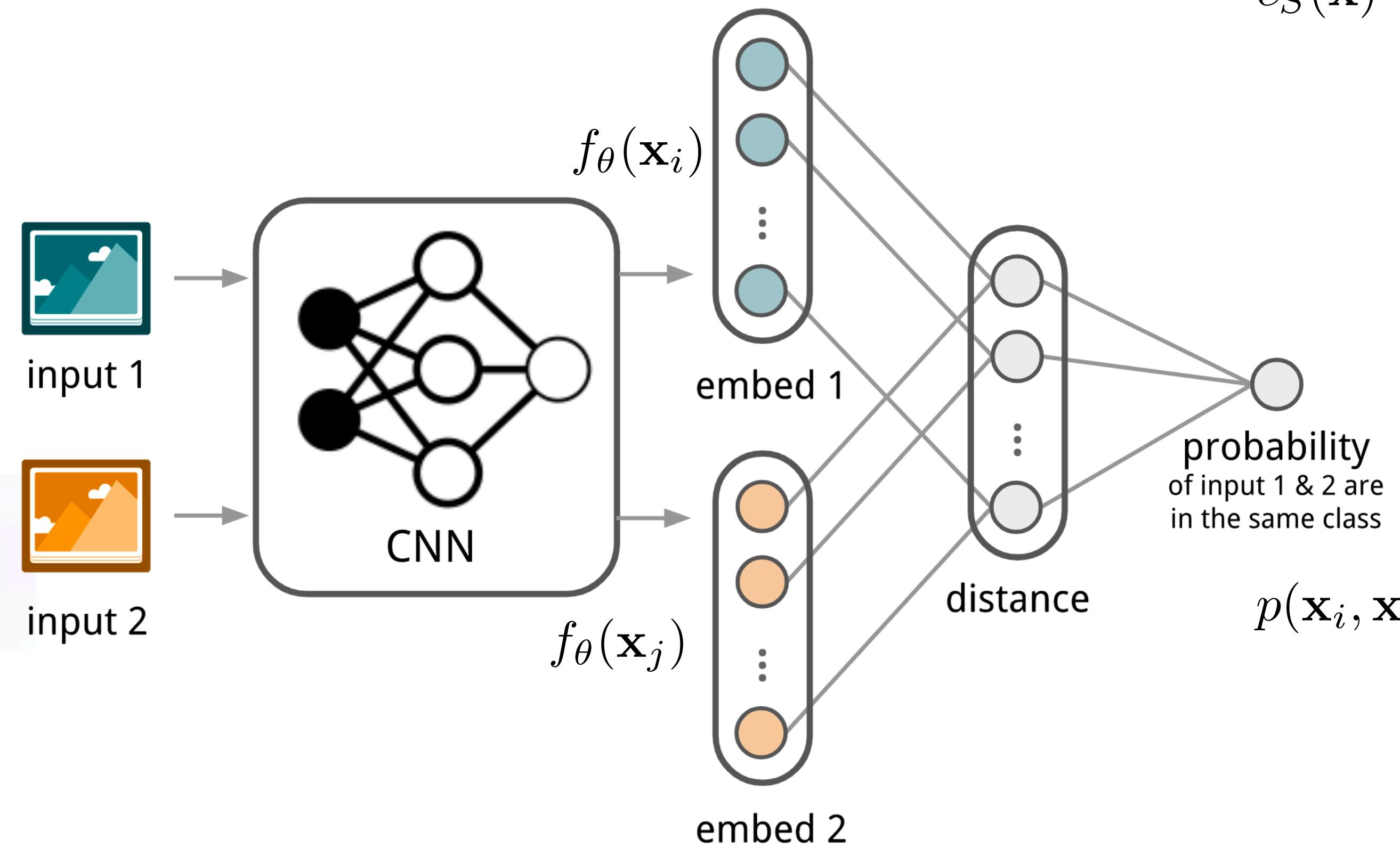


Metric-Based Meta-Learning – Siamese Neural Nets

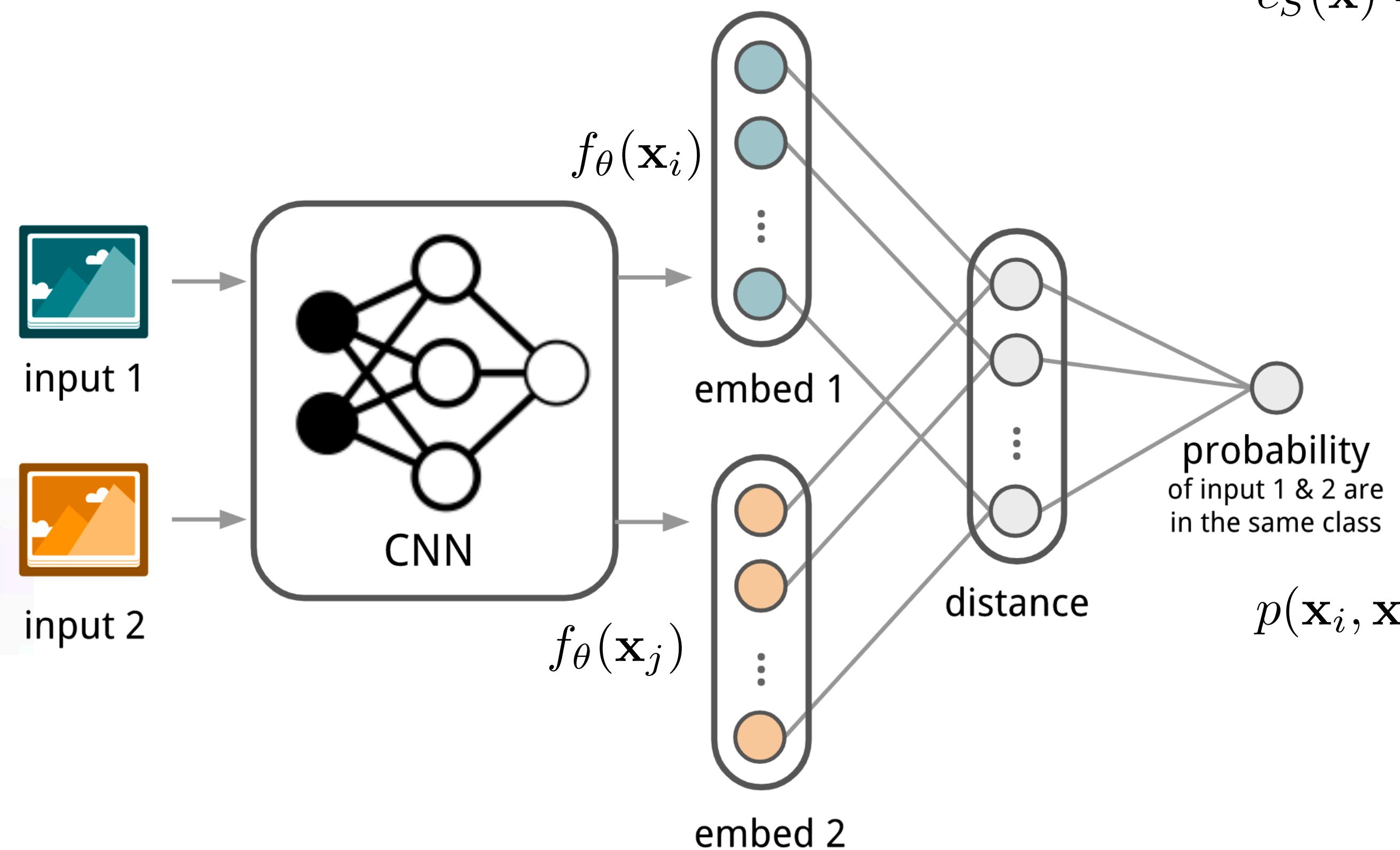


Training Objective: do two images belong to same class

Siamese Neural Nets



Siamese Neural Nets



Training Objective: do two images belong to same class

$$\mathcal{L}(B) = \sum_{(\mathbf{x}_i, \mathbf{x}_j, y_i, y_j) \in B} \mathbf{1}_{y_i=y_j} \log p(\mathbf{x}_i, \mathbf{x}_j) + (1 - \mathbf{1}_{y_i=y_j}) \log(1 - p(\mathbf{x}_i, \mathbf{x}_j))$$

