

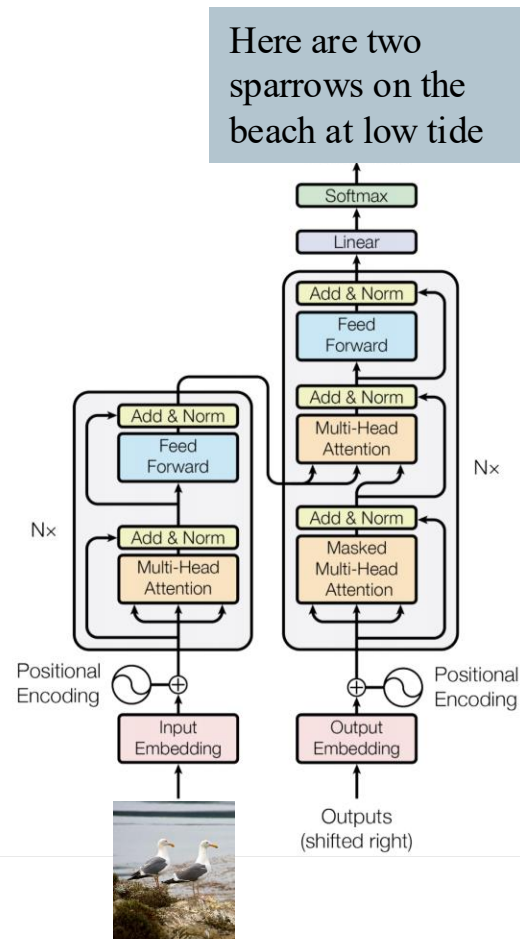
Vision-Language Models

Part II:

VLMs using LLMs

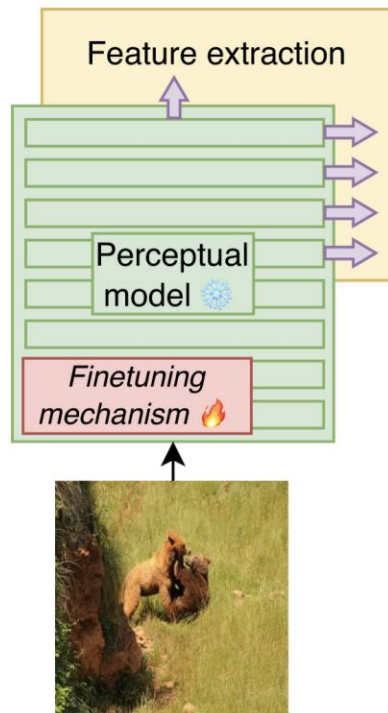
1. Vision-Language Models in the era of LLMs

- Unimodal models with connection
- *One model for all*



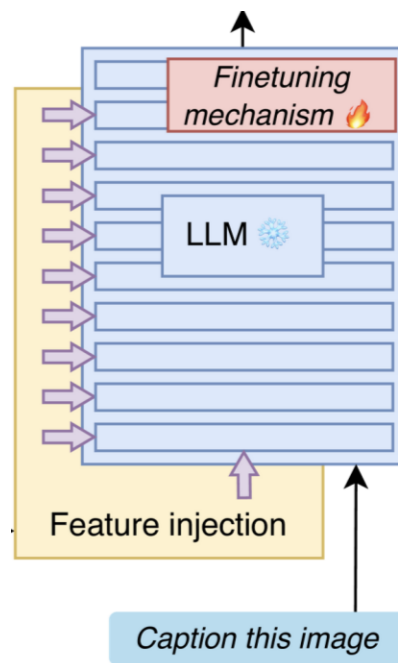
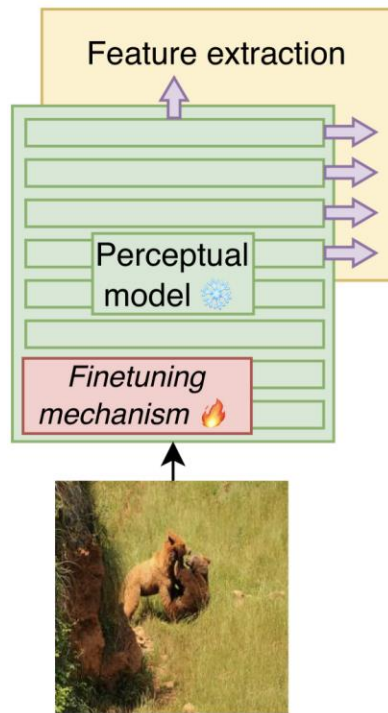
Vision Encoder + LLM Decoder

Image as input, textual caption as output



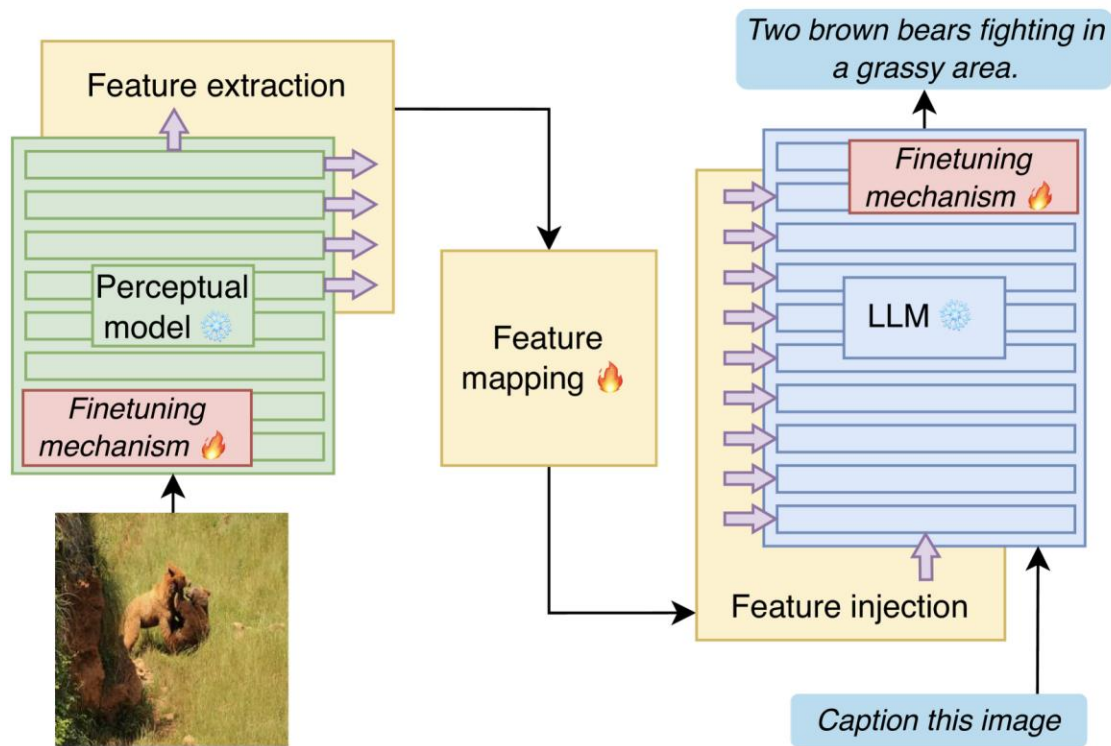
Vision Encoder + LLM Decoder

Image as input, textual caption as output



Vision Encoder + LLM Decoder

Image as input, textual caption as output

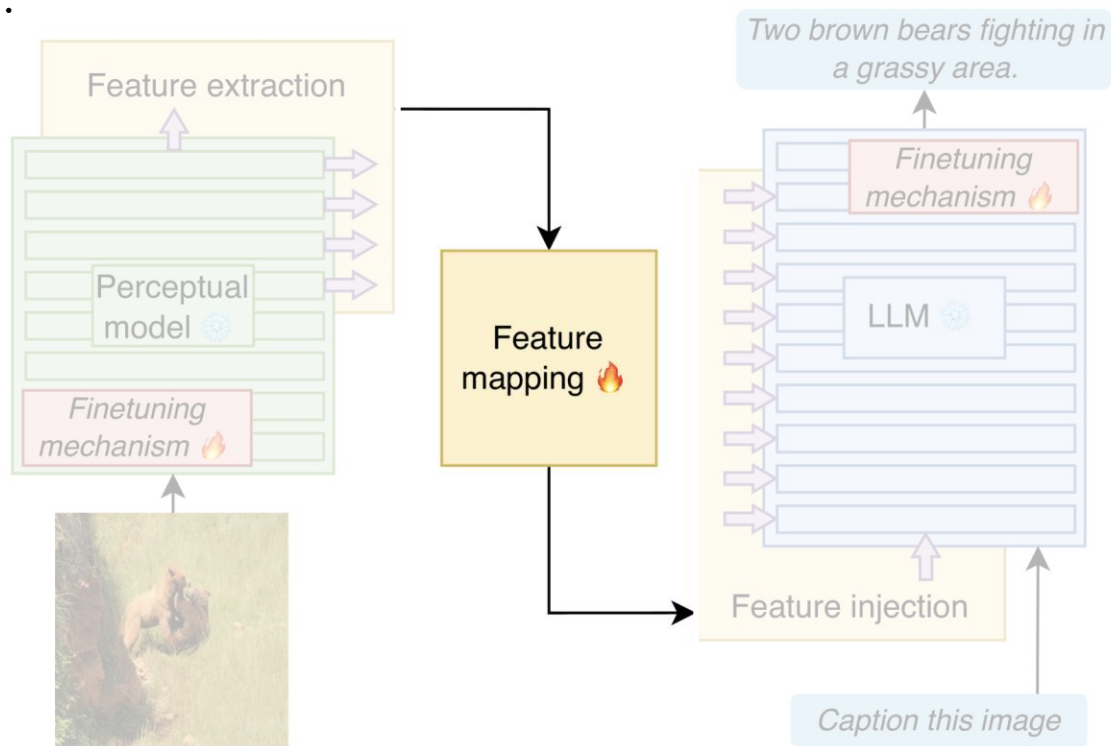


Why this modeling? Because the best LLM ever designed (and the plug&play update if a new LLM is released)

Vision Encoder + LLM Decoder

Feature mapping module?

