# Large Scale Vision-Language Pretraining: Models and Applications

Wangmeng Zuo

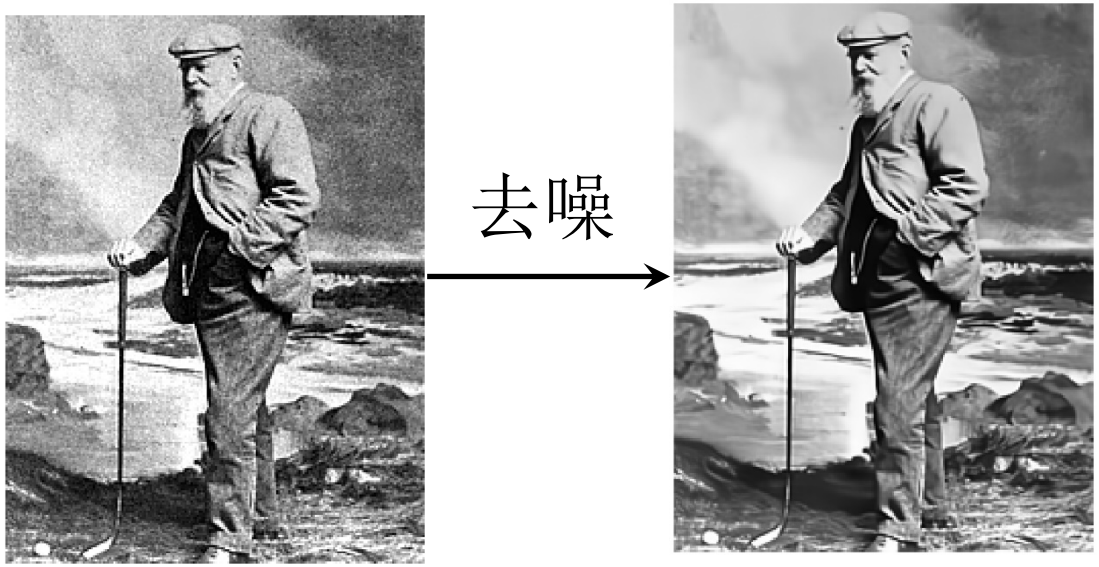
Centre on Machine Learning Research

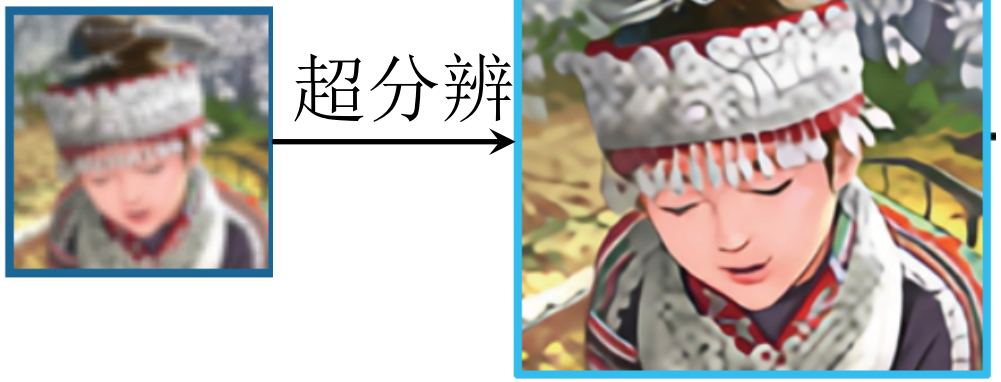
Harbin Institute of Technology

# Vision-Language Pretraining

- Vision-Language Models
- Classification: Parameter-efficient Fine-tuning
- More Challenging Tasks
- More Visual Modalities

# 典型视觉学习任务（底层视觉）

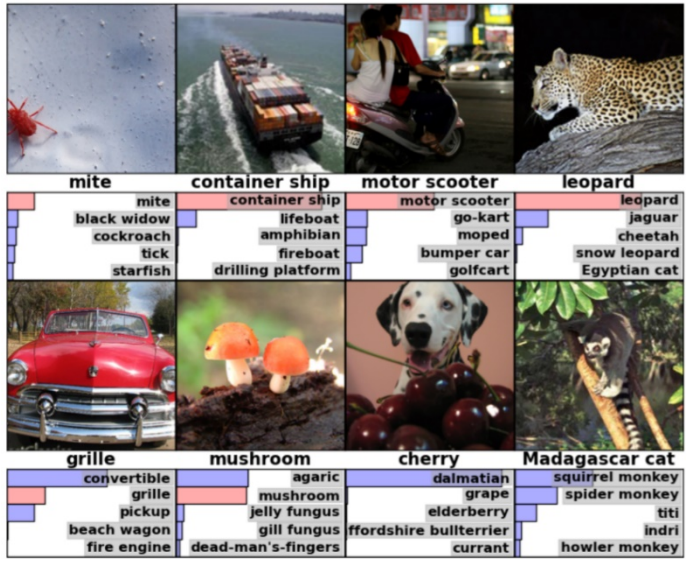



风格迁移

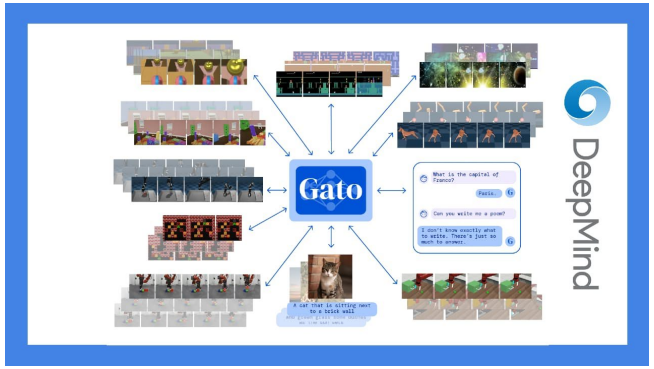

修复

# 典型视觉学习任务（视觉理解）




图像级分类


边界框级检测





物体关系预测

像素级分割

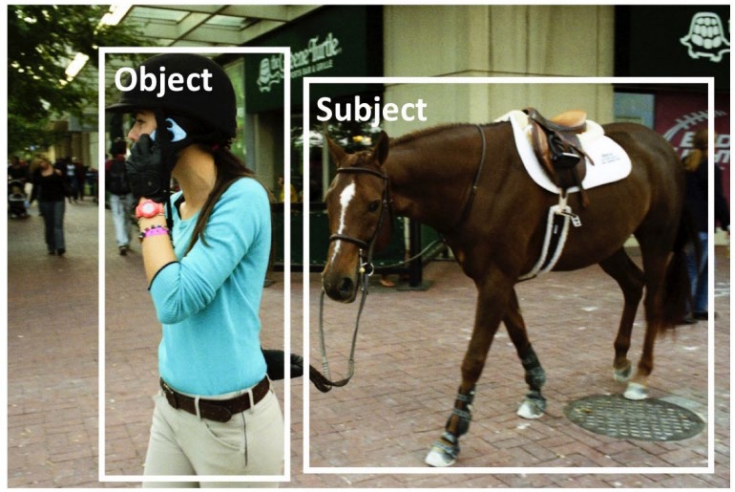
# 典型视觉任务（视觉-语言）

自然语言描述

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."




指代消解

视觉问答

Q: Does this foundation have any sunscreen?
A: yes

Q: What is this?
A: 10 euros

Q: What color is this?
A: green


Q: Please can you tell me what this item is?
A: butternut squash red pepper soup

Q: Is it sunny outside?
A: yes

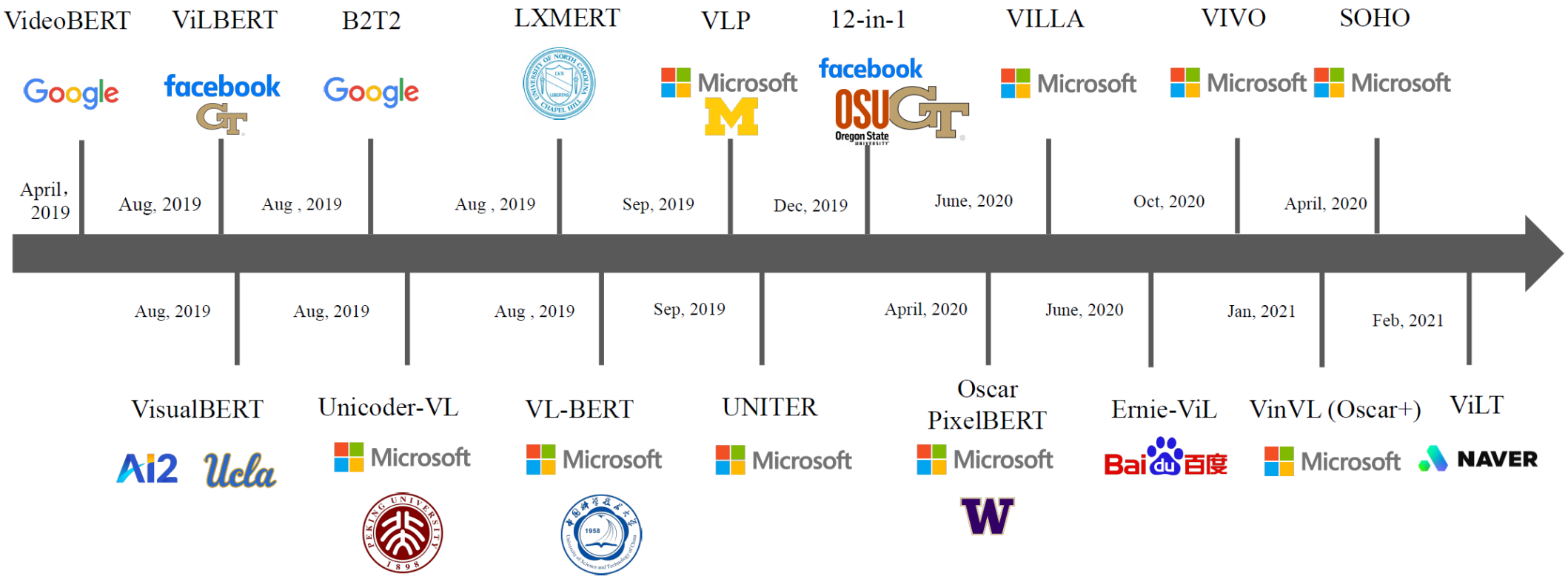
Q: Is this air conditioner on fan, dehumidifier, or air conditioning?
A: air conditioning

# Vision-Language Pretraining

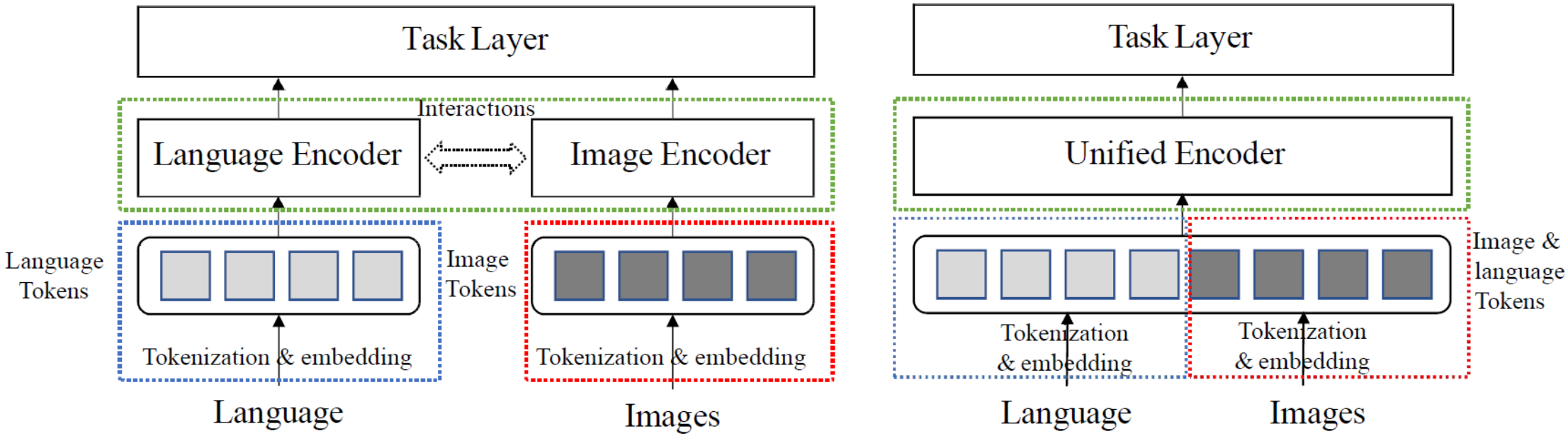
- Early VL Pretraining

- Recent Breakthrough

# Landscape of V-L Pretraining Methods



# Architectures of VLP Models



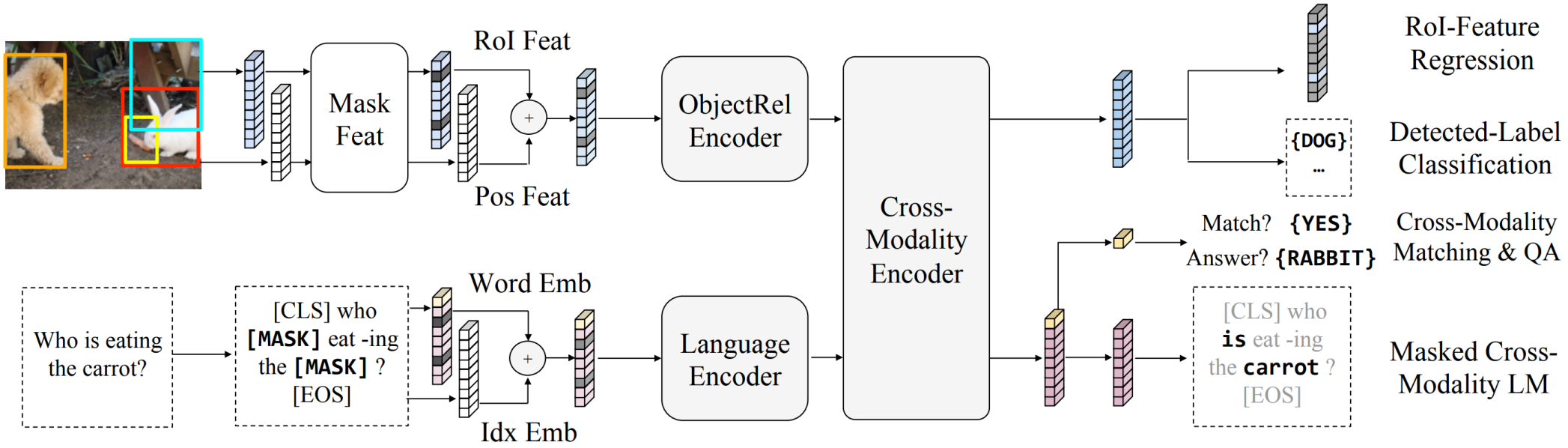

(a) Dual Stream Models

(b) Single Stream Models

# Two-stream Architectures

| | | | | | |
|---|---|---|---|---|---|
| Dual-Stream | ViLBERT (2019) | BUTD | ITM, MLM<br>MRC-kl | CC3M | VQA, VR, RE<br>IR, Zero-shot IR |
| | LXMERT (2019) | BUTD | ITM, MLM, MRP, MRC | COCO, VG (2017a), VQA v2<br>GQA, Visual7W (2016) | VQA, VR |
| | 12-in-1 (2020) | BUTD | ITM,MLM,MRC-kl | VQA v2, Flickr30k<br>SNLI-VE, COCO<br>GuessWhat, VG<br>RefCOCO, RefCOCO+,RefCOCOG<br>Visual 7W, GQA, NLVR$^2$ | VQA ,IR, RE, VE, VR |
| | Ernie-ViL (2020) | BUTD | Object Prediction<br>Attribute Prediction<br>Relationship Prediction<br>ITM, MLM,MRC-kl | CC3M, SBU (out-of-domain)<br>COCO, VG (in-domain) | VQA, VR, IT, TR, RE |

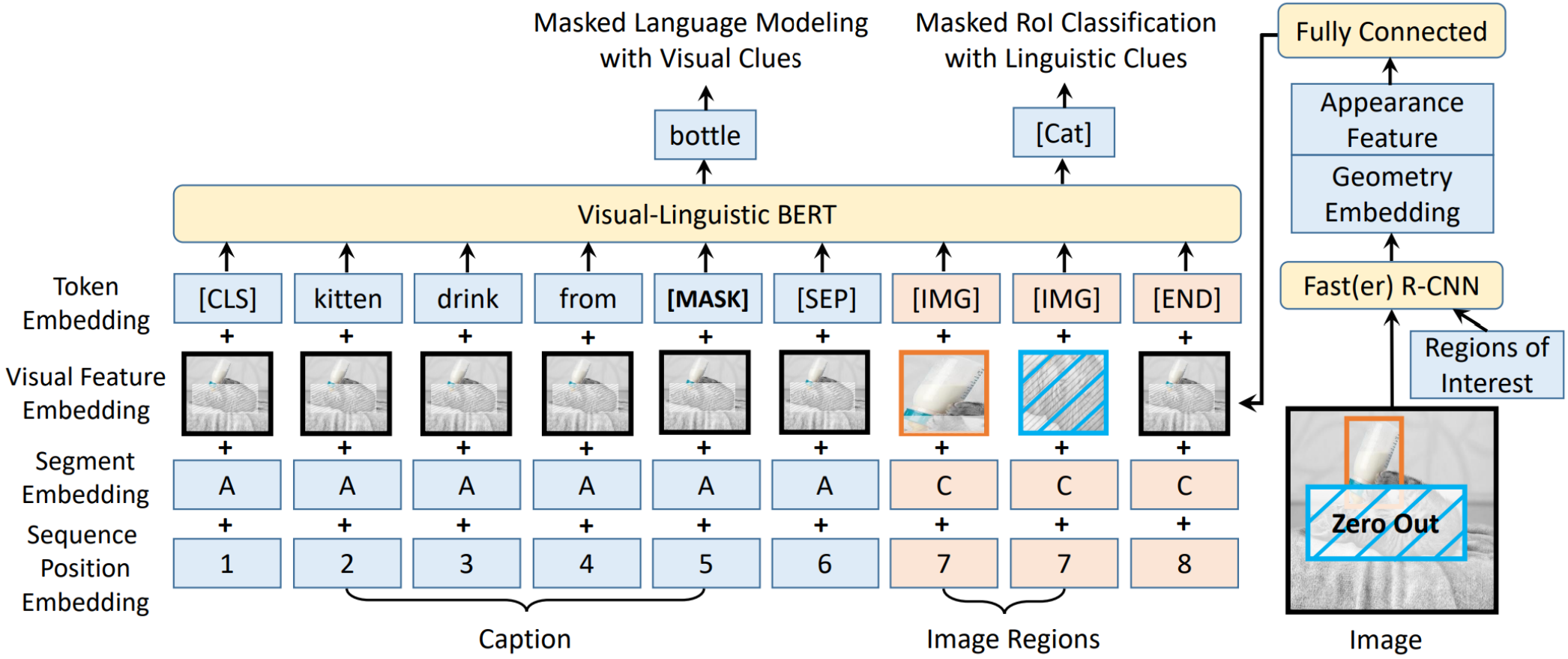
# LXMERT Training



# One-stream Architectures

| | | | | | |
|---|---|---|---|---|---|
| Single-Stream | VideoBERT (2019a) | S3D (2017b) | ITM, MLM, MVM | YouTube cooking videos (2017b) | Zero-shot Action prediction (2017b)<br>Video Captioning (2017b) |
| | VisualBERT (2019b) | Pre-trained Fast R-CNN | ITM, MLM | COCO | VQA, VR, PG |
| | B2T2 (2019) | ResNet-152 (2016b) | ITM, MLM | CC3M | VR |
| | Unicoder-VL (2019a) | Pre-trained<br>Faster-RCNN (2018) | ITM, MLM, MRC | CC3M, SBU | IR, TR, VR<br>Zero-shot IR/TR |
| | VL-BERT (2019) | BUTD | MLM, MRP | CC3M, English Wikipedia<br>BooksCorpus (2015) | VQA, VR, RE |
| | Unified VLP (2020) | BUTD variant<br>(with ResNext-101<br>backbone) (2017a) | MLM (Sequentially<br>and bidirectionally) | CC3M | IC, VQA |
| | UNITER (2019) | BUTD | ITM, MLM, MRP, MRC<br>MRC-kl | COCO, VG, CC3M, SBU | VQA, VR, VE, IR, TR, RE |
| | Oscar (2020b) | BUTD+tags | MLM (include tags)<br>ITM (pollute tags) | COCO, VG, CC3M,<br>SBU, fliker30k, GQA | VQA, VR, VE, IR, TR<br>RE, IC |
| | PixelBERT (2020) | Pixel feature embedding (2015) | MLM, ITM | COCO, VG | VQA, IR, TR, VR |
| | VILLA (2020) | BUTD | ITM, MLM, MRC-kl | COCO, VG, CC3M, SBU | VQA, VR, VE, IR, TR, RE |
| | VIVO (2020) | BUTD | Mask Tag Prediction<br>(Hungarian match loss) | Open Images V5 (2018; 2019) | IC |
| | SOHO (2021) | VD | ITM, MLM<br>MVMVD | COCO, VG | IR, TR, VQA<br>VR, VE (based on VD) |
| | ViLT (2021) | Patch Projection | ITM, MLM | COCO, VG, SBU, CC | VQA, VR,<br>IR, TR,<br>Zero-shot IR&TR |
| Hybrid | SemVLP (2021a) | Pre-tained Faster-RCNN | MLM, MRP, VQA | COCO, VG, VQA v2,<br>GQA, Visual7W | VQA, VR, IR, TR |

# VL-BERT

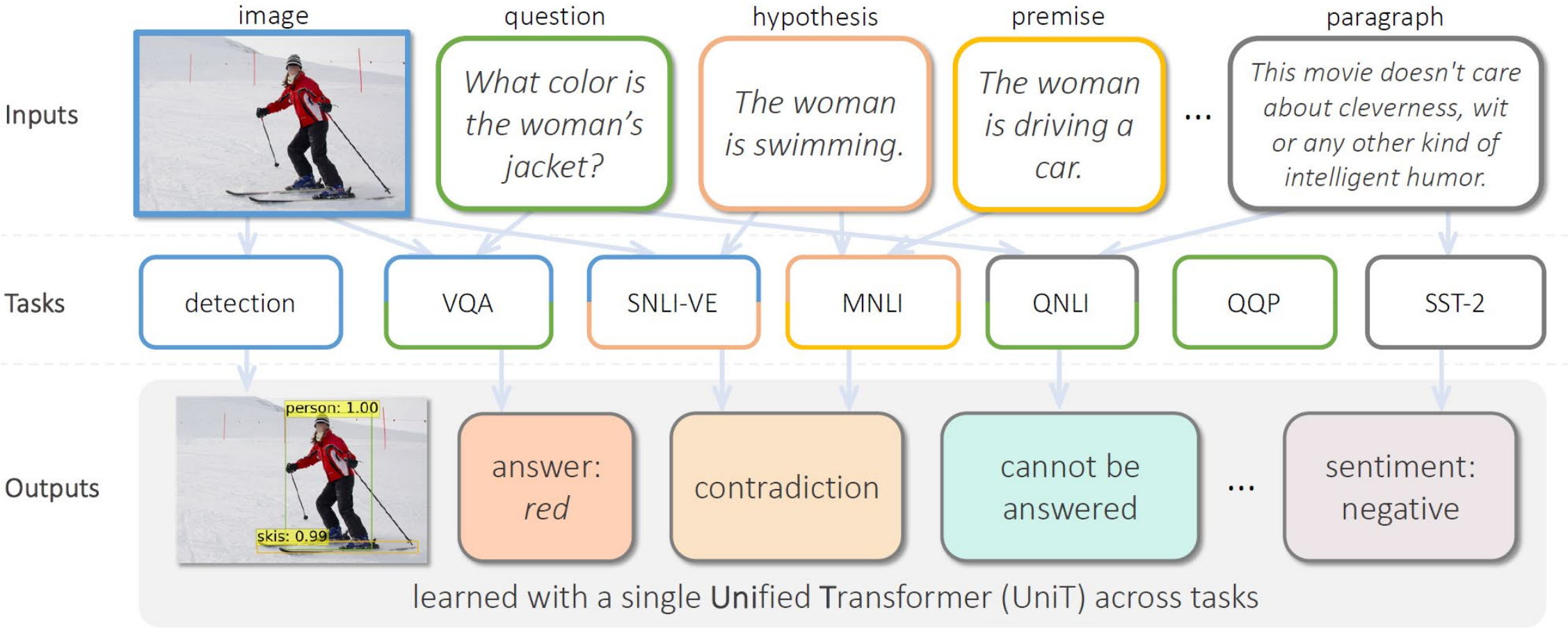

Masked Language Modeling with Visual Clues
Masked RoI Classification with Linguistic Clues
bottle
[Cat]
Visual-Linguistic BERT
Token Embedding
Visual Feature Embedding
Segment Embedding
Sequence Position Embedding
[CLS]
kitten
drink
from
[MASK]
[SEP]
[IMG]
[IMG]
[END]
A
C
1
2
3
4
5
6
7
8
Caption
Image Regions
Fully Connected
Appearance Feature
Geometry Embedding
Fast(er) R-CNN
Regions of Interest
Zero Out
Image

