Multimodal NLP

Author: Matthieu Futeral

- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - Text-to-Image diffusion models

- Visual language models
  - **Tasks that require processing images (or videos) & text**
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - Text-to-Image diffusion models

Multimodal tasks - Examples

A realistic painting of students in a classroom listening to their professor.

Multimodal tasks - Examples

Image Captioning

"man in black shirt is playing guitar."

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

Visual question Answering

What is the mustache made of?

AI System

bananas

Vision & Language Navigation

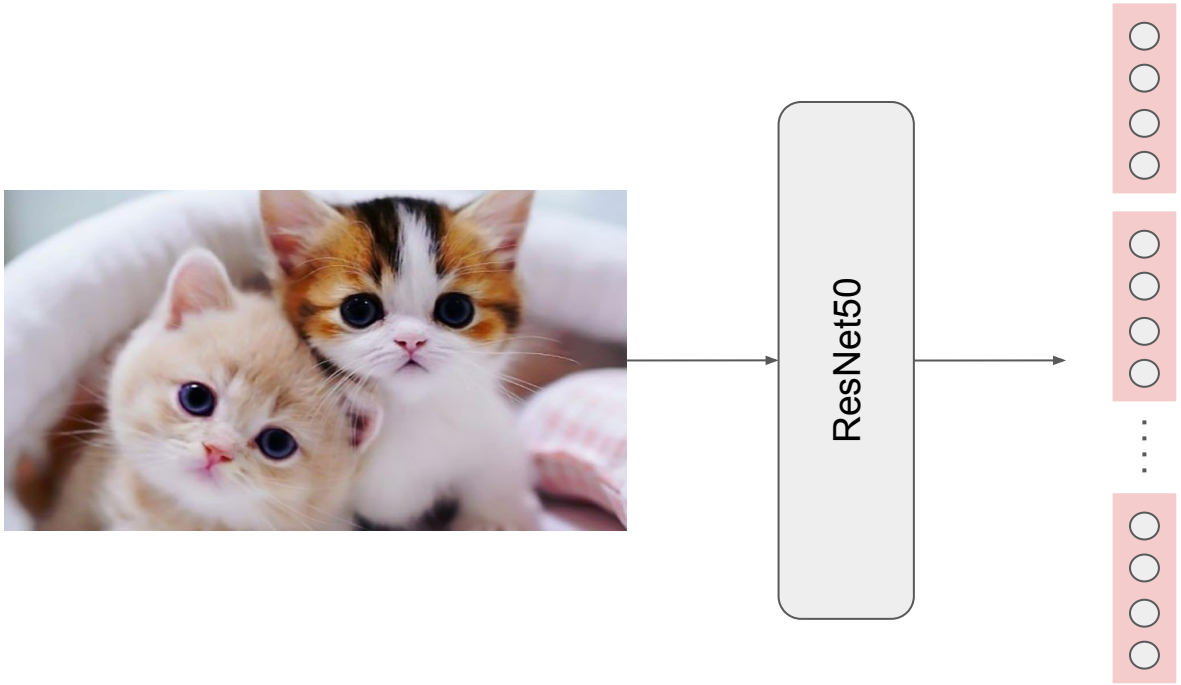
Text-to-Image Generation

And others…: Image-text retrieval, Visual commonsense reasoning, Grounding Referring Expressions *etc…*

- Visual language models
  - Tasks that require processing images (or videos) & text
  - **Visual input representations & training objectives**
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - Text-to-Image diffusion models

Visual input representations

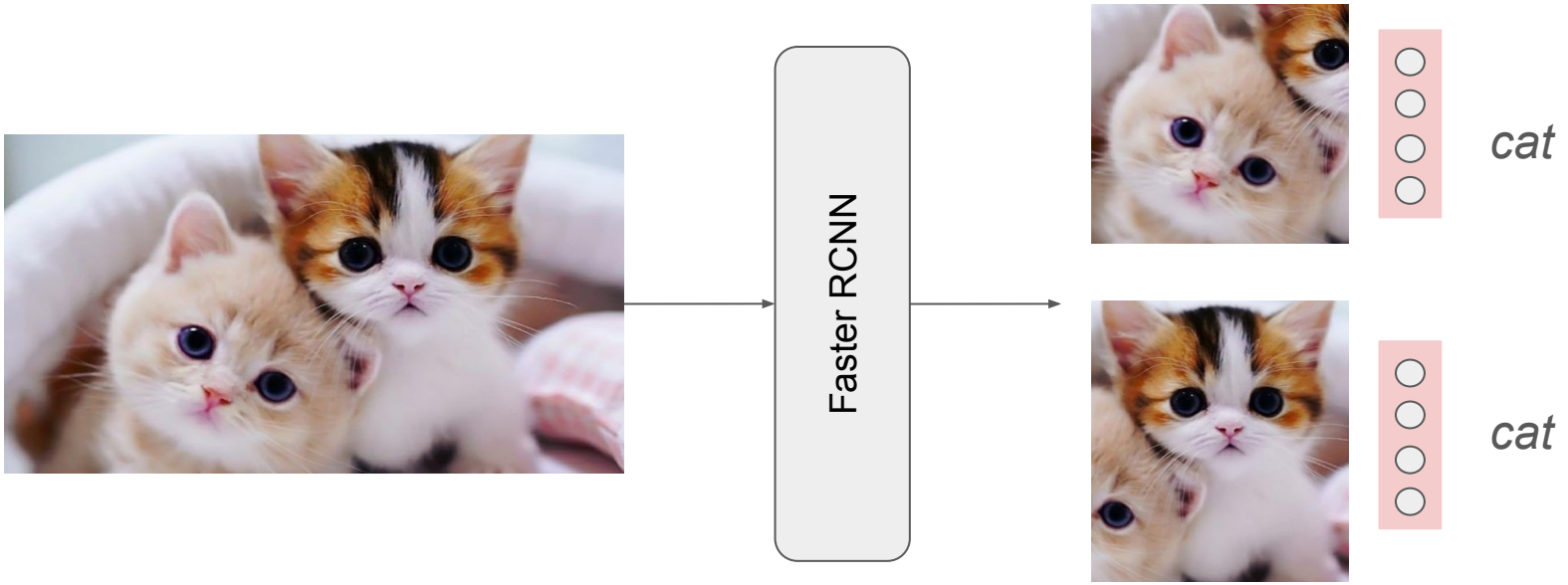
Continuous embeddings from the whole image (Grid)



Pixel-BERT (Huang et al. 2020), SOHO (Huang et al. 2021), SimVLM (Wang et al. 2022) …

Visual input representations

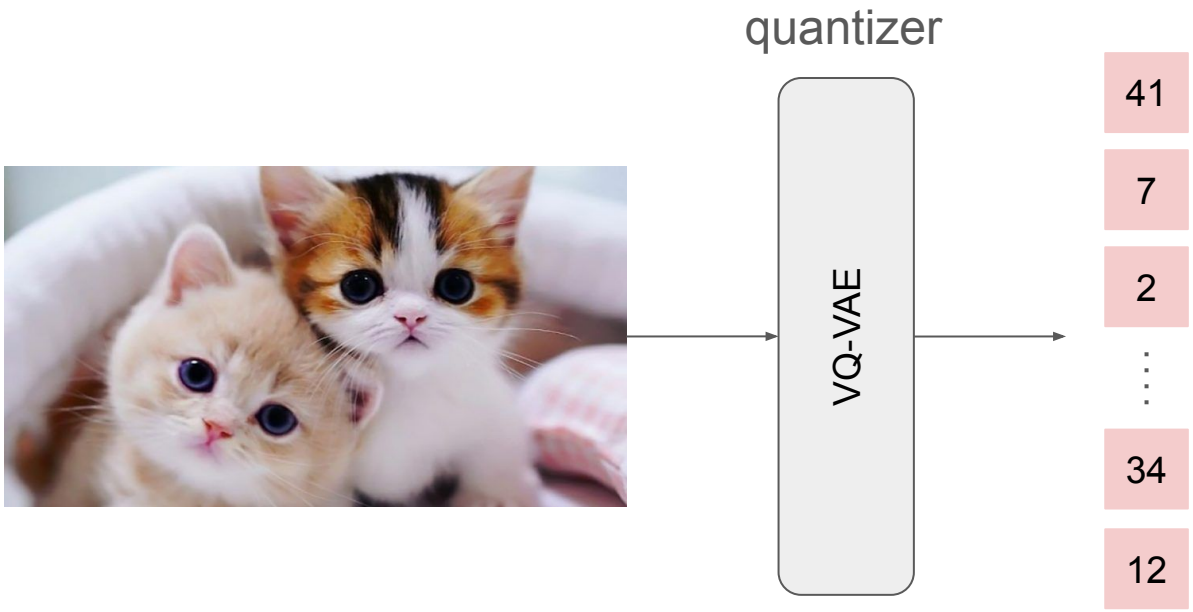
Bounding boxes



VisualBERT (Li et al. 2019), LXMERT (Tan et al. 2019), VL-BERT (Su et al. 2019) …