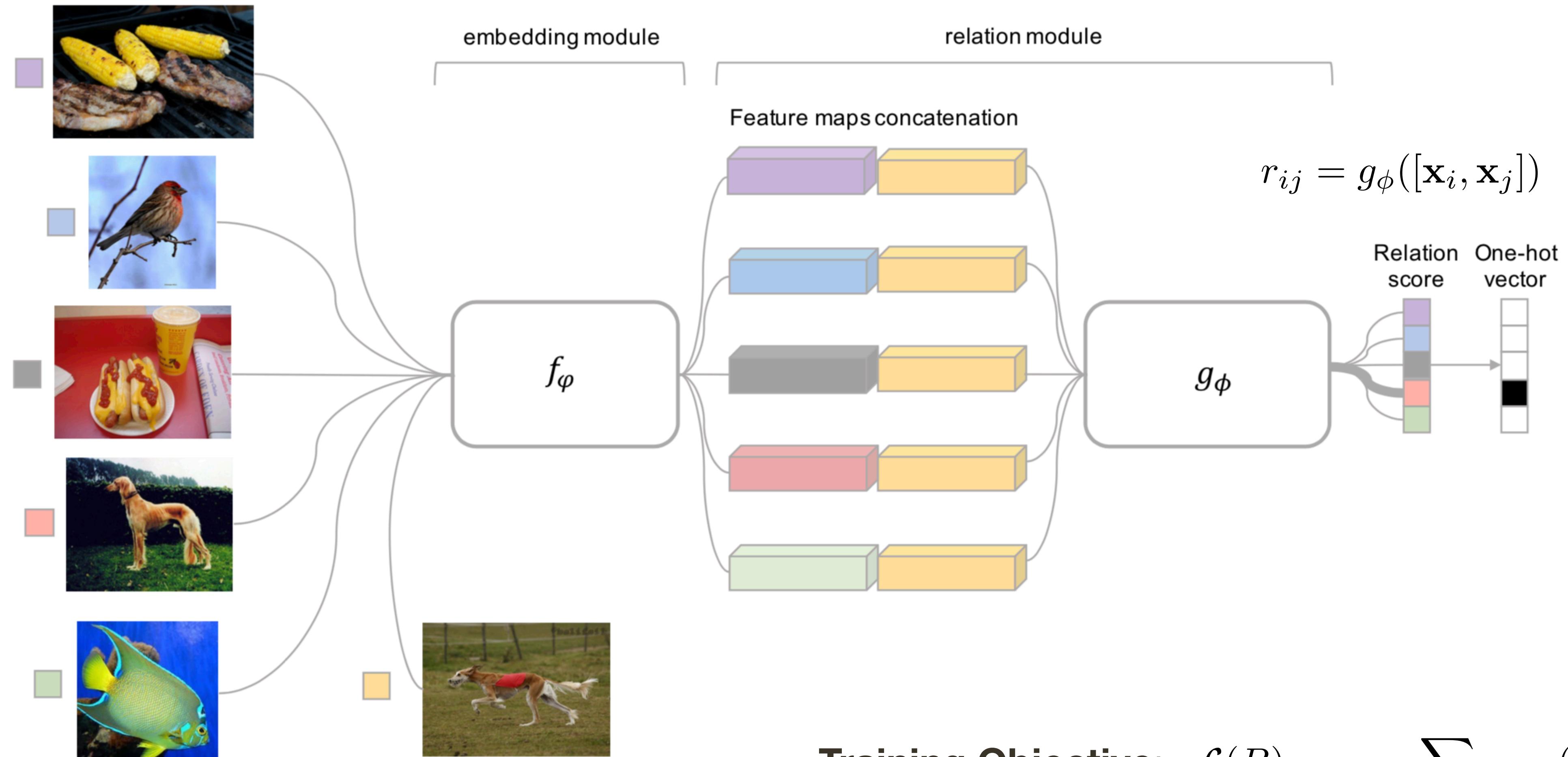
Inference: output the label of the most similar support image (i.e., nearest neighbor)

$$\hat{c}_S(\mathbf{x}) = c(\arg \max_{\mathbf{x}_i \in S} P(\mathbf{x}, \mathbf{x}_i))$$



[Koch et al. 2015]

Metric-Based Meta-Learning – Relation Networks

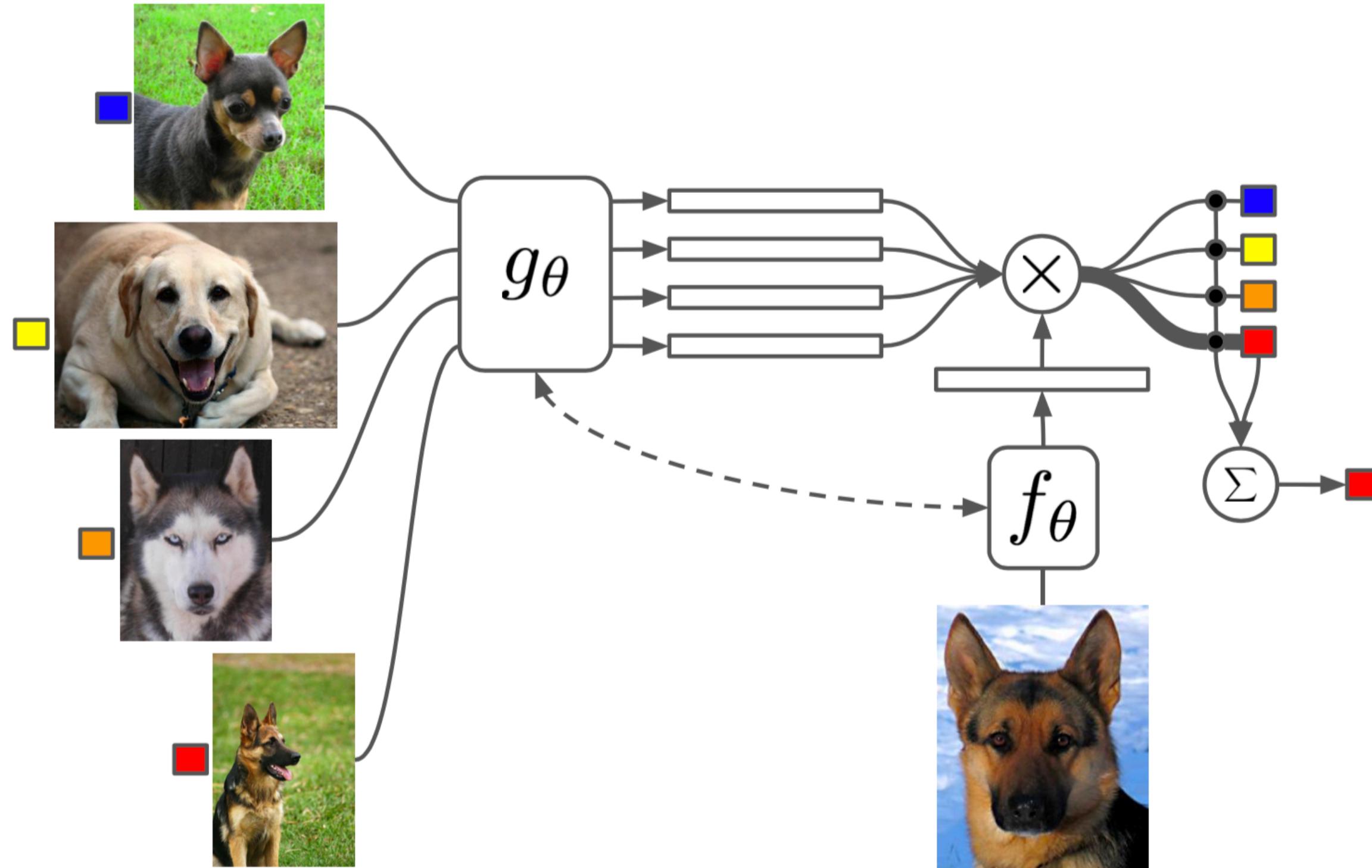


Training Objective: $\mathcal{L}(B) = \sum_{(\mathbf{x}_i, \mathbf{x}_j, y_i, y_j) \in B} (r_{ij} - \mathbf{1}_{y_i=y_j})^2$

Metric-Based Meta-Learning – Marching Networks

Inference: output the label distribution is simply sum of labels from support set, weighted by similarity/relevance

