


Language Is Not All You Need

Aligning Perception with Language Models



What's in this picture?

Looks like a duck.

That's not a duck. Then what's it?


Looks more like a bunny.

Why?

It has bunny ears.

Description of three toed woodpecker: It has black and white stripes throughout the body and a yellow crown.

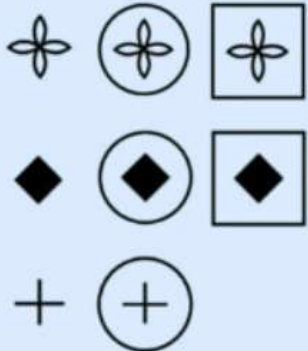
Description of downy woodpecker: It has white spots on its black wings and some red on its crown.






Question: what is the name of the woodpecker in the picture?




Downy

Here are eight images:



The following image is:

A  B  C 

D  E  F 

Multimodal Large Language Models

Multimodal perception is Key step toward artificial general intelligence!

MLLMs: perceive general modalities (text, image, audio...), learn in context (few-shot) & follow instructions(zero-shot) → generate output

KOSMOS-1



What's in this picture?

sausage roll.

How to cook it?

Soak sausage roll in ketchup. Bake in oven for 15 min. Cut and enjoy.

Can I put cheese in the dish?

Sure. But make sure it is melted.



What kind of animal is in the picture?

It's a zebra.

Where do they live in?

In Africa.

Tell me a joke about zebras.

Zebras get in trouble for crossing the road.



What's in this picture?

A screenshot of the Windows 10 shutdown window

I would like to restart my computer. Which button should I click?

Press OK.

Now I would not like to restart. What can I do?

Click Cancel.

Input of KOSMOS-1

Flatten input as sequence decorated with **special tokens**

`<s>` `</s>`: start and end of sequence

→ **Why are they necessary? (except output sequence)**

`<image>` `</image>`: beginning and end of encoded image embeddings

→ model can distinguish between text and image embeddings

Datasets	Format Examples
Text	<code><s></code> KOSMOS-1 can perceive multimodal input, learn in context, and generate output. <code></s></code>
Image-Caption	<code><s></code> <code><image></code> Image Embedding <code></image></code> WALL-E giving potted plant to EVE. <code></s></code>
Multimodal	<code><s></code> <code><image></code> Image Embedding <code></image></code> This is WALL-E. <code><image></code> Image Embedding <code></image></code> This is EVE. <code></s></code>

Table 21: The examples of the data format to train the KOSMOS-1 model.

