



# Deep Learning 1

2025-2026 – Pascal Mettes

## Lecture 9

*Multi-modal deep learning*

# Previous lecture

Lecture	Title	Lecture	Title
1	Intro and history of deep learning	2	AutoDiff
3	Deep learning optimization I	4	Deep learning optimization II
5	Convolutional deep learning	6	Attention-based deep learning
7	Graph deep learning	8	From supervised to unsupervised deep learning
9	Multi-modal deep learning	10	Generative deep learning
11	What doesn't work in deep learning	12	Non-Euclidean deep learning
13	Q&A	14	Deep learning for videos

# This lecture

Vision-language models.

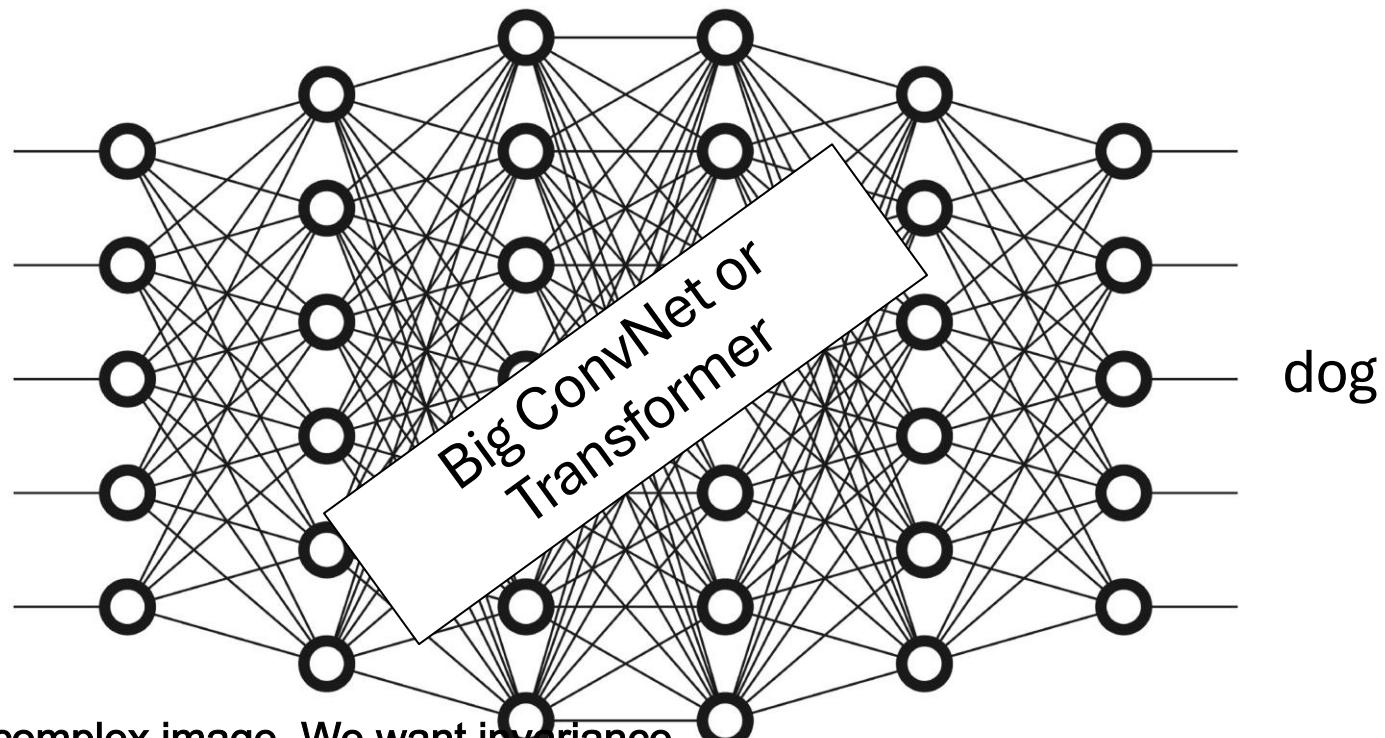
Prompting and improving vision-language models.

Multi-modal LLMs.

# Canonical supervised learning



Also I can generalize only to thing that I've seen during training



here we have a very simple label for a very complex image. We want invariance

If I add different labels for classification the time to classify increase.

Even if I have more labels they are going to be not enough and the computational complexity increase

# Issues with supervised classification

Labels are deemed independent.

*A false assumption. Any mistake is equally bad, which leads to real-world issues.*

An image is more complex than a label.

*An image is a complex scene, with multiple objects in interaction.*

What happens when we see something we didn't train on?

*We can never generalize to new settings.*

# Deep learning beyond class labels: the origins

some tasks which  
seems very difficult  
becomes doable if we  
have an intermediate  
description



Which bird species is this?

American Robin  
*White head*  
*White belly*

Crow  
*Black head*  
*Black belly*

Painted Bunting (male)  
*Blue head*  
*Red belly*

so I want to built an attribute classifier in order to generalize to any new category

ML model from input to attributes and then an hard coded inference systems that goes from attributes to class

# Zero-shot learning

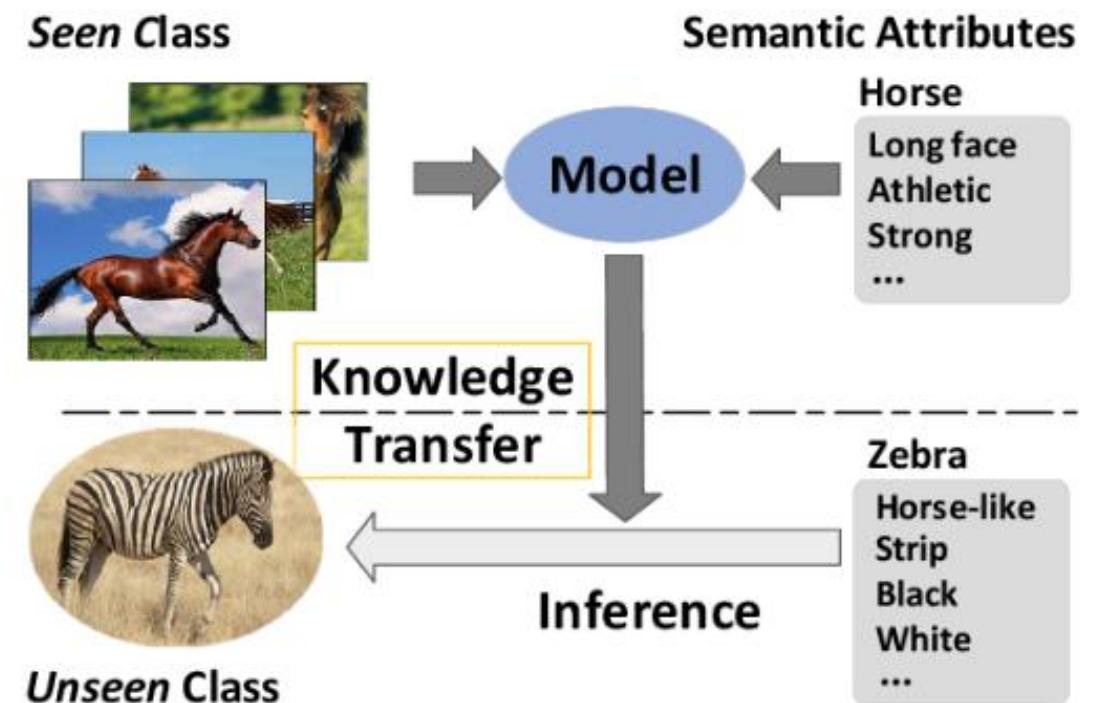
the inference systems looks at the nearest neighbor in the attributes space

**Original idea:** Instead of predicting a class label per image, predict a set of shared attribute labels.

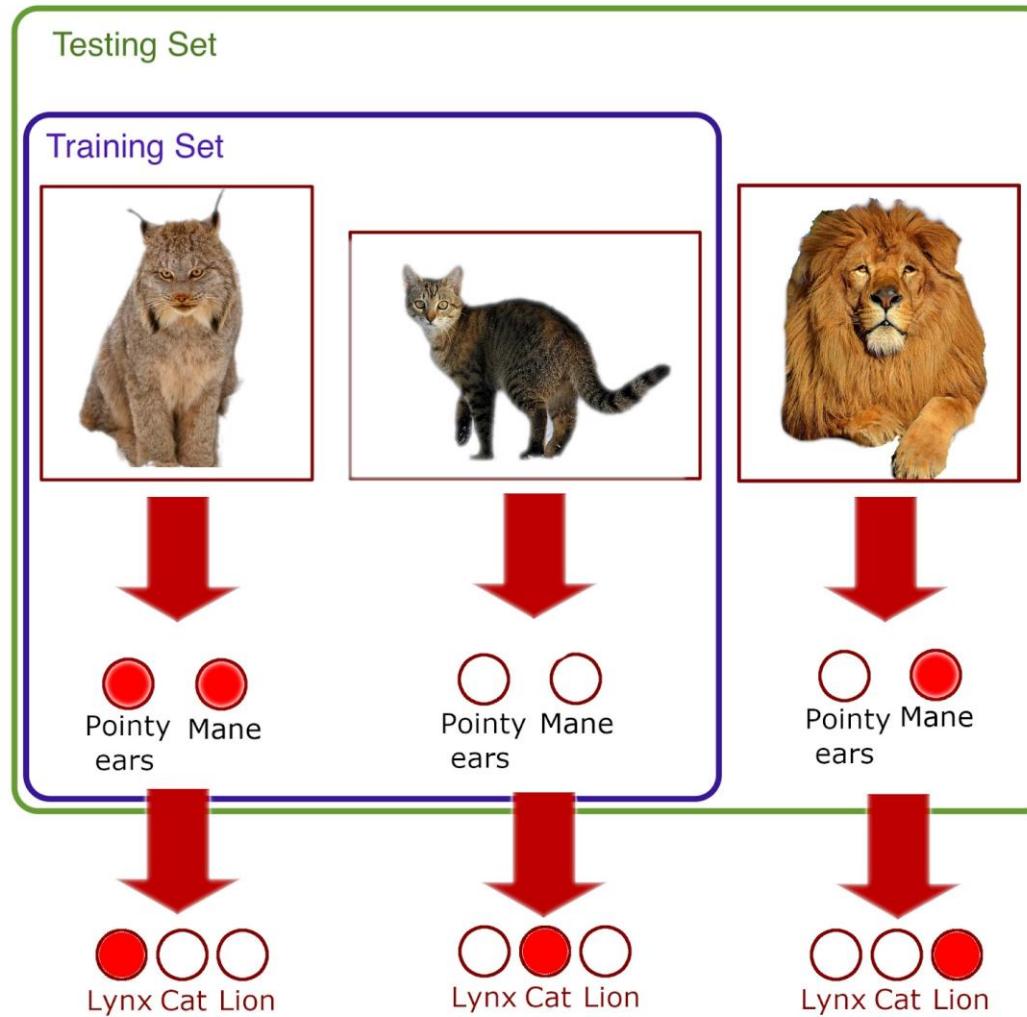
For each class, pre-specify the correct attributes.

