

Topics:

- Early VLMs

# **CS 8803-VLM**

## **ZSOLT KIRA**

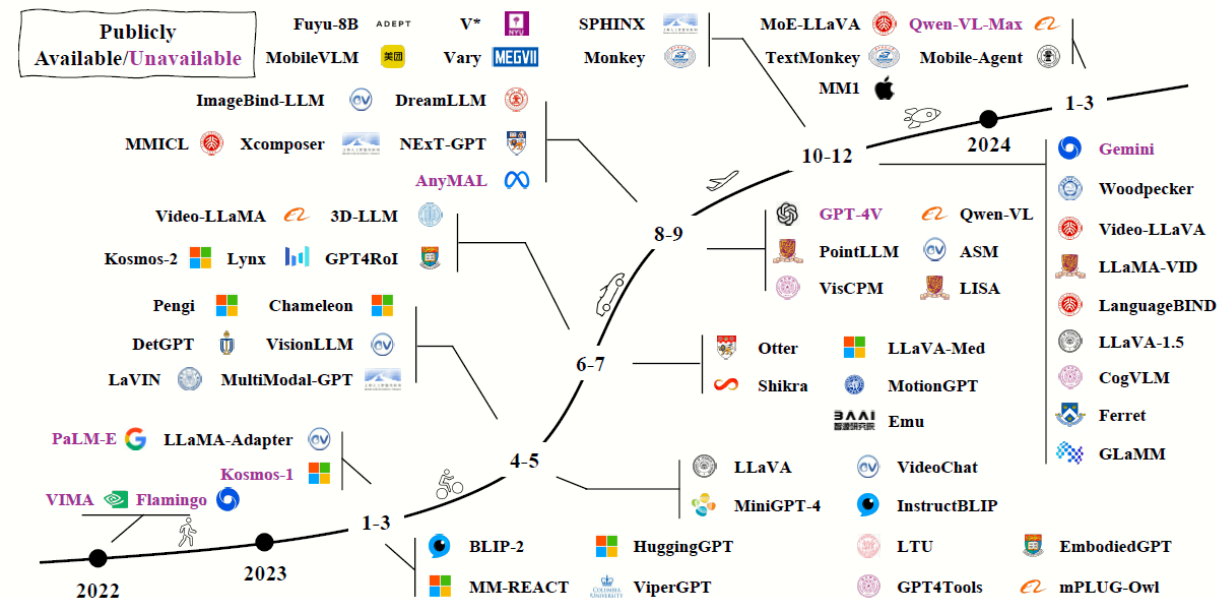
- Schedule out, please sign up if you haven't done so by Friday!
  - Sessions are around “topics” so feel free to suggest additions to paper list for that topic.
  - First job is to define main paper (that will be reviewed) and will make up majority of presentation details
  - Other auxiliary/context papers should be covered more concisely (e.g. a few slides)
- Reminder:
  - **15% Class Participation – Attendance and participation in-class/Ed**
  - **20% Paper Reviews** – Due night before paper presentation
    - **First one due Monday Aug 2 11:59pm**
  - **15% Paper Discussion - Presentation**
  - **50% Project**
    - Pick teams by Sept 11
    - Proposal (~5 min presentation) to be presented Sept 18
- All deliverables to be submitted on Canvas, see instructions/rubrics there

Pre-Trained  
Vision Model

?

Pre-Trained  
Language  
Model

- Several different flavors of multi-modal models
  - Training of aligned encoders (CLIP)
  - Text-conditioned image generation (flow-based, diffusion, GANs, VAEs)
  - Image-conditioned text generation (captioning, vision-question answering)
    - Specialised captioning/VQA models (more engineered encoders, e.g. object detection, etc)
  - Full-fledged any-input (text or image) -> language generation
- My hope: Any to any models!



# CLIP: Connecting text and images

[Read paper ↗](#)

[View code ↗](#)

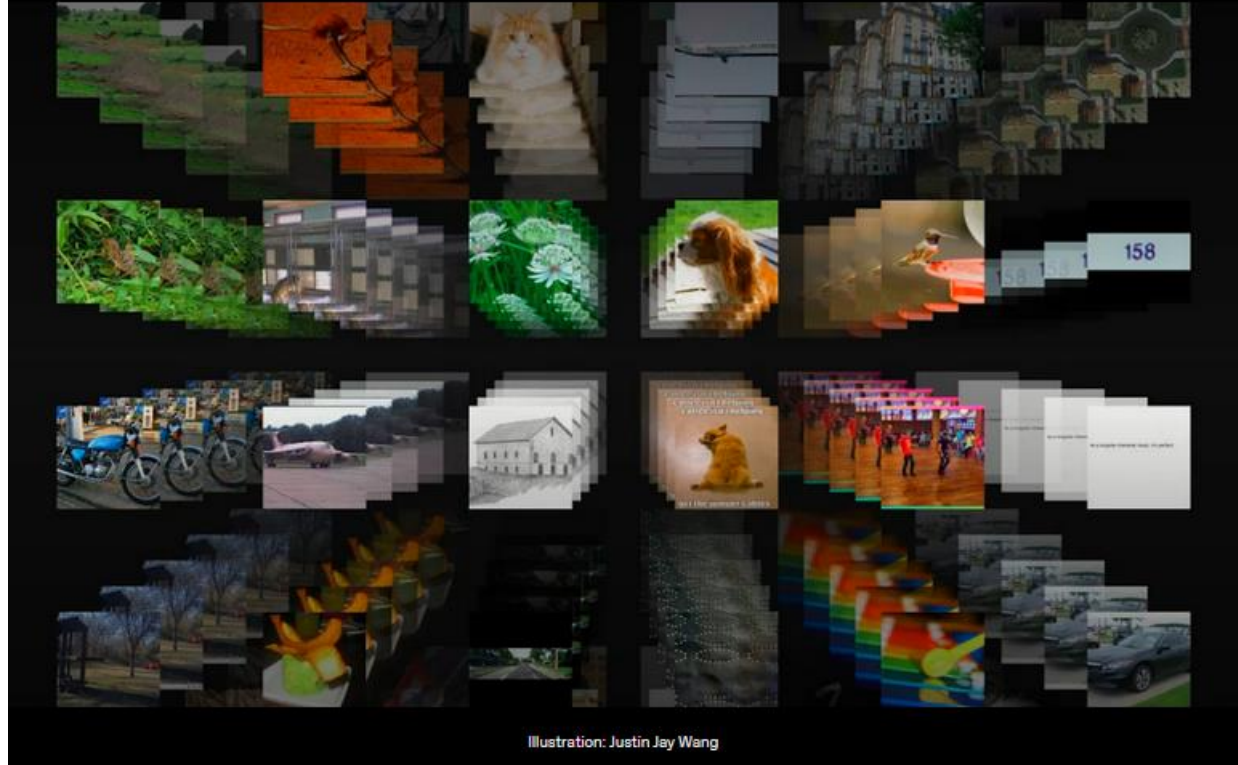


Illustration: Justin Jay Wang

## Learning Transferable Visual Models From Natural Language Supervision

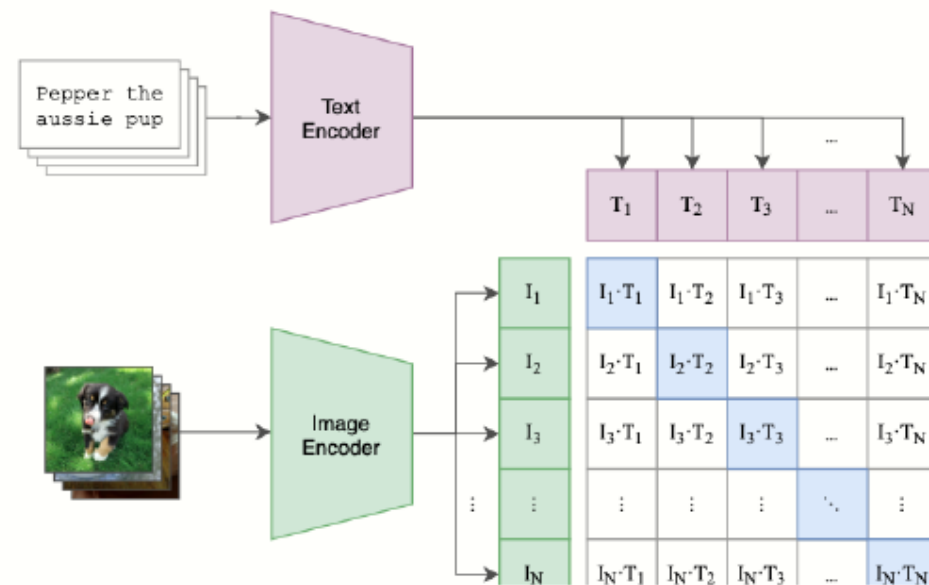
[Alec Radford](#), [Jong Wook Kim](#), [Chris Hallacy](#), [Aditya Ramesh](#), [Gabriel Goh](#),  
[Sandhini Agarwal](#), [Girish Sastry](#), [Amanda Aspell](#), [Pamela Mishkin](#), [Jack](#)  
[Clark](#), [Gretchen Krueger](#), [Ilya Sutskever](#)

<https://openai.com/index/clip/>

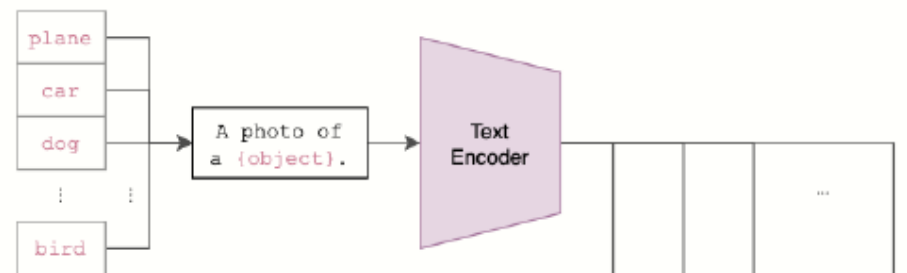
# Approach

- What is CLIP? Contrastive Language-Image Pre-training
- 400M (image, text) pairs collected from various internet sources
- Image encoder piece: Modified ResNet or Vision Transformer (ViT)
  - Picked based on performance
- Text encoder: Transformer with 63M parameters

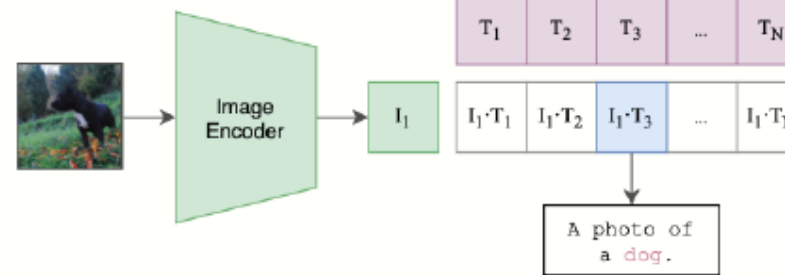
(1) Contrastive pre-training



(2) Create dataset classifier from label text

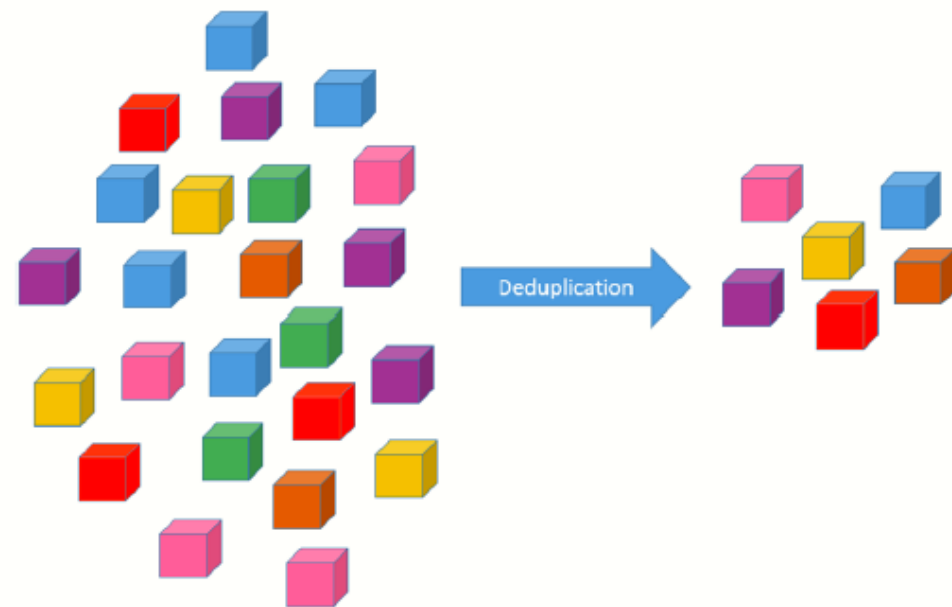


(3) Use for zero-shot prediction



# Approach (Data Collection)

- Raw web pairs aren't going to be perfect
  - Plenty of noise and even mismatches, abstract pairs
  - Either way → CLIP gets stronger with weird stuff
- CLIP filtering
  - 500,000 unique internet queries to cover all domains
  - Pulled in captions, descriptions, comments any kind of data paired with images
  - 1 query could produce max most relevant 20k image-text pairs, ensuring diversity
- De-duplication
  - Image text pairs underwent de-duplication which just ensures overlap is minimal
  - Each sample should ideally be unique
  - Also lowers overlap with benchmarking datasets, → real evaluation and generalization capabilities





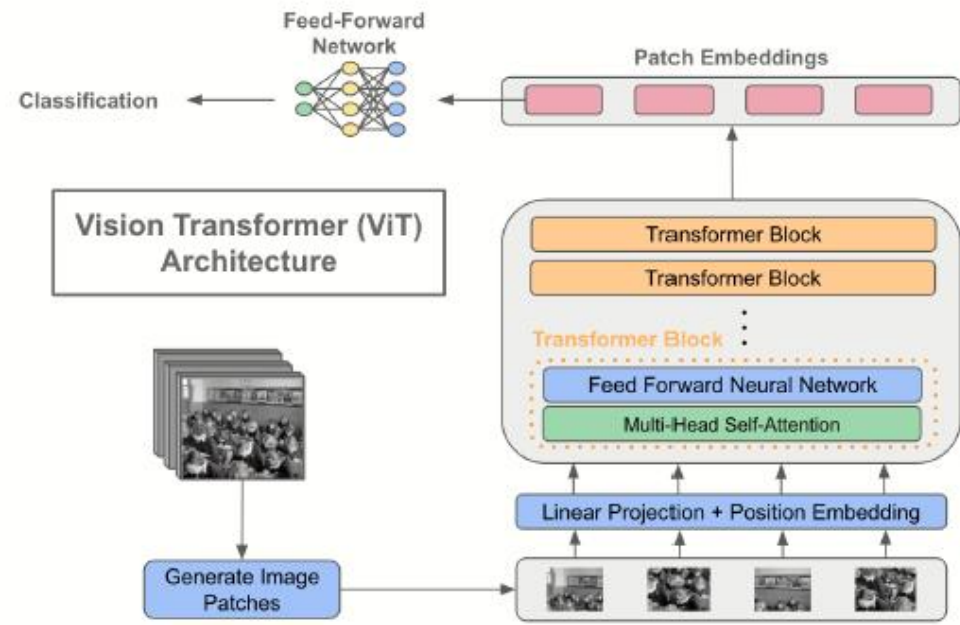
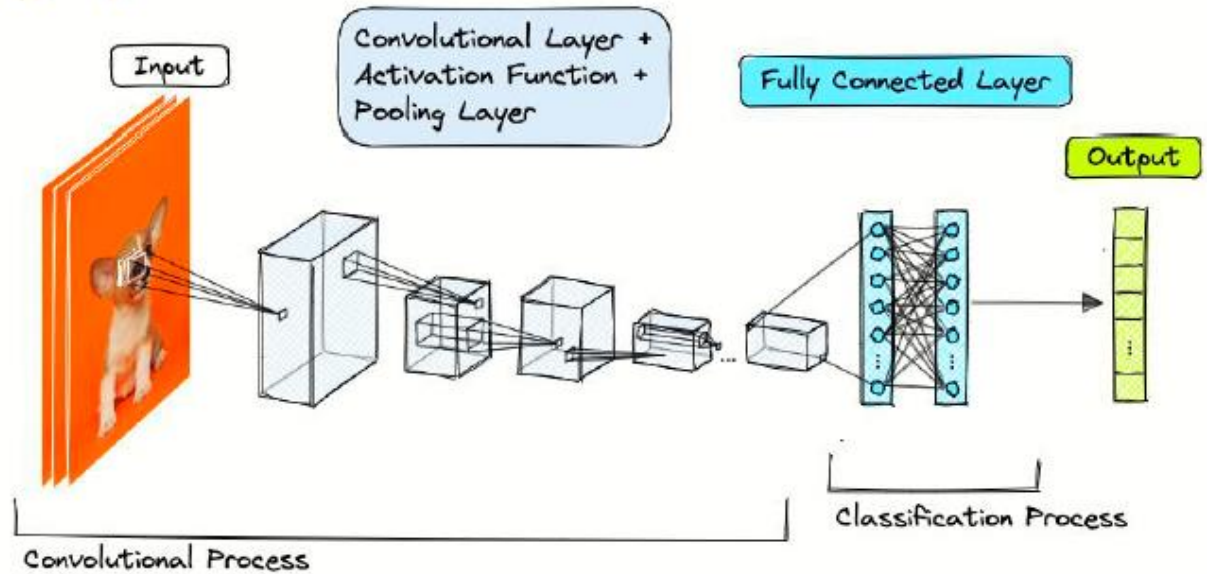
# Approach (Image Encodings)

- ResNet encoder

- CNN architecture, conv layers + pooling → feature vector
- Linear layer for final embedding, L2 normed for ease of similarity