A classic MLP, or:

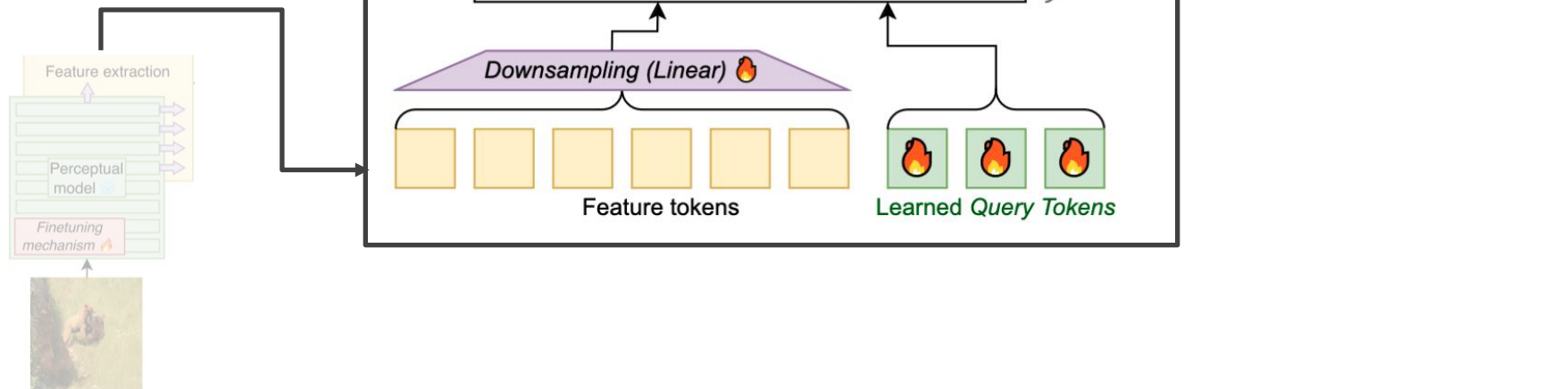


Vision Encoder + LLM Decoder

Feature mapping module?

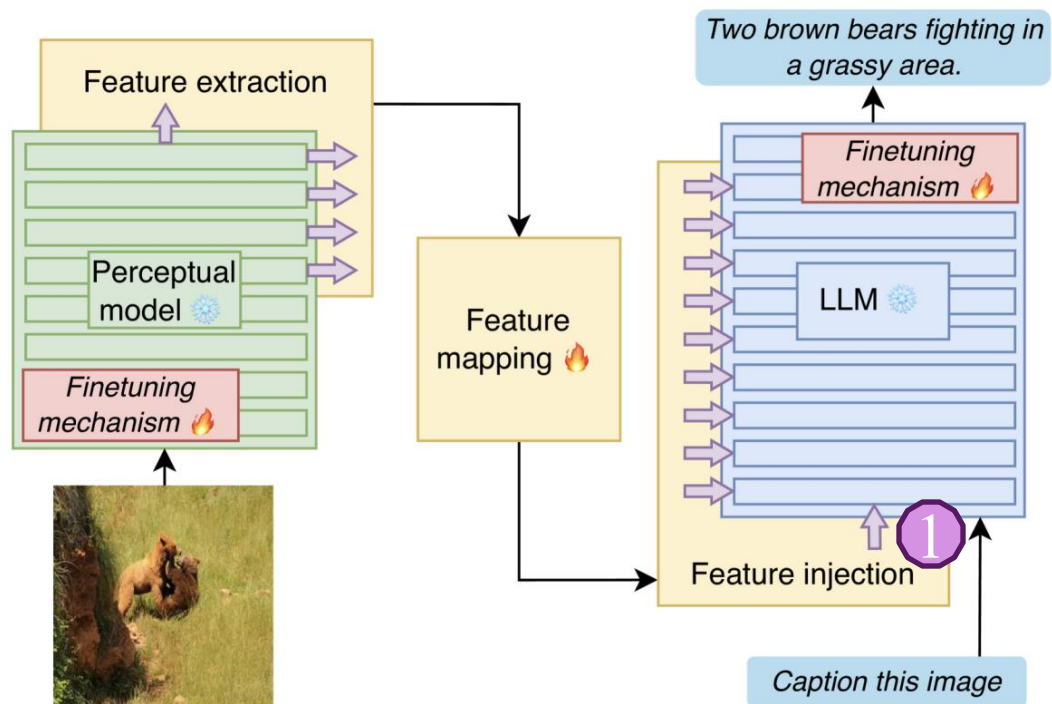
A classic MLP,

Or ViT like with extra tokens



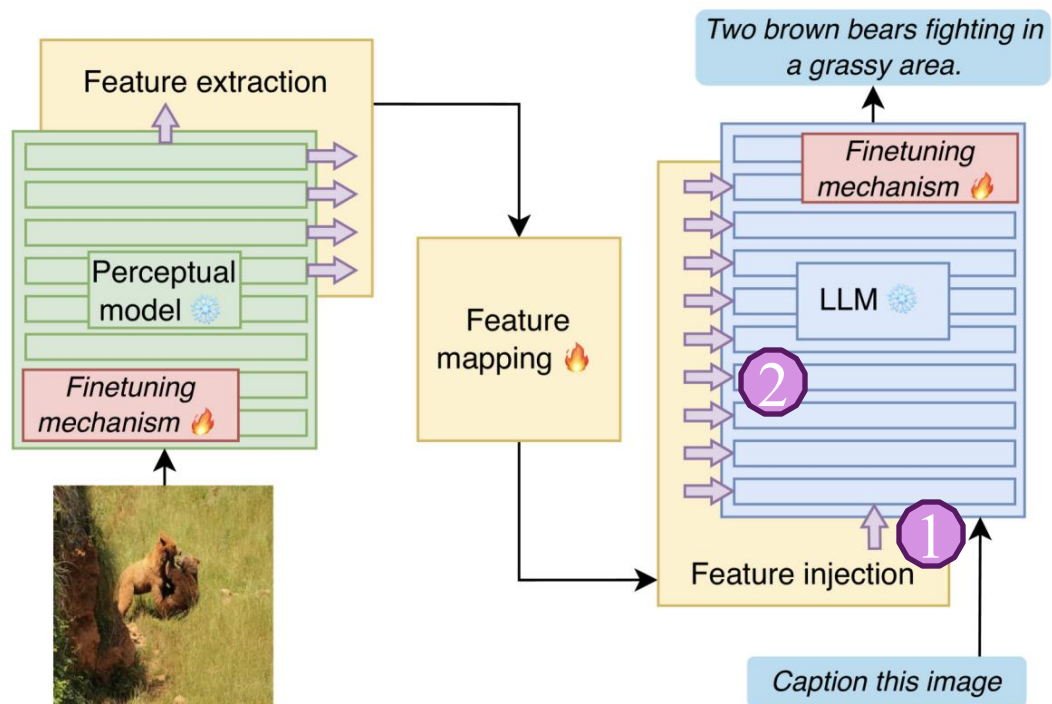
Vision Encoder + LLM Decoder

After feature mapping, feature **injection**!



Vision Encoder + LLM Decoder

After feature mapping, feature **injection**!



input: img
output: text caption

img → visual CLIP encoder → features extraction

from last CLIP layer
or n last layers

FEATURE MAPPING *

it can be before
the LLM or in the
cross-attention of
intermediate blocks

feature injection

prompt: "add a caption" →

LLM

caption

Another model that
learns a mapping:

ViT with extra tokens:
(or MLP)

FEATURES + LEARNED QUERY TOKENS

to match:
transformer's
dimension

downsample
transformer blocks
upsample

match LLM's
dimension

In classification we add
a CLS token for the head,
same thing for the added query tokens
(to pull all the visual information
towards the added tokens)

~~FEATURES~~ + ~~LEARNED QUERY TOKENS~~ ✓
in LLM

Vision encoder + LLM decoder can be modified in any step:

- ideal: freeze LLM, backbone

train mapping in a limited training set

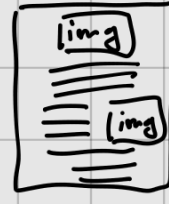
- many unimodal models → multimodal

- build a huge multimodal dataset

to scale (Billions)

- train (vision encoder → mapping → visual tokens → LLM)

- works very well when the output is text



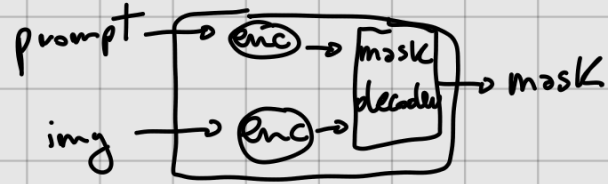
or OFA (One For All)