# Vision-Language Pretraining

- Early VL Pretraining

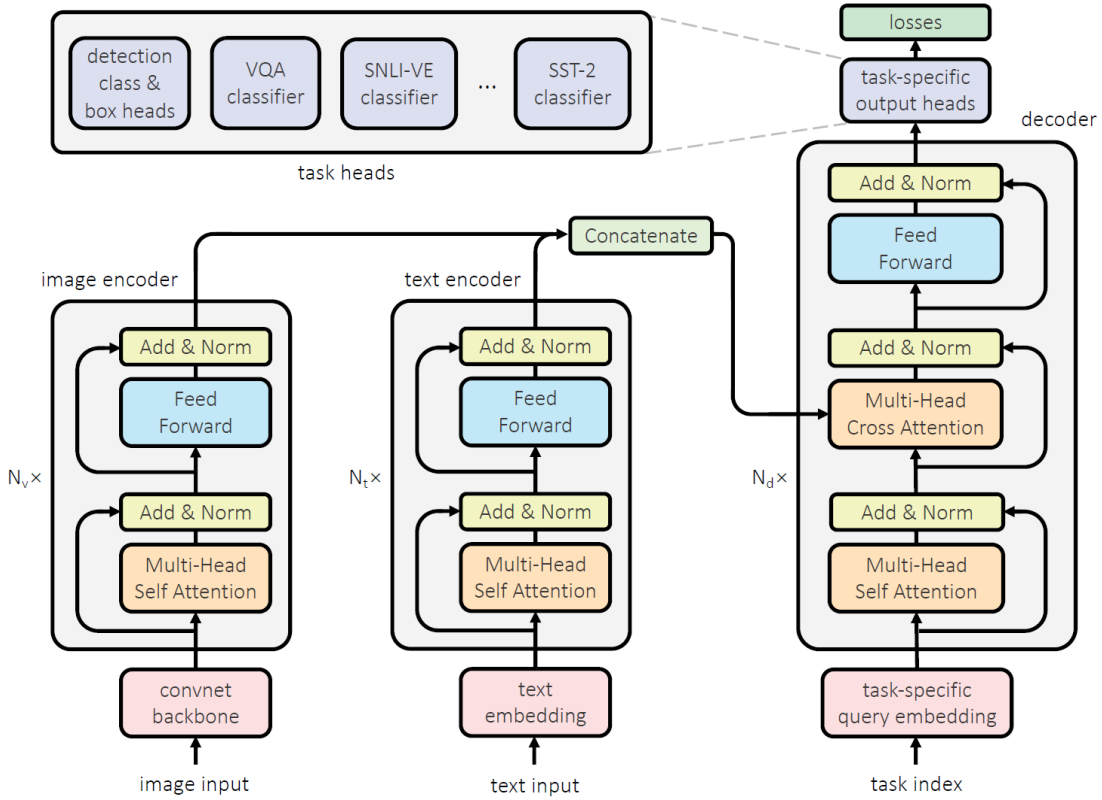
- Recent Breakthrough
  - Unified Foundation Model
  - VL Pre-training

# UniT (Facebook)

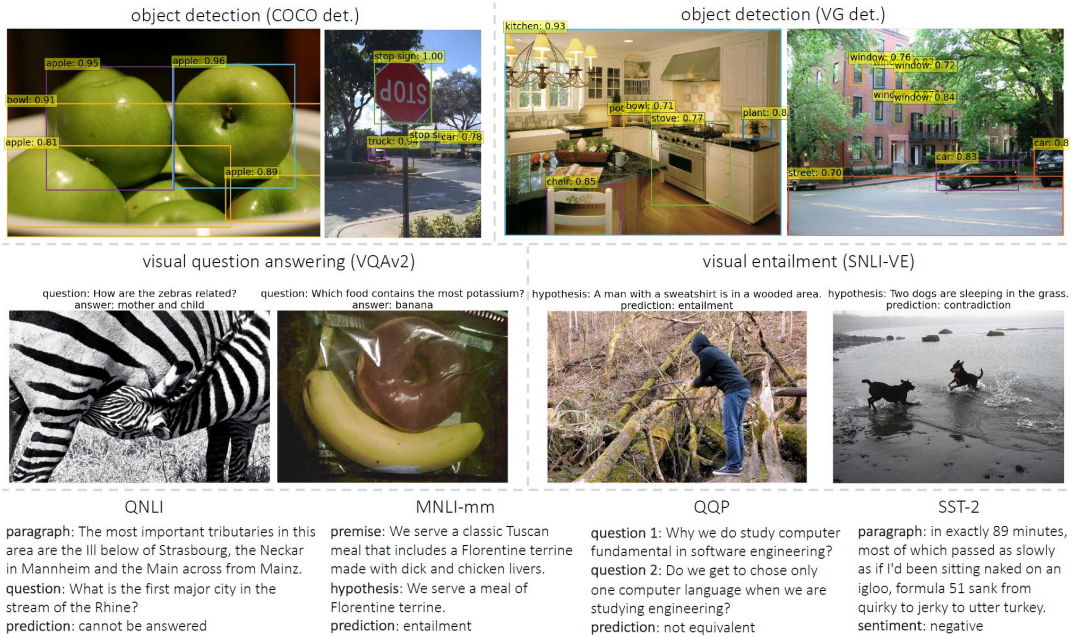



Unit: Multimodal multitask learning with a unified transformer, CVPR 2021

# UniT (Facebook)

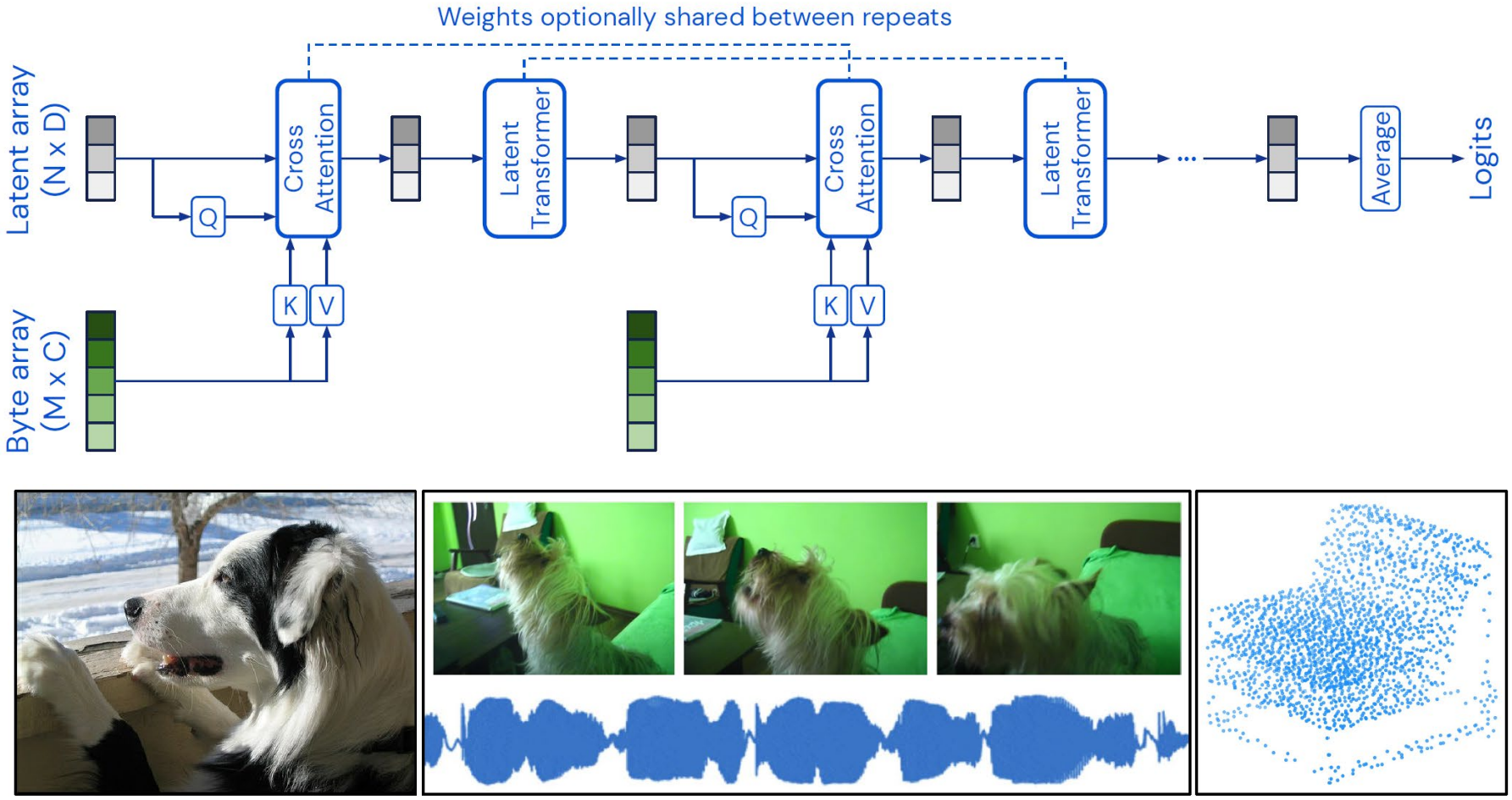



| # | decoder setup | COCO det. mAP | VG det. mAP | VQAv2 accuracy |
|---|---|---|---|---|
| 1 | single-task training | 40.6 / – | 3.87 | 66.38 / – |
| 2 | separate | **40.8** / – | 3.91 | **68.84** / – |
| 3 | shared | 37.2 / – | 4.05 | 68.79 / – |
| 4 | shared (COCO init.) | **40.8** / 41.1 | **4.53** | 67.30 / 67.47 |



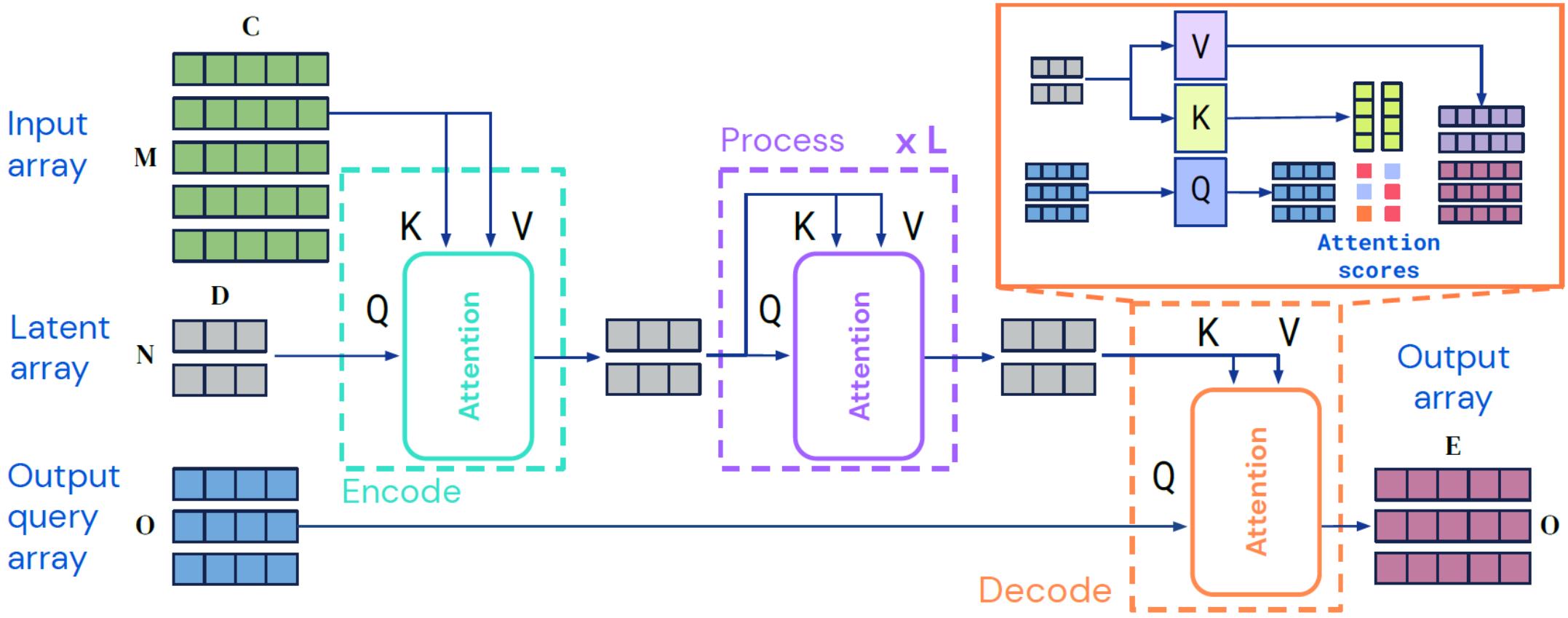

Unit: Multimodal multitask learning with a unified transformer, CVPR 2021

# Perceiver (Deepmind)




Perceiver: General Perception with Iterative Attention, Arxiv 2021

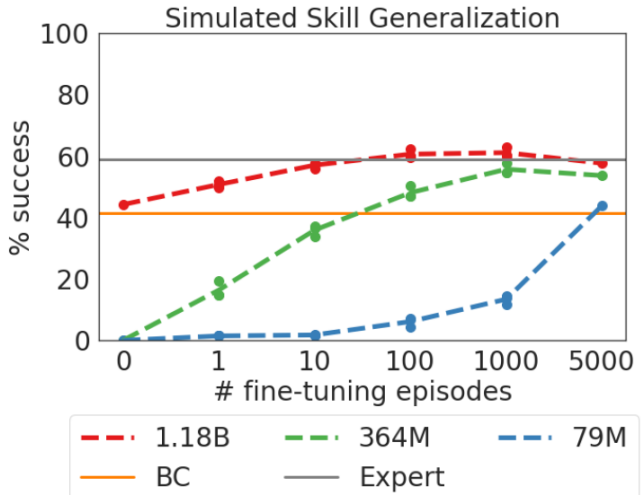
# Perceiver IO (Deepmind)

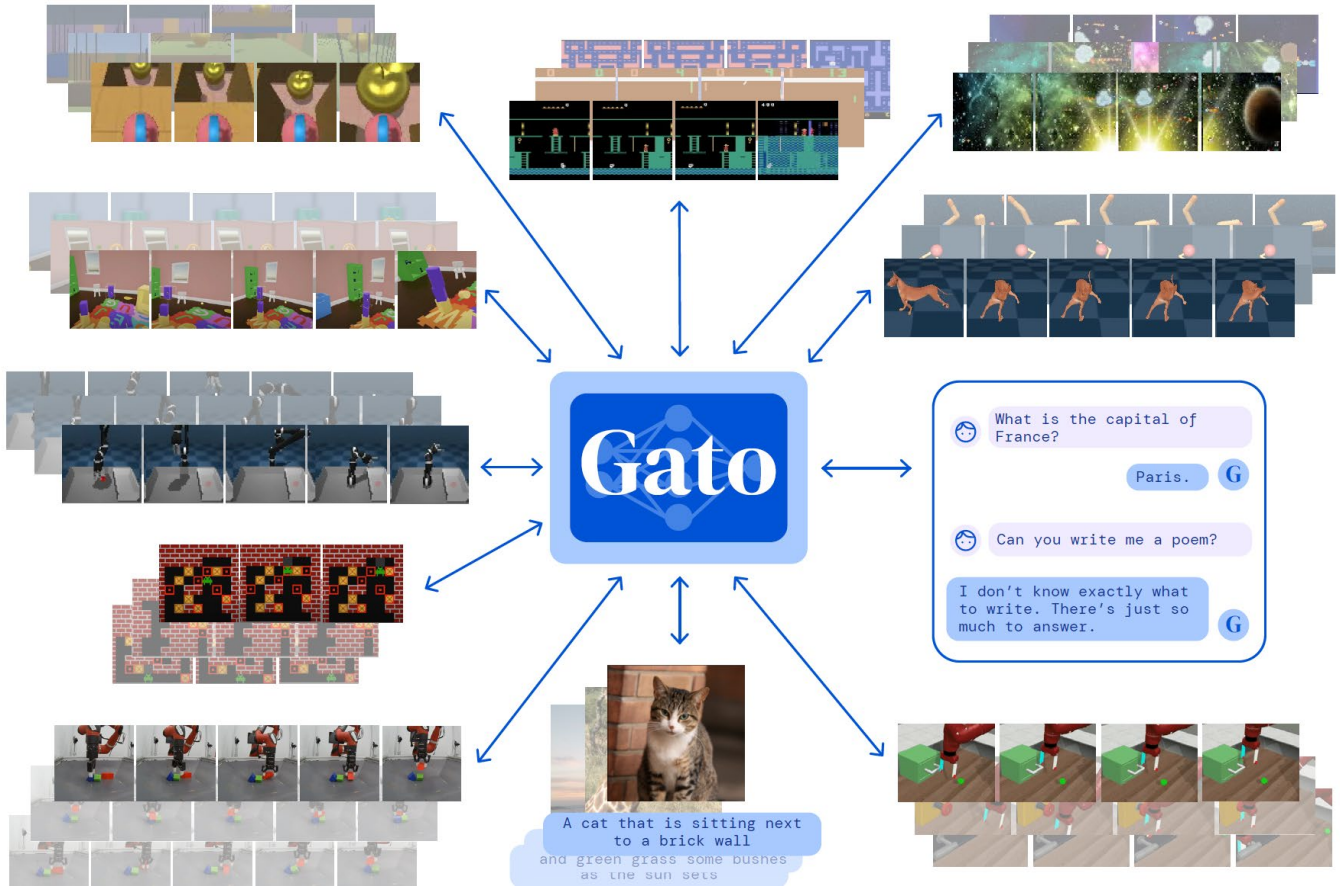



Perceiver IO: A General Architecture for Structured Inputs & Outputs. Arxiv 2021

# GATO (Deepmind)

- Multi-modal
- Multi-task
- Multi-embodiment

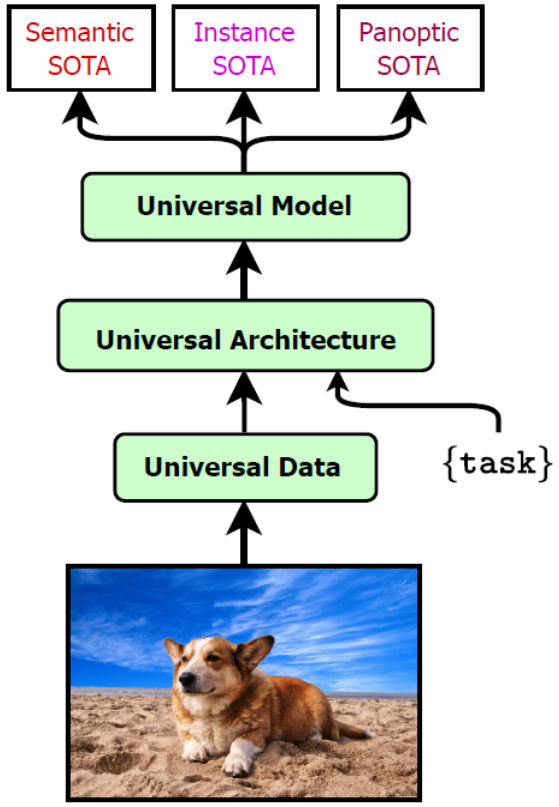


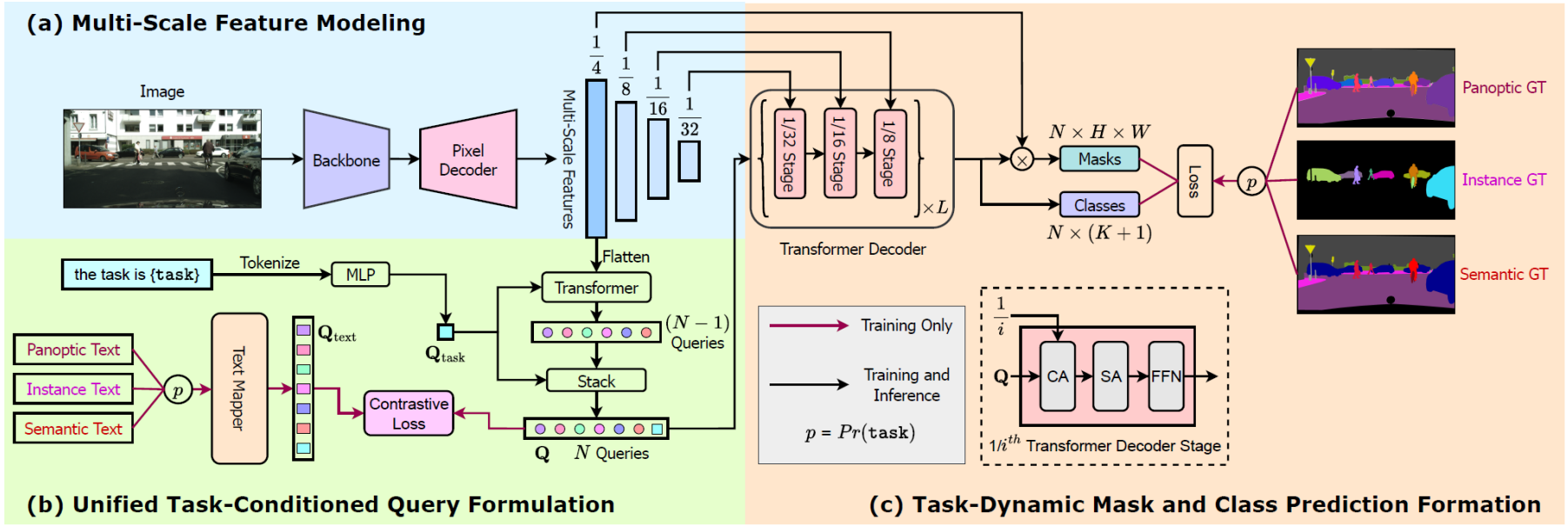




A generalist agent, Arxiv 2022.

# OneFormer (UIUC)

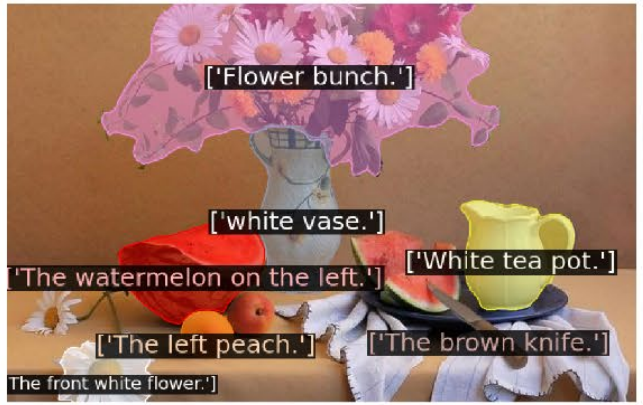
1 architecture, 1 model & 1 dataset




OneFormer: One Transformer to Rule Universal Image Segmentation, Arxiv 2022.