Visual input representations

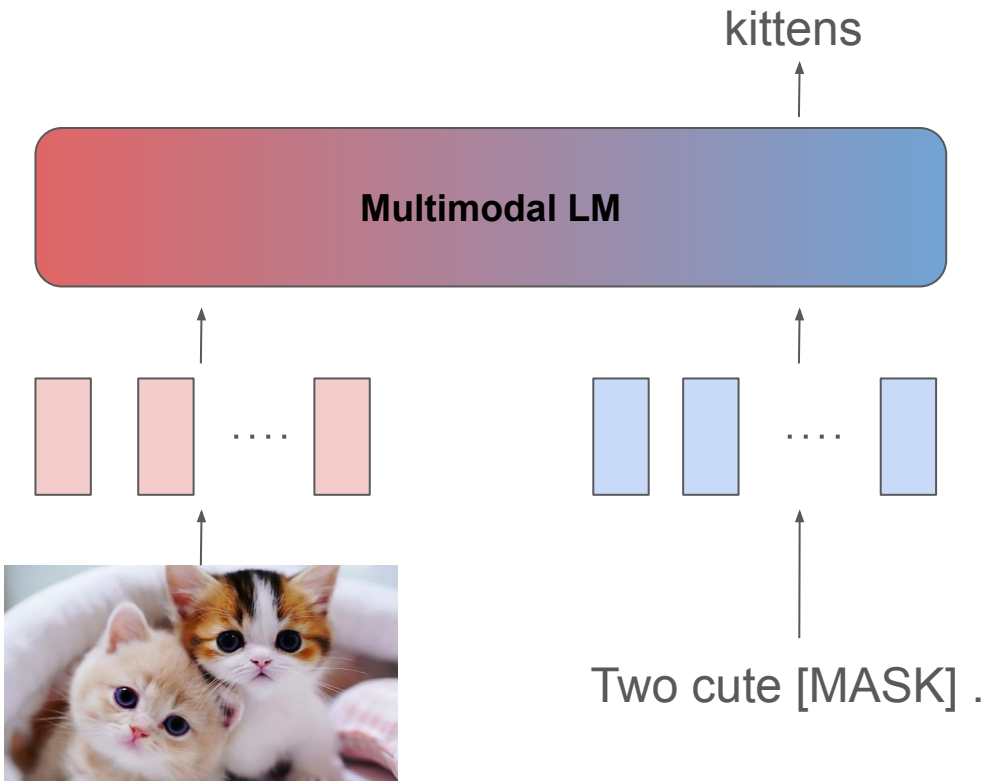
Discrete tokens



DALL-E (Ramesh et al. 2021), Parti (Yu et al. 2022)

Training objectives

Masked language modelling

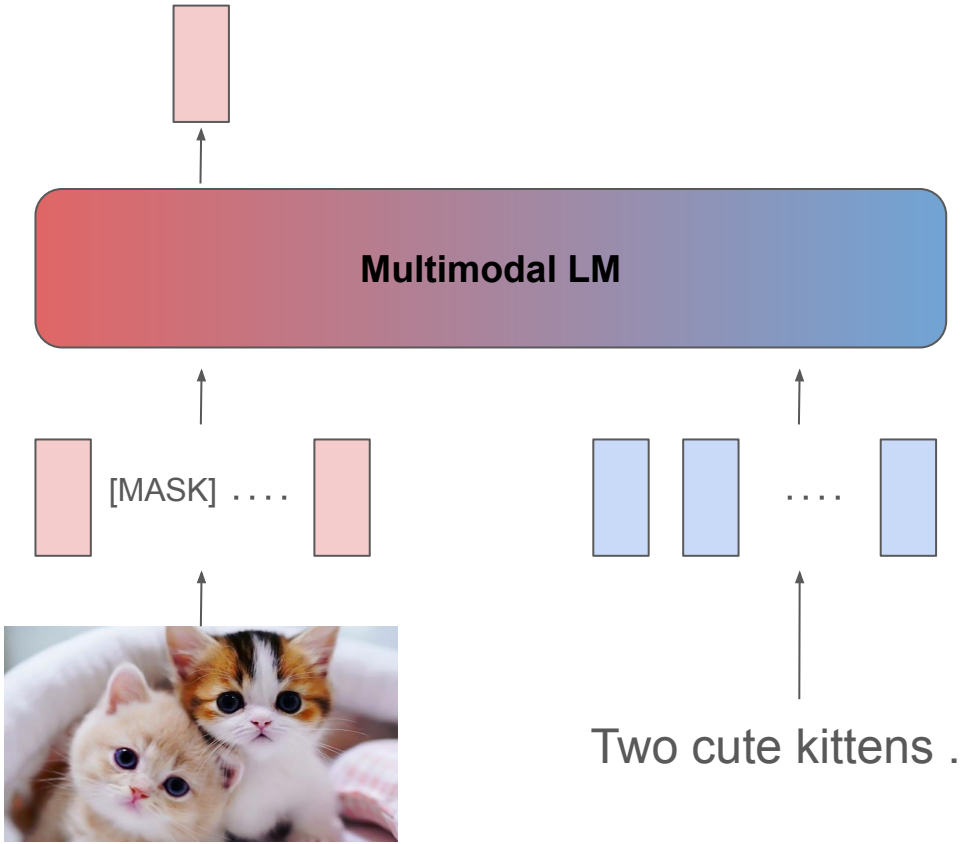


Generative or multiple choice tasks:

Image captioning, Visual question answering etc.

Training objectives

Masked Image modelling

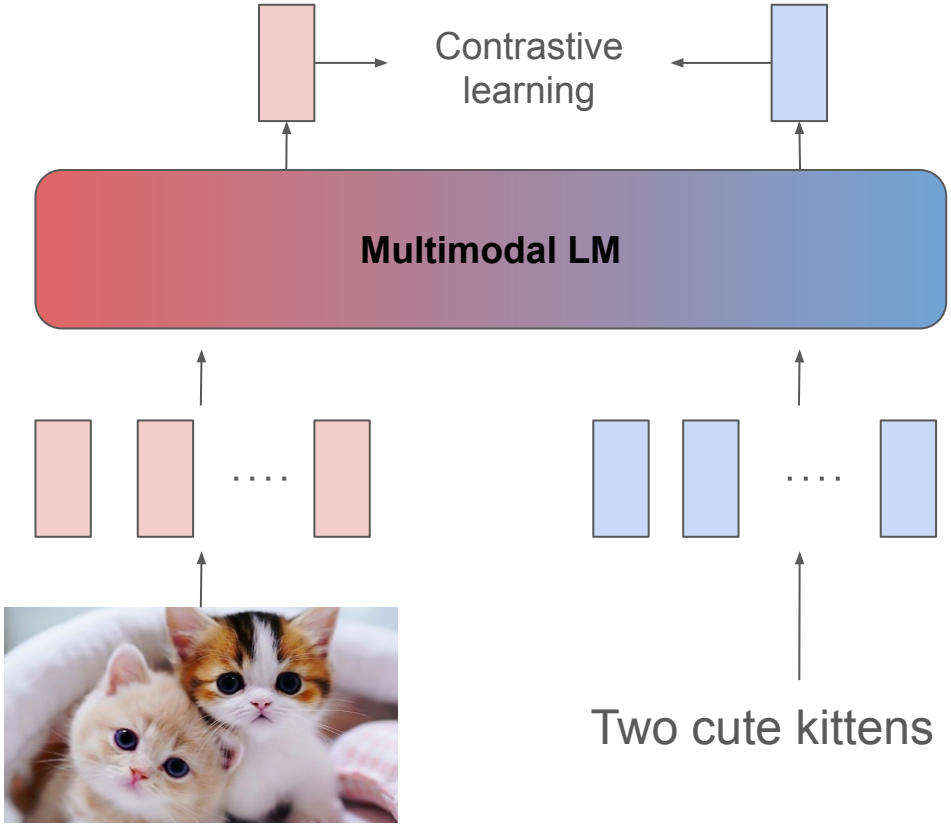


Generative or multiple choice tasks:

Image captioning, Visual question answering etc.

Training objectives

Image-Text matching



Retrieval tasks:

Image-text retrieval, Zero-shot image classification etc.