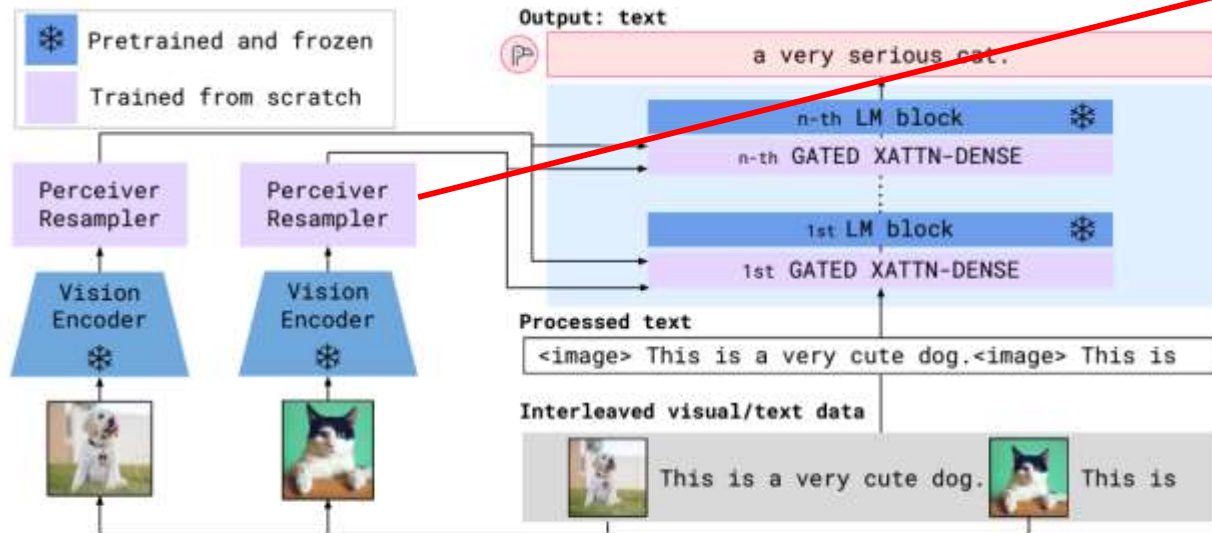
Embedding module

Encode both tokens and other input modalities(image) into vectors

Use a lookup table → map the input tokens into embeddings

Continuous signal → discrete code & regard them as “foreign languages” (BEiT)

Input image → vision encoder (Embedding module) → **Resampler (Flamingo)**



Produces a fixed number of visual outputs
→ computational complexity ↓

Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

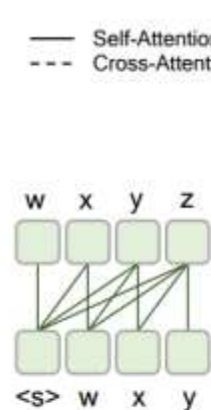
META LM

META LM: Causal, Non causal → semi-causal

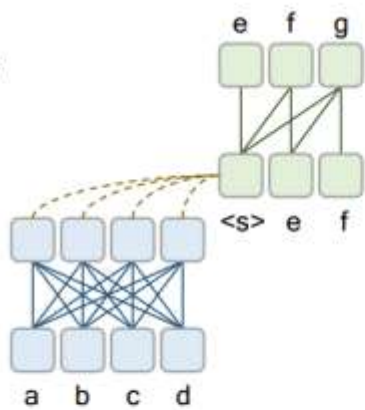
KOSMOS follow META LM → employ a vision encoder as the embedding module

BUT causal model!

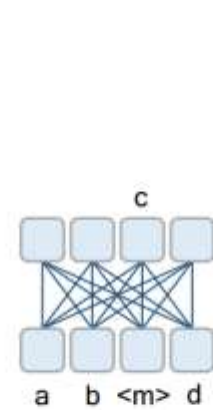
— Self-Attention
--- Cross-Attention



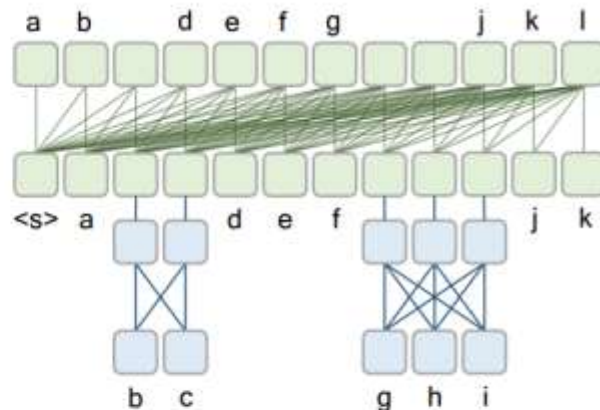
(a) Causal LM
(Unidirectional)



(b) Prefix LM
(Encoder-Decoder
with Cross-Attention)

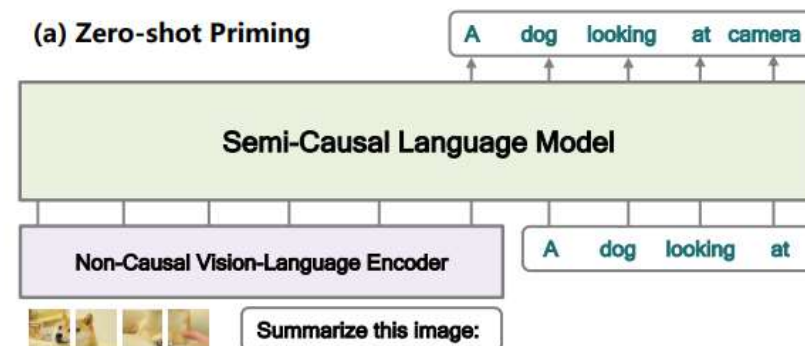


(c) Non-Causal LM
(Bidirectional)