# X-Decoder (Microsoft)




Referring Segmentation

Open-Vocabulary Semantic Segmentation

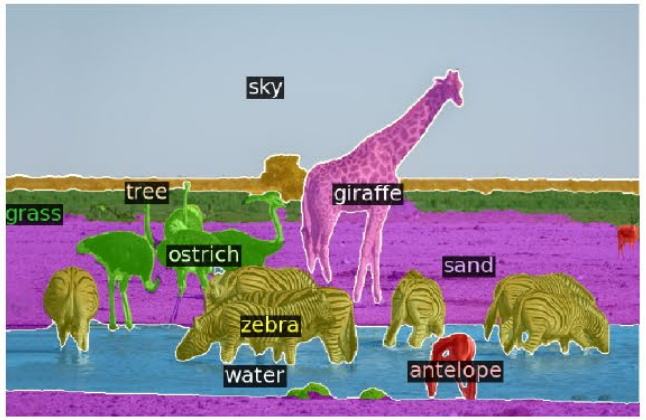




Image Captioning

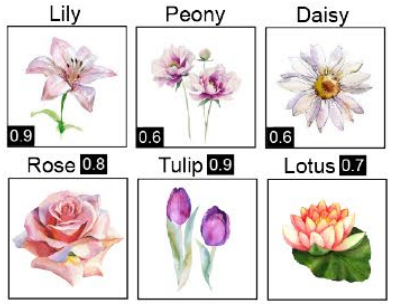



Image-Text Retrieval

X-Decoder

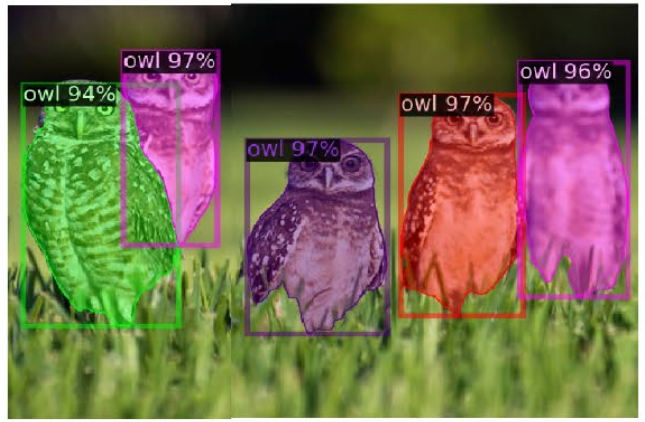
Referring Captioning


Referring Image Editing



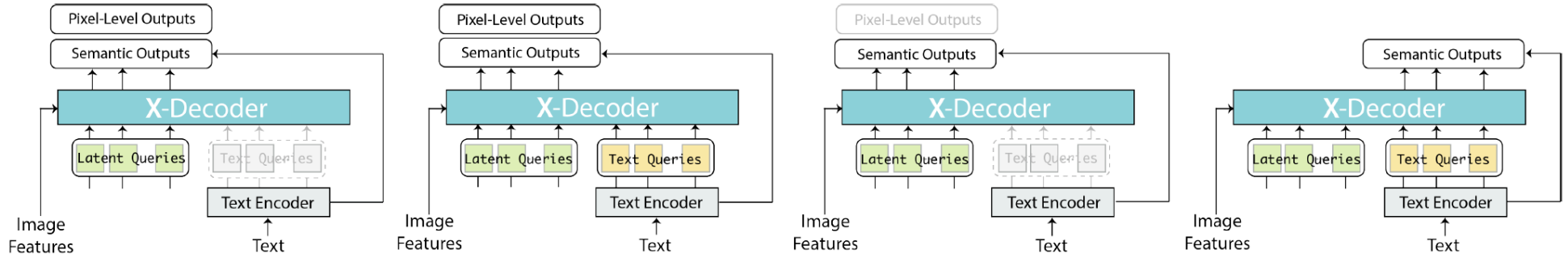
Open-Vocabulary Panoptic Segmentation

Open-Vocabulary Instance Segmentation




Generalized Decoding for Pixel, Image, and Language, Arxiv 2022

# X-Decoder (Microsoft)




| Method | Type | *Generic Segmentation* ADE PQ | ADE mAP | ADE mIoU | COCO PQ | COCO mAP | COCO mIoU | *Referring* g-Ref cIoU | *Retrieval* COCO-Karpathy IR@1 | COCO-Karpathy TR@1 | F30k-Karpathy IR@1 | F30k-Karpathy TR@1 | *Captioning* COCO-Karpathy CIDEr | COCO-Karpathy BLEU | *VQA* VQAv2-test dev | VQAv2-test std |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mask2Former (T) [12] | Segmentation | 39.7 | 26.4 | 47.7 | 53.2 | 43.3 | 63.2 | - | - | - | - | - | - | - | - | - |
| Mask2Former (B) [12] | | ⋆ | ⋆ | 53.9 | 56.4 | 46.3 | 67.1 | - | - | - | - | - | - | - | - | - |
| Mask2Former (L) [12] | | 48.1 | 34.2 | 56.1 | 57.8 | **48.6** | 67.4 | - | - | - | - | - | - | - | - | - |
| Pano/SegFormer (B) [47, 80] | | ⋆ | ⋆ | 51.0 | 55.4 | ⋆ | ⋆ | - | - | - | - | - | - | - | - | - |
| kMaX-DeepLab (L) [93] | | 48.7 | ⋆ | 54.8 | **58.1** | ⋆ | ⋆ | - | - | - | - | - | - | - | - | - |
| LAVT (B) [86] | | - | - | - | - | - | - | 61.2 | - | - | - | - | - | - | - | - |
| UNITER (B) [10] | Vision Language (VL) | - | - | - | - | - | - | - | 50.3 | 64.4 | 72.5 | 85.9 | - | - | 72.7 | 72.9 |
| UNITER (L) [10] | | - | - | - | - | - | - | - | 52.9 | 65.6 | 75.6 | 87.3 | - | - | 73.8 | 74.0 |
| VinVL (B) [96] | | - | - | - | - | - | - | - | 58.1 | 74.6 | ⋆ | ⋆ | 129.3 | 38.2 | 76.0 | 76.1 |
| VinVL (L) [96] | | - | - | - | - | - | - | - | **58.8** | 75.4 | ⋆ | ⋆ | 130.8 | 38.5 | 76.5 | 76.6 |
| ALBEF-4M (B) [43] | | - | - | - | - | - | - | - | 56.8 | 73.1 | 82.8 | 94.3 | ⋆ | ⋆ | 74.5 | 74.7 |
| METER-Swin (B) [21] | | - | - | - | - | - | - | - | 54.9 | 73.0 | 79.0 | 92.4 | ⋆ | ⋆ | 76.4 | 76.4 |
| UViM (L) [38] | General Purpose | ⋆ | ⋆ | ⋆ | 45.8 [1] | ⋆ | ⋆ | - | - | - | - | - | - | - | - | - |
| UniT (T) [32] | | - | - | - | - | - | - | - | - | - | - | - | - | - | 67.6 | ⋆ |
| GPV (T) [27] | | - | - | - | - | - | - | - | - | - | - | - | 102.3 [2] | ⋆ | 62.5 | ⋆ |
| UniTAB (B) [85] | | - | - | - | - | - | - | - | - | - | - | - | 119.8 | 36.1 | 70.7 | 71.0 |
| Pix2Seq v2 (B) [7] | | - | ⋆ | - | - | 38.2 | - | - | - | - | - | - | ⋆ | 34.9 | - | - |
| Unified-IO (B) [54] | | - | ⋆ | - | - | ⋆ | - | - | - | - | - | - | ⋆ | ⋆ | 61.8 | ⋆ |
| Unified-IO (L) [54] | | - | ⋆ | - | - | ⋆ | - | - | - | - | - | - | ⋆ | ⋆ | 67.8 | ⋆ |
| GLIPv2 (T) [95] | | - | ⋆ | - | - | -/42.0 | - | ⋆ | - | - | - | - | 122.1 | ⋆ | 71.6 | 71.8 |
| GLIPv2 (B) [95] | | - | ⋆ | - | - | -/45.8 | - | ⋆ | - | - | - | - | 128.5 | ⋆ | 73.1 | 73.3 |
| GLIPv2 (H) [95] | | - | ⋆ | - | - | -/48.9 | - | ⋆ | - | - | - | - | 131.0 | ⋆ | 74.6 | 74.8 |
| X-Decoder (T) | | 41.6 | 27.7 | 51.0 | 52.6 | 41.3/42.3 | 62.4 | 59.8 \| 61.9 | 49.3 | 66.7 | 74.4 | 89.1 | 122.3 | 37.8 | 70.6 | 70.9 |
| X-Decoder (B) | | 46.8 | 33.5 | 54.6 | 56.2 | 45.8/45.8 | 66.0 | 62.4 \| 64.5 | 54.5 | 71.2 | 80.8 | 93.2 | 129.0 | 39.6 | 74.1 | 74.2 |
| X-Decoder (L) | | **49.6** | **35.8** | **58.1** | 56.9 | 46.7/47.1 | **67.5** | **64.6** \| **64.6** | 58.6 | **76.1** | **84.4** | **94.4** | **132.1** | **40.2** | **76.8** | **77.0** |

Generalized Decoding for Pixel, Image, and Language, Arxiv 2022

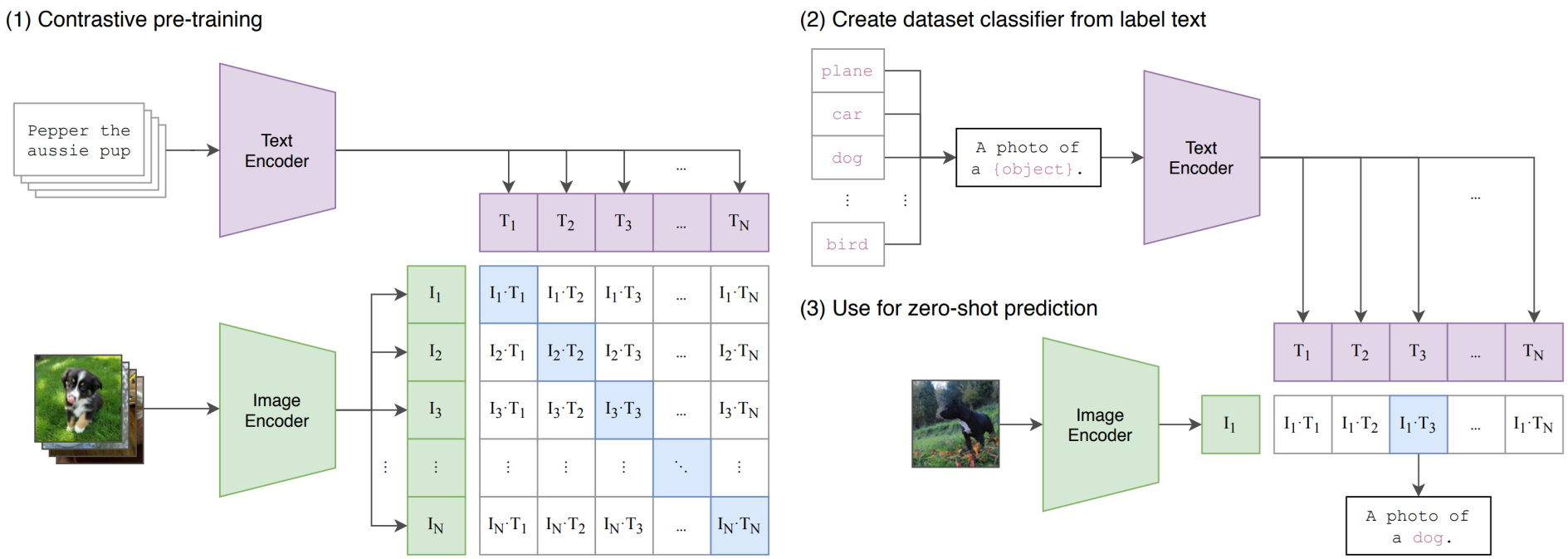
# Vision-Language Pretraining

- Early VL Pretraining

- Recent Breakthrough
  - Unified Foundation Model
  - VL Pre-training

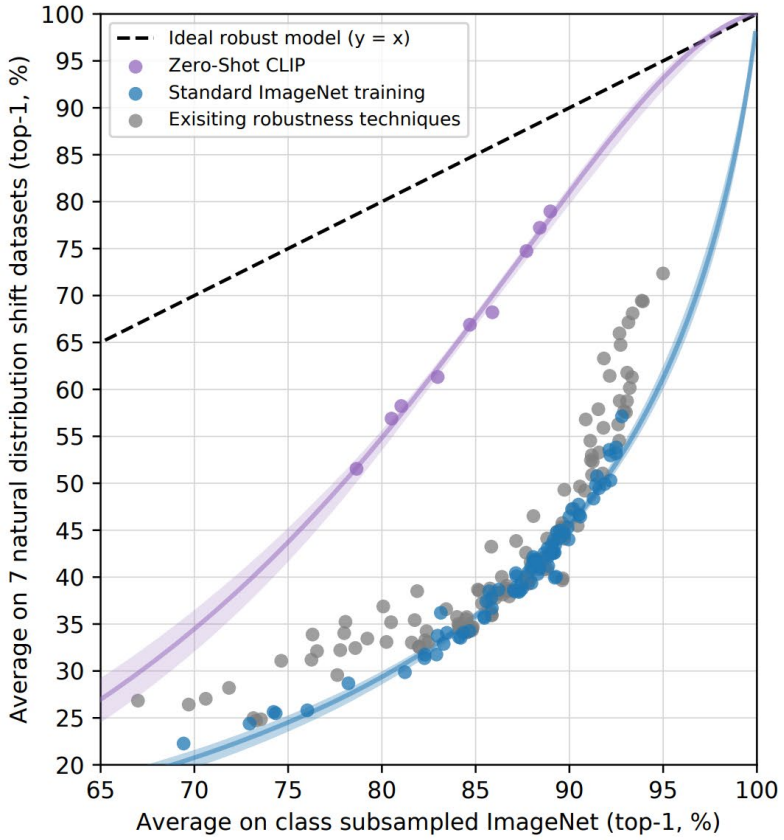
# VL Pretraining: CLIP (OpenAI)