Training objectives - Contrastive learning

i_1  Two cute kittens. t_1

i_2  Two people playing video games. t_2

i_3  A photo of the Eiffel tower. t_3

Training objectives - Contrastive learning

i_1

i_2

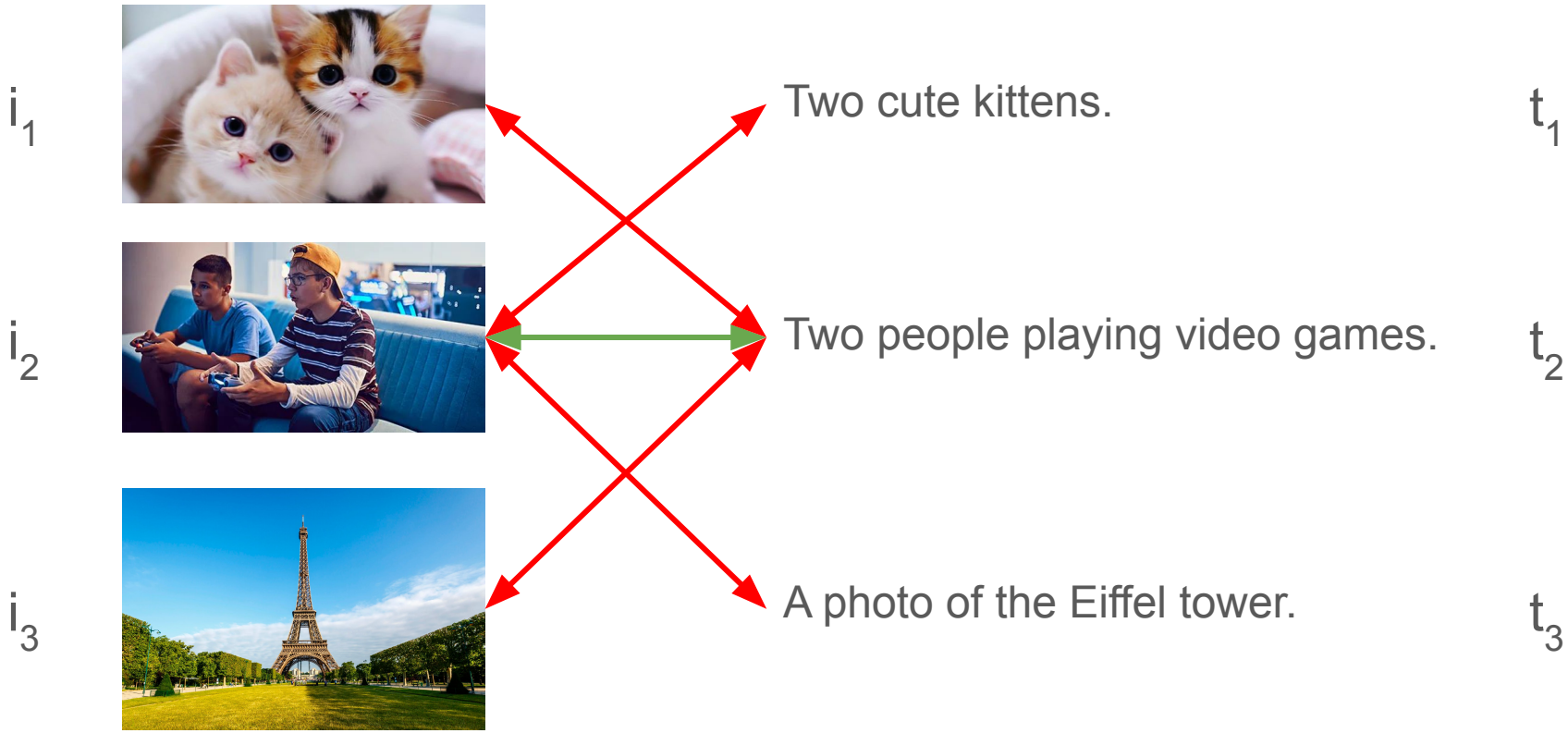
i_3

Two cute kittens. t_1

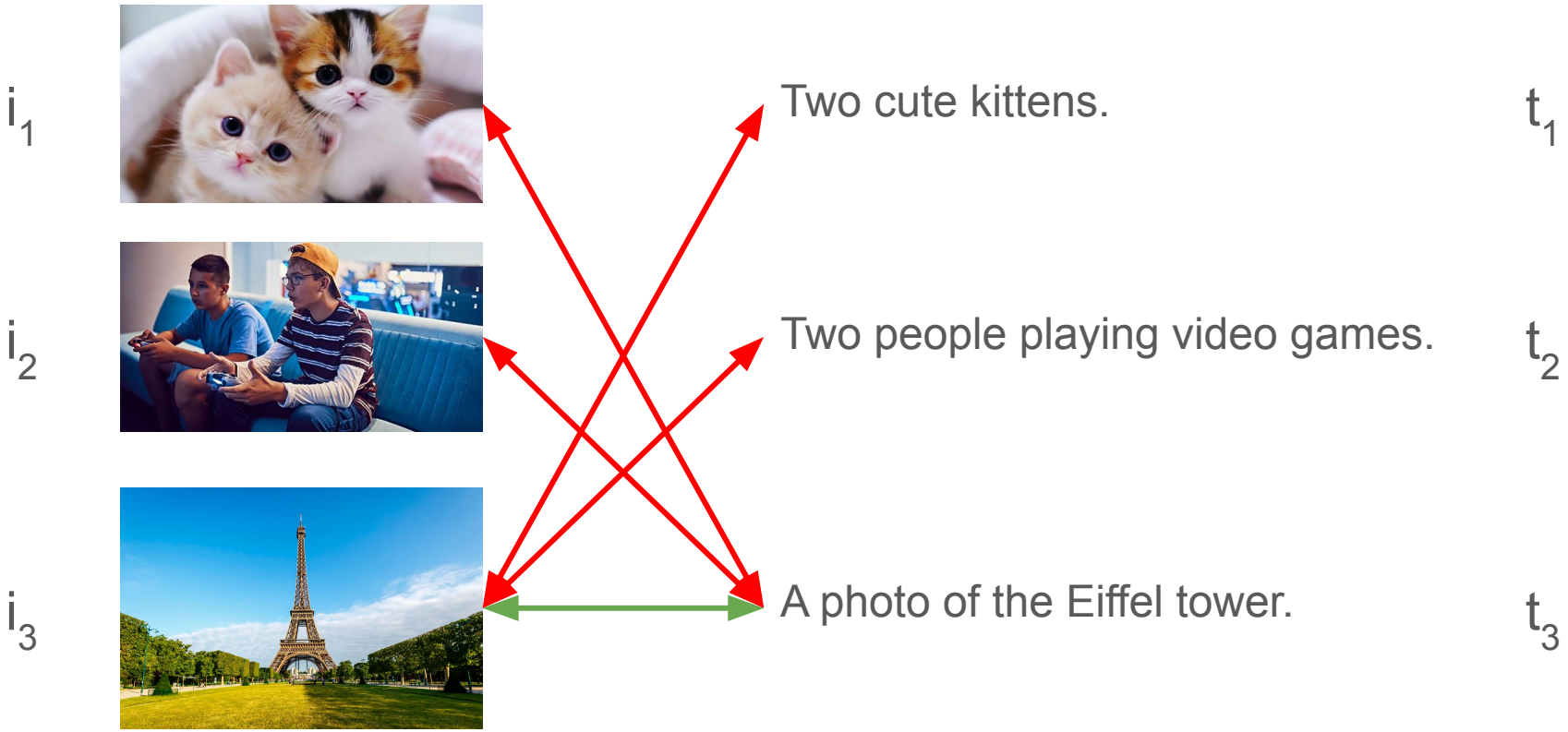
Two people playing video games. t_2

A photo of the Eiffel tower. t_3

Training objectives - Contrastive learning



Training objectives - Contrastive learning



Training objectives - Contrastive learning

image-to-text

$$\frac{\exp(i_1 . t_1)}{\sum_{k=1,2,3} \exp(i_1 . t_k)}$$

text-to-image

$$\frac{\exp(i_1 . t_1)}{\sum_{k=1,2,3} \exp(i_k . t_1)}$$

Training objectives - Contrastive learning, general case

image-to-text

$$\frac{\exp(i_n . t_n)}{\sum_{k=1...N} \exp(i_n . t_k)}$$

text-to-image

$$\frac{\exp(i_n . t_n)}{\sum_{k=1...N} \exp(i_k . t_n)}$$

Negative samples are usually drawn from the same batch.

- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - **Overview of VLM architectures and fusion method**
  - Leveraging pretrained language models
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - Text-to-Image diffusion models

Overview of VLM architectures and fusion methods

Previous section: how to represent images (or videos) into a visual language model (bounding boxes, grid features, vision tokens…)

Now, how to model vision and language information jointly?

Overview of VLM architectures and fusion methods

Dual Encoder

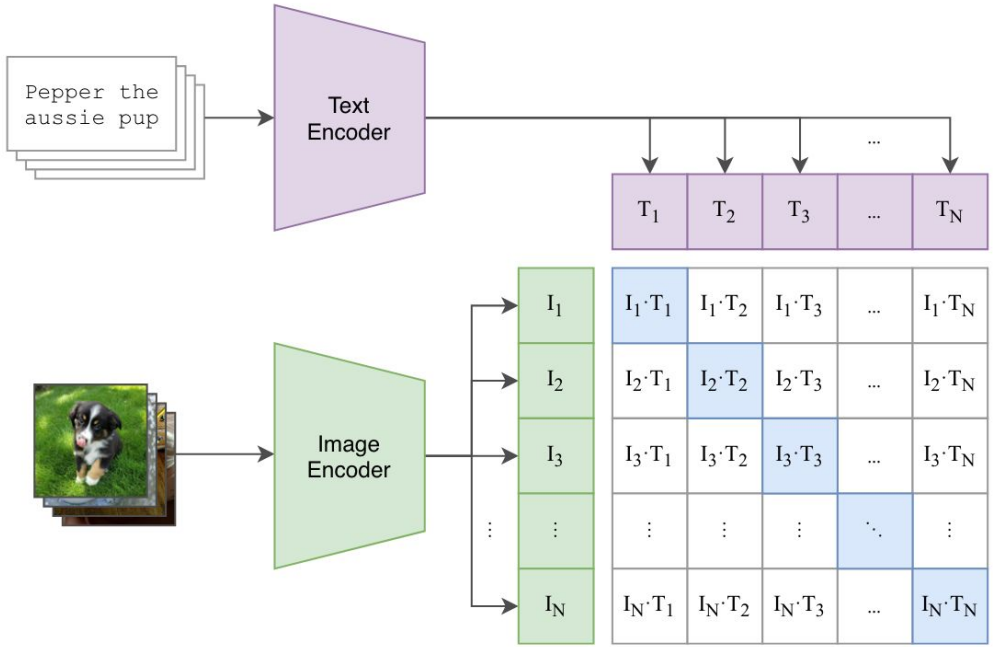


Image from https://openai.com/research/clip

Overview of VLM architectures and fusion methods

Dual Encoder with cross-attention

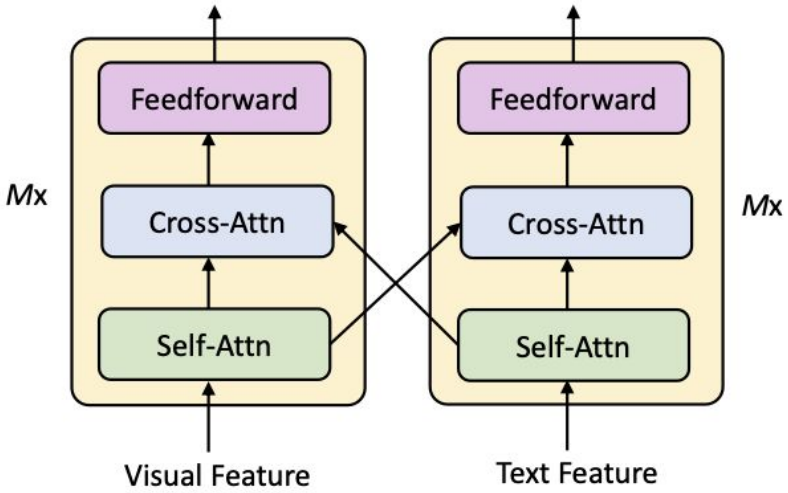


Image from Gan et al. (2022)

Overview of VLM architectures and fusion methods

Single Encoder with joint self-attention

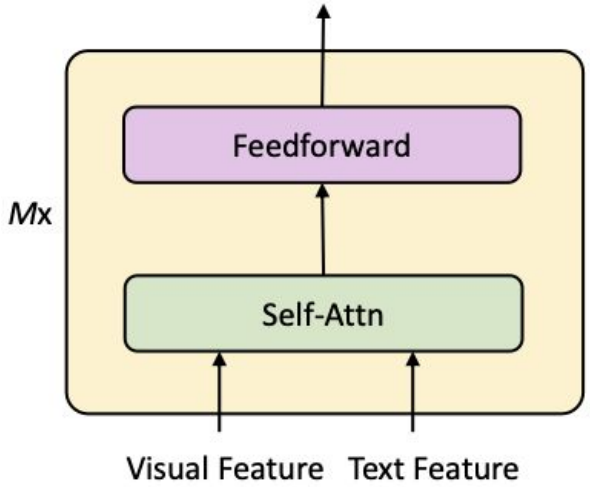


Image from Gan et al. (2022)