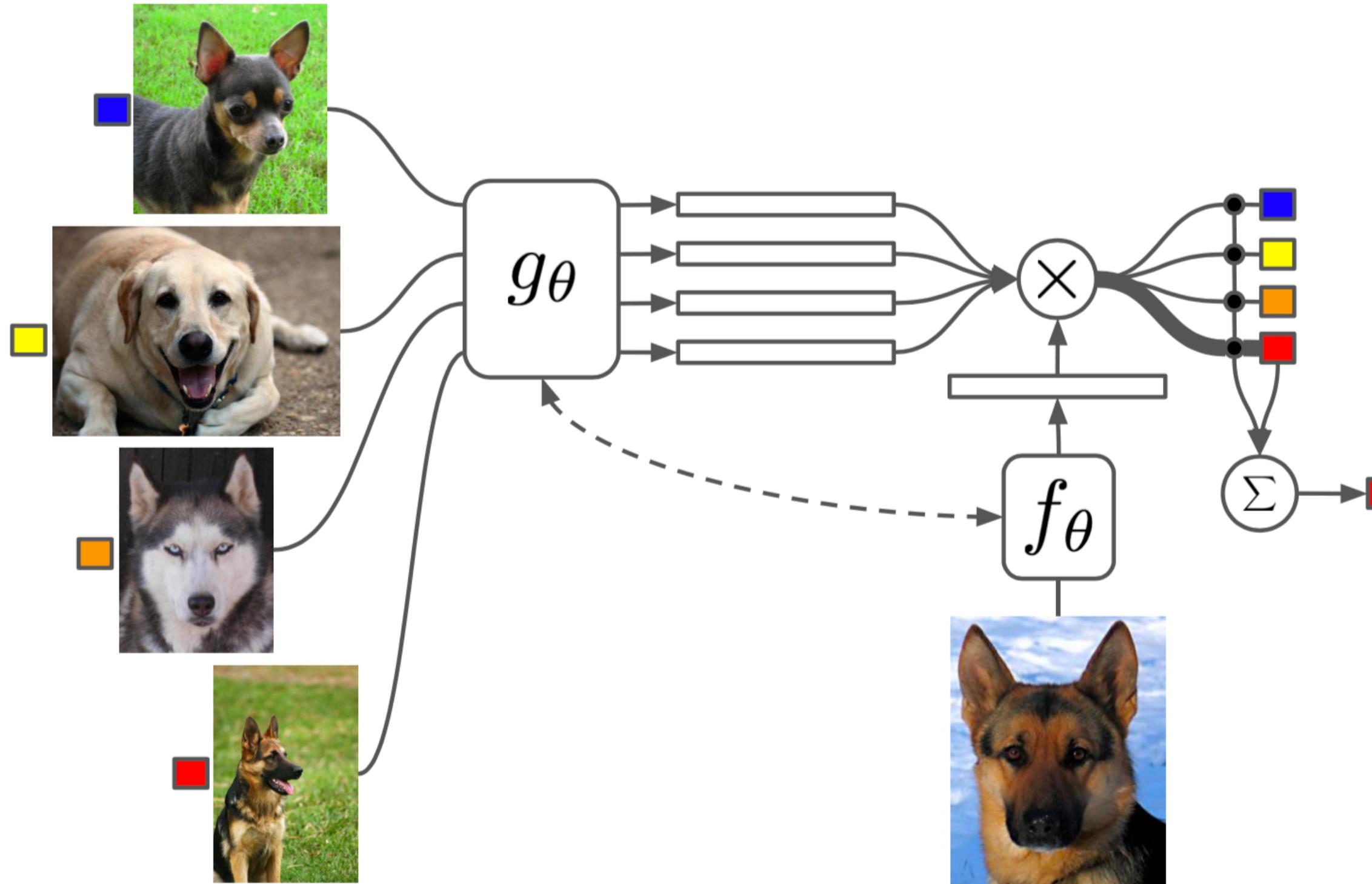
$$c_S(\mathbf{x}) = P(y|\mathbf{x}, S) = \sum_{i=1}^k a(\mathbf{x}, \mathbf{x}_i)y_i, \text{ where } S = \{(\mathbf{x}_i, y_i)\}_{i=1}^k$$



Metric-Based Meta-Learning – Marching Networks

Inference: output the label distribution is simply sum of labels from support set, weighted by similarity/relevance

$$c_S(\mathbf{x}) = P(y|\mathbf{x}, S) = \sum_{i=1}^k a(\mathbf{x}, \mathbf{x}_i) y_i, \text{ where } S = \{(\mathbf{x}_i, y_i)\}_{i=1}^k$$

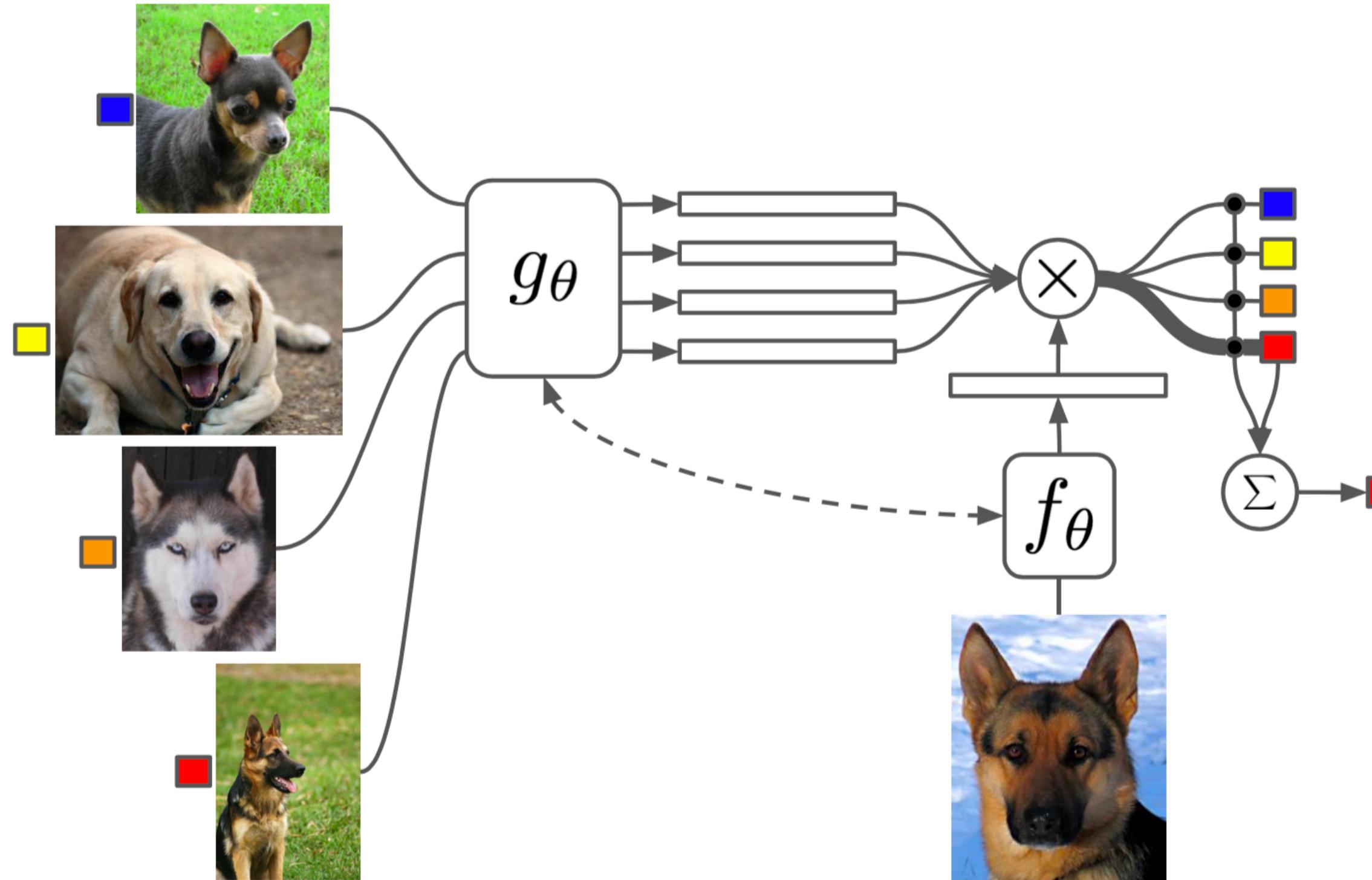


$$a(\mathbf{x}, \mathbf{x}_i) = \frac{\exp(\cosine(f(\mathbf{x}), g(\mathbf{x}_i)))}{\sum_{j=1}^k \exp(\cosine(f(\mathbf{x}), g(\mathbf{x}_j)))}$$

Metric-Based Meta-Learning – Marching Networks

Inference: output the label distribution is simply sum of labels from support set, weighted by similarity/relevance

$$c_S(\mathbf{x}) = P(y|\mathbf{x}, S) = \sum_{i=1}^k a(\mathbf{x}, \mathbf{x}_i) y_i, \text{ where } S = \{(\mathbf{x}_i, y_i)\}_{i=1}^k$$



$$a(\mathbf{x}, \mathbf{x}_i) = \frac{\exp(\cosine(f(\mathbf{x}), g(\mathbf{x}_i)))}{\sum_{j=1}^k \exp(\cosine(f(\mathbf{x}), g(\mathbf{x}_j)))}$$

