VLM Architectures 핵심 요약

(94개의 Model)

https://github.com/gokayfem/awesome-vlm-architectures?tab=readme-ov-file#qwen2-vl-a-powerful-open-source-vision-language-model-for-image-and-video-understanding

1. Unicoder-VL (AAAI 2020)

- **Universal** Encoder
    - Universal = **Vision + Language**
- 3개의 pretraining task
    - Masked language modeling (**MLM**)
    - Masked object classification (**MOC**)
    - Visual-linguistic matching (**VLM**)

1. Unicoder-VL (AAAI 2020)

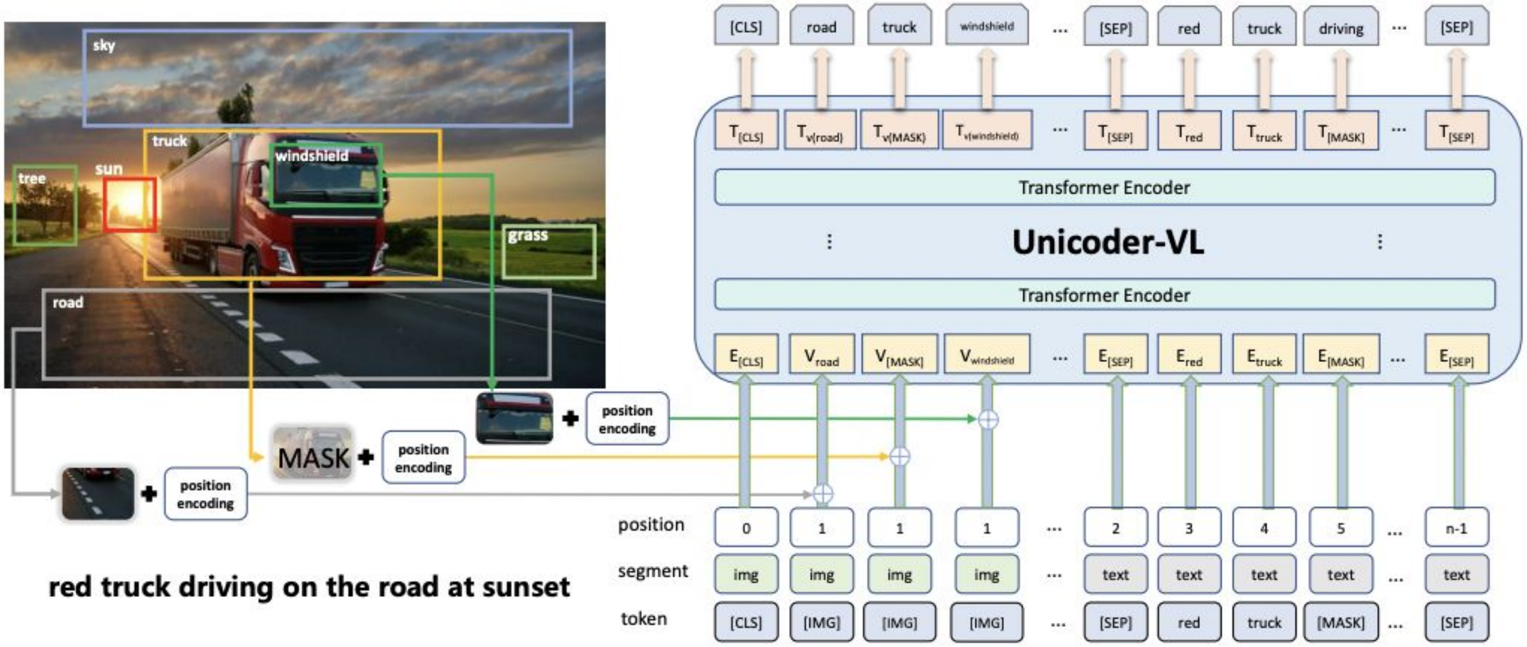
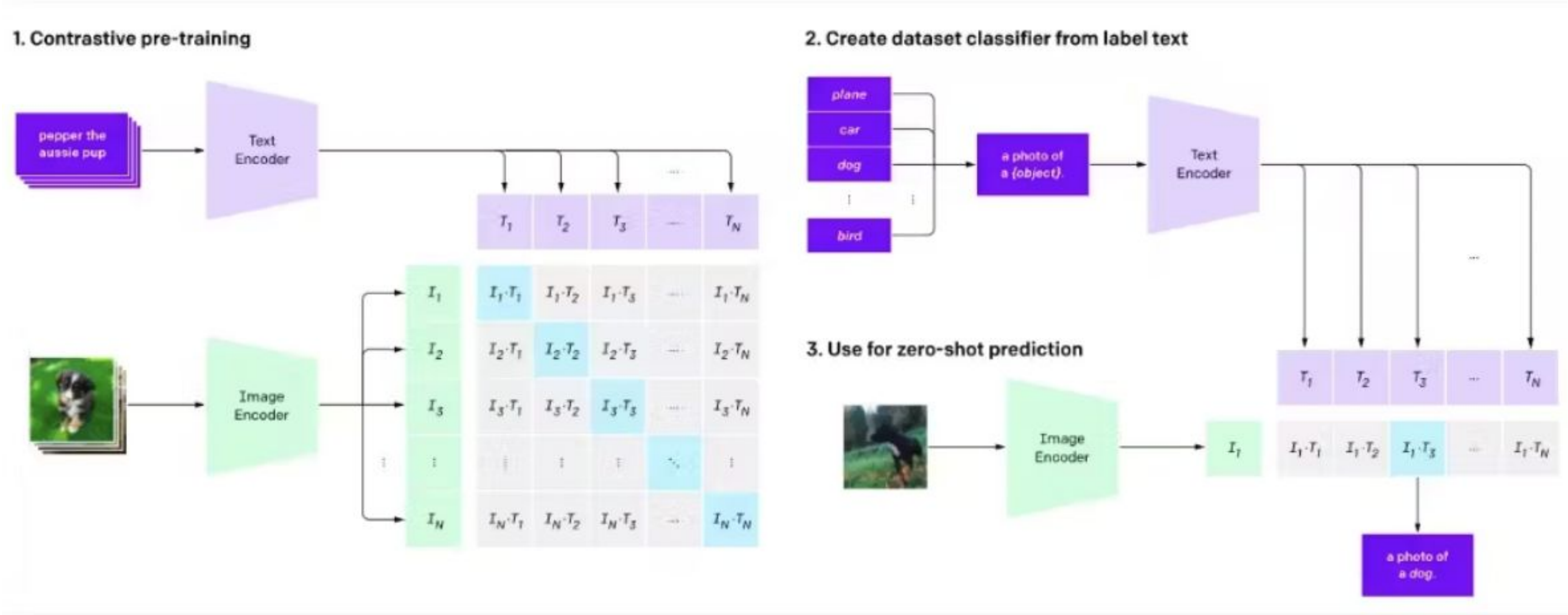


Figure 1: Illustration of Unicoder-VL in the context of an object and text masked token prediction, or *cloze*, task. Unicoder-VL contains multiple Transformer encoders which are used to learn viusal and linguistic representation jointly.

Learning Transferable Visual Models From Natural Language Supervision

2. CLIP (ICML 2021)

- Language & Image “**Contrastive Learning**”


1. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
T_1 T_2 T_3 ... T_N
3. Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

3. MetaCLIP (ICLR 2024)

- Refine the **data curation** process
- Leverage CLIP-derived metadata
- To create a **balanced & high-quality dataset** from vast sources

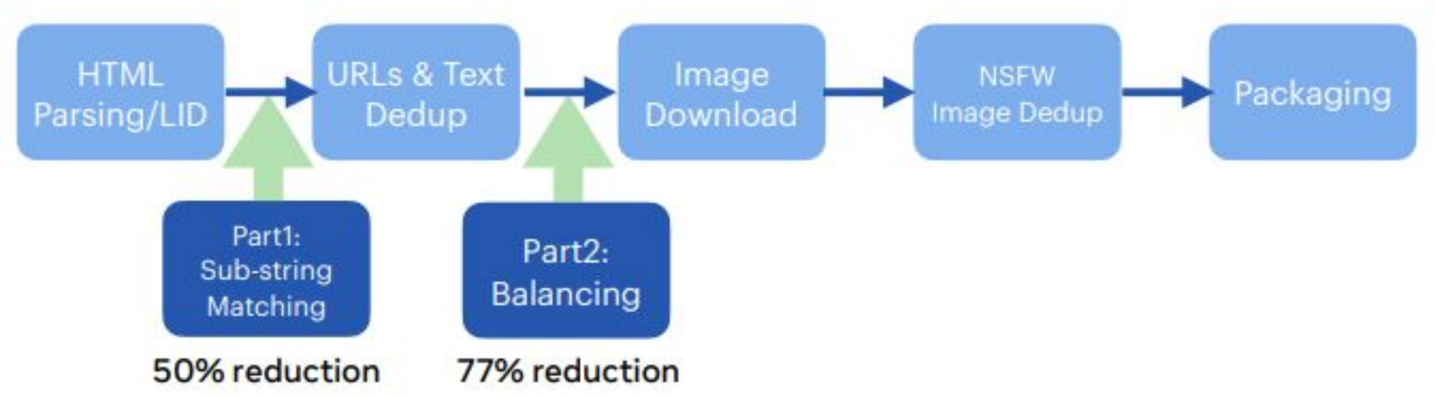
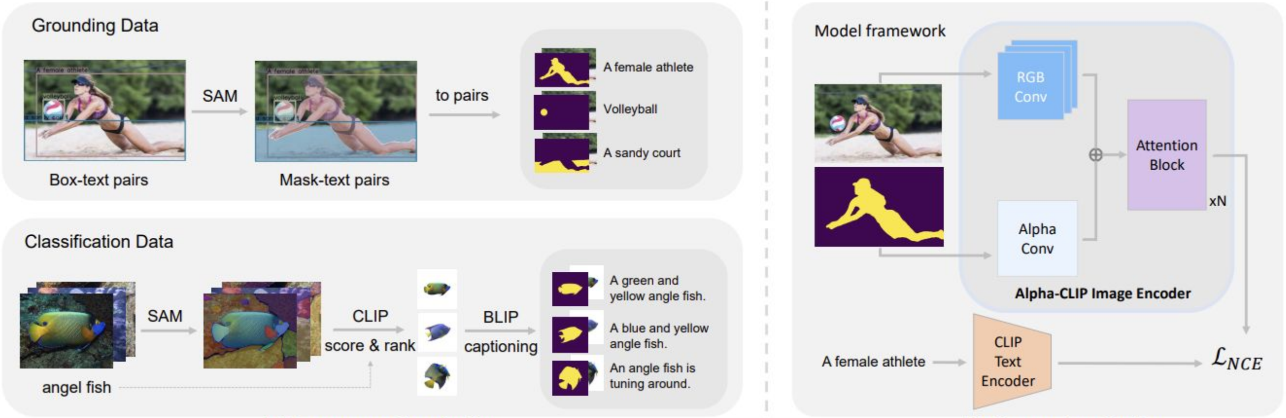


Figure 5: Case study: Curation implementation in our data pipeline.

4. Alpha-CLIP (CVPR 2024)

- CLIP + **Region awareness**
  - Addition of an **alpha channel** to the CLIP encoder
- Allows for detailed **segmentation** & region-specific processing



(a) Data generation pipeline

(b) Fine-tuning pipeline

Figure 3. **The pipeline of our data generation method and model architecture.** **(a)** Our method generates millions of RGBA-region text pairs. **(b)** Alpha-CLIP modifies the CLIP image encoder to take an additional alpha channel along with RGB.

5. GLIP (CVPR 2022)

- GLIP = **Grounded** + LIP
- Pretraining by unifying **object detection & phrase grounding**

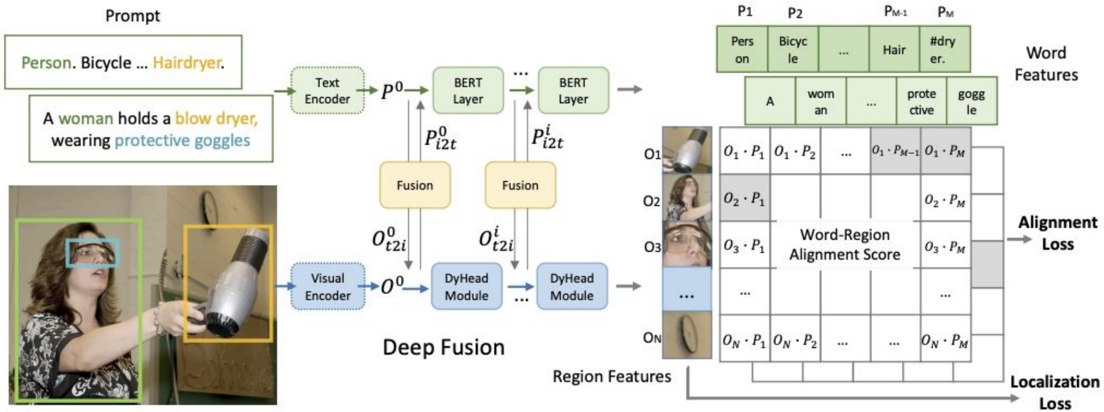
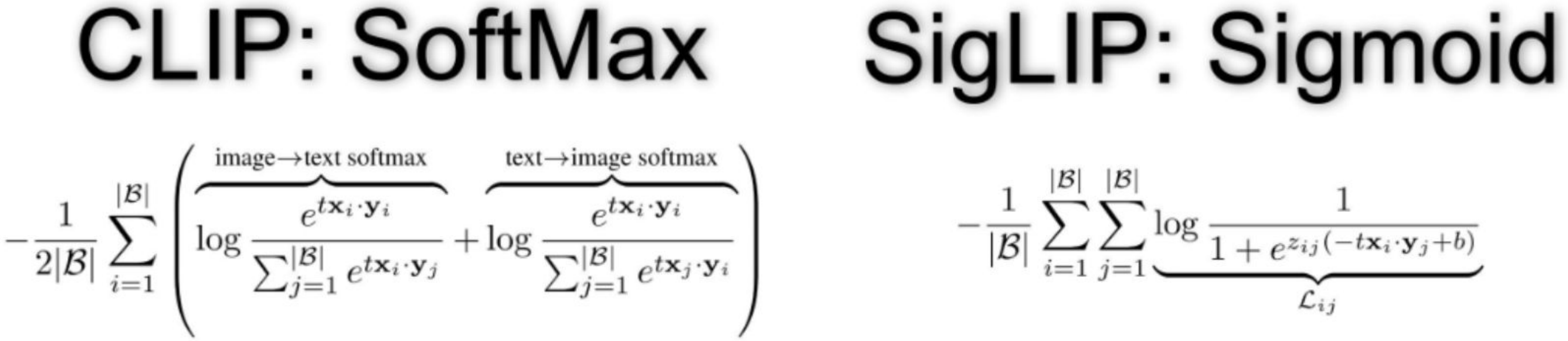


Figure 2. A unified framework for detection and grounding. Unlike a classical object detection model which predicts a categorical class for each detected object, we reformulate detection as a grounding task by aligning each region/box to phrases in a text prompt. GLIP jointly trains an image encoder and a language encoder to predict the correct pairings of regions and words. We further add the cross-modality deep fusion to early fuse information from two modalities and to learn a language-aware visual representation.

6. SigLIP (CVPR 2022)

- Simple pairwise **sigmoid** loss
- Scalable training with **large batch sizes**



CLIP: SoftMax

$$-\frac{1}{2|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left(\overbrace{\log\frac{e^{t\mathbf{x}_i\cdot\mathbf{y}_i}}{\sum_{j=1}^{|\mathcal{B}|}e^{t\mathbf{x}_i\cdot\mathbf{y}_j}}}^{\text{image}\rightarrow\text{text softmax}}+\overbrace{\log\frac{e^{t\mathbf{x}_i\cdot\mathbf{y}_i}}{\sum_{j=1}^{|\mathcal{B}|}e^{t\mathbf{x}_j\cdot\mathbf{y}_i}}}^{\text{text}\rightarrow\text{image softmax}}\right)$$

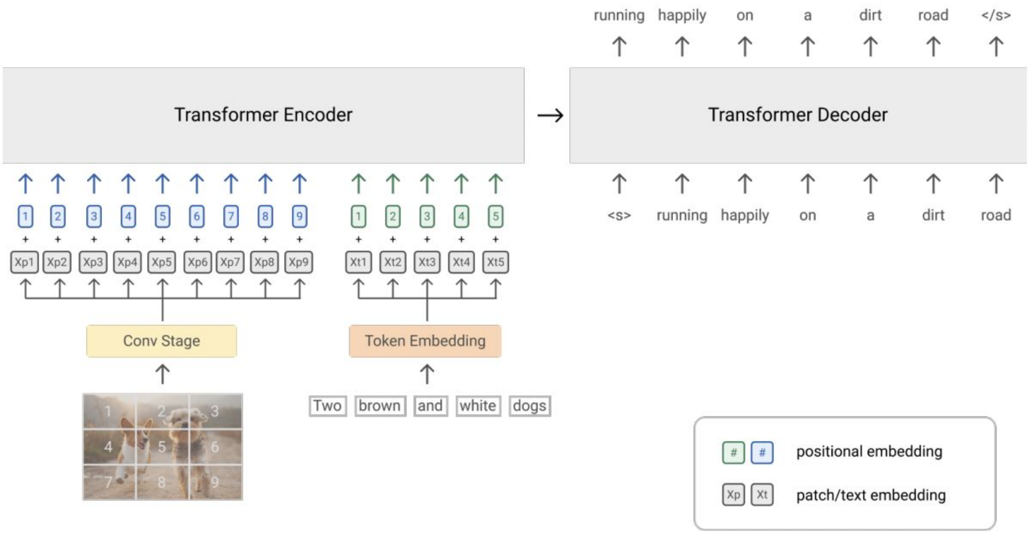
SigLIP: Sigmoid

$$-\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\sum_{j=1}^{|\mathcal{B}|}\underbrace{\log\frac{1}{1+e^{z_{ij}(-t\mathbf{x}_i\cdot\mathbf{y}_j+b)}}}_{\mathcal{L}_{ij}}$$

SimVLM: Simple Visual Language Model Pretraining with Weak Supervision

7. SimVLM (ICLR 2022)

- **Prefix**LM
  - X = **(Prefix) Image** + Text
  - Y = Text
- Task: **Next word prediction**



8. VirTex (CVPR 2021)

- Pretraining task = **Image Captioning**

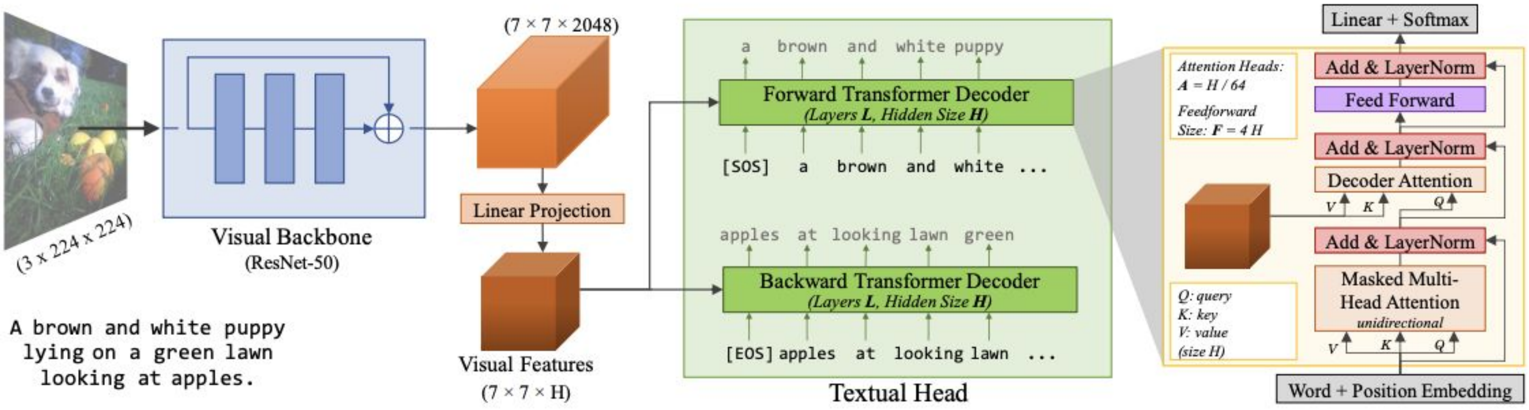


Figure 3: **VirTex pretraining setup:** Our model consists of a *visual backbone* (ResNet-50), and a *textual head* (two unidirectional Transformers). The visual backbone extracts image features, and textual head predicts captions via bidirectional language modeling (*bicaptioning*). The Transformers perform masked multiheaded self-attention over caption features, and multiheaded attention over image features. Our model is trained end-to-end from scratch. After pretraining, the visual backbone is transferred to downstream visual recognition tasks.