4M (Massively Multimodal Masked Modeling)

any input \rightarrow any output

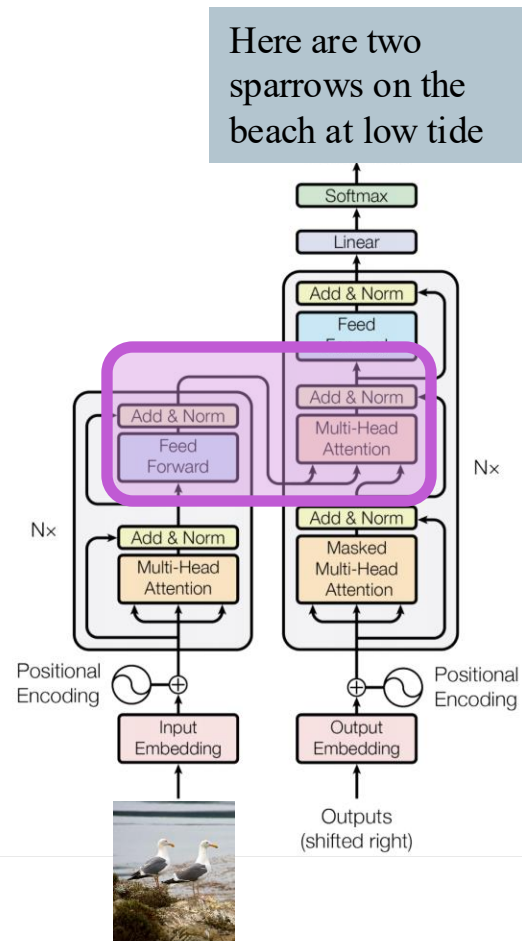
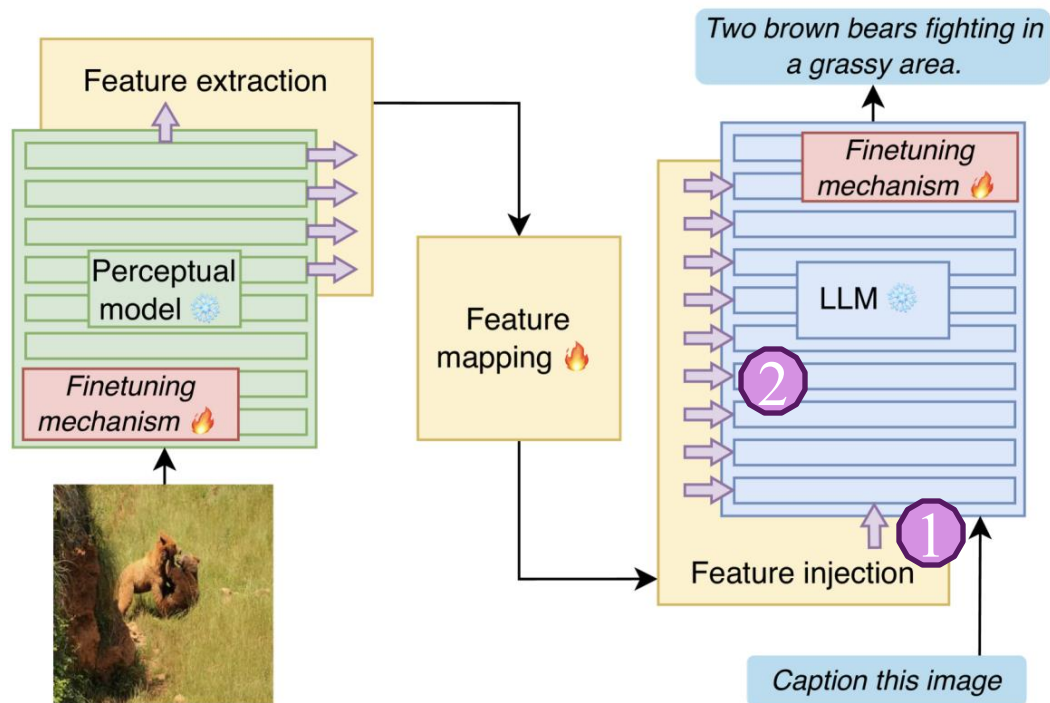
composed by many
models that can
solve many tasks

like SAM (Segment Anything Model)

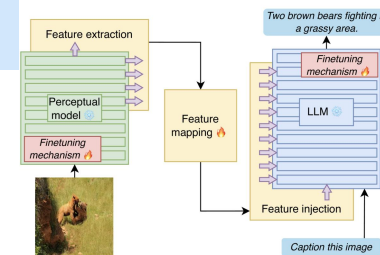


Vision Encoder + LLM Decoder

After feature mapping, feature **injection**!

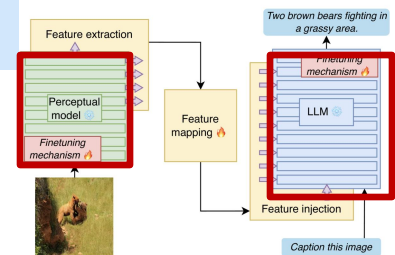


Vision Encoder + LLM Decoder



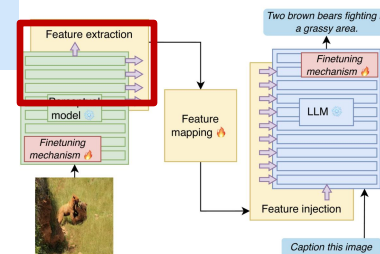
Method	Backbones		Adaptation mechanism				# Tr.
	LLMs	Perceptual Enc.	Feature extraction	Feature mapping	Feature injection	Fine-tuning mechanisms	params.
Flamingo [1]	Chinchilla [33]	NFNet [5]	Tokens from last layer	Perceiver Resampler (Transformer)	GATED XATTN-DENSE (Cross-attention)	–	10B
BLIP-2 [43]	OPT [92], FlanT5 [13]	CLIP [65]	Tokens from last layer	Q-Former	1st layer token injection	–	1.2B
MAGMA [22]	GPT-J 6B [86]	CLIP [65] / NFNet [5]	Tokens from last layer	MLP	1st layer token injection	fine-tuning of perceptual model	243M
MAPL [58]	GPT-J 6B [86]	CLIP-L [65]	Tokens from last layer	QPMapper ($d_{\text{embed}}=256$, 4 layers)	1st layer token injection	–	3.4M
PromptFuse [46]	BART [42]	ViT [19]	Tokens from last layer	<i>nothing</i>	–	prompt tuning	15K
LiMBer [60]	GTP-J 6B [86]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	–	12.5M
eP-ALM [72]	OPT-2.7B/6.7B [92]	ViT [77], AST [27], TimeFormer [4]	CLS tokens from n last layers	(Shared) linear projection	Token injection in intermediate layers	prompt tuning	4.2M
LLaMA-Adapter [25, 91]	LLaMA[82]	CLIP [65]	Tokens from last layer	Linear projection	Token injection in intermediate layers	inner-layer prompt tuning, bias tuning, norm tuning	14M
Frozen [84]	GPT-like [66]	NFNet [5]	Pooled output tokens	<i>nothing</i>	1st layer token injection	Fine-tune the NFNet	40.3M
ClipCap [61]	GPT-2[66]	CLIP [65]	Tokens from last layer	Transformer	1st layer token injection	–	43M
VL-Adapter [79]	BART [42], T5 [67]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	Adapters	5.8M
AnyMAL [62]	Llama 2-70B-chat [83],	CLIP [65], CLAP [23]	Tokens from last layer	Perceiver Resampler, or linear projection	1st layer token injection	LoRA [34]	–
DePALM ^{QP,inner}	OPT-6.7B [92], LLaMA [82]	CLIP-L [65], DINOv2 [63], MAViL [36] TimeFormer [4]	Tokens from n last layers	QPMapper	Token injection in intermediate layers	prompt tuning	18.1M
DePALM							17.9M
DePALM ^{R-rand,L0} , DePALM ^{R-linear,L0} , DePALM ^{R-QPMapper,L0} , DePALM ^{R-avgpool,L0}			Tokens from last layer	Linear projection + Resampler	1st layer token injection		21M, 88M 18M, 21M
DePALM ^{c-attn}			Tokens from n last layers	Projection + Small Transformer	Gated cross-attention		17.9M