Overview of VLM architectures and fusion methods

Encoder vs Encoder-Decoder

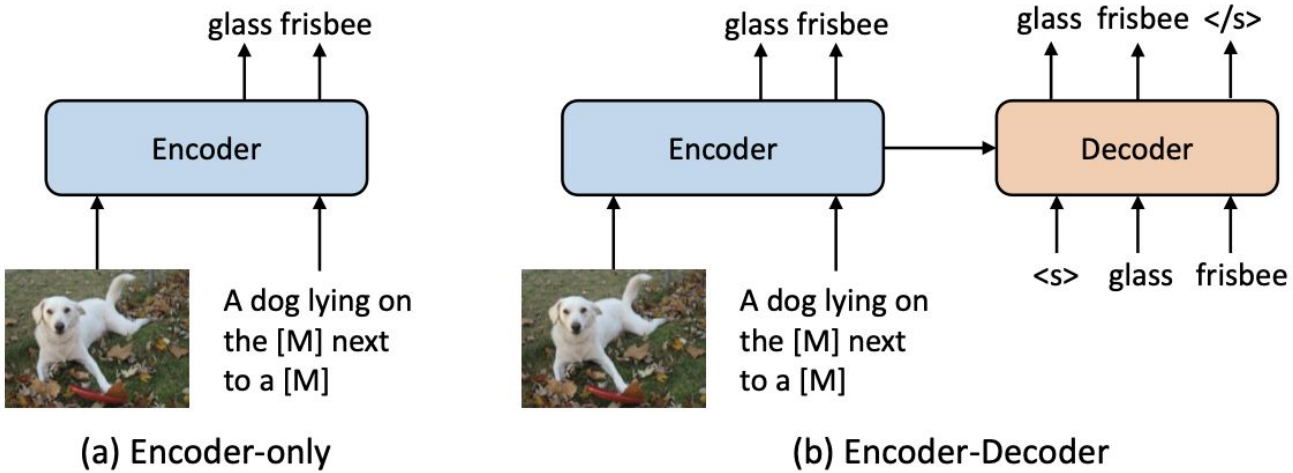


Image from Gan et al. (2022)

Overview of VLM architectures and fusion methods

Global overview of current VLM

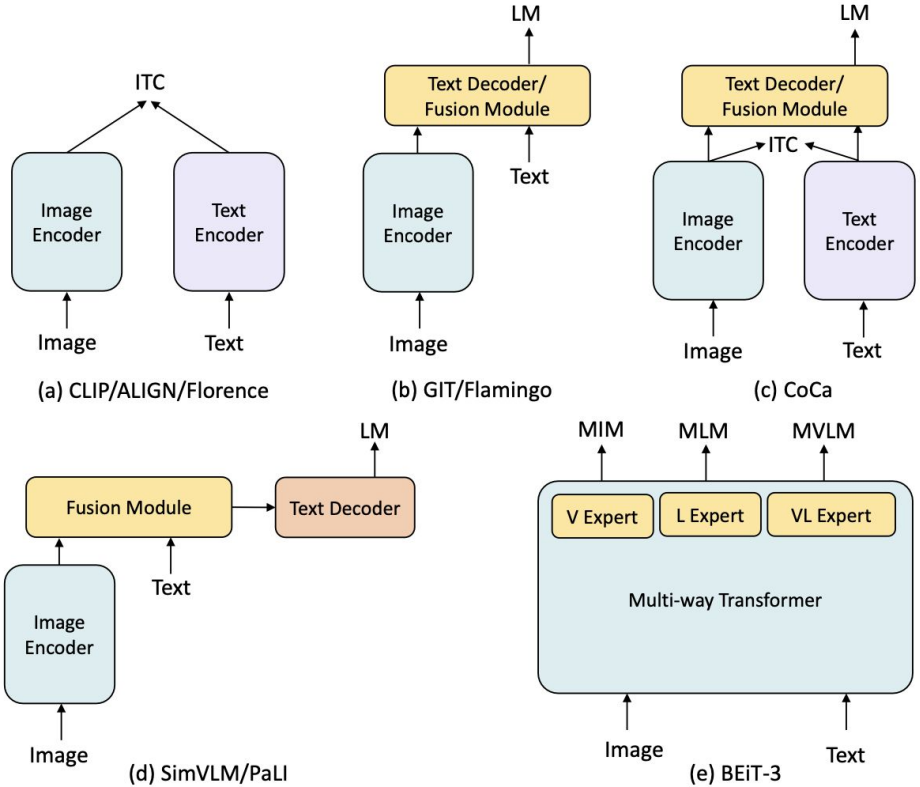


Image from Gan et al. (2022)

- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - **Leveraging pretrained language models**
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - Text-to-Image diffusion models

Leveraging pretrained language models

All the previous models are trained from images and their captions. It has some limitations:

1) Models are only trained on captiony text data

2) Far more text only available than image + captions data

3) Limit prompting abilities as models never seen raw plain text

Leveraging pretrained language models - Flamingo

Flamingo model ->

1) Start from the weights of a language model frozen
2) Train on whole document data instead of just image+caption

Multimodal document 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. …

Image from Laurençon et al. (2023)

Leveraging pretrained language models - Flamingo

Flamingo model ->

1) Start from the weights of a language model frozen
2) Train on whole document data instead of just image+caption

Start from a strong language model

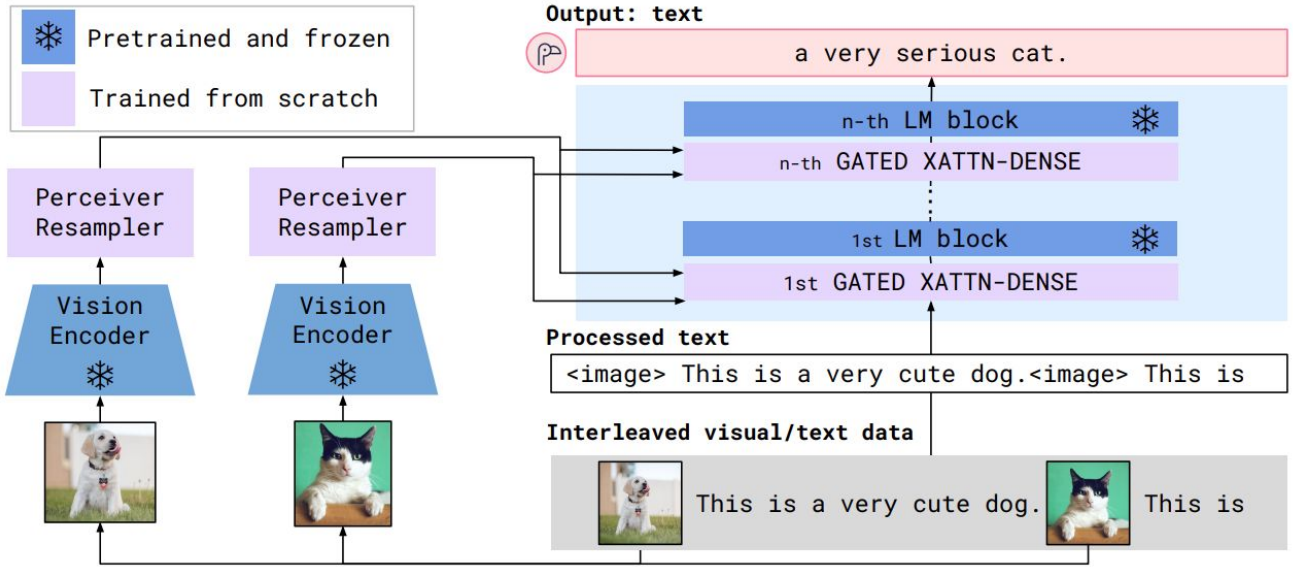


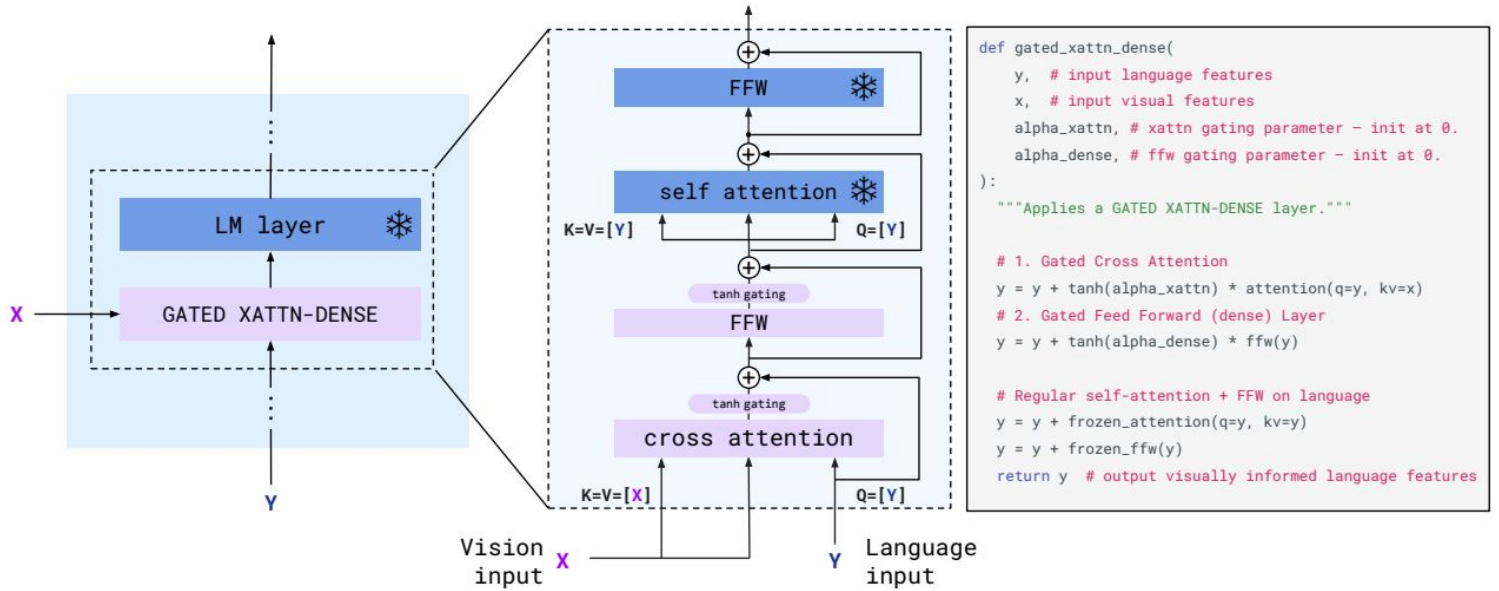
Image from Alayrac et al. (2023)