During inference, predict all attributes and select class with highest attribute similarity.

we want the labels in a semantic space represented as a continuous vector



# Recognition beyond the training set



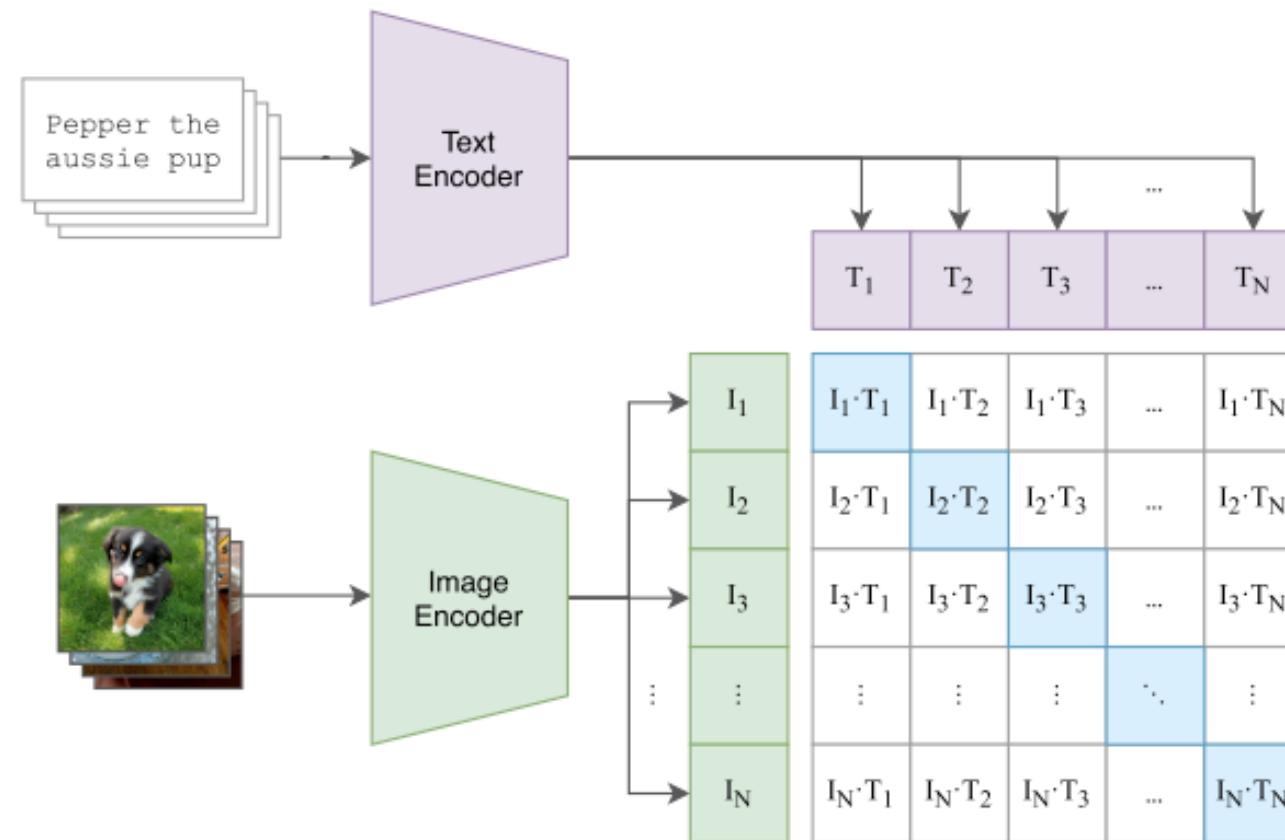
# Vision-language models

extremely simple

we are gonna treat semantic as important, giving it a separate NN.

The label instead of being an element of a set is now a sentence

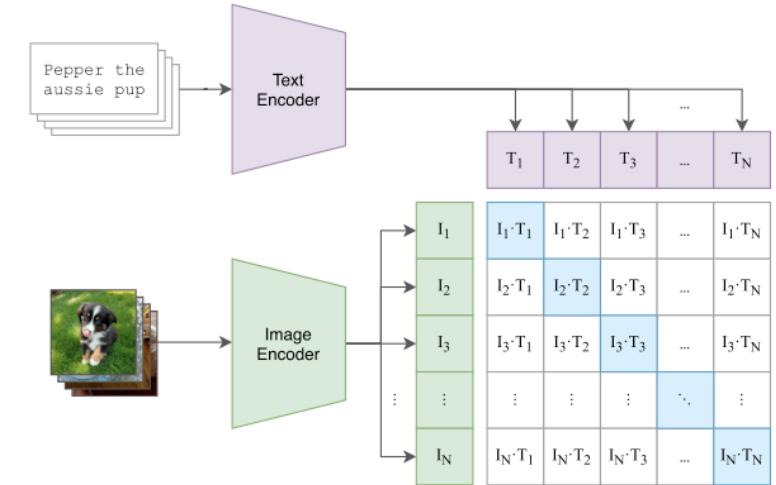
CLIP: Contrastive Language-Image Pre-training



# The idea behind CLIP

semantic is more than just a label

Treat semantics beyond labels.



Two encoders: one for an image, one for a sentence describing it.

we are gonna use contrastive learning so similar samples are gonna be pushed togheeter, while different samples will push the different ones away from each other

Align both with a contrastive loss.

*Pull image-text pair together, push other pairs in batch away.*

# Powerful approach, simple implementation

```
# image_encoder - ResNet or Vision Transformer
# text_encoder - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l] - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T) #[n, d_t]

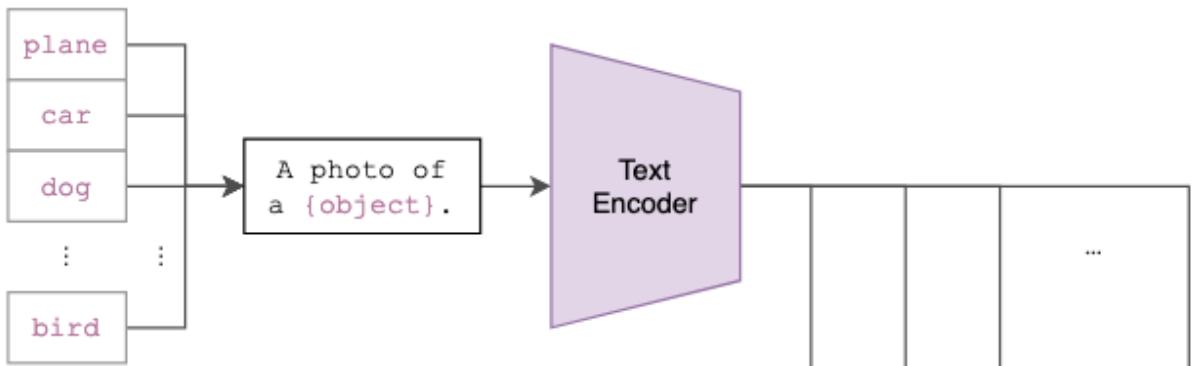
# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1) what the l2 normalization does is
# scaled pairwise cosine similarities [n, n] that only the directions matters, it
logits = np.dot(I_e, T_e.T) * np.exp(t) projects them on the unit sphere

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2 t excacerbaet the means of the norm, (ask),
return very ora distribution
```

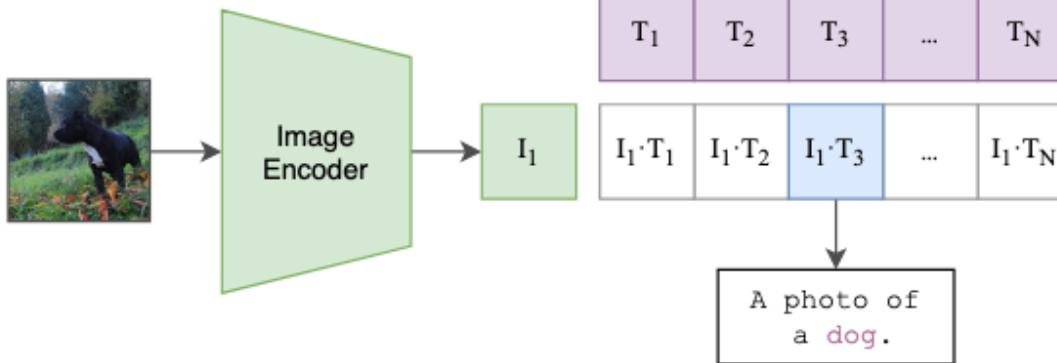
this works really well, the amount of data for which I can get a label now is much higher

# How to use CLIP for “normal” classification?

(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



just add at the beginning the string:

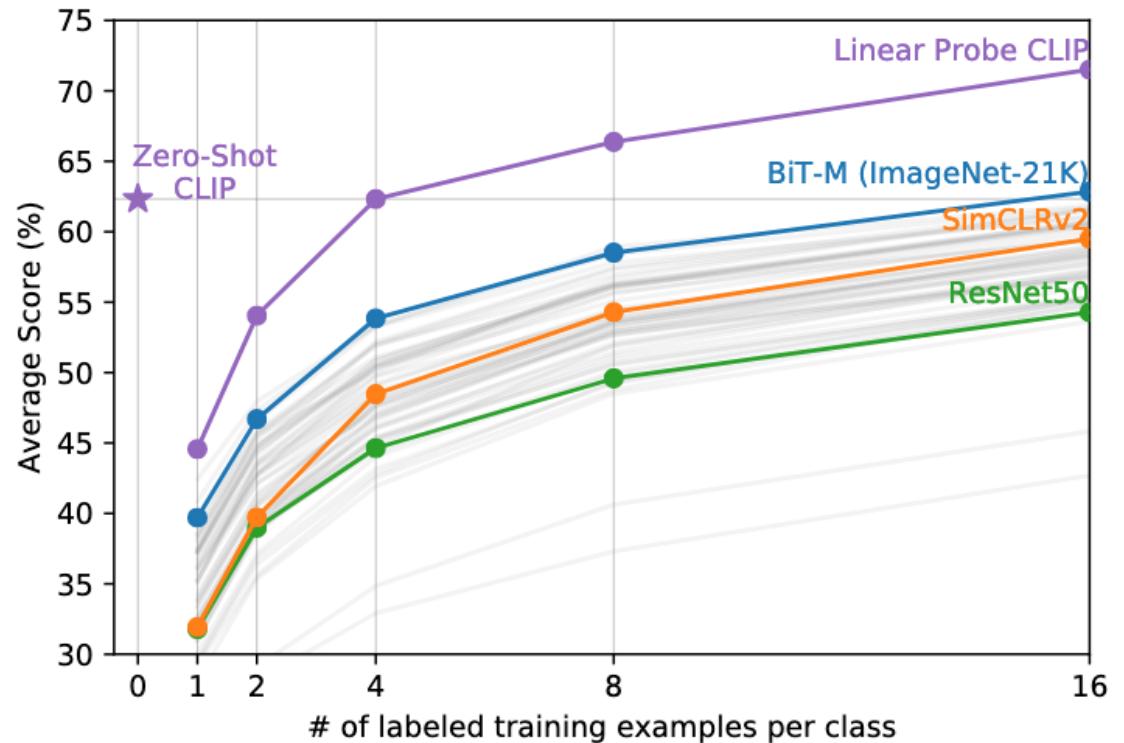
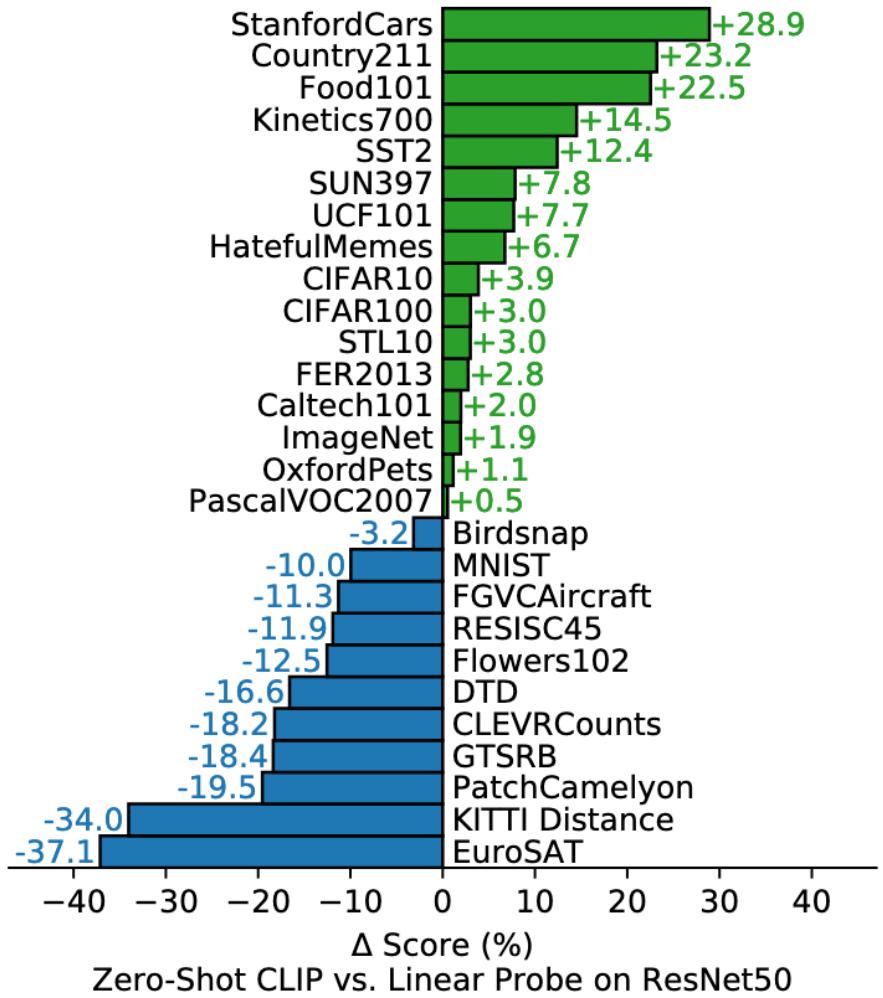
"A photo of a " {lcass}

Could I use the predictions as before, obtain the predicted sentence and comput a similarity with the calss?