Vision Encoder + LLM Decoder



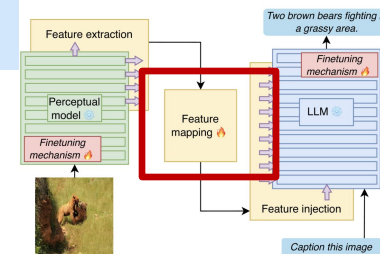
Method	Backbones		Adaptation mechanism				# Tr.
	LLMs	Perceptual Enc.	Feature extraction	Feature mapping	Feature injection	Fine-tuning mechanisms	params.
Flamingo [1]	Chinchilla [33]	NFNet [5]	Tokens from last layer	Perceiver Resampler (Transformer)	GATED XATTN-DENSE (Cross-attention)	–	10B
BLIP-2 [43]	OPT [92], FlanT5 [13]	CLIP [65]	Tokens from last layer	Q-Former	1st layer token injection	–	1.2B
MAGMA [22]	GPT-J 6B [86]	CLIP [65] / NFNet [5]	Tokens from last layer	MLP	1st layer token injection	fine-tuning of perceptual model	243M
MAPL [58]	GPT-J 6B [86]	CLIP-L [65]	Tokens from last layer	QPMapper ($d_{\text{embed}}=256$, 4 layers)	1st layer token injection	–	3.4M
PromptFuse [46]	BART [42]	ViT [19]	Tokens from last layer	<i>nothing</i>	–	prompt tuning	15K
LiMBer [60]	GTP-J 6B [86]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	–	12.5M
eP-ALM [72]	OPT-2.7B/6.7B [92]	ViT [77], AST [27], TimeFormer [4]	CLS tokens from n last layers	(Shared) linear projection	Token injection in intermediate layers	prompt tuning	4.2M
LLaMA-Adapter [25, 91]	LLaMA[82]	CLIP [65]	Tokens from last layer	Linear projection	Token injection in intermediate layers	inner-layer prompt tuning, bias tuning, norm tuning	14M
Frozen [84]	GPT-like [66]	NFNet [5]	Pooled output tokens	<i>nothing</i>	1st layer token injection	Fine-tune the NFNet	40.3M
ClipCap [61]	GPT-2[66]	CLIP [65]	Tokens from last layer	Transformer	1st layer token injection	–	43M
VL-Adapter [79]	BART [42], T5 [67]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	Adapters	5.8M
AnyMAL [62]	Llama 2-70B-chat [83],	CLIP [65], CLAP [23]	Tokens from last layer	Perceiver Resampler, or linear projection	1st layer token injection	LoRA [34]	–
DePALM ^{QP,inner}	OPT-6.7B [92], LLaMA [82]	CLIP-L [65], DINOv2 [63], MAViL [36] TimeFormer [4]	Tokens from n last layers	QPMapper	Token injection in intermediate layers	prompt tuning	18.1M
DePALM							17.9M
DePALM ^{R-rand,L0} , DePALM ^{R-linear,L0} , DePALM ^{R-QPMapper,L0} , DePALM ^{R-avgpool,L0}			Tokens from last layer	Linear projection + Resampler	1st layer token injection		21M, 88M 18M, 21M
DePALM ^{c-attn}			Tokens from n last layers	Projection + Small Transformer	Gated cross-attention		17.9M

Vision Encoder + LLM Decoder



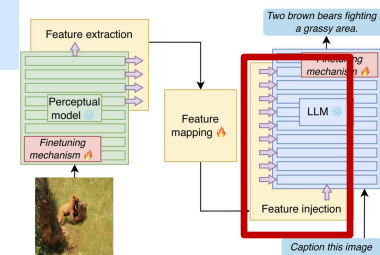
Method	Backbones		Adaptation mechanism				# Tr. params.
	LLMs	Perceptual Enc.	Feature extraction	Feature mapping	Feature injection	Fine-tuning mechanisms	
Flamingo [1]	Chinchilla [33]	NFNet [5]	Tokens from last layer	Perceiver Resampler (Transformer)	GATED XATTN-DENSE (Cross-attention)	–	10B
BLIP-2 [43]	OPT [92], FlanT5 [13]	CLIP [65]	Tokens from last layer	Q-Former	1st layer token injection	–	1.2B
MAGMA [22]	GPT-J 6B [86]	CLIP [65] / NFNet [5]	Tokens from last layer	MLP	1st layer token injection	fine-tuning of perceptual model	243M
MAPL [58]	GPT-J 6B [86]	CLIP-L [65]	Tokens from last layer	QPMapper ($d_{\text{embed}}=256$, 4 layers)	1st layer token injection	–	3.4M
PromptFuse [46]	BART [42]	ViT [19]	Tokens from last layer	<i>nothing</i>	–	prompt tuning	15K
LiMBer [60]	GTP-J 6B [86]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	–	12.5M
eP-ALM [72]	OPT-2.7B/6.7B [92]	ViT [77], AST [27], TimeFormer [4]	CLS tokens from n last layers	(Shared) linear projection	Token injection in intermediate layers	prompt tuning	4.2M
LLaMA-Adapter [25, 91]	LLaMA[82]	CLIP [65]	Tokens from last layer	Linear projection	Token injection in intermediate layers	inner-layer prompt tuning, bias tuning, norm tuning	14M
Frozen [84]	GPT-like [66]	NFNet [5]	Pooled output tokens	<i>nothing</i>	1st layer token injection	Fine-tune the NFNet	40.3M
ClipCap [61]	GPT-2[66]	CLIP [65]	Tokens from last layer	Transformer	1st layer token injection	–	43M
VL-Adapter [79]	BART [42], T5 [67]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	Adapters	5.8M
AnyMAL [62]	Llama 2-70B-chat [83],	CLIP [65], CLAP [23]	Tokens from last layer	Perceiver Resampler, or linear projection	1st layer token injection	LoRA [34]	–
DePALM ^{QP,inner}	OPT-6.7B [92], LLaMA [82]	CLIP-L [65], DINOv2 [63], MAViL [36] TimeFormer [4]	Tokens from n last layers	QPMapper	Token injection in intermediate layers	prompt tuning	18.1M
DePALM							17.9M
DePALM ^{R-rand,L0} , DePALM ^{R-linear,L0} , DePALM ^{R-QPMapper,L0} , DePALM ^{R-avgpool,L0}			Tokens from last layer	Linear projection + Resampler	1st layer token injection		21M, 88M 18M, 21M
DePALM ^{c-attn}			Tokens from n last layers	Projection + Small Transformer	Gated cross-attention		17.9M

Vision Encoder + LLM Decoder



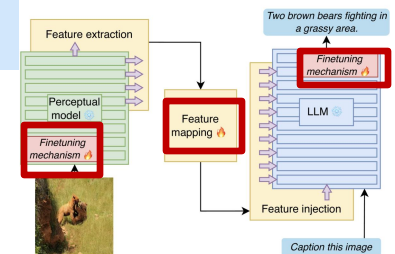
Method	Backbones		Adaptation mechanism				# Tr. params.
	LLMs	Perceptual Enc.	Feature extraction	Feature mapping	Feature injection	Fine-tuning mechanisms	
Flamingo [1]	Chinchilla [33]	NFNet [5]	Tokens from last layer	Perceiver Resampler (Transformer)	GATED XATTN-DENSE (Cross-attention)	–	10B
BLIP-2 [43]	OPT [92], FlanT5 [13]	CLIP [65]	Tokens from last layer	Q-Former	1st layer token injection	–	1.2B
MAGMA [22]	GPT-J 6B [86]	CLIP [65] / NFNet [5]	Tokens from last layer	MLP	1st layer token injection	fine-tuning of perceptual model	243M
MAPL [58]	GPT-J 6B [86]	CLIP-L [65]	Tokens from last layer	QPMapper ($d_{\text{embed}}=256$, 4 layers)	1st layer token injection	–	3.4M
PromptFuse [46]	BART [42]	ViT [19]	Tokens from last layer	nothing	–	prompt tuning	15K
LiMBer [60]	GTP-J 6B [86]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	–	12.5M
eP-ALM [72]	OPT-2.7B/6.7B [92]	ViT [77], AST [27], TimeFormer [4]	CLS tokens from n last layers	(Shared) linear projection	Token injection in intermediate layers	prompt tuning	4.2M
LLaMA-Adapter [25, 91]	LLaMA[82]	CLIP [65]	Tokens from last layer	Linear projection	Token injection in intermediate layers	inner-layer prompt tuning, bias tuning, norm tuning	14M
Frozen [84]	GPT-like [66]	NFNet [5]	Pooled output tokens	nothing	1st layer token injection	Fine-tune the NFNet	40.3M
ClipCap [61]	GPT-2[66]	CLIP [65]	Tokens from last layer	Transformer	1st layer token injection	–	43M
VL-Adapter [79]	BART [42], T5 [67]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	Adapters	5.8M
AnyMAL [62]	Llama 2-70B-chat [83],	CLIP [65], CLAP [23]	Tokens from last layer	Perceiver Resampler, or linear projection	1st layer token injection	LoRA [34]	–
DePALM ^{QP,inner}	OPT-6.7B [92], LLaMA [82]	CLIP-L [65], DINOv2 [63], MAViL [36] TimeFormer [4]	Tokens from n last layers	QPMapper	Token injection in intermediate layers	prompt tuning	18.1M
DePALM							17.9M
DePALM ^{R-rand,L0} , DePALM ^{R-linear,L0} , DePALM ^{R-QPMapper,L0} , DePALM ^{R-avgpool,L0}			Tokens from last layer	Linear projection + Resampler	1st layer token injection		21M, 88M, 18M, 21M
DePALM ^{c-attn}			Tokens from n last layers	Projection + Small Transformer	Gated cross-attention		17.9M