Leveraging pretrained language models - Flamingo

Flamingo model ->

1) Start from the weights of a language model frozen
2) Train on whole document data instead of just image+caption

Gated XAttn layer

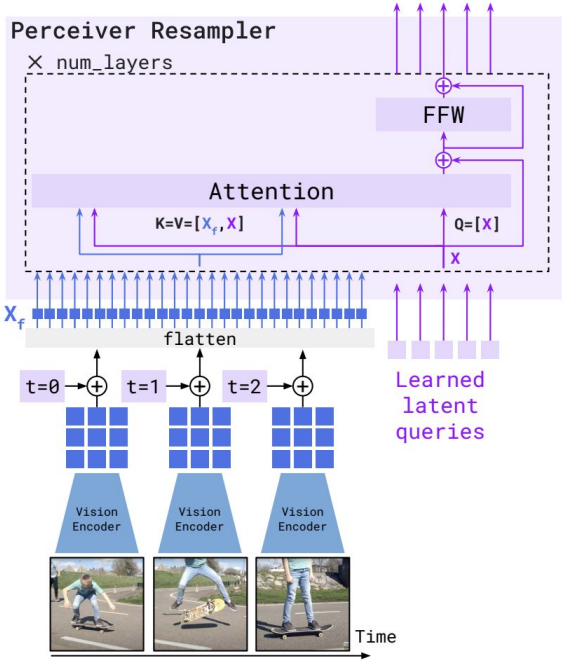



```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Image from Alayrac et al. (2023)

Leveraging pretrained language models - Flamingo

Visual connector => Perceiver Resampler




```python
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
```

Figure 5: **The Perceiver Resampler** module maps a *variable* size grid of spatio-temporal visual features output by the Vision Encoder to a *fixed* number of output tokens (five in the figure), independently from the input image resolution or the number of input video frames. This transformer has a set of learned latent vectors as queries, and the keys and values are a concatenation of the spatio-temporal visual features with the learned latent vectors.

Image from Alayrac et al. (2023)

Leveraging pretrained language models - Flamingo

Zero-shot results - Emergence of In context learning capabilities

Method	FT	Shot	OKVQA (I)	VQAv2 (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	✗	(X)	[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	**42.8**	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	**60.8**
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	**55.6**	36.5	30.8	68.6	-
	✗	32	**57.8**	**67.6**	**113.8**	**52.3**	**65.1**	**49.8**	**75.4**	**31.0**	**45.3**	**86.8**	42.2	**55.6**	**37.9**	**33.5**	**70.0**	-
Pretrained FT SOTA	✔	(X)	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.

Table from Alayrac et al. (2023)

- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - **Visual programming**
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - Text-to-Image diffusion models

Visual programming - ViperGPT & Visual programming

Hard for end-to-end model to answer all types of visual queries because requires visual processing and reasoning.

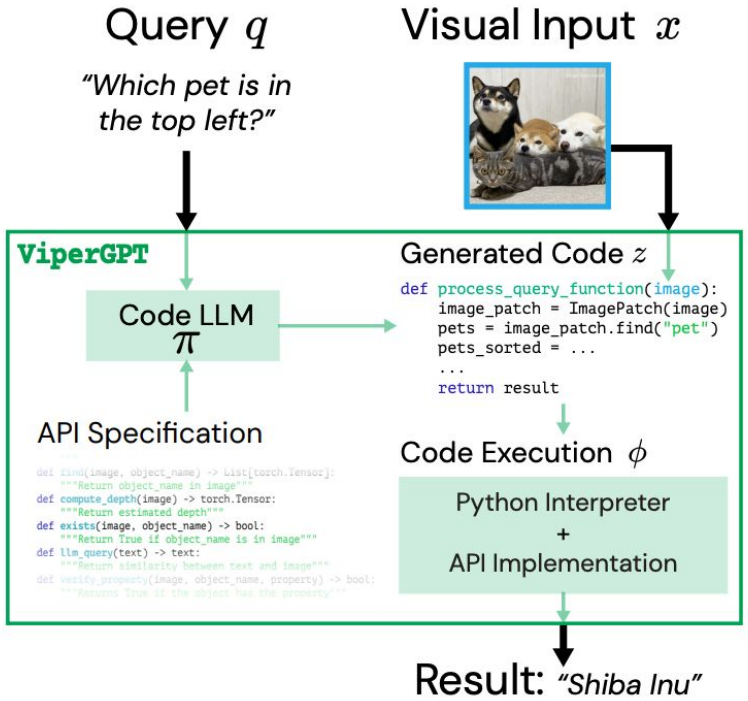


Image from Suris et al. (2023)

Visual programming - ViperGPT & Visual programming

Hard for end-to-end model to answer all types of visual queries because requires visual processing and reasoning.

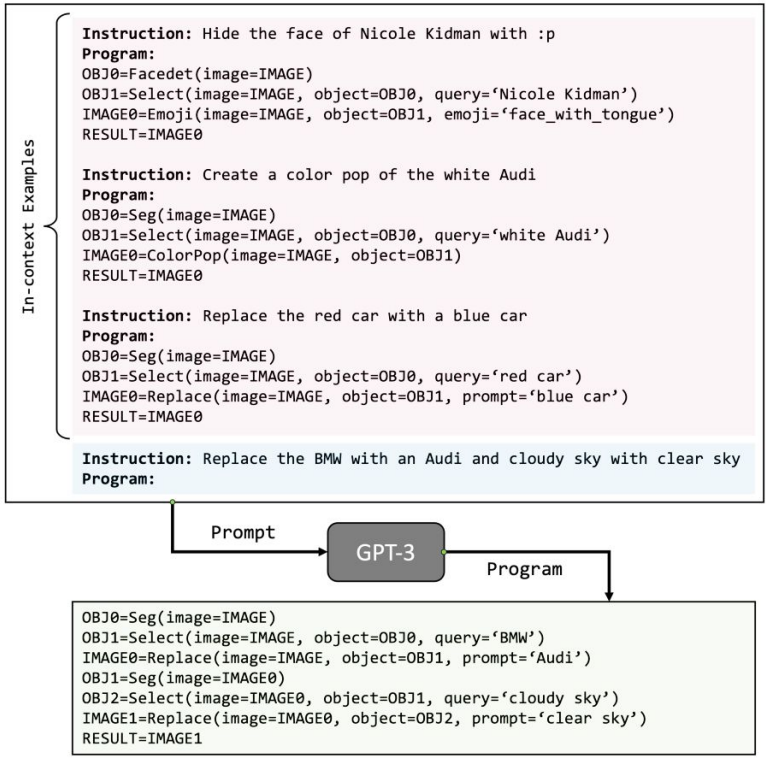


Image from Gupta et al. (2023)

Visual programming - ViperGPT & Visual programming

Some examples from VisProg and ViperGPT

Natural Language Image Editing

IMAGE:

Prediction: IMAGE1

Instruction: Hide Daniel Craig with 8) and Sean Connery with ;)

Program:

```
OBJ0=FaceDet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Daniel Craig', category=None)
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='smiling_face_with_sunglasses')
OBJ2=Select(image=IMAGE, object=OBJ0, query='Sean Connery', category: None)
IMAGE1=Emoji(image=IMAGE0, object=OBJ2, emoji='winking_face')
RESULT=IMAGE1
```

VisProg

Query: How many muffins can each kid have for it to be fair?

Generated Code