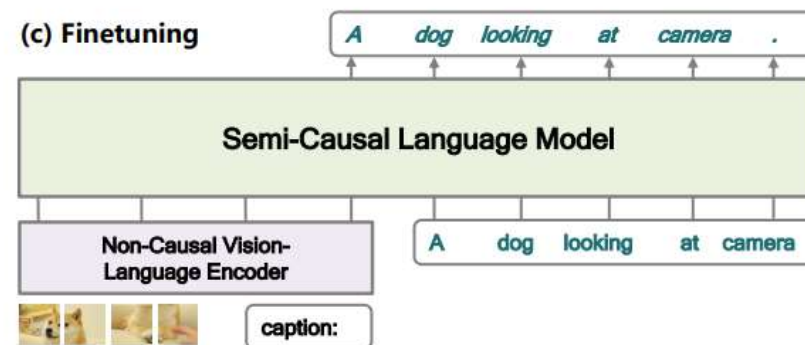


(d) Semi-Causal LM

(a) Zero-shot Priming



(c) Finetuning



Transformer-based decoder

Embeddings of an input sequence → Transformer-based decoder

Auto-regressive manner, produces the next token by conditioning on past timesteps

causal masking → mask out future information

softmax classifier → generate tokens over the vocabulary

KOSMOS-1

in-context learning and instruction following (Language models naturally inherit)

perception is aligned with language models (training on multimodal corpora)

Transformer architecture

TorchScale library: designed for large scale model training, adopts **Magneto** as backbone

→ Improve **Scalability**

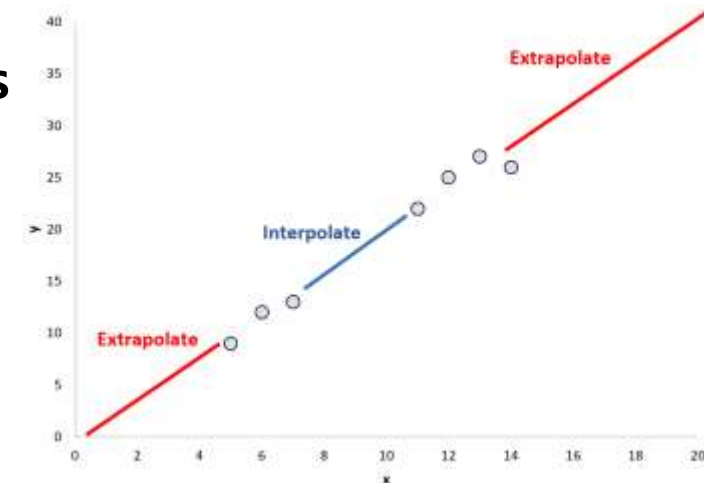
= Improve generality, stability, and efficiency during scaling up the model size

xPos: relative position encoding for better **long-context modeling**

Ex) training on short while testing on longer sequences

Optimizes attention resolution → position information can be captured more precisely

Efficient and effective in both interpolation and extrapolation settings

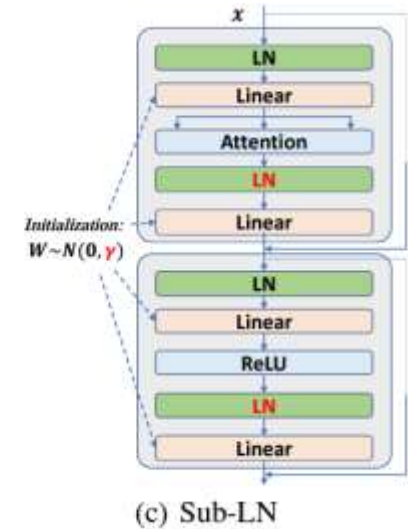
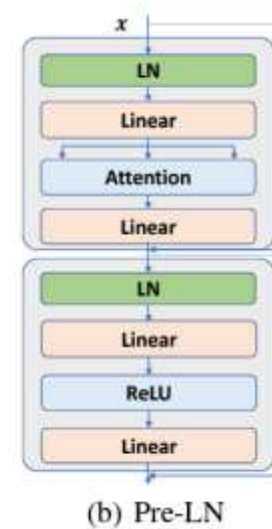
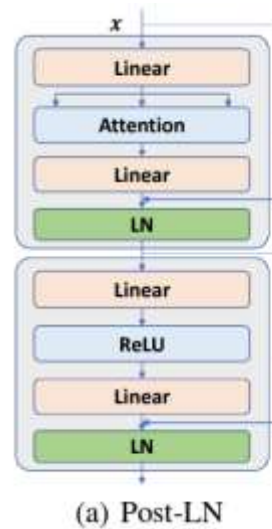