9. Frozen (NeurIPS 2021)

- PrefixLM (e.g., SimVLM)
  - X = (Prefix) Image + Text
  - Y = Text
- **Frozen PrefixLM**
  - Use “Pretrained” LLM

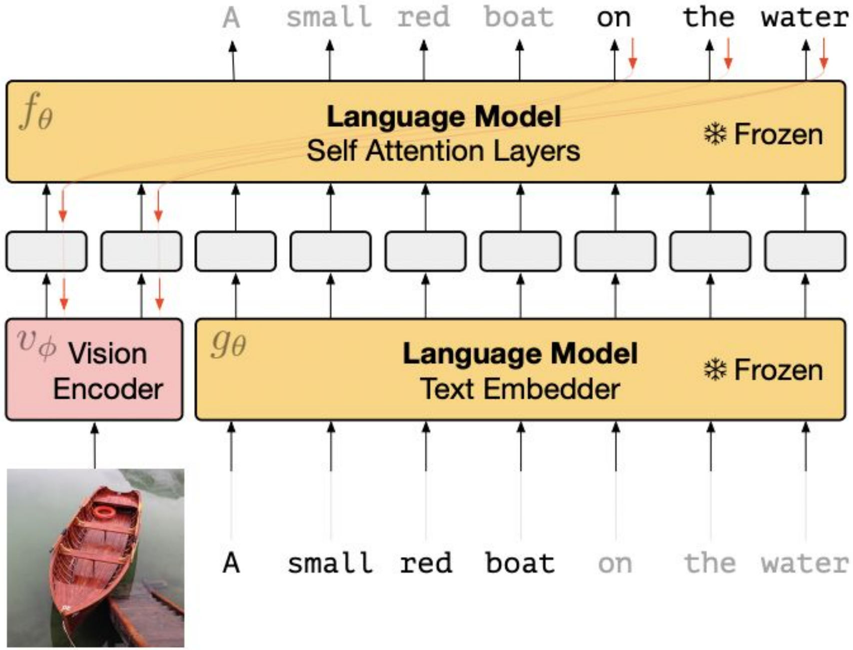


Figure 2: Gradients through a frozen language model's self attention layers are used to train the vision encoder.

9. Frozen (NeurIPS 2021)

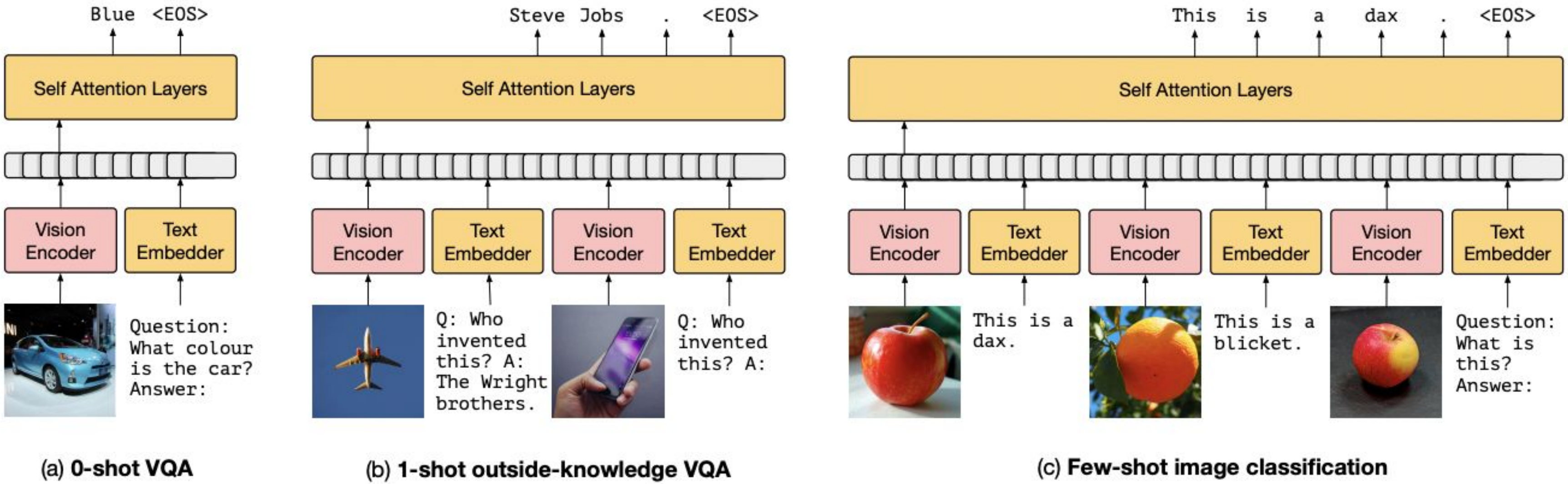


Figure 3: Inference-Time interface for *Frozen*. The figure demonstrates how we can support (a) visual question answering, (b) outside-knowledge question answering and (c) few-shot image classification via in-context learning.

10. Flamingo (NeurIPS 2022)

- Model Architecture
    - LLM (**freeze**): Chinchilla
    - Vision (**freeze**): CLIP
- **Arbitrarily interleaved** visual & textual tokens
- **Cross-attention** for **multimodal fusion**
- **Perceiver Resampler**
    - Input: Visual features
    - Output: **Fixed number** of visual tokens

10. Flamingo (NeurIPS 2022)

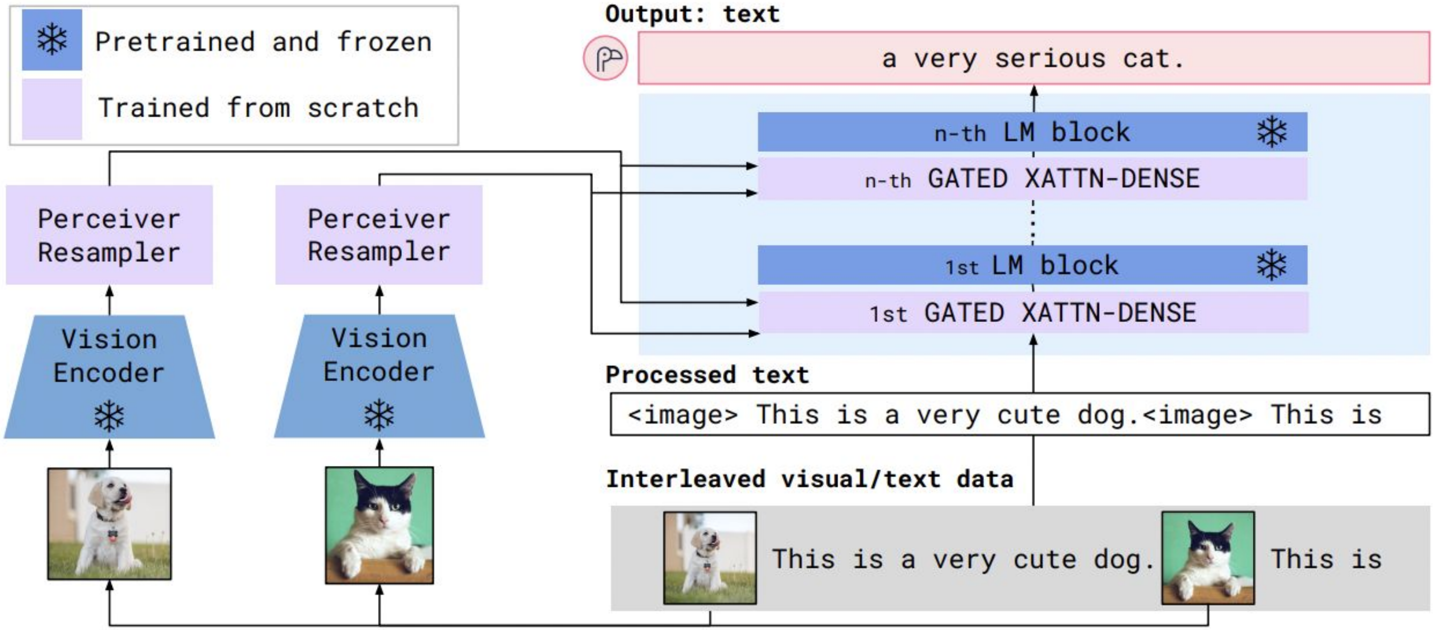


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

11. VisualGPT (CVPR 2022)

- **Cross-attention** for **multimodal fusion**
- Architecture
  - Encoder: Visual encoder
  - Decoder: LLM
- **Self-resurrecting** enc-dec attention
  - To adapt LLM for visual tasks

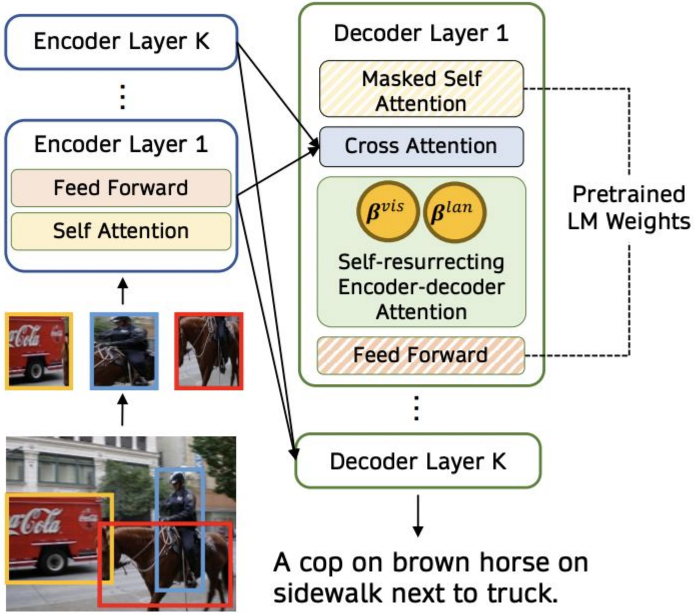
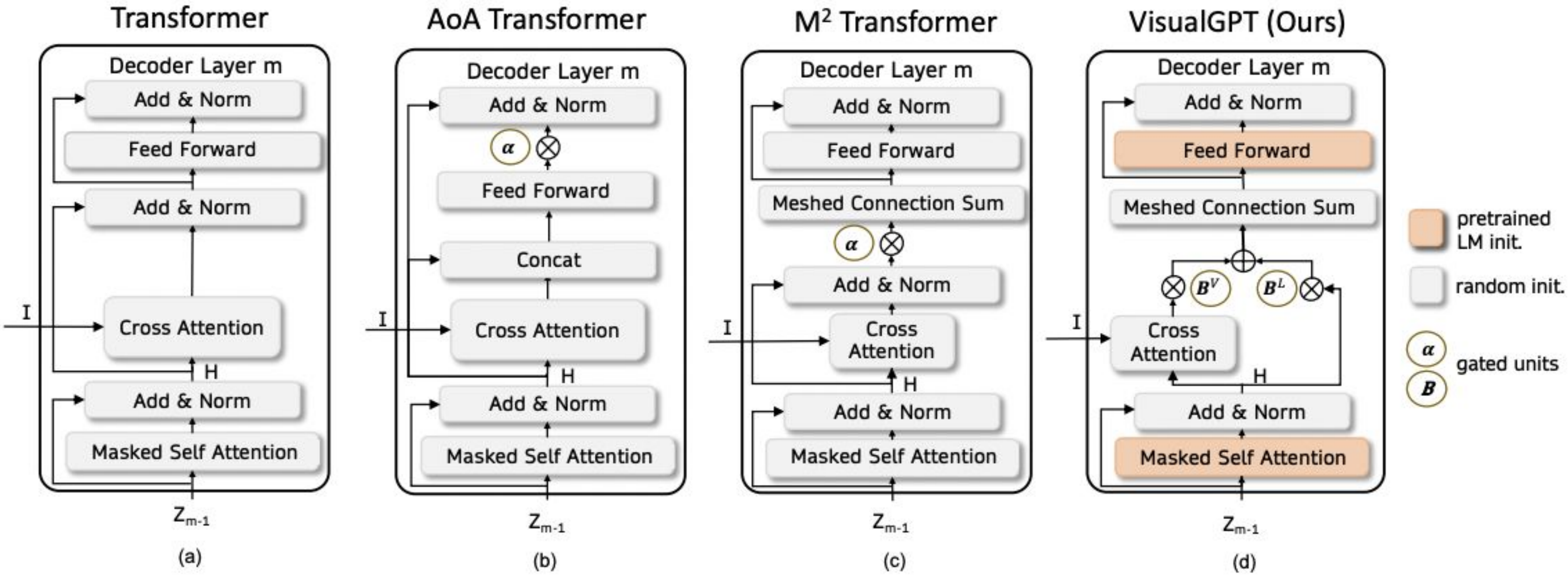


Figure 1. Our VisualGPT model transfers the knowledge from a pre-trained language model to the caption decoder. A self-resurrecting encoder-decoder attention is designed to connect the multi-level visual features and caption decoder.

11. VisualGPT (CVPR 2022)



VisualBERT: A Simple and Performant Baseline for Vision and Language

12. VisualBERT (ACL 2020)

- Two pretraining tasks = **MLM + ITM**

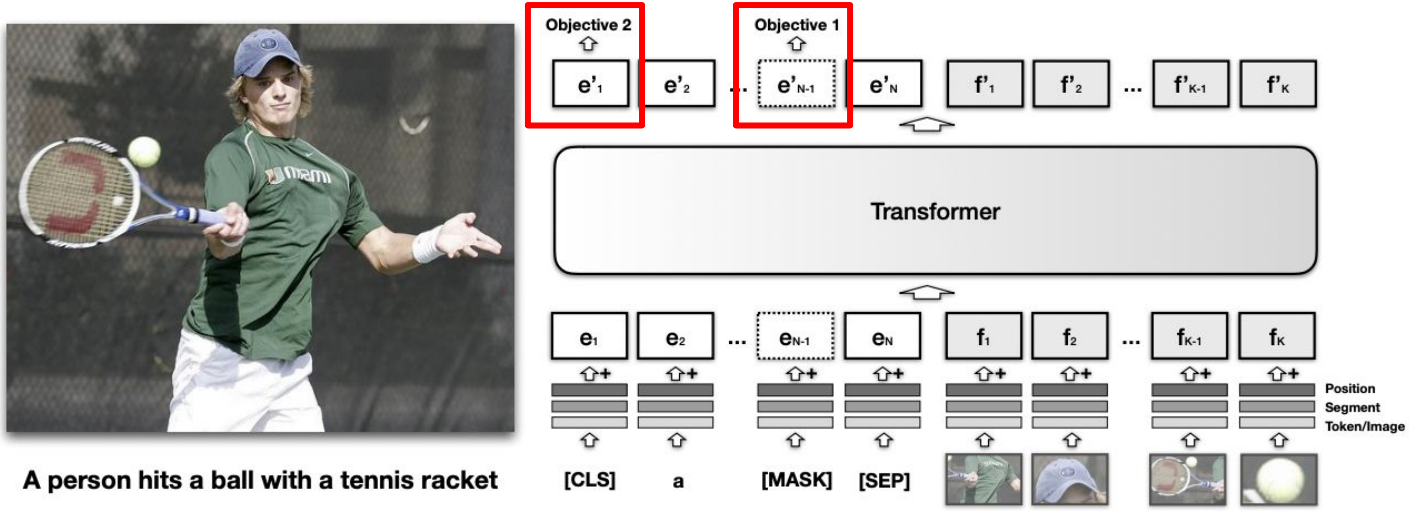
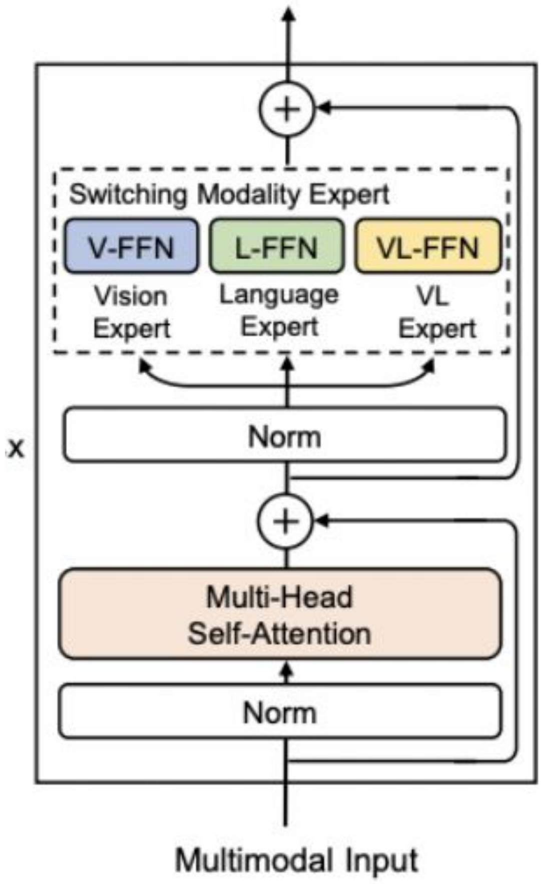


Figure 2: The architecture of VisualBERT. Image regions and language are combined with a Transformer to allow the self-attention to discover implicit alignments between language and vision. It is pre-trained with a masked language modeling (Objective 1), and sentence-image prediction task (Objective 2), on caption data and then fine-tuned for different tasks. See §3.3 for more details.