```
def execute_command(image):
    image_patch = ImagePatch(image)
    muffin_patches = image_patch.find("muffin")
    kid_patches = image_patch.find("kid")
    return str(len(muffin_patches) // len(kid_patches))
```

ViperGPT

Image from Gupta et al. (2023) & Suris et al. (2023)

Visual programming - ViperGPT & Visual programming

How does it work under the hood? **No training!** Only in-context learning and predefined programs

- VisProg -> GPT3 -> Pure handcrafted in context examples
- ViperGPT -> Codex -> API specification as input (function name and description)

Visual programming - ViperGPT & Visual programming

Table 2. **GQA Results**. We report accuracy on the test-dev set.

		Accuracy (%) ↑
Sup.	LGCN [20]	55.8
	LXMERT [51]	60.0
	NSM [24]	63.0
	CRF [39]	72.1
ZS	BLIP-2 [30]	44.7
	ViperGPT (ours)	**48.1**

Table 3. **OK-VQA Results.**

		Accuracy (%) ↑
Sup.	TRiG [13]	50.5
	KAT [16]	54.4
	RA-VQA [32]	54.5
	REVIVE [33]	58.0
	PromptCap [21]	58.8
ZS	PNP-VQA [52]	35.9
	PICa [60]	43.3
	BLIP-2 [30]	45.9
	Flamingo [1]	50.6
	ViperGPT (ours)	**51.9**

Tables from Suris et al. (2023)

- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - Visual programming
  - **State-of-the-art Vision language models (VLMs)**
  - From images to videos
  - Text-to-Image diffusion models

State-of-the-art Vision Language models (VLMs)

Some ablations studies to find the best ways to scale VLMs (and obtains the best performances)

- What matters when building vision-language models? Laurençon et al., 2024
- MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training, McKinzie et al., 2024

Several questions these papers tried to answers:

1) What types of data (and which proportion) to train VLMs?
2) What kind of neural architectures?
3) What kind of visual connectors?
4) How many image tokens to introduce into the VLM?

State-of-the-art Vision Language models (VLMs)

What type of data?

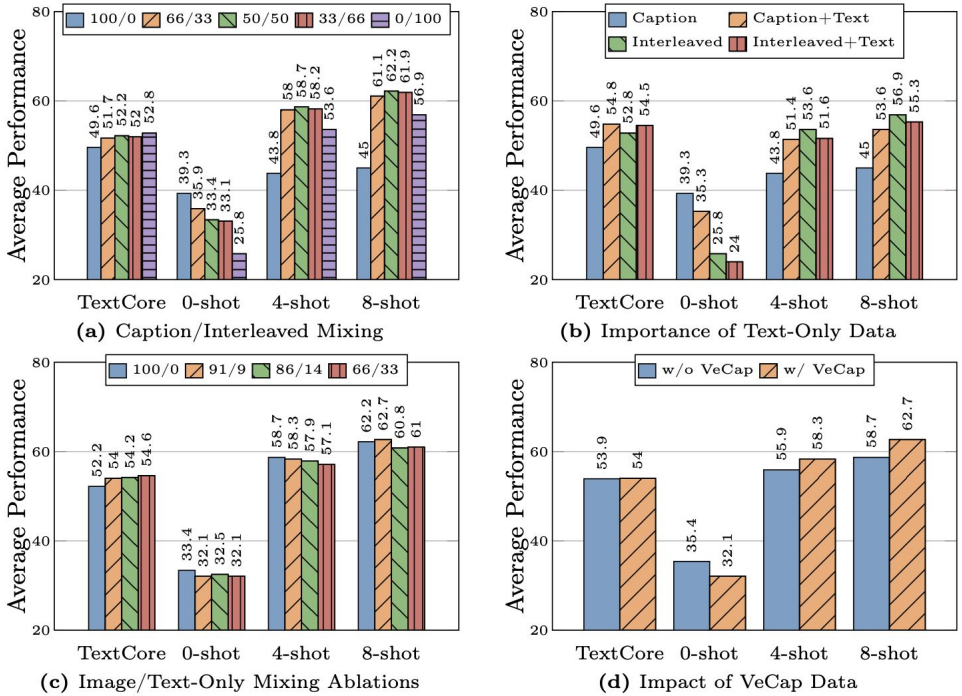


Fig. 5: Data Ablations. For each ablation, we present four different metrics: TextCore, 0-shot, 4-shot, and 8-shot. **(a)** Results with image data where we present five different mixing ratios between interleaved and captioned data. **(b)** Results with and without text-only data. We mix the text-only data separately with captioned and interleaved data. **(c)** Results with different mixing ratios between image data (caption and interleaved) and text-only data. **(d)** Results with and without including VeCap as part of caption data.

State-of-the-art Vision Language models (VLMs)

What kind of neural architecture?

Architecture	Backbones training	Avg. score