Maybe not because it would require to train a new encoder to embed the classes, I want to use a model already pre trained

Issues:  
What about the length of the sequence?

# Pre-training is so powerful, no training needed



# Keys behind the success of CLIP

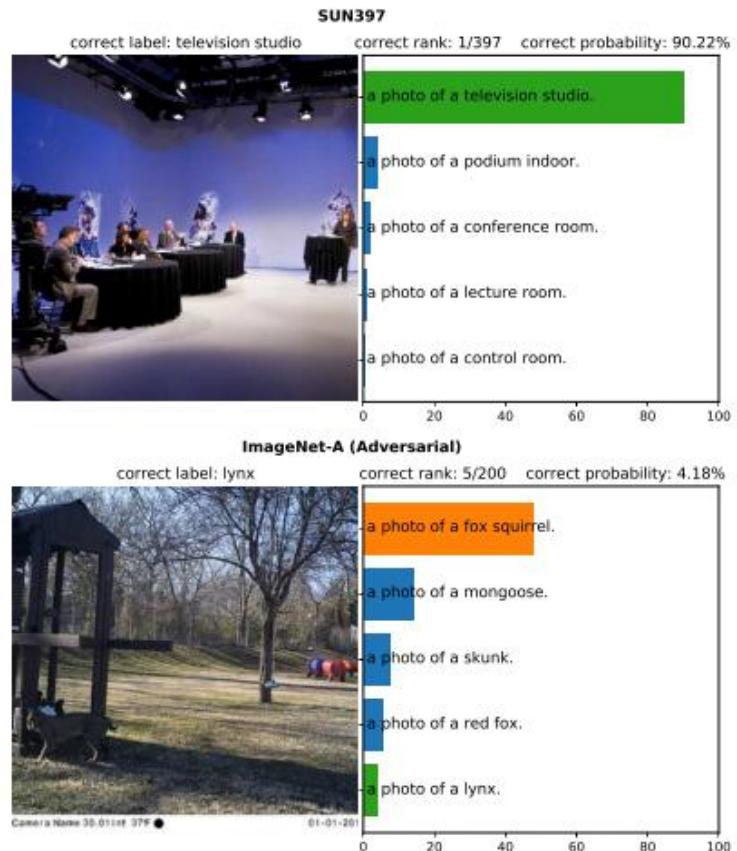
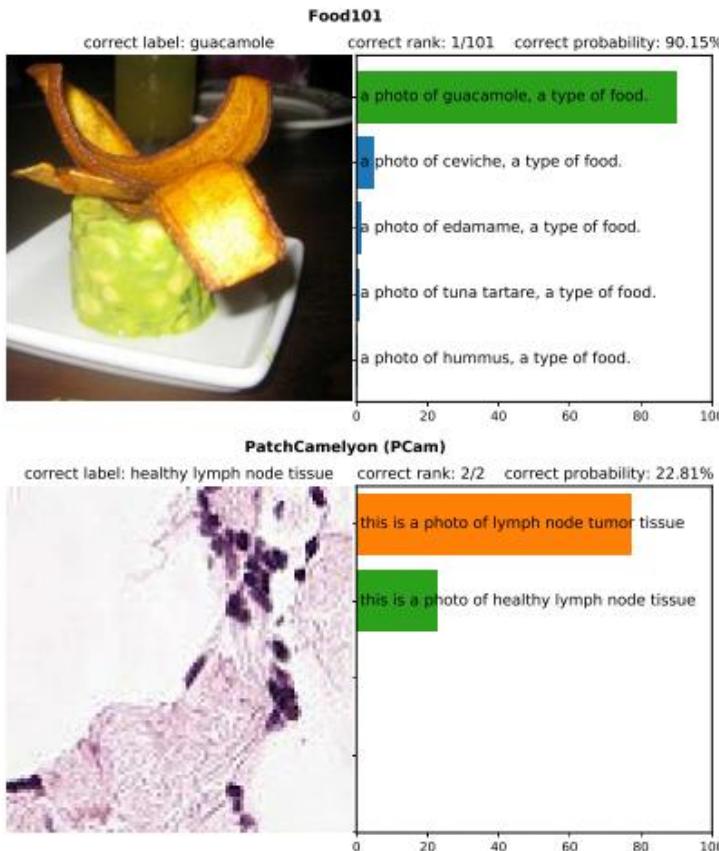
Pre-training on up to 5 billion image-text pairs (~100x bigger than ImageNet).

Sentences are more complex than labels.

Semantics is treated as continuous instead of discrete.

# Zero-shot examples of CLIP

Predicted probability of top 5 classes for each example.



# Robustness of CLIP

	Dataset Examples					ImageNet	Zero-Shot	ResNet101	CLIP	$\Delta$ Score
	ImageNet	ImageNetV2	ImageNet-R	ObjectNet	ImageNet Sketch					
ImageNet						<b>76.2</b>	<b>76.2</b>	0%		
ImageNetV2						64.3	<b>70.1</b>	+5.8%		
ImageNet-R						37.7	<b>88.9</b>	+51.2%		
ObjectNet						32.6	<b>72.3</b>	+39.7%		
ImageNet Sketch						25.2	<b>60.2</b>	+35.0%		
ImageNet-A						2.7	<b>77.1</b>	+74.4%		

Visualizing distribution shift for bananas, a class shared across 5 of the 7 natural distribution shift datasets.

		Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Caltech101	Pets	Flowers	MNIST	FER2013	STL10*	EuroSAT	REISISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMeme:	SSCF	ImageNet
LM RN50	LM RN50	81.3	82.8	61.7	44.2	69.6	74.9	44.9	85.5	71.5	82.8	85.5	91.1	96.6	60.1	95.3	93.4	84.0	73.8	70.2	19.0	82.9	76.4	51.9	51.2	65.2	76.8	65.2
CLIP-RN	50	86.4	88.7	70.3	56.4	73.3	78.3	49.1	87.1	76.4	88.2	89.6	96.1	98.3	64.2	96.6	95.2	87.5	82.4	70.2	25.3	82.7	81.6	57.2	53.6	65.7	72.6	73.3
	101	88.9	91.1	73.5	58.6	75.1	84.0	50.7	88.0	76.3	91.0	92.0	96.4	98.4	65.2	97.8	95.9	89.3	82.4	<b>73.6</b>	26.6	82.8	84.0	60.3	50.3	68.2	73.3	75.7
	50x4	91.3	90.5	73.0	65.7	77.0	85.9	57.3	88.4	79.5	91.5	92.5	97.8	98.5	68.1	97.8	96.4	89.7	85.5	59.4	30.3	83.0	85.7	62.6	52.5	68.0	76.6	78.2
	50x16	93.3	92.2	74.9	72.8	79.2	88.7	62.7	<b>89.0</b>	79.1	93.5	93.7	98.3	<b>98.9</b>	68.7	98.6	97.0	91.4	89.0	69.2	34.8	83.5	88.0	66.3	53.8	71.1	<b>80.0</b>	81.5
	50x64	94.8	94.1	78.6	77.2	81.1	90.5	67.7	<b>88.9</b>	<b>82.0</b>	94.5	95.4	98.9	<b>98.9</b>	<b>71.3</b>	99.1	97.1	92.8	90.2	69.2	40.7	83.7	89.5	69.1	55.0	<b>75.0</b>	<b>81.2</b>	83.6
CLIP-ViT	B/32	88.8	95.1	80.5	58.5	76.6	81.8	52.0	87.7	76.5	90.0	93.0	96.9	<b>99.0</b>	69.2	98.3	97.0	90.5	85.3	66.2	27.8	83.9	85.5	61.7	52.1	66.7	70.8	76.1
	B/16	92.8	96.2	83.1	67.8	78.4	86.7	59.5	<b>89.2</b>	79.2	93.1	94.7	98.1	<b>99.0</b>	69.5	99.0	97.1	92.7	86.6	67.8	33.3	83.5	88.4	66.1	<b>57.1</b>	70.3	75.5	80.2
	L/14	95.2	98.0	87.5	77.0	<b>81.8</b>	<b>90.9</b>	69.4	<b>89.6</b>	<b>82.1</b>	<b>95.1</b>	<b>96.5</b>	99.2	<b>99.2</b>	<b>72.2</b>	<b>99.7</b>	<b>98.2</b>	94.1	<b>92.5</b>	64.7	42.9	85.8	<b>91.5</b>	72.0	<b>57.8</b>	<b>76.2</b>	<b>80.8</b>	83.9
	L/14-336px	<b>95.9</b>	97.9	87.4	79.9	<b>82.2</b>	<b>91.5</b>	<b>71.6</b>	<b>89.9</b>	<b>83.0</b>	<b>95.1</b>	<b>96.0</b>	99.2	<b>99.2</b>	<b>72.9</b>	<b>99.7</b>	<b>98.1</b>	<b>94.9</b>	<b>92.4</b>	69.2	46.4	85.6	<b>92.0</b>	<b>73.0</b>	<b>60.3</b>	<b>77.3</b>	<b>80.5</b>	85.4

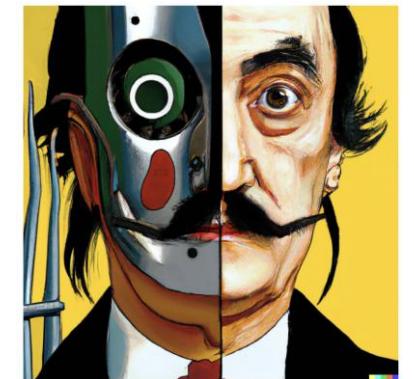
Performance of various pre-trained models over 27 datasets

# Using CLIP in other models

The joint multimodal embedding space of CLIP enables its usage in many other downstream tasks in a zero-shot fashion.

- DALLE-2<sup>1</sup> – a text-guided image generation model,
- CLIP4Clip<sup>2</sup> - video-language retrieval model,
- GroupViT<sup>3</sup> – semantic segmentation model - in a zero-shot manner.

“Vibrant portrait painting of Salvador Dalí with a robotic half.”