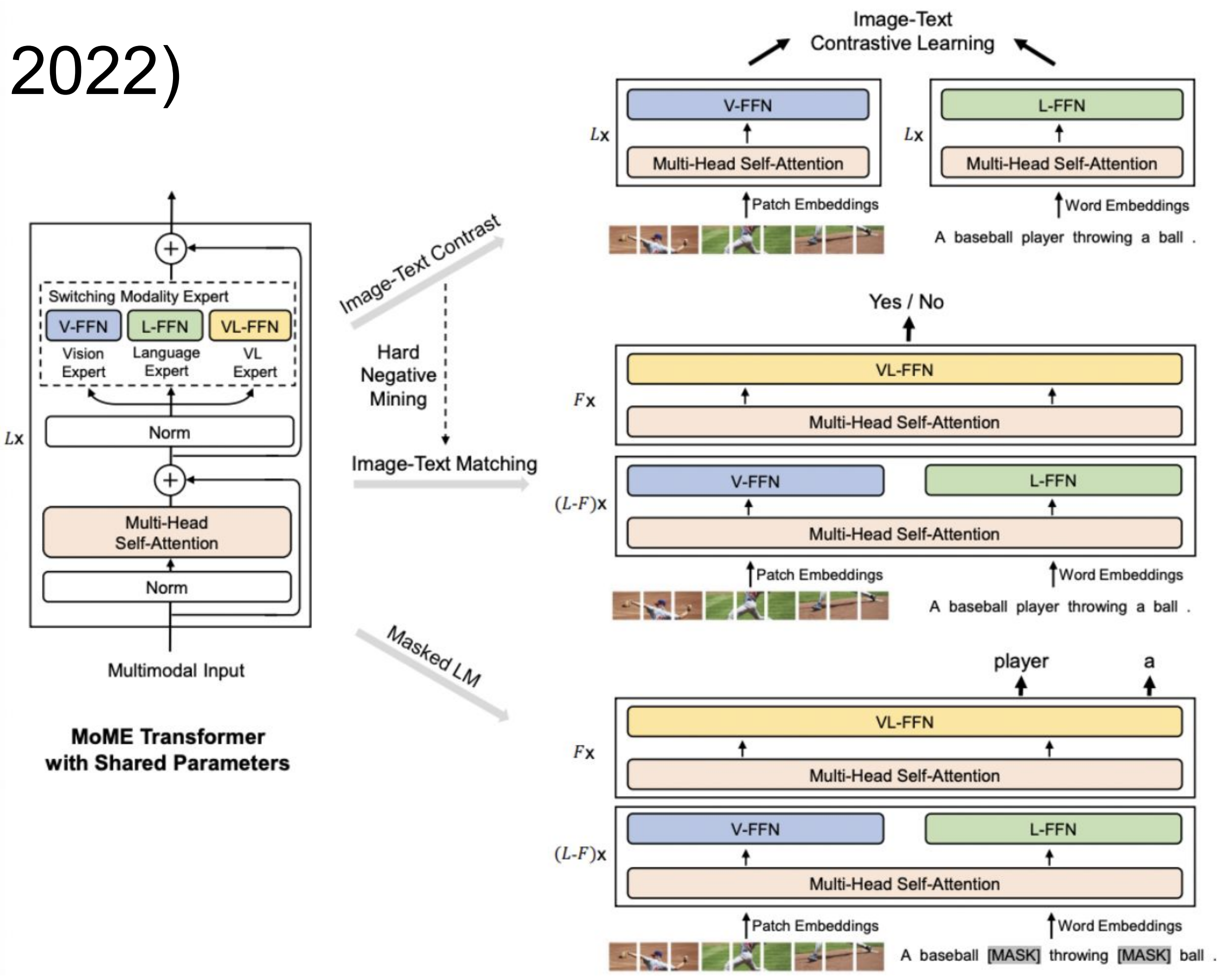
13. VLMo (NeurIPS 2022)

- **MoME** (Mixture-of-modality-experts)
  - **각 Modality 별**로 expert 존재
  - V용 / L용 / VL용
- 두 가지 역할 모두 가능!
  - **Dual** encoder: 각각 embedding
  - **Fusion** encoder: 같이 embedding



13. VLMo (NeurIPS 2022)

- 세 가지 pretraining
    - **ITC, ITM, LM**


Switching Modality Expert
V-FFN
L-FFN
VL-FFN
Vision Expert
Language Expert
VL Expert
Norm
Multi-Head Self-Attention
Norm
Lx
Multimodal Input
MoME Transformer with Shared Parameters
Image-Text Contrast
Hard Negative Mining
Image-Text Matching
Masked LM
Image-Text Contrastive Learning
V-FFN
Multi-Head Self-Attention
L-FFN
Multi-Head Self-Attention
Lx
Lx
Patch Embeddings
Word Embeddings
A baseball player throwing a ball .
Yes / No
VL-FFN
Multi-Head Self-Attention
Fx
V-FFN
L-FFN
Multi-Head Self-Attention
(L-F)x
Patch Embeddings
Word Embeddings
A baseball player throwing a ball .
player
a
VL-FFN
Multi-Head Self-Attention
Fx
V-FFN
L-FFN
Multi-Head Self-Attention
(L-F)x
Patch Embeddings
Word Embeddings
A baseball [MASK] throwing [MASK] ball .

ASIF: Coupled Data Turns Unimodal Models to Multimodal Without Training

14. ASIF (NeurIPS 2023)

- **Training-free**

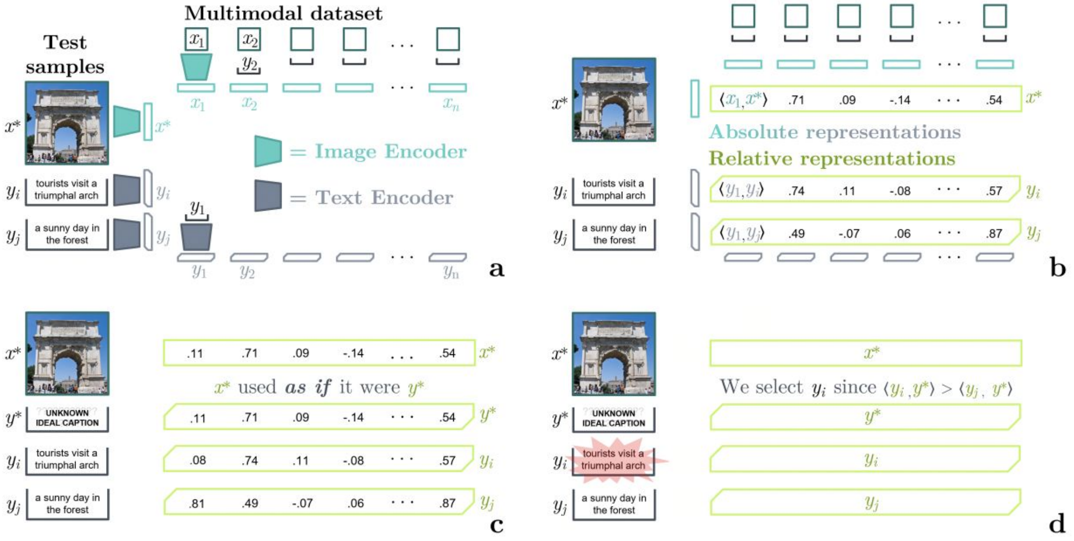

Multimodal dataset
Test samples
x^*
x_1
x_2
y_2
x_n
y_1
y_i
y_j
tourists visit a triumphal arch
a sunny day in the forest
= Image Encoder
= Text Encoder
y_n
a
$\langle x_1, x^* \rangle$ .71 .09 -.14 54
Absolute representations
Relative representations
$\langle y_1, y_i \rangle$ .74 .11 -.08 57
$\langle y_1, y_j \rangle$ .49 -.07 .06 87
b
.11 .71 .09 -.14 54
x^* used *as if* it were y^*
y^*
UNKNOWN IDEAL CAPTION
.11 .71 .09 -.14 54
.08 .74 .11 -.08 57
.81 .49 -.07 .06 87
c
We select y_i since $\langle y_i, y^* \rangle > \langle y_j, y^* \rangle$
d

Figure 4: **Zero shot classification with ASIF.** In this example we determine the best caption for the image x^* from the two candidates, y_i and y_j. **a.** Compute and store the embeddings of the multimodal dataset and the test samples. **b.** Compute the relative representation of the test image and the candidate captions. **c.** We consider the relative representation of x^* with respect to the image collection $x_1, \dots, x_n$ *as if* it was the relative representation of y^* – the ideal caption for x^* – with respect to the corresponding caption collection $y_1, \dots, y_n$. **d.** We choose the candidate caption most similar to the ideal one.

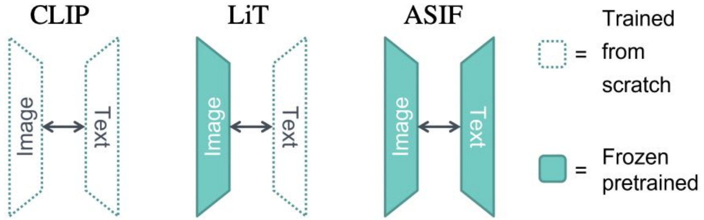
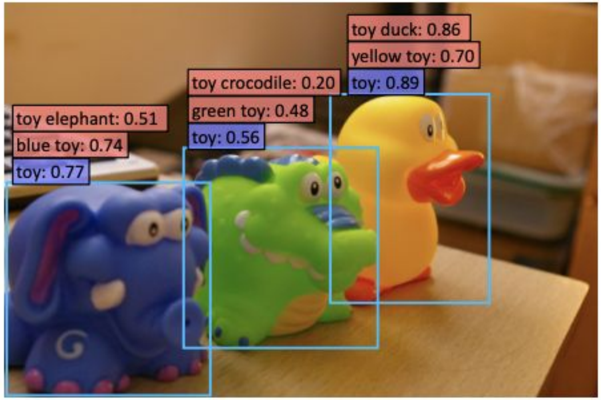
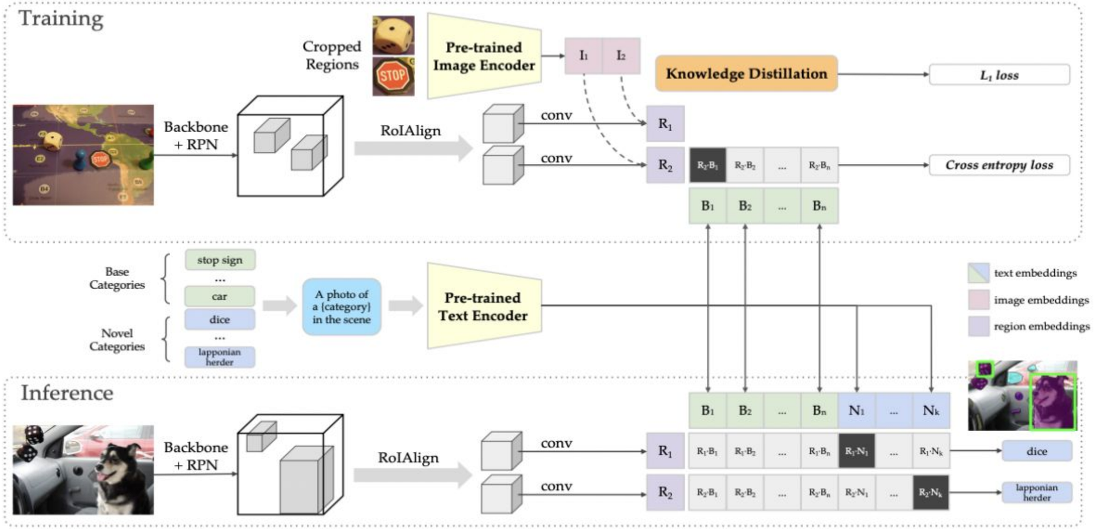


Figure 1: ASIF is a simple recipe to align the representations of two frozen pre-trained encoders.

15. ViLD (ICLR 2022)

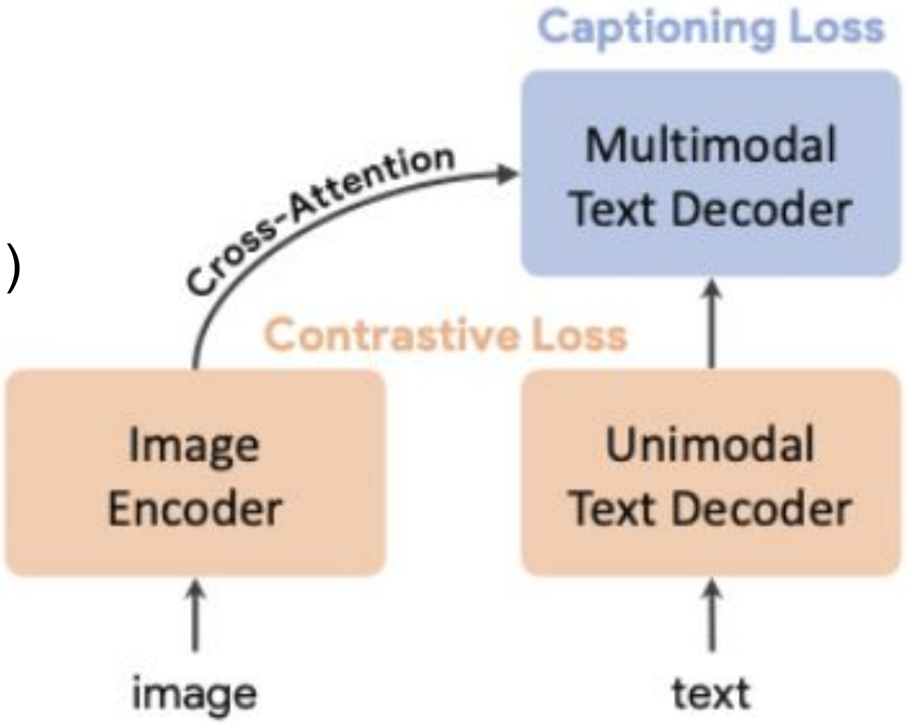
- **Knowledge Distillation**
  - Teacher = (Pretrained) **Open-vocab** CLS model
  - Student = Two-stage detector





16. CoCa (TMLR 2022)

- CoCa = Contrastive Captioner
- 핵심: **2가지의 joint loss**
  - **Contrastive** loss (feat. CLIP)
  - **Captioning** loss (feat. SimVLM)