- ViT encoder

- Patches over image, flattened and projected into embedding (like with text)
- Positional encodings for those patches, multi-head self attention + feedforward neural nets are strong
- A classification token is added onto the patch embeddings sequence, then normalized too



```

# 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

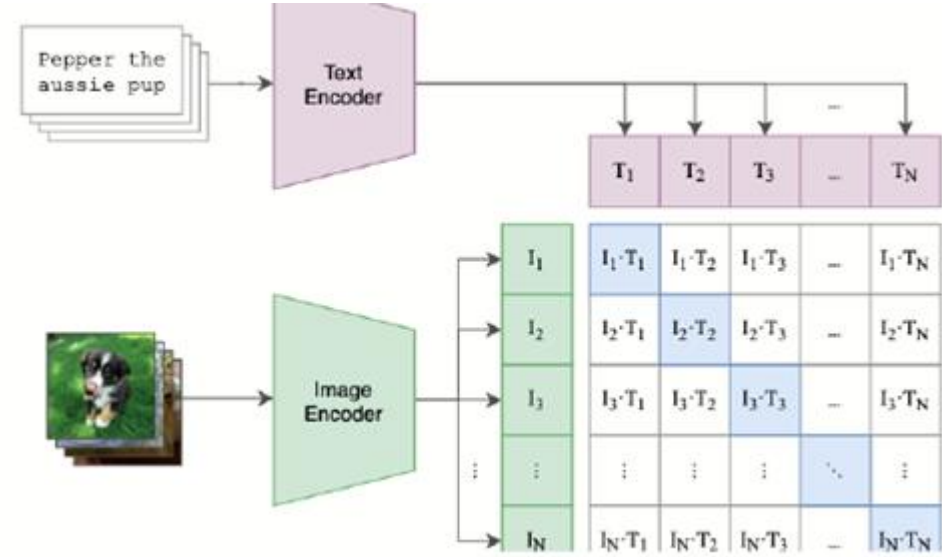
# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# 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)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# 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

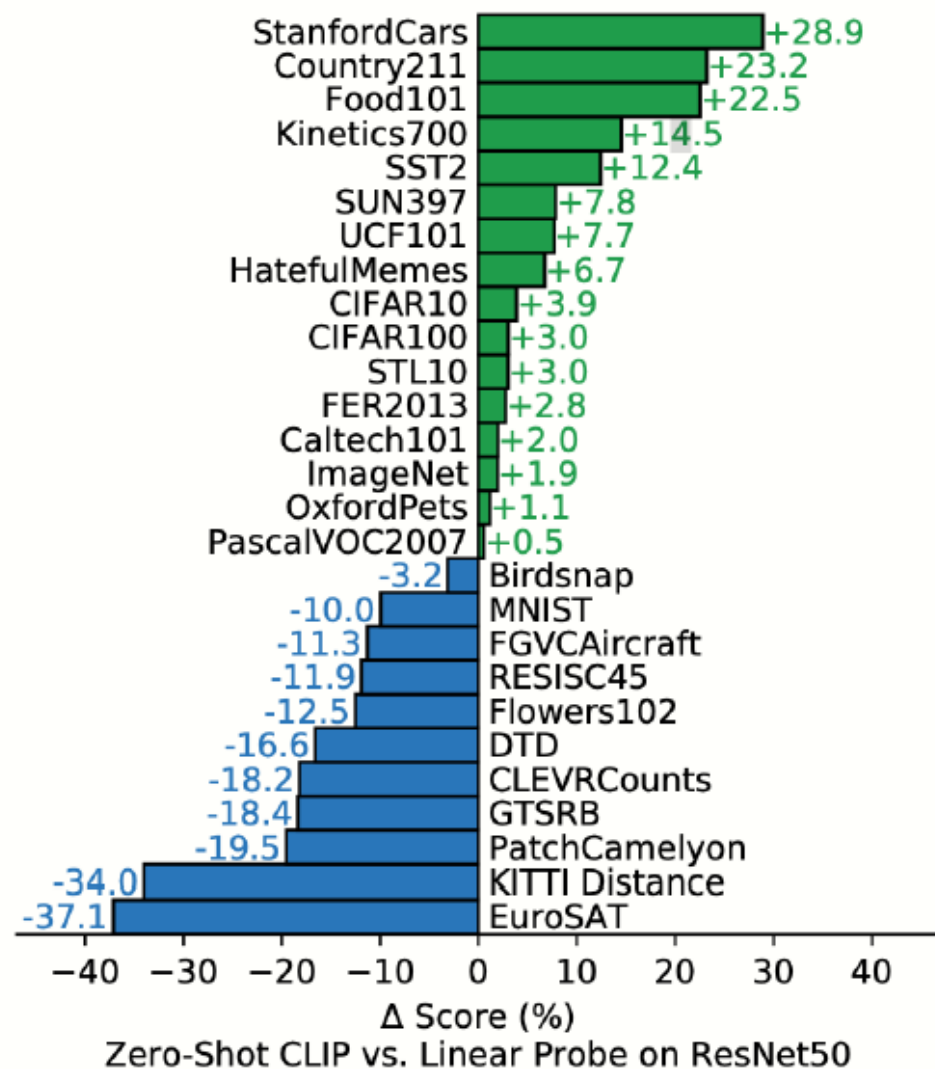
```





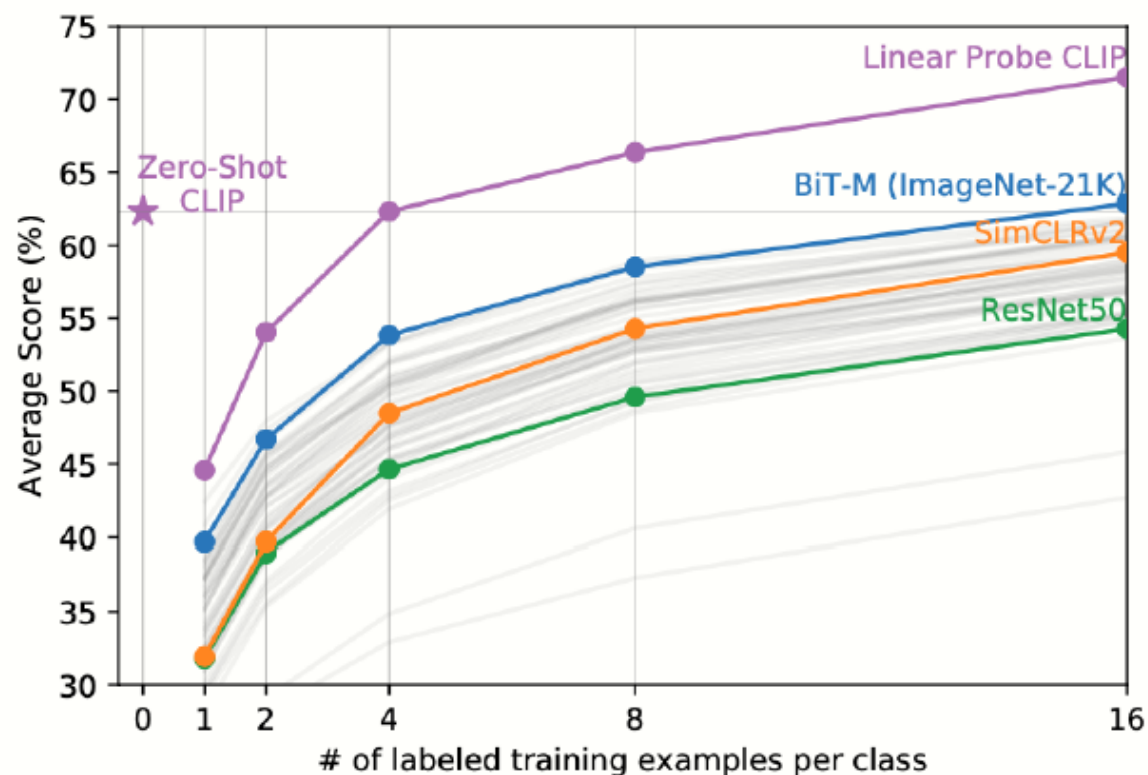
# Experiments and Results (Linear Probe)

- Linear probe is a simple classifier (log reg) added to pre-trained features some labeled data
  - Beat logistic regression on ResNet50 features on 16/27 datasets – multimodal training power
  - Significance? ROBUST, no task-specific data or fine-tuning needed
  - Particularly good at general object recognition Food101, StandfordCars
  - Specific context-based understanding like EuroSAT and Satellite Imagery give CLIP more trouble



# Experiments and Results (Few-Shot)

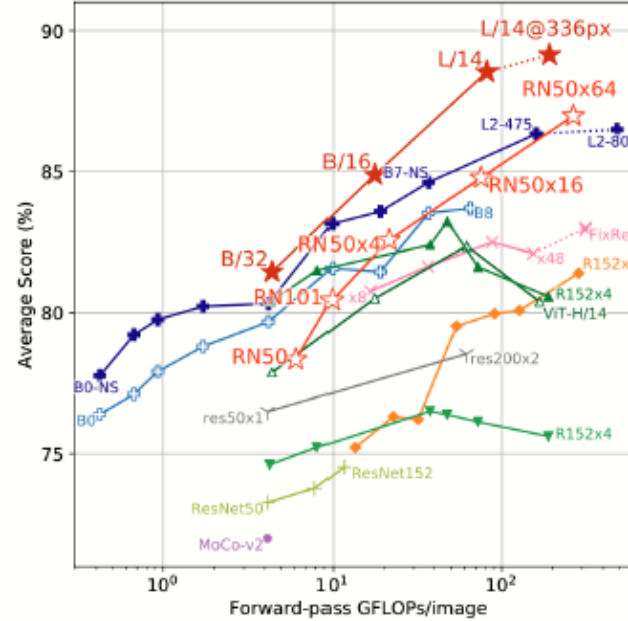
- Few-shot learning is training on a couple of examples per class
- Outperforms 16-shot classifiers using features from other models
  - Embeddings learned by CLIP capture a plenty of transferable knowledge and can generalize to out of domain concepts



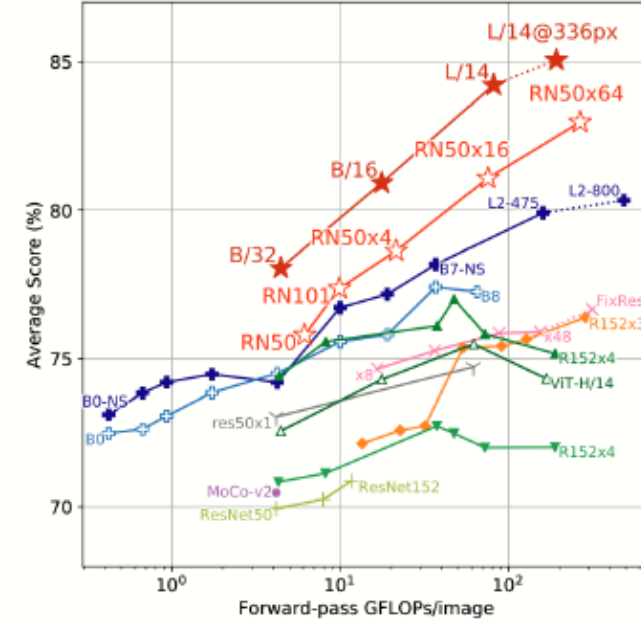
# Experiments and Results (Scaling)

- Scaling:
  - ViT's scale well with compute + data
    - ResNets... not so much
  - Learned representations that are not just specific to one type of data
  - Largest CLIP model (ViT-L/14@336px) outperforms existing models by a significant margin
  - 2.6% average improvement
    - CLIP benefits from larger models but strong architectures too which better capture complex relationships

Linear probe average over Kornblith et al.'s 12 datasets



Linear probe average over all 27 datasets



# Strengths, Weaknesses, Relationships (including limitations)

- Pros:

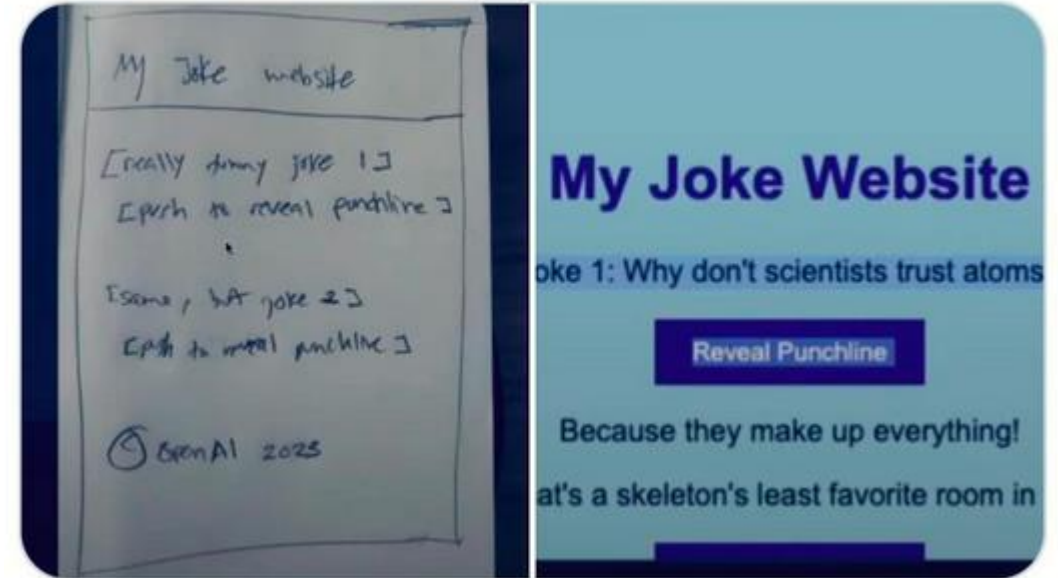
- Pre-trained robust zero-shot learning with encoder-backed supervision
- Adaptable to distribution shifts
- Scales well with compute
- Many tasks learned and classifying classes without explicit supervision

- Cons:

- Not SOTA on all tasks Satellite Imaging, (EuroSAT, RESISC45) etc
- 1000x more compute to reach SOTA?!
- “Prompt engineering” effects, like adding “child” to categories list reduced misclassification of young people into incorrect categories from 32.3% to 8.7%
- Societal issues – surveillance/privacy
- Data Overlap (3.2% testing dataset avg)

# Leveraging Large Language Models

- MLLMs: Combining LLMs with other modalities (e.g., vision)
  - Key capabilities:
  - Writing stories from images
  - OCR-free math reasoning
  - Decision-making
  - Embodied/Robotics
  - Rapid development since 2022
- Some concepts for today:
  - How can we leverage pre-trained MLLMs (early proof of concepts – Frozen)
  - How should modalities interact? (BLIP-2: Q-former, Flamingo: Perceiver-I/O and cross-attention)
  - Can we support interleaved **inputs** and hence **few-shot prompting (in-context learning)**?



GPT4-V Demo



# VL-Bert and ViLBert - 2019

- Despite their names, both papers were developed independently.
- VL-Bert: Bert backbone remains the same, extracts image ROIs with Faster RCNN for fine-grained features
- ViLBert: Model consists of two parallel BERT models (one for images, one for text) which mix information in co-attention blocks
- Both models performed well on multimodal tasks (VQA, captioning) but were task-specific.

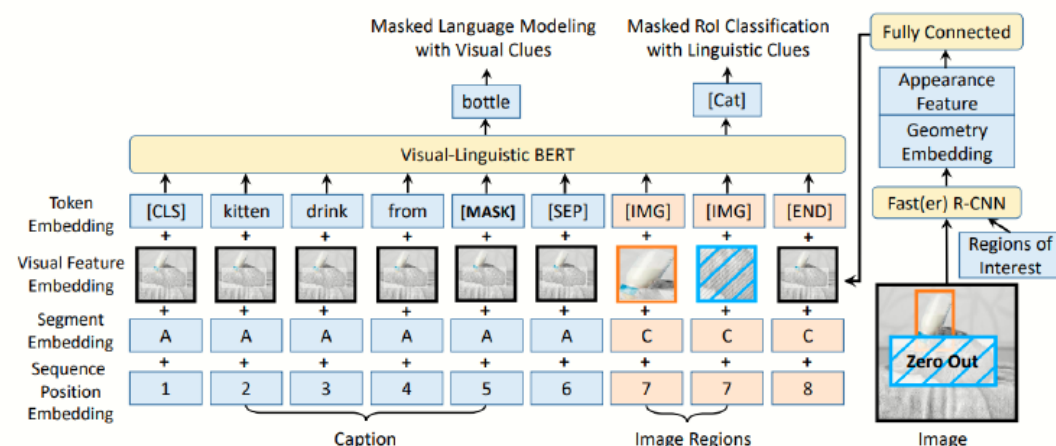
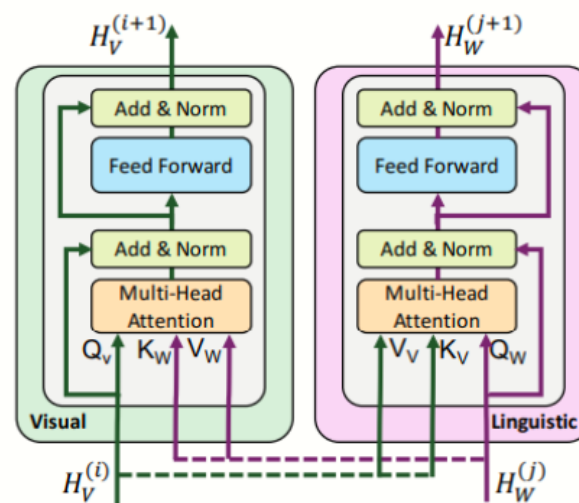
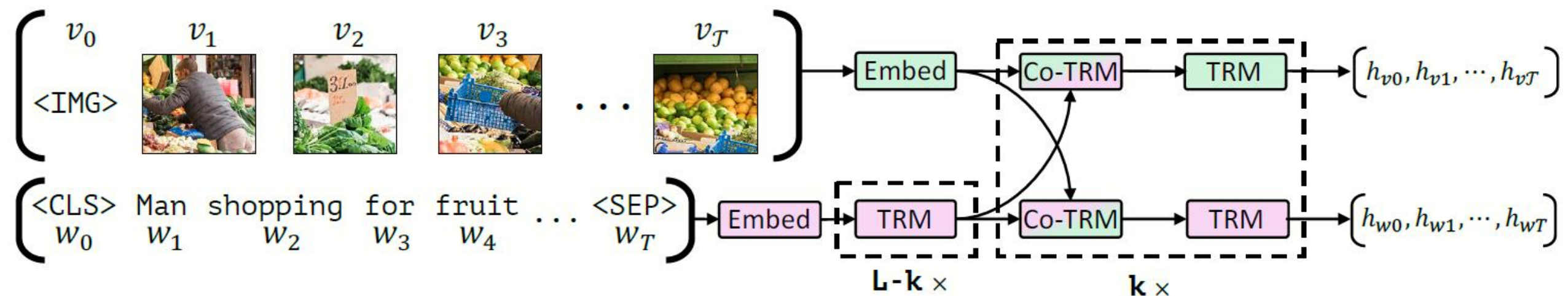


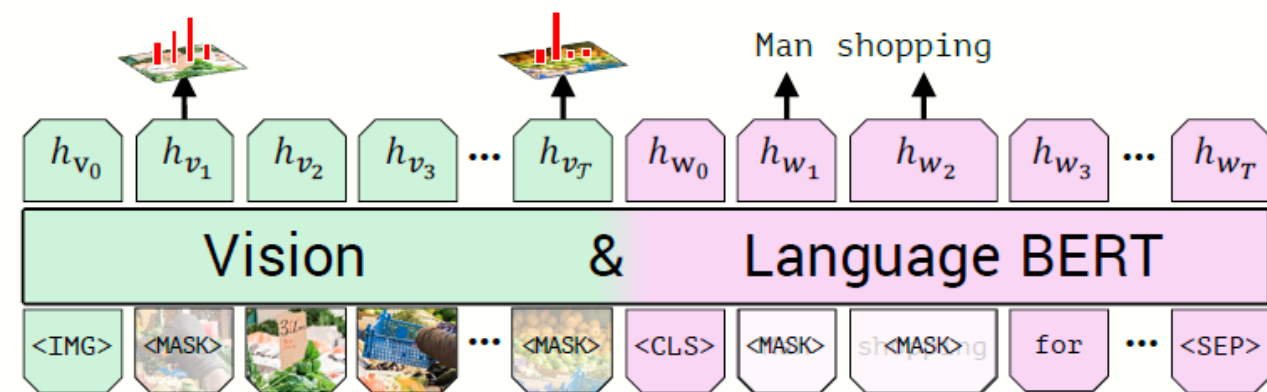
Figure 1: Architecture for pre-training VL-BERT. All the parameters in this architecture including VL-BERT and Fast R-CNN are jointly trained in both pre-training and fine-tuning phases.