- 400 million (image, text) pairs

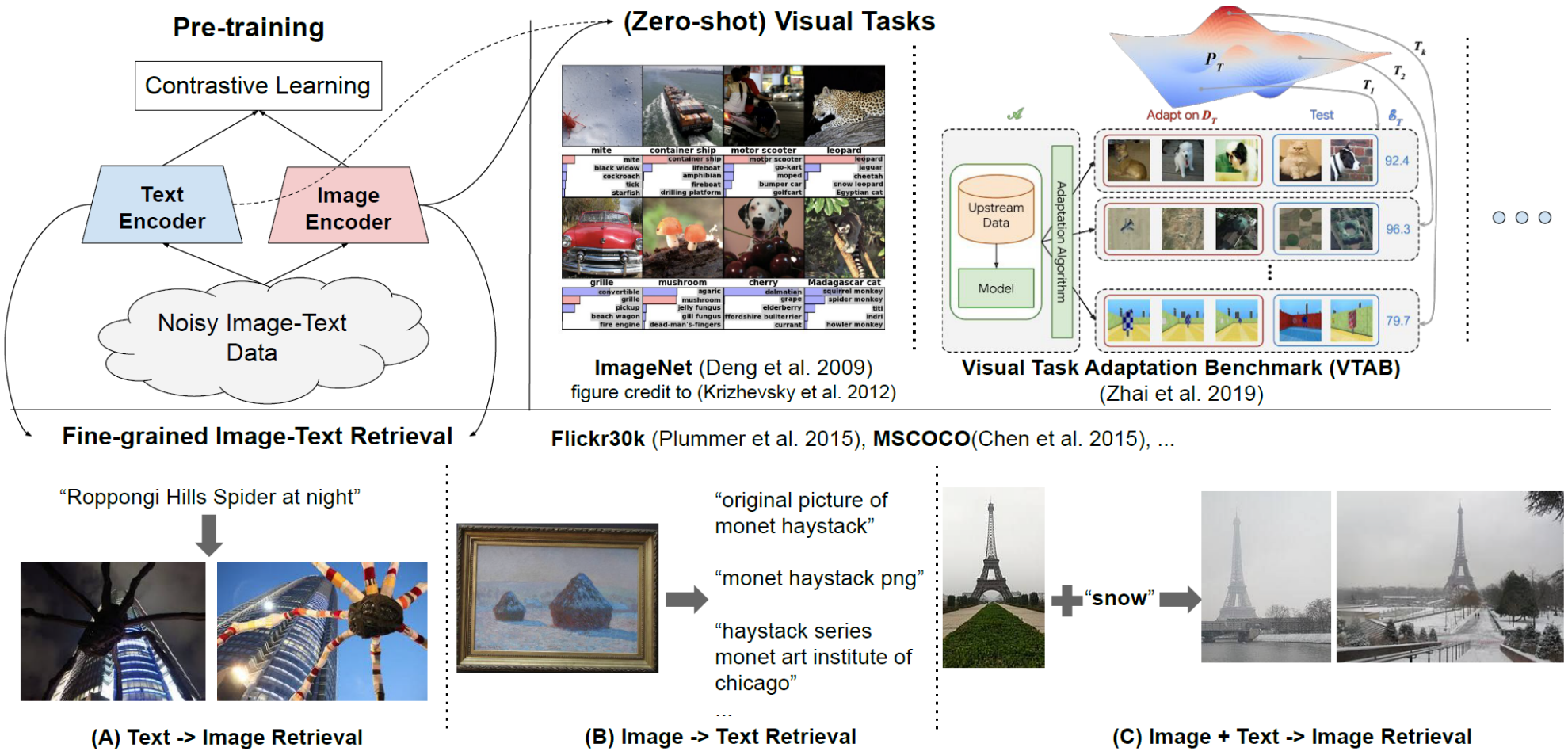



Learning Transferable Visual Models from Natural Language Supervision, ICML 2021

# Performance

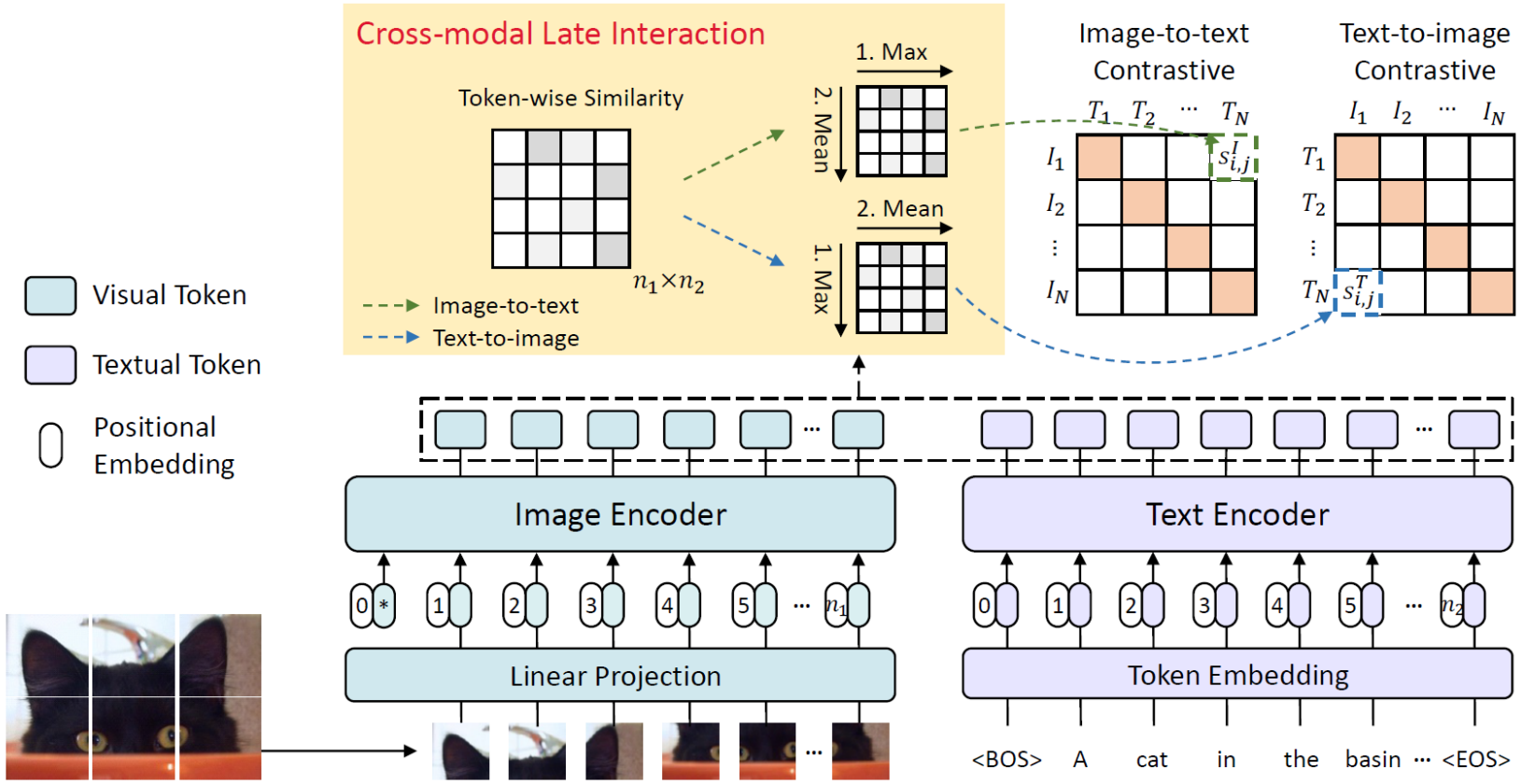



# VL Pretraining: ALIGN (Google)



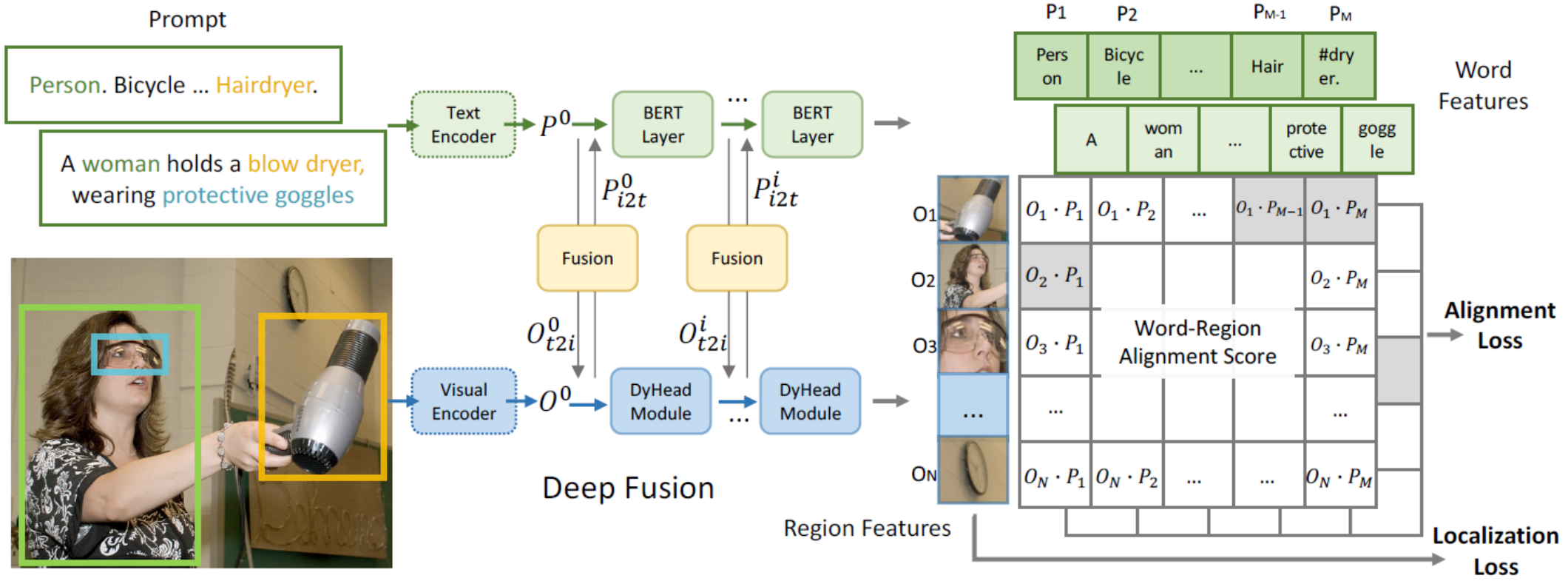

Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision, Arxiv 2021

# FILIP (Huawei)



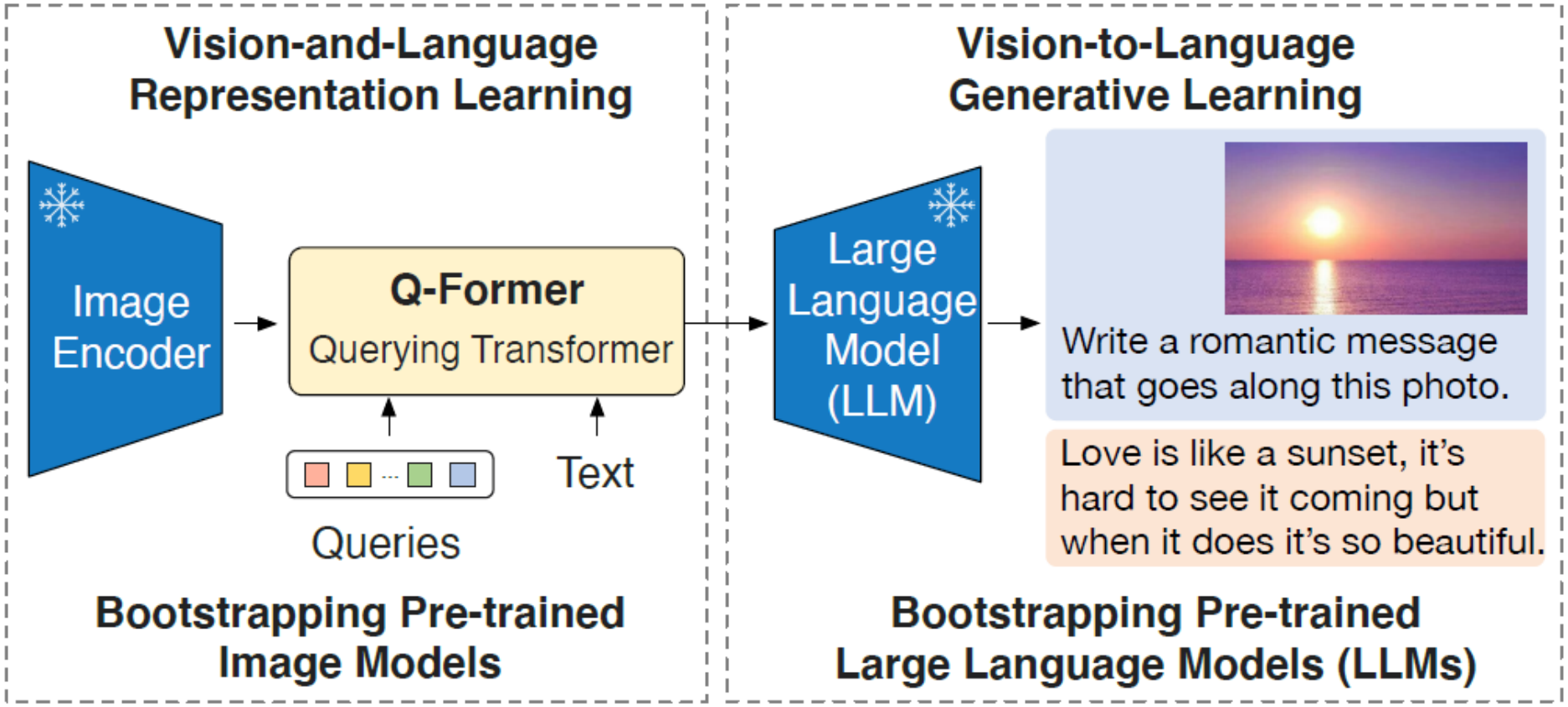

FILIP: fine-grained interactive language-image pre-training, ICLR 2022

# GLIP (Microsoft)




Grounded Language-Image Pre-training, CVPR 2022.

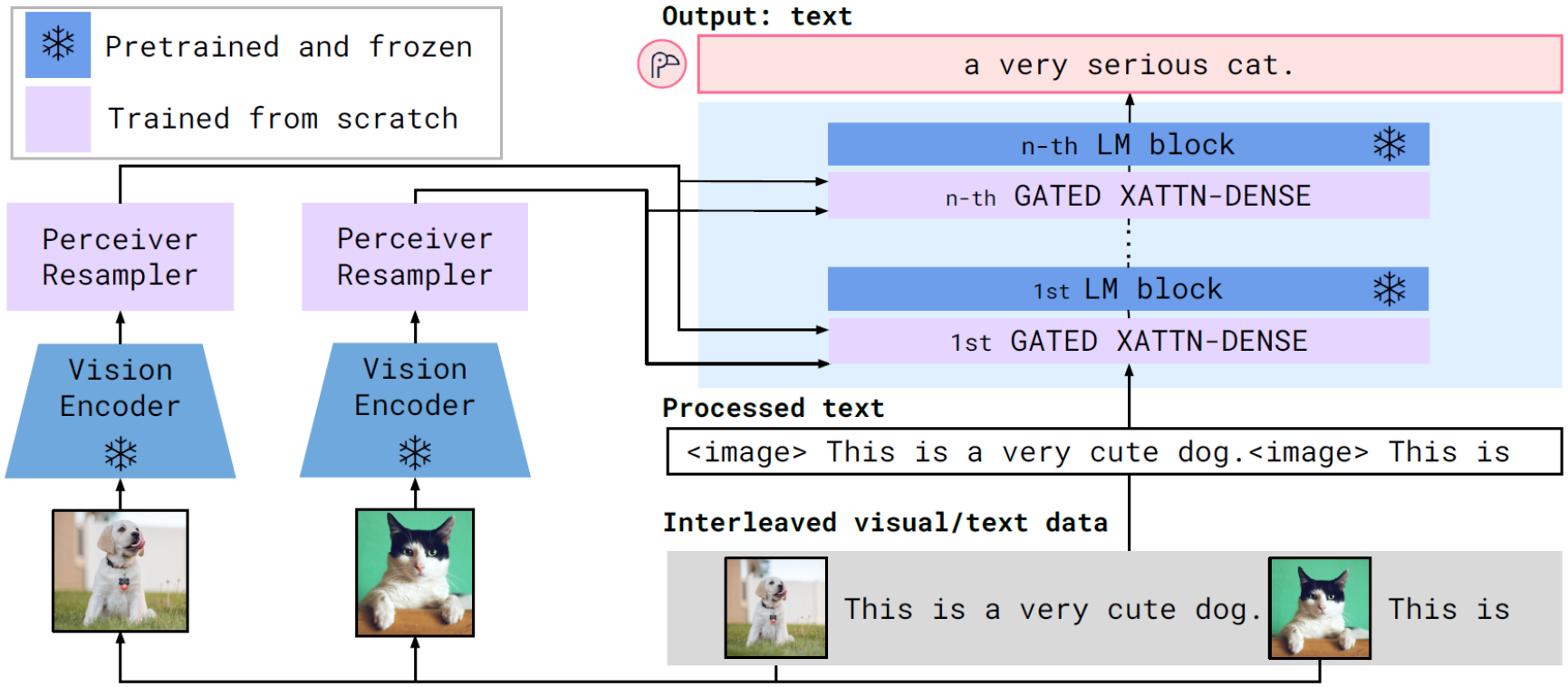
# BLIP-2 (SalesForce)




BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, Arxiv 2023.

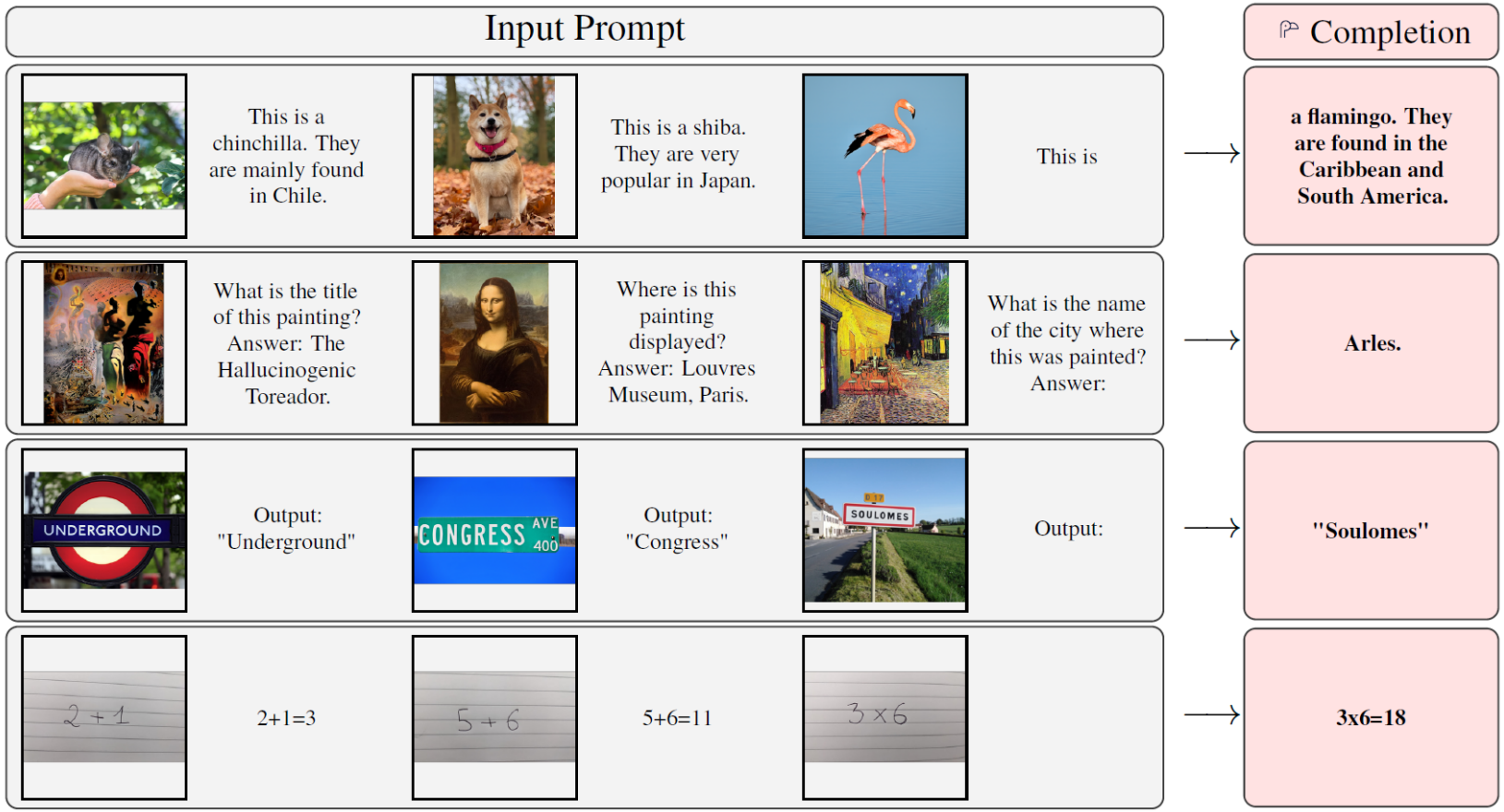
# Flamingo (Deepmind)

- In-context learning



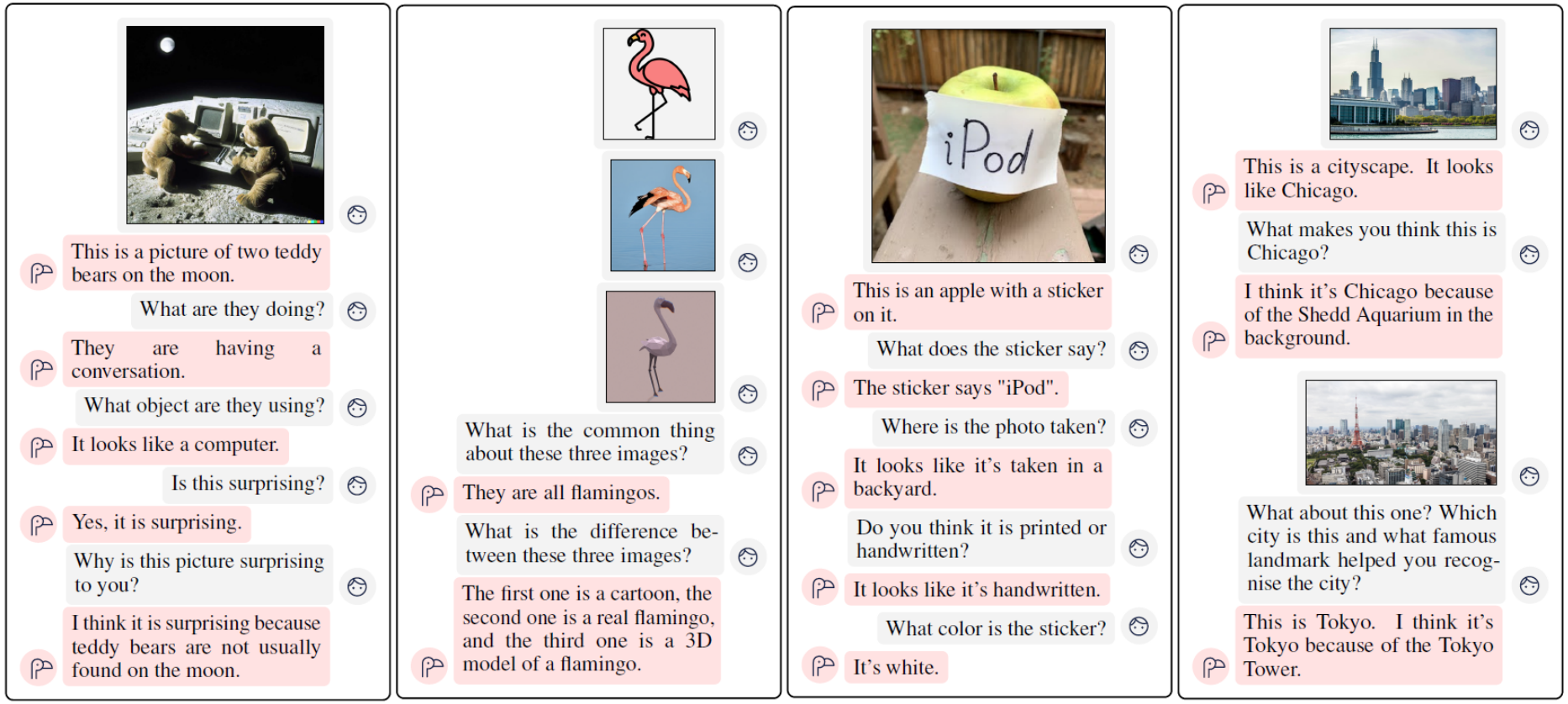

Flamingo - a Visual Language Model for Few-Shot Learning, NeurIPS 2022

# Flamingo (Deepmind)




Flamingo - a Visual Language Model for Few-Shot Learning, NeurIPS 2022

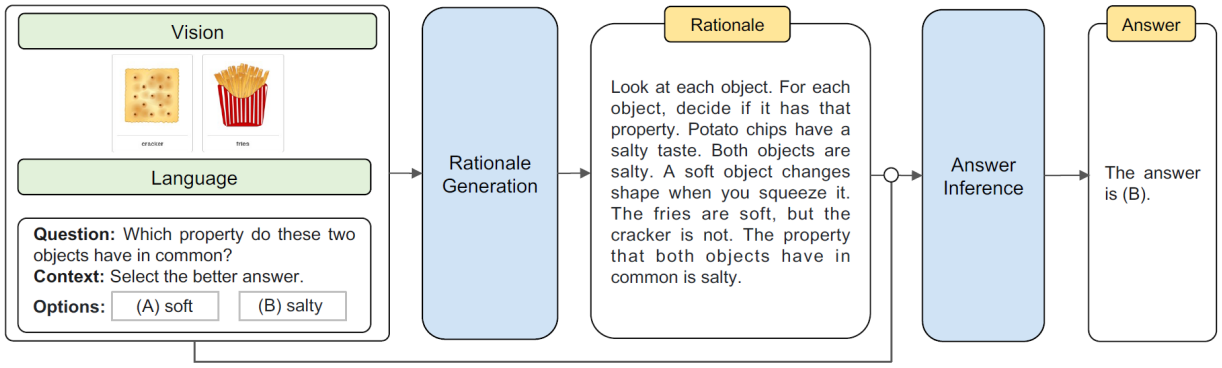
# Flamingo (Deepmind)




Flamingo - a Visual Language Model for Few-Shot Learning, NeurIPS 2022

# ChatGPT & GPT-4 (OpenAI)

- Chain of Thoughts
- RLHF
- In-context learning




Multimodal Chain-of-Thought Reasoning in Language Models, Arxiv 2023

**Example of GPT-4 visual input:**

User What is funny about this image? Describe it panel by panel.

Source: https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

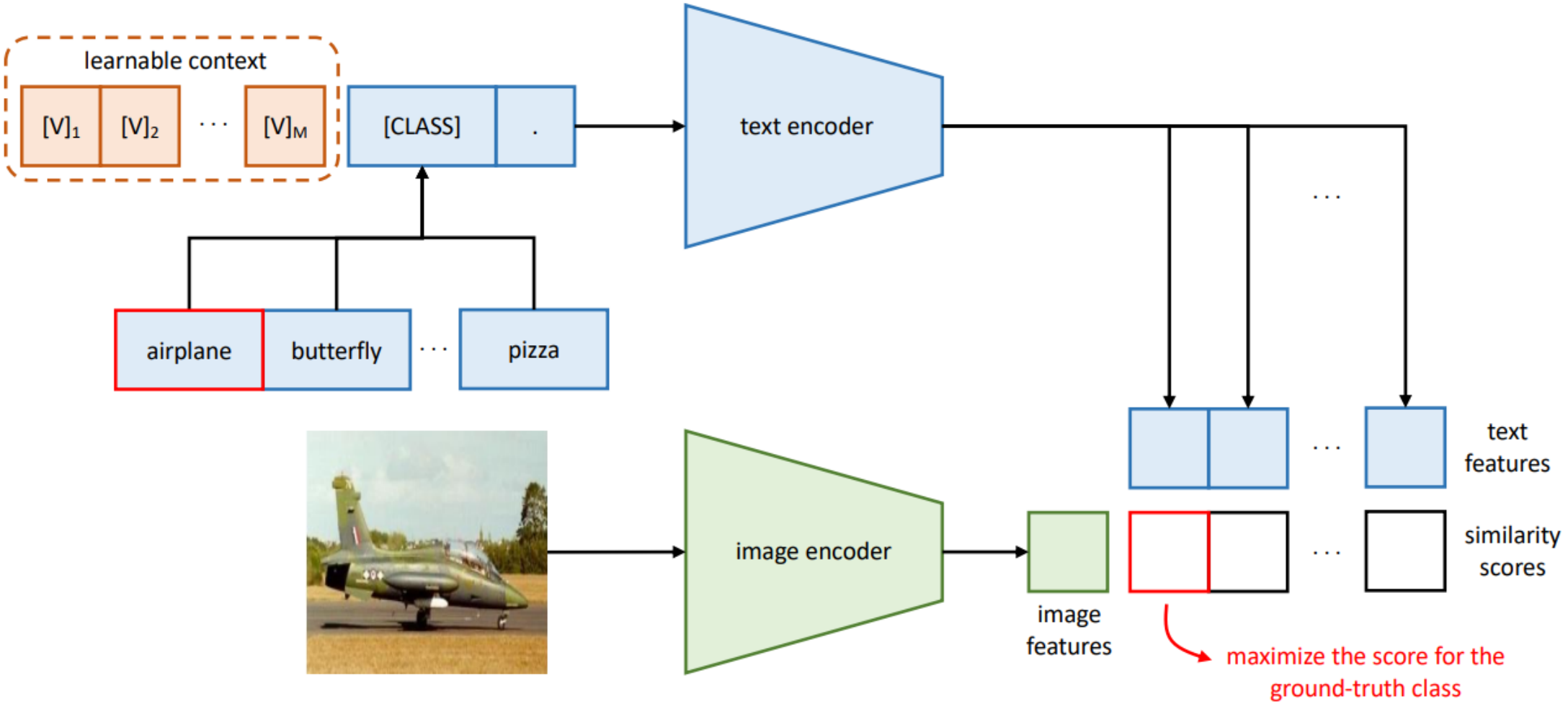
# Vision-Language Pretraining

- Vision-Language Models
- Classification: Parameter-efficient Fine-tuning
- More Challenging Tasks
- More Visual Modalities

# Contents

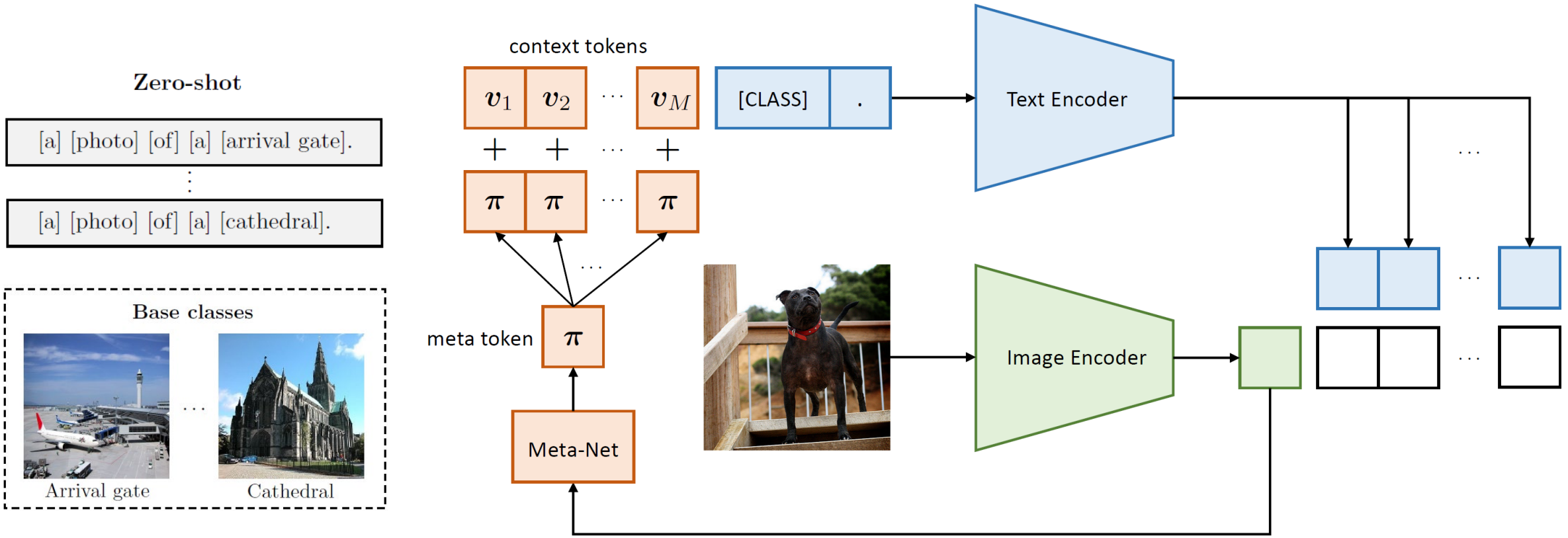
- Prompt
- Adapter
- Controllable FineTuning