CoCa; Contrastive Captioners are Image-Text Foundation Models

16. CoCa (TMLR 2022)

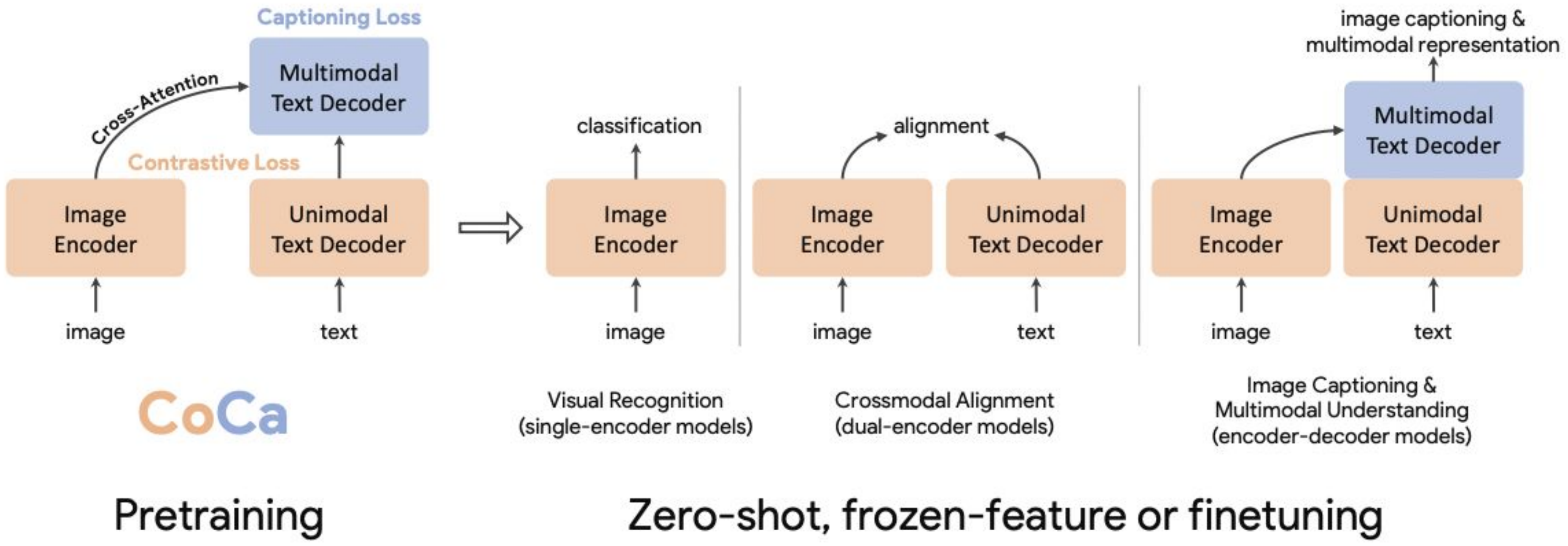
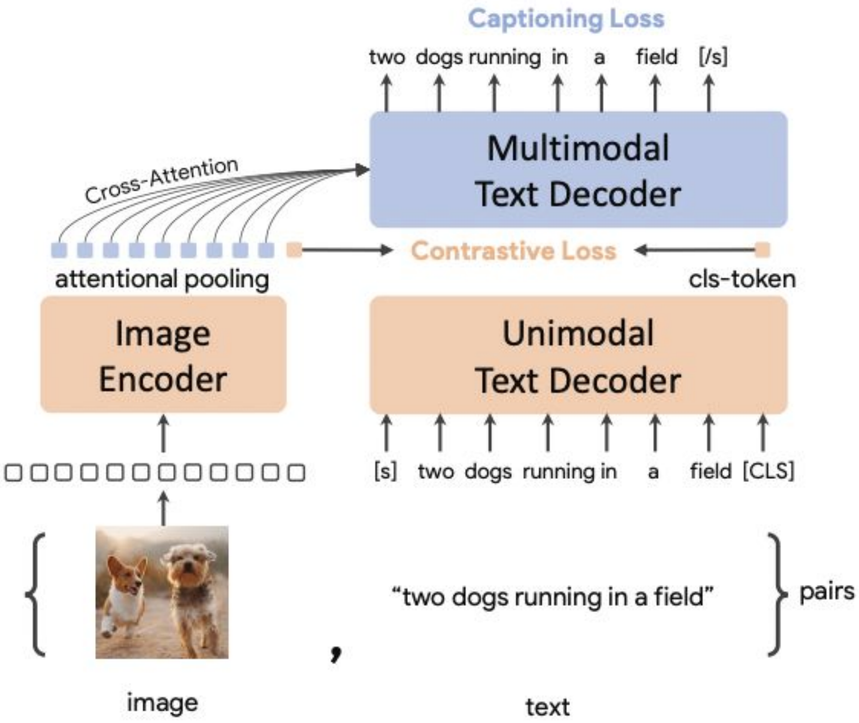
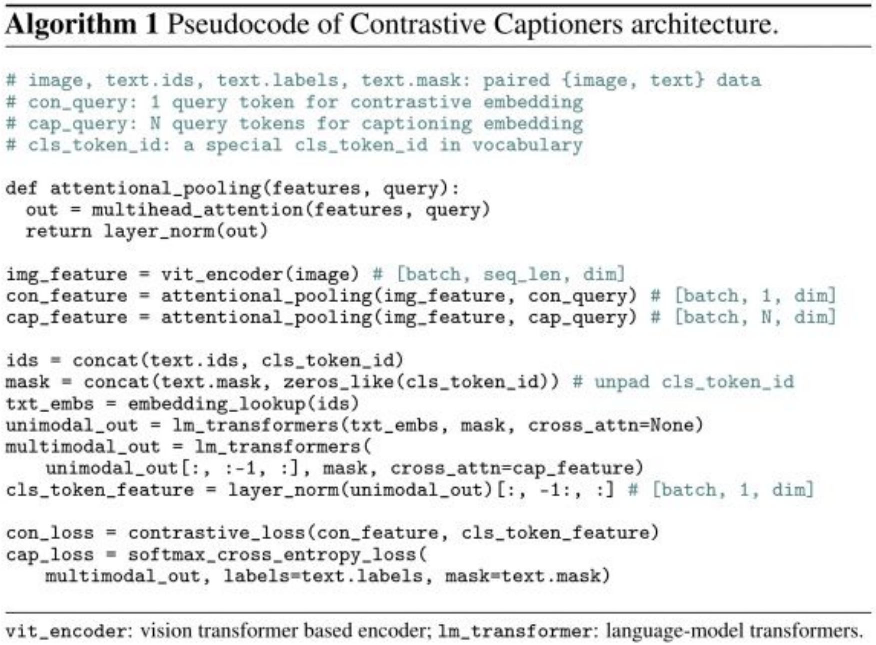


Figure 1: Overview of Contrastive Captioners (CoCa) pretraining as image-text foundation models. The pretrained CoCa can be used for downstream tasks including visual recognition, vision-language alignment, image captioning and multimodal understanding with zero-shot transfer, frozen-feature evaluation or end-to-end finetuning.

CoCa; Contrastive Captioners are Image-Text Foundation Models

16. CoCa (TMLR 2022)





Algorithm 1 Pseudocode of Contrastive Captioners architecture.

```
# image, text.ids, text.labels, text.mask: paired {image, text} data
# con_query: 1 query token for contrastive embedding
# cap_query: N query tokens for captioning embedding
# cls_token_id: a special cls_token_id in vocabulary

def attentional_pooling(features, query):
  out = multihead_attention(features, query)
  return layer_norm(out)

img_feature = vit_encoder(image) # [batch, seq_len, dim]
con_feature = attentional_pooling(img_feature, con_query) # [batch, 1, dim]
cap_feature = attentional_pooling(img_feature, cap_query) # [batch, N, dim]

ids = concat(text.ids, cls_token_id)
mask = concat(text.mask, zeros_like(cls_token_id)) # unpad cls_token_id
txt_embs = embedding_lookup(ids)
unimodal_out = lm_transformers(txt_embs, mask, cross_attn=None)
multimodal_out = lm_transformers(
    unimodal_out[:, :-1, :], mask, cross_attn=cap_feature)
cls_token_feature = layer_norm(unimodal_out)[:, -1:, :] # [batch, 1, dim]

con_loss = contrastive_loss(con_feature, cls_token_feature)
cap_loss = softmax_cross_entropy_loss(
    multimodal_out, labels=text.labels, mask=text.mask)
```

vit_encoder: vision transformer based encoder; lm_transformer: language-model transformers.

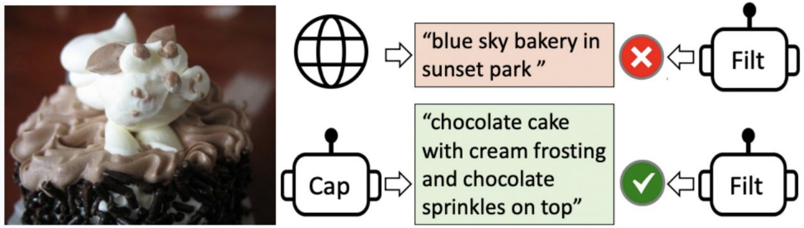
Figure 2: Detailed illustration of CoCa architecture and training objectives.

17. BLIP (ICML 2022)

- Proposal 2개
  - Model 측면: Multimodal mixture of Encoder-Decoder (**MED**)
  - Data 측면: **CapFilt**
- 세 가지 Pretraining
  - ITC loss: CL loss
  - ITM loss: Binary CE loss
  - LM: CE loss

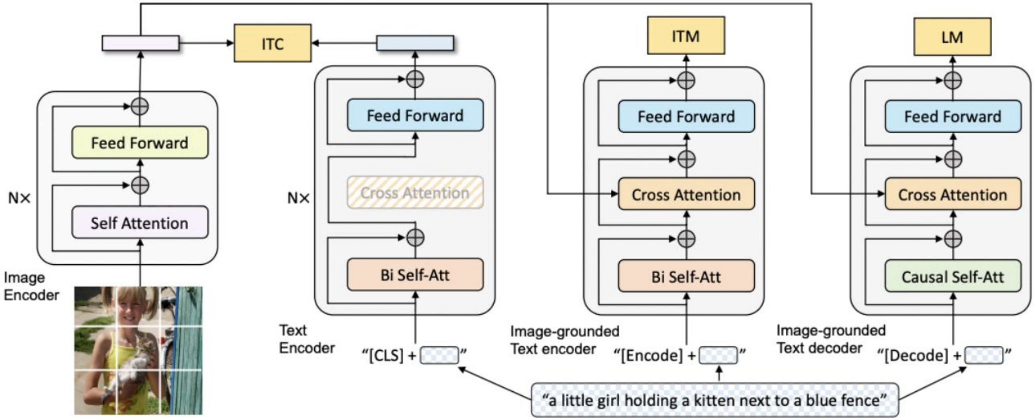
17. BLIP (ICML 2022)

- [1] CapFilt: **Bootstrap the caption**!
  - 배경: Noisy image-text pairs
  - (1) **Captioner**: Synthetic caption 생성
    - Image-grounded text decoder
    - LM으로 fine-tune
  - (2) **Filter**: Noisy caption 제거
    - Image grounded text encoder
    - ITC & ITM으로 fine-tune


"blue sky bakery in sunset park"
✗
Filt
Cap
"chocolate cake with cream frosting and chocolate sprinkles on top"
✓
Filt

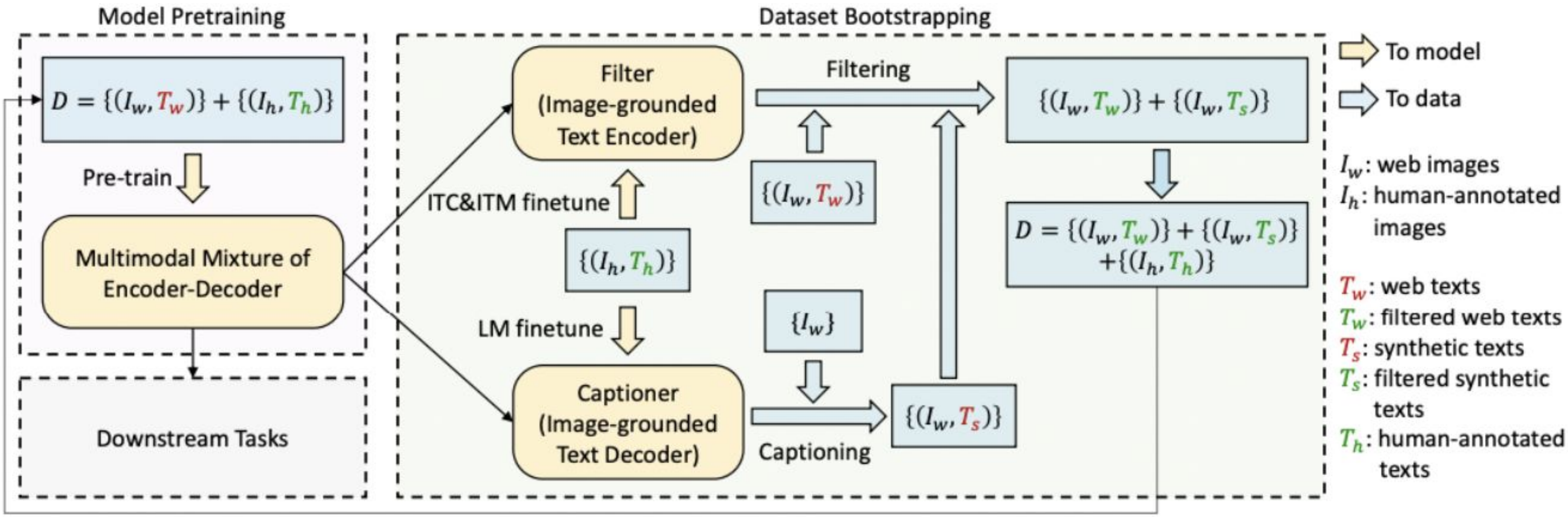
17. BLIP (ICML 2022)

- [2] **MED**: 아래의 3개 중 하나의 기능으로 작동 가능
  - Unimodal encoder
  - Image-grounded text encoder
  - Image-grounded text decoder



17. BLIP (ICML 2022)

- Learning framework of BLIP

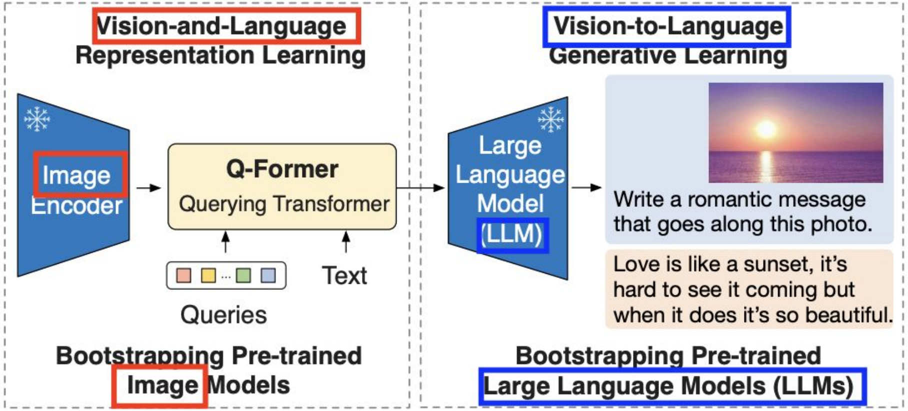

Model Pretraining
$D = \{(I_w, T_w)\} + \{(I_h, T_h)\}$
Pre-train
Multimodal Mixture of Encoder-Decoder
Downstream Tasks
Dataset Bootstrapping
Filter (Image-grounded Text Encoder)
ITC&ITM finetune
$\{(I_h, T_h)\}$
LM finetune
Captioner (Image-grounded Text Decoder)
Filtering
$\{(I_w, T_w)\}$
$\{I_w\}$
Captioning
$\{(I_w, T_s)\}$
$\{(I_w, T_w)\} + \{(I_w, T_s)\}$
$D = \{(I_w, T_w)\} + \{(I_w, T_s)\} + \{(I_h, T_h)\}$
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

18. BLIP-2 (ICML 2023)

- **Q(Querying)-Former**
- 목적: Modality gap을 bridge
- 2단계 학습
  - 1) (Frozen image 인코더)

    VL representation learning
  - 2) (Frozen language 인코더)

    V-to-L generation learning



Write a romantic message that goes along this photo.

Love is like a sunset, it's hard to see it coming but when it does it's so beautiful.

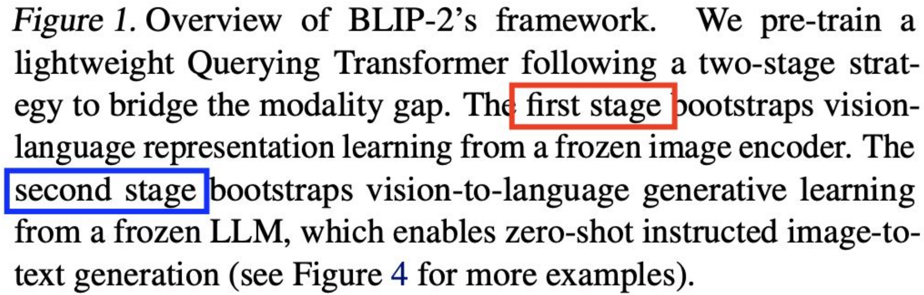
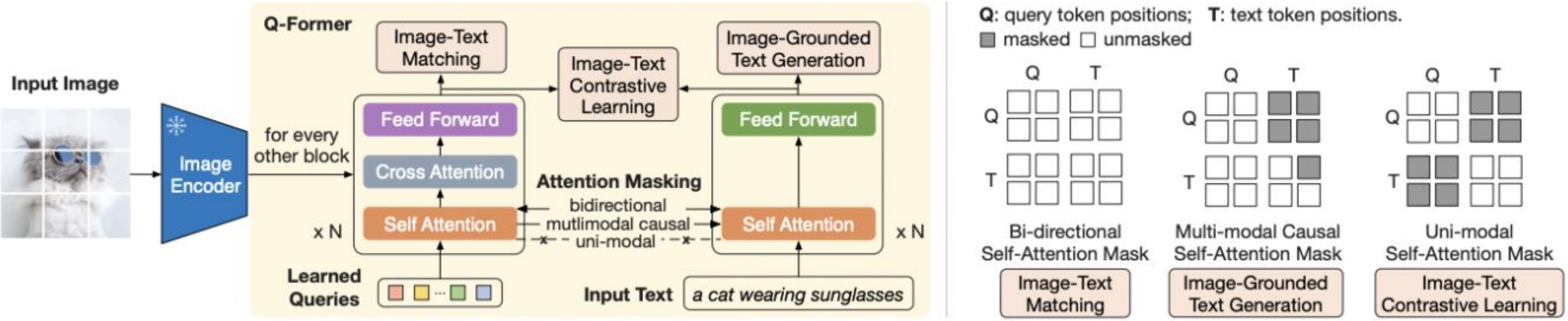


Figure 1. Overview of BLIP-2's framework. We pre-train a lightweight Querying Transformer following a two-stage strategy to bridge the modality gap. The first stage bootstraps vision-language representation learning from a frozen image encoder. The second stage bootstraps vision-to-language generative learning from a frozen LLM, which enables zero-shot instructed image-to-text generation (see Figure 4 for more examples).