Vision Encoder + LLM Decoder



Method	Backbones		Adaptation mechanism				# Tr.
	LLMs	Perceptual Enc.	Feature extraction	Feature mapping	Feature injection	Fine-tuning mechanisms	params.
Flamingo [1]	Chinchilla [33]	NFNet [5]	Tokens from last layer	Perceiver Resampler (Transformer)	GATED XATTN-DENSE (Cross-attention)	–	10B
BLIP-2 [43]	OPT [92], FlanT5 [13]	CLIP [65]	Tokens from last layer	Q-Former	1st layer token injection	–	1.2B
MAGMA [22]	GPT-J 6B [86]	CLIP [65] / NFNet [5]	Tokens from last layer	MLP	1st layer token injection	fine-tuning of perceptual model	243M
MAPL [58]	GPT-J 6B [86]	CLIP-L [65]	Tokens from last layer	QPMapper ($d_{\text{embed}}=256$, 4 layers)	1st layer token injection	–	3.4M
PromptFuse [46]	BART [42]	ViT [19]	Tokens from last layer	nothing	–	prompt tuning	15K
LiMBer [60]	GTP-J 6B [86]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	–	12.5M
eP-ALM [72]	OPT-2.7B/6.7B [92]	ViT [77], AST [27], TimeFormer [4]	CLS tokens from n last layers	(Shared) linear projection	Token injection in intermediate layers	prompt tuning	4.2M
LLaMA-Adapter [25, 91]	LLaMA[82]	CLIP [65]	Tokens from last layer	Linear projection	Token injection in intermediate layers	inner-layer prompt tuning, bias tuning, norm tuning	14M
Frozen [84]	GPT-like [66]	NFNet [5]	Pooled output tokens	nothing	1st layer token injection	Fine-tune the NFNet	40.3M
ClipCap [61]	GPT-2[66]	CLIP [65]	Tokens from last layer	Transformer	1st layer token injection	–	43M
VL-Adapter [79]	BART [42], T5 [67]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	Adapters	5.8M
AnyMAL [62]	Llama 2-70B-chat [83],	CLIP [65], CLAP [23]	Tokens from last layer	Perceiver Resampler, or linear projection	1st layer token injection	LoRA [34]	–
DePALM ^{QP,inner}	OPT-6.7B [92], LLaMA [82]	CLIP-L [65], DINOv2 [63], MAViL [36] TimeFormer [4]	Tokens from n last layers	QPMapper	Token injection in intermediate layers	prompt tuning	18.1M
DePALM							17.9M
DePALM ^{R-rand,L0} , DePALM ^{R-linear,L0} , DePALM ^{R-QPMapper,L0} , DePALM ^{R-avgpool,L0}			Tokens from last layer	Linear projection + Resampler	1st layer token injection		21M, 88M 18M, 21M
DePALM ^{c-attn}			Tokens from n last layers	Projection + Small Transformer	Gated cross-attention		17.9M

Vision Encoder + LLM Decoder



Method	Backbones		Adaptation mechanism				# Tr. params.
	LLMs	Perceptual Enc.	Feature extraction	Feature mapping	Feature injection	Fine-tuning mechanisms	
Flamingo [1]	Chinchilla [33]	NFNet [5]	Tokens from last layer	Perceiver Resampler (Transformer)	GATED XATTN-DENSE (Cross-attention)	–	10B
BLIP-2 [43]	OPT [92], FlanT5 [13]	CLIP [65]	Tokens from last layer	Q-Former	1st layer token injection	–	1.2B
MAGMA [22]	GPT-J 6B [86]	CLIP [65] / NFNet [5]	Tokens from last layer	MLP	1st layer token injection	fine-tuning of perceptual model	243M
MAPL [58]	GPT-J 6B [86]	CLIP-L [65]	Tokens from last layer	QPMapper ($d_{\text{embed}}=256$, 4 layers)	1st layer token injection	–	3.4M
PromptFuse [46]	BART [42]	ViT [19]	Tokens from last layer	<i>nothing</i>	–	prompt tuning	15K
LiMBer [60]	GTP-J 6B [86]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	–	12.5M
eP-ALM [72]	OPT-2.7B/6.7B [92]	ViT [77], AST [27], TimeFormer [4]	CLS tokens from n last layers	(Shared) linear projection	Token injection in intermediate layers	prompt tuning	4.2M
LLaMA-Adapter [25, 91]	LLaMA[82]	CLIP [65]	Tokens from last layer	Linear projection	Token injection in intermediate layers	inner-layer prompt tuning, bias tuning, norm tuning	14M
Frozen [84]	GPT-like [66]	NFNet [5]	Pooled output tokens	<i>nothing</i>	1st layer token injection	Fine-tune the NFNet	40.3M
ClipCap [61]	GPT-2[66]	CLIP [65]	Tokens from last layer	Transformer	1st layer token injection	–	43M
VL-Adapter [79]	BART [42], T5 [67]	CLIP [65]	Tokens from last layer	Linear projection	1st layer token injection	Adapters	5.8M
AnyMAL [62]	Llama 2-70B-chat [83],	CLIP [65], CLAP [23]	Tokens from last layer	Perceiver Resampler, or linear projection	1st layer token injection	LoRA [34]	–
DePALM ^{QP,inner}	OPT-6.7B [92], LLaMA [82]	CLIP-L [65], DINOv2 [63], MAViL [36] TimeFormer [4]	Tokens from n last layers	QPMapper	Token injection in intermediate layers	prompt tuning	18.1M
DePALM							17.9M
DePALM ^{R-rand,L0} , DePALM ^{R-linear,L0} , DePALM ^{R-QPMapper,L0} , DePALM ^{R-avgpool,L0}			Tokens from last layer	Linear projection + Resampler	1st layer token injection		21M, 88M 18M, 21M
DePALM ^{c-attn}			Tokens from n last layers	Projection + Small Transformer	Gated cross-attention		17.9M

Vision Encoder + LLM Decoder

Parameter efficient approaches:

Leave the LLM and backbone frozen,

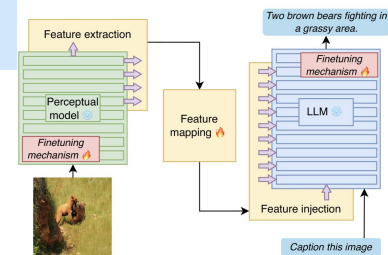
Train the mapping on (very) limited training sets to obtain very good results

Simple design choices works best!

ie. passing all perceptual tokens at the input to the LLM

compress perceptual to a few “summary tokens”

4 times faster to train and on par results

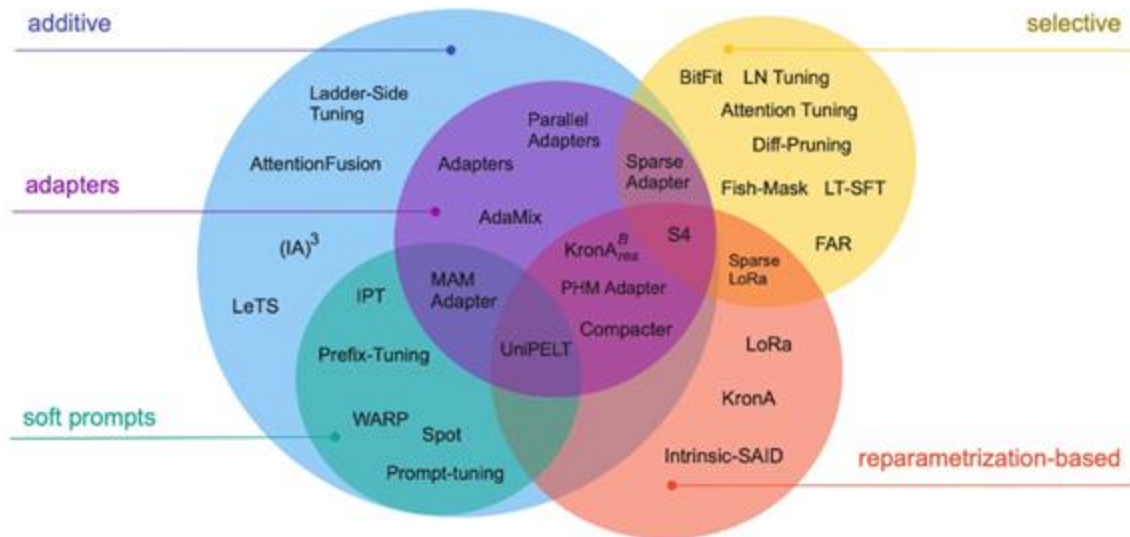
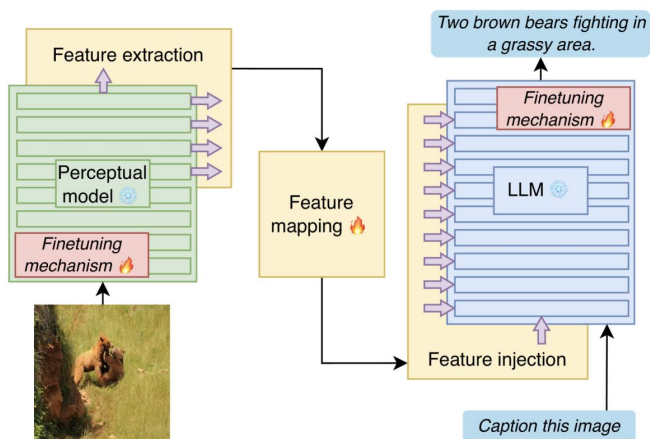


Vision Encoder + LLM Decoder

Many things to do on top of (pretrained) foundations models (if/when available)

Leverage **unimodal** models to build efficient **multimodal** models works well

Efficient finetuning: parameter efficiency, data efficiency, ...



Vision Encoder + LLM Decoder

How to get the best VLM?

Relax the **efficiency** constraint

1/ Build a huge multimodal dataset

Vision Encoder + LLM Decoder

How to get the best VLM?

Relax the **efficiency** constraint

1/ Build a huge multimodal dataset

Image-Text Pairs

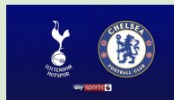


Tottenham vs Chelsea Live Streaming



Tottenham Spurs vs Chelsea Live Streaming

Multimodal Document



The match between Tottenham Spurs vs Chelsea will kick off from 16:30 at Tottenham Hotspur Stadium, London.