# CoOp



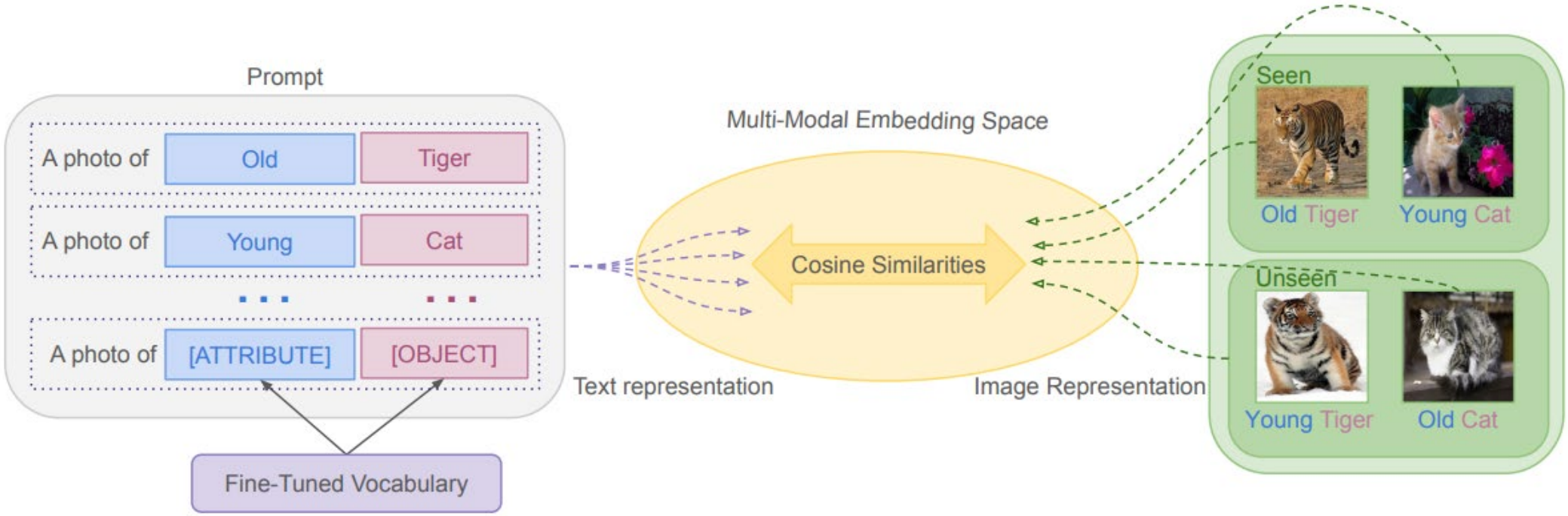

Learning to Prompt for Vision-Language Models, IJCV 2022

# CoCoOp: Conditional Prompt Learning

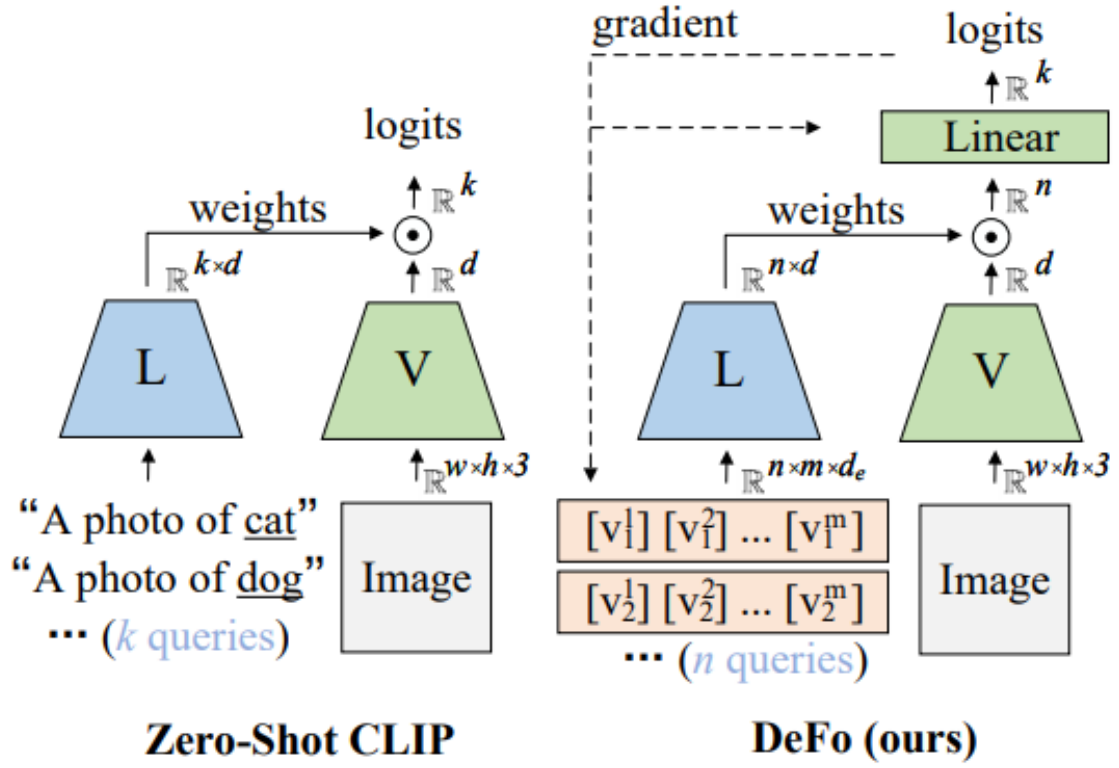



Conditional Prompt Learning for Vision-Language Models, Arxiv 2022

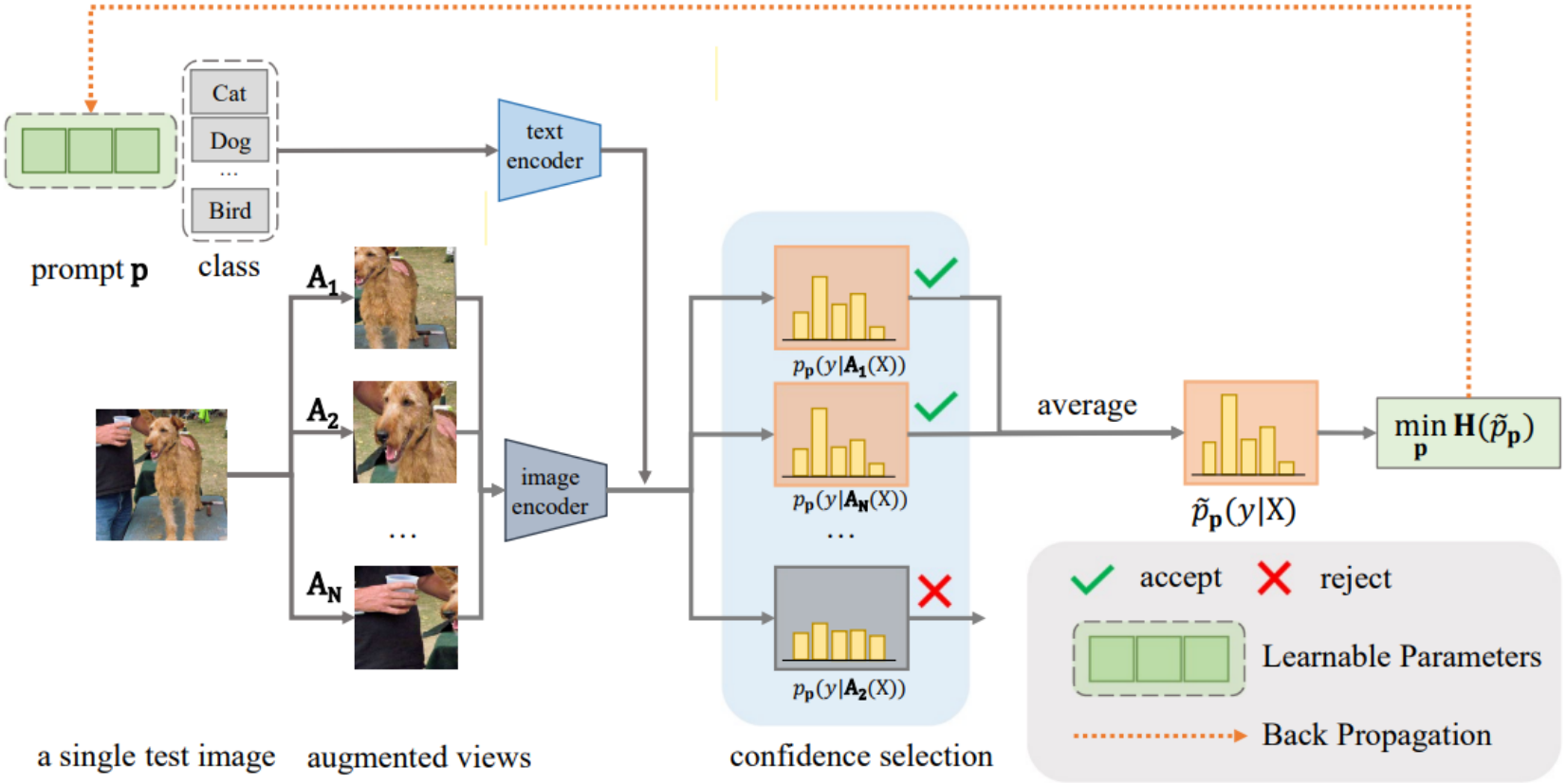
# Learning to Compose Soft Prompts for Compositional Zero-Shot Learning (ICLR2023)



# DeFo (ICLR2023)

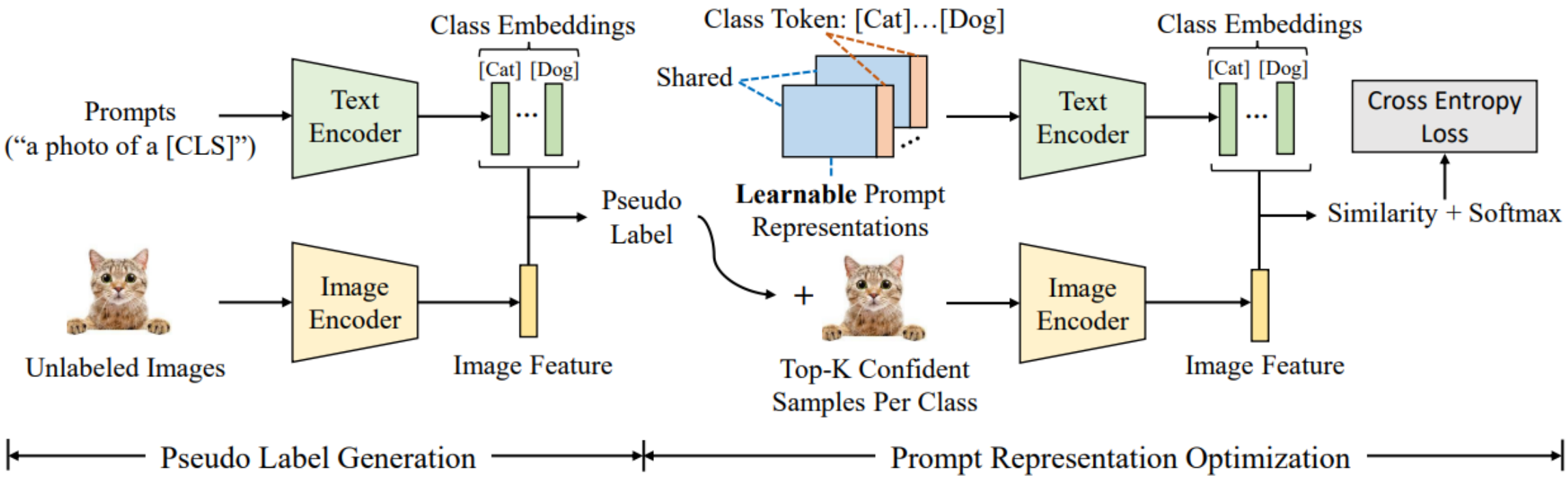


# Test-time Prompt Tuning (NIPS2022)




测试图像也是一种unsupervised setting

# Unsupervised Prompt Learning




模型确信的往往分类正确，迭代self-training有提升

# DualCoOp: Adapting CLIP to Multi-Label

- Previous single-label classification

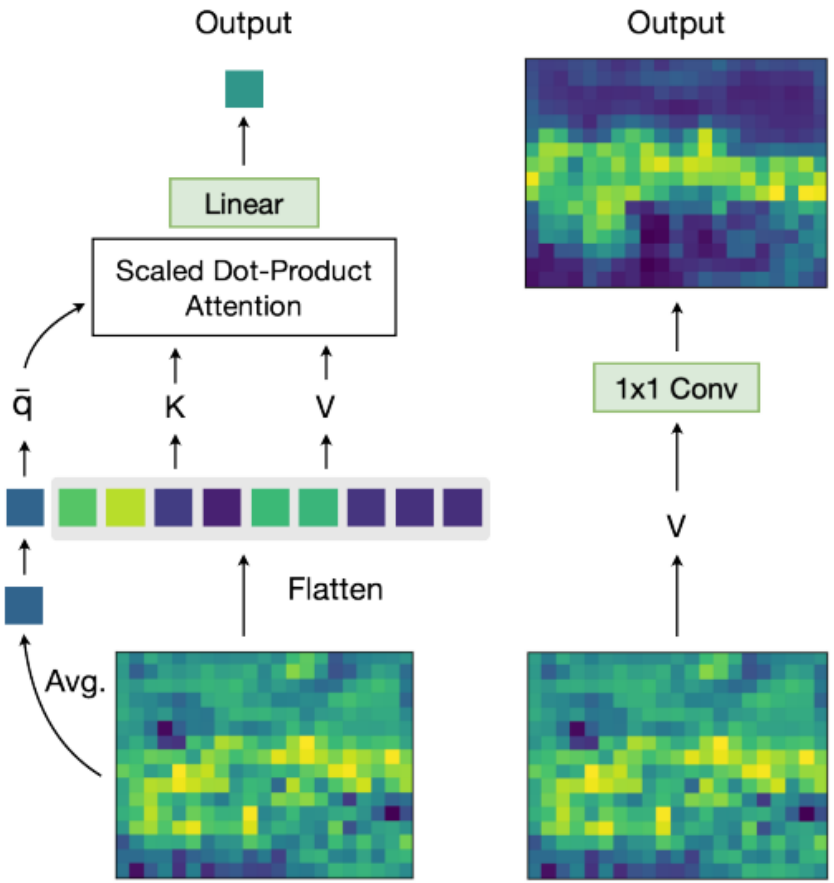
$$p(y=i|\boldsymbol{x}) = \frac{\exp(<\boldsymbol{w_i}, \boldsymbol{f}>/\tau)}{\sum_{j=1}^{K}\exp(<\boldsymbol{w_j}, \boldsymbol{f}>/\tau)},$$

- Multi-label binary classification

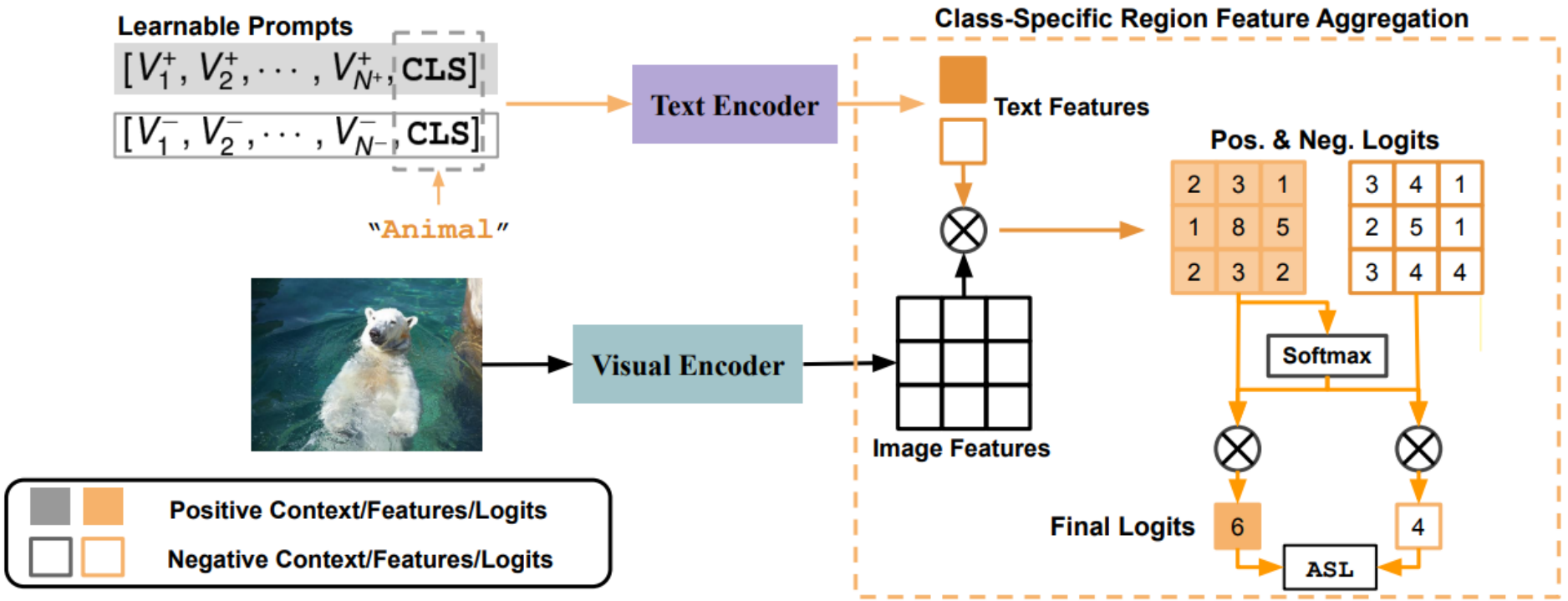
$$p(y=i \mid x) = \frac{\exp(<w_i^+, f>/\tau)}{\exp(<w_i^+, f>/\tau) + \exp(<w_i^+, f>/\tau)}$$

# DualCoOp: Adapting CLIP to Multi-Label

- Preserve localized features



# DualCoOp: Adapting CLIP to Multi-Label



# Texts as Images

Train

$V_1, V_2, V_3, ..., V_M, [dog]$

Test

A photo of a dog.
A dog sitting on grass.
…

Train

$V_1, V_2, V_3, ..., V_M, [dog]$

Test

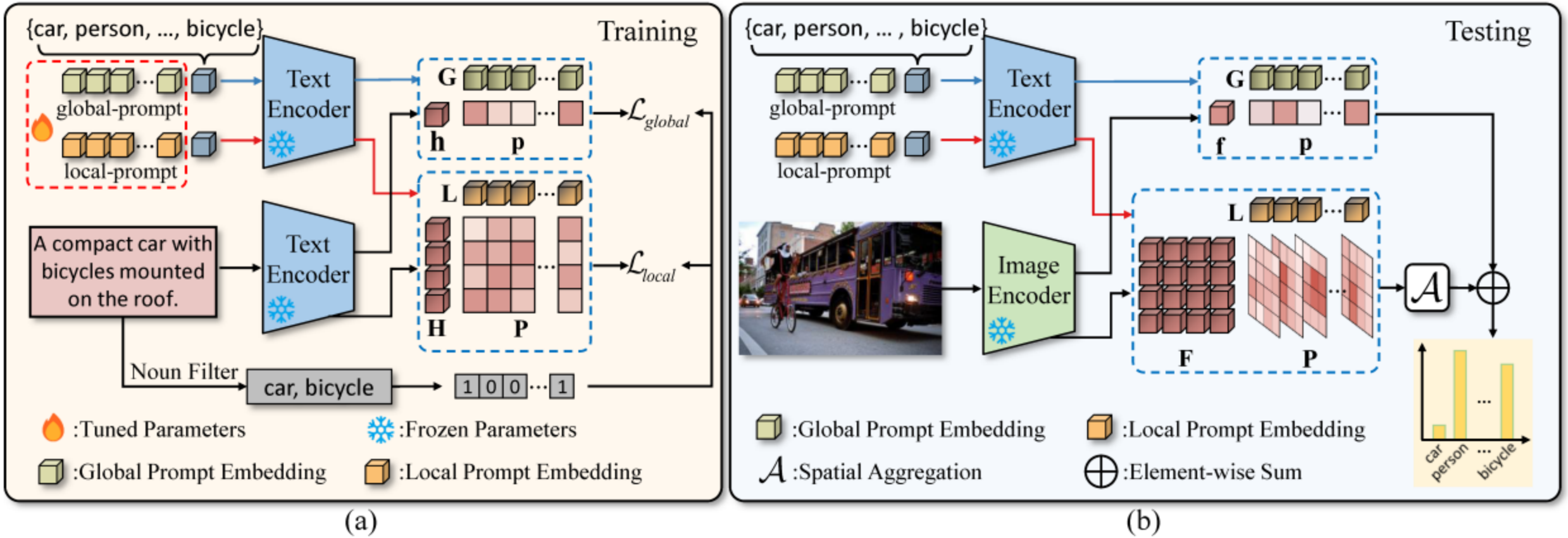
Woman on a horse jumping over a pole jump.

Train

$V_1, V_2, V_3, ..., V_M, [person]$
$V_1, V_2, V_3, ..., V_M, [horse]$

Test

# TaI-DPT


{car, person, ..., bicycle}
Training
global-prompt
local-prompt
Text Encoder
G
h
p
$\mathcal{L}_{global}$
L
H
P
$\mathcal{L}_{local}$
A compact car with bicycles mounted on the roof.
Text Encoder
Noun Filter
car, bicycle
1 0 0 ··· 1
:Tuned Parameters
:Frozen Parameters
:Global Prompt Embedding
:Local Prompt Embedding
(a)
{car, person, ... , bicycle}
Testing
global-prompt
local-prompt
Text Encoder
G
f
p
L
F
P
Image Encoder
$\mathcal{A}$
car
person
bicycle
:Global Prompt Embedding
:Local Prompt Embedding
$\mathcal{A}$ :Spatial Aggregation
:Element-wise Sum
(b)

# Loss

- Ranking Loss

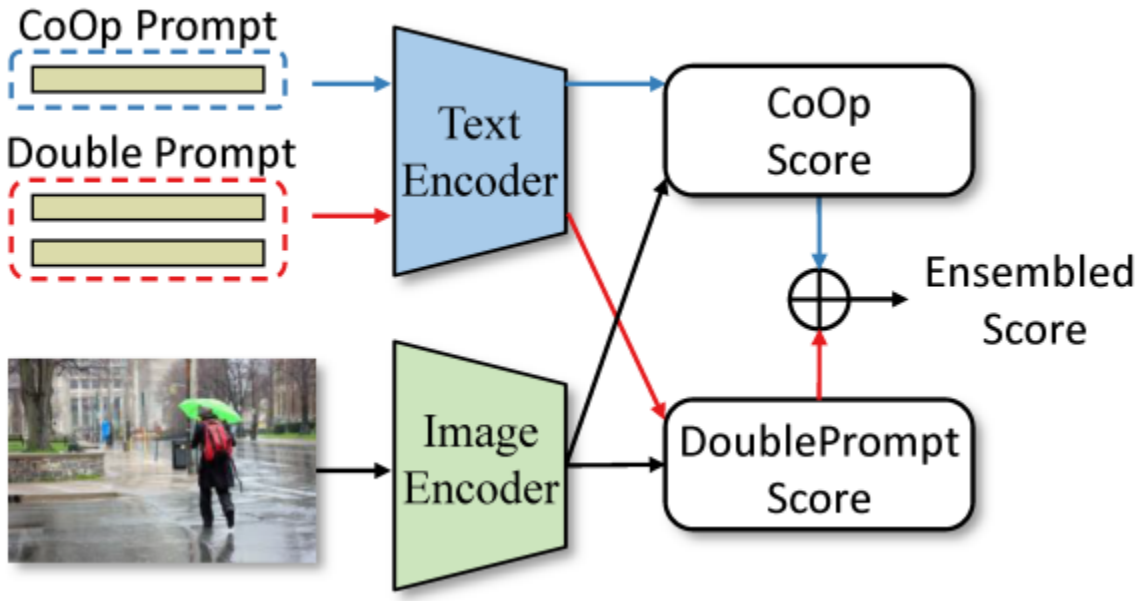
$$\mathcal{L}_{global} = \sum_{i \in \{c^+\}} \sum_{j \in \{c^-\}} \max(0, m - \boldsymbol{p}_i + \boldsymbol{p}_j),$$
$$\mathcal{L}_{local} = \sum_{i \in \{c^+\}} \sum_{j \in \{c^-\}} \max(0, m - \boldsymbol{p}'_i + \boldsymbol{p}'_j)$$