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

18. BLIP-2 (ICML 2023)

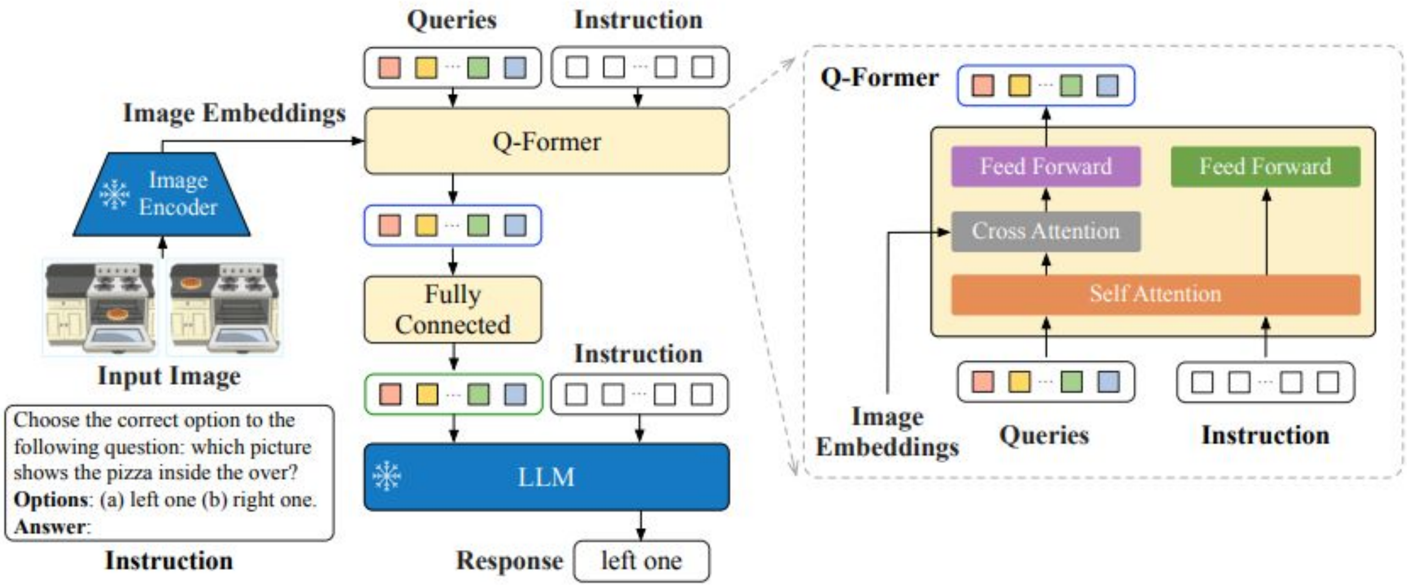
- 2개의 transformer = **Image** transformer + **Text** transformer
- Pretraining
  - (BLIP과 비슷하게) **ITC + ITG + ITM**
  - 각 objective별로 다른 attention masking strategy



InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning

19. InstructBLIP (NeurIPS 2023)

- BLIP2를 개선함.
- **Q-Former를 instruction tuning**함

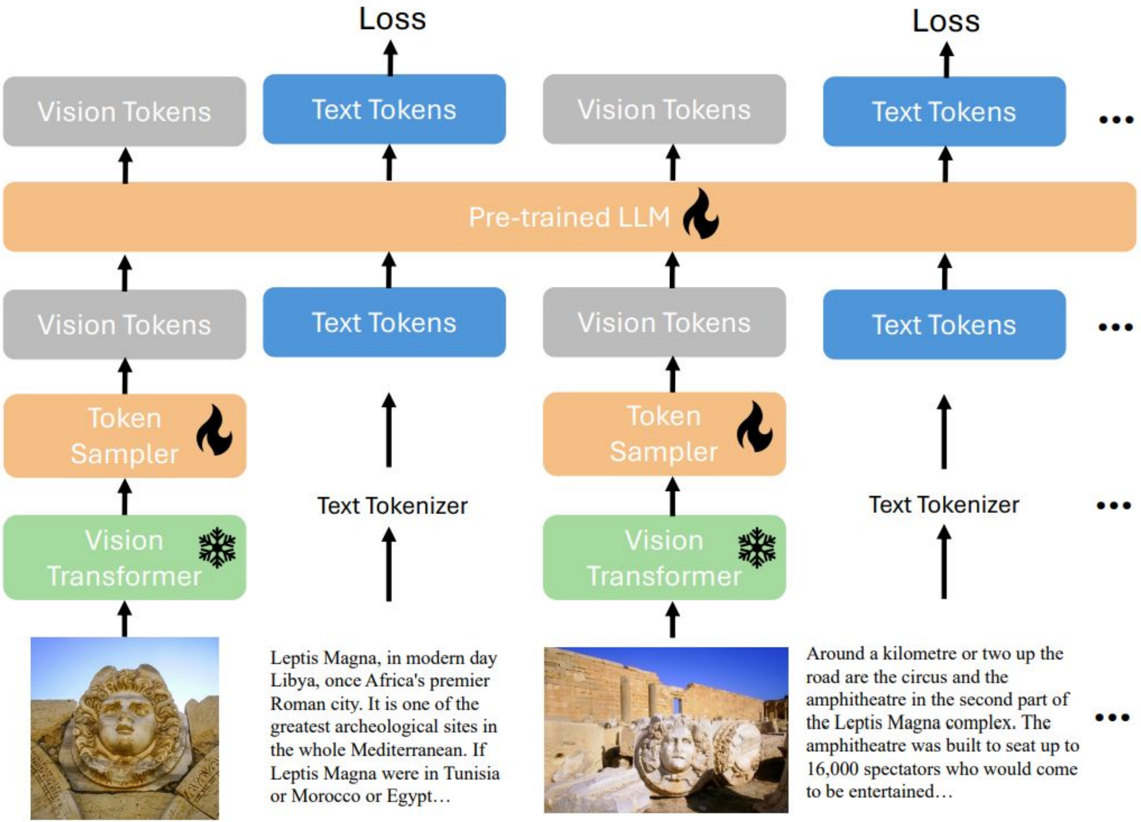


Choose the correct option to the following question: which picture shows the pizza inside the over? **Options**: (a) left one (b) right one. **Answer**:

20. xGen-MM (BLIP-3) (ICML 2023)

- Diverse, large-scale, high-quality multimodal data
- **BLIP-2**:
  - (a) **Q-Former**
  - (b) **Multiple pretraining objectives**
- **BLIP-3**:
  - (a) More scalable vision token sampler (**perceiver resampler**)
  - (b) Unifying the training objective to a **single autoregressive loss**

20. xGen-MM (BLIP-3) (ICML 2023)

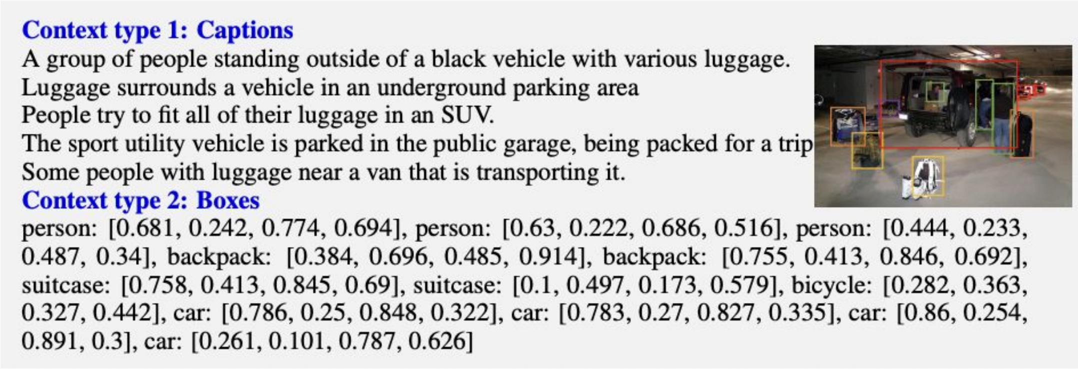


Leptis Magna, in modern day Libya, once Africa's premier Roman city. It is one of the greatest archeological sites in the whole Mediterranean. If Leptis Magna were in Tunisia or Morocco or Egypt…

Around a kilometre or two up the road are the circus and the amphitheatre in the second part of the Leptis Magna complex. The amphitheatre was built to seat up to 16,000 spectators who would come to be entertained…

21. LLaVA (NeurIPS 2023)

- Goal: **Multimodal 버전의 Instruction-following dataset**을 생성하자!
- How: **“text” 데이터만**을 사용해서 (feat. GPT-4)
- (1) + (2) 두 가지 정보를 준 뒤, 문제/답변 생성을 LLM에 요청
  - **(1) Caption**
  - **(2) Bounding boxes**
- 생성한 3종류의 데이터셋:
  - Conversation
  - Detailed description
  - Complex Reasoning



Context type 1: Captions

A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

Context type 2: Boxes

person: [0.681, 0.242, 0.774, 0.694], person: [0.63, 0.222, 0.686, 0.516], person: [0.444, 0.233, 0.487, 0.34], backpack: [0.384, 0.696, 0.485, 0.914], backpack: [0.755, 0.413, 0.846, 0.692], suitcase: [0.758, 0.413, 0.845, 0.69], suitcase: [0.1, 0.497, 0.173, 0.579], bicycle: [0.282, 0.363, 0.327, 0.442], car: [0.786, 0.25, 0.848, 0.322], car: [0.783, 0.27, 0.827, 0.335], car: [0.86, 0.254, 0.891, 0.3], car: [0.261, 0.101, 0.787, 0.626]

21. LLaVA (NeurIPS 2023)

- Model architecture
  - LLM: Vicuna
  - Vision encoder: CLIP (ViT-L/14) + **Linear projection layer**

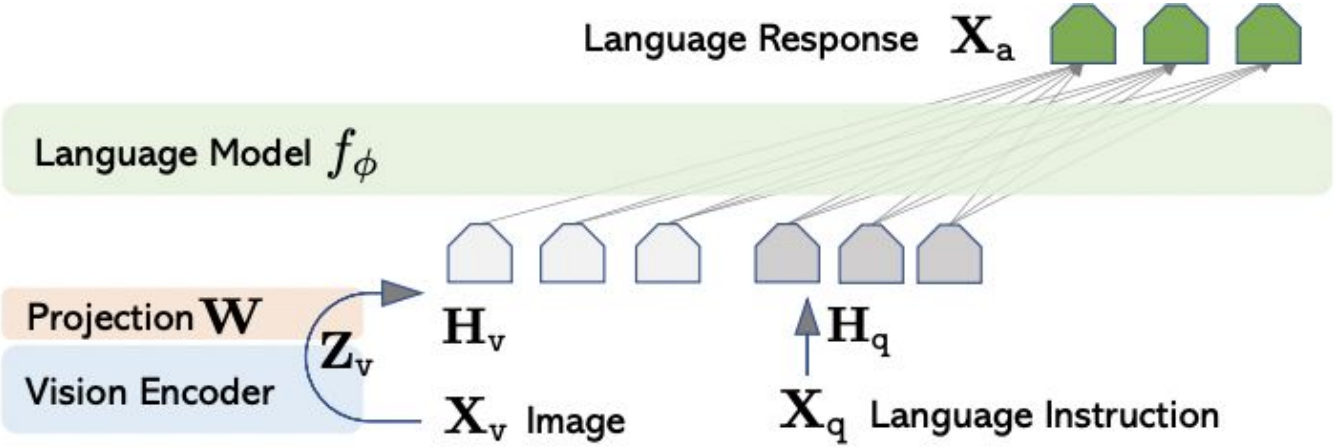
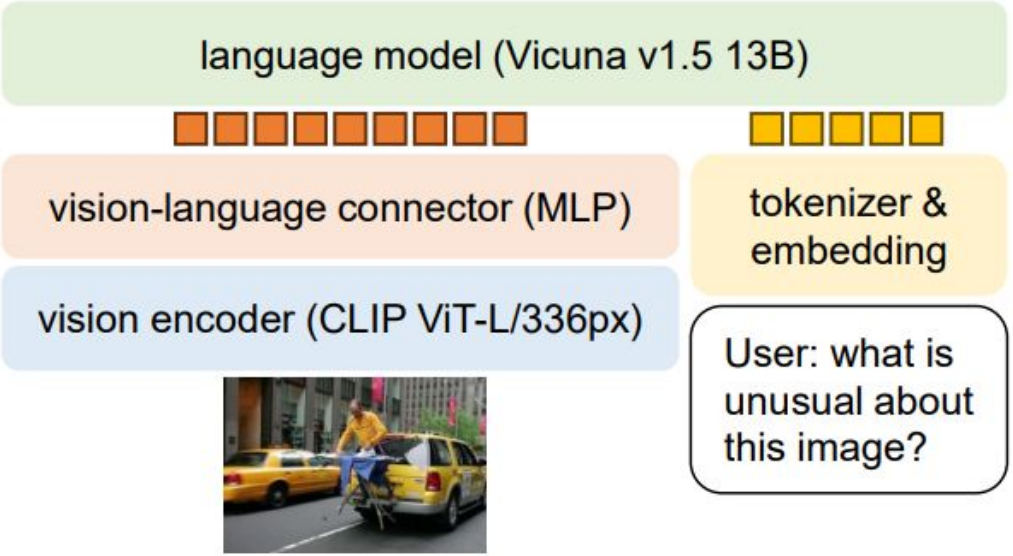


Figure 1: LLaVA network architecture.

22. LLaVA 1.5 (CVPR 2024)

- LLaVA: **Linear** projection layer
- LLaVA 1.5: **MLP** projection layer


language model (Vicuna v1.5 13B)
vision-language connector (MLP)
tokenizer & embedding
vision encoder (CLIP ViT-L/336px)
User: what is unusual about this image?

23. LLaVA-NeXT (LLaVA 1.6) (blog)

- LLaVA1.5 + alpha
  - **Dynamic high resolution**: High-resolution image processing
  - OCR capabilities
  - Efficient deployment & inference

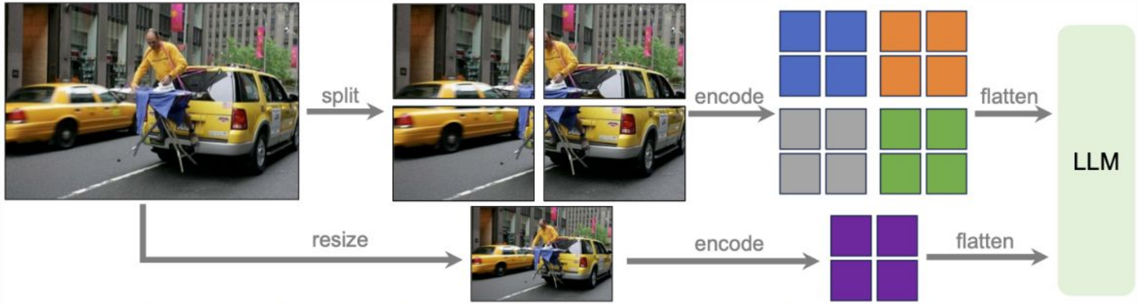
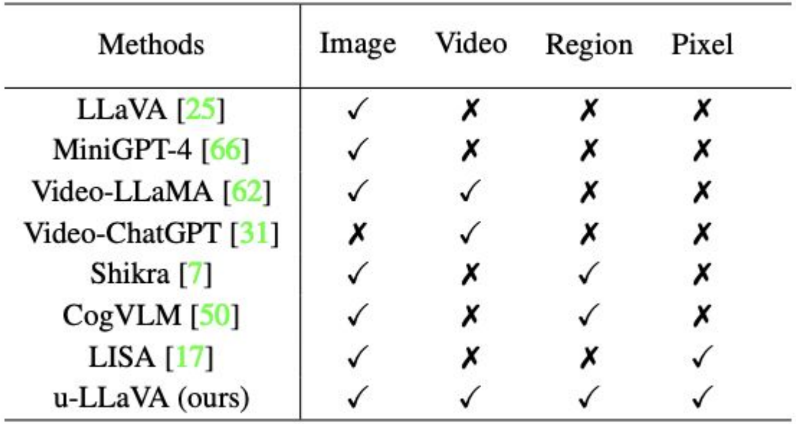


Illustration of dynamic high resolution scheme: a grid configuration of 2×2

24. u-LLaVA (arxiv 2023)

- Projector-based architecture
- Unifies multi-modal tasks
  - Integrates **pixel**, **regional**, and **global** features
  - By connecting (specialized) experts with (central) LLM



Methods	Image	Video	Region	Pixel
LLaVA [25]	✓	✗	✗	✗
MiniGPT-4 [66]	✓	✗	✗	✗
Video-LLaMA [62]	✓	✓	✗	✗
Video-ChatGPT [31]	✗	✓	✗	✗
Shikra [7]	✓	✗	✓	✗
CogVLM [50]	✓	✗	✓	✗
LISA [17]	✓	✗	✗	✓
u-LLaVA (ours)	✓	✓	✓	✓

24. u-LLaVA (arxiv 2023)

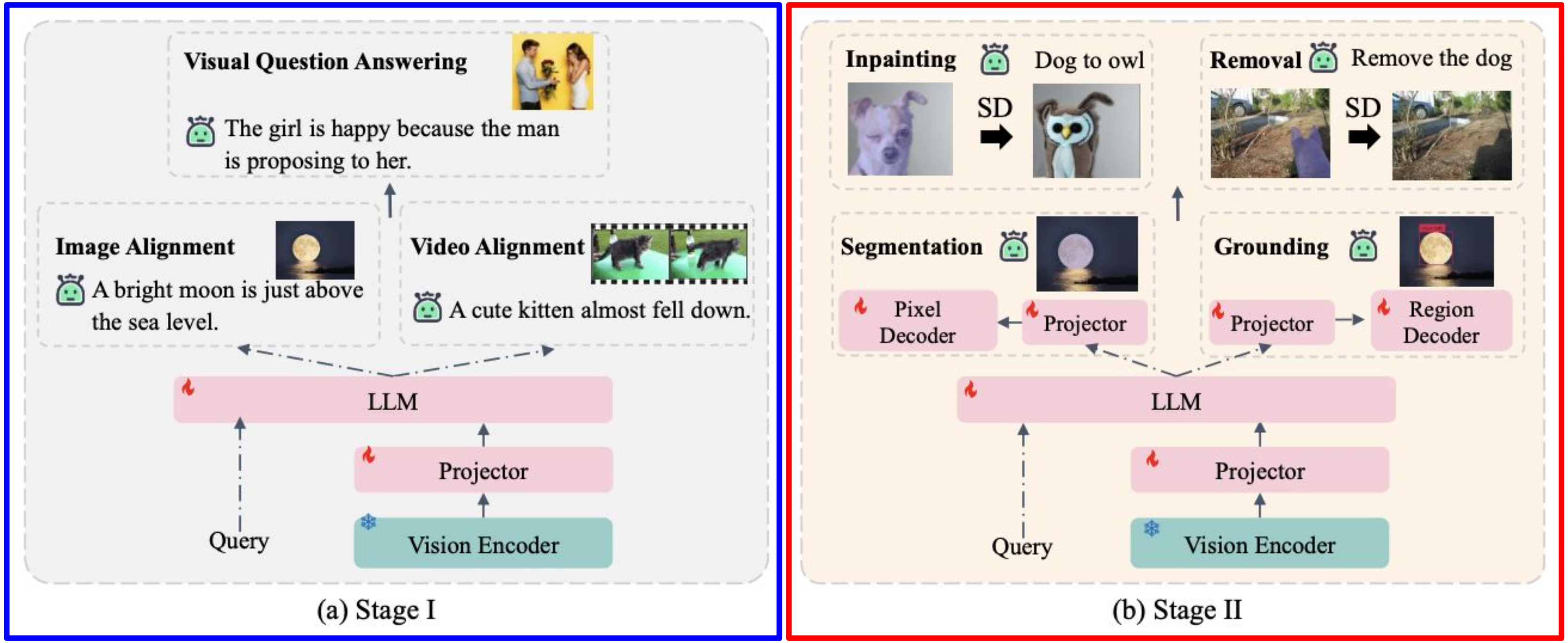


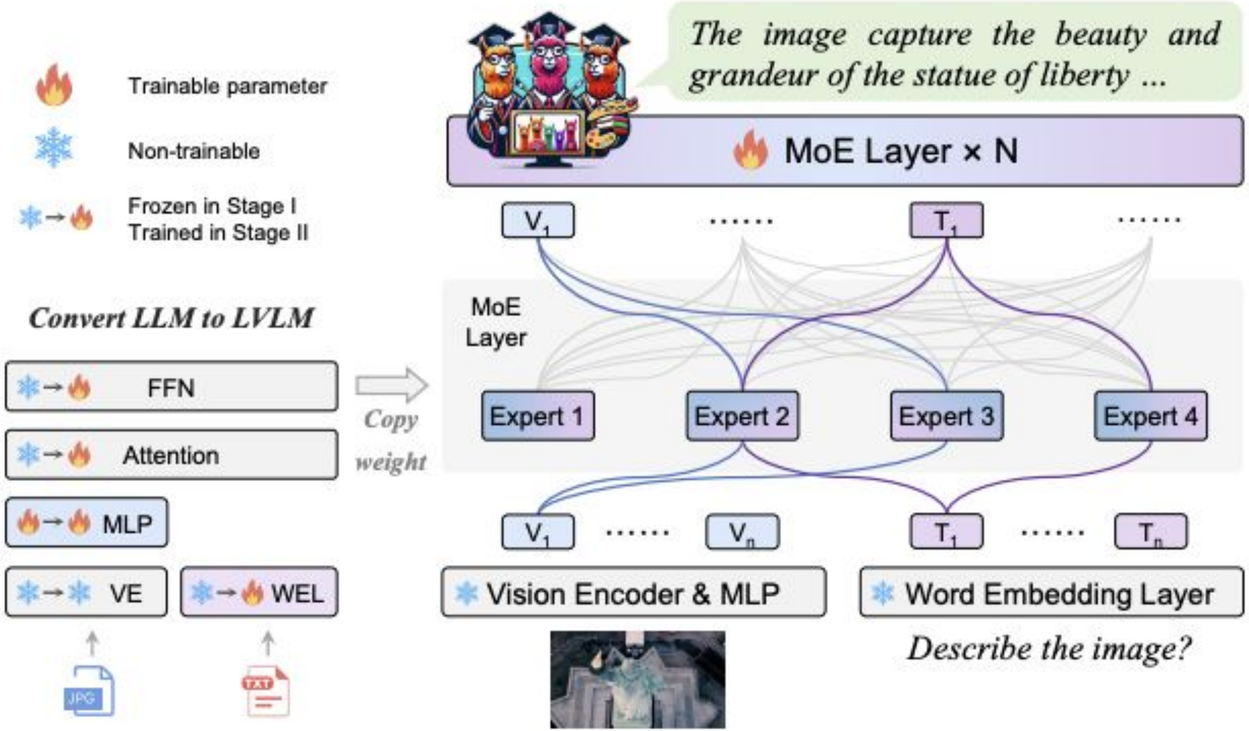
Figure 1: Overview of u-LLaVA. In stage I, image and spatio-temporal features are used to efficiently boost the general-purpose modality alignment. In Stage II, task-specific projectors and decoders are jointly trained for region and pixel-level understanding. Further, additional modules such as stable diffusion [39] can be easily patched for downstream tasks.