Left: co-attention block of ViLBert

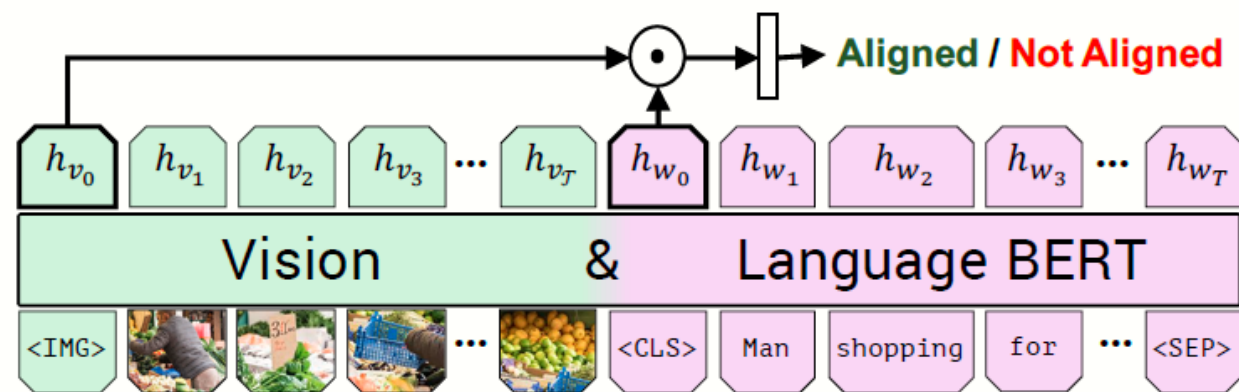


## Training: Masked Prediction + Alignment



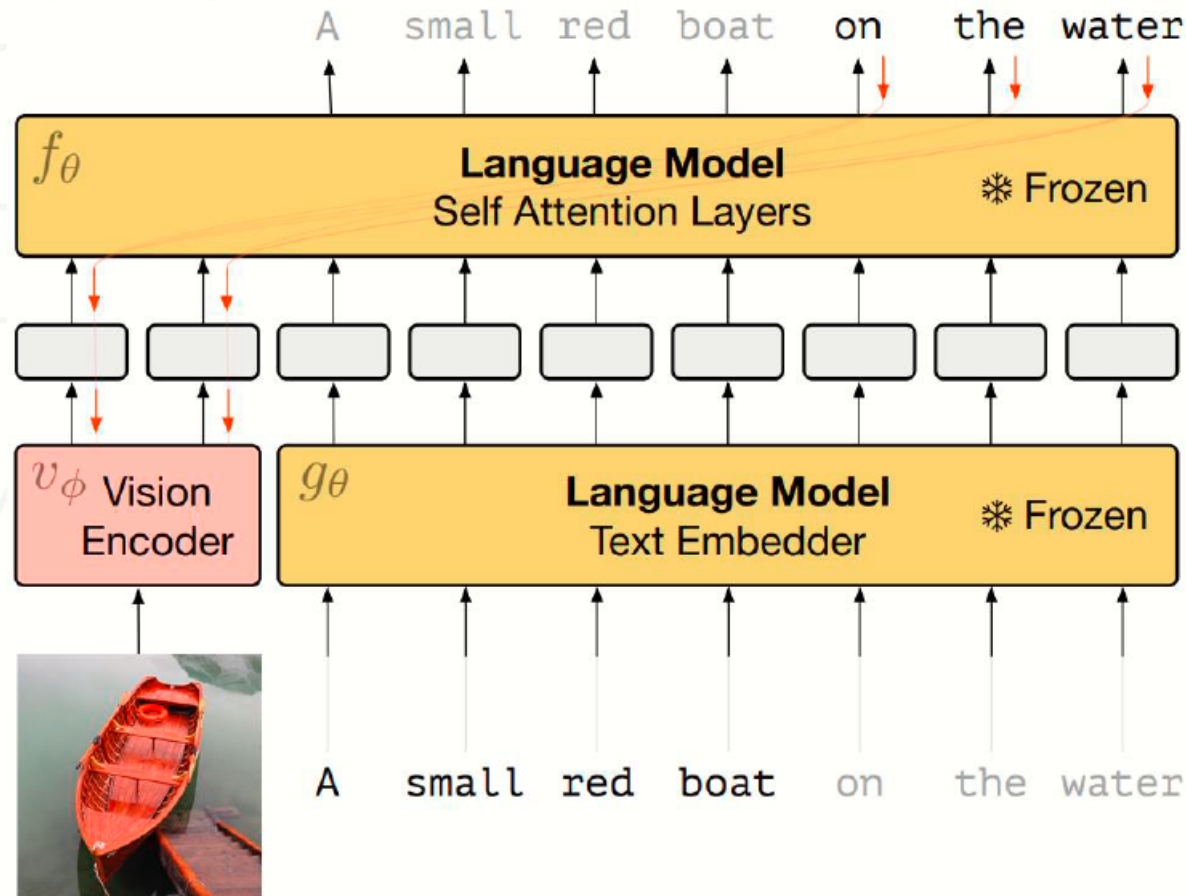
(a) Masked multi-modal learning

## Interaction/Fusion: Cross-Attention



(b) Multi-modal alignment prediction

## Frozen: Approach



- fine-tuning  $\theta$  hurts generalization (because the LM datasets size  $\gg$  text/image coupled datasets)
- Modularity : plug-n-play any LLM !
- Proof on concept : small scale (7B model), but enough to show interesting properties for few-shot
- Training objective : for only one image ! But at inference multiple images supported (thanks to relative pos. enc.)

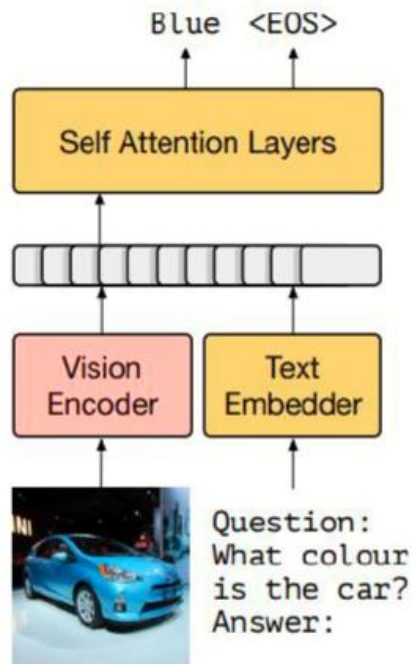
A simple architecture : a completely frozen LLM, conversion of the image w/ Resnet into 2 tokens (~prefix tuning). Gradient flows through LLM

# Architectures

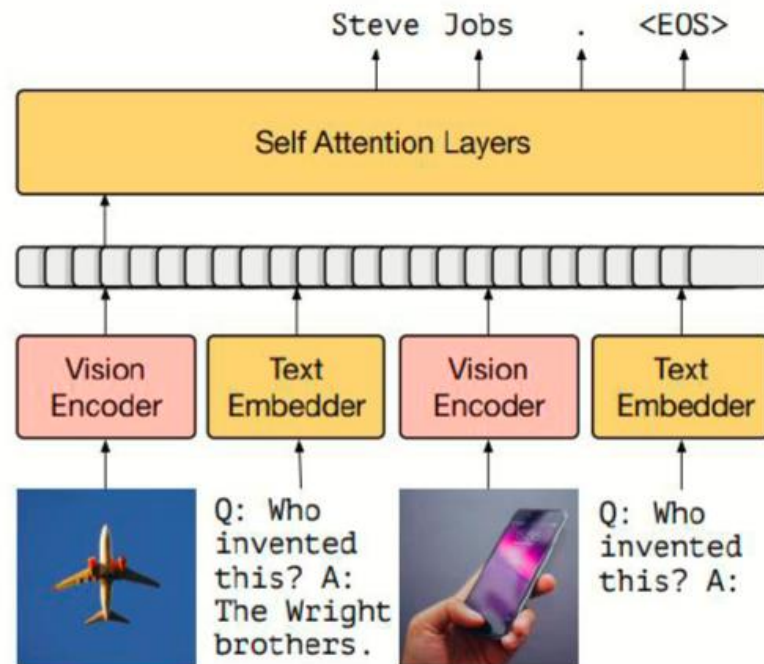
- Language model: GPT-like ~7B model trained on C4
  - C4: Cleaned up version of common crawl by Google/Meta
- Vision Model: NF-ResNet-50 architecture



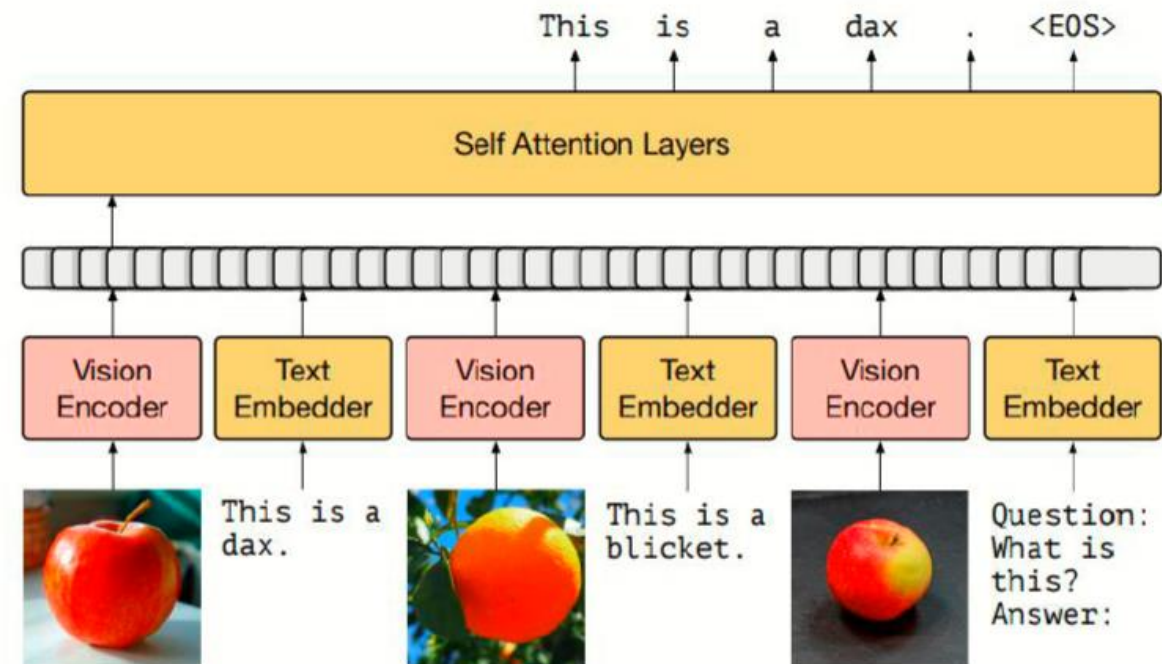
# Frozen: Supports Interleaved Inputs!



(a) 0-shot VQA



(b) 1-shot outside-knowledge VQA



(c) Few-shot image classification

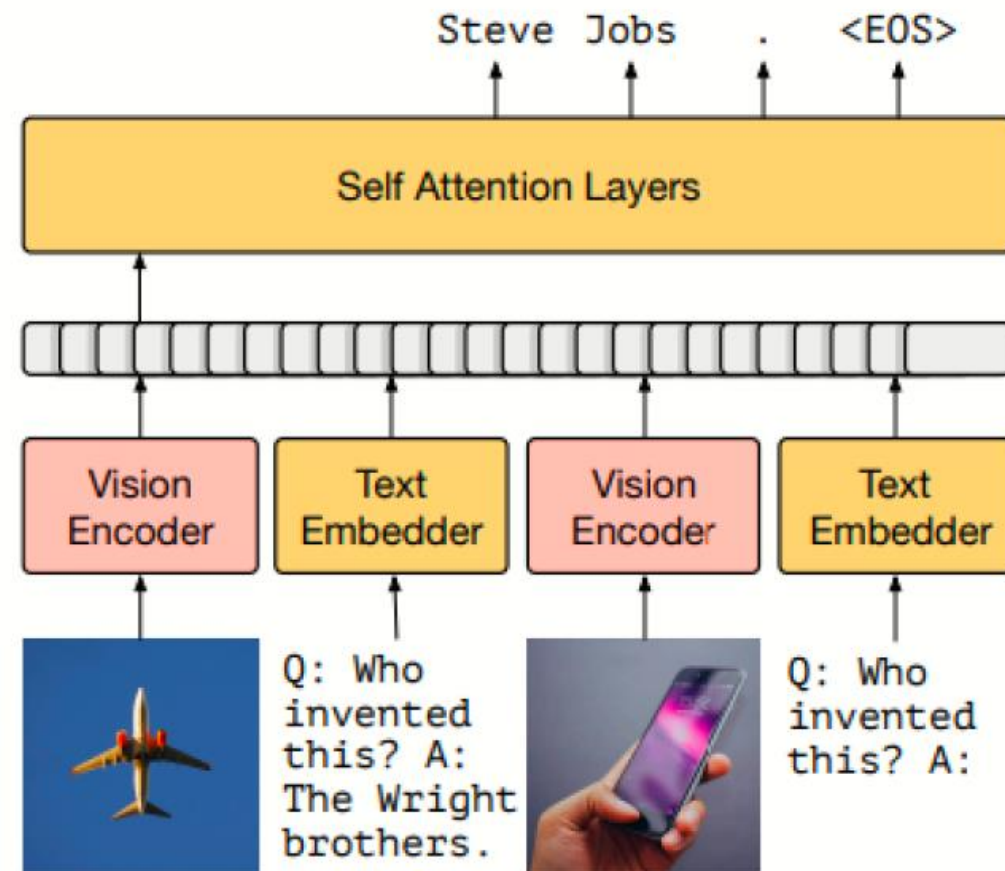
Possible thanks to Position encodings !



# Frozen: Supports Few-Shot Learning!

n-shot Acc.	n=0	n=1	n=4	$\tau$
<i>Frozen</i>	5.9	9.7	12.6	$\times$
<i>Frozen</i> 400mLM	4.0	5.9	6.6	$\times$
<i>Frozen</i> finetuned	4.2	4.1	4.6	$\times$
<i>Frozen</i> train-blind	3.3	7.2	0.0	$\times$
<i>Frozen</i> VQA	19.6	—	—	$\times$
<i>Frozen</i> VQA-blind	12.5	—	—	$\times$
MAVE <sub>x</sub> [42]	39.4	—	—	✓

Table 2: Transfer from Conceptual Captions to OKVQA. The  $\tau$  column indicates if a model uses training data from the OKVQA training set. *Frozen* does not train on VQAv2 except in the baseline row, and it never trains on OKVQA.



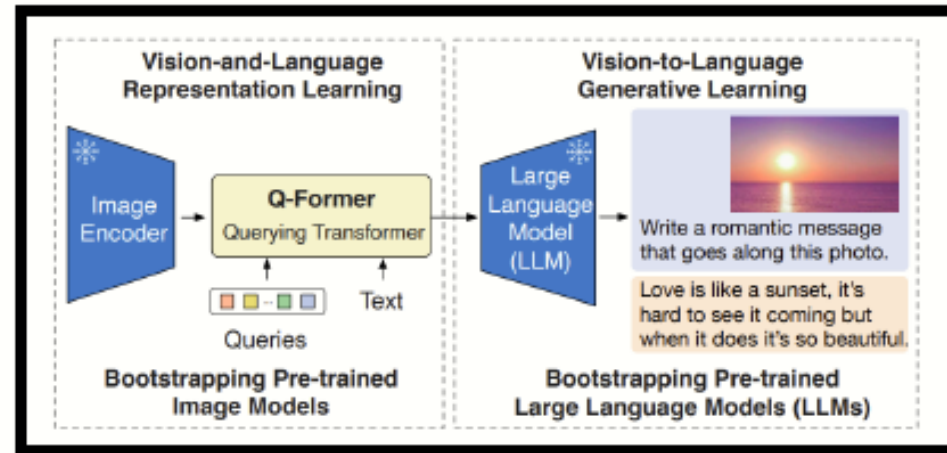
(b) 1-shot outside-knowledge VQA

# Limitations & Conclusion

- Far from SOTA performance : proof of concept for new area of research "few shot multi-modal learning"
- Unexplored question : Could performance be improved with more elaborate architectures for mixing vision and language ?
- System only trained for captioning, but capable of open-ended interpretation of images and multimodal few-shot learning



# • BLIP2



Language Model

Connection Module

Vision Encoder

Pre-trained: FLAN-T5/OPT

Q-Former: Lightweight  
Querying Transformer

Contrastive pre-trained:  
EVA/CLIP

# Problems With Current Vision-Language Pretraining

- No unified architecture for multi-task vision-language pre-training
  - Encoder only models
    - CLIP, ALBEF
    - Not directly applicable to text generation tasks
  - Encoder-Decoder models
    - VL-T5, SimVLM
    - Can't perform image-text retrieval
- Noisy image captions are suboptimal for vision-language pretraining

BLIP

- High computational costs during pre-training
- Pre-trained encoders experience catastrophic forgetting

BLIP 2

Diffusion Transformer

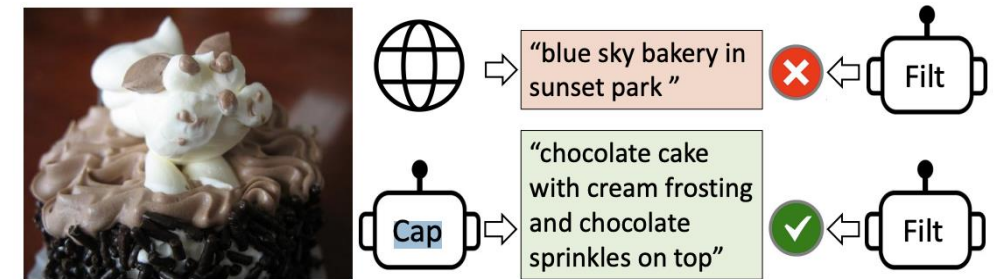
BLIP 3



# Proposed Solutions

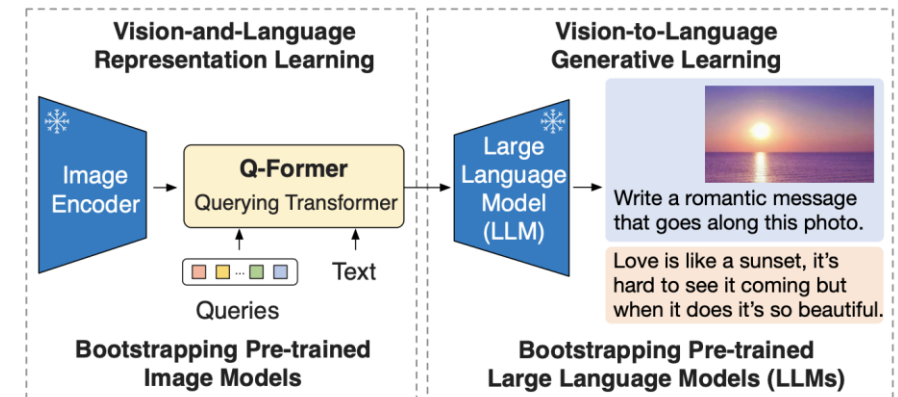
- BLIP: Bootstrapping Language Image Pretraining