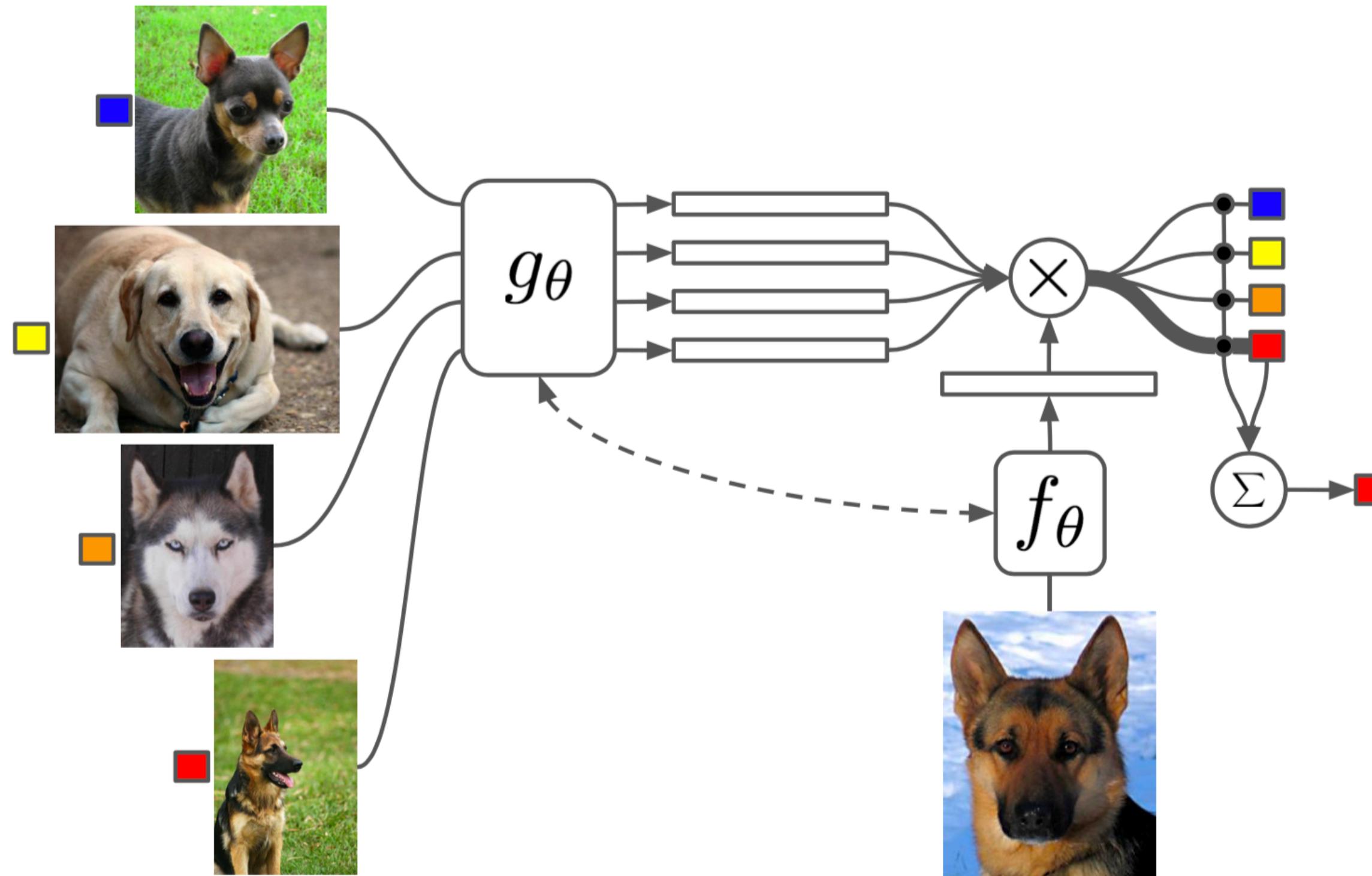
Simple version: $f = g$
+ soft-attention

Key = features of support set images
Value = labels of support set images
Query = features of test image

Metric-Based Meta-Learning – Marching Networks

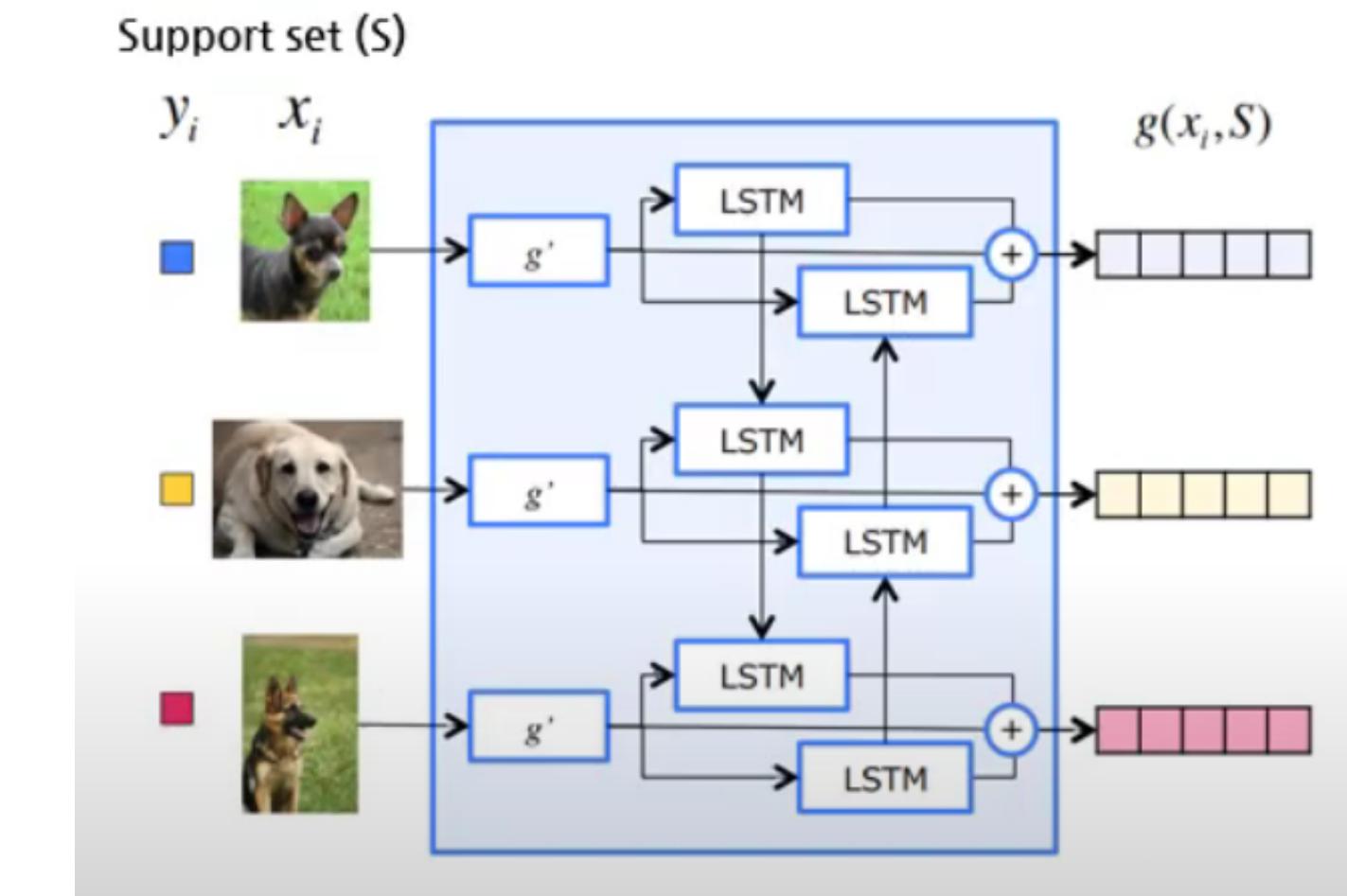
Inference: output the label distribution is simply sum of labels from support set, weighted by similarity/relevance

$$c_S(\mathbf{x}) = P(y|\mathbf{x}, S) = \sum_{i=1}^k a(\mathbf{x}, \mathbf{x}_i) y_i, \text{ where } S = \{(\mathbf{x}_i, y_i)\}_{i=1}^k$$



$$a(\mathbf{x}, \mathbf{x}_i) = \frac{\exp(\text{cosine}(f(\mathbf{x}), g(\mathbf{x}_i)))}{\sum_{j=1}^k \exp(\text{cosine}(f(\mathbf{x}), g(\mathbf{x}_j)))}$$