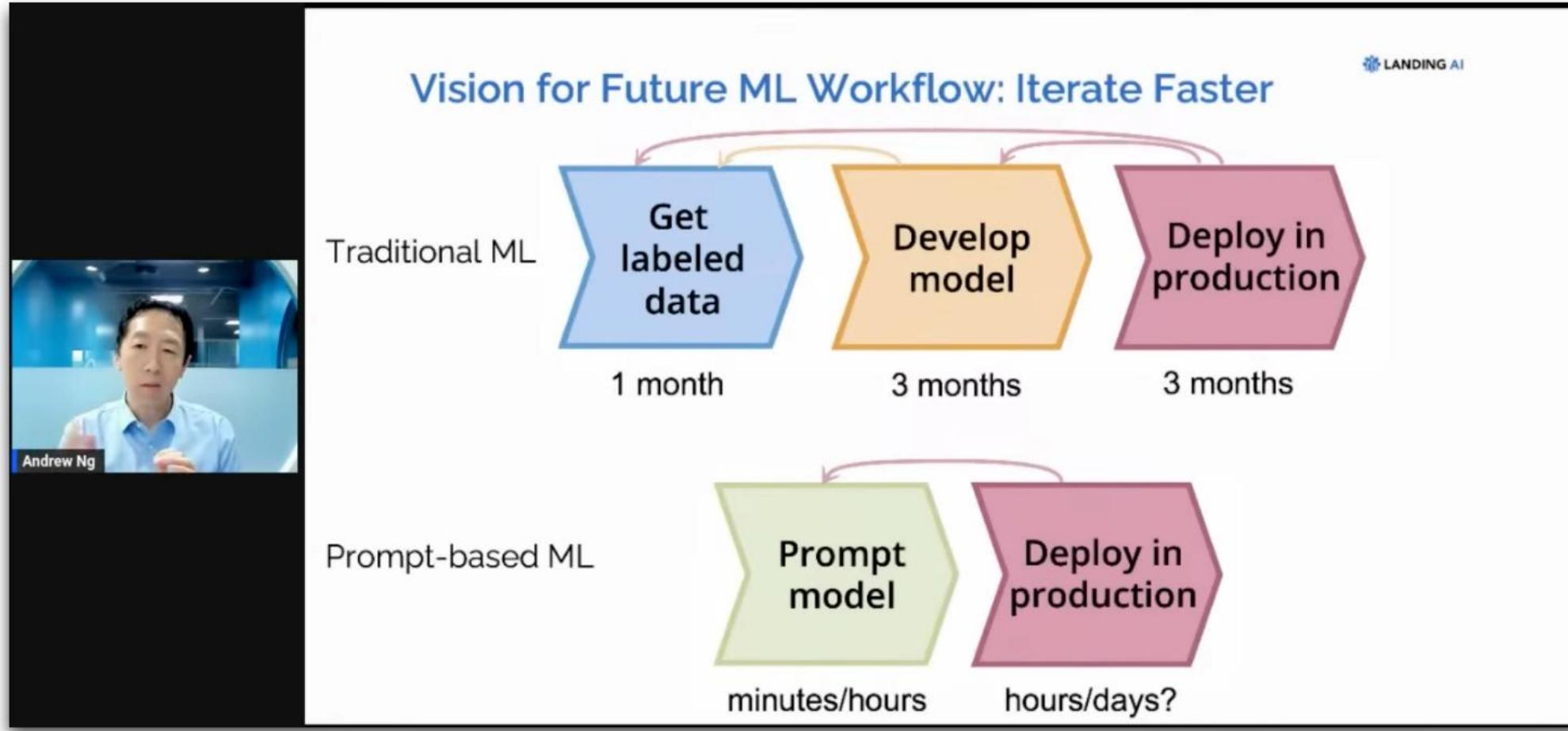


<sup>1</sup> Hierarchical Text-Conditional Image Generation with CLIP Latents, Ramesh et al. (2022)

<sup>2</sup> CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval , Luo et al. (2022)

<sup>3</sup> GroupViT: Semantic Segmentation Emerges from Text Supervision, Xu et al. (2022)

# A peek into the future



# Limitations of CLIP

Models like CLIP simply provide a similarity score between text and image.

Lack the ability to generate language - less suitable to more open-ended tasks.

Require a fixed prompt mechanism to deal with standard classification.

Uses short context windows, ignore hierarchies.

Fine-tuning can actually decrease performance.

how do we formulate the sentence has a huge impact on the performances

# Prompt engineering

a bad photo of a {}.  
a photo of many {}.  
a sculpture of a {}.  
a photo of the hard to see {}.  
a low resolution photo of the {}.  
a rendering of a {}.  
graffiti of a {}.  
a bad photo of the {}.  
a cropped photo of the {}.  
a tattoo of a {}.  
the embroidered {}.  
a photo of a hard to see {}.  
a bright photo of a {}.  
a photo of a clean {}.  
a photo of a dirty {}.  
a dark photo of the {}.  
a drawing of a {}.  
a photo of my {}.  
the plastic {}.  
a photo of the cool {}.  
a close-up photo of a {}.  
a black and white photo of the {}.  
a painting of the {}.  
a painting of a {}.

a pixelated photo of the {}.  
a sculpture of the {}.  
a bright photo of the {}.  
a cropped photo of a {}.  
a plastic {}.  
a photo of the dirty {}.  
a jpeg corrupted photo of a {}.  
a blurry photo of the {}.  
a photo of the {}.  
a good photo of the {}.  
a rendering of the {}.  
a {} in a video game.  
a photo of one {}.  
a doodle of a {}.  
a close-up photo of the {}.  
a photo of a {}.  
the origami {}.  
the {} in a video game.  
a sketch of a {}.  
a doodle of the {}.  
a origami {}.  
a low resolution photo of a {}.  
the toy {}.  
a rendition of the {}.

a photo of the clean {}.  
a photo of a large {}.  
a rendition of a {}.  
a photo of a nice {}.  
a photo of a weird {}.  
a blurry photo of a {}.  
a cartoon {}.  
art of a {}.  
a sketch of the {}.  
a embroidered {}.  
a pixelated photo of a {}.  
itap of the {}.  
a jpeg corrupted photo of the {}.  
a good photo of a {}.  
a plushie {}.  
a photo of the nice {}.  
a photo of the small {}.  
a photo of the weird {}.  
the cartoon {}.  
art of the {}.  
a drawing of the {}.  
a photo of the large {}.  
a black and white photo of a {}.  
the plushie {}.

**A slight change in wording could lead to big changes in performance**

[https://github.com/openai/CLIP/blob/main/notebooks/Prompt\\_Engineering\\_for\\_ImageNet.ipynb](https://github.com/openai/CLIP/blob/main/notebooks/Prompt_Engineering_for_ImageNet.ipynb)

# The effect of prompt engineering

A way to solve the fact that different templates got different scores is to use an ensemble of templates and average the accuracies obtained

Dataset	Prompt	Accuracy
Caltech101	a [CLASS].	82.68
	a photo of [CLASS].	80.81
	a photo of a [CLASS].	86.29
	[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>91.83</b>
Flowers102	a photo of a [CLASS].	60.86
	a <b>flower</b> photo of a [CLASS].	65.81
	a photo of a [CLASS], a <b>type of flower</b> .	66.14
	[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>94.51</b>
Describable Textures (DTD)	a photo of a [CLASS].	39.83
	a photo of a [CLASS] <b>texture</b> .	40.25
	[CLASS] texture.	42.32
	[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>63.58</b>
EuroSAT	a photo of a [CLASS].	24.17
	a <b>satellite</b> photo of [CLASS].	37.46
	a centered satellite photo of [CLASS].	37.56
	[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>83.53</b>

**A slight change in wording could lead to big changes in performance**

# What makes a good prompt?

A person riding a  
motorcycle on a dirt road.



Two dogs play in the grass.



# Can we learn to prompt instead?

Caltech101



Prompt	Accuracy
a [CLASS].	82.68
a photo of [CLASS].	80.81
a photo of a [CLASS].	86.29
[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>91.83</b>

(a)

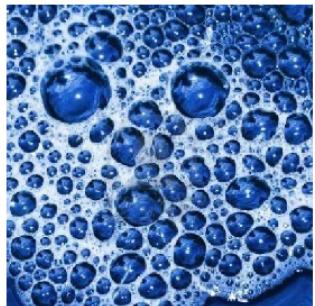
Flowers102



Prompt	Accuracy
a photo of a [CLASS].	60.86
a flower photo of a [CLASS].	65.81
a photo of a [CLASS], a type of flower.	66.14
[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>94.51</b>

(b)

Describable Textures (DTD)



Prompt	Accuracy
a photo of a [CLASS].	39.83
a photo of a [CLASS] texture.	40.25
[CLASS] texture.	42.32
[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>63.58</b>

(c)

EuroSAT

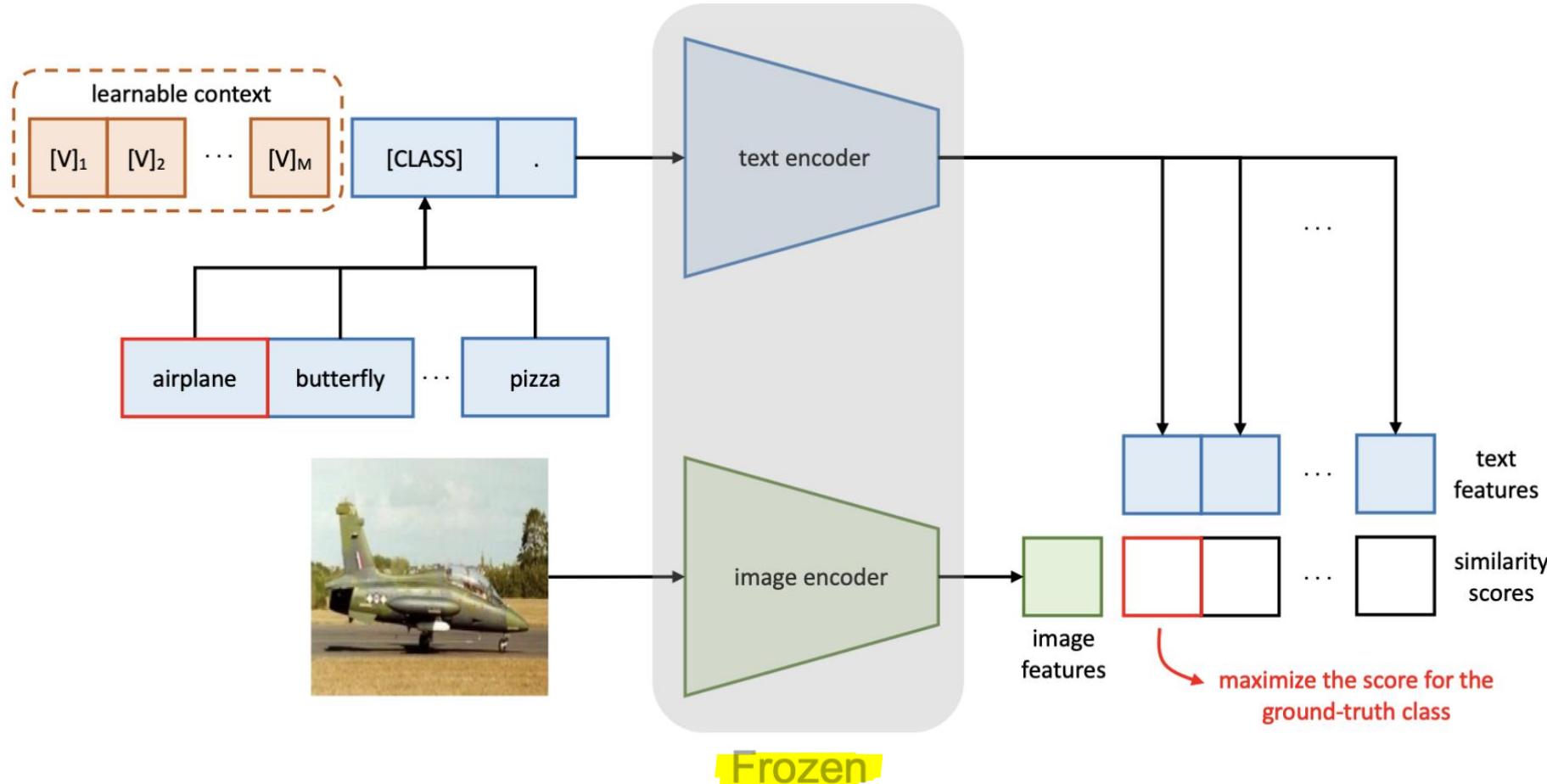


Prompt	Accuracy
a photo of a [CLASS].	24.17
a satellite photo of [CLASS].	37.46
a centered satellite photo of [CLASS].	37.56
[V] <sub>1</sub> [V] <sub>2</sub> ... [V] <sub>M</sub> [CLASS].	<b>83.53</b>

(d)

# Learning to prompt for vision-language models

the token are not anymore words, we are gonna let go semantic and maximize my performance on the task



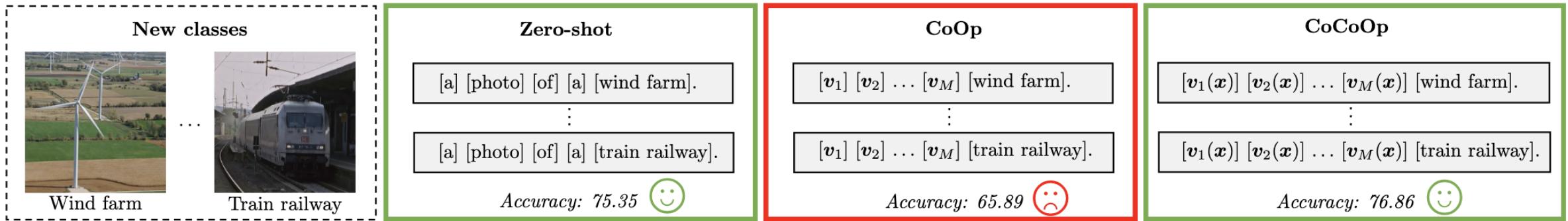
# CoCoOp: Beyond statically learned prompts

overfits and does not generalize to wider unseen classes

Issue in CoOp: The prompt is overfit to wider unseen classes.