- Binary Cross Entropy Loss

$$\mathcal{L} = \mathrm{BCE}(\boldsymbol{p}, \boldsymbol{y}) + \mathrm{BCE}(\boldsymbol{p}', \boldsymbol{y}),$$
$$\mathrm{BCE}(\boldsymbol{q}, \boldsymbol{y}) = -\frac{1}{C} \sum_{i=1}^{C} [\boldsymbol{y}_i \cdot \log \boldsymbol{q}_i + (1 - \boldsymbol{y}_i) \cdot \log (1 - \boldsymbol{q}_i)]$$

| Loss | VOC2007 | MS-COCO | NUSWIDE |
|---|---|---|---|
| BCE | 84.9 | 59.0 | 40.5 |
| ASL [1] | 84.6 | 56.9 | 36.0 |
| RL [6] | **88.3** | **65.1** | **46.5** |

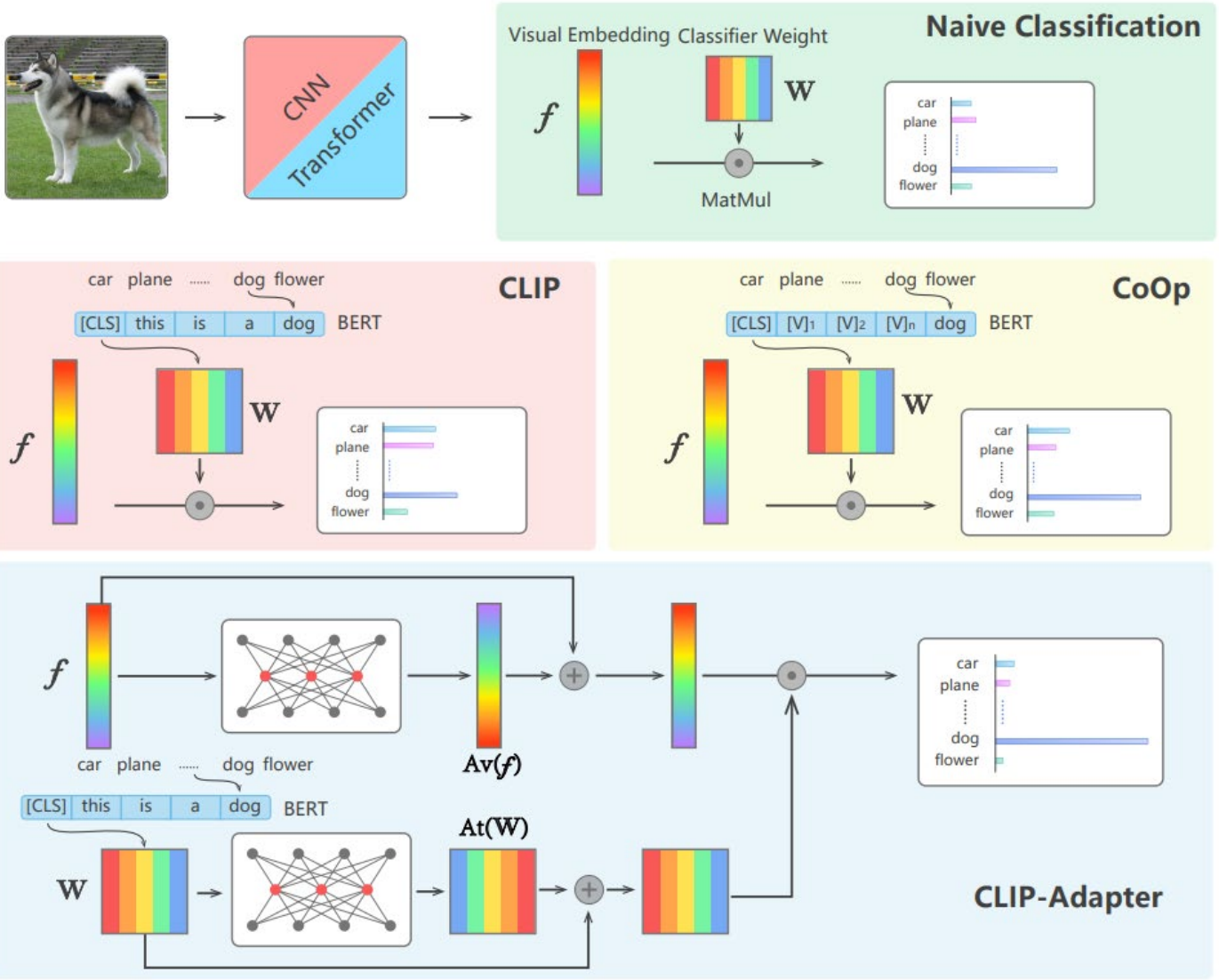
# Integration with Other Methods



# Contents

- Prompt
- Adapter
- Controllable FineTuning

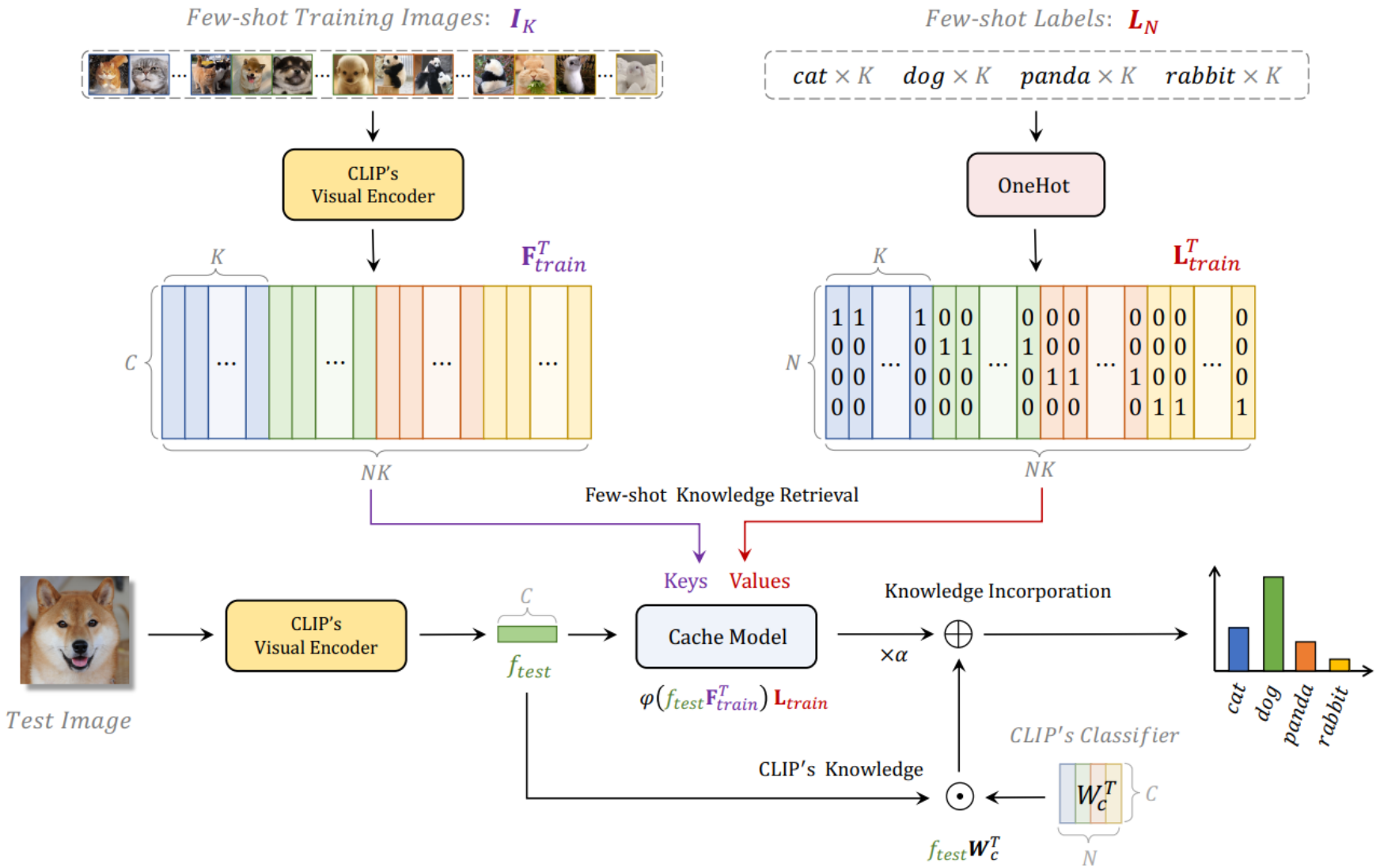
# CLIP-Adapter




$$A_v(f) = \mathrm{ReLU}(f^T \mathbf{W}_1^v)\mathbf{W}_2^v,$$
$$A_t(\mathbf{W}) = \mathrm{ReLU}(\mathbf{W}^T \mathbf{W}_1^t)\mathbf{W}_2^t.$$

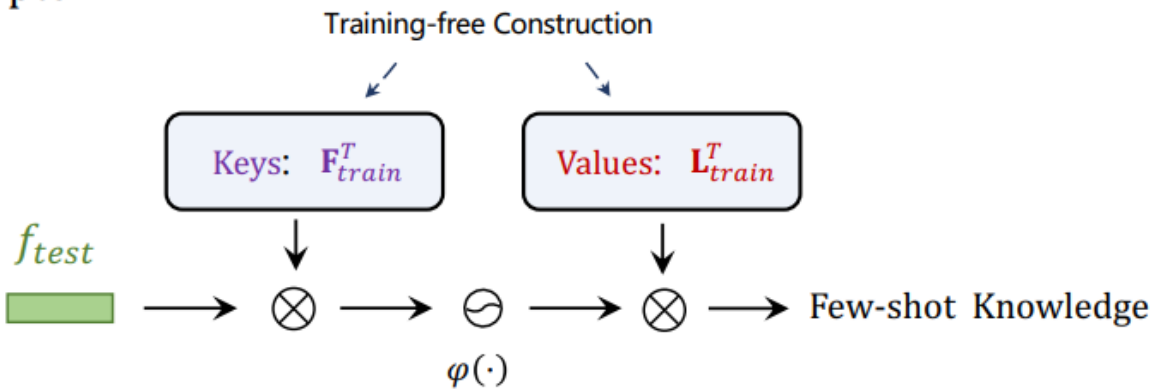
$$f^\star = \alpha A_v(f)^T + (1-\alpha) f,$$
$$\mathbf{W}^\star = \beta A_t(\mathbf{W})^T + (1-\beta)\mathbf{W}.$$

# Tip-Adapter

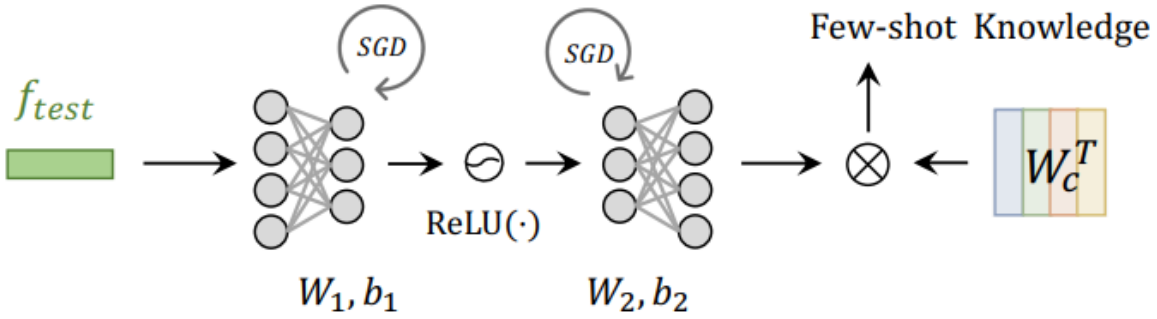

Few-shot Training Images: $I_K$
Few-shot Labels: $L_N$
cat × K
dog × K
panda × K
rabbit × K
CLIP's Visual Encoder
OneHot
K
$F^T_{train}$
$L^T_{train}$
C
N
NK
Few-shot Knowledge Retrieval
Keys
Values
Test Image
CLIP's Visual Encoder
C
$f_{test}$
Cache Model
$\varphi(f_{test}F^T_{train})L_{train}$
×α
Knowledge Incorporation
cat
dog
panda
rabbit
CLIP's Classifier
CLIP's Knowledge
$W^T_c$
$f_{test}W^T_c$

# Relations with CLIP-Adapter

Tip-Adapter:



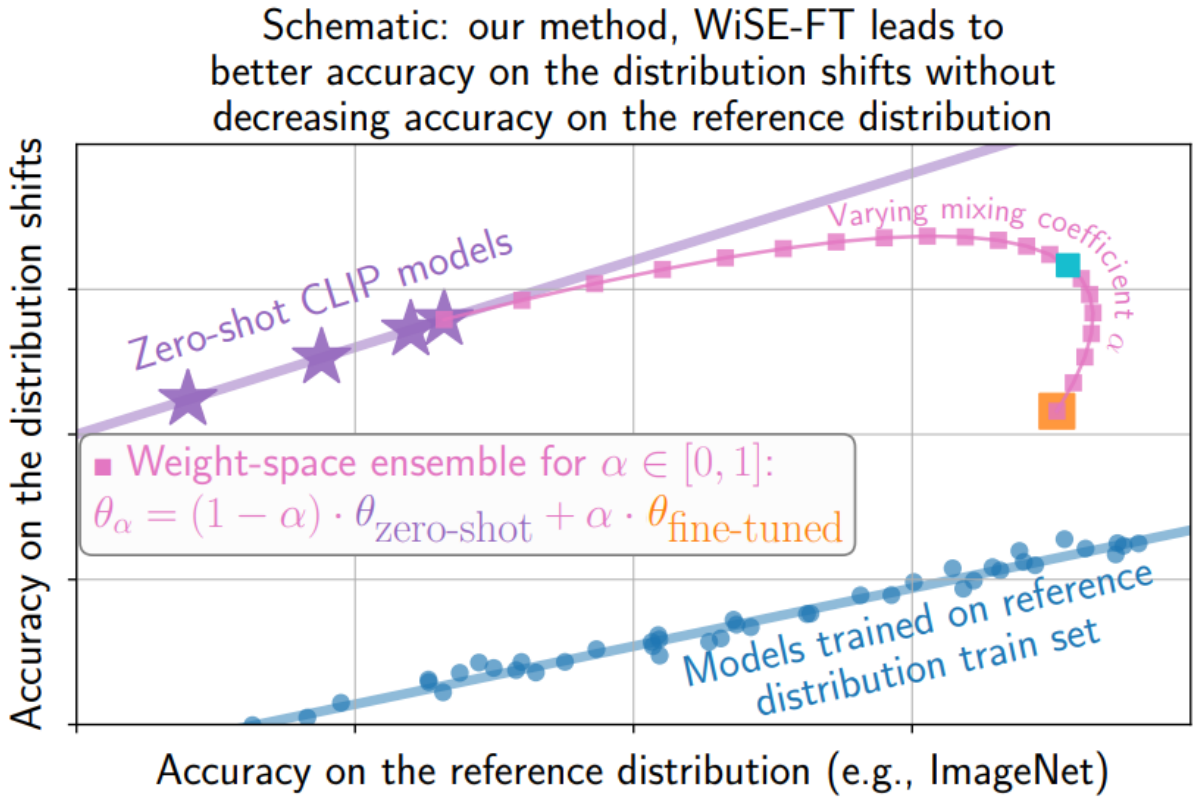

CLIP-Adapter:



# Contents

- Prompt
- Adapter
- Controllable FineTuning

# Robust fine-tuning of zero-shot models (CVPR2022)




Finetune后将模型参数插值

# Full ImageNet

| Method | RN-50 | RN-101 | ViT-B/32 | ViT-B/16 |
|---|---|---|---|---|
| Zero-Shot CLIP (Radford et al., 2021) | 58.2 | 61.5 | 62.0 | 66.9 |
| Linear-Probing CLIP | 72.8 | 75.5 | 76.0 | 79.5 |
| Prompt Ensembling | 60.4† | 62.5† | 63.7† | 68.7† |
| CoOp (Zhou et al., 2021) | 65.6 | 67.8 | 68.0 | 72.4 |
| Target Optimization | 71.4 | 73.2 | 74.0 | 78.1 |
| CLIP-Adapter (Gao et al., 2021) | 61.3‡ | - | - | - |
| DeFo (ours) | **73.2** | **75.5** | **76.2** | **80.2** |

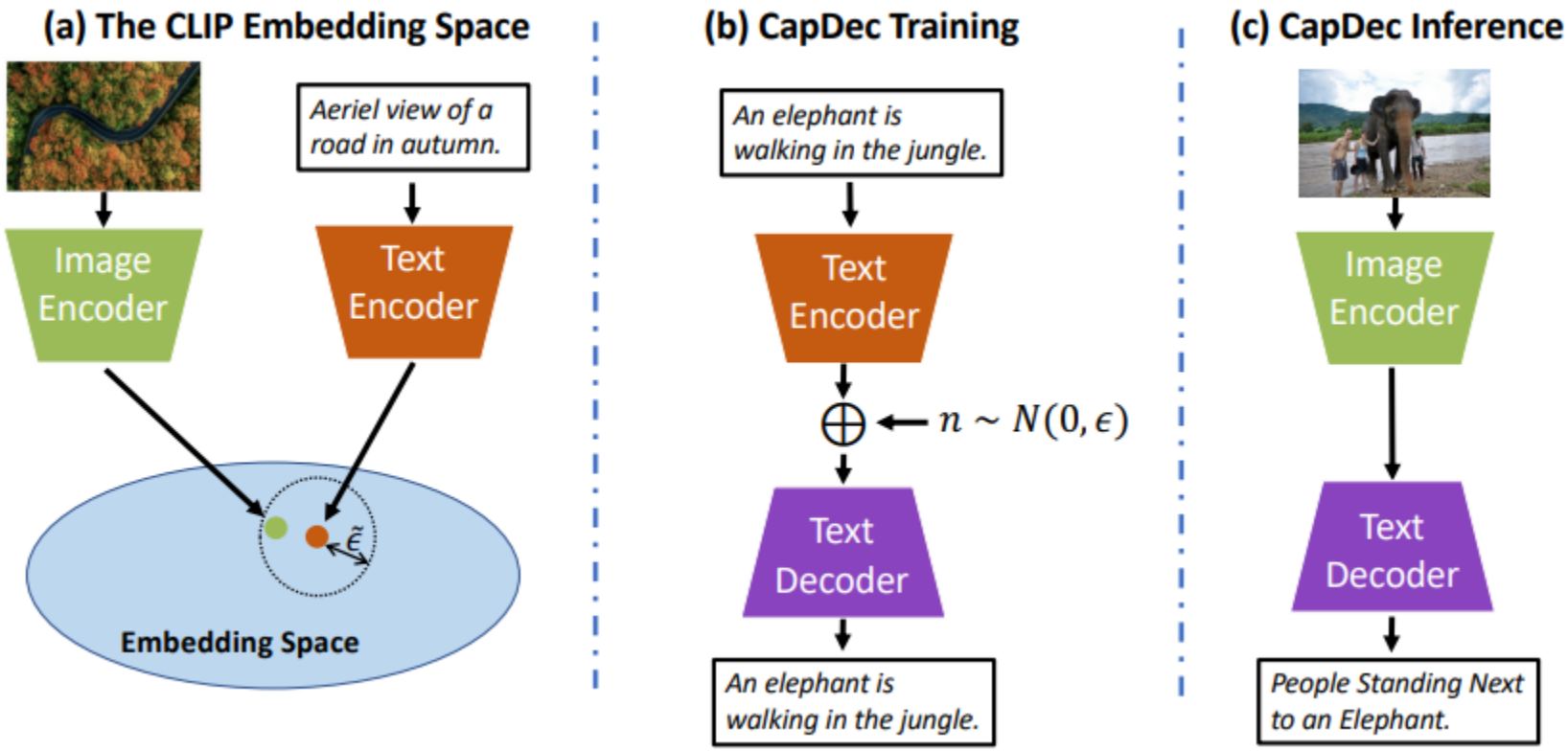
| | |
|---|---|
| Wise-FT (LP, α = 0.5) | <u>78.2</u> |
| Wise-FT (LP, optim. α) | 80.0 |
| Wise-FT (E2E, α = 0.5) | **82.6** |
| Wise-FT (E2E, optim. α) | 81.7 |

数据较充足的条件下，Finetune backbone在训练数据集上效果较好

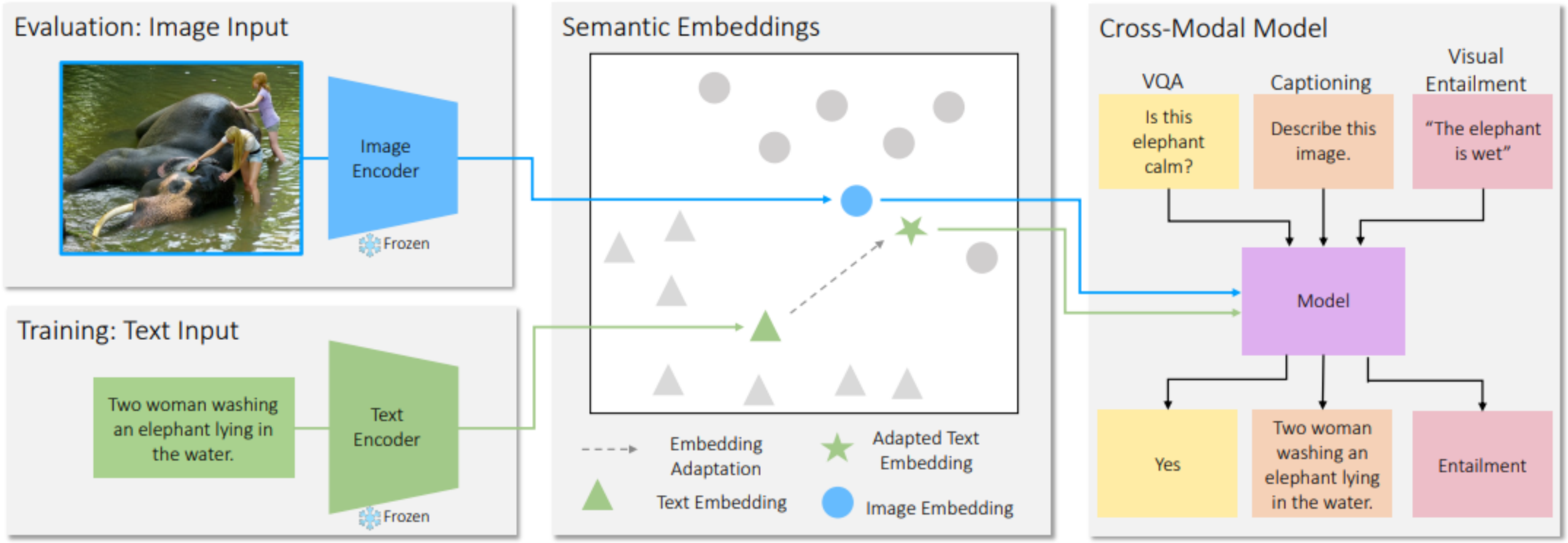
# Vision-Language Pretraining

- Vision-Language Models
- Classification: Parameter-efficient Fine-tuning
- More Challenging Tasks
- More Visual Modalities