TorchScale

Library designed for large scale model training, adopts **Magneto** as backbone

By using extra LayerNorm to each sublayer

xPos



Training data

Training on web-scale multimodal corpora

1. Text Corpora: Pile, Common Crawl, RealNews datasets...

→ purged of duplicate, filtered to exclude downstream task data

2. Image-Caption pairs: LAION-2B, LAION-400M, COYO-700M...

These are made by using Common Crawl

3. Interleaved Image-Text Data: from Common Crawl

Limit number of images to 5, discard half of the documents that only have one image...

To reduce noise and redundancy & increase the diversity

Datasets	Tokens (billion)	Weight (%)	Epochs
OpenWebText2	14.8	21.8%	1.47
CC-2021-04	82.6	17.7%	0.21
Books3	25.7	16.2%	0.63
CC-2020-50	68.7	14.7%	0.21
Pile-CC	49.8	10.6%	0.21
Realnews	21.9	10.2%	0.46
Wikipedia	4.2	5.4%	1.29
BookCorpus2	1.5	1.1%	0.75
Gutenberg (PG-19)	2.7	1.0%	0.38
CC-Stories	5.3	1.0%	0.19
NIH ExPorter	0.3	0.2%	0.75

Table 20: Language datasets used to train the KOSMOS-1 model.

Training Setup

24 layers with 2,048 hidden dimensions, 8,192 FFN intermediate size, and 32 attention heads

use **Magneto's** initialization for optimization stability for optimization stability

Pretrained **CLIP ViT-L/14** model with 1,024 feature dimensions
→ frozen except for the last layer

batch size is 1.2 million tokens (0.5 0.5 0.2), 300k steps, 375 warming-up steps

SentencePiece → tokenize the text

RoBERT → preprocess the data in the “full-sentence” format

:packs each input sequence with full sentences that are sampled continuously from one or more documents

Language-Only Instruction Tuning

Presented by FLAN

Chat GPT, GPT-3... used instruction Tuning

Instruction dataset:

Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor, FLANv2

instructions, inputs, and outputs

→ predict outputs

instructions and inputs loss X

Example	Correct?
Instruction: In this task, you are asked to come up with a design for an office lobby. The client has provided specific instructions on what they want in the lobby, and it is your job to create a floor plan that meets all of their requirements. Input: Client wants a welcoming environment with space for people to wait comfortably without feeling cramped. They also would like some sort of area where people can do work if they need to kill some time before their next meeting. Output: [...] Based on these requirements, a possible floor plan [...] The lobby would have a reception area [...] with access to outlets, and a coffee station.	✓
Instruction: You will be given several pieces of information about an event, and you have to determine whether or not it is a cause-and-effect relationship. If the given statements are related by cause and effect, then output 'True'. Otherwise, output 'False'. Input: Statements: ['The tornado damaged the city', 'Many people were left homeless']. Output: True	✓
Instruction: You are to determine if the statement is true or false. Input: Text: Santa Claus lives at the North Pole. Output: False	✗
Instruction: You are given a scientific experiment scenario and your job is to determine which variable(s) the scientist should control in order to test the hypothesis. Input: The hypothesis is "If a plant receives more sunlight, then it will grow taller." Which variable(s) should the scientist control? Output: A	✗

Result

Does the model really “perceive” images?

Or just effect of the text in image caption or Interleaved data

→ Image encoder ablation study

Is MLLM better than LLM in language task?

Effect of instruction training (text only) on image recognition