Why? The prompt embeddings ignore the visual instance.

it has a bias towards whatever it has seen most often before



the prompt embedding ignores the visual instance. All I'm gonna do here is.

each token will be also dependent on my image. The image embedding is not only used to compute the similarity logits but also to compute a meta token that will be combined with the tokens that I'm gonna learn

“Conditional Prompt Learning for Vision-Language Models.” Zhou et al. CVPR 2022.

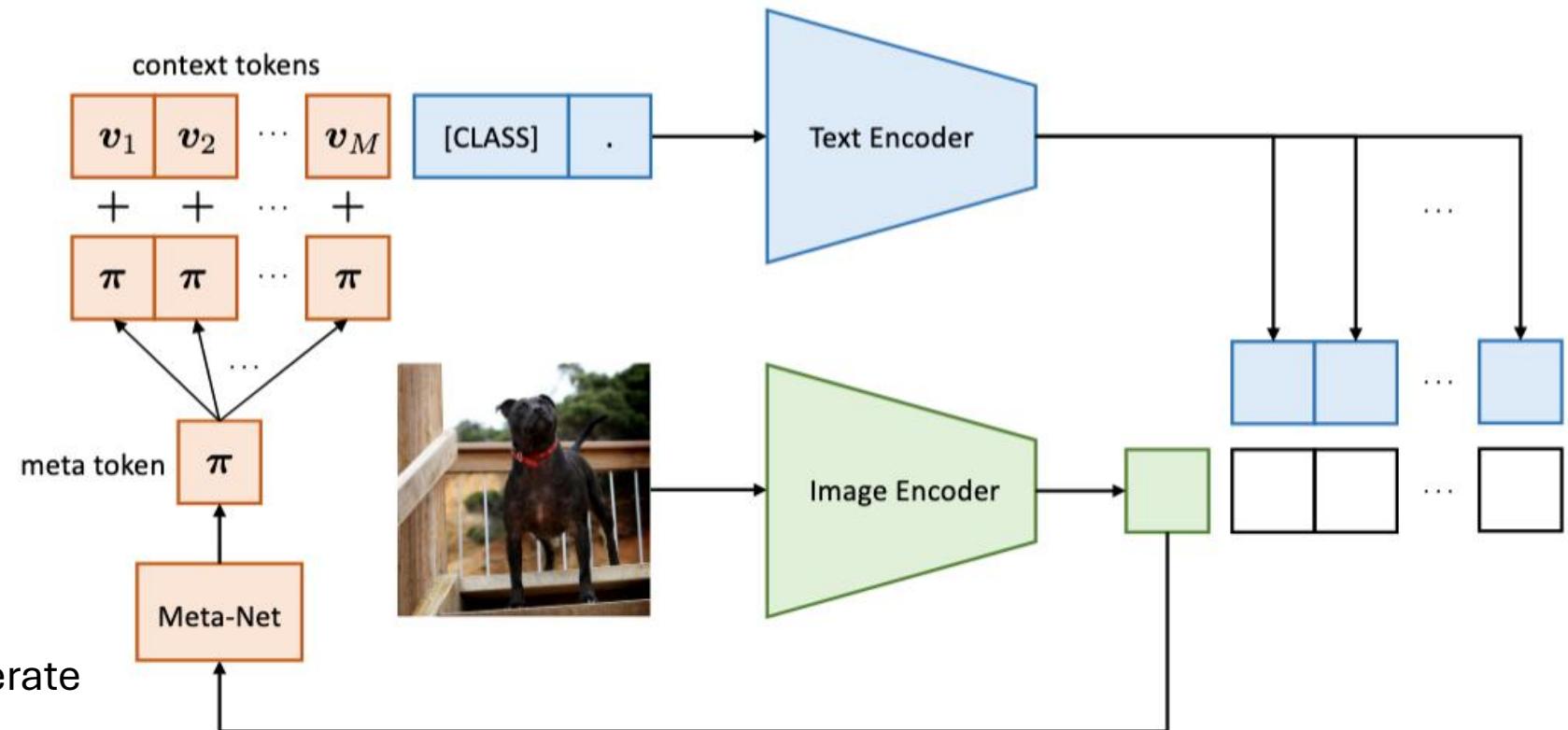
# Conditional prompt learning

I'm going to bias the tokens with what im seeing in the image, hoping that this will debias the scores

I have no more any interpretability, this are just random tokens

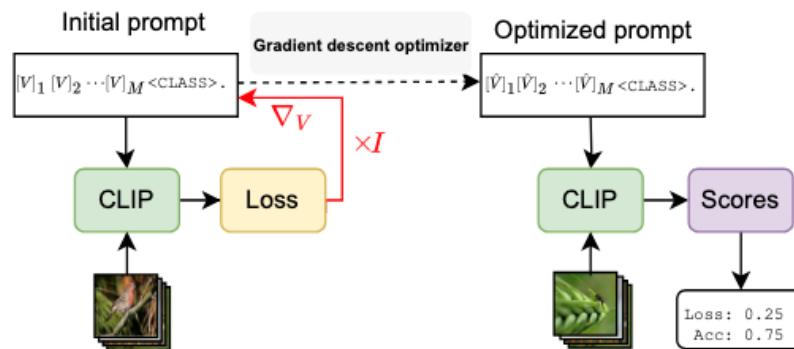
Component 1:  
Context vectors (ala CoOp).

Component 2:  
Small network that learns to generate  
an image-conditional token.

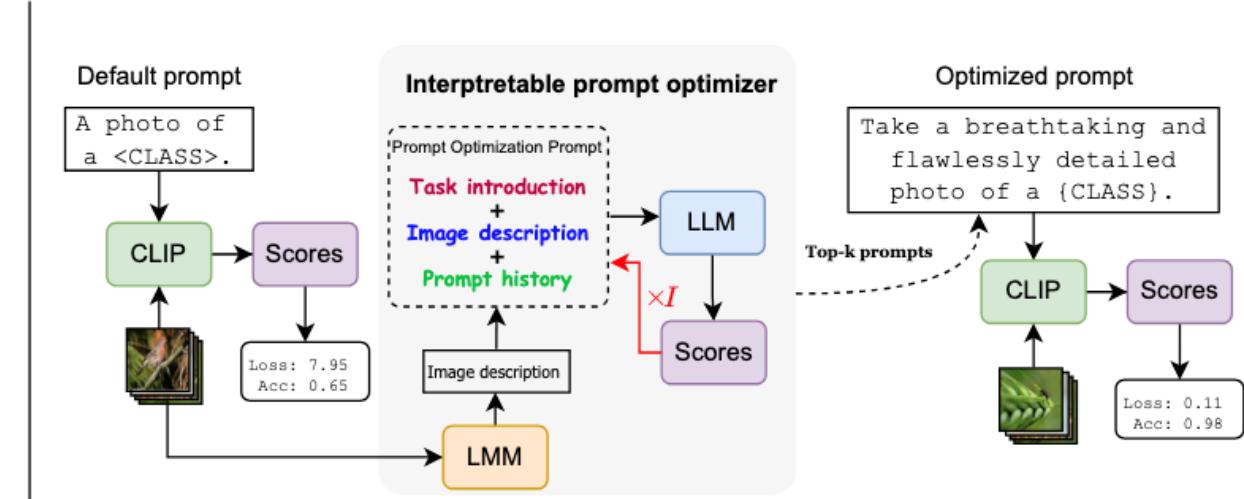


# Interpretable prompt learning

Issue with all prompt learners: we have no clue what they learned.



(a) Gradient-based prompt optimization.



(b) Interpretable prompt optimization.

in context learning, the LLm works to refine the sentences in order to improve the scores, at each iteration wed compute new scores from new sentences and in the end the new sentences really improve the original scvores

“IPO: Interpretable Prompt Optimization for Vision-Language Models.” Du et al. NeurIPS 2024.

# Summarizing prompts

Prompts make it possible to have in-context generalization.

First attempt: manually curate a prompt.

Prompt learning optimizes prompt tokens, massively improving results.

Interpretable prompt learning makes prompt learning understandable, but require LLM probes (which can become quite expensive).

# Break

# Looking beyond CLIP

In CLIP, the text encoder is learnt from scratch, why not start from an LLM?

How to efficiently fine-tune VLMs?

What about other modalities?

CLIP is flat, but the real-world is hierarchical. How can we fix this?

# Using an LLM: Flamingo

Flamingo is a Transformer-based architecture for multimodal few-shot tasks (image captioning, visual dialogue or visual question answering).

Able to learn from only a few input/output examples i.e., *in few-shot settings*.

*It processes arbitrarily interleaved images and text as prompt;*

*And it generates output text in an open-ended manner.*

Performs in-context learning (like GPT) but with images and text as context.



# Pre-training Flamingo

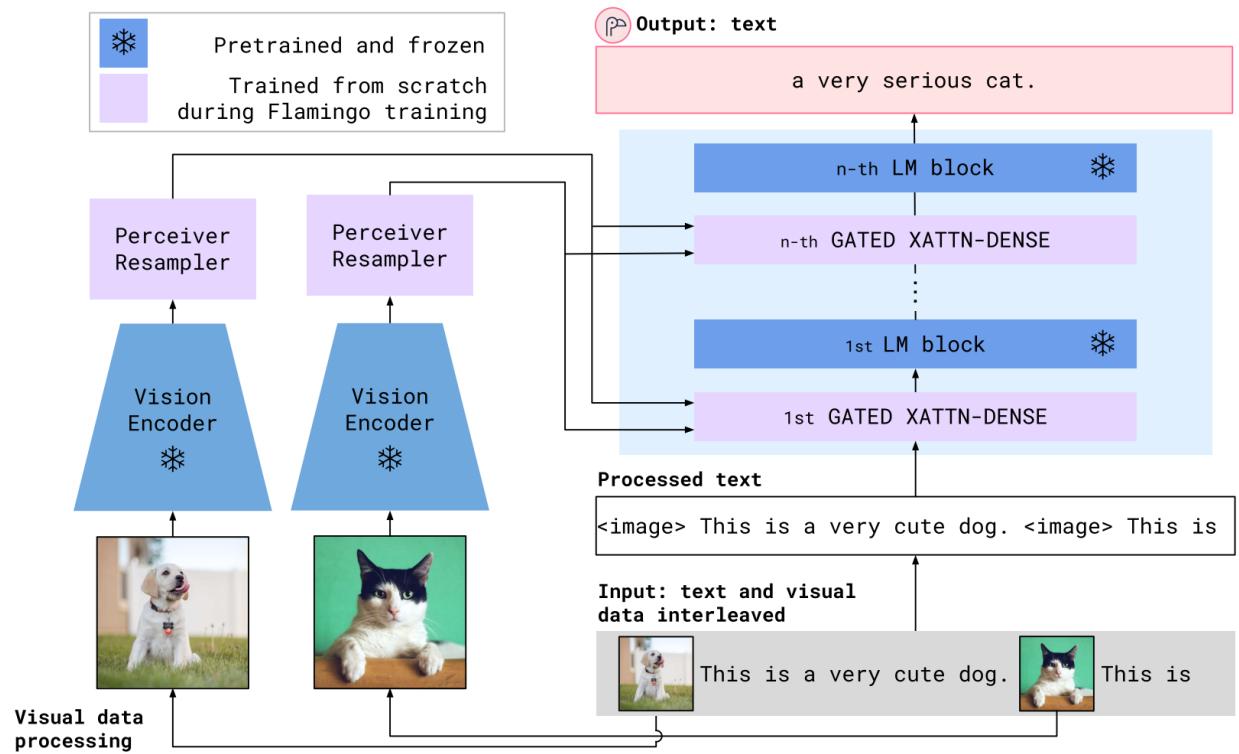
Vision side: an encoder with contrastive text-image approach, à la CLIP.