MoE-LLaVA: Mixture of Experts for Large Vision-Language Models

25. MoE-LLaVA (arxiv 2024)

- **MoE** + VLM

  (Overview)



(a) Stage I and Stage II

(b) Stage III

25. MoE-LLaVA (arxiv 2024)

- **MoE** + VLM

  (Detail)

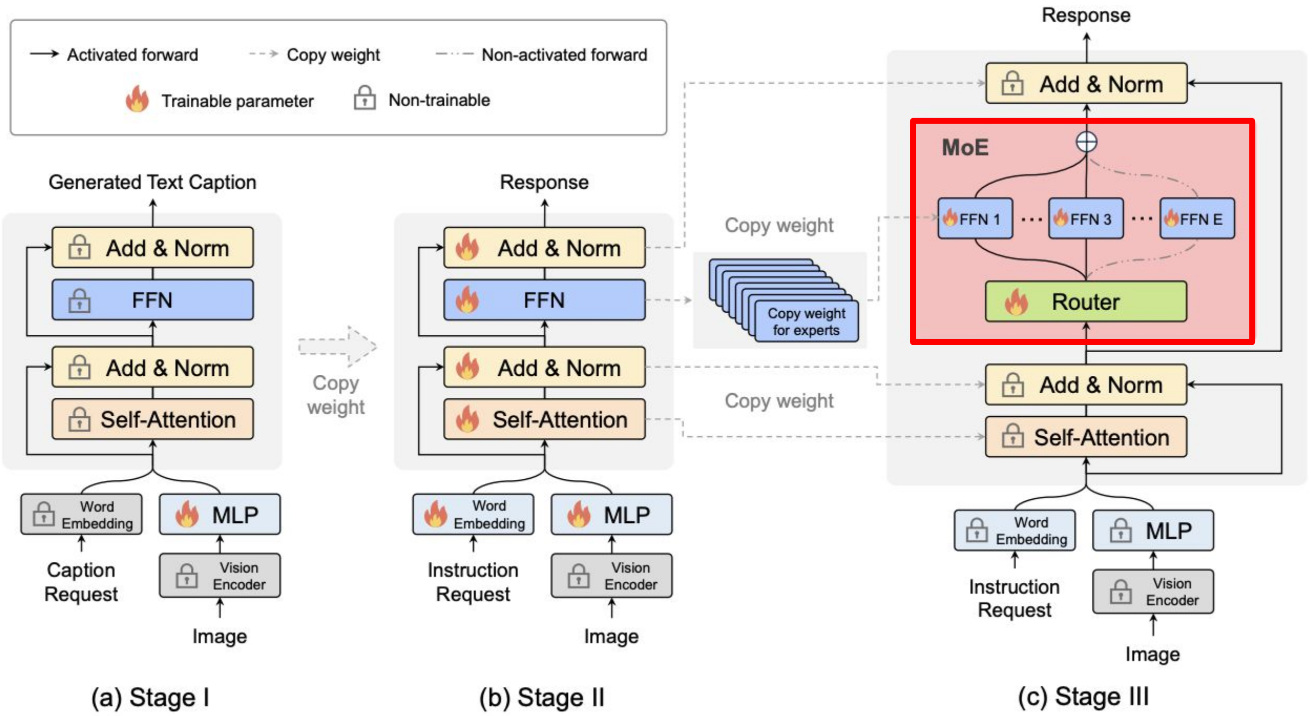
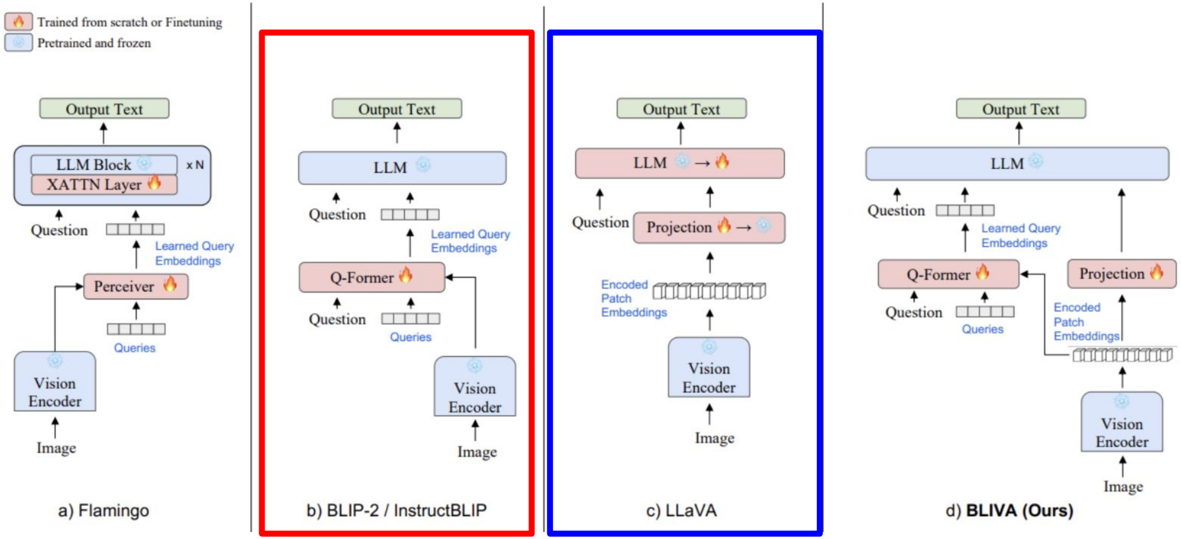


Figure 3. **Training framework and strategy.** MoE-LLaVA adopts a three-stage training strategy. (a) We solely train the MLP to adapt the LLM to visual inputs. (b) Training the LLM backend empowers multi-modal understanding capability and MoE layers are not involved. (c) In this stage, we replicate the weights of the FFN to initialize each expert.

26. BLIVA (AAAI 2024)

- Augments “**InstructBLIP**” with “**Visual Assistant**”
  - InstructBLIP: **Learned query embeddings** (feat. Q-Former)
  - Visual Assistant: **Encoded patch embeddings**



a) Flamingo

b) BLIP-2 / InstructBLIP

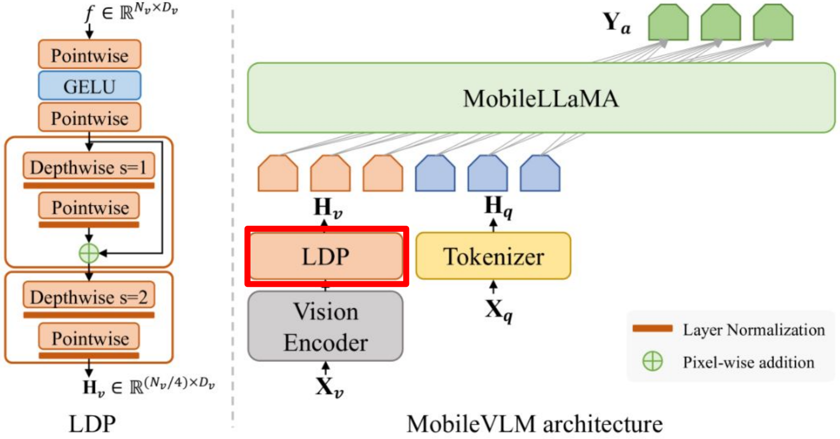
c) LLaVA

d) **BLIVA (Ours)**

MobileVLM: A Fast, Strong and Open Vision Language Assistant for Mobile Devices

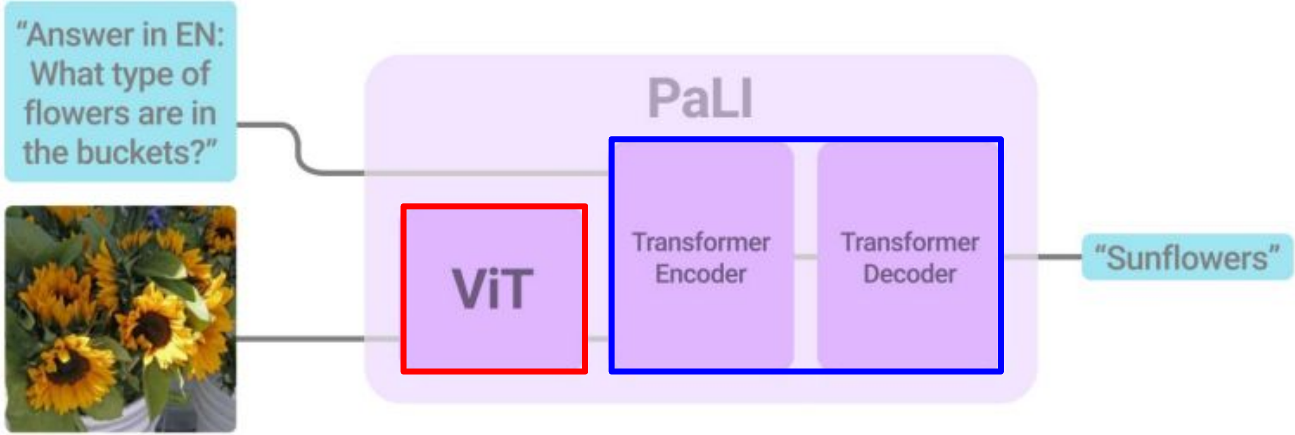
27. MobileVLM (arxiv 2023)

- Mobile-optimized VLM
- Architecture
  - (1) Vision: CLIP ViT-L/14
  - (2) LLM: MobileLLaMA
  - Combine (1)->(2) with **LDP**
- **lightweight downsample projector**



28. PaLI (ICLR 2023)

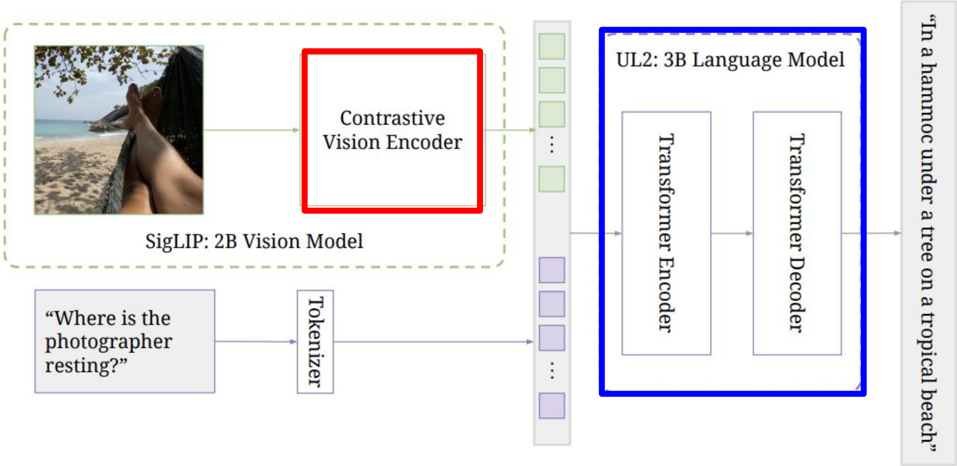
- Architecture
  - (Image) **ViT**
  - (LLM) **mT5**



PaLI-3 Vision Language Models: Smaller, Faster, Stronger

29. PaLI-3 (arxiv 2023)

- Architecture
    - (Image) **SigLIP-2B**
    - (LLM) **UL2-3B**
- Training:
    - Stage 0) **Unimodal pretraining**
        - Pretrain image encoder
        - Task: (image-text) CL
    - Stage 1) **Multimodal pretrianing**
        - Pretrain LLM
        - Task: Mixture of tasks
    - Stage 2) **Resolution increase**


Contrastive Vision Encoder
SigLIP: 2B Vision Model
"Where is the photographer resting?"
Tokenizer
UL2: 3B Language Model
Transformer Encoder
Transformer Decoder
"In a hammoc under a tropical beach"

30. PaliGemma (arxiv 2024)

- Architecture
  - (Image) **SigLIP**
  - (LLM) **Gemma**

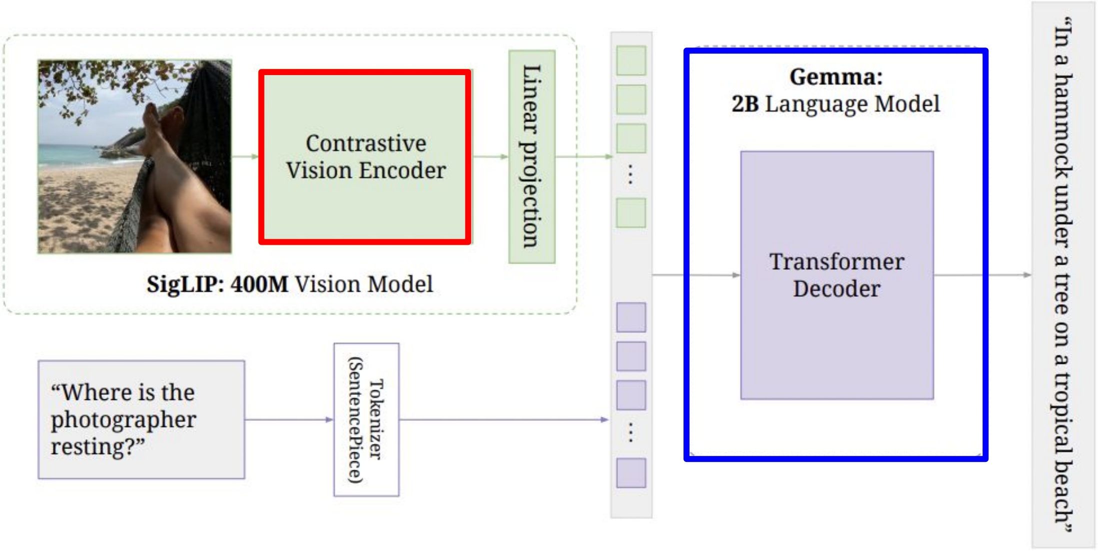

Contrastive Vision Encoder
Linear projection
SigLIP: 400M Vision Model
"Where is the photographer resting?"
Tokenizer (SentencePiece)
Gemma:
2B Language Model
Transformer Decoder
"In a hammock under a tree on a tropical beach"

Figure 1 | PaliGemma's architecture: a SigLIP image encoder feeds into a Gemma decoder LM.

PaliGemma 2: A Family of Versatile VLMs for Transfer

31. PaliGemma 2 (arxiv 2024)

- PaliGemma 개선
  - Gemma2-**2B** => Gemma2-**27B**
  - 3개의 서로 다른 resolution

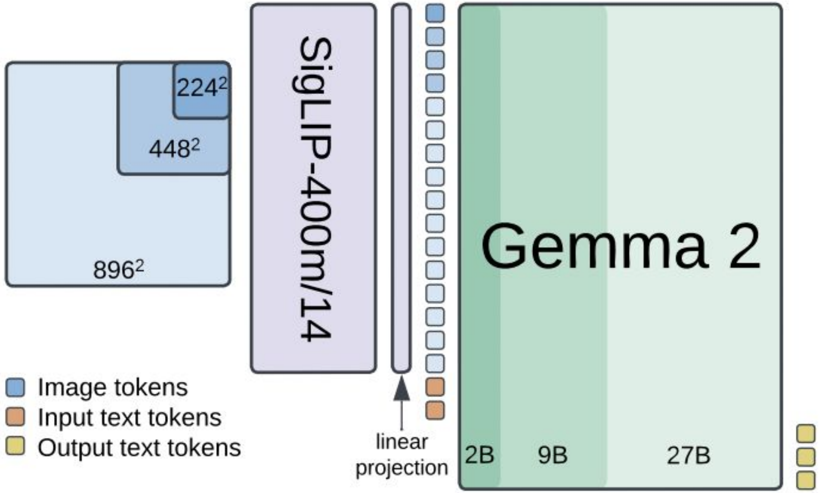


Figure 1 | PaliGemma 2 processes a 224px^2/448px^2/896px^2 image with a SigLIP-400m encoder with patch size 14px^2, yielding 256/1024/4096 tokens. After a linear projection, the image tokens are concatenated with the input text tokens and Gemma 2 autoregressively completes this prefix with an answer.