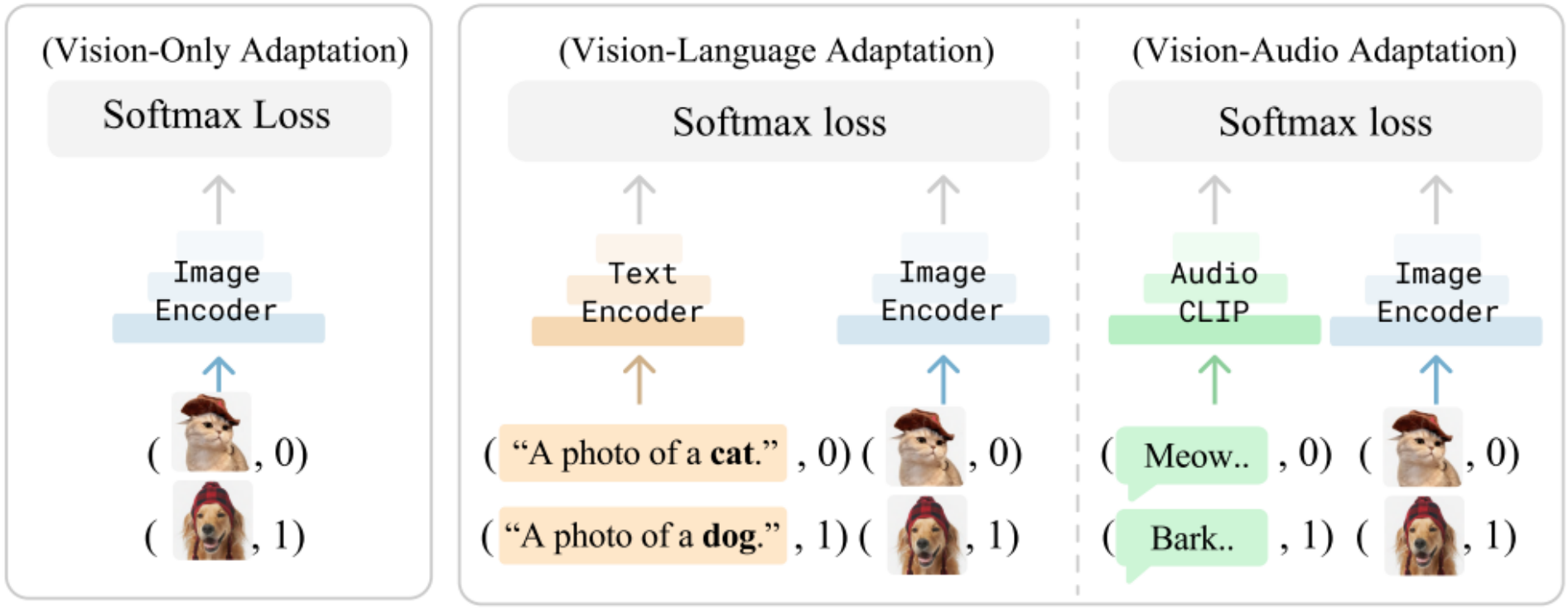
# CapDec（EMNLP2022）



# Learning Visual Tasks Using only Language Data

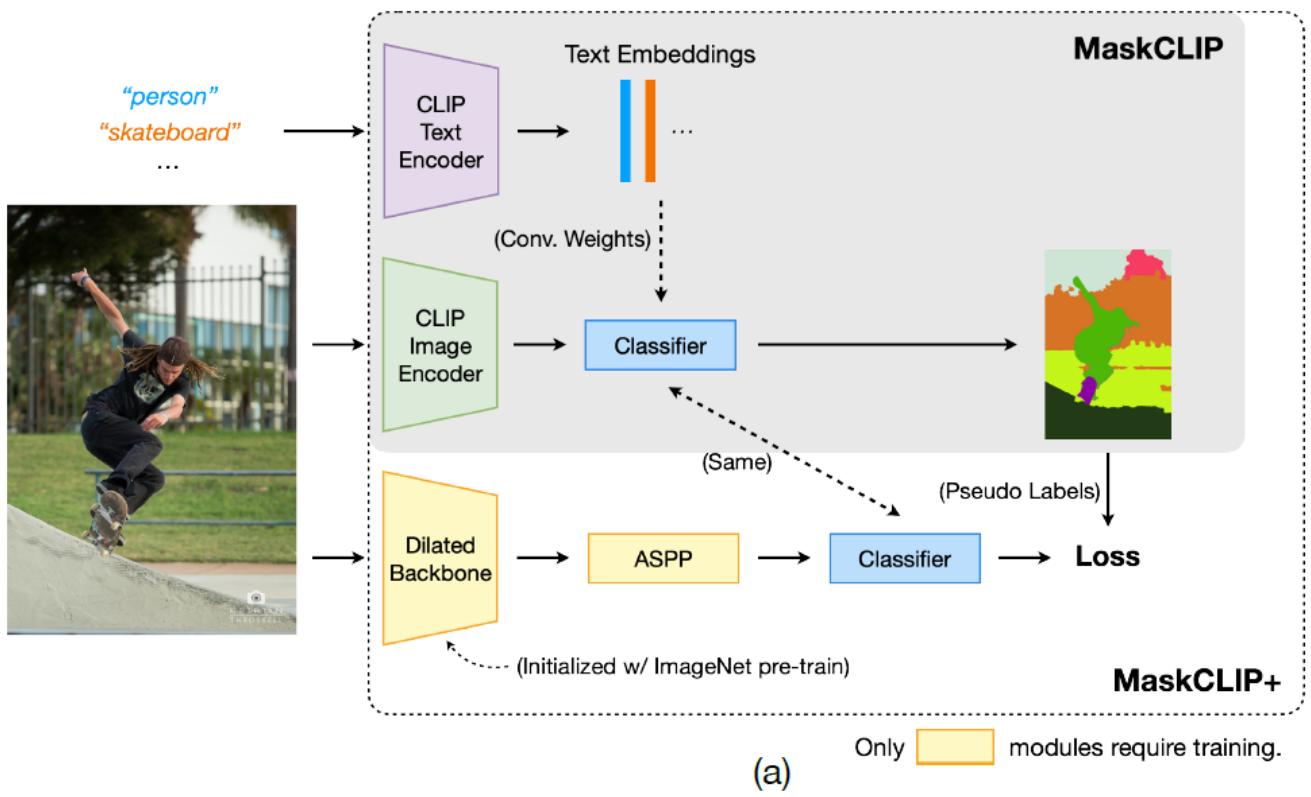


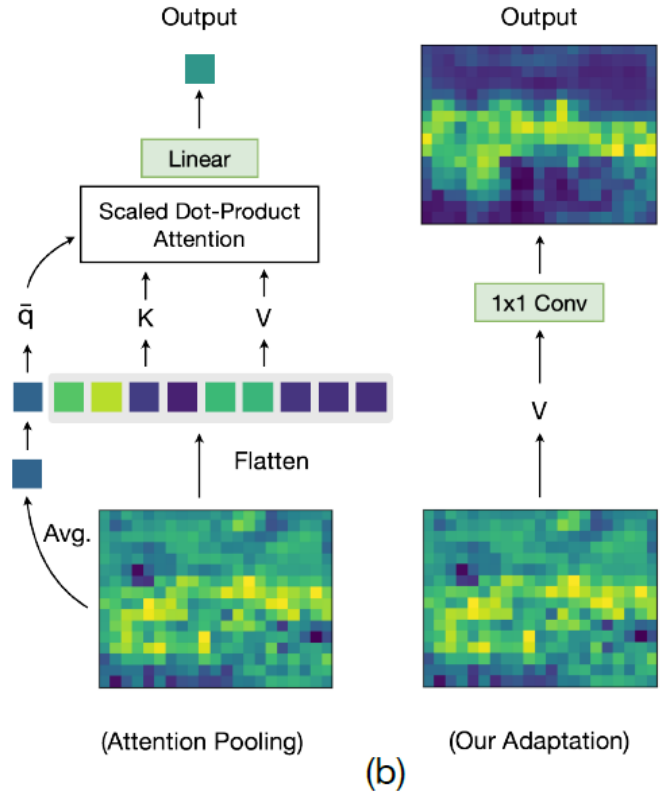
# Multimodality Helps Unimodality: Cross-Modal Few-Shot Learning with Multimodal Models




扩展了任务设定，Multi-modal training & Multi-modal testing

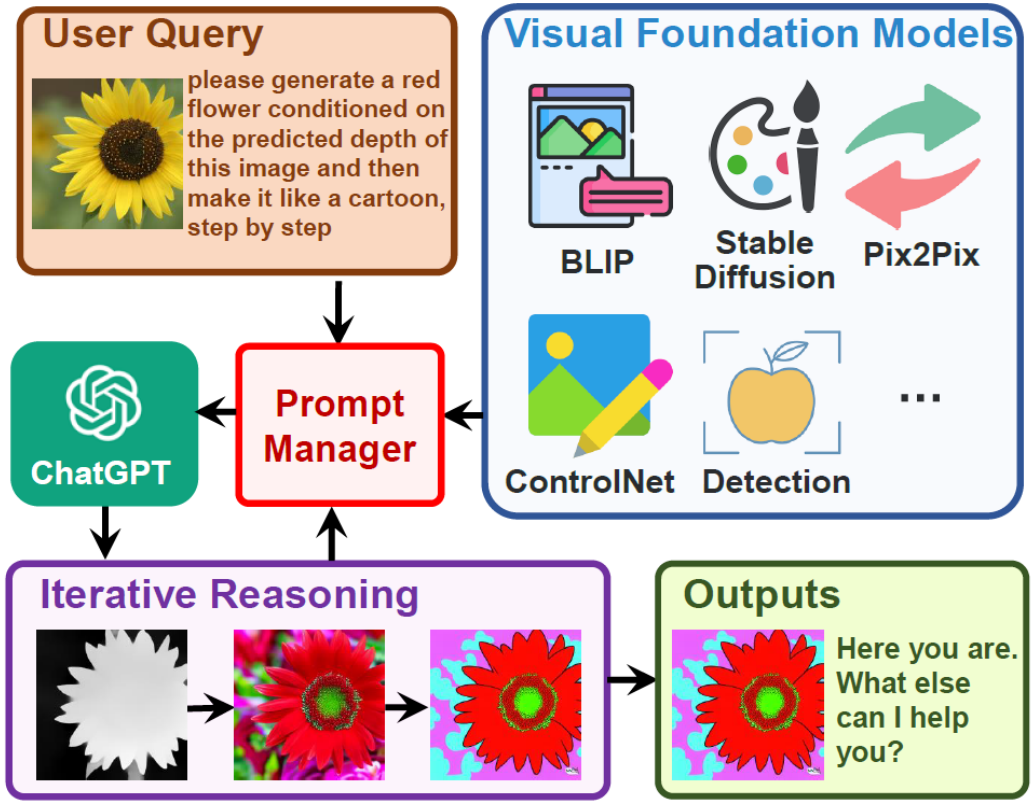
# MaskCLIP (NTU)

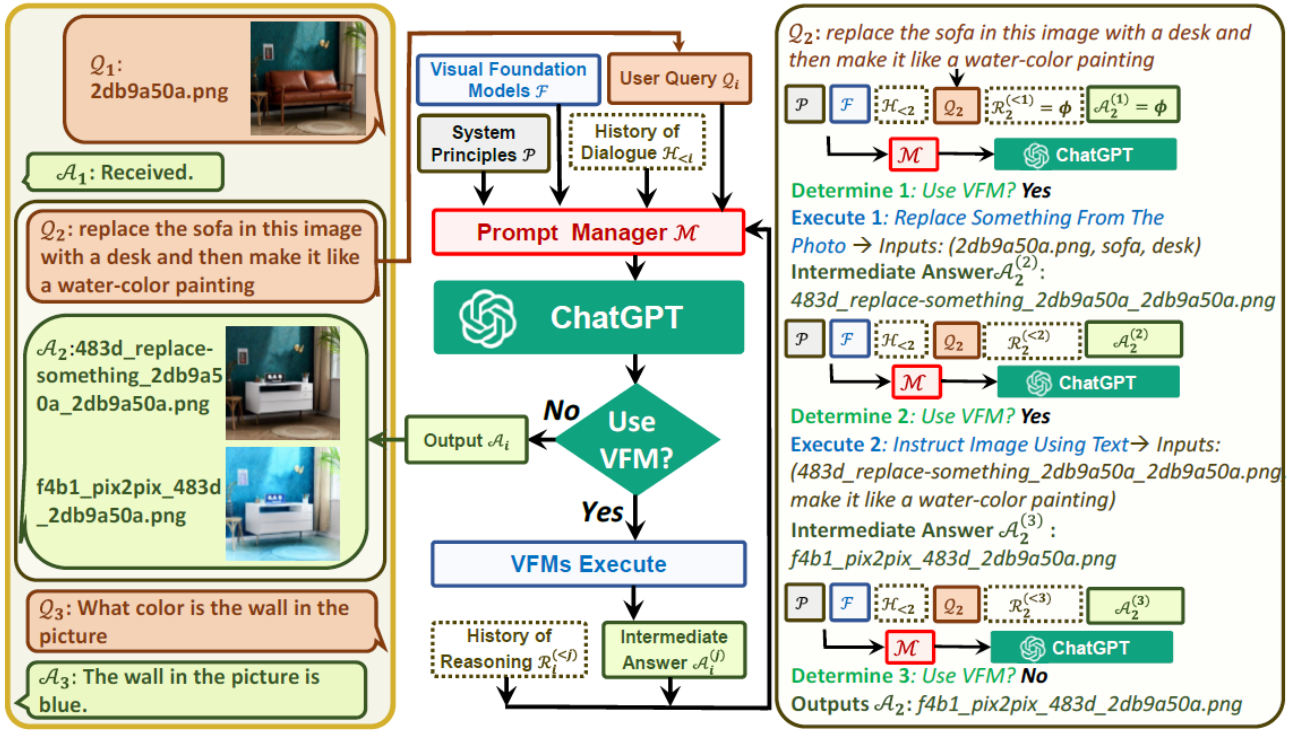



Extract Free Dense Labels from CLIP, ECCV 2022

# Visual ChatGPT



# Visual ChatGPT

$\mathcal{Q}_1$ : Hello, who are you?

$\mathcal{A}_1$ : Hi there! I'm Visual ChatGPT. I'm here to help you with a wide range of text and visual related tasks. How can I help you?

$\mathcal{Q}_2$: I like drawing, but I'm not good at drawing, can you help me? like drawing an apple.

$\mathcal{A}_2$ : Sure, I can help you draw an apple. Here is the image I generated for you:

image/a3453226.png

$\mathcal{Q}_3$: Draw an image:

$\mathcal{A}_3$ : Received. What do you want to know / operate it? You can use image/dcd3af54.png to represent this image in our dialogue.

$\mathcal{Q}_4$: The image/dcd3af54.png is my sketch of an apple and a drinking glass, can you please help me to improve it?

$\mathcal{A}_4$ : I generated a new image based on your sketch. The new image is saved as:

image/569b_scribble2image_dcd3af54_dcd3af54.png

$\mathcal{Q}_5$: Looks good. Can you make the image/569b_scribble2image_dcd3af54_dcd3af54.png into a watercolor painting?

$\mathcal{A}_5$ : I have used the Instruct Image Using Text tool to generate a watercolor painting based on your input. The new image is saved as:

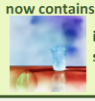
image/0849_pix2pix_569b_dcd3af54.png

$\mathcal{Q}_6$: Wow this is very beautiful. Can you tell me what color this background is?

$\mathcal{A}_6$ : The background color of the image/0849_pix2pix_569b_dcd3af54.png is blue.

$\mathcal{Q}_7$: Correct answer. Can you remove this apple in this picture? and then describe the image?

$\mathcal{A}_7$ : The image/2ca6_remove-something_0849_dcd3af54.png now contains a drinking glass with a blue background.



image/2ca6_remove-something_0849_dcd3af54.png

$\mathcal{Q}_8$: There are still some shadows on the table in the image/2ca6_remove-something_0849_dcd3af54.png, can you help me to replace the table with a black table?

$\mathcal{A}_8$ : I have replaced the table with a black table. The image is saved as

image/7d56_replace-something_2ca6_dcd3af54.png

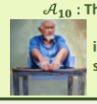
$\mathcal{Q}_9$: Good. Now help me remove the cup too.

$\mathcal{A}_9$ : The image/6c04_remove-something_7d56_dcd3af54.png now contains a black table with a blue background and no cup.

image/6c04_remove-something_7d56_dcd3af54.png

$\mathcal{Q}_{10}$: Can you replace the background of image/6c04_remove-something_7d56_dcd3af54.png with an old man sitting with a lot of vicissitudes in his face?

$\mathcal{A}_{10}$ : The new image is saved as



image/5cd1_replace-something_6c04_dcd3af54.png

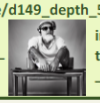
$\mathcal{Q}_{11}$: This picture looks good. Can you describe the color of the old man's clothes?

$\mathcal{A}_{11}$ : The old man's clothes are blue.

$\mathcal{Q}_{12}$: Correct. I want you to detect the depth map of the image/5cd1_replace-something_6c04_dcd3af54 first, and show the depth image, then generate a new image based on this depth map, and finally describe the final image.

$\mathcal{A}_{12}$ : The image/38a5_depth2image_5cd1_dcd3af54.png, which contains 1 chair, 1 person. The person is wearing a white shirt and has a beard. The depth image of the of image/5cd1_replace-something_6c04_dcd3af54 is image/d149_depth_5cd1_dcd3af54.png.

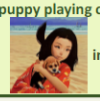
image/d149_depth_5cd1_dcd3af54.png



image/38a5_depth2image_5cd1_dcd3af54.png

$\mathcal{Q}_{13}$: Ok, forget apples and old people. I want you to help me generate a picture of a girl and a puppy playing on the beach.

$\mathcal{A}_{13}$ : I have generated a picture of a girl and a puppy playing on the beach. The image is



image/ab0da279.png

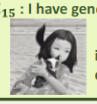
$\mathcal{Q}_{14}$: It looks very nice. Can you replace this puppy with a kitten?

$\mathcal{A}_{14}$ : I have replaced the puppy with a kitten in the image. The new image is

image/d1e1_replace-something_ab0da279_ab0da279.png

$\mathcal{Q}_{15}$: Well done. Can you turn this image/d1e1_replace-something_ab0da279_ab0da279.png into a pencil drawing?

$\mathcal{A}_{15}$ : I have generated a pencil drawing of a girl and a kitten playing on the beach. The image is

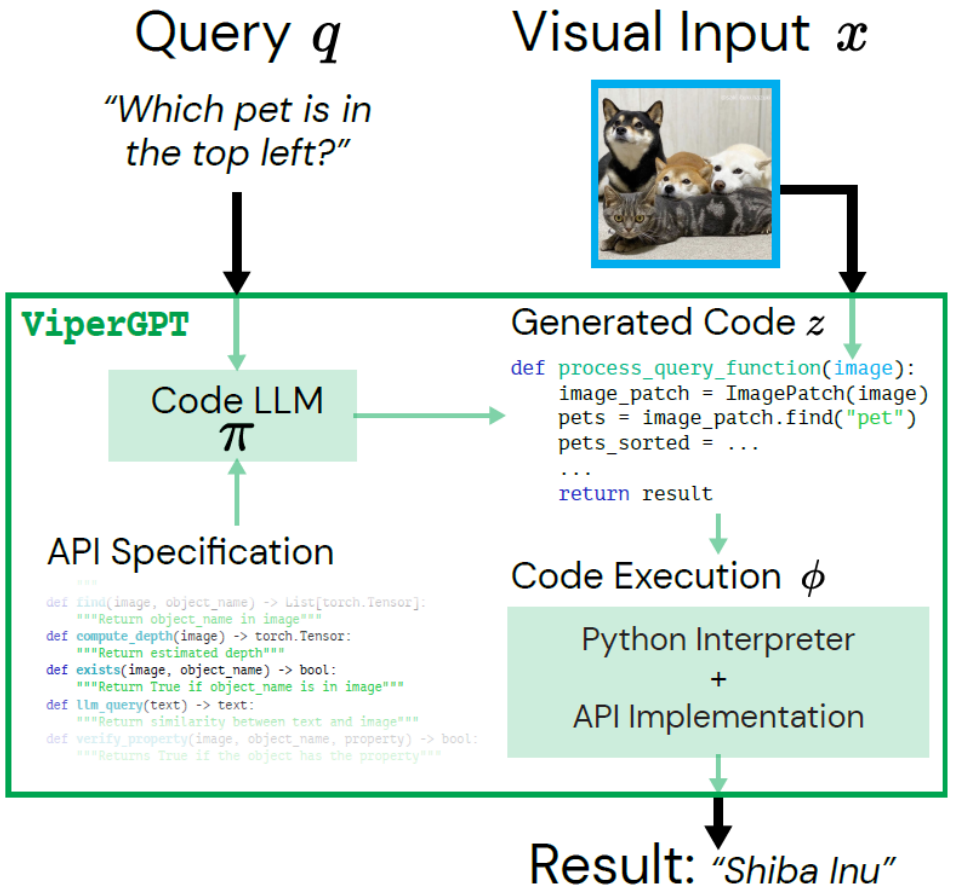


image/6c71_pix2pix_d1e1_ab0da279.png

$\mathcal{Q}_{16}$: Thanks very much. You did a good job. Goodbye!

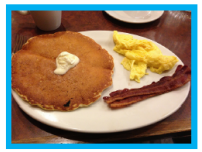
$\mathcal{A}_{16}$ : You're welcome! Have a great day!

# ViperGPT




**Query:** Does that pancake look brown and round?

In:



Generated code

```
def execute_command(image):
    image_patch = ImagePatch(image)
    pancake_patches = image_patch.find("pancake")
    is_brown = pancake_patches[0].verify_property("pancake", "brown")
    is_round = pancake_patches[0].verify_property("pancake", "round")
    return bool_to_yesno(is_brown and is_round)
```

Execution

```
pancake_patches = image_patch.
    find("pancake")
▶ pancake_patches[0] = {ImagePatch}
```

```
...verify_property("pancake", "brown")
▶ is_brown = {bool} True
...verify_property("pancake", round)
▶ is_round = {bool} True
▶ is_brown and is_round = {bool} True
Result: "yes"
```

**Query:** Are there water bottles to the right of the bookcase that is made of wood?

In:

Generated code

```
def execute_command(image):
    image_patch = ImagePatch(image)
    bookcase_patches = image_patch.find("bookcase")
    for bookcase_patch in bookcase_patches:
        is_wood = bookcase_patch.verify_property("bookcase", "wood")
        if is_wood:
            water_bottle_patches = image_patch.find("water bottle")
            for water_bottle_patch in water_bottle_patches:
                if water_bottle_patch.horizontal_center > \
                    bookcase_patch.horizontal_center:
                    return "yes"
            return "no"
    return "no"
```

Execution

```
bookcase_patches= image_patch.
    find("bookcase")
▶ bookcase_patches[0] = {ImagePatch}
```

```
▶ bookcase_patches[0].
    horizontal_center = {float} 239.0
...verify_property("bookcase","wood")
▶ is_wood = {bool} True
```

```
water_bottle_patches = image_patch.
    find("water bottle")
▶ water_bottle_patches[0]
    = {ImagePatches}
```

```
▶ water_bottle_patches[0].
    horizontal_center = {float} 608.5
▶ water_bottle_patch.horizontal_center >
bookcase_patch.horizontal_center =
{bool} True
```

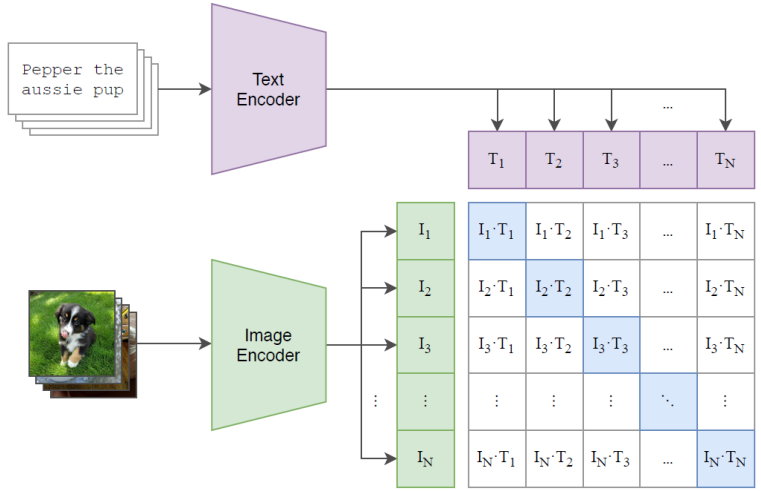
Result: "yes"