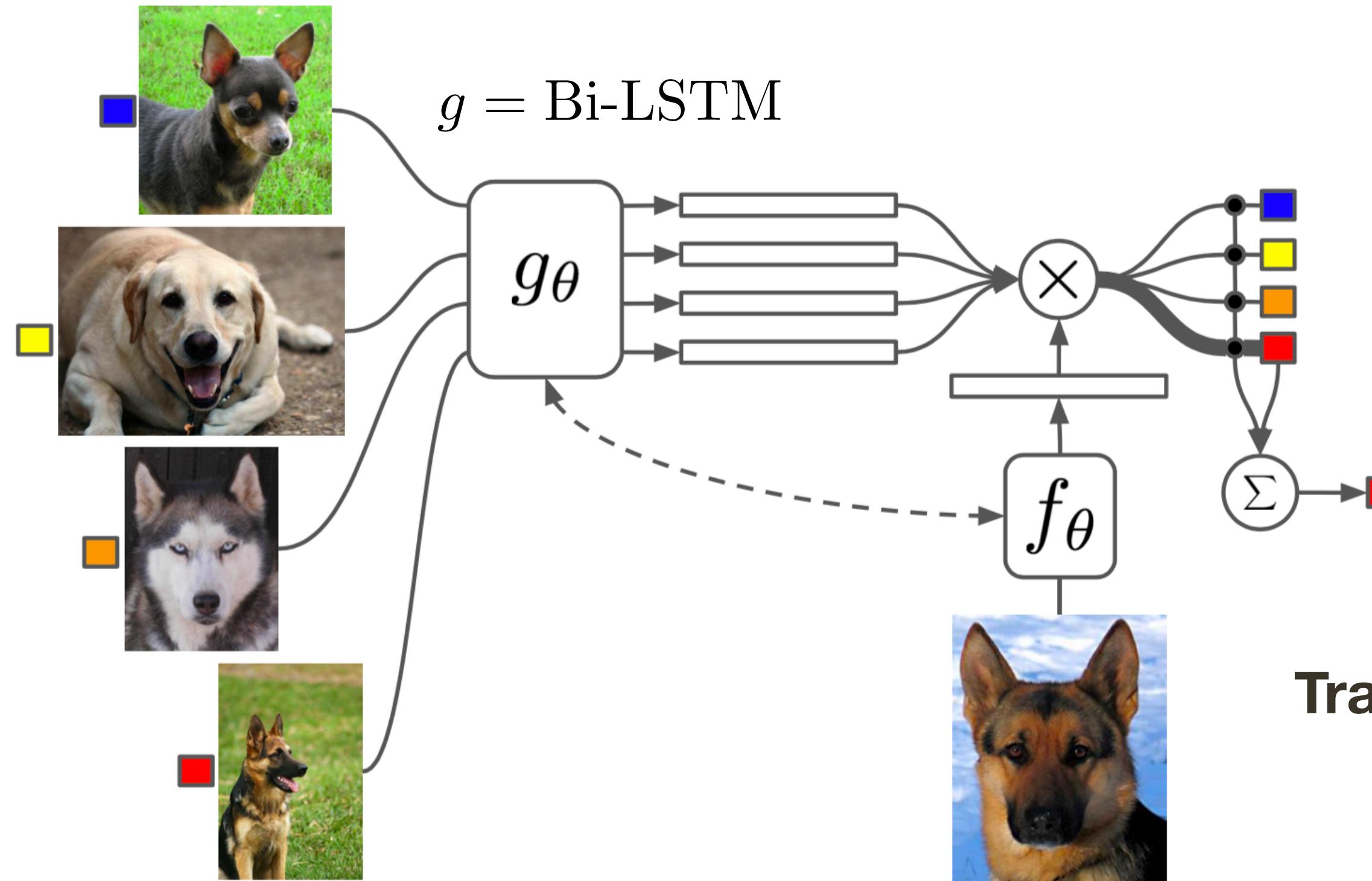
$g = \text{Bi-LSTM}$



Metric-Based Meta-Learning – Marching Networks

Inference: output the label distribution is simply sum of labels from support set, weighted by similarity/relevance

$$c_S(\mathbf{x}) = P(y|\mathbf{x}, S) = \sum_{i=1}^k a(\mathbf{x}, \mathbf{x}_i) y_i, \text{ where } S = \{(\mathbf{x}_i, y_i)\}_{i=1}^k$$



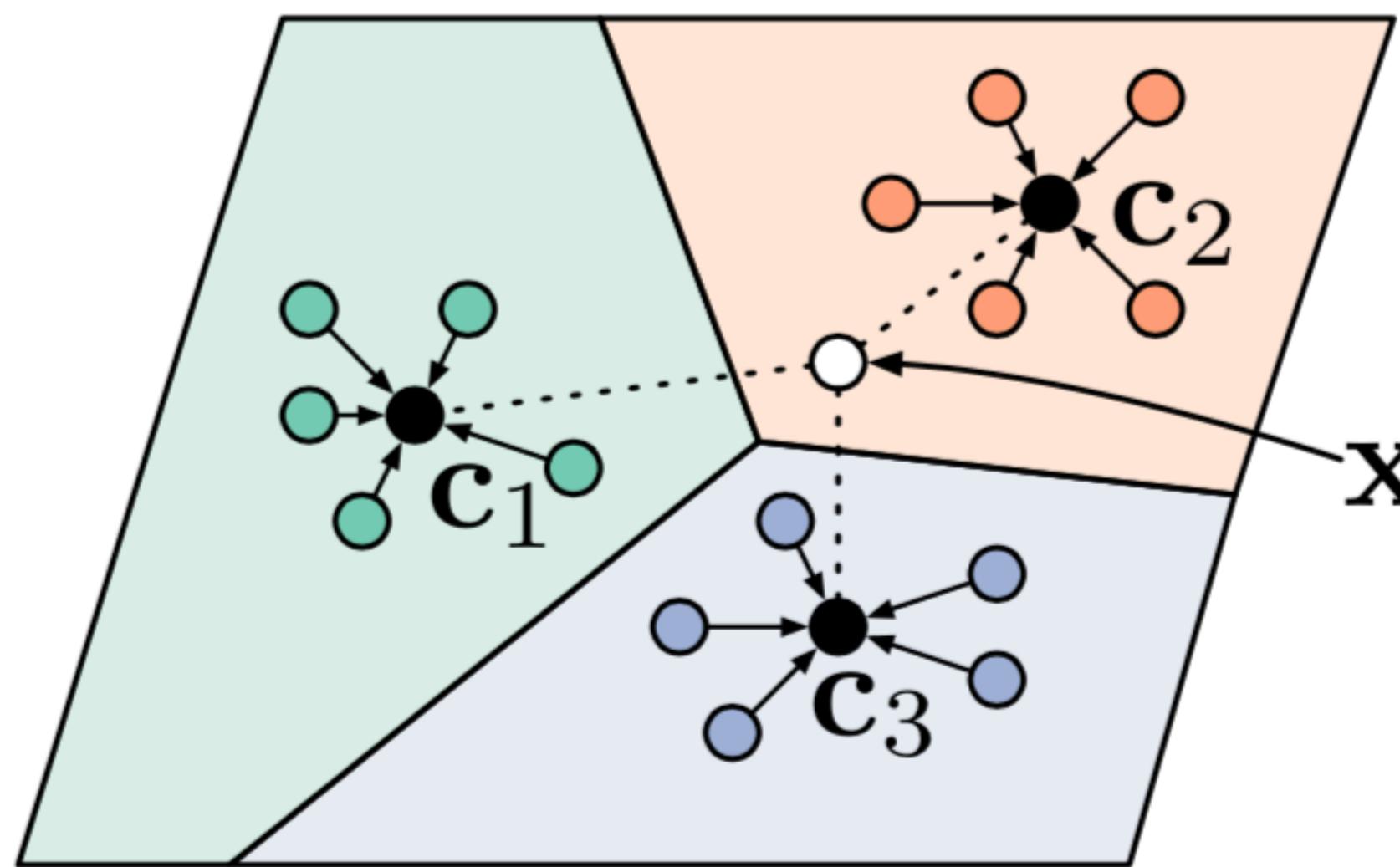
$$a(\mathbf{x}, \mathbf{x}_i) = \frac{\exp(\cosine(f(\mathbf{x}), g(\mathbf{x}_i)))}{\sum_{j=1}^k \exp(\cosine(f(\mathbf{x}), g(\mathbf{x}_j)))}$$

Training Objective: correct classification of query examples

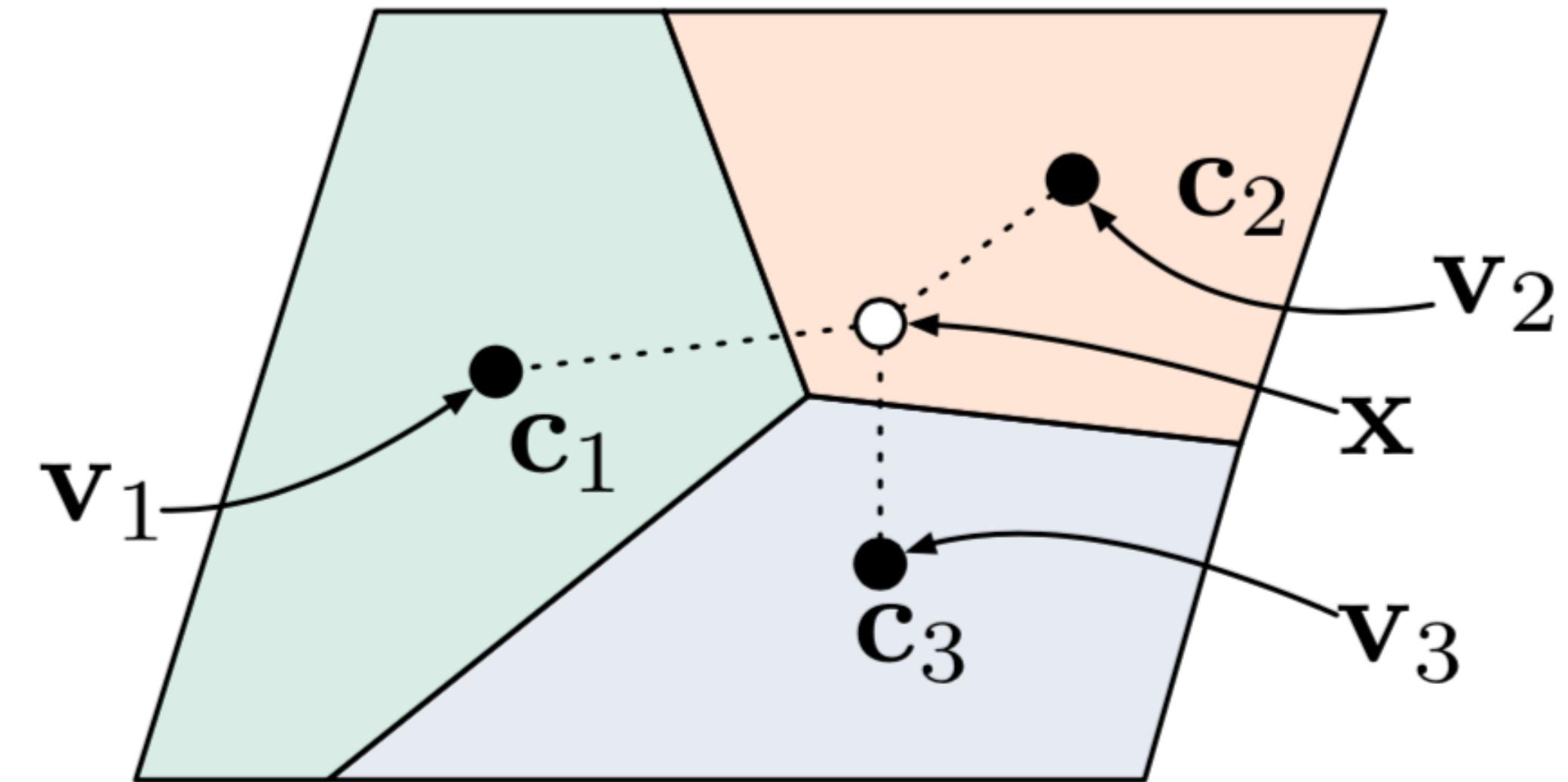
$$\theta^* = \arg \max_{\theta} \mathbb{E}_{L \subset \mathcal{L}} [\mathbb{E}_{S^L \subset \mathcal{D}, B^L \subset \mathcal{D}} [\sum_{(\mathbf{x}, y) \in B^L} P_\theta(y|\mathbf{x}, S^L)]]$$