Language side: existing autoregressive LM trained on a large text corpus.

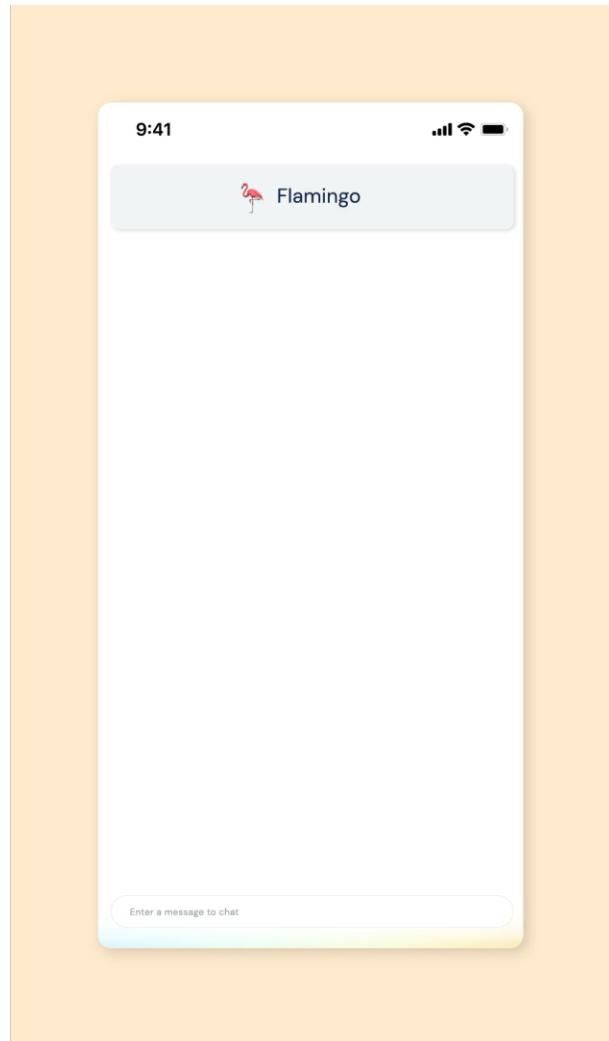
Linked via a learnable attention component (the Perceiver).

*It outputs a fixed-size set of visual tokens.*

*Which are used to condition the frozen LM, trained to generate text.*



# Example output of Flamingo



Which city is this?

This is a street sign in New York City.

Which street corner is it?

It is the intersection of Madison Avenue and East 42nd Street.

Which subway station is nearby?

The closest subway station is Grand Central Station.

I would like you to read the color and then say the color it is written in. For example:

PURPLE

Color is "Purple" and it is written in red.

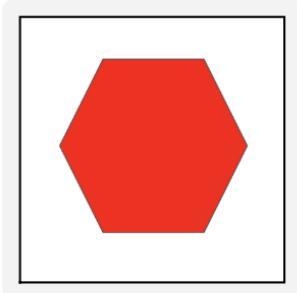
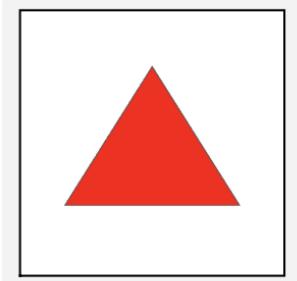
GREEN

Color is "Green" and it is written in blue.

YELLOW

Color is "Yellow" and it is written in green.

BLACK



What is the difference between these two images?

The first one is a triangle and the second one is a hexagon.

Which one has larger internal angles?

The hexagon has larger internal angles.

# Looking beyond CLIP

In CLIP, the text encoder is learnt from scratch, why not start from an LLM?

How to efficiently fine-tune VLMs?

What about other modalities?

CLIP is flat, but the real-world is hierarchical. How can we fix this?

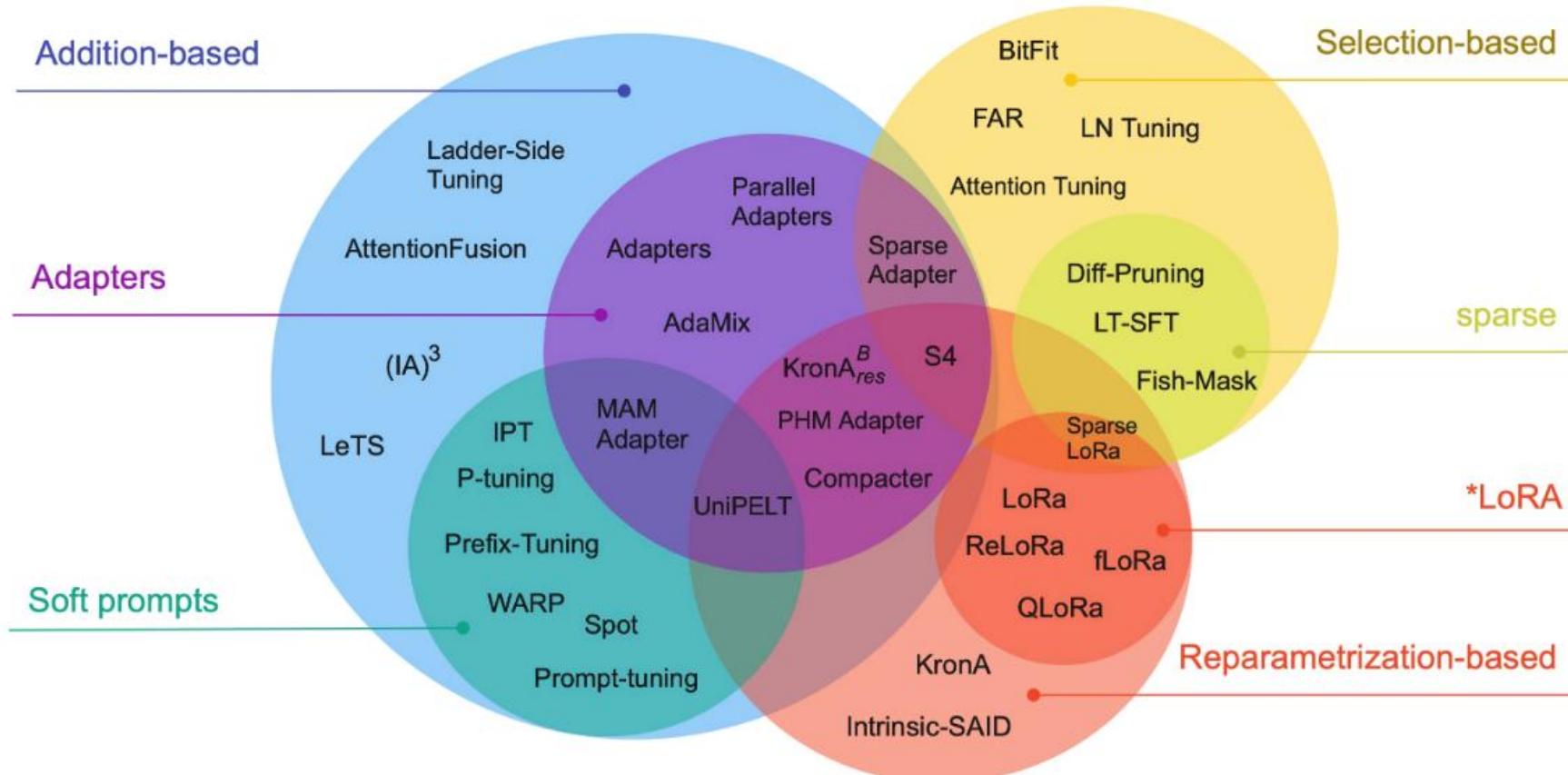
# Fine-tuning VLMs

Sometimes, in-context learning is not enough, we need to alter the parameters.

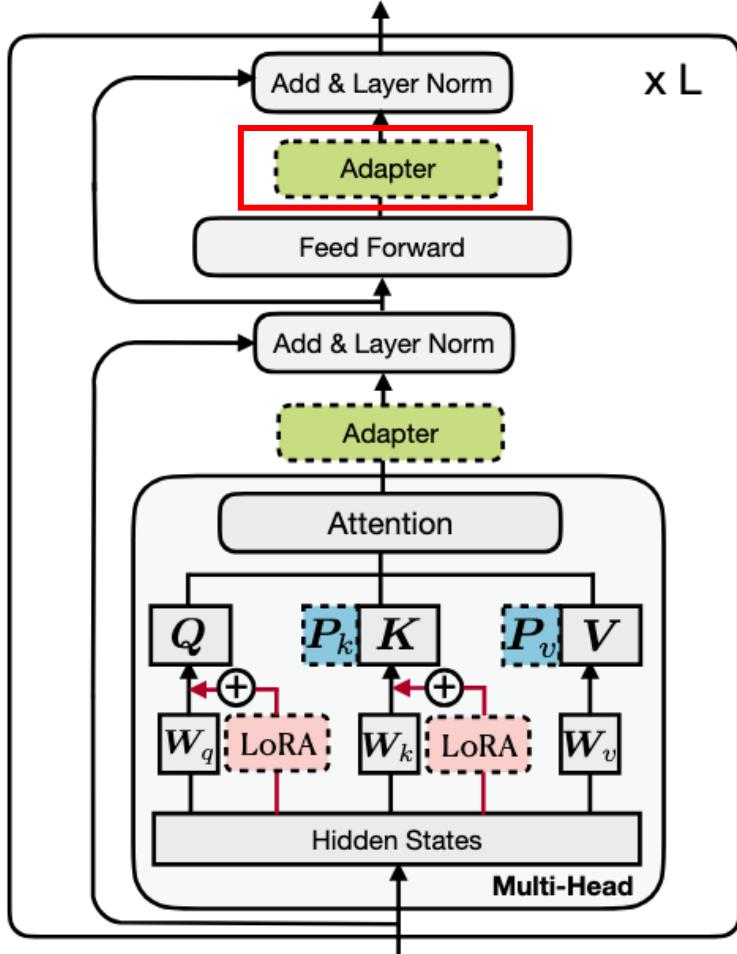
E.g., when exposed to a new task, new language, or any other new scope.

Problem! LLMs have huge networks. Us poor common folk cannot simply update a 10B parameter network.