The derby had been played 54 times and the Blues have dominated the Spurs. Out of 54 matches played, Chelsea has won 28 times and Spurs had only won 7 times. The remaining 19 matches had ended in draw.

However, in recent 5 meetings, Spurs had won 3 times where Chelsea had won the other two times. ...

+Add synthesized data ...

Dataset	Images	% unique images	Docs	Tokens	Open
KOSMOS-1	-	-	71M	-	✗
M3W	185M	-	43M	-	✗
mmc4-ff	385M	60.6%	79M	34B	✓
mmc4	585M	-	103M	43B	✓
OBELICS	353M	84.3%	141M	115B	✓

Table 1: General statistics of **OBELICS** and the current largest alternatives.

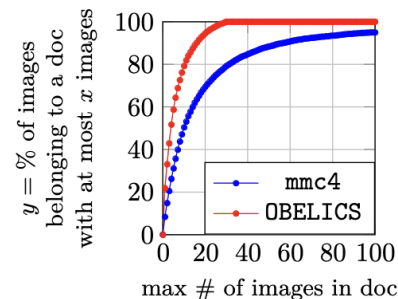


Figure 3: Distribution of images.

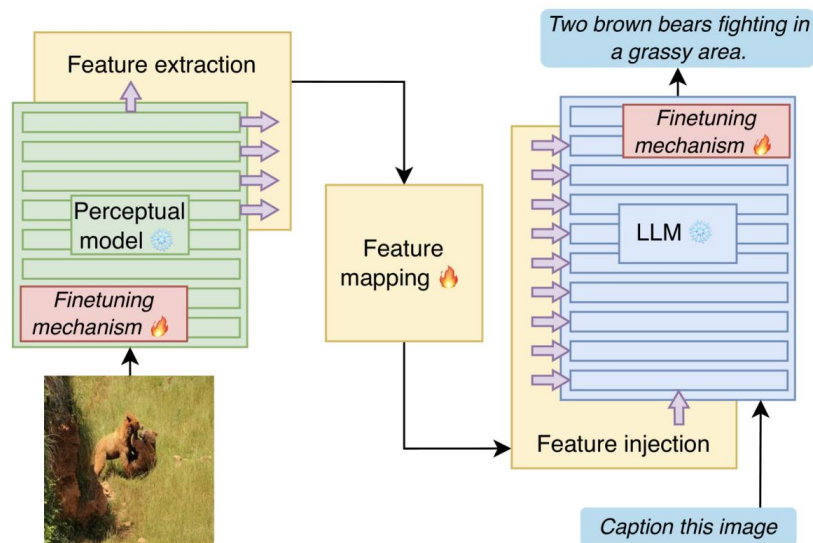
Vision Encoder + LLM Decoder

How to get the best VLM?

Relax the **efficiency** constraint

1/ Build a huge multimodal dataset

2/ Train your best model:



Best architecture?

Vision encoder

Feature Mapping to the LLM input space

Visual tokens (64 in our standard configuration)

interleaved with the input sequence of text embeddings

LLM

Vision Encoder + LLM Decoder

Evaluation very important, not easy for Generative models


Quantitative results:

Model	Size	Archi.	# tokens per image	VQAv2	TextVQA	OKVQA	COCO
OpenFlamingo	9B	CA	-	54.8	29.1	41.1	96.3
Idefics1	9B	CA	-	56.4	27.5	47.7	97.0
Flamingo	9B	CA	-	58.0	33.6	50.0	99.0
MM1	7B	FA	144	63.6	46.3	51.4	116.3
Idefics2-base	8B	FA	64	70.3	57.9	54.6	116.0

Qualitative results:

Prompt

Describe the image



Idefics2 output

The image shows two golden retriever puppies sitting in a field of flowers. They are sitting next to each other, looking at the camera, and appear to be very happy. The puppies are adorable, and their fur is a beautiful golden color. The flowers surrounding them are yellow and add a vibrant touch to the scene.

Model	Size	# tokens per image	MMMU	MathVista	TextVQA	MMBench
LLaVA-NeXT	13B	2880	36.2/-	35.3	67.1	70.0
DeepSeek-VL	7B	576	36.6/-	36.1	64.4	73.2
MM1-Chat	7B	720	37.0/35.6	35.9	72.8	72.3
Idefics2	8B	64	43.5/37.9	51.6	70.4	76.8
Idefics2	8B	320	43.0/37.7	51.4	73.0	76.7

Vision Encoder + LLM Decoder

Qualitative results:

Prompt

Extract the name, email, current job, and education.
Output a JSON format.

YANN KARPATHY

 yann.karpathy@gmail.com  Paris, France

Accomplished machine learning researcher with over 5 years of specialized experience at Hugging Face, focusing on developing cutting-edge AI technologies. Proven track record of innovation and leadership in artificial intelligence, seeking to leverage expertise to tackle new challenges in machine learning and AI development.

EXPERIENCE

05/2018 - Current *Machine Learning Engineer, Company: Hugging Face*
Vision Language Models.

EDUCATION

2012 - 2018 **Ph.D. in Computer Science**
Stanford University
Advancements in Neural Network Architectures for Image Recognition

2010 - 2012 **Master of Science in Artificial Intelligence**
Massachusetts Institute of Technology
Machine Learning Algorithms and Applications

2006 - 2010 **Bachelor of Science in Computer Science**
University of Toronto
Data Structures and Algorithm Design

SKILLS

- Proficient in Python, PyTorch, TensorFlow, and Keras
- Extensive experience with NLP and computer vision
- Strong background in statistical data analysis and model validation
- Excellent communication and leadership skills

LANGUAGES

French, English: First Language



Idefics2 output

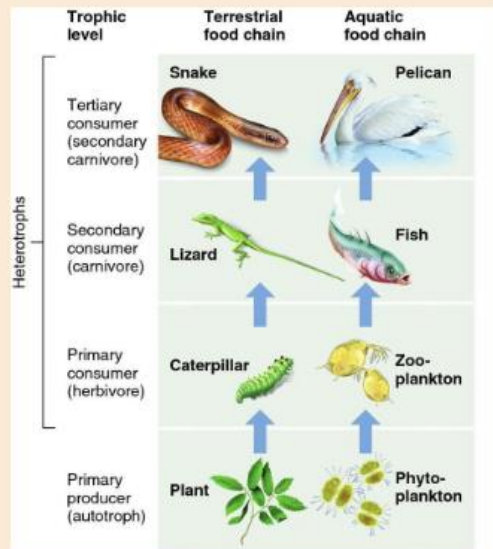
```
{
  "name": "Yann Karpathy",
  "email": "yann.karpathy@gmail.com",
  "current_job": "Machine Learning Engineer, Hugging Face",
  "education": [
    {
      "year": "2012 - 2018",
      "degree": "Ph.D. in Computer Science",
      "university": "Stanford University"
    },
    {
      "year": "2010 - 2012",
      "degree": "Master of Science in Artificial Intelligence",
      "university": "Massachusetts Institute of Technology"
    },
    {
      "year": "2006 - 2010",
      "degree": "Bachelor of Science in Computer Science",
      "university": "University of Toronto"
    }
  ]
}
```

Vision Encoder + LLM Decoder

Qualitative results:

Prompt

What happens to fish if pelicans increase?