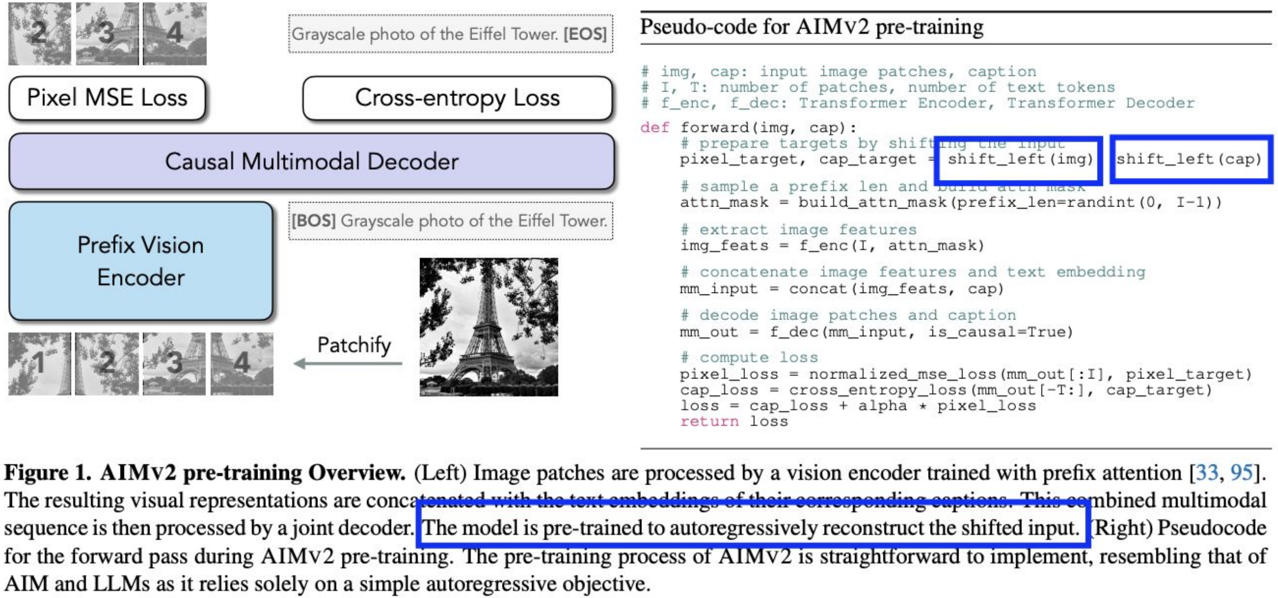
Multimodal autoregressive pre-training of large vision encoders

32. AIMv2 (arxiv 2024)

- A family of generalist vision encoders
- Vision Encoder + Multimodal decoder

$$L_{\text{img}} = \frac{1}{I} \sum_{i=1}^{I} \|\hat{x}_i(x_{<i}; \theta) - x_i\|_2^2,$$

$$L_{\text{text}} = -\frac{1}{T} \sum_{t=I+1}^{I+T} \log P(x_t | x_{<t}; \theta).$$


2
3
4
Grayscale photo of the Eiffel Tower. [EOS]
Pixel MSE Loss
Cross-entropy Loss
Causal Multimodal Decoder
Prefix Vision Encoder
[BOS] Grayscale photo of the Eiffel Tower.
1
2
3
4
Patchify

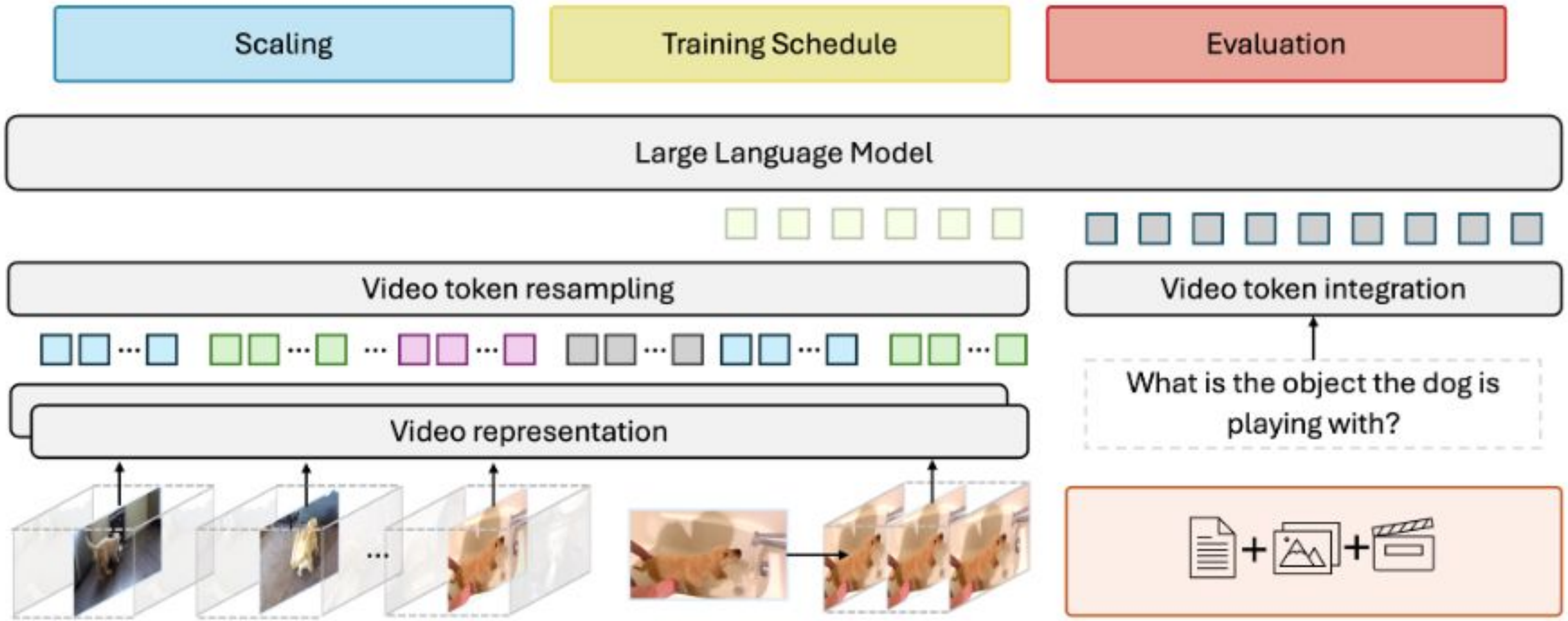
Pseudo-code for AIMv2 pre-training

```
# img, cap: input image patches, caption
# I, T: number of patches, number of text tokens
# f_enc, f_dec: Transformer Encoder, Transformer Decoder
def forward(img, cap):
    # prepare targets by shifting the input
    pixel_target, cap_target = shift_left(img), shift_left(cap)
    # sample a prefix len and build attn mask
    attn_mask = build_attn_mask(prefix_len=randint(0, I-1))
    # extract image features
    img_feats = f_enc(I, attn_mask)
    # concatenate image features and text embedding
    mm_input = concat(img_feats, cap)
    # decode image patches and caption
    mm_out = f_dec(mm_input, is_causal=True)
    # compute loss
    pixel_loss = normalized_mse_loss(mm_out[:I], pixel_target)
    cap_loss = cross_entropy_loss(mm_out[-T:], cap_target)
    loss = cap_loss + alpha * pixel_loss
    return loss
```

Figure 1. AIMv2 pre-training Overview. (Left) Image patches are processed by a vision encoder trained with prefix attention [33, 95]. The resulting visual representations are concatenated with the text embeddings of their corresponding captions. This combined multimodal sequence is then processed by a joint decoder. The model is pre-trained to autoregressively reconstruct the shifted input. (Right) Pseudocode for the forward pass during AIMv2 pre-training. The pre-training process of AIMv2 is straightforward to implement, resembling that of AIM and LLMs as it relies solely on a simple autoregressive objective.

33. Apollo (arxiv 2024)

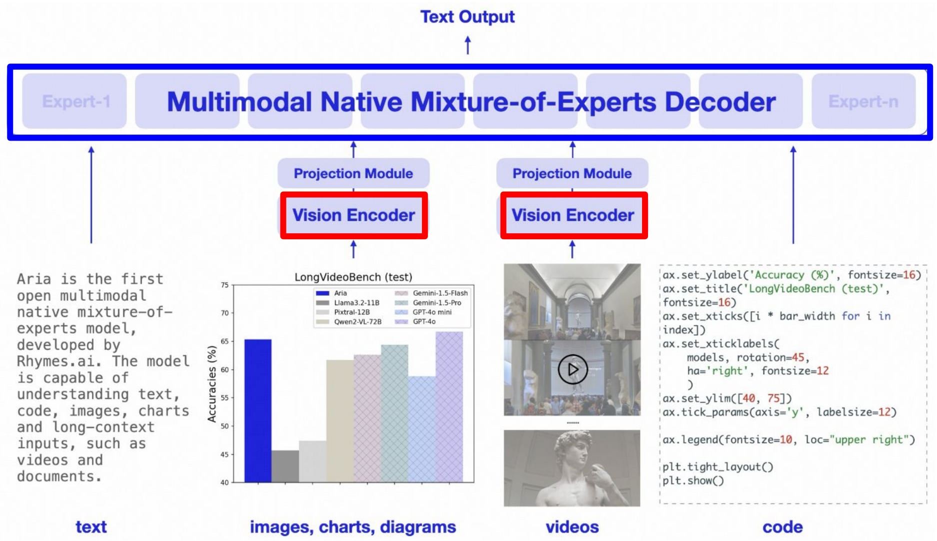
- SOTA LMMs for **video understanding**



Aria: An Open Multimodal Native Mixture-of-Experts Model

34. ARIA (arxiv 2024)

- **Multimodal MoE**
  - Text, Code, Images, Video …
- Architecture
  - (Fine-grained) **MoE decoder**
  - (Lightweight) **Visual encoder**



Text Output

Expert-1 | Multimodal Native Mixture-of-Experts Decoder | Expert-n

Projection Module

Vision Encoder

Projection Module

Vision Encoder

Aria is the first open multimodal native mixture-of-experts model, developed by Rhymes.ai. The model is capable of understanding text, code, images, charts and long-context inputs, such as videos and documents.

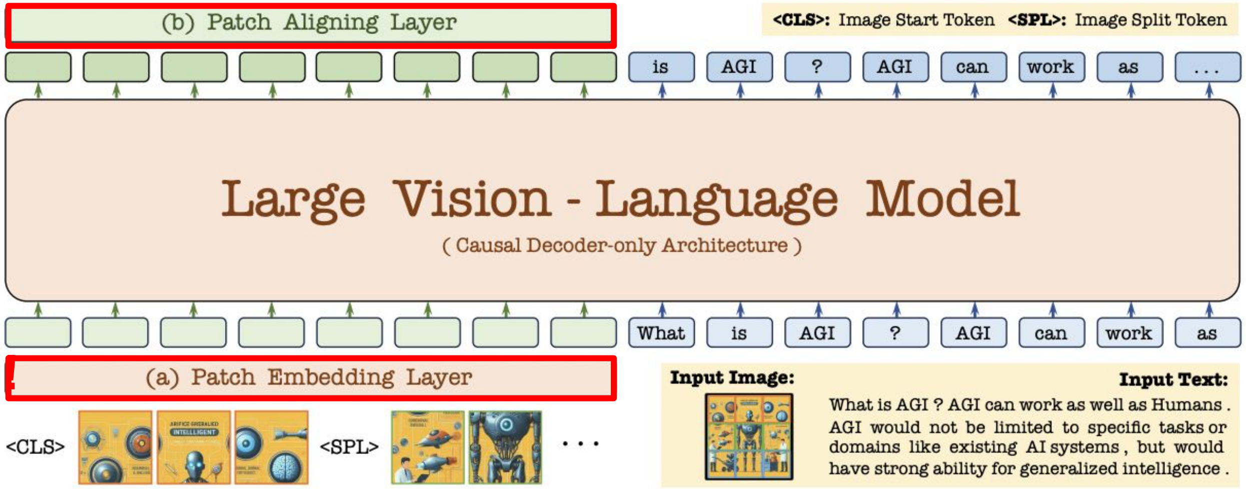
```
ax.set_ylabel('Accuracy (%)', fontsize=16)
ax.set_title('LongVideoBench (test)',
fontsize=16)
ax.set_xticks([i * bar_width for i in
index])
ax.set_xticklabels(
    models, rotation=45,
    ha='right', fontsize=12
    )
ax.set_ylim([40, 75])
ax.tick_params(axis='y', labelsize=12)

ax.legend(fontsize=10, loc="upper right")

plt.tight_layout()
plt.show()
```

35. EVE (NeurIPS 2024)

- Encoder-free VLM
  - **별도의 vision encoder 필요 없음**
- Image & Text를 동시에 “unified” “decoder-only” 아키텍처에 넣음


(b) Patch Aligning Layer
<CLS>: Image Start Token <SPL>: Image Split Token
is
AGI
?
AGI
can
work
as
. . .
Large Vision - Language Model
(Causal Decoder-only Architecture)
What
is
AGI
?
AGI
can
work
as
(a) Patch Embedding Layer
<CLS>
<SPL>
. . .
Input Image:
Input Text:

What is AGI ? AGI can work as well as Humans . AGI would not be limited to specific tasks or domains like existing AI systems , but would have strong ability for generalized intelligence .

이거면 끝!

35. EVE (NeurIPS 2024)

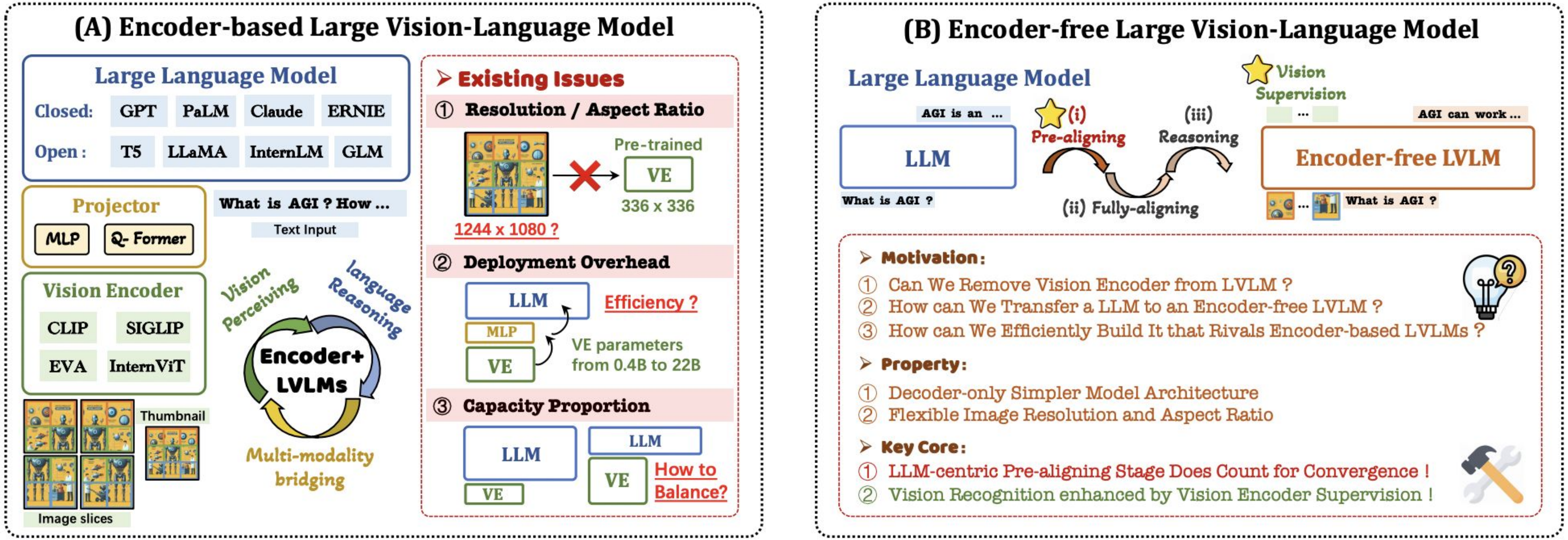

(A) Encoder-based Large Vision-Language Model
Large Language Model
Closed: GPT PaLM Claude ERNIE
Open : T5 LLaMA InternLM GLM
Projector
MLP Q- Former
What is AGI ? How ...
Text Input
Vision Encoder
CLIP SIGLIP
EVA InternViT
Image slices
Thumbnail
Vision Perceiving
language Reasoning
Encoder+ LVLMs
Multi-modality bridging
Existing Issues
① Resolution / Aspect Ratio
Pre-trained
VE
336 x 336
1244 x 1080 ?
② Deployment Overhead
LLM
MLP
VE
Efficiency ?
VE parameters from 0.4B to 22B
③ Capacity Proportion
LLM
VE
LLM
VE
How to Balance?
(B) Encoder-free Large Vision-Language Model
Large Language Model
AGI is an ...
LLM
What is AGI ?
(i) Pre-aligning
(ii) Fully-aligning
(iii) Reasoning
Vision Supervision
AGI can work ...
Encoder-free LVLM
What is AGI ?
Motivation:
① Can We Remove Vision Encoder from LVLM ?
② How can We Transfer a LLM to an Encoder-free LVLM ?
③ How can We Efficiently Build It that Rivals Encoder-based LVLMs ?
Property:
① Decoder-only Simpler Model Architecture
② Flexible Image Resolution and Aspect Ratio
Key Core:
① LLM-centric Pre-aligning Stage Does Count for Convergence !
② Vision Recognition enhanced by Vision Encoder Supervision !

Figure 1: Overview of large (A) encoder-based and (B) encoder-free vision-language models. **Encoder-based VLMs** contain vision encoders (VE) and large language models (LLM), with a projector as a vision-language bridge, while **encoder-free VLMs** exclude vision encoders and handle vision perception and linguistic instruction simultaneously with one unified architecture.