- Multimodal mixture of Encoder-Decoder (MED)
  - Unimodal encoder
  - Image-grounded text encoder
  - Image-grounded text decoder
- Captioning and Filtering (CapFilt)



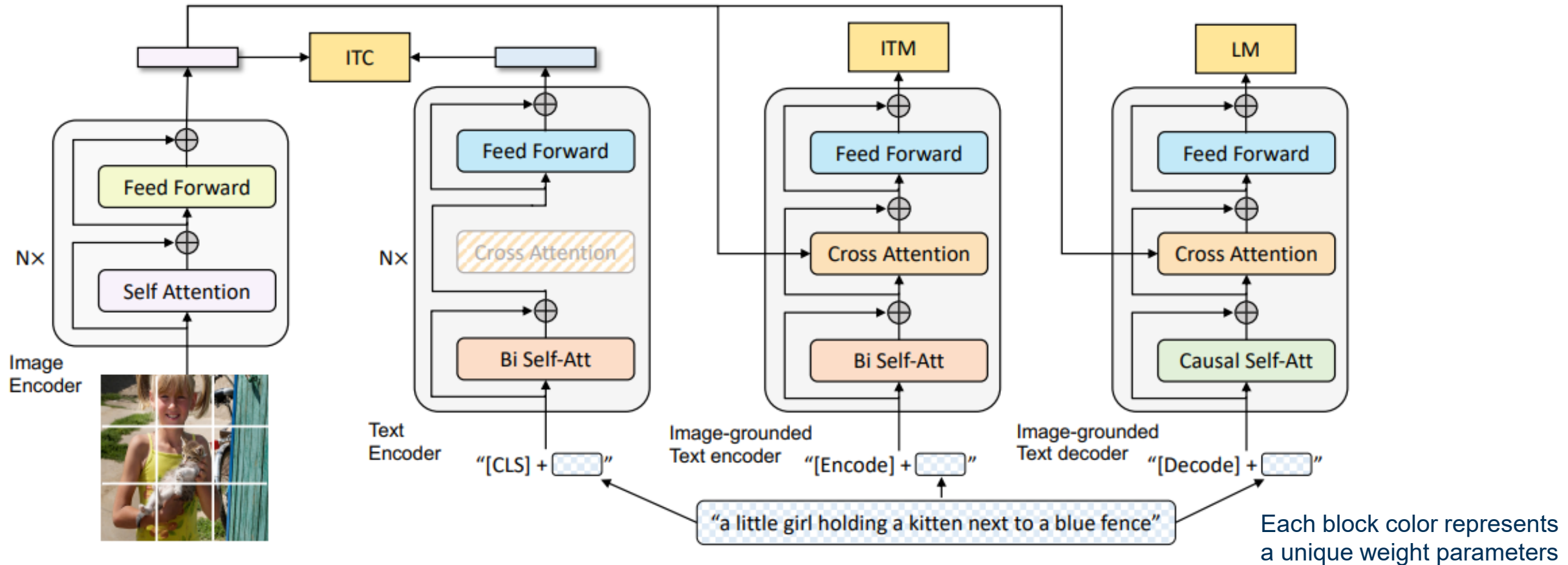
- BLIP 2: BLIP with Frozen Unimodal Models

- Modality bridge with Q-Former
- Frozen Unimodal Encoders
  - Compute Efficient
  - Very less forgetting





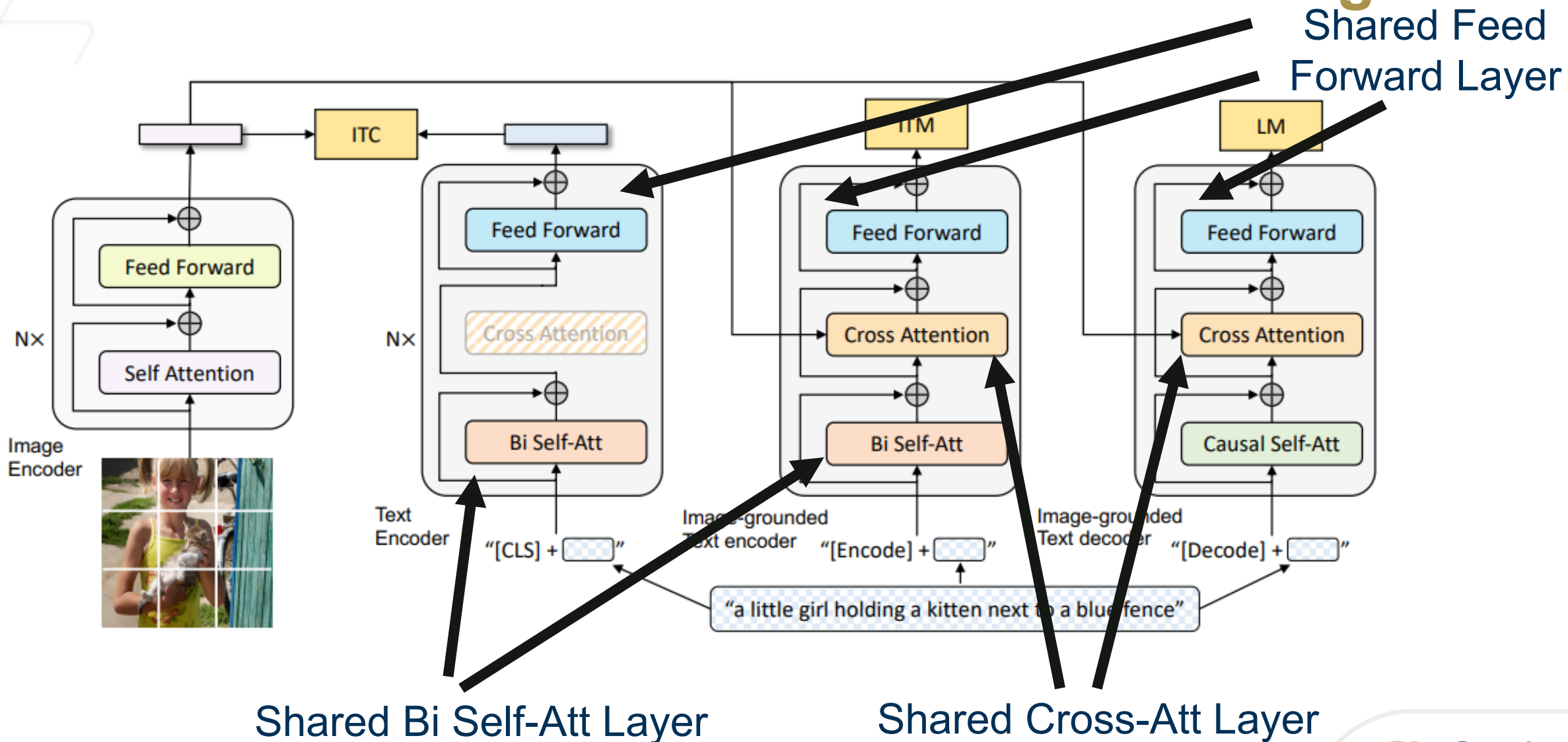
# BLIP Model Architecture



1. Unimodal Encoder (Image & Text Encoder)
2. Image-Grounded Text Encoder
3. Image-Grounded Text Decoder

ViT-B/16 or ViT-L/16

# BLIP Text Encoder/Decoder Parameter Sharing



# Pre-training Loss Function (ITC & ITM)

## Image-Text Contrastive Loss (ITC) [1]

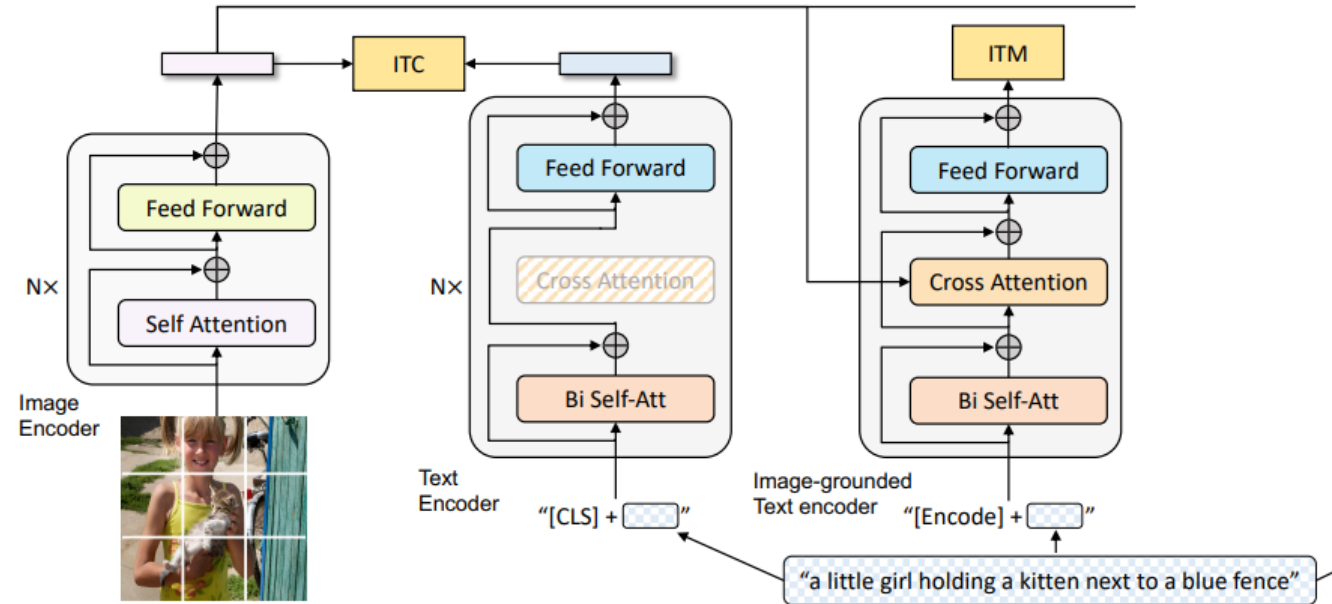
Align the feature space of the visual transformer and text transformer

Encourage positive image-text pairs, having similar representation

## Image-Text Matching Loss (ITM) [1]

Capture the fine-grained alignment b/w vision & language

Binary classify of whether an image-text pair matched or unmatched



Bi-directional self-attention block -> build representations of current tokens

# Pre-training Loss Function (LM)

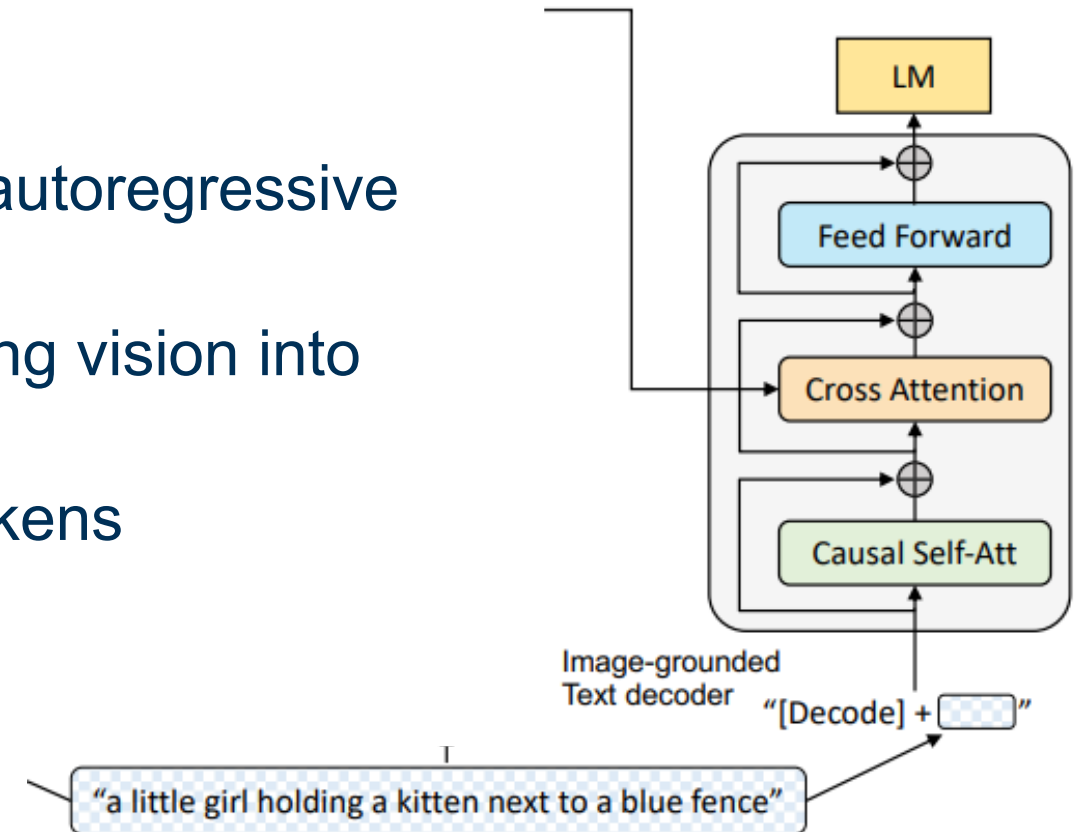
## Language Modeling Loss (LM)

Generate text captions given an image

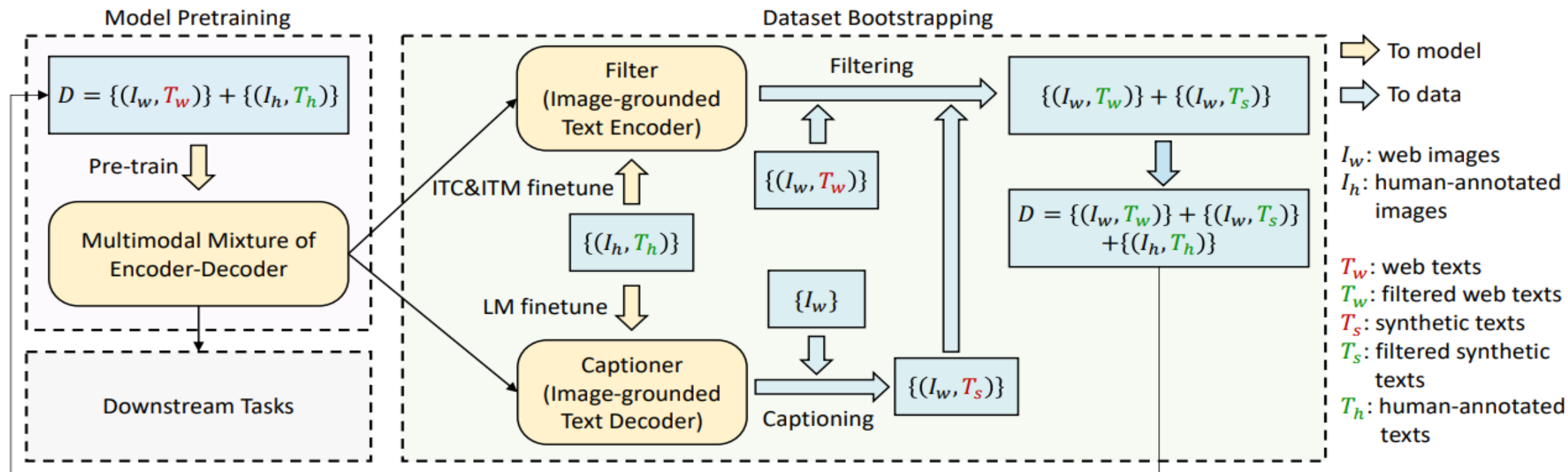
Maximize the likelihood of the caption in an autoregressive manner

Enables generalization capability of converting vision into coherent caption

Causal self-attention block -> predict next tokens



# Captioning and Filtering (CapFilt)



Create synthetic captions & filter out noisy captions

Gap: Image-text pairs from web data are noisy -> suboptimal performance in VLM

Captioner: Decoder Finedtuned with LM Loss.

Filter: Encoder Finetuned with ITC and ITM Loss. Filter text (both from web and synthetic data) if ITM is 0.



# Diversity is The Key

## Nucleus Sampling (Stochastic Search) for Caption Selection

Each token is sampled from a set of tokens whose cumulative probability mass exceeds a threshold  $p$  (0.9).

Noisier data, but better performance.

*More diverse and surprise data for better robustness?*

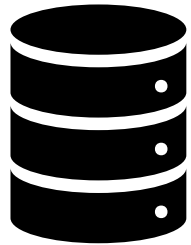
Generation method	Noise ratio	Retrieval-FT (COCO)		Retrieval-ZS (Flickr)		Caption-FT (COCO)		Caption-ZS (NoCaps)	
		TR@1	IR@1	TR@1	IR@1	B@4	CIDEr	CIDEr	SPICE
None	N.A.	78.4	60.7	93.9	82.1	38.0	127.8	102.2	13.9
Beam	19%	79.6	61.9	94.1	83.1	38.4	128.9	103.5	14.2
Nucleus	25%	80.6	63.1	94.8	84.9	38.6	129.7	105.1	14.4

Table 2. Comparison between beam search and nucleus sampling for synthetic caption generation. Models are pre-trained on 14M images.