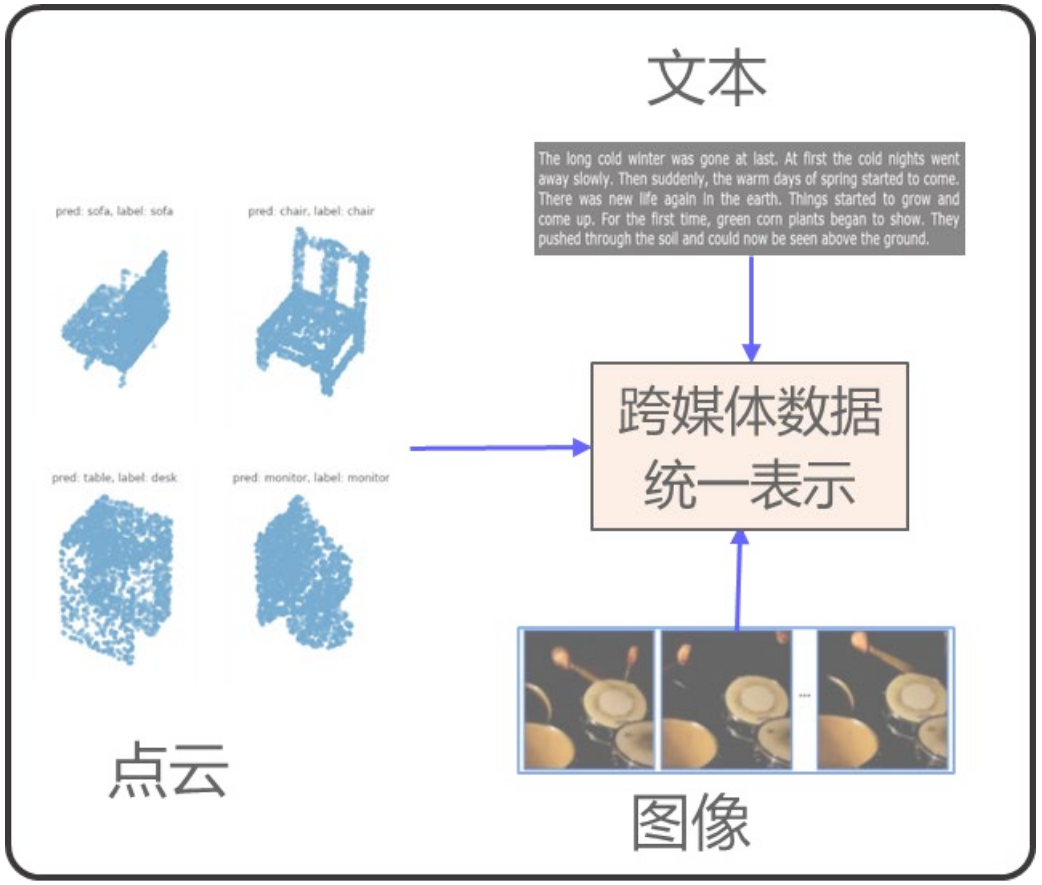
# Vision-Language Pretraining

- Vision-Language Models
- Classification: Parameter-efficient Fine-tuning
- More Challenging Tasks
- More Visual Modalities

# 跨媒体数统一表示

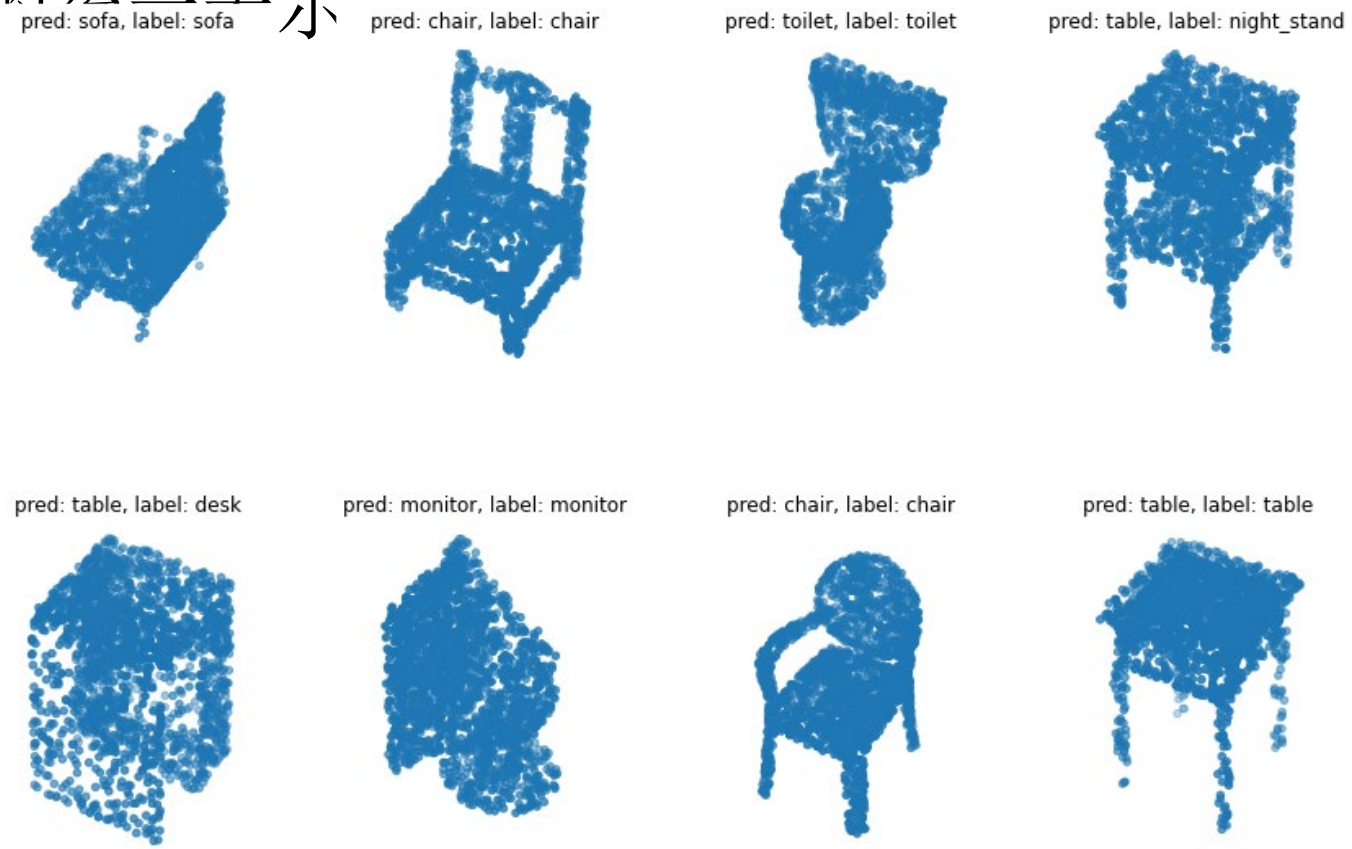
- 图像-文本的统一表示
  - CLIP
  - ALIGN
- 其他视觉模态



# 跨媒体数统一表示

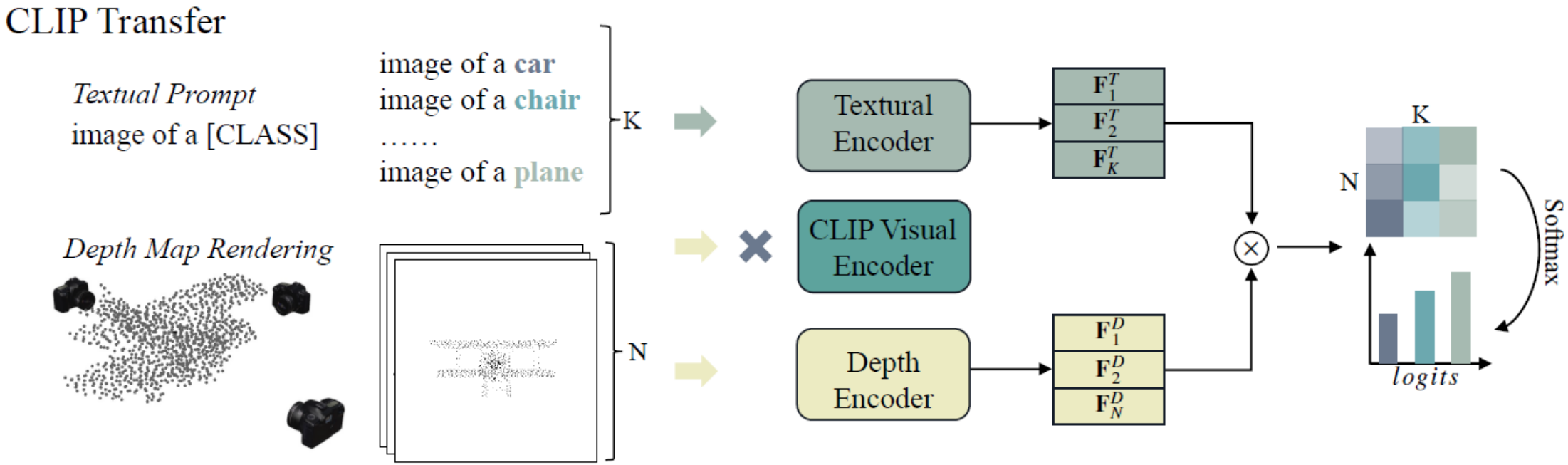
- 3D点云-图像-文本的统一表示
  - 点云分类
  - 点云检测
  - 点云分割
  - ……



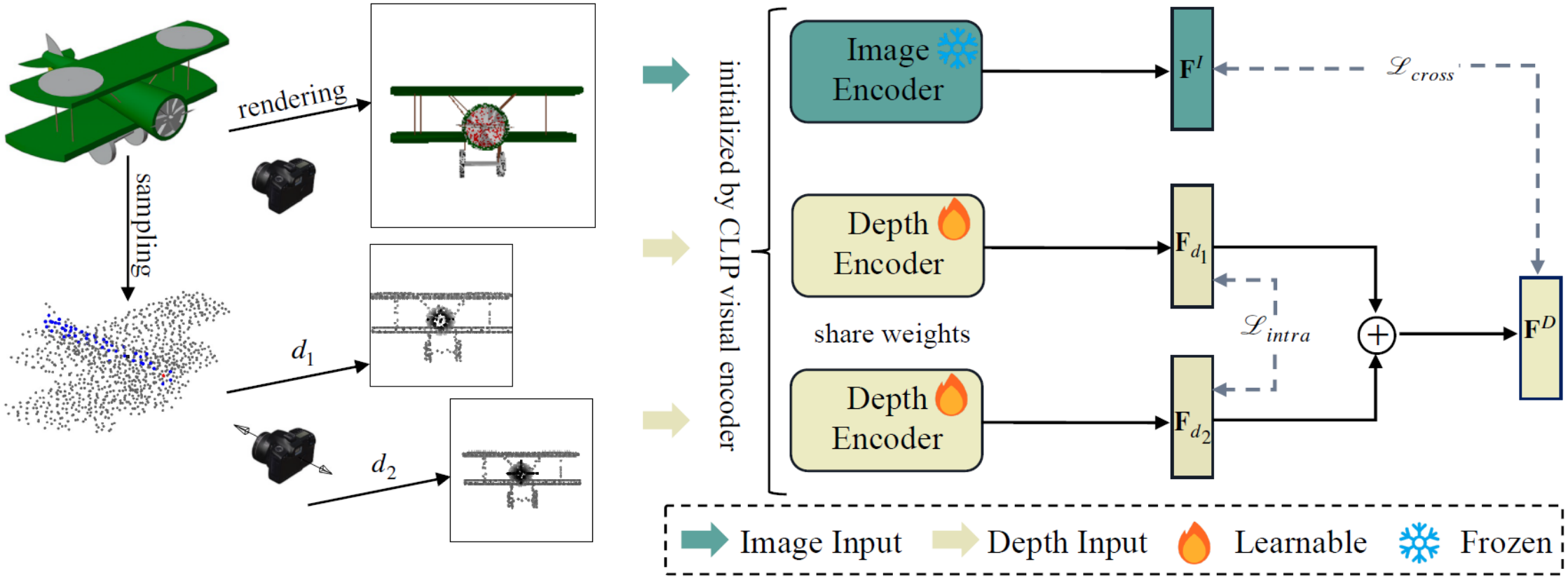

CLIP2Point: Transfer CLIP to Point Cloud Classification with Image-Depth Pre-training, Arxiv 2022.

# 跨媒体数统一表示

- 直接方法存在的问题



# CLIP2Point

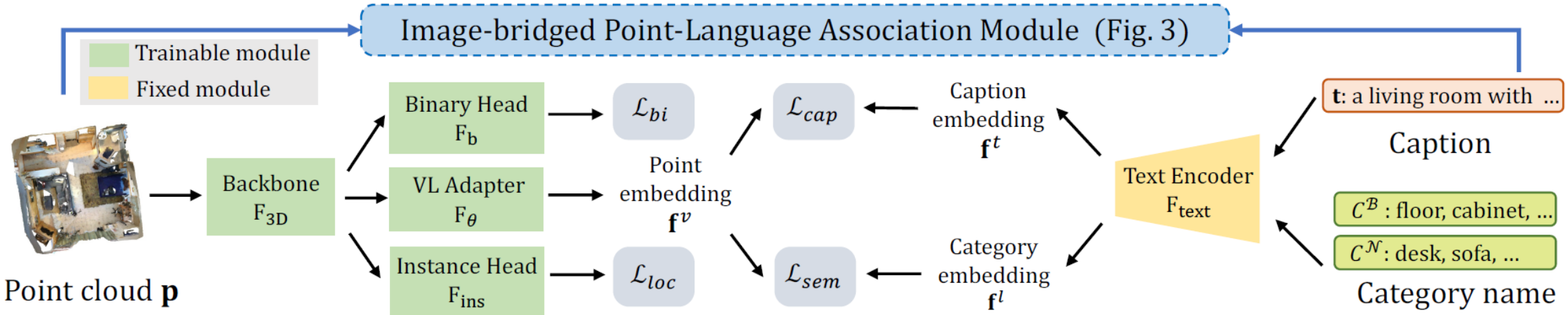



$$\mathcal{L}_{intra} = \frac{1}{2N}\sum_{i=1}^{N}(l^i_{intra}(d_1, d_2) + l^i_{intra}(d_2, d_1))$$

$$\mathcal{L}_{cross} = \frac{1}{2N}\sum_{i=1}^{N}(l^i_{cross}(D, I) + l^i_{cross}(I, D))$$

# 3D Scene Understanding

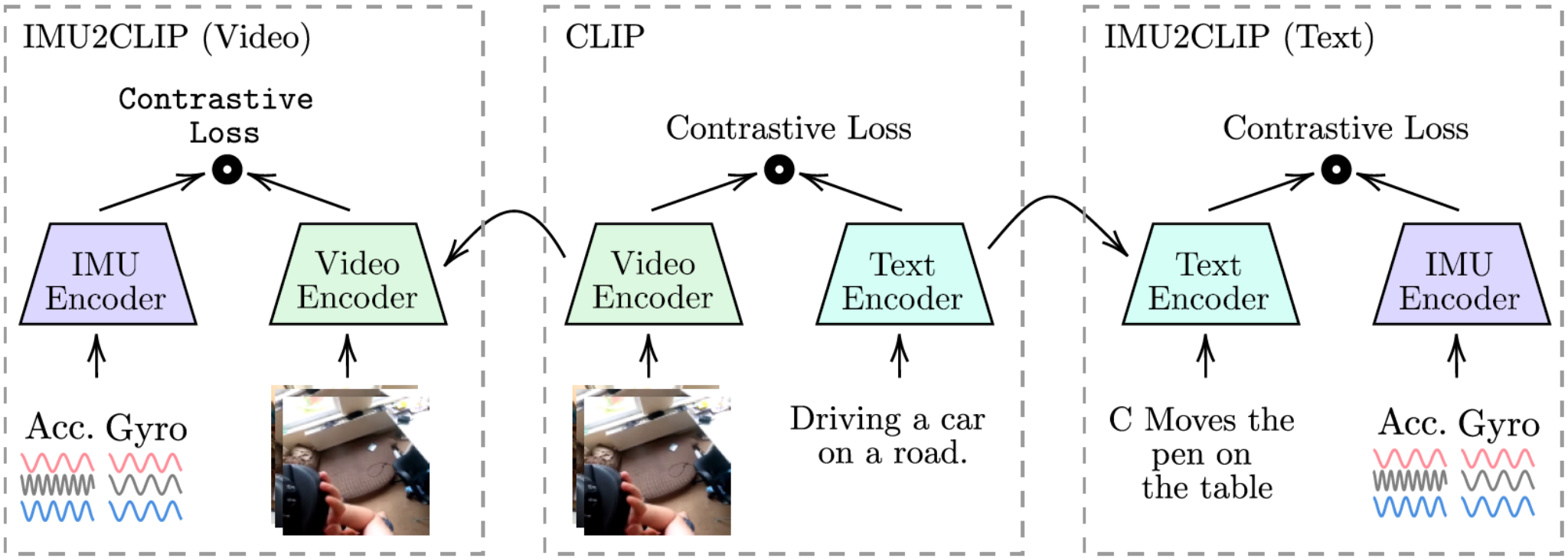
- 字节+港大




Language-driven Open-Vocabulary 3D Scene Understanding, Arxiv 2022

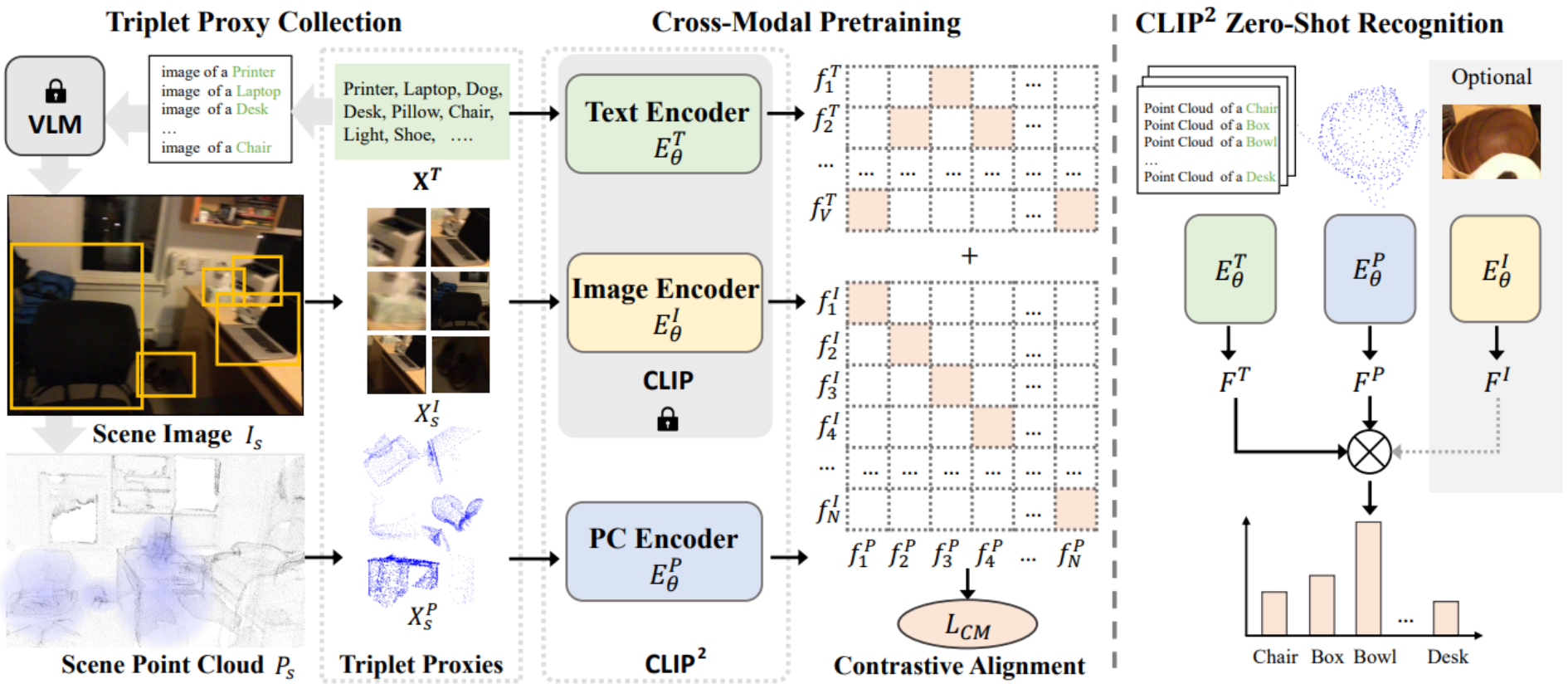
# IMU2CLIP

- Meta AI IMU2CLIP

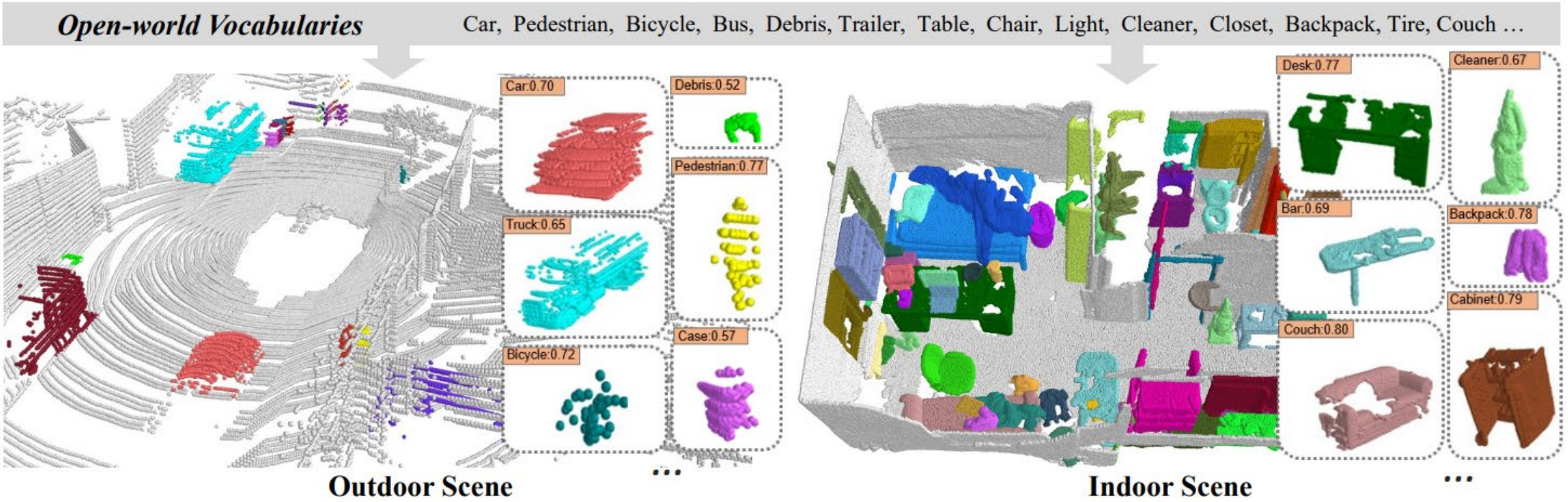



IMU2CLIP: Multimodal Contrastive Learning for IMU Motion Sensors from Egocentric Videos and Text, Arxiv 2022

# CLIP$^2$

- 
Triplet Proxy Collection
Cross-Modal Pretraining
CLIP² Zero-Shot Recognition
VLM
image of a Printer
image of a Laptop
image of a Desk
…
image of a Chair
Printer, Laptop, Dog, Desk, Pillow, Chair, Light, Shoe, ….
$X^T$
Scene Image $I_s$
$X_s^I$
Scene Point Cloud $P_s$
$X_s^P$
Triplet Proxies
Text Encoder $E_\theta^T$
Image Encoder $E_\theta^I$
CLIP
PC Encoder $E_\theta^P$
CLIP²
$f_1^T$ $f_2^T$ … $f_V^T$
$f_1^I$ $f_2^I$ $f_3^I$ $f_4^I$ … $f_N^I$
$f_1^P$ $f_2^P$ $f_3^P$ $f_4^P$ … $f_N^P$
$L_{CM}$
Contrastive Alignment
Point Cloud of a Chair
Point Cloud of a Box
Point Cloud of a Bowl
…
Point Cloud of a Desk
Optional
$E_\theta^T$
$E_\theta^P$
$E_\theta^I$
$F^T$
$F^P$
$F^I$
Chair Box Bowl … Desk

# CLIP$^2$


Open-world Vocabularies
Car, Pedestrian, Bicycle, Bus, Debris, Trailer, Table, Chair, Light, Cleaner, Closet, Backpack, Tire, Couch …
Car:0.70
Debris:0.52
Pedestrian:0.77
Truck:0.65
Case:0.57
Bicycle:0.72
Outdoor Scene
Desk:0.77
Cleaner:0.67
Bar:0.69
Backpack:0.78
Cabinet:0.79
Couch:0.80
Indoor Scene

# Summary

- Rapid progress has been made in VL pre-training
- Challenges and opportunities