Fully autoreg. no Perceiver	Frozen	51.8
Fully autoreg.	Frozen	60.3
Cross-attention	Frozen	66.7
Cross-attention	LoRA	67.3
Fully autoreg.	LoRA	69.5

Table 3: Ablation for the architecture and method of training.

State-of-the-art Vision Language models (VLMs)

What kind of visual connectors?

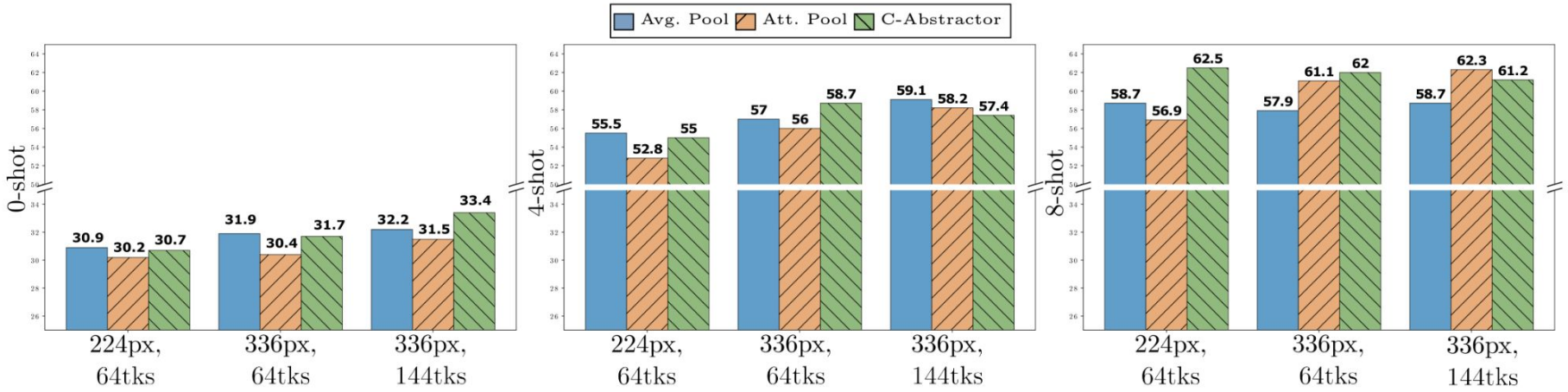
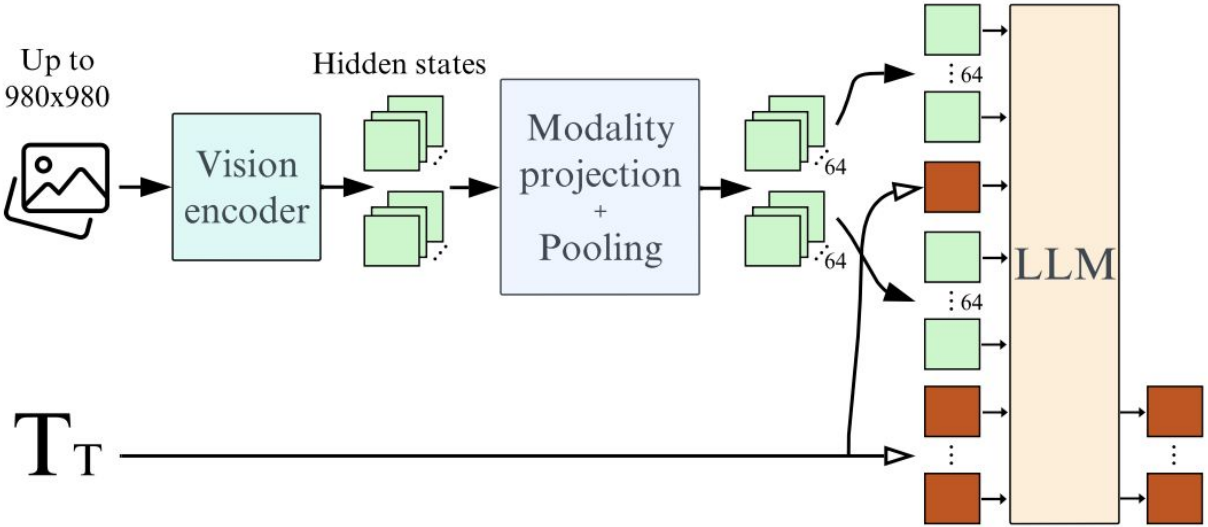


Fig. 4: 0-shot, 4-shot, and 8-shot ablations across different visual-language connectors for two image resolutions, and two image token sizes.

State-of-the-art Vision Language models (VLMs)

How many image tokens?



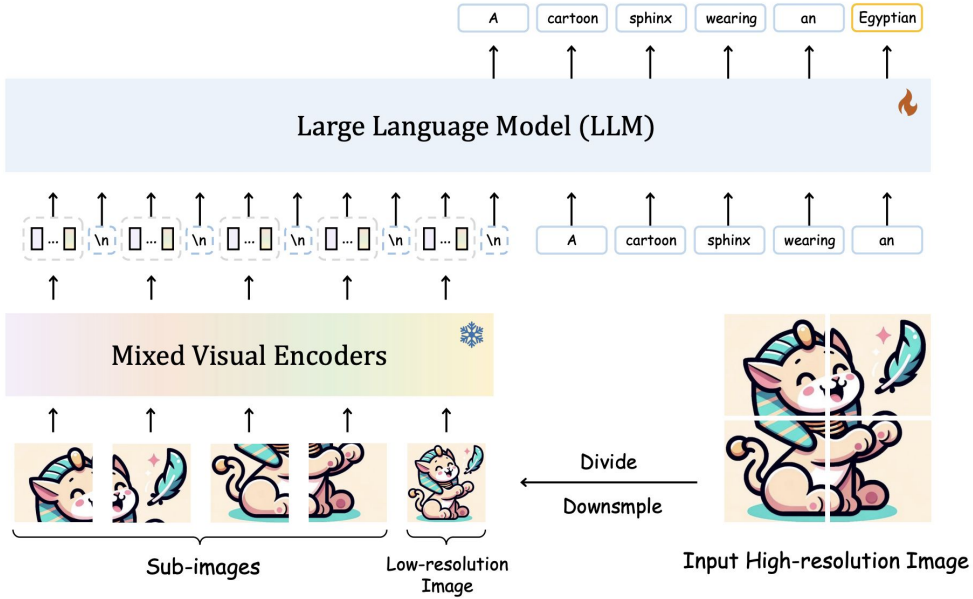
Pooling	# vis. tok.	Avg. score
Perceiver	128	71.2
Perceiver	64	71.7

Table 4: Ablation on the pooling strategy.

Figure 2: Idefics2 fully-autoregressive architecture: Input images are processed by the Vision encoder. The resulting visual features are mapped (and optionally pooled) to the LLM input space to get the visual tokens (64 in our standard configuration). They are concatenated (and potentially interleaved) with the input sequence of text embeddings (green and red column). The concatenated sequence is fed to the language model (LLM), which predicts the text tokens output.

State-of-the-art Vision Language models (VLMs)

How many image tokens? The special case of Document understanding. It requires high resolution images!



OCR data	Res.	DocVQA
W/o	384	22.6
W/o	768	42.9
W/	768	49.9

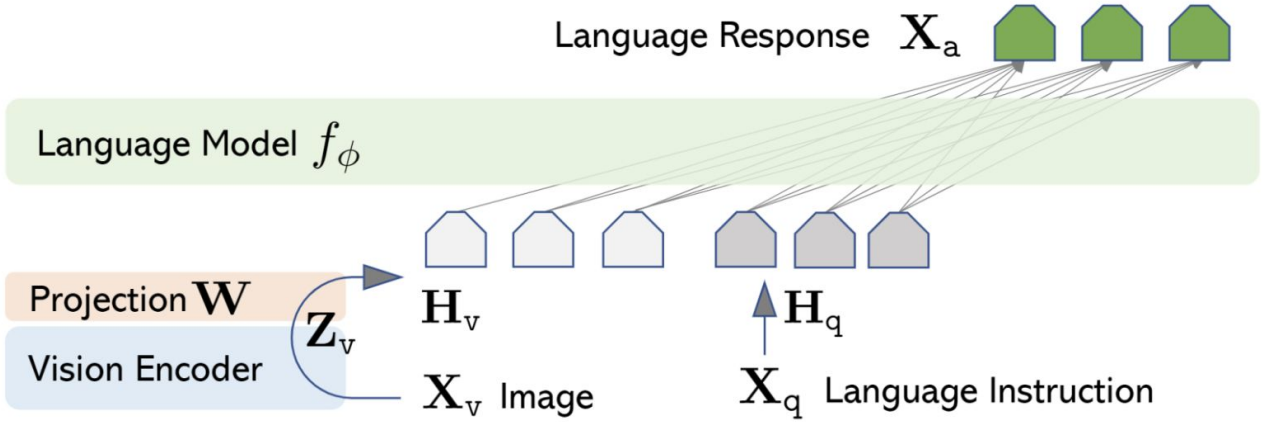
Table 7: Ablation on the synergy between OCR data and image resolution. We pre-trained the models for 5'500 steps, followed by 500 steps of fine-tuning on DocVQA.

State-of-the-art Vision Language models (VLMs) - Training efficiency with multimodal instruction tuning data

Llava suite of VLMs models which are on far less data while performing similar to SOTA VLMs.

- **Stage 1: Pre-training for Feature Alignment.** Only the projection matrix is updated, based on a subset of CC3M.
- **Stage 2: Fine-tuning End-to-End.** Both the projection matrix and LLM are updated for two different use senarios:
  - **Visual Chat**: LLaVA is fine-tuned on our generated multimodal instruction-following data for daily user-oriented applications.
  - **Science QA**: LLaVA is fine-tuned on this multimodal reasonsing dataset for the science domain.

Please check out our [Model Zoo].



State-of-the-art Vision Language models (VLMs) - Training efficiency with multimodal instruction tuning data

Llava suite of VLMs models which are on far less data while performing similar to SOTA VLMs.

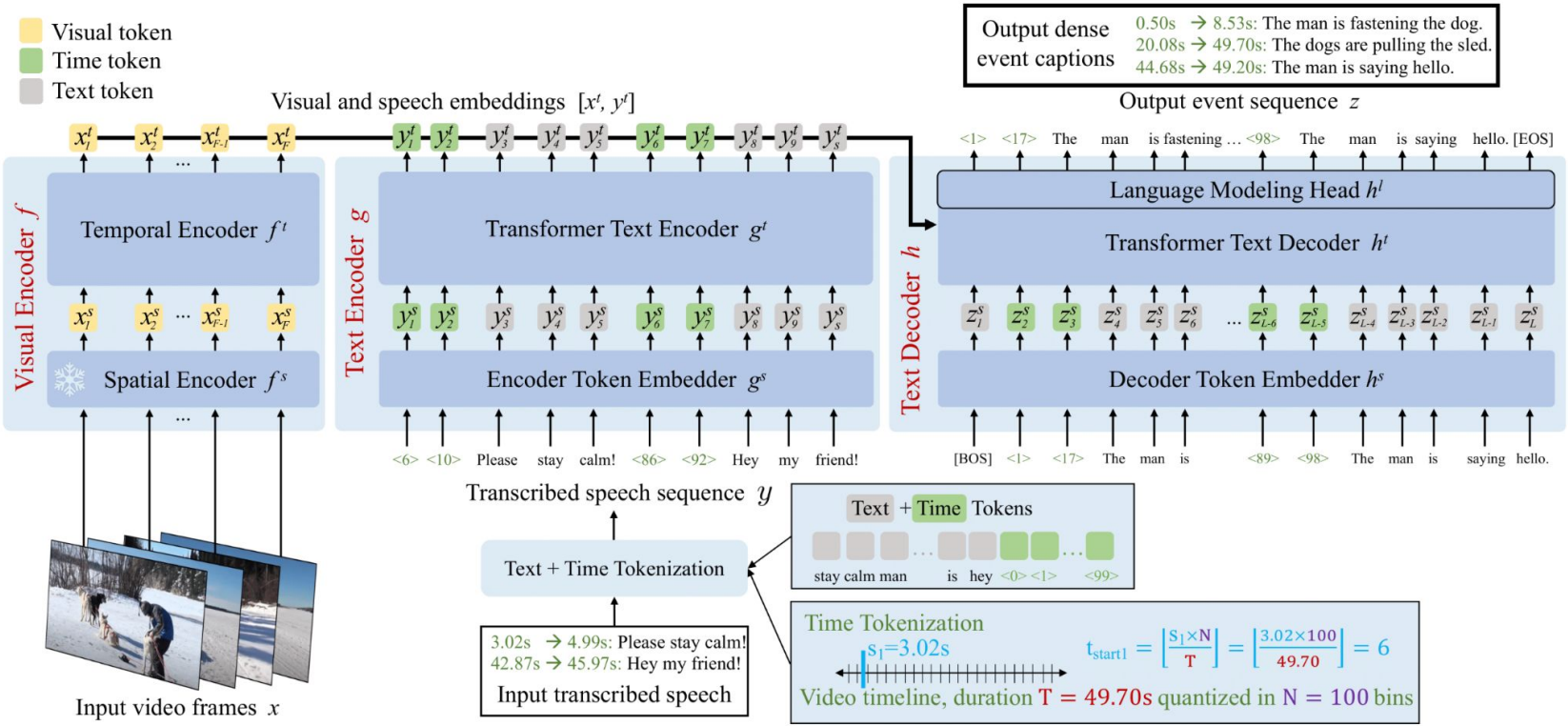
Open-Source Proprietary