# Pre-training & Finetuning Procedure

N times

web-data &  
human-annotated  
data



➡ To model

➡ To data

$I_w$ : web images

$I_h$ : human-annotated  
images

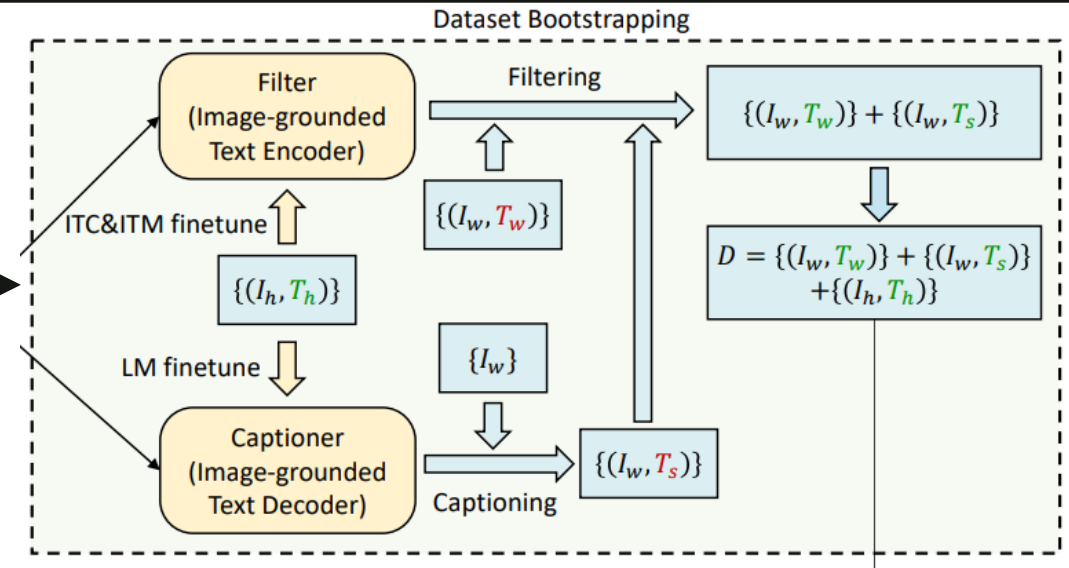
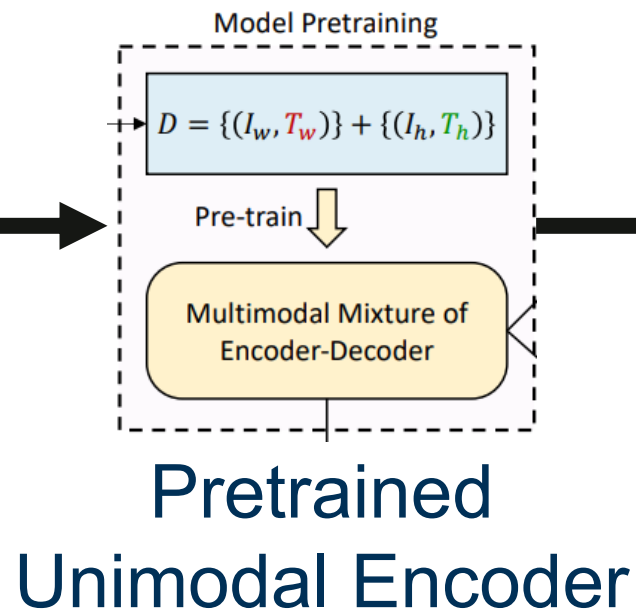
$T_w$ : web texts

$T_w$ : filtered web texts

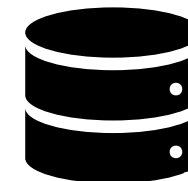
$T_s$ : synthetic texts

$T_s$ : filtered synthetic  
texts

$T_h$ : human-annotated  
texts



Finetuned CapFile



Filtered web-data  
& synthetic data

# Experiments and Results

- Image Transformer
  - ViT-B/16 or ViT-L/16 architecture pre-trained on ImageNet
- Text Transformer
  - BERT<sub>base</sub>
- Pre-training Dataset (14M, 129M Images)
  - 2 human-annotated datasets: COCO & Visual Genome
  - 3 web datasets: Conceptual Captions, Conceptual 12M, SBU captions
- Finetune to downstream datasets
- Image Resolution (384 x 384)

# Effect of CapFilt

## Metrics

TR: Image-text retrieval

IR: Image caption

B@4: BLEU

## Settings

FT: Finetuning

ZS: Zero-Shot

Pre-train dataset	Bootstrap		Vision backbone	Retrieval-FT (COCO)		Retrieval-ZS (Flickr)		Caption-FT (COCO)		Caption-ZS (NoCaps)	
	C	F		TR@1	IR@1	TR@1	IR@1	B@4	CIDEr	CIDEr	SPICE
COCO+VG +CC+SBU (14M imgs)	✗	✗	ViT-B/16	78.4	60.7	93.9	82.1	38.0	127.8	102.2	13.9
	✗	✓ <sub>B</sub>		79.1	61.5	94.1	82.8	38.1	128.2	102.7	14.0
	✓ <sub>B</sub>	✗		79.7	62.0	94.4	83.6	38.4	128.9	103.4	14.2
	✓ <sub>B</sub>	✓ <sub>B</sub>		80.6	63.1	94.8	84.9	38.6	129.7	105.1	14.4
COCO+VG +CC+SBU +LAION (129M imgs)	✗	✗	ViT-B/16	79.6	62.0	94.3	83.6	38.8	130.1	105.4	14.2
	✓ <sub>B</sub>	✓ <sub>B</sub>		81.9	64.3	96.0	85.0	39.4	131.4	106.3	14.3
	✓ <sub>L</sub>	✓ <sub>L</sub>		81.2	64.1	96.0	85.5	39.7	133.3	109.6	14.7
	✗	✗	ViT-L/16	80.6	64.1	95.1	85.5	40.3	135.5	112.5	14.7
	✓ <sub>L</sub>	✓ <sub>L</sub>		82.4	65.1	96.7	86.7	40.4	136.7	113.2	14.8

The use of captioning or/and filtering improves the performance across all tasks

Performance scales with more data (14M -> 129M) and more parameters (ViT-B/16 -> ViT-L/16)

# Sharing Parameters Results

Layers shared	#parameters	Retrieval-FT (COCO)		Retrieval-ZS (Flickr)		Caption-FT (COCO)		Caption-ZS (NoCaps)	
		TR@1	IR@1	TR@1	IR@1	B@4	CIDEr	CIDEr	SPICE
All	224M	77.3	59.5	93.1	81.0	37.2	125.9	100.9	13.1
All except CA	252M	77.5	59.9	93.1	81.3	37.4	126.1	101.2	13.1
All except SA	252M	78.4	60.7	93.9	82.1	38.0	127.8	102.2	13.9
None	361M	78.3	60.5	93.6	81.9	37.8	127.4	101.8	13.9

Sharing all parameters except for self-attention (SA) in the text encoder and decoder -> best performance & reduce model size

**Reasoning:** Sharing SA layer would degrade performance due to conflicting objective between the encoder and decoder.



# Image Text Retrieval

- Evaluate BLIP for both image-to-text retrieval (TR) and text-to-image retrieval (IR), fine tuned on image-text contrastive loss (ITC) and image text matching loss (ITM).
- Select k candidates based on the image-text feature similarity, and then rank the selected candidates based on their pairwise ITM scores.

Method	Pre-train # Images	COCO (5K test set)						Flickr30K (1K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER (Chen et al., 2020)	4M	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 (Gan et al., 2020)	4M	-	-	-	-	-	-	87.9	97.5	98.8	76.3	94.2	96.8
OSCAR (Li et al., 2020)	4M	70.0	91.1	95.5	54.0	80.8	88.5	-	-	-	-	-	-
UNIMO (Li et al., 2021b)	5.7M	-	-	-	-	-	-	89.4	98.9	99.8	78.0	94.2	97.1
ALIGN (Jia et al., 2021)	1.8B	77.0	93.5	96.9	59.9	83.3	89.8	95.3	99.8	100.0	84.9	97.4	98.6
ALBEF (Li et al., 2021a)	14M	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	14M	80.6	95.2	97.6	63.1	85.3	91.1	96.6	99.8	<b>100.0</b>	87.2	97.5	98.8
BLIP	129M	<b>81.9</b>	95.4	97.8	<b>64.3</b>	85.7	91.5	<b>97.3</b>	<b>99.9</b>	<b>100.0</b>	87.3	97.6	<b>98.9</b>
BLIP <sub>CapFilt-L</sub>	129M	81.2	<b>95.7</b>	<b>97.9</b>	64.1	<b>85.8</b>	<b>91.6</b>	97.2	<b>99.9</b>	<b>100.0</b>	<b>87.5</b>	<b>97.7</b>	<b>98.9</b>
BLIP <sub>ViT-L</sub>	129M	82.4	95.4	97.9	65.1	86.3	91.8	97.4	99.8	99.9	87.6	97.7	99.0

34 *Table 5.* Comparison with state-of-the-art image-text retrieval methods, finetuned on COCO and Flickr30K datasets. BLIP<sub>CapFilt-L</sub> pre-trains a model with ViT-B backbone using a dataset bootstrapped by captioner and filter with ViT-L. <sup>a</sup>

# Image Captioning

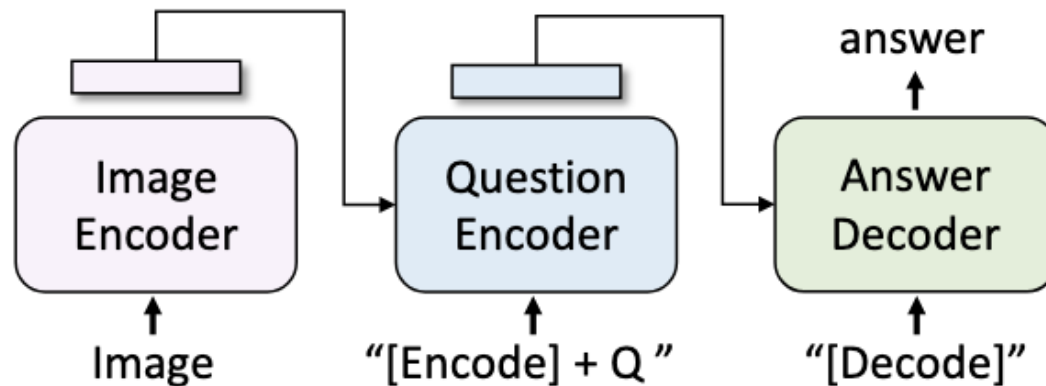
- Similar as SimvIm (Wang et al. (2021)), add a prompt “a picture of” at the beginning of each caption, which leads to slightly better results.

Method	Pre-train #Images	NoCaps validation								COCO Caption Karpathy test	
		in-domain		near-domain		out-domain		overall		B@4	C
		C	S	C	S	C	S	C	S		
Enc-Dec (Changpinyo et al., 2021)	15M	92.6	12.5	88.3	12.1	94.5	11.9	90.2	12.1	-	110.9
VinVL <sup>†</sup> (Zhang et al., 2021)	5.7M	103.1	14.2	96.1	13.8	88.3	12.1	95.5	13.5	38.2	129.3
LEMON <sub>base</sub> <sup>†</sup> (Hu et al., 2021)	12M	104.5	14.6	100.7	14.0	96.7	12.4	100.4	13.8	-	-
LEMON <sub>base</sub> <sup>†</sup> (Hu et al., 2021)	200M	107.7	14.7	106.2	14.3	107.9	13.1	106.8	14.1	<b>40.3</b>	<b>133.3</b>
BLIP	14M	111.3	15.1	104.5	14.4	102.4	13.7	105.1	14.4	38.6	129.7
BLIP	129M	109.1	14.8	105.8	14.4	105.7	13.7	106.3	14.3	39.4	131.4
BLIP <sub>CapFilt-L</sub>	129M	<b>111.8</b>	<b>14.9</b>	<b>108.6</b>	<b>14.8</b>	<b>111.5</b>	<b>14.2</b>	<b>109.6</b>	<b>14.7</b>	39.7	<b>133.3</b>
LEMON <sub>large</sub> <sup>†</sup> (Hu et al., 2021)	200M	116.9	15.8	113.3	15.1	111.3	14.0	113.4	15.0	40.6	135.7
SimVLM <sub>huge</sub> (Wang et al., 2021)	1.8B	113.7	-	110.9	-	115.2	-	112.2	-	40.6	143.3
BLIP <sub>ViT-L</sub>	129M	114.9	15.2	112.1	14.9	115.3	14.4	113.2	14.8	40.4	136.7

# Visual Question Answering

- An answer generation task, which enables open-ended VQA.
- During finetuning, they rearrange the pre-trained model, where an image-question is first encoded into multimodal embeddings and then given to an answer decoder. The VQA model is finetuned with the LM loss using ground-truth answers as targets.

(a) VQA

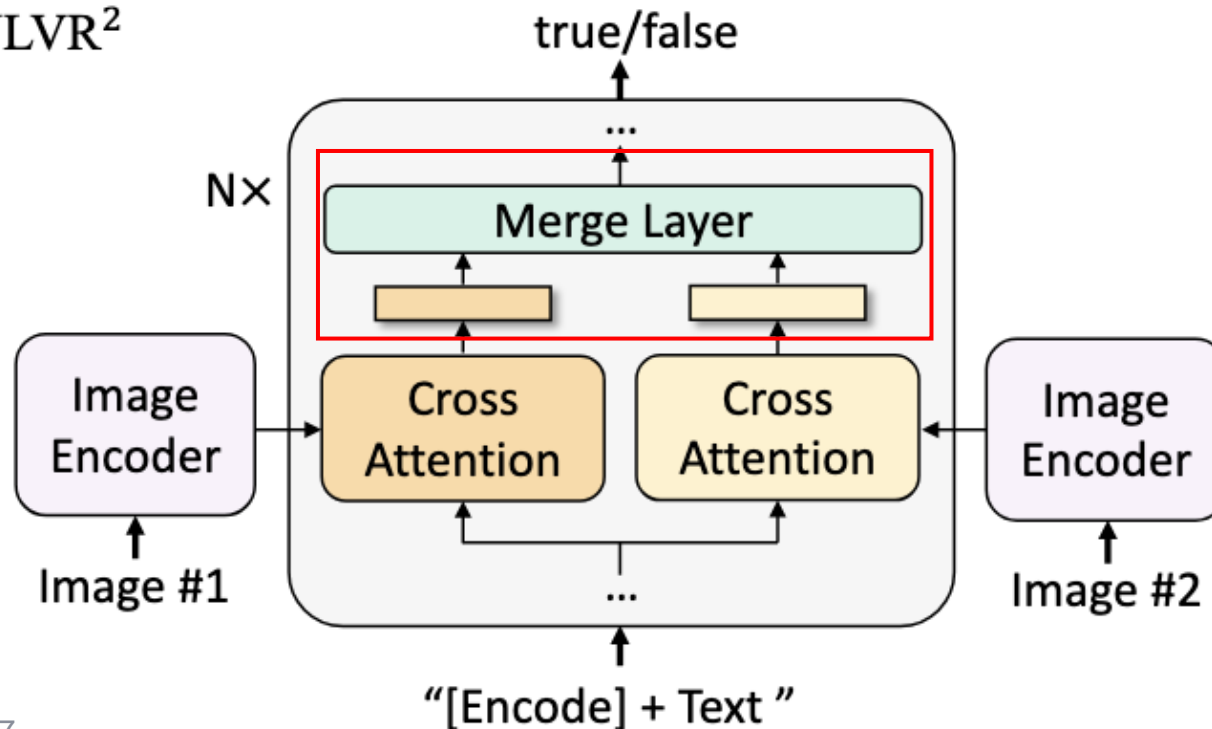


Method	Pre-train #Images	VQA		NLVR <sup>2</sup>	
		test-dev	test-std	dev	test-P
LXMERT	180K	72.42	72.54	74.90	74.50
UNITER	4M	72.70	72.91	77.18	77.85
VL-T5/BART	180K	-	71.3	-	73.6
OSCAR	4M	73.16	73.44	78.07	78.36
SOHO	219K	73.25	73.47	76.37	77.32
VILLA	4M	73.59	73.67	78.39	79.30
UNIMO	5.6M	75.06	75.27	-	-
ALBEF	14M	75.84	76.04	82.55	83.14
SimVLM <sub>base</sub> <sup>†</sup>	1.8B	77.87	78.14	81.72	81.77
BLIP	14M	77.54	77.62	<b>82.67</b>	82.30
BLIP	129M	78.24	78.17	82.48	<b>83.08</b>
BLIP <sub>CapFilt-L</sub>	129M	<b>78.25</b>	<b>78.32</b>	82.15	82.24

# Natural Language Visual Reasoning

- NLVR (Suhr et al., 2019) asks the model to predict whether a sentence describes a pair of images.
- Make a "simple" modification to our pre-trained model -> a more computational-efficient architecture.

(b) NLVR<sup>2</sup>



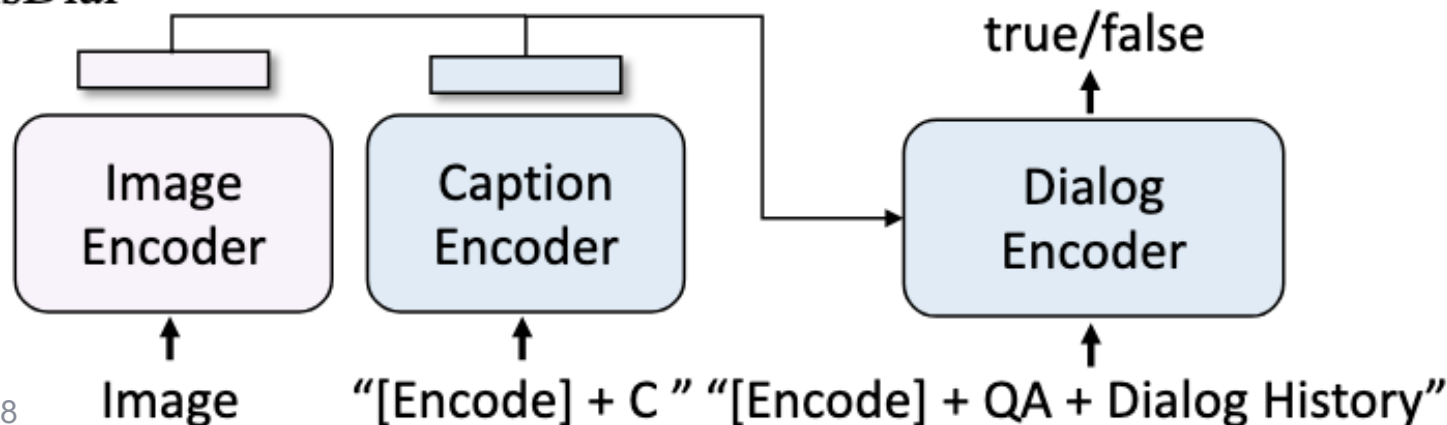
Method	Pre-train #Images	VQA		NLVR <sup>2</sup>	
		test-dev	test-std	dev	test-P
LXMERT	180K	72.42	72.54	74.90	74.50
UNITER	4M	72.70	72.91	77.18	77.85
VL-T5/BART	180K	-	71.3	-	73.6
OSCAR	4M	73.16	73.44	78.07	78.36
SOHO	219K	73.25	73.47	76.37	77.32
VILLA	4M	73.59	73.67	78.39	79.30
UNIMO	5.6M	75.06	75.27	-	-
ALBEF	14M	75.84	76.04	82.55	83.14
SimVLM <sub>base</sub> <sup>†</sup>	1.8B	77.87	78.14	81.72	81.77
BLIP	14M	77.54	77.62	<b>82.67</b>	82.30
BLIP	129M	78.24	78.17	82.48	<b>83.08</b>
BLIP <sub>CapFilt-L</sub>	129M	<b>78.25</b>	<b>78.32</b>	82.15	82.24

# Visual Dialog

- VisDial (Das et al., 2017) extends VQA in a natural conversational setting, where the model needs to predict an answer not only based on the image-question pair, but also considering the dialog history and the image's caption.
- Follow the discriminative setting where the model ranks a pool of answer candidates.