Metric-Based Meta-Learning – Prototypical Networks

$$\mathbf{v}_c = \frac{1}{|S_c|} \sum_{(\mathbf{x}_i, y_i) \in S_c} f_\theta(\mathbf{x}_i)$$



(a) Few-shot



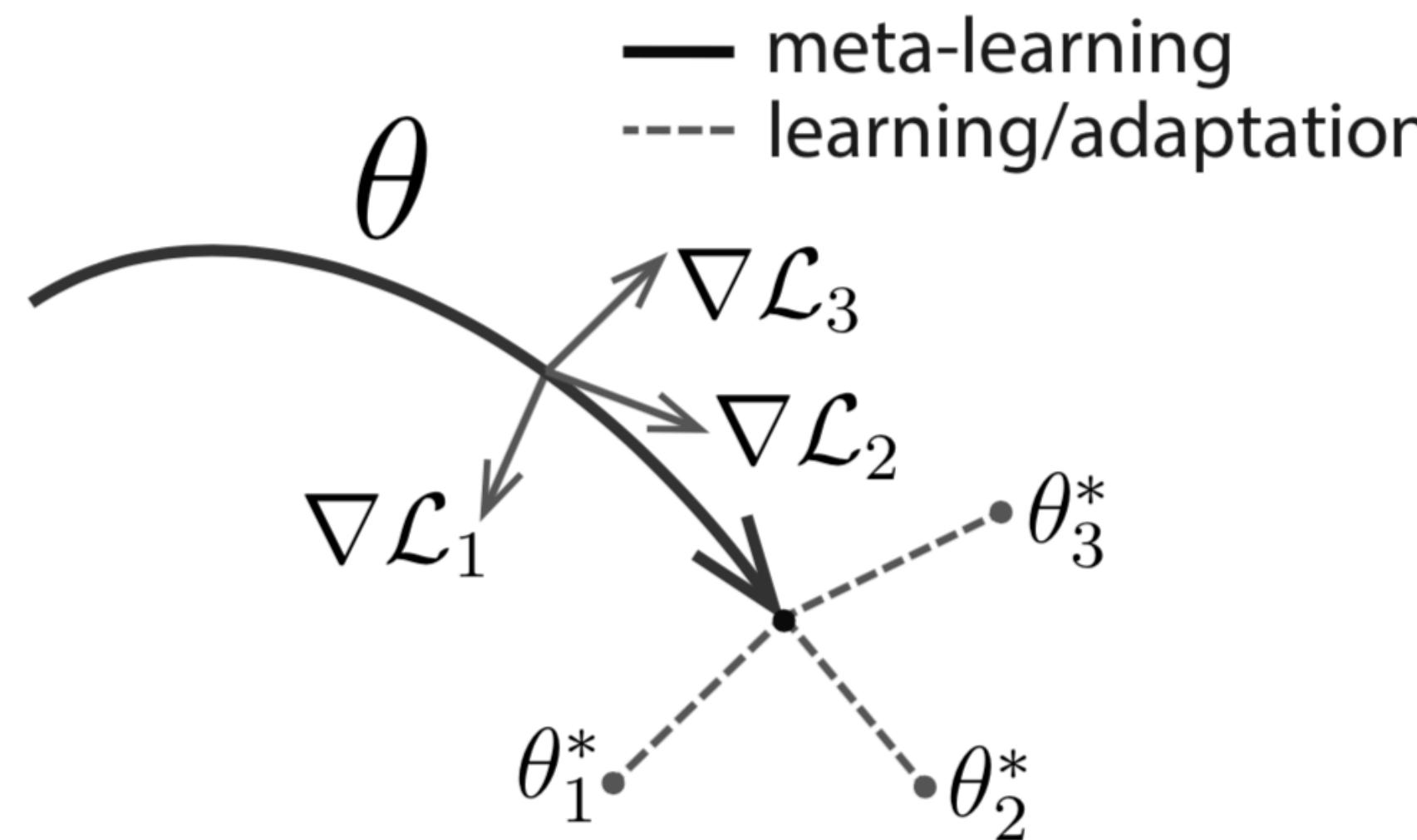
(b) Zero-shot

$$P(y = c|\mathbf{x}) = \text{softmax}(-d_\varphi(f_\theta(\mathbf{x}), \mathbf{v}_c)) = \frac{\exp(-d_\varphi(f_\theta(\mathbf{x}), \mathbf{v}_c))}{\sum_{c' \in \mathcal{C}} \exp(-d_\varphi(f_\theta(\mathbf{x}), \mathbf{v}_{c'}))}$$

$$\mathcal{L}(\theta) = -\log P_\theta(y = c|\mathbf{x})$$

Optimization-Based Meta-Learning — MAML

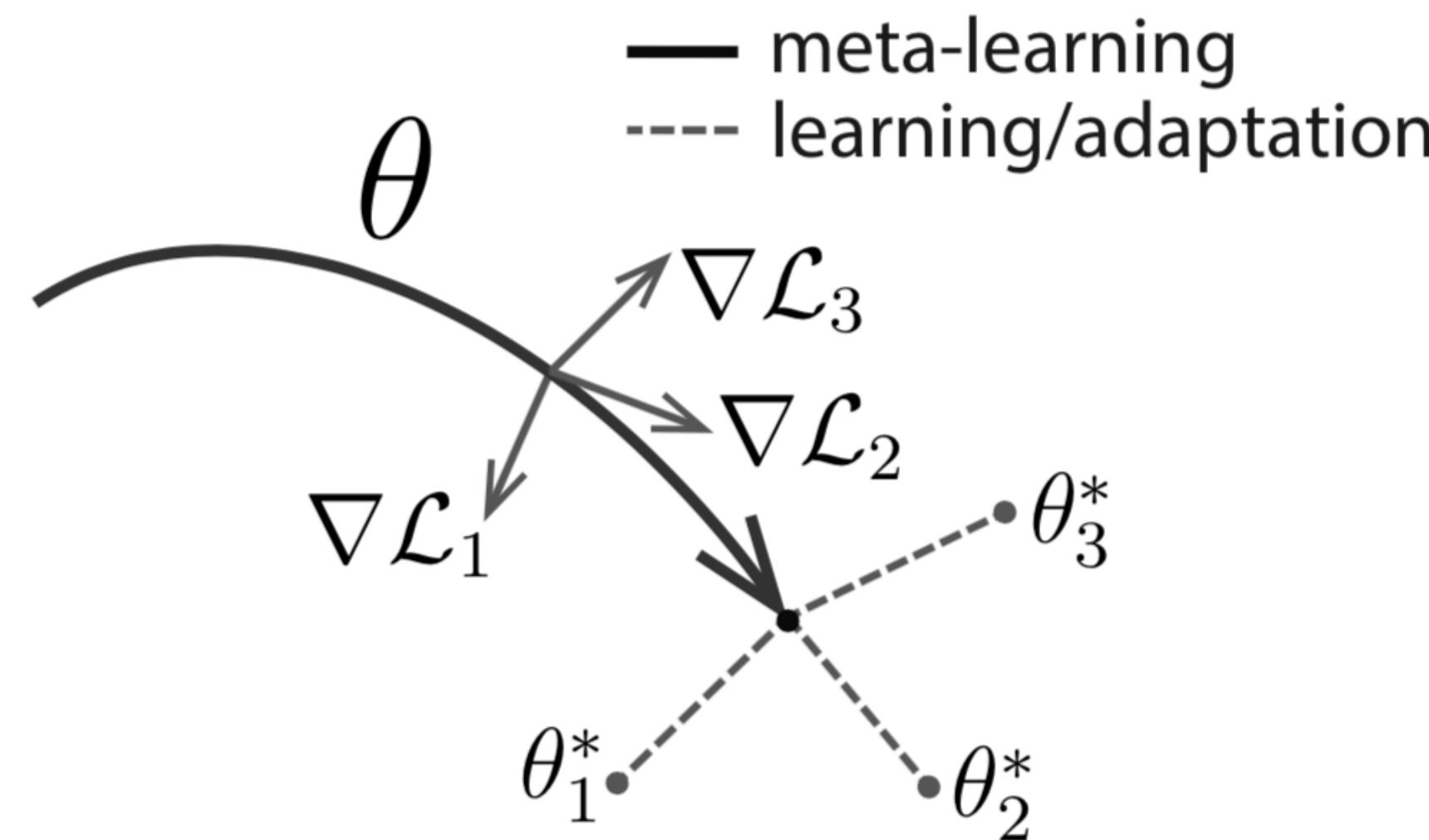
Idea: learn model initialization from which one can rapidly adopt to ANY meta-task