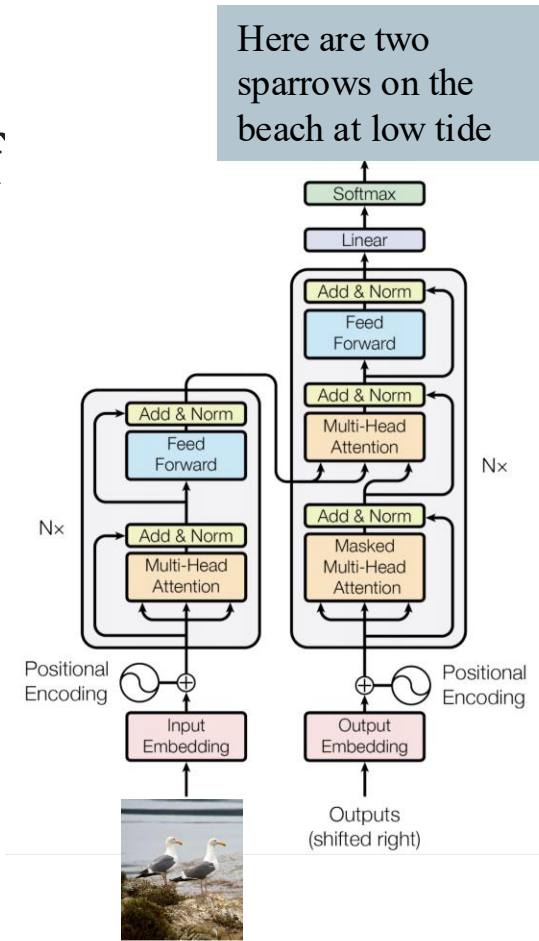


Idedics2 output

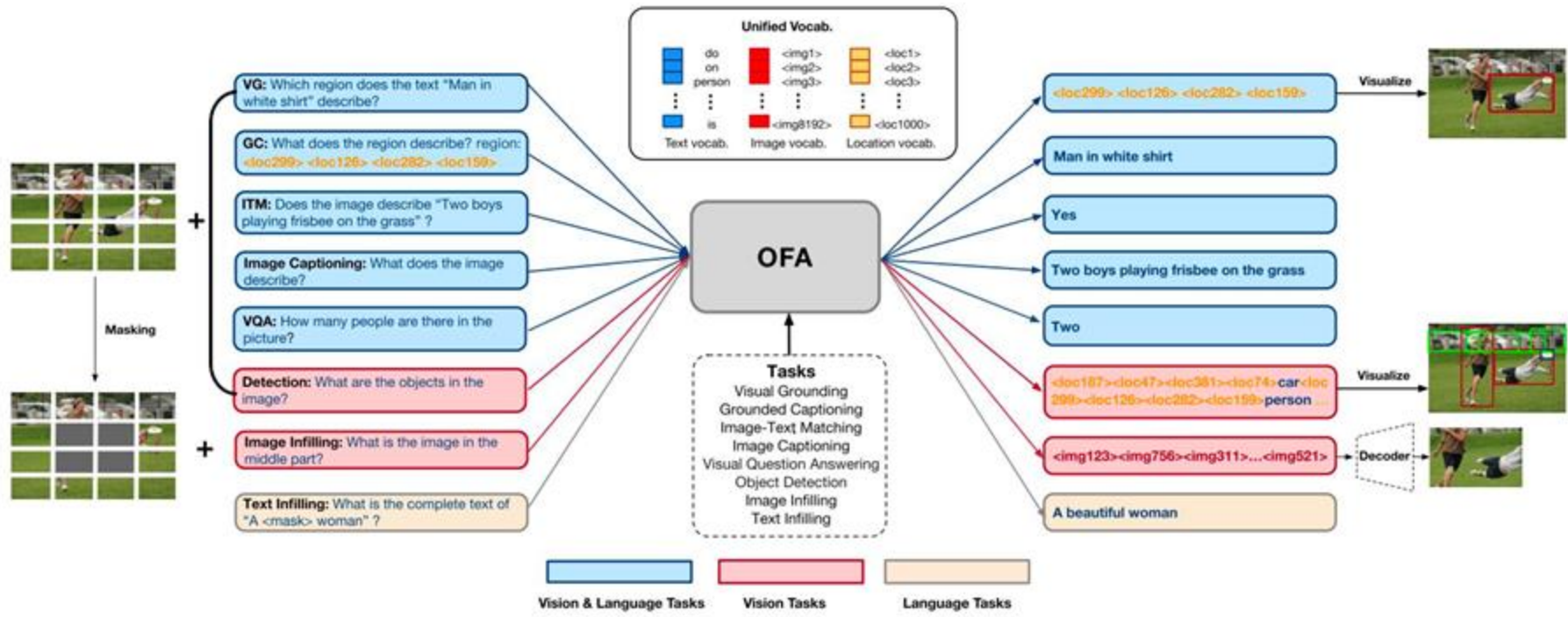
If pelicans increase, they may consume more fish, which could lead to a decrease in the fish population or an imbalance in the ecosystem. This could potentially affect other species that rely on fish for food, such as seals, dolphins, and humans who fish for consumption.

1. Vision-Language Models in the era of LLMs

- Unimodal models with connection
- **One model for all**



One model with: many inputs / many outputs / many tasks



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitrary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

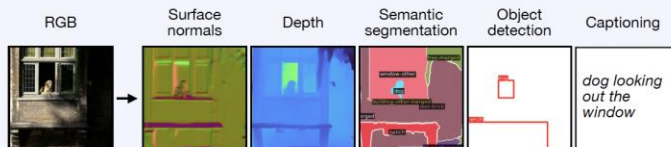
4M: Massively Multimodal Masked Modeling

NeurIPS 2023/24

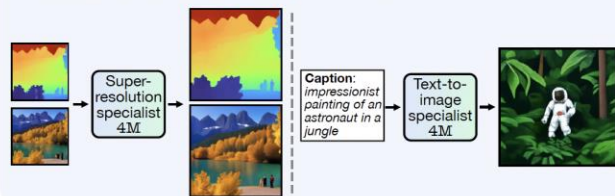
David Mizrahi^{1,2*} Roman Bachmann^{1*} Oguzhan Fatih Kar¹Teresa Yeo¹ Mingfei Gao² Afshin Dehghan² Amir Zamir¹¹Swiss Federal Institute of Technology Lausanne (EPFL) ²Apple

A generalist vision model that can...

... perform a diverse set of vision tasks out of the box



... be easily fine-tuned into specialist variants

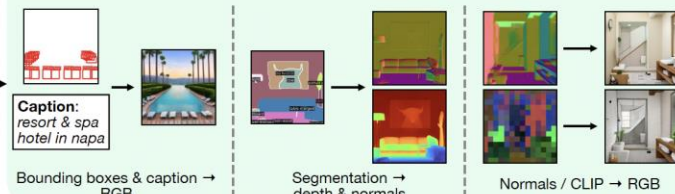


... transfer well to unseen tasks and modalities

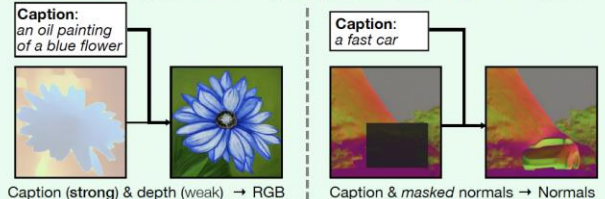


A multimodal generative model that can...

... generate any modalities conditioned on any other(s) ...



... with varying conditioning weights and from partial inputs ...

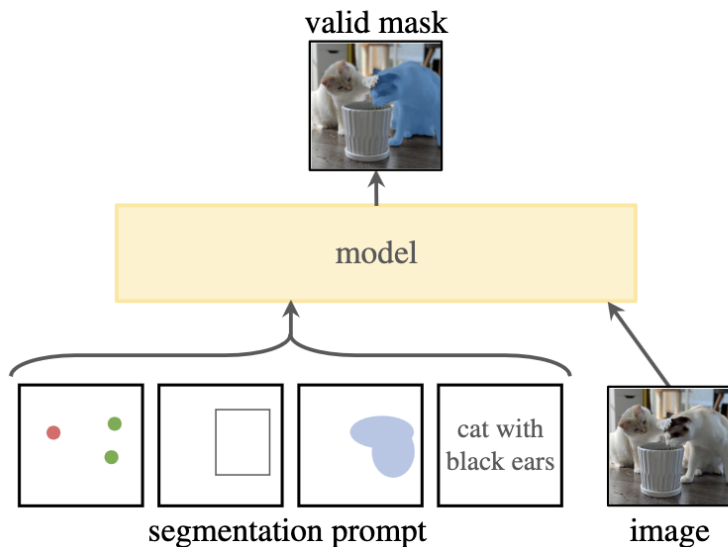


... enabling precise user control through multimodal editing chains

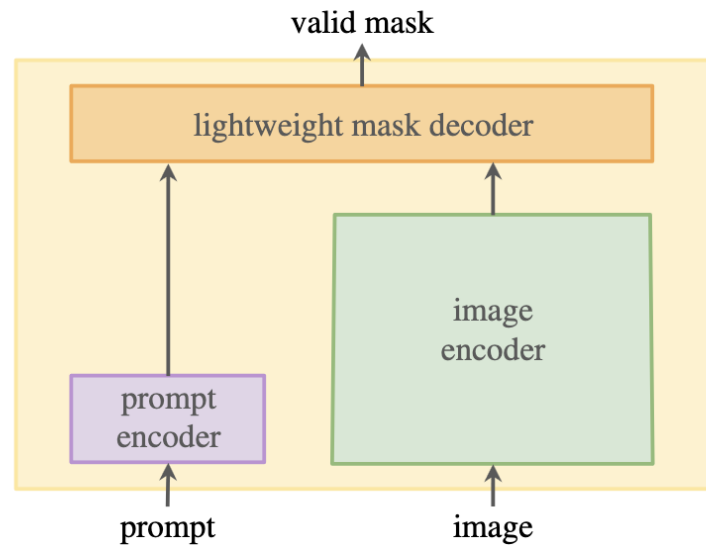


Segment Anything

Alexander Kirillov^{1,2,4} Eric Mintun² Nikhila Ravi^{1,2} Hanzi Mao² Chloe Rolland³ Laura Gustafson³
Tete Xiao³ Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár⁴ Ross Girshick⁴