# Parameter efficient fine-tuning



# Adding adapters

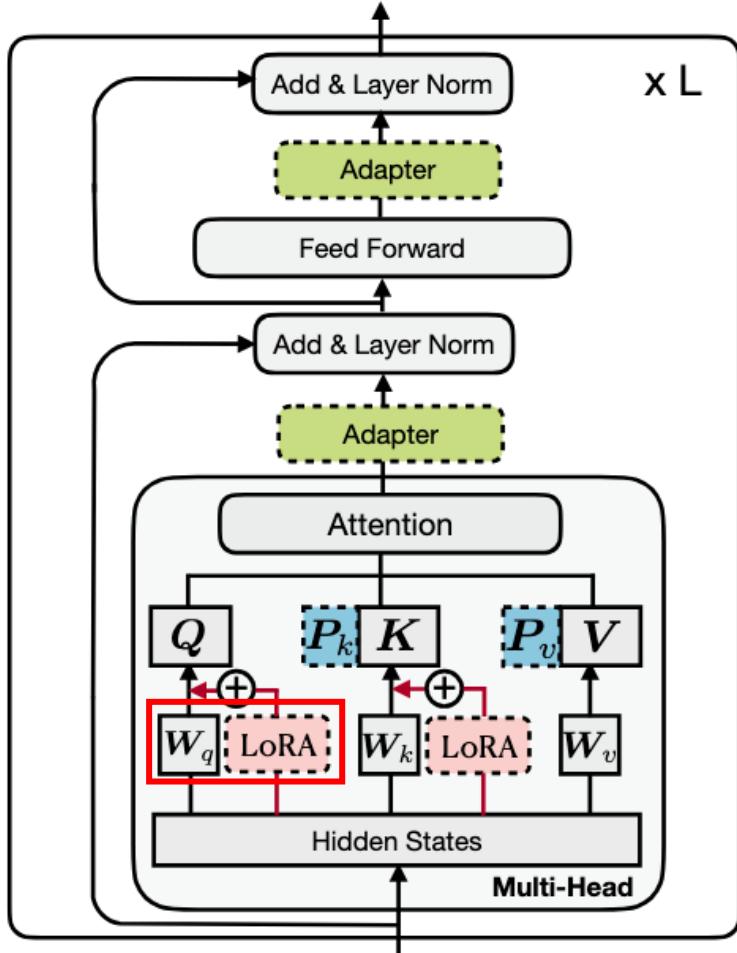


Let's add some small network blocks.

During fine-tuning, we keep the main architecture fixed and only tune the adapters.

Strong performance while only needing a full computation graph of a small subset.

# Adding LoRA

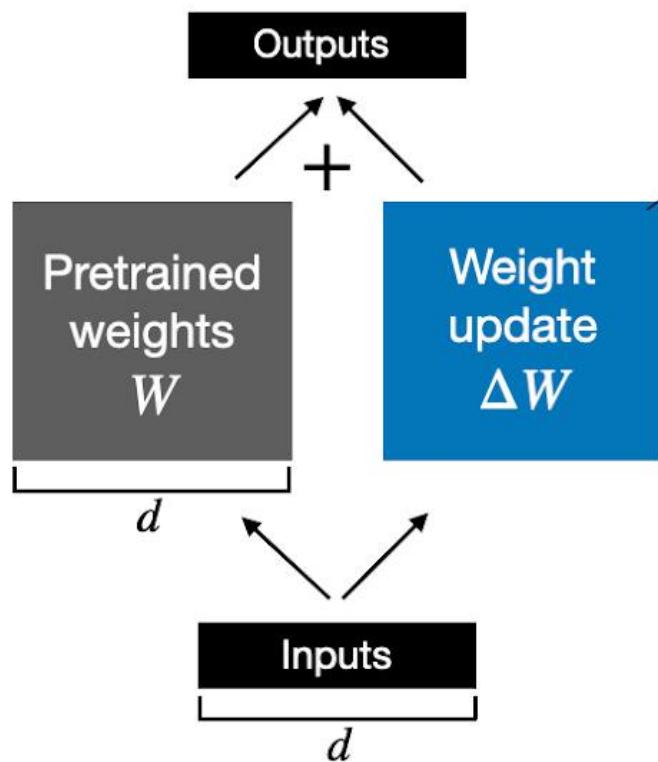


Most of the parameters of transformers are in the linear layers. Maybe these such matrices has a lower intrinsic dimensionality?

Can we learn a low-rank approximation of these huge matrices during fine-tuning?

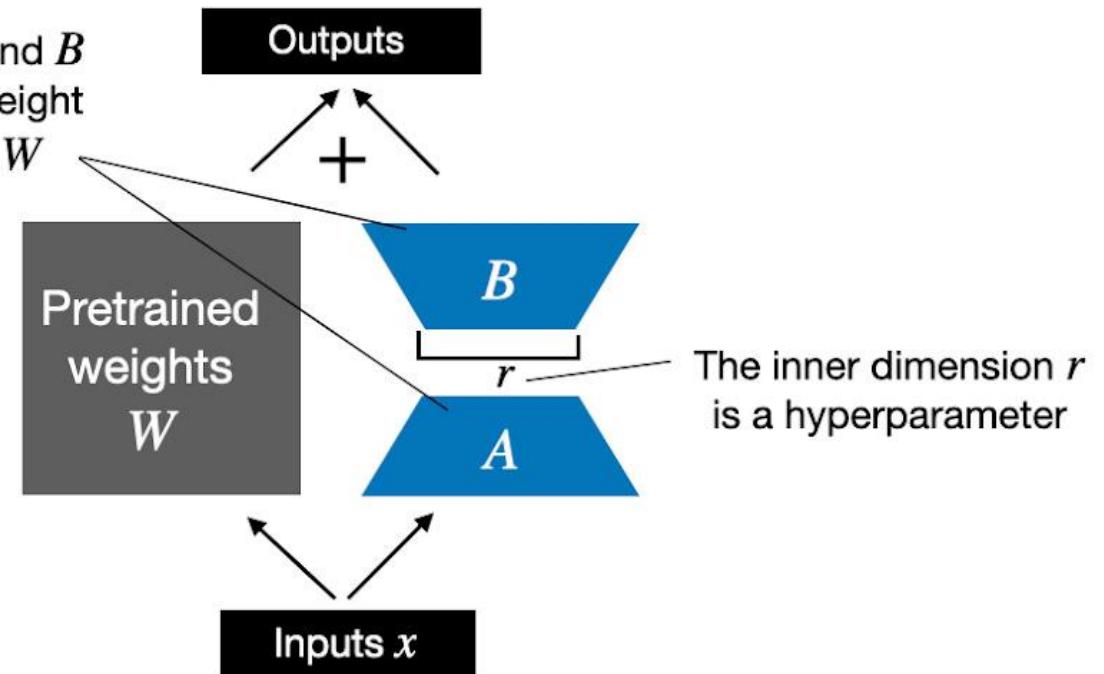
# LoRA

## Weight update in regular finetuning



LoRA matrices  $A$  and  $B$  approximate the weight update matrix  $\Delta W$

## Weight update in LoRA



The inner dimension  $r$  is a hyperparameter

# Looking beyond CLIP

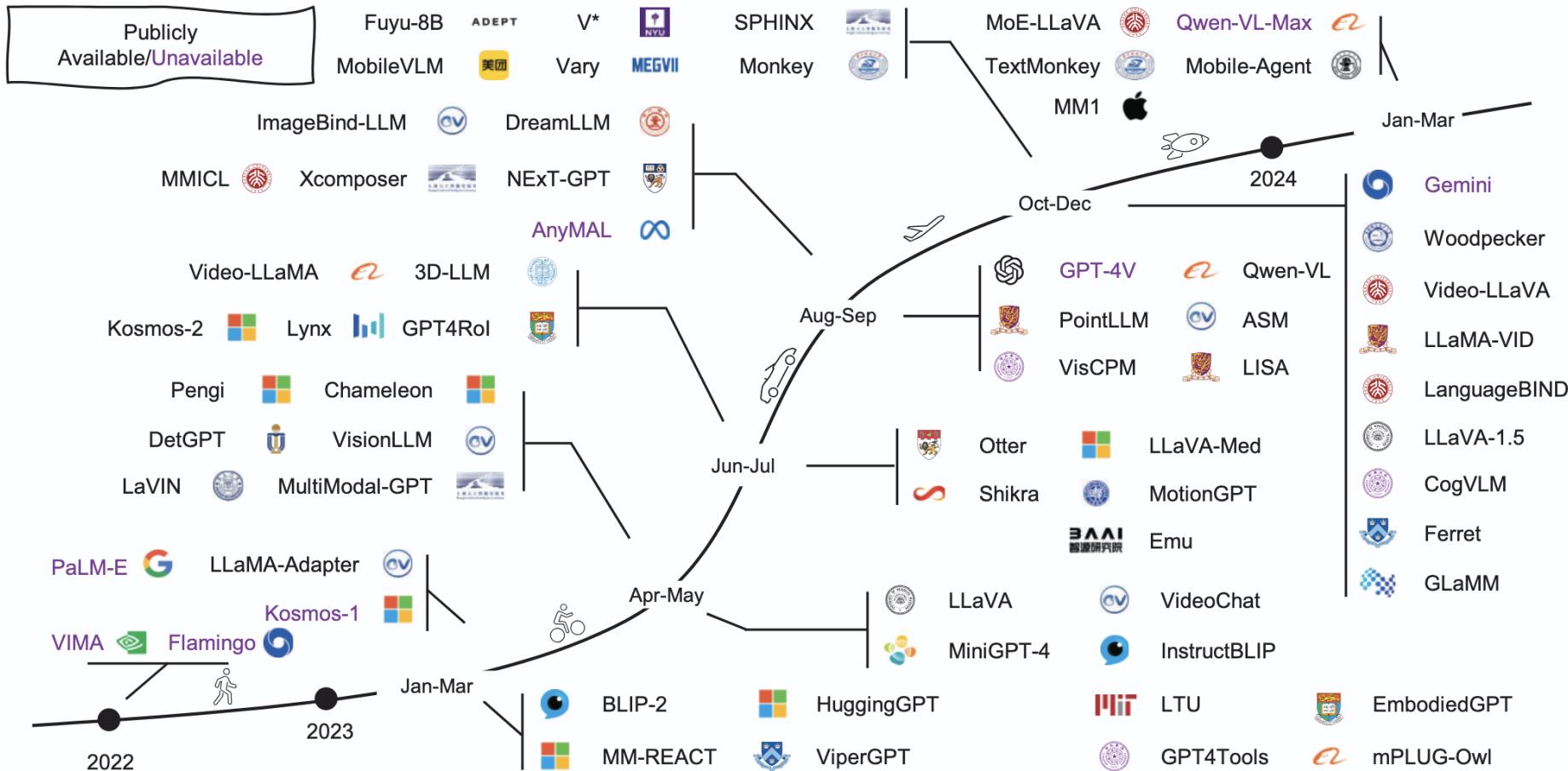
In CLIP, the text encoder is learnt from scratch, why not start from an LLM?

How to efficiently fine-tune VLMs?

What about other modalities?

CLIP is flat, but the real-world is hierarchical. How can we fix this?

# Multimodal Large Language Models



“A survey on multimodal large language models.” Yin et al. NSR 2024.