Optimization-Based Meta-Learning — MAML

Idea: learn model initialization from which one can rapidly adopt to ANY meta-task

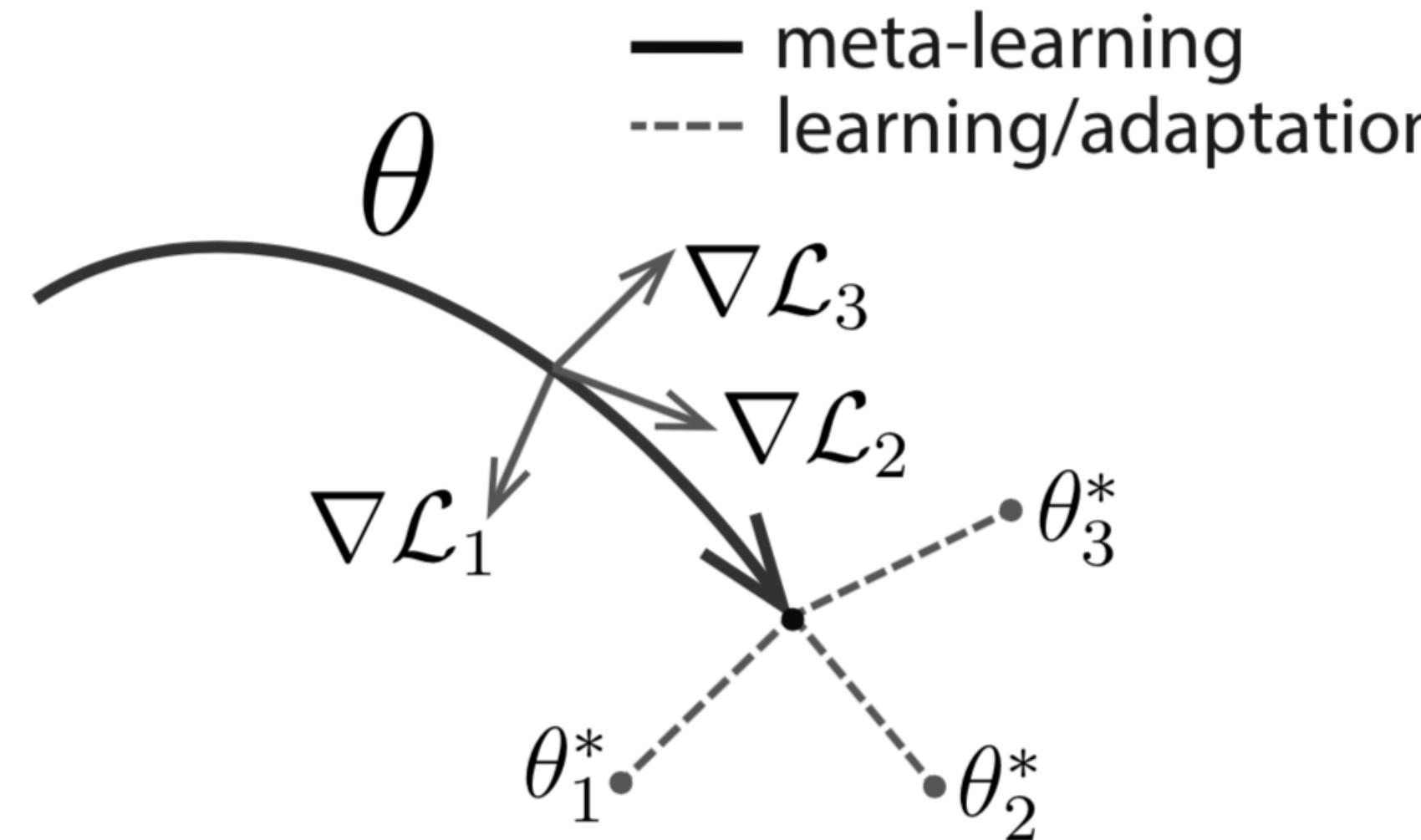
Means: find model parameters that are sensitive to changes in the task



Optimization-Based Meta-Learning — MAML

Idea: learn model initialization from which one can rapidly adopt to ANY meta-task

Means: find model parameters that are sensitive to changes in the task



Adapting parameters for one task for k steps:

$$\theta_0 = \theta_{\text{meta}}$$

$$\theta_1 = \theta_0 - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_0)$$

$$\theta_2 = \theta_1 - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_1)$$

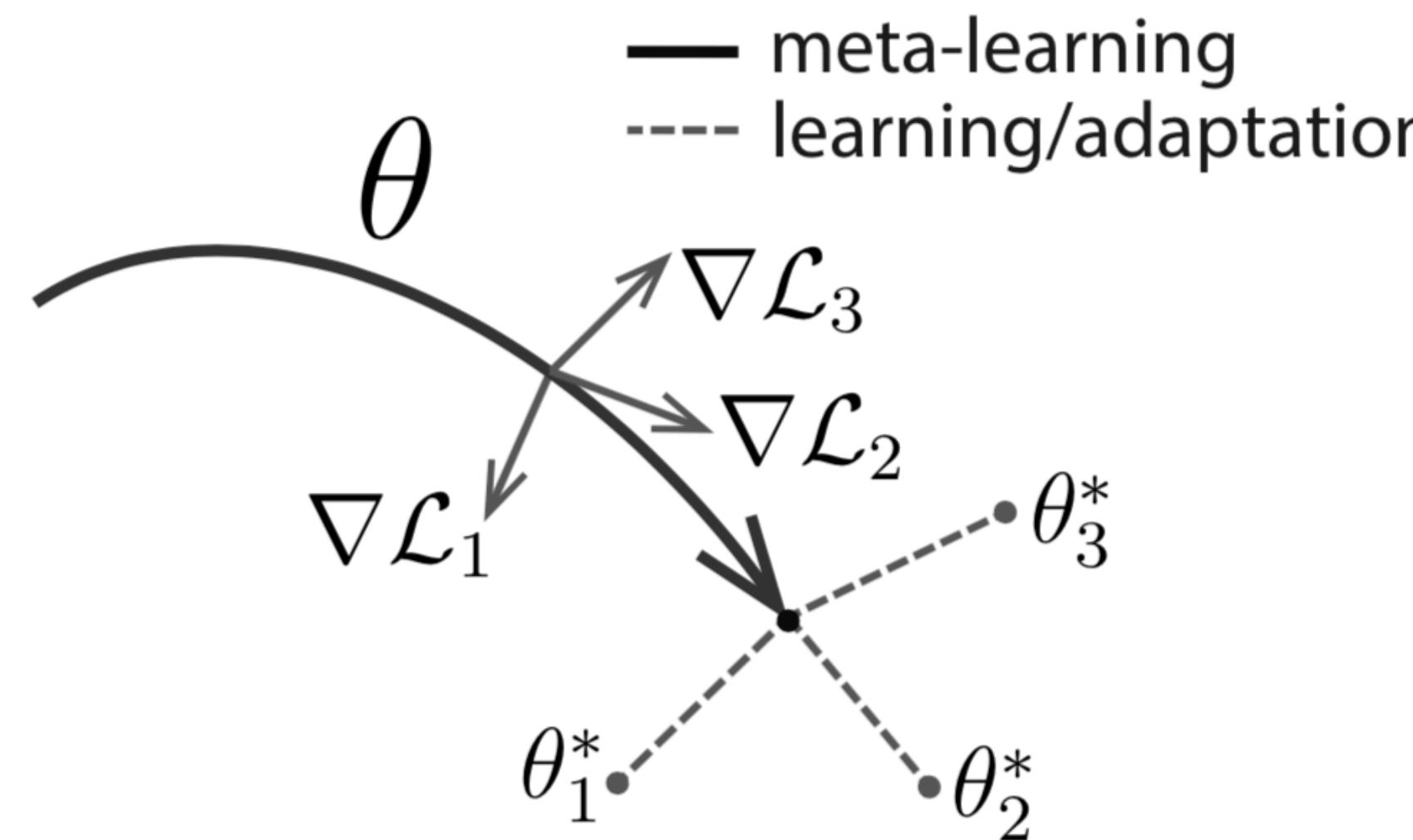
...

$$\theta_k = \theta_{k-1} - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_{k-1})$$

Optimization-Based Meta-Learning — MAML

Idea: learn model initialization from which one can rapidly adopt to ANY meta-task

Means: find model parameters that are sensitive to changes in the task



Adapting parameters for one task for k steps:

$$\theta_0 = \theta_{\text{meta}}$$

$$\theta_1 = \theta_0 - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_0)$$

$$\theta_2 = \theta_1 - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_1)$$

...

$$\theta_k = \theta_{k-1} - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_{k-1})$$

Optimizing meta-parameters:

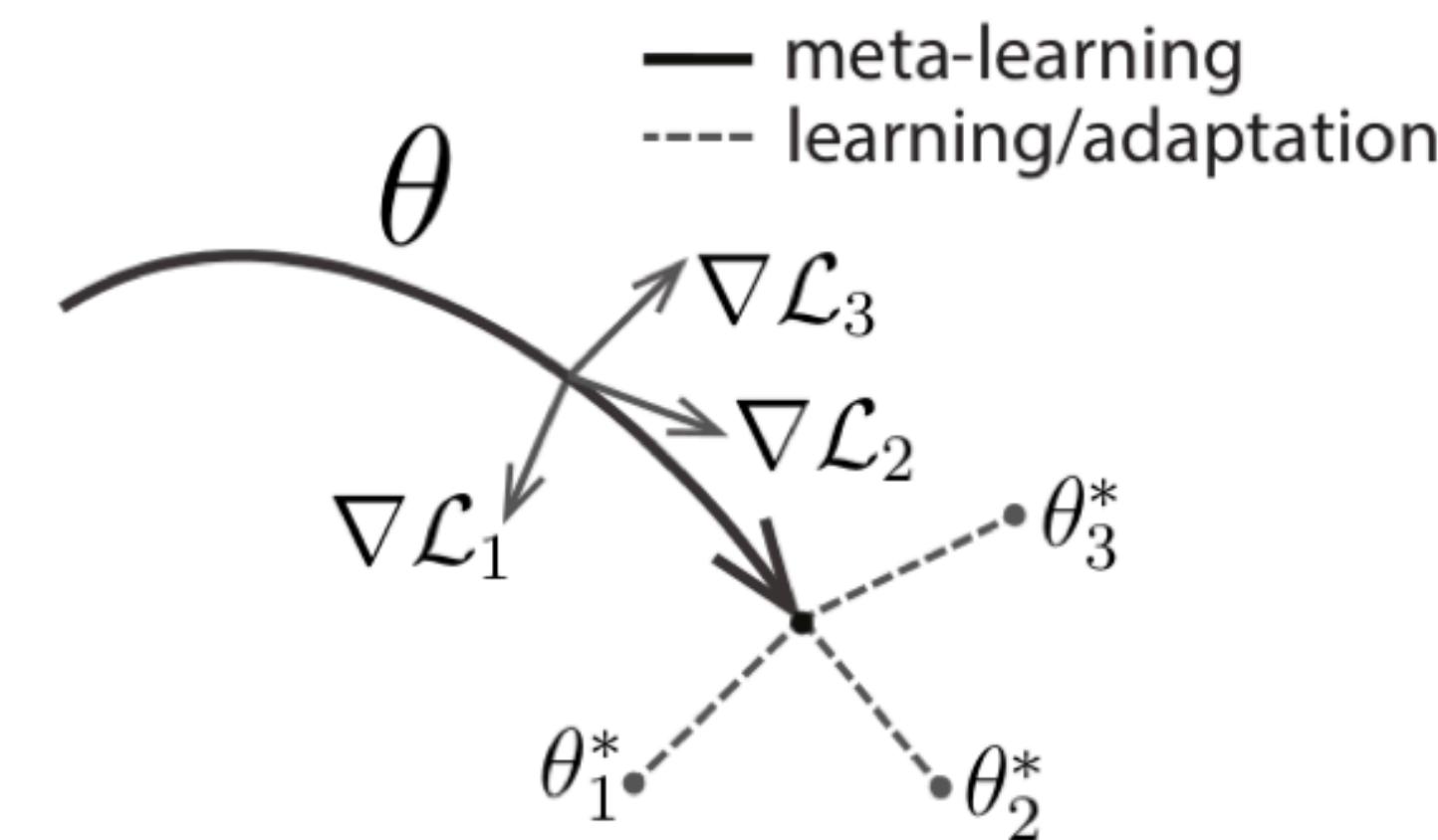
$$\theta_{\text{meta}} \leftarrow \theta_{\text{meta}} - \beta g_{\text{MAML}}$$

Optimization-Based Meta-Learning — MAML

Inner Loop: Update the model for a task from an initialization

Outer Loop: Optimize for the performance of all **inner loop** models on **all tasks**

Intuition: We want achieve a low loss after only a few updates on a task



MAML — Algorithm

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks

Require: α, β : step size hyperparameters

1: randomly initialize θ

2: **while** not done **do**

3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$

MAML — Inner Loop

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks

Require: α, β : step size hyperparameters

- 1: randomly initialize θ
- 2: **while** not done **do**
- 3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
- 4: **for all** \mathcal{T}_i **do**
- 5: Evaluate $\nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$ with respect to K examples
- 6: Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$
- 7: **end for**

Inner Loop: Update the model for a task from an initialization

$$\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$$

Simple gradient update on the sampled task

MAML — Outer Loop

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks

Require: α, β : step size hyperparameters

- 1: randomly initialize θ
 - 2: **while** not done **do**
 - 3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
 - 4: **for all** \mathcal{T}_i **do**
 - 5: Evaluate $\nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$ with respect to K examples
 - 6: Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$
 - 7: **end for**
 - 8: Update $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$
 - 9: **end while**
-

Meta-objective:

$$\min_{\theta} \underbrace{\sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})}_{\text{Total loss of all updated models}} = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})})$$

Meta-update:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \underbrace{\sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})}_{\text{Total loss of all updated models}}$$

MAML — Issues

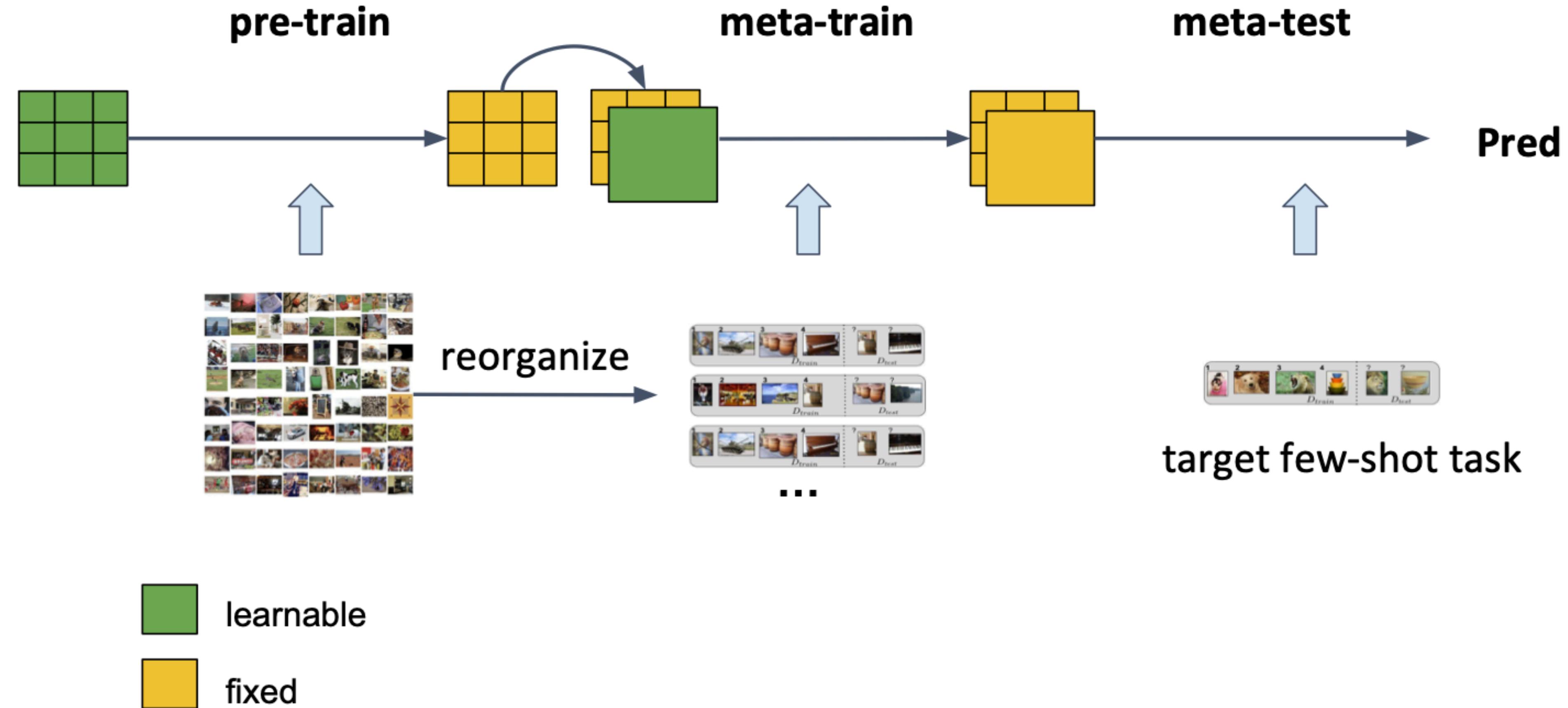
- Hard to train with deep feature extractor networks

Solution: Meta-transfer learning

- Slow training

Solution: Hard task sampling (will not cover)

A more typical pipeline ...



Multi-modal Few-shot Learners — Flamingo