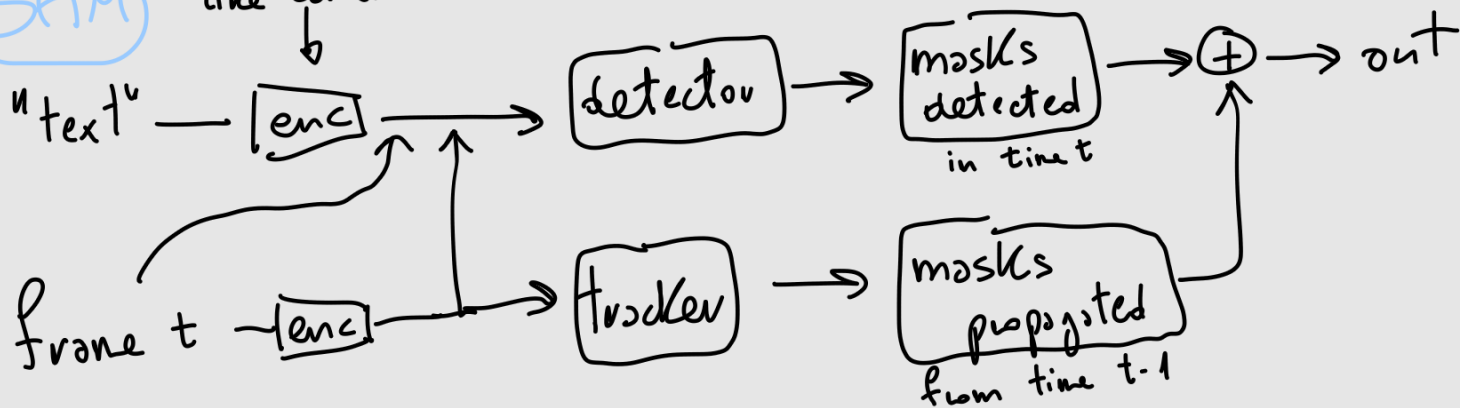
(a) **Task:** promptable segmentation



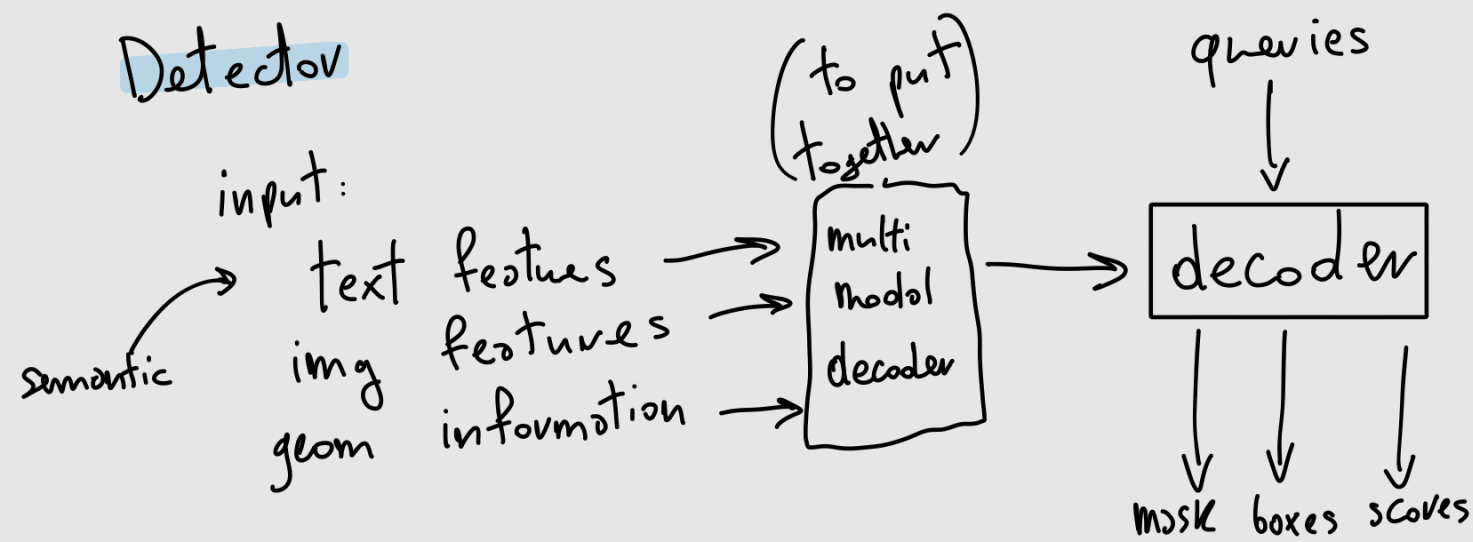
(b) **Model:** Segment Anything Model (SAM)

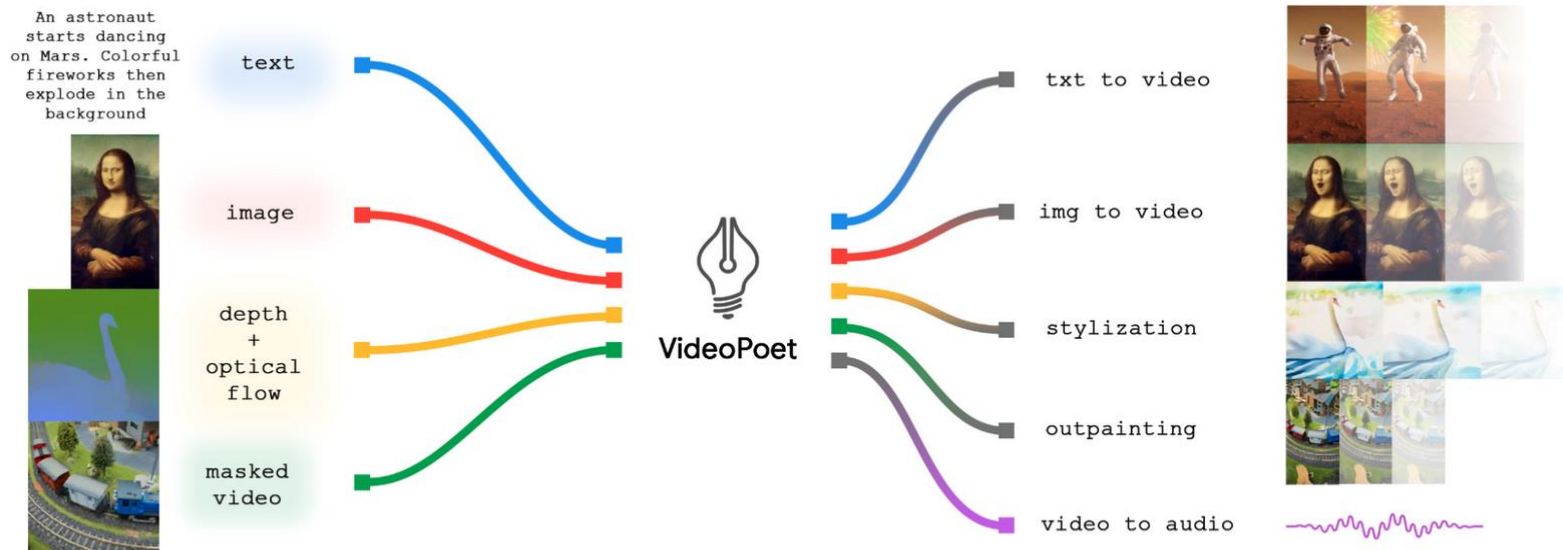
SAM

like LBP encoder

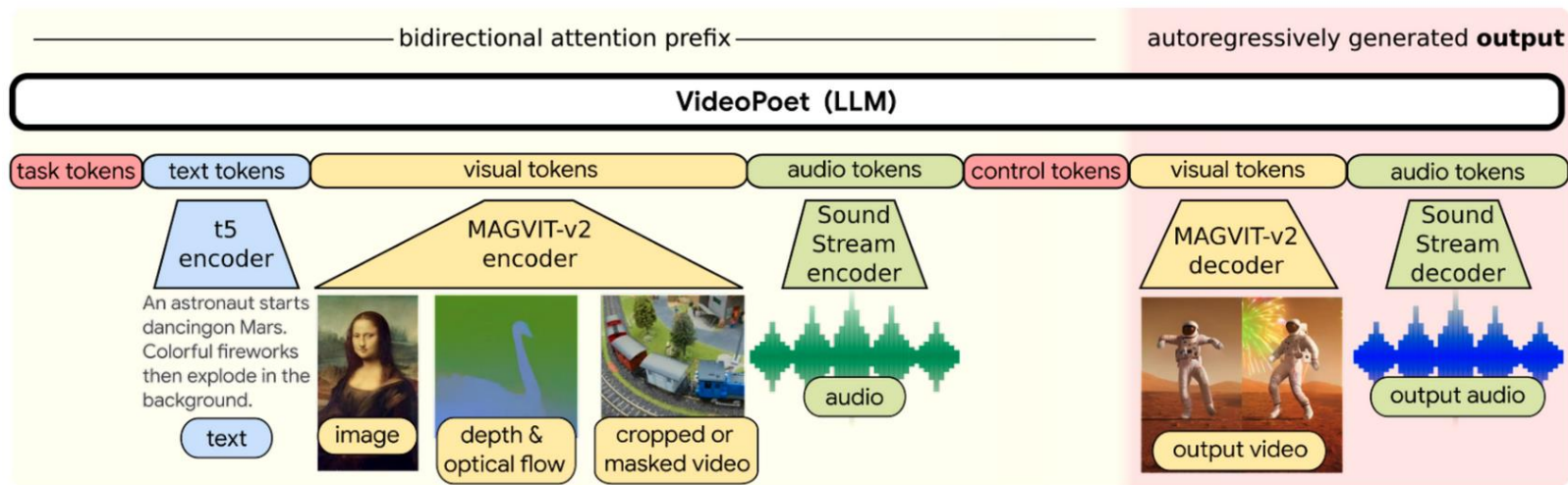


Detector





An overview of VideoPoet, capable of multitasking on a variety of video-centric inputs and outputs. The LLM can optionally take text as input to guide generation for text-to-video, image-to-video, video-to-audio, stylization, and outpainting tasks. Resources used: [Wikimedia Commons](#) and [DAVIS](#).



A detailed look at the VideoPoet task design, showing the training and inference inputs and outputs of various tasks. Modalities are converted to and from tokens using tokenizer encoder and decoders. Each modality is surrounded by boundary tokens, and a task token indicates the type of task to perform.