Method	MRR↑	R@1↑	R@5↑	R@10↑	MR↓
VD-BERT	67.44	54.02	83.96	92.33	3.53
VD-ViLBERT†	69.10	55.88	85.50	93.29	3.25
BLIP	<b>69.41</b>	<b>56.44</b>	<b>85.90</b>	<b>93.30</b>	<b>3.20</b>

(c) VisDial





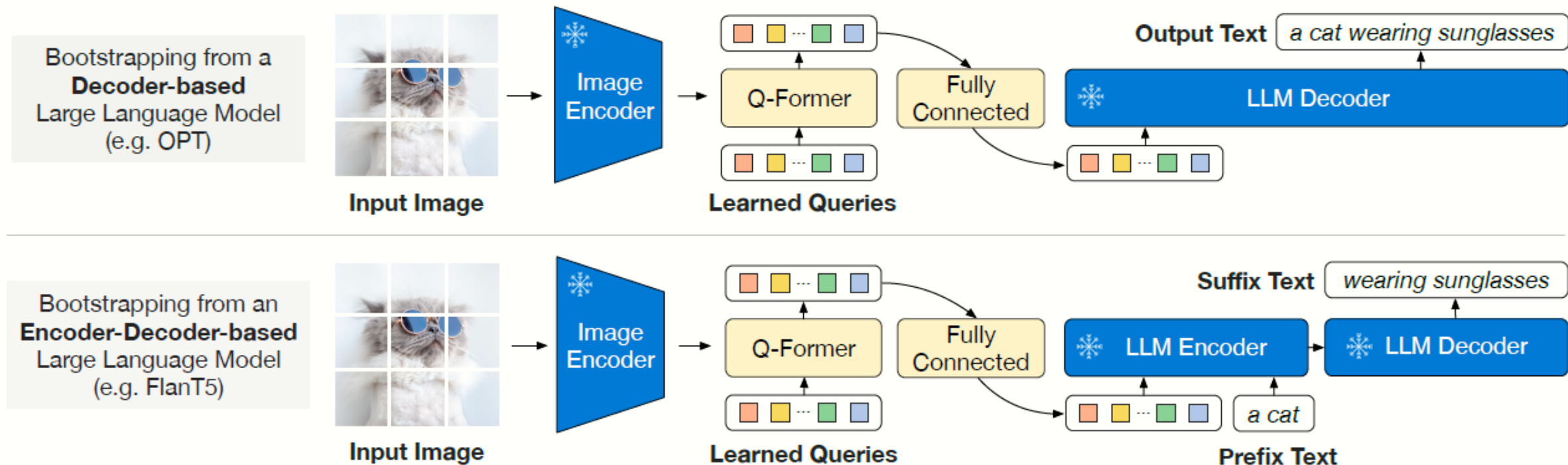
# Zero-Shot Transfer to Video Language

- Perform zero-shot transfer to text-to-video retrieval and video question answering, where we directly evaluate the models trained on COCO-retrieval and VQA, respectively.
- To process video input, we uniformly sample  $n$  frames per video ( $n = 8$  for retrieval and  $n = 16$  for QA) and concatenate the frame features into a single sequence. Note that this simple approach ignores all temporal information.

Method	R1↑	R5↑	R10↑	MdR↓
<i>zero-shot</i>				
ActBERT (Zhu & Yang, 2020)	8.6	23.4	33.1	36
SupportSet (Patrick et al., 2021)	8.7	23.0	31.1	31
MIL-NCE (Miech et al., 2020)	9.9	24.0	32.4	29.5
VideoCLIP (Xu et al., 2021)	10.4	22.2	30.0	-
FiT (Bain et al., 2021)	18.7	39.5	51.6	10
BLIP	<b>43.3</b>	<b>65.6</b>	<b>74.7</b>	<b>2</b>
<i>finetuning</i>				
ClipBERT (Lei et al., 2021)	22.0	46.8	59.9	6
VideoCLIP (Xu et al., 2021)	30.9	55.4	66.8	-

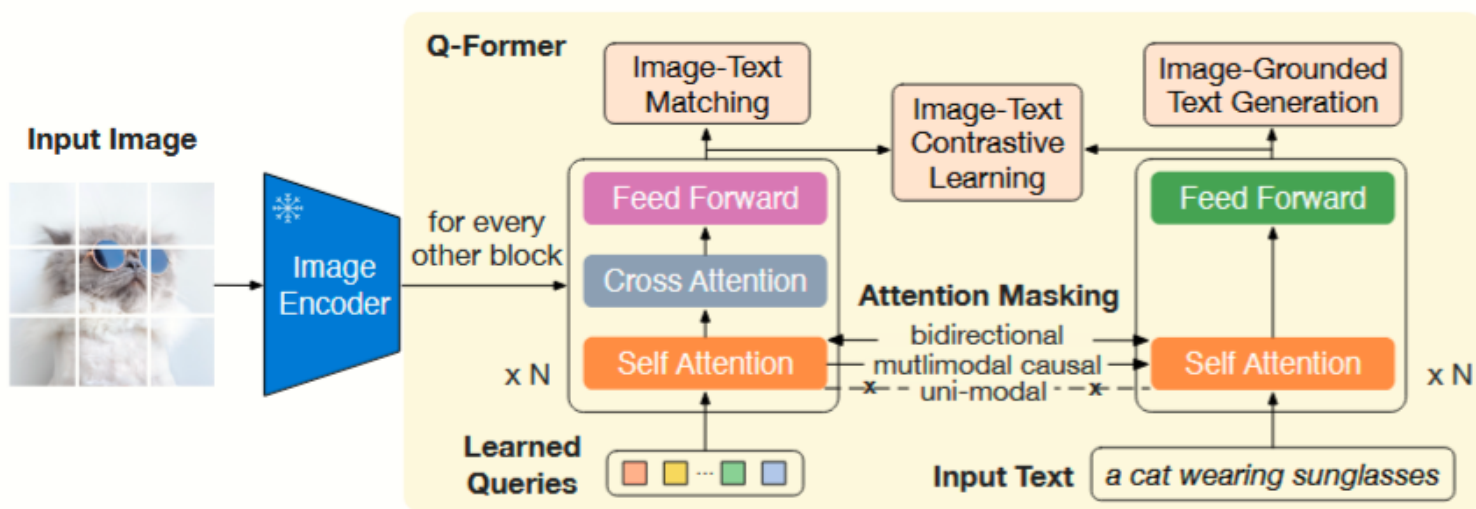
Method	MSRVTT-QA	MSVD-QA
<i>zero-shot</i>		
VQA-T (Yang et al., 2021)	2.9	7.5
BLIP	19.2	35.2
<i>finetuning</i>		
HME (Fan et al., 2019)	33.0	33.7
HCRN (Le et al., 2020)	35.6	36.1
VQA-T (Yang et al., 2021)	41.5	46.3

# BLIP2: Introduces Q-Former



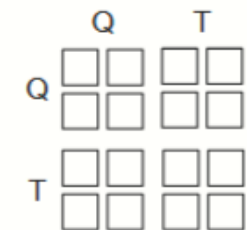
Li et al., BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

# BLIP2: Q-Former



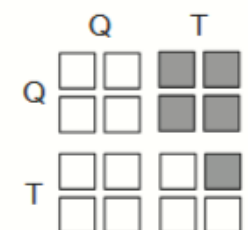
**Q**: query token positions; **T**: text token positions.

■ masked □ unmasked



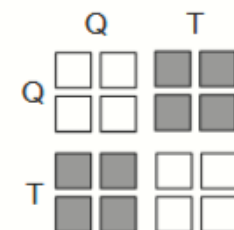
Bi-directional Self-Attention Mask

Image-Text Matching



Multi-modal Causal Self-Attention Mask

Image-Grounded Text Generation



Uni-modal Self-Attention Mask

Image-Text Contrastive Learning

# BLIP2 Test Results

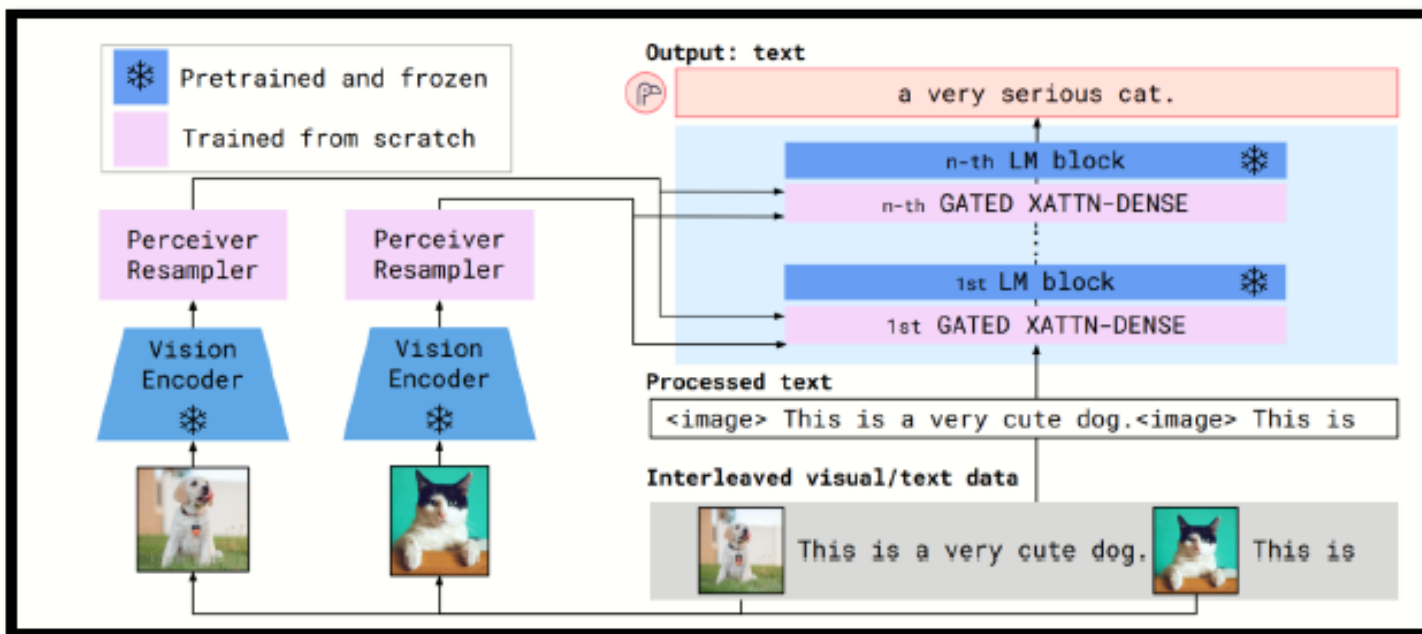
Models	#Trainable Params	Open- sourced?	Visual Question Answering	Image Captioning		Image-Text Retrieval	
			VQAv2 (test-dev) VQA acc.	NoCaps (val) CIDEr	SPICE	Flickr (test) TR@1	IR@1
BLIP (Li et al., 2022)	583M	✓	-	113.2	14.8	96.7	86.7
SimVLM (Wang et al., 2021b)	1.4B	✗	-	112.2	-	-	-
BEIT-3 (Wang et al., 2022b)	1.9B	✗	-	-	-	94.9	81.5
Flamingo (Alayrac et al., 2022)	10.2B	✗	56.3	-	-	-	-
BLIP-2	188M	✓	<b>65.0</b>	<b>121.6</b>	<b>15.8</b>	<b>97.6</b>	<b>89.7</b>

# Summary of Strengths, Weaknesses, Relationships

- New pre-training (fine-tuning) techniques
- Synthetic data (less noise/variance)
- Scalability (In terms of performance)
- Network architecture is modular
- Very specific finetuning procedure
- Introduced new architectures after pre-training
- Very specific testing setup with a lot of model changes
- Synthetic data (more bias)



- Flamingo:



Language Model

Connection Module

Vision Encoder

Pre-trained: 70B Chinchilla

Perceiver Resampler  
Gated Cross-attention + Dense

Pre-trained: Nonormalizer-Free ResNet (NFNet)

Flamingo

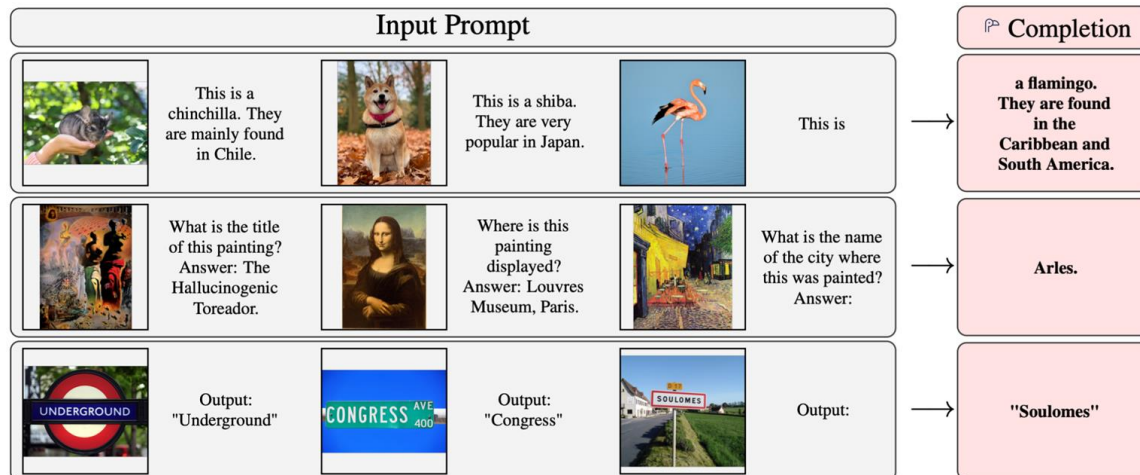
Slide by Chunyuan Li

# Flamingo: a Visual Language Model for Few-Shot Learning (Alayrac et al.)

## Goal: Few-shot learning to perform novel multimodal tasks

### Implications

- Key element of human intelligence
- Don't need to fine-tune models
  - Resource intensive
  - Task-specific annotated data



### Contributions

- **Flamingo**: family of VLMs [1]
  - Connect frozen vision-only and language-only models
  - Interactive, generates open-ended text
- State-of-the-art learning on 16 tasks (Q)
  - Using just examples
  - VQA, captioning, visual dialogue, etc.

Q: Can it localize objects?

# Related Works

## Adapting models to novel tasks

### Partial Fine-Tuning

- Adapter modules [2]
  - Few trainable parameters per task
  - Original network parameters stay fixed
- BitFit [3]
  - Only modifies bias term
  - Competitive performance to fine-tuned models

### Prompt-Based Approach

- GPT-3 [4]
  - Show in-context examples within prompt
  - Scaled-up language model
- Prompt-Tuning [5] (Q)
  - Prompt optimization through gradient descent
  - Learn “soft prompts” to influence frozen LM to perform tasks

Q: Since prompt-tuning achieved better few-shot learning performance than GPT-3, could it also achieve better performance in multimodal space?

# Related Works

## Chinchilla: Base Language Model [6]

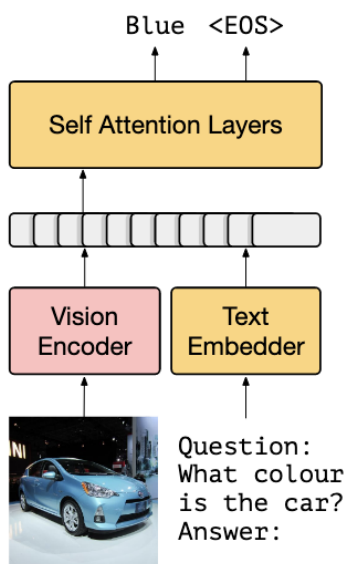
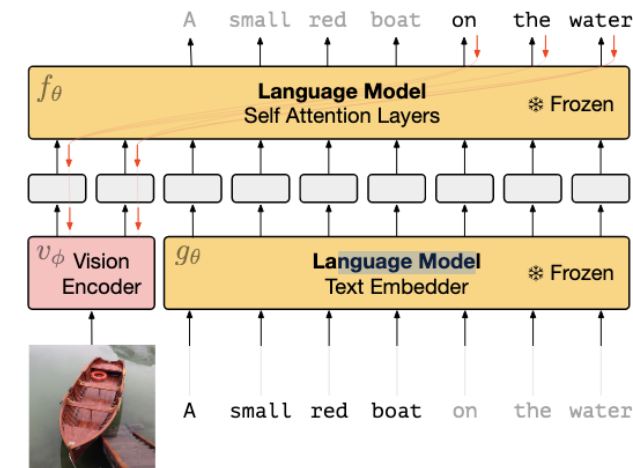
- SOTA accuracy on MMLU
  - MMLU: Exam-like questions on academic subjects
- Scaled training tokens at same rate as model size
- Trained on *MassiveText* [7]

---

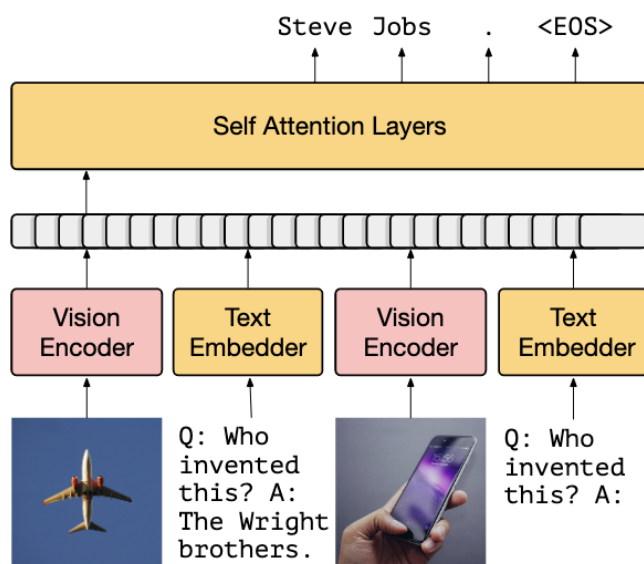
Random	25.0%
Average human rater	34.5%
GPT-3 5-shot	43.9%
<i>Gopher</i> 5-shot	60.0%
<b><i>Chinchilla</i> 5-shot</b>	<b>67.6%</b>
Average human expert performance	89.8%