Data (PT)	Data (IT)	Model	MMMU (val)	Math-Vista	MMB-ENG	MMB-CN	MM-Vet	LLaVA-Wild	SEED-IMG
N/A	N/A	GPT-4V	56.8	49.9	75.8	73.9	67.6	-	71.6
N/A	N/A	Gemini Ultra	59.4	53	-	-	-	-	-
N/A	N/A	Gemini Pro	47.9	45.2	73.6	74.3	64.3	-	70.7
1.4B	50M	Qwen-VL-Plus	45.2	43.3	-	-	55.7	-	65.7
1.5B	5.12M	CogVLM-30B	32.1	-	-	-	56.8	-	-
125M	~1M	Yi-VL-34B	45.9	-	-	-	-	-	-
558K	665K	LLaVA-1.5-13B	36.4	27.6	67.8	63.3	36.3	72.5	68.2
558K	760K	LLaVA-NeXT-34B	51.1	46.5	79.3	79	57.4	89.6	75.9

- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - **From images to videos**
  - Text-to-Image diffusion models

From images to videos

Vid2Seq - An event captioning video model



From images to videos

Vid2Seq - An event captioning video model

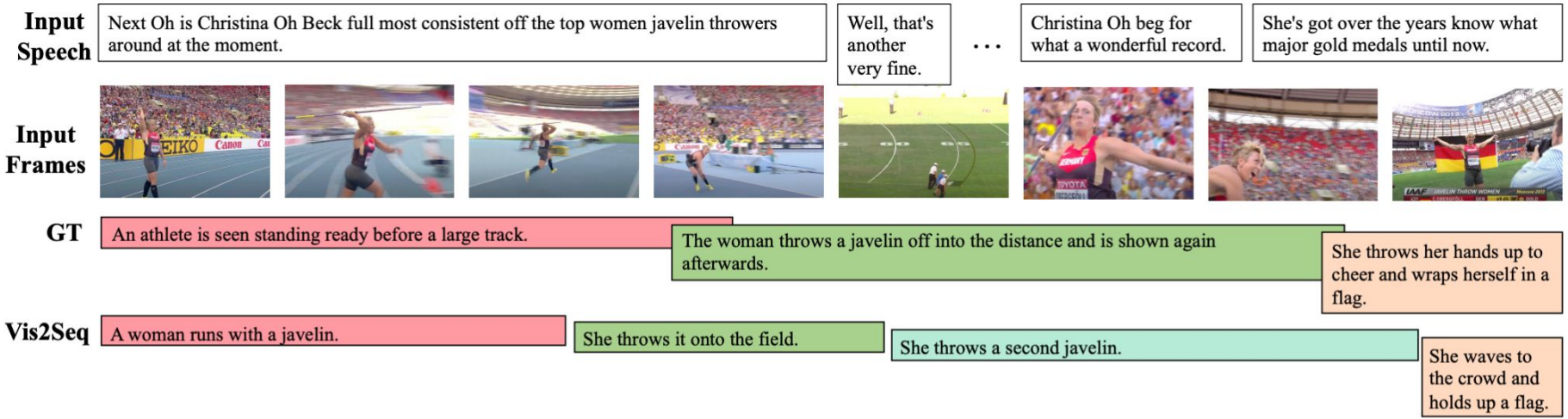
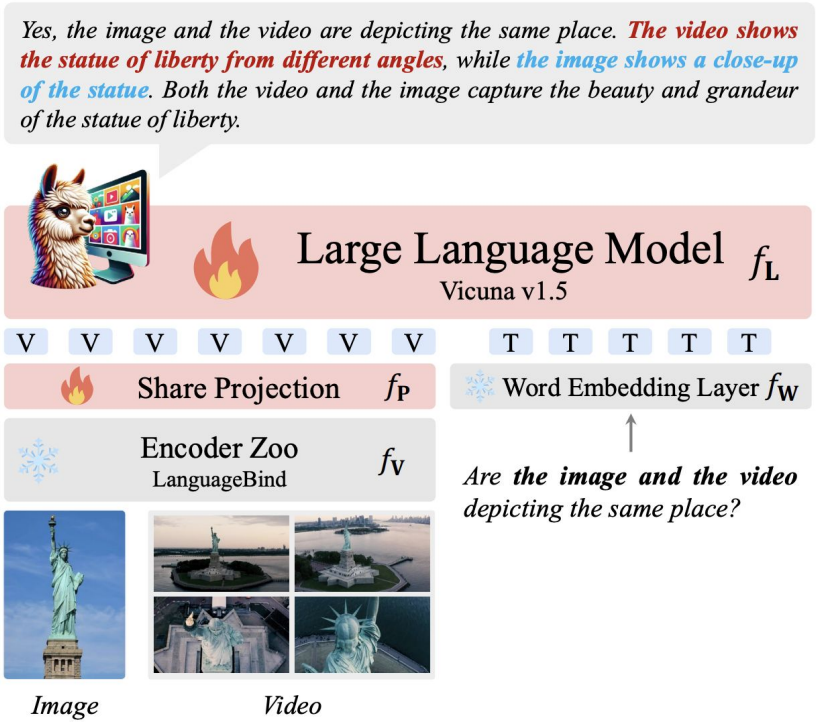


Figure 4. Example of dense event captioning predictions of Vid2Seq on ActivityNet Captions validation set, compared with ground-truth.

Trained on YT-Temporal-1B => 18M
YouTube videos with narrated content

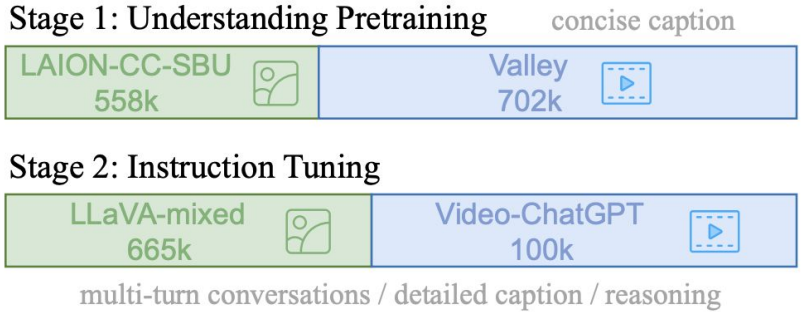
From images to videos

Video-LLaVA



(a) Illustration of Video-LLaVA

- LanguageBind is similar to CLIP but also adapted to videos



- Visual language models
  - Tasks that require processing images (or videos) & text
  - Visual input representations & training objectives
  - Overview of VLM architectures and fusion method
  - Leveraging pretrained language models
  - Visual programming
  - State-of-the-art Vision language models (VLMs)
  - From images to videos
  - **Text-to-Image diffusion models**

Text-to-image diffusion models: brief overview

What are diffusion models?

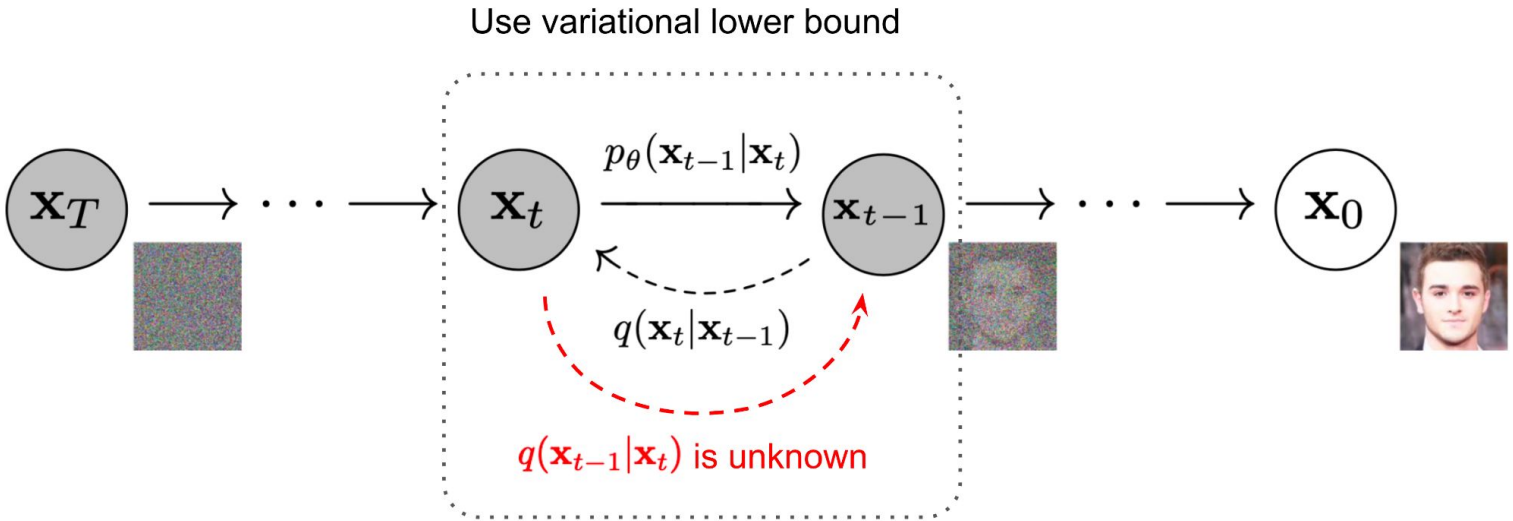


Image from Ho et al. (2020)

Text-to-image diffusion models: brief overview

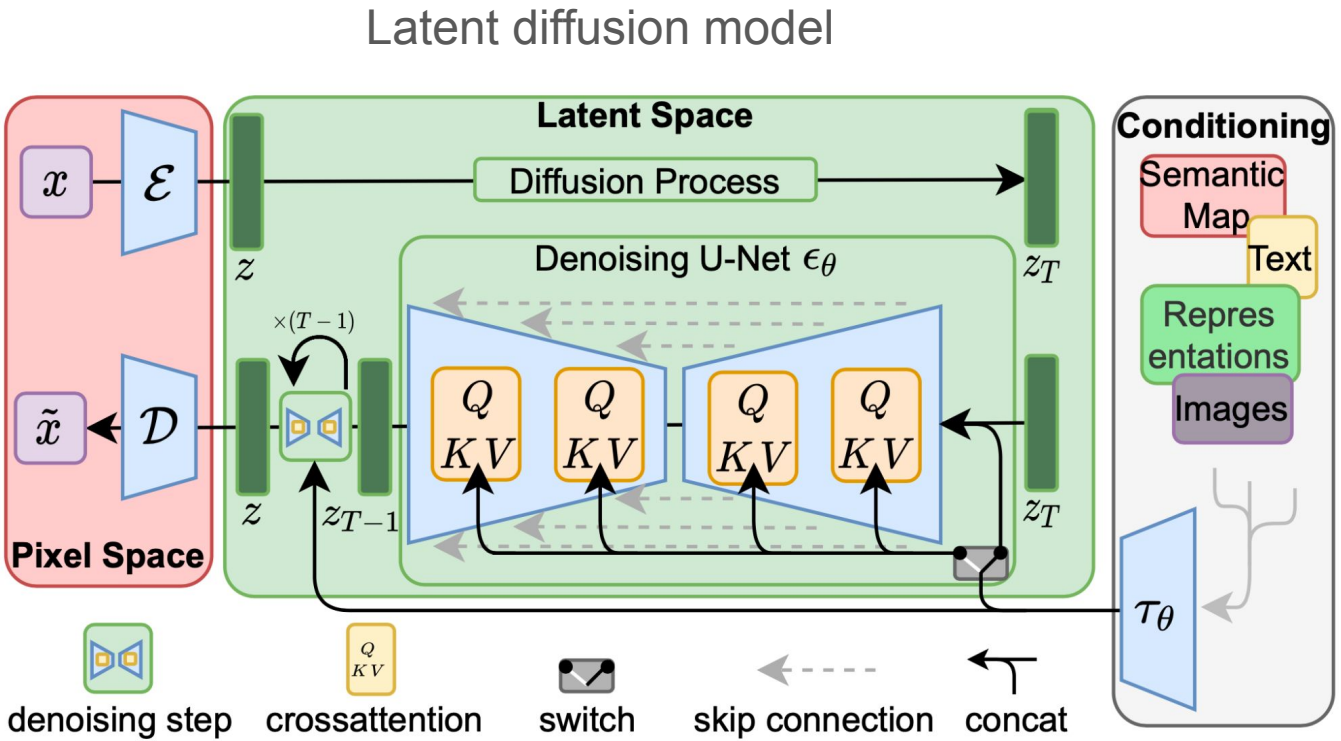


Image from Rombach & Blattmann, et al. (2022)

Text-to-image diffusion models: brief overview

From unconditional to text conditioned diffusion models:
Classifier Guided Diffusion

Given a class y (a class from ImageNet for instance):

Train a classifier: $f_\phi(y|\mathbf{x}_t, t)$

Use gradient to guide the diffusion process: $\nabla_{\mathbf{x}} \log f_\phi(y|\mathbf{x}_t)$

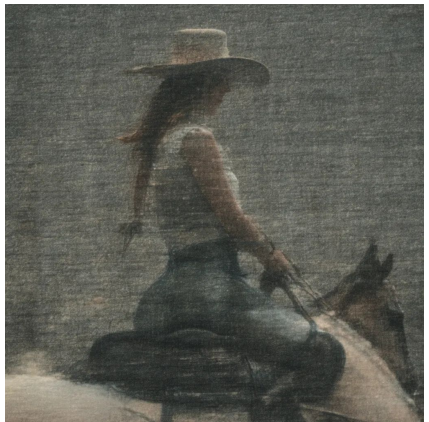
Text-to-image diffusion models: brief overview

From unconditional to text conditioned diffusion models:
Classifier Free guidance

$$\bar{\epsilon}_\theta(\mathbf{x}_t, t, y) = \epsilon_\theta(\mathbf{x}_t, t, y) + w\Big(\epsilon_\theta(\mathbf{x}_t, t, y) - \epsilon_\theta(\mathbf{x}_t, t)\Big)$$

Example with SD3.5 : A woman with a cowboy hat riding a horse.

Cfg 1



Cfg 2.5

Cfg 4.5

Cfg 7.5