# Main idea of MLLMs

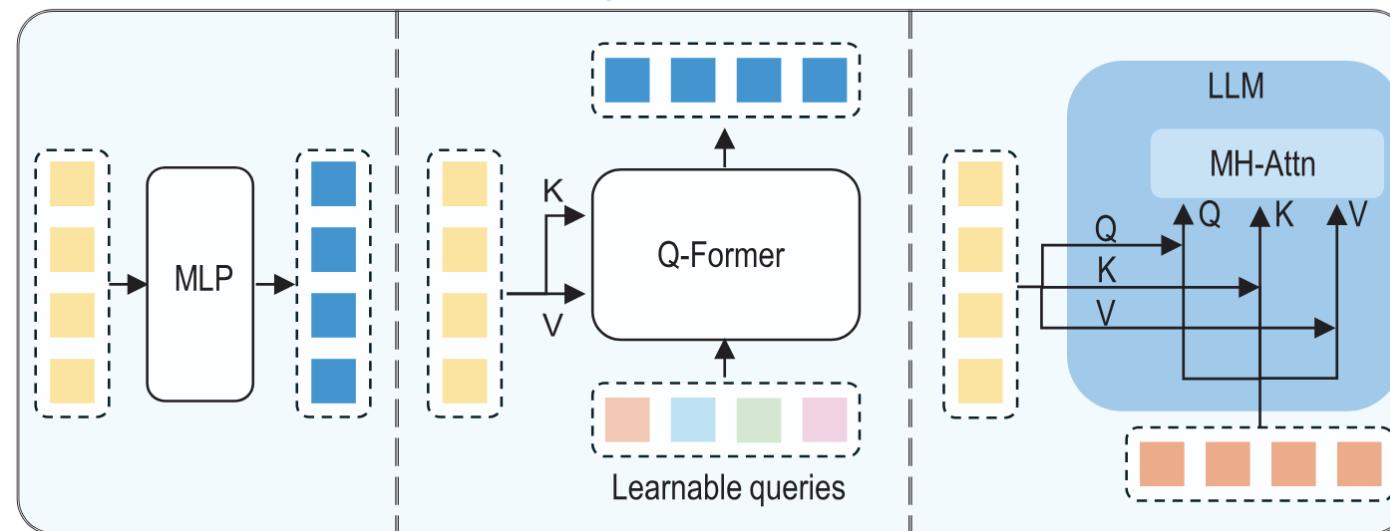
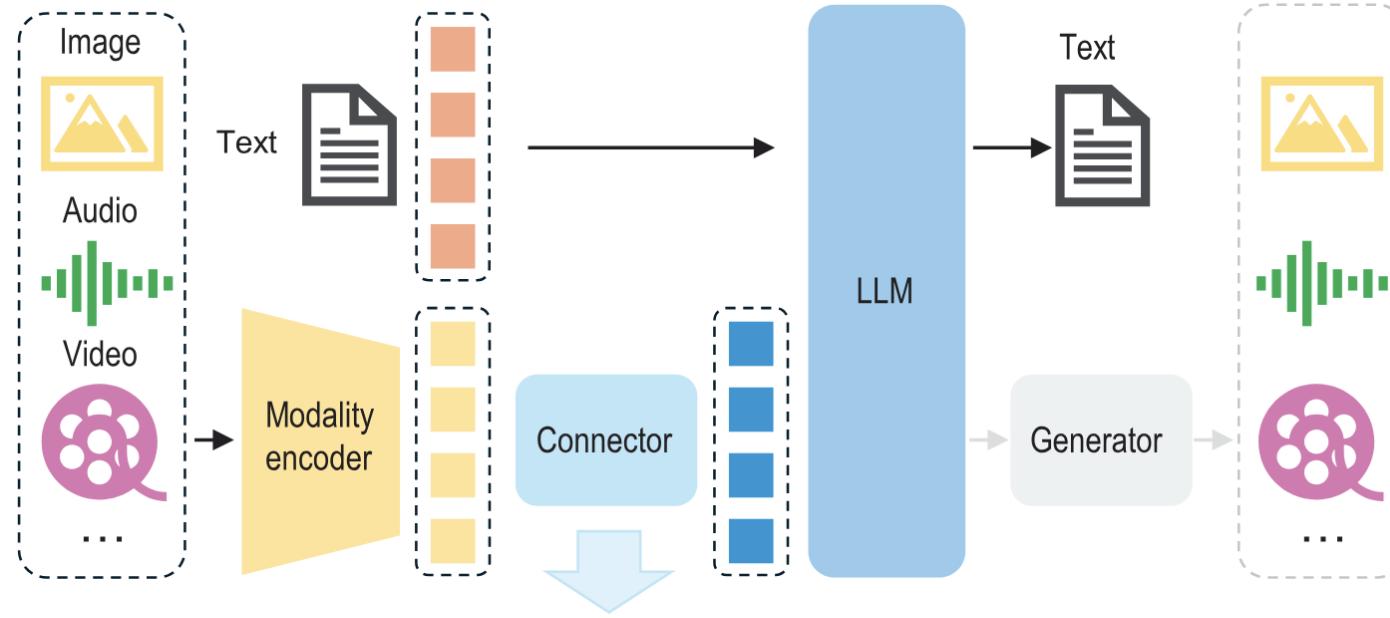
More is more: different modalities provide complimentary views.

For example, visually a bike is close to a motor and far from a leafblower.

In audio however, the motor and leafblower are close, far from a bike.

You can soon expect extensions of popular LLMs to video, audio, and more.

# Multimodal Large Language Model framework



# Case study 1: MLLMs and video popularity

Video-based MLLMs can deal with concrete tasks, such as object recognition.

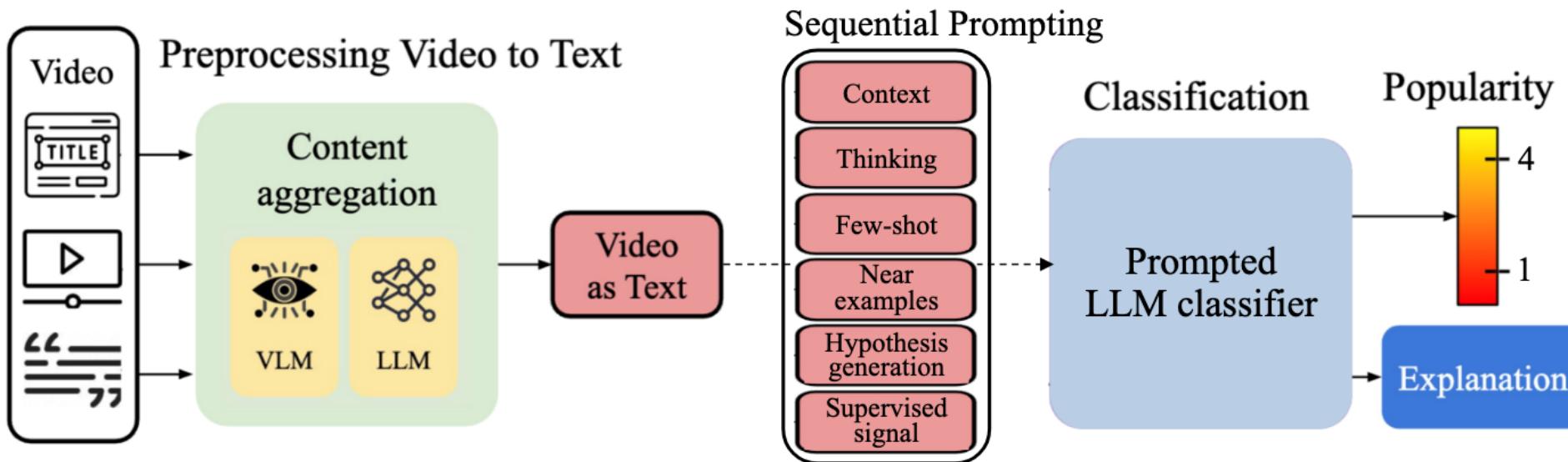
Video popularity is however cultural, social, and time-dependent.

Is it possible for an MLLM to predict whether a video will be popular, without fine-tuning the modal to this task?

“Large Language Models Are Natural Video Popularity Predictors.” Kayal et al. ACL 2025.

# Case study 1: MLLMs and video popularity

Main idea: bring all the multi-modal inputs to one modality, namely text.



# Case study 1: MLLMs and video popularity

Outcome: prompting an MLLM works better than training a modal specifically to solve the task at hand, with explainability as a bonus.

Video content



Title: Mexico vs. Brazil Highlights | International Friendly

Text summary

The video starts with Brazil attacking. Andreas Pereira scores a goal for Brazil after Mexico's Edson Alvarez dives in ....

retrieved videos



LLM hypothesis

1. highlight intense matches, finals, or close games
2. feature well-known or popular teams/players

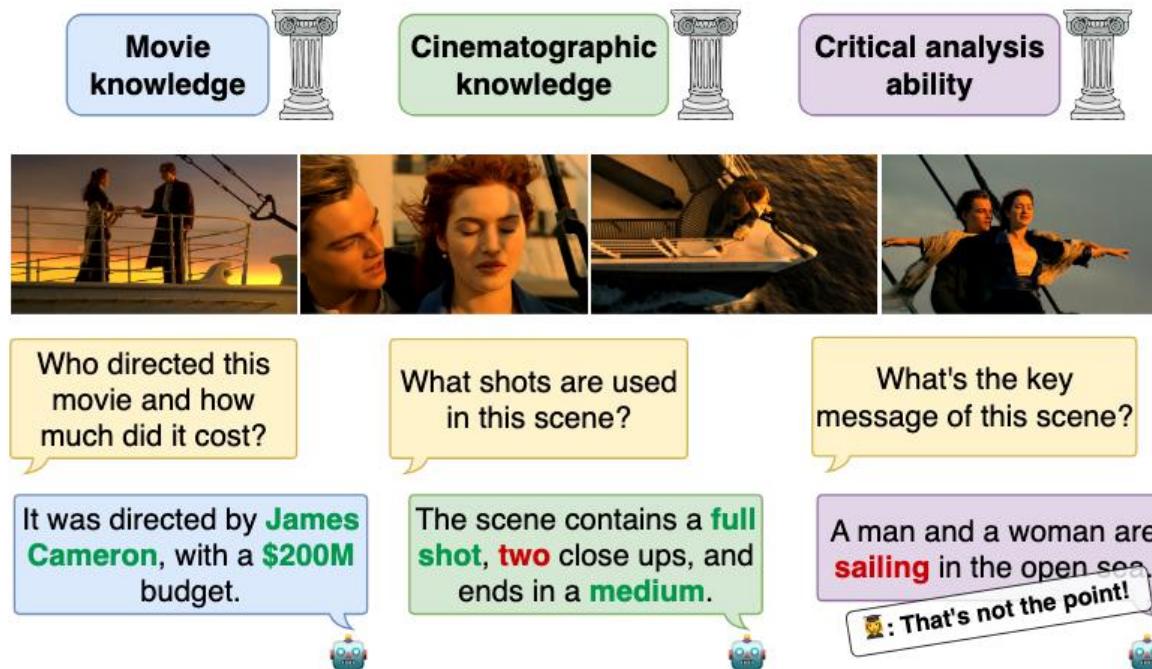
Prediction : 4 / 4

LLM explanation

1. The match is between Brazil and Mexico, Brazil being one of the biggest national teams in the world.
2. The video features several star players including Vinicius and Richarlison.

# Case study 2: MLLMs and movies

Are MLLMs Movie Buffs? No.



“Are Multimodal LMMs Movie Buffs.” Brett et al. under review.

# Case study 2: MLLMs and movies

MLLMs are decent at predicting meta-categories and basic cinematographics.



GT: Action, Adventure, Sci-Fi  
Pred: Action, Adventure, Sci-Fi



GT: 100M+ USD  
Pred: 10-50M USD



GT: Extreme close-up  
Pred: Extreme close-up



GT: Close-up  
Pred: Extreme close-up

But did they get this from viewing the visuals, or because the LLM scanned IMDb?

# Case study 2: MLLMs and movies

We can ask the MLLM to act as a film studies student. Then we can give its output to a film studies teacher to evaluate.



[...] The scene begins with a close-up of a woman lying on an operating table, wearing a hospital gown and an oxygen mask. The camera then pans out to reveal a group of surgeons dressed in surgical gowns, masks, and caps, preparing for the operation. The lighting is dim, with a blue hue dominating the scene, creating a clinical and sterile atmosphere. [...]



Incorrect, the first scene is a full shot of the whole group.

Accurate comment on the lighting!

No comments on the fact that she's flying and the surgeons turn into aliens?



The use of shadows and low lighting creates an atmosphere of suspense [...] The camera angles are carefully chosen to emphasize the power dynamics at play, with the man in the suit often positioned higher than the police officer. [...] The use of slow motion during key moments adds to the dramatic effect [...]



Comments on lighting and tone are accurate.

Good assessment of the power dynamics based on positioning!

There is no slow motion.



Conclusion: MLLMs don't really use the visual modality and hallucinate everything.

# Looking beyond CLIP

In CLIP, the text encoder is learnt from scratch, why not start from an LLM?

How to efficiently fine-tune VLMs?

What about other modalities?

CLIP is flat, but the real-world is hierarchical. How can we fix this?

# Next lecture

Lecture	Title	Lecture	Title
1	Intro and history of deep learning	2	AutoDiff
3	Deep learning optimization I	4	Deep learning optimization II
5	Convolutional deep learning	6	Attention-based deep learning
7	Graph deep learning	8	From supervised to unsupervised deep learning
9	Multi-modal deep learning	10	Generative deep learning
11	What doesn't work in deep learning	12	Non-Euclidean deep learning
13	Q&A	14	Deep learning for videos

# Learning and reflection

Multi-modal deep learning not covered in Understanding Deep Learning book.

Thank you