---

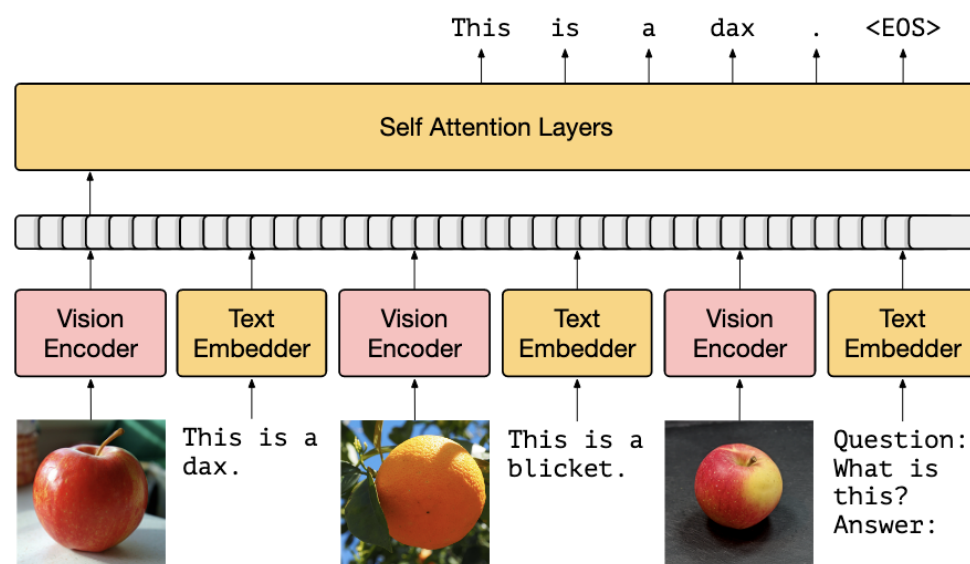
# Training:



(a) 0-shot VQA



(b) 1-shot outside-knowledge VQA



(c) Few-shot image classification



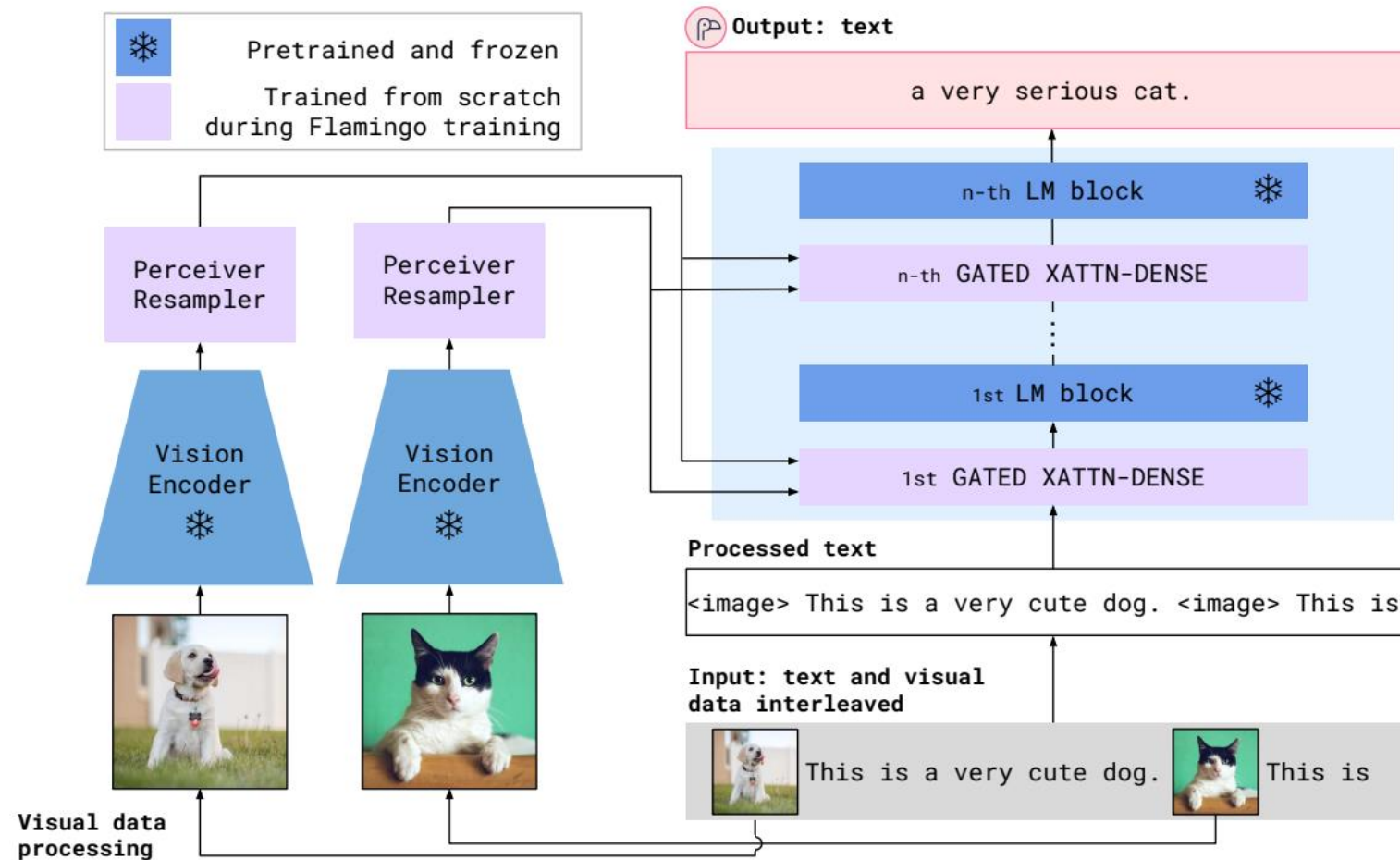
# Flamingo VL

- Three challenges for training with image/video and text.
  - **Supporting both images and videos**
    - Images /videos :2D structure with high dimensionality.
    - Text: 1D sequence
    - Sol.: Introduce Perceiver Resample module.
  - **The interaction with image/video and text**
    - keep the pretrained model's language understanding and generation capabilities fully intact
    - Sol.: Interleave cross-attention layers with frozen self-attention. gating mechanism.
  - **Obtaining multimodal dataset to induce good generalist capabilities**
    - Dataset with weak matching problem
    - Sol.: combine dataset with standard strong related paired image/text and video/text datasets



# Flamingo VLM

- Overview of the Flamingo Model

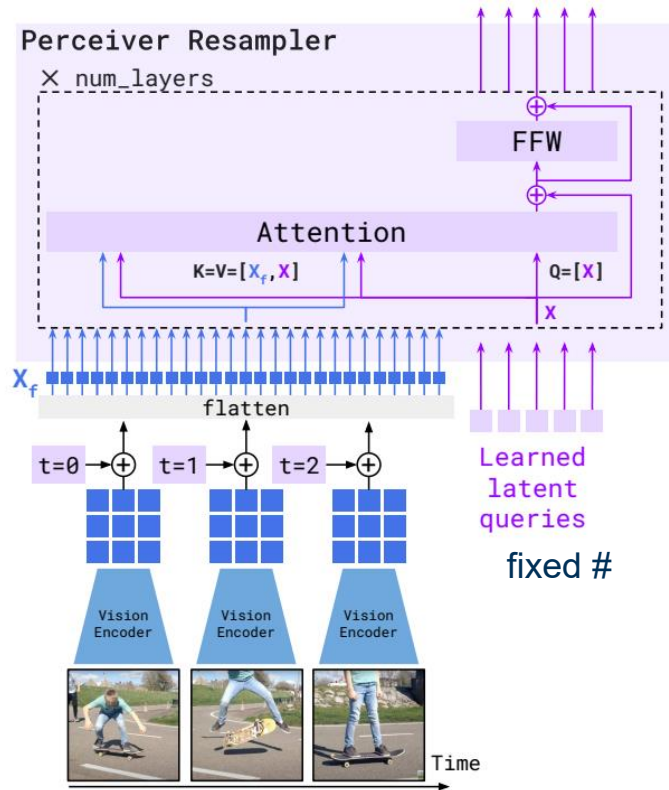


- Each image is encoded individually

Slide by Azade Farshad and Mei Sun

# Flamingo VL

- Model structure - **Supporting both images and videos**



```
def perceiver_resampler(
    x_f, # The [T, S, d] visual features (T=time, S=space)
    time_embeddings, # The [T, 1, d] time pos embeddings.
    x, # R learned latents of shape [R, d]
    num_layers, # Number of layers
):
    """The Perceiver Resampler model."""

    # Add the time position embeddings and flatten.
    x_f = x_f + time_embeddings
    x_f = flatten(x_f) # [T, S, d] -> [T * S, d]
    # Apply the Perceiver Resampler layers.
    for i in range(num_layers):
        # Attention.
        x = x + attention_i(q=x, kv=concat([x_f, x]))
        # Feed forward.
        x = x + ffw_i(x)
    return x
```

pseudo code

- Using pre-trained ResNet to get visual features  $X_f$
- Compress the encode image into R tokens
- Core of this module : Attention .
  - Query: the learned latent token X
  - Key=Value: the concatenation of  $X_f$  and the learned latent token X
  - Better performance by concatenating keys and values obtained from latent
- If the input is video
  - $X_f$  will add time embeddings

Maps a **variable size grid of visual features** from the Vision Encoder to a **fixed number of output token** (5 in the

figure

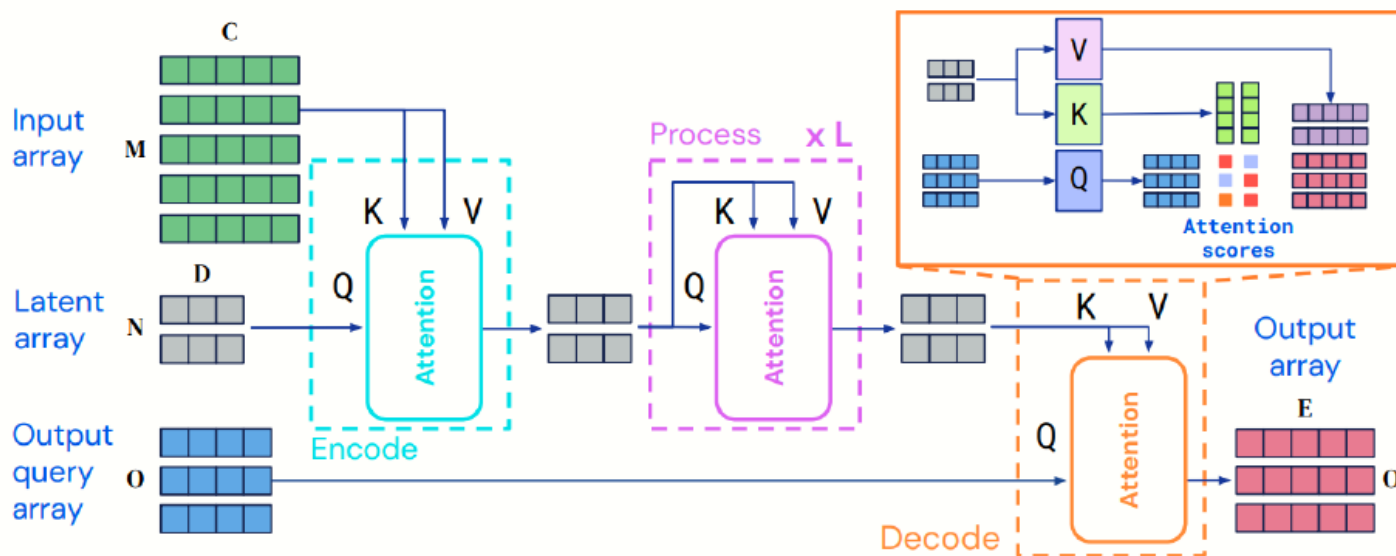
August 28,  
2025

Slide by Azade Farshad and Mei Sun

# Perceiver / Perceiver IO:

## Transformer for general data perception

- General data processing method given data can be mapped into sequence of vectors
- Use cross attention to fetch information from input
- Self attention to process input.
- Use cross attention to fetch relevant information and send to output.

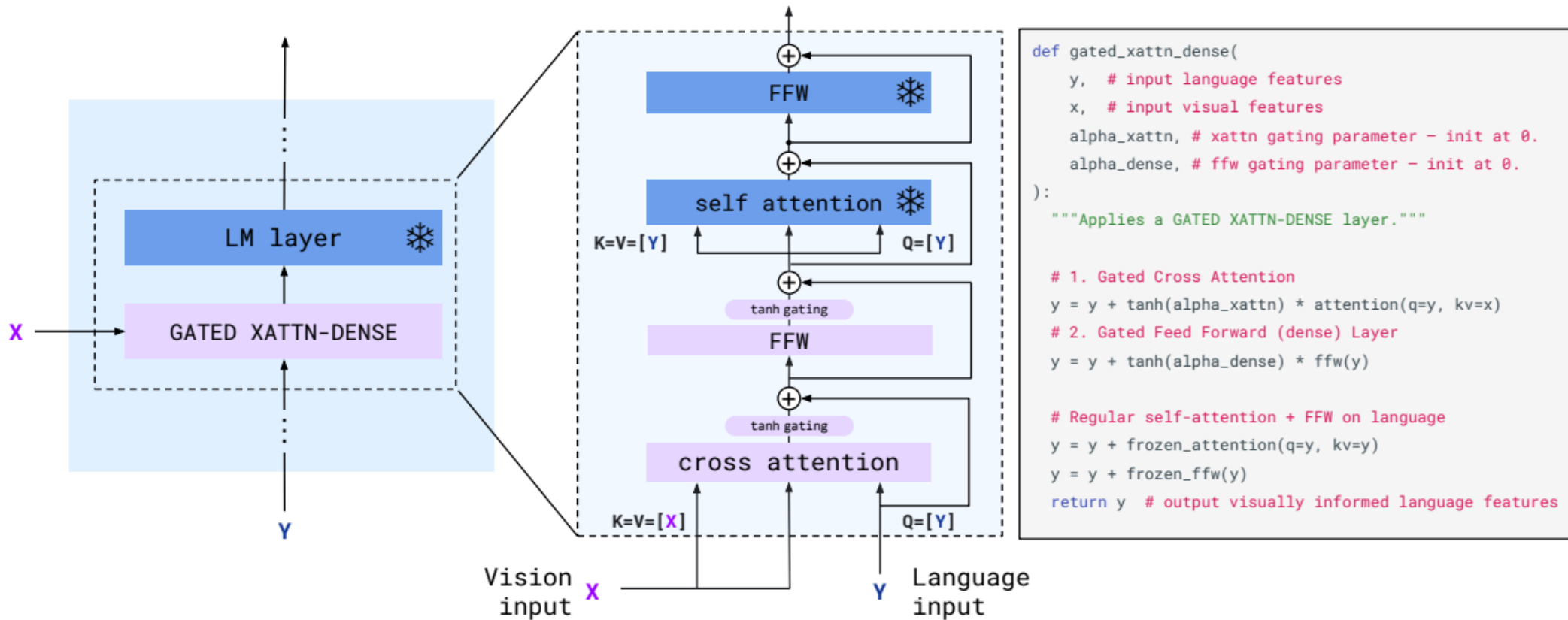


Jaegle, Andrew, et al. "Perceiver: General perception with iterative attention." *ICML*, 2021.  
Jaegle, Andrew, et al. "Perceiver io: A general architecture for structured inputs & outputs." *ICLR*, 2021



# Flamingo VLM

- Model structure - The interaction with image/video and text



A **Gated Cross attention** mechanism is proposed to fuse images and text.

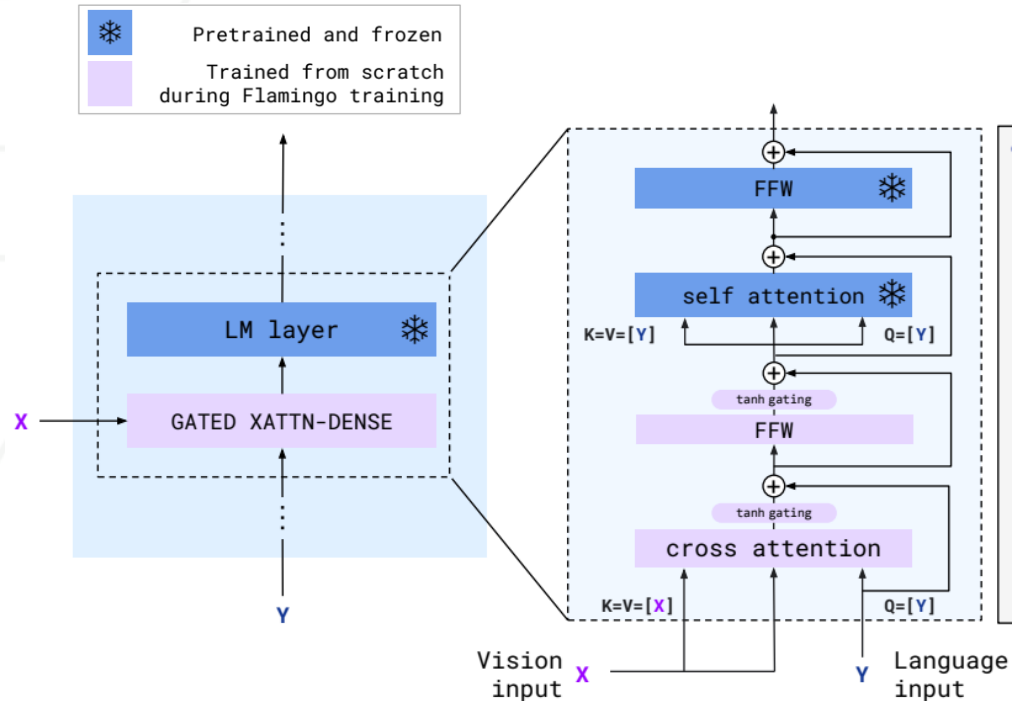
Slide by Azade Farshad and Mei Sun





# Flamingo VLM

- Model structure - The interaction with image/video and text

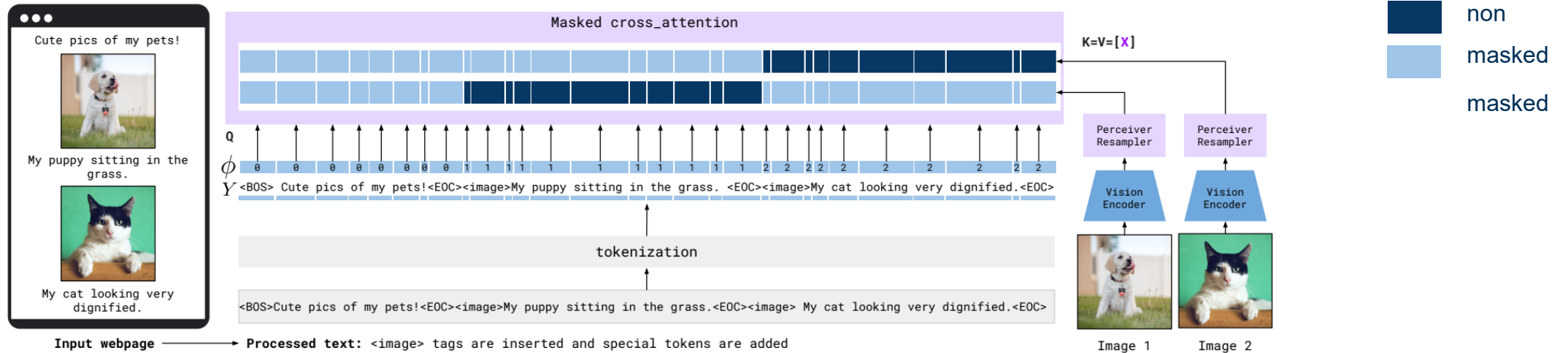


- Frozen LM layers
  - LM: 70B parameter Chinchilla
  - keep pretrained LM's language understanding
- Gated Cross Attention:
  - Query: Y, Key=Value: X
  - Tanh Gating: Initialized with 0 then gradually increases
  - Transitions from a fully trained text-only model to a visual language model.
- The LM can generate text conditioned on the above visual tokens



# Flamingo VLM

- Model structure - **Interleaved visual data and text support**



- Multi-visual input support: per-image/video attention masking
- During Cross-attention,
  - each text can only focus on one image before it.
  - Function  $\phi$  : for each token what is the index of the last preceding image
- During final prediction,

# Flamingo VLM

- Model structure - **Obtaining multimodal dataset to induce good generalist capabilities**



Image-Text Pairs dataset  
[N=1, T=1, H, W, C]



Video-Text Pairs dataset  
[N=1, T>1, H, W, C]



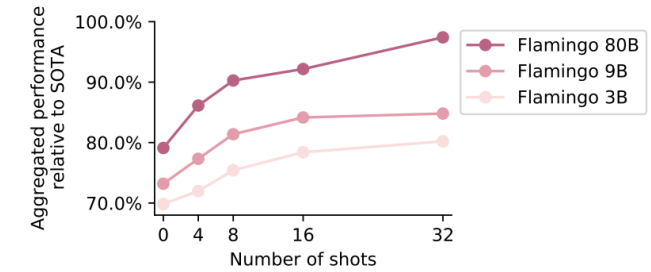
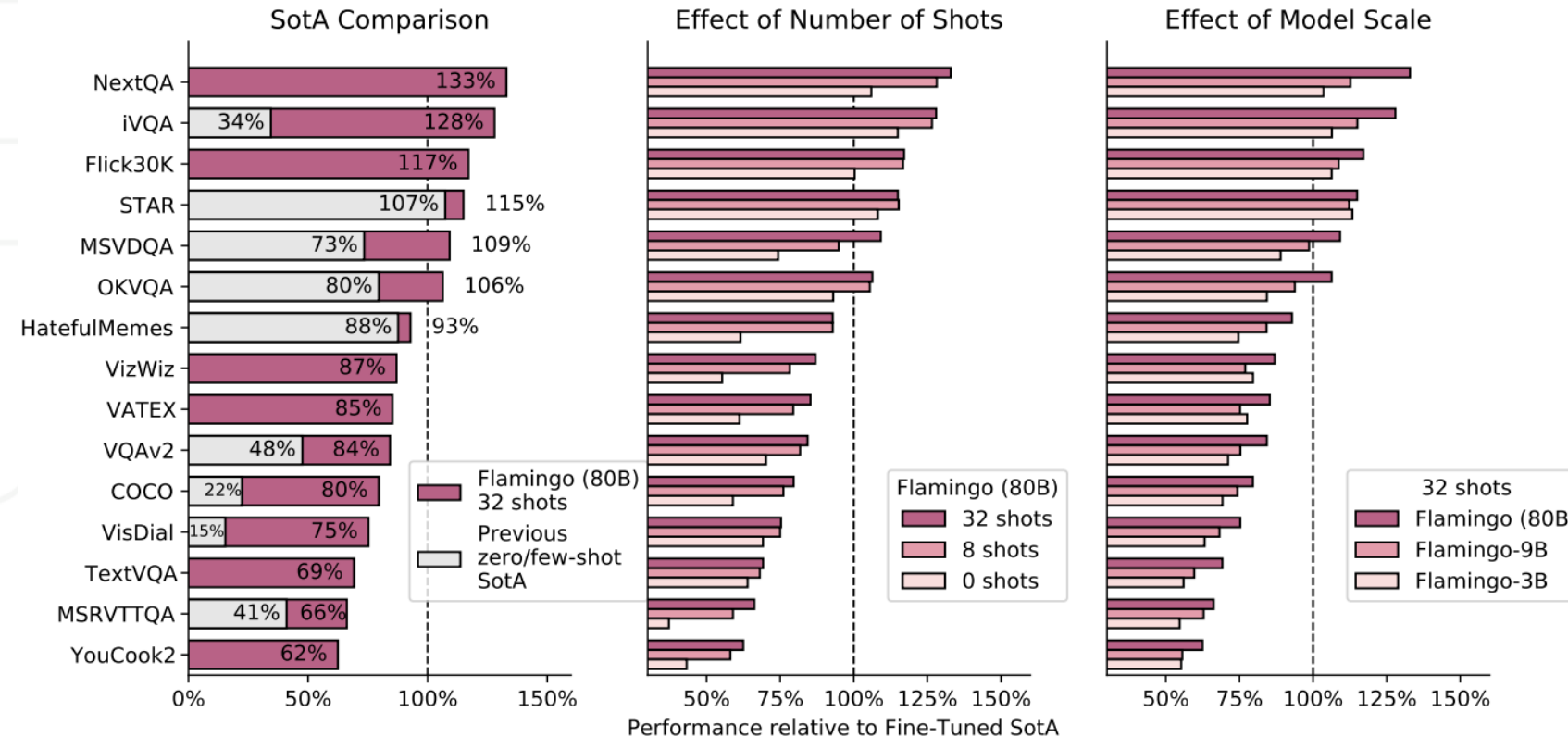
Multi-Modal Massive Web (M3W) dataset  
[N>1, T=1, H, W, C]

- M3W: Scrapping 43 million webpages from the Internet
- Training on a mixture of vision and language datasets
  - M3W(185M images+ 182G text)
  - ALIGN(1.8B images with alt-text)
  - LTIP (312M images/text)
  - VTP(27M short video/text)



# Flamingo VLM

- Result: Overview of the results of the Flamingo models



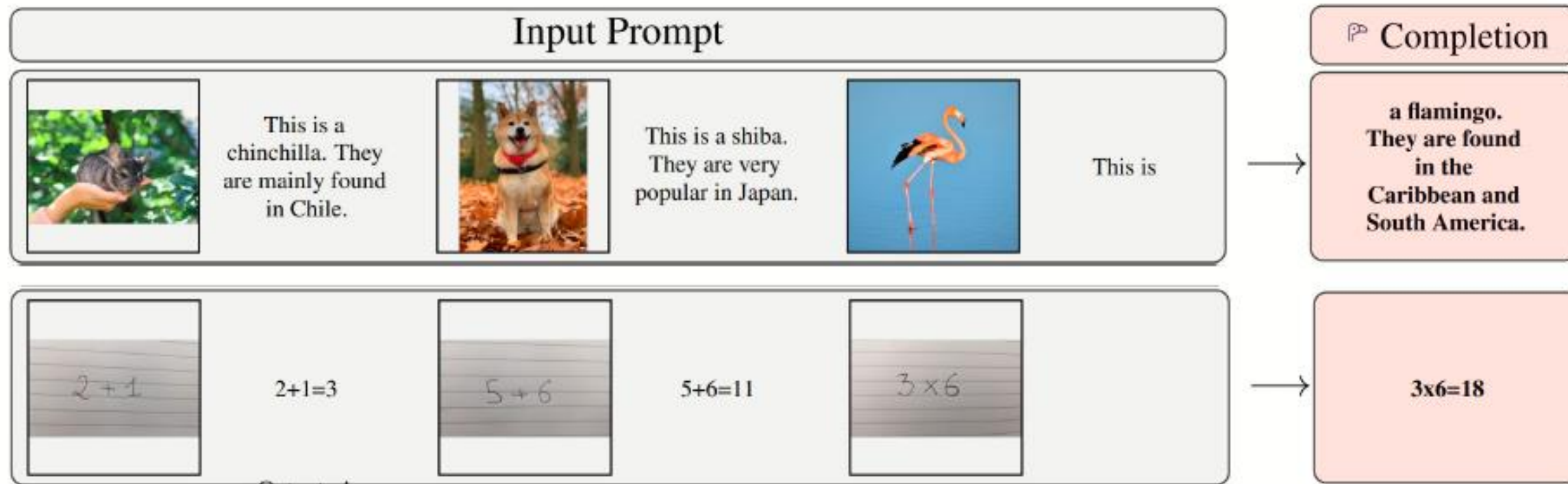
- Larger model sizes and more few-shot examples lead to better performance

- Performance of Flamingo model using different numbers of shots and of different sizes, (without fine-tuned) in comparison with SoTA fine-tuned baseline.

Slide by Azade Farshad and Mei Sun

- Flamingo: Multimodal In-Context-Learning

Emerging  
Property





# Approach

## Training on a mixture of vision and language datasets

- Datasets

- M3W: Interleaved image and text dataset.
- ALIGN: 1.8B text-to-image
- LTIP: 312M long-text and image
- VTP: 27M short-video and text



Figure 9: **Training datasets.** Mixture of training datasets of different formats.  $N$  corresponds to the number of visual inputs for a single example. For paired image (or video) and text datasets,  $N = 1$ .  $T$  is the number of video frames ( $T = 1$  for images).  $H$ ,  $W$ , and  $C$  are height, width and color channels.

- Multi-objective training and optimisation strategy.

- Tuning the per-dataset weights  $\lambda_m$  is key to performance.
- Below weights were obtained empirically at a small model scale and kept fixed afterwards.

Dataset	M3W	ALIGN	LTIP	VTP
$\lambda_m$	1.0	0.2	0.2	0.03

# Experiments and Results

## Zero/Few-shot Performance

Method	FT	Shot	OKVQA (I)	VQA <sub>v2</sub> (I)	COCO (I)	MSVDQA (V)	VATEX (V)	VizWiz (I)	Flick30K (I)	MSRVTTQA (V)	iVQA (V)	YouCook2 (V)	STAR (V)	VisDial (I)	TextVQA (I)	NextQA (I)	HatefulMemes (I)	RareAct (V)
Zero/Few shot SOTA	✗		[34] 43.3 (16)	[114] 38.2 (4)	[124] 32.2 (0)	[58] 35.2 (0)	-	-	-	[58] 19.2 (0)	[135] 12.2 (0)	-	[143] 39.4 (0)	[79] 11.6 (0)	-	-	[85] 66.1 (0)	[85] 40.7 (0)
Flamingo-3B	✗	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	✗	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	✗	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	47.3	30.6	26.1	56.3	-
Flamingo-9B	✗	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	✗	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	<u>42.8</u>	50.4	33.6	24.7	62.7	-
	✗	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	50.4	32.6	28.4	63.5	-
Flamingo	✗	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	<u>60.8</u>
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	<b>55.6</b>	36.5	30.8	68.6	-
	✗	32	<b>57.8</b>	<b>67.6</b>	<b>113.8</b>	<b>52.3</b>	<b>65.1</b>	<b>49.8</b>	<b>75.4</b>	<b>31.0</b>	<b>45.3</b>	<b>86.8</b>	42.2	<b>55.6</b>	<b>37.9</b>	<b>33.5</b>	<b>70.0</b>	-
Pretrained FT SOTA	✓		54.4 [34] (10K)	80.2 [140] (444K)	143.3 [124] (500K)	47.9 [28] (27K)	76.3 [153] (500K)	57.2 [65] (20K)	67.4 [150] (30K)	46.8 [51] (130K)	35.4 [135] (6K)	138.7 [132] (10K)	36.7 [128] (46K)	75.2 [79] (123K)	54.7 [137] (20K)	25.2 [129] (38K)	79.1 [62] (9K)	-

Table 1: **Comparison to the state of the art.** A *single* Flamingo model reaches the state of the art on a wide array of image (I) and video (V) understanding tasks with few-shot learning, significantly outperforming previous best zero- and few-shot methods with as few as four examples. More importantly, using only 32 examples and without adapting any model weights, Flamingo *outperforms* the current best methods – fine-tuned on thousands of annotated examples – on seven tasks. Best few-shot numbers are in **bold**, best numbers overall are underlined.

# Experiments and Results

## Fine-Tuning Performance

Method	VQAV2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA		HatefulMemes
	test-dev	test-std			test-dev	test-std		valid	test-std		valid	test-std	
🦄 32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	70.0
🦄 Fine-tuned	<b>82.0</b>	<b>82.1</b>	138.1	<b>84.2</b>	<b>65.7</b>	<b>65.4</b>	<b>47.4</b>	61.8	59.7	118.6	<b>57.1</b>	54.1	<b>86.6</b>
SotA	81.3 <sup>†</sup>	81.3 <sup>†</sup>	<b>149.6<sup>†</sup></b>	81.4 <sup>†</sup>	57.2 <sup>†</sup>	60.6 <sup>†</sup>	46.8	<b>75.2</b>	<b>75.4<sup>†</sup></b>	<b>138.7</b>	54.7	<b>73.7</b>	84.6 <sup>†</sup>
	[133]	[133]	[119]	[153]	[65]	[65]	[51]	[79]	[123]	[132]	[137]	[84]	[152]

Table 2: **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* does not achieve SotA with few-shot learning. *Flamingo* sets a new SotA on five of them, outperforming methods (marked with <sup>†</sup>) that use tricks such as model ensembling or domain-specific metric optimisation (e.g., CIDEr optimisation).

# Experiments and Results

## Ablation Study

Ablated setting	Flamingo-3B original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
<b>Flamingo-3B model</b>			3.2B	1.74s	86.5	42.1	55.8	36.3	53.4	<b>70.7</b>
(i)	Training data	All data	w/o Video-Text pairs	3.2B	1.42s	84.2	43.0	53.9	34.5	67.3
			w/o Image-Text pairs	3.2B	0.95s	66.3	39.2	51.6	32.0	60.9
			Image-Text pairs → LAION	3.2B	1.74s	79.5	41.4	53.5	33.9	66.4
			w/o M3W	3.2B	1.02s	54.1	36.5	52.7	31.4	53.4
(ii)	Optimisation	Accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	33.2	62.9
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	66.5
(iv)	Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	66.9
			GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	63.1
(v)	Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	29.1	59.8
			Every 4th	2.3B	1.02s	82.3	42.7	55.1	34.6	68.8
			Every 2nd	2.6B	1.24s	83.7	41.0	55.8	34.5	68.2
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	66.6
			Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	66.7
(vii)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	64.9
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	62.7
(viii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	57.8
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	62.7

Table 3: **Ablation studies.** Each row should be compared to the baseline Flamingo run (top row). Step time measures the time spent to perform gradient updates on all training datasets.


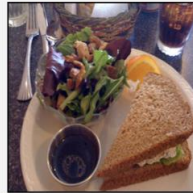
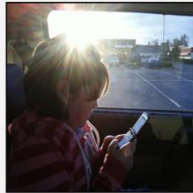
# Limitations

## Functional Limitations

- Hallucinations (Q)
- Poor generalization for long sequences
- Worse than contrastive models in classification
- Sensitivity to examples

## Practical Limitations

- Text interface inconvenient for some tasks
- Expensive to train

Input Prompt			
	Question: What is on the phone screen? Answer:	Question: What can you see out the window? Answer:	Question: Whom is the person texting? Answer:
Output	A text message from a friend.	A parking lot.	The driver.

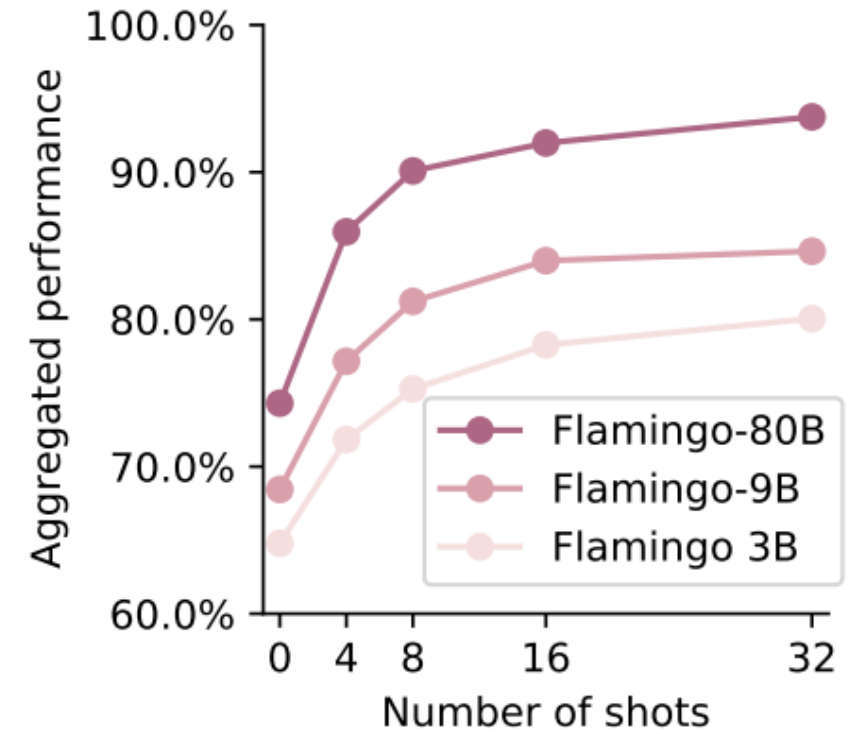
Q: Is the model simply inferring answers through the prompts without using images?



# Limitations

## Learning new task or identifying trained task?

- Performance plateaus as number of examples reach 32
- Non-trivial performance without images (Q)
- Examples may be locating task in memory (Q)
  - “Task Location” [8]



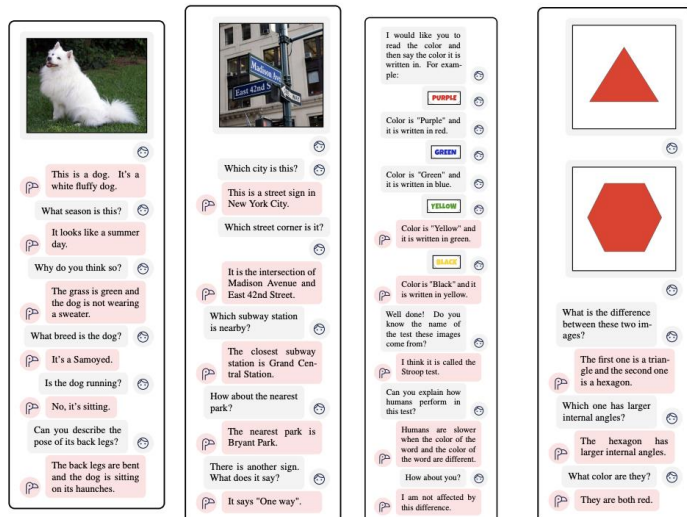
Q: Is the model learning a new task at inference or just identifying a task learned during training?

Q: Is it possible that the model's success is just due to the capabilities of the LM?

# Strengths

## Accessibility

- Few-shot task learning
- Chat interface
  - Non-expert use
  - Handles open-vocabulary prompts
  - Explainability and interpretability



## Reusability

- Repurpose pretrained frozen models
  - Practical and environmental benefits
- New modalities can be introduced
- Only used 5 datasets for design decisions

# Weaknesses

## Performance Dependencies

- Weights of mixture dataset
- Large model size and large pretraining dataset size

## Minor Issues

- Lack of detailed settings on downstream tasks, e.g. will `<image>` token also cross-attend to visual conditions?

- Please do the reading and paper reviews!
  - **First one due Monday Sept 1 11:59pm**