35. EVE (NeurIPS 2024)

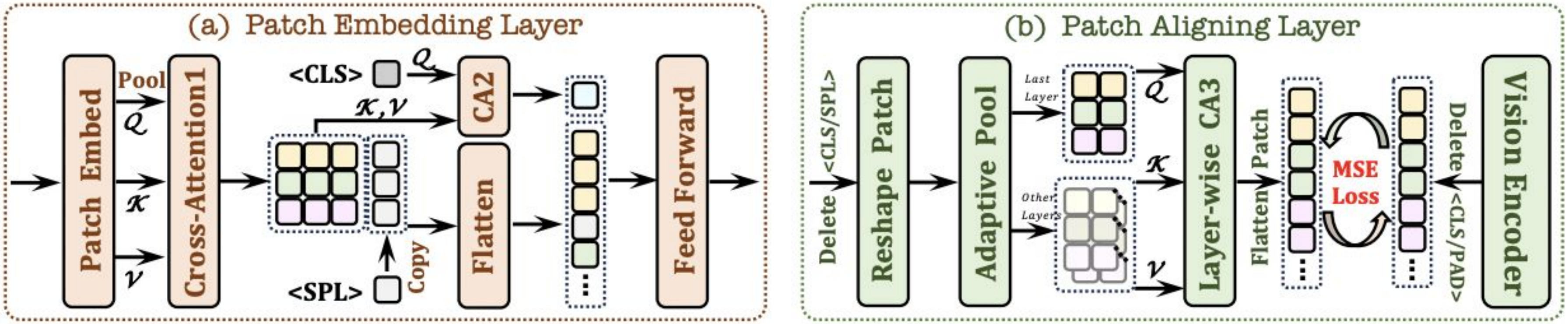


Figure 3: Overview of (a) patch embedding and (b) patch aligning layer. The former layer encodes images into patch features and employs cross-attention (CA1) within a limited receptive field to enhance representations. Meanwhile, a special <CLS> token provides a holistic view of each patch feature in the subsequent backbone. The latter layer removes all meaningless tokens and adjusts the size of patch features to align with the semantics of the same positional features in the vision encoder.

35. EVE (NeurIPS 2024)

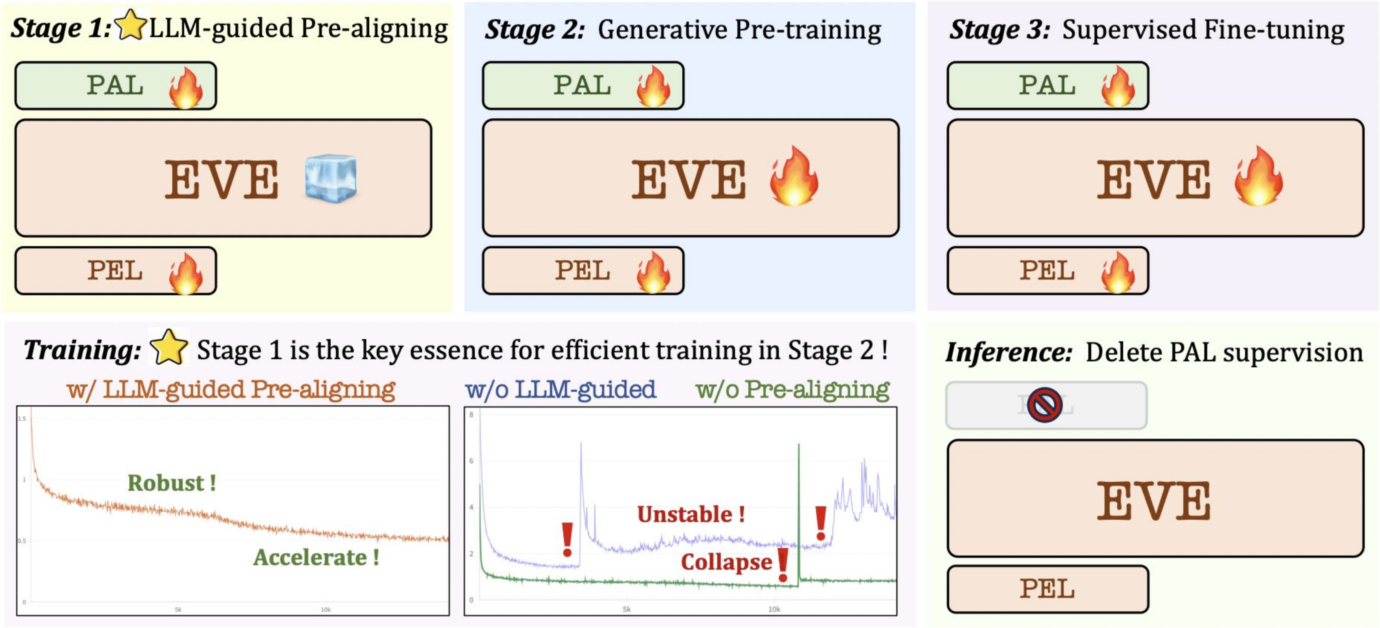

Stage 1: ⭐ LLM-guided Pre-aligning
PAL
EVE
PEL
Stage 2: Generative Pre-training
PAL
EVE
PEL
Stage 3: Supervised Fine-tuning
PAL
EVE
PEL
Training: ⭐ Stage 1 is the key essence for efficient training in Stage 2 !
w/ LLM-guided Pre-aligning
w/o LLM-guided
w/o Pre-aligning
Robust !
Accelerate !
Unstable !
Collapse !
1.5
1
0.5
0
5k
10k
8
6
4
2
0
Inference: Delete PAL supervision
PAL
EVE
PEL

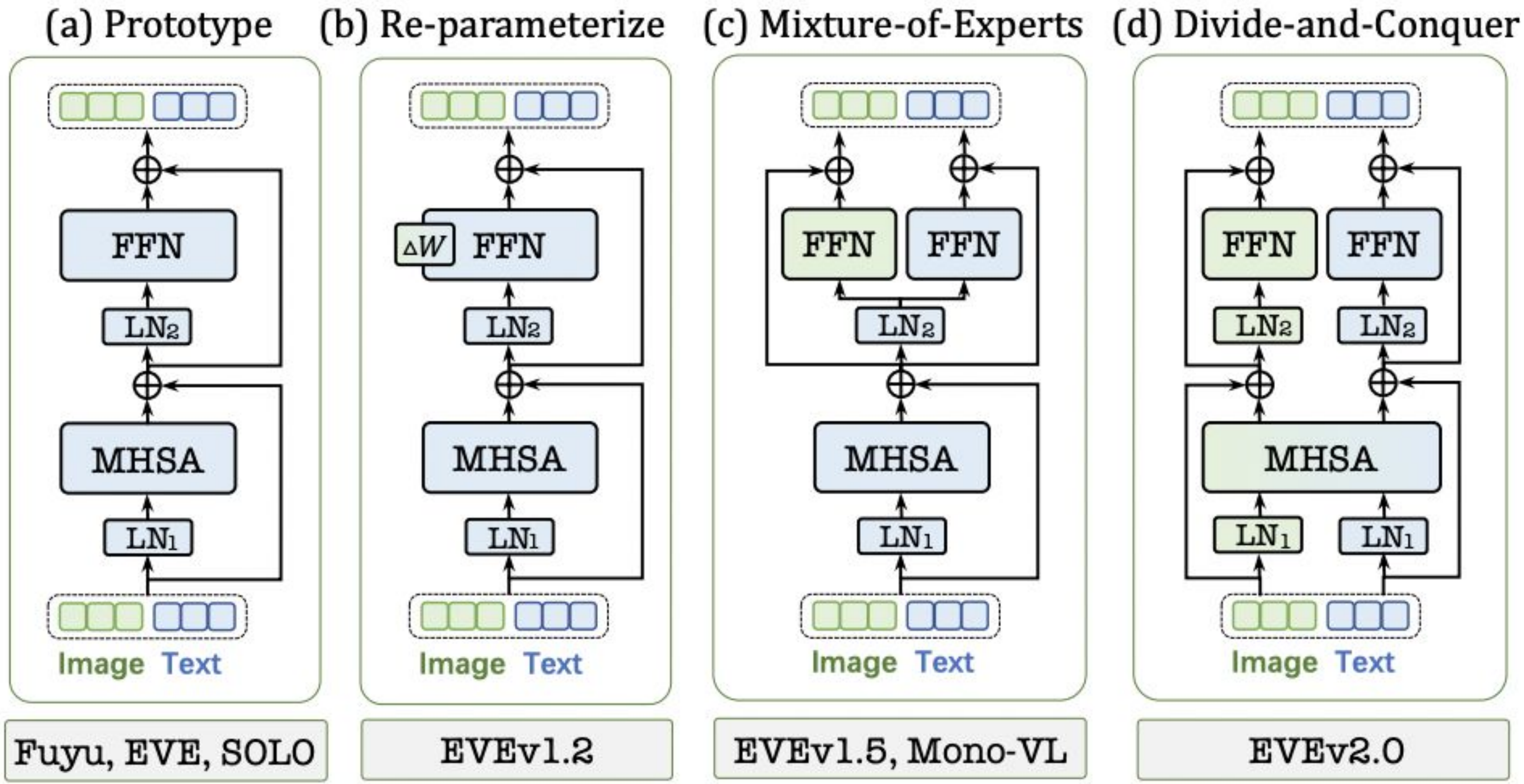
Figure 4: Overview of training procedure with three successive stages. We perform initial vision-language alignment guided by a frozen LLM in Stage 1, and then update the entire backbone for Stage 2 and Stage 3. We empirically find that Stage 1 is quite crucial to avoid collapse and accelerate convergence, thereby enhancing training efficiency. Notably, PAL is removed during inference.

36. EVEv2 (arxiv 2025)

- EVE와 마찬가지로 **Encoder-free VLM**
- **Divide-and-conquer** 아키텍처 도입
  - **“Modality-specific”** component

    (Text & Vision에 각기 다른 Att, LN, FFN)
- 효과
  - Maximize scaling efficiency
  - Reduce inter-modality interference
  - Superior data efficiency

36. EVEv2 (arxiv 2025)



36. EVEv2 (arxiv 2025)

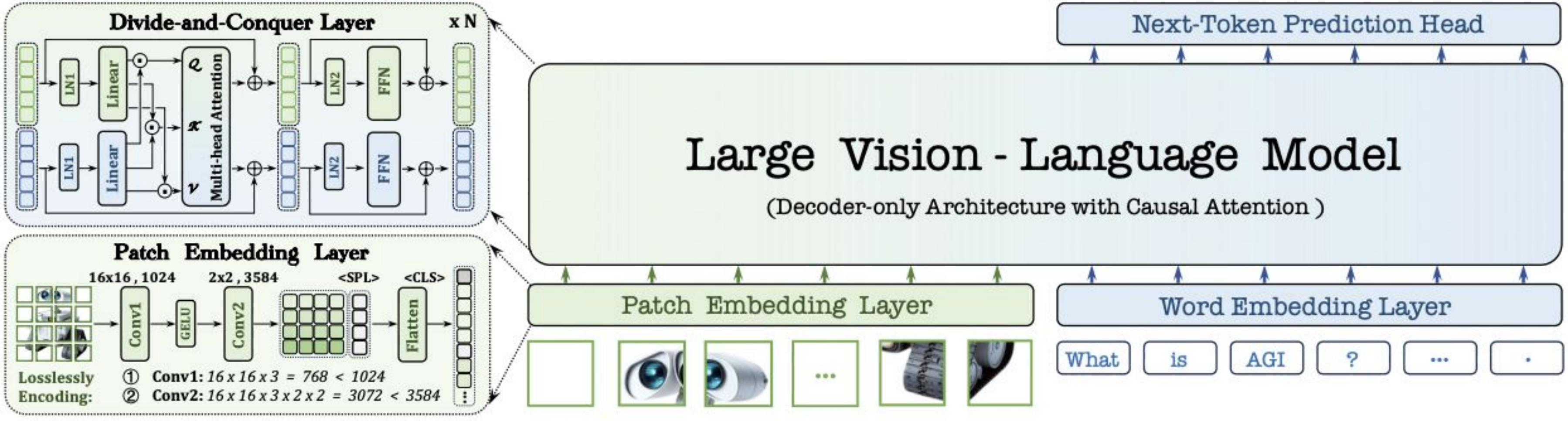


Figure 3. Overview of our proposed EVEv2.0 framework. We first adopt a patch embedding layer to encode images losslessly, and then concatenate visual and textual tokens into a unified decoder-only vision-language model. Here, it extends the standard autoregressive transformer by incorporating modality-specific weights for each multi-head self-attention layer, feed-forward layer, and layer normalization.

Janus: Decoupling Visual Encoding for Unified Multimodal Understanding and Generation

37. Janus (arxiv 2024)

- AR framework that unifies multimodal understanding and generation

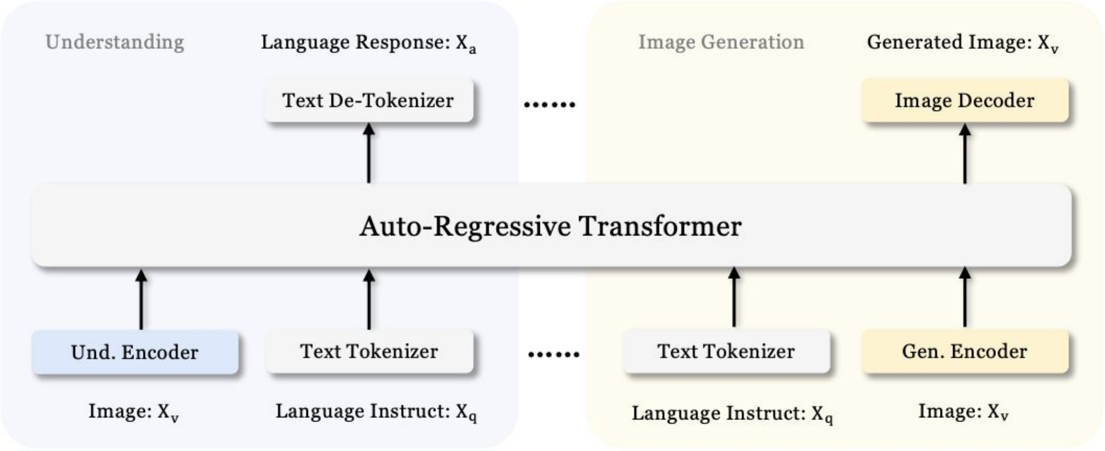


Figure 2 | **Architecture of our Janus.** Different from previous approaches [77, 85] that typically assume visual understanding and generation require the same visual encoder, our Janus decouples visual encoding for visual understanding and visual generation. "Und. Encoder" and "Gen. Encoder" are abbreviations for "Understanding Encoder" and "Generation Encoder", respectively. Best viewed in color.

37. Janus (arxiv 2024)

- AR framework that unifies multimodal understanding and generation

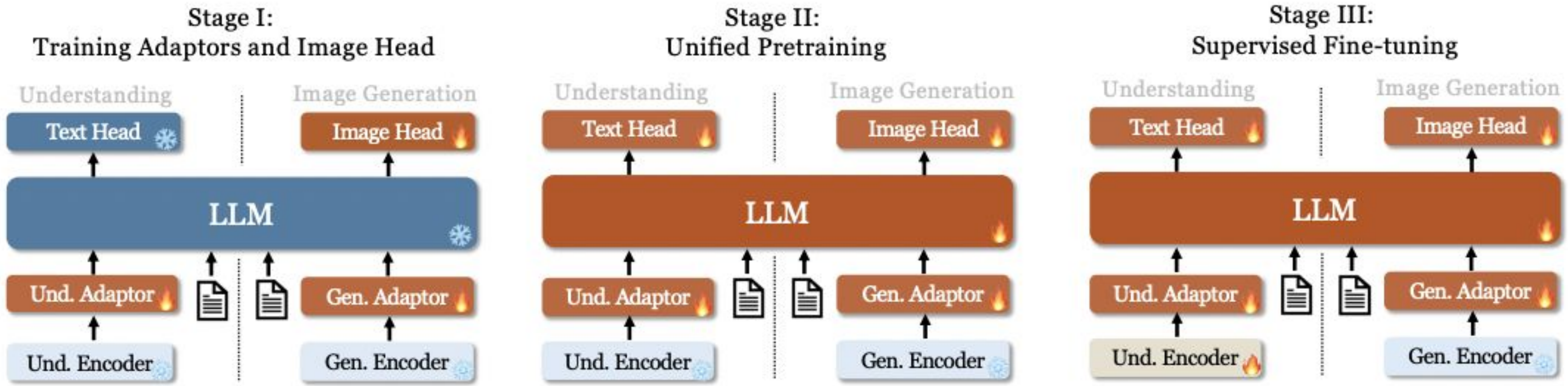
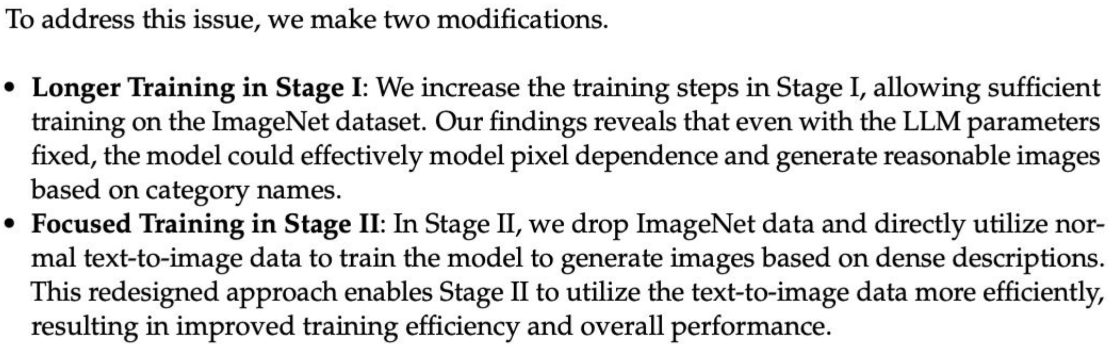


Figure 3 | **Our Janus adopts a three-stage training procedure.** We use flame symbols/snowflake symbols in the diagram to indicate the module updates/does not update its parameters.

38. Janus-Pro (arxiv 2025)

- **Janus를 발전 시킴**
  - (1) Training strategy (minor) 변화



To address this issue, we make two modifications.

- **Longer Training in Stage I**: We increase the training steps in Stage I, allowing sufficient training on the ImageNet dataset. Our findings reveals that even with the LLM parameters fixed, the model could effectively model pixel dependence and generate reasonable images based on category names.
- **Focused Training in Stage II**: In Stage II, we drop ImageNet data and directly utilize normal text-to-image data to train the model to generate images based on dense descriptions. This redesigned approach enables Stage II to utilize the text-to-image data more efficiently, resulting in improved training efficiency and overall performance.

  - (2) Training data 키우기
  - (3) Model scale-up

38. Janus-Pro (arxiv 2025)

- 아키텍처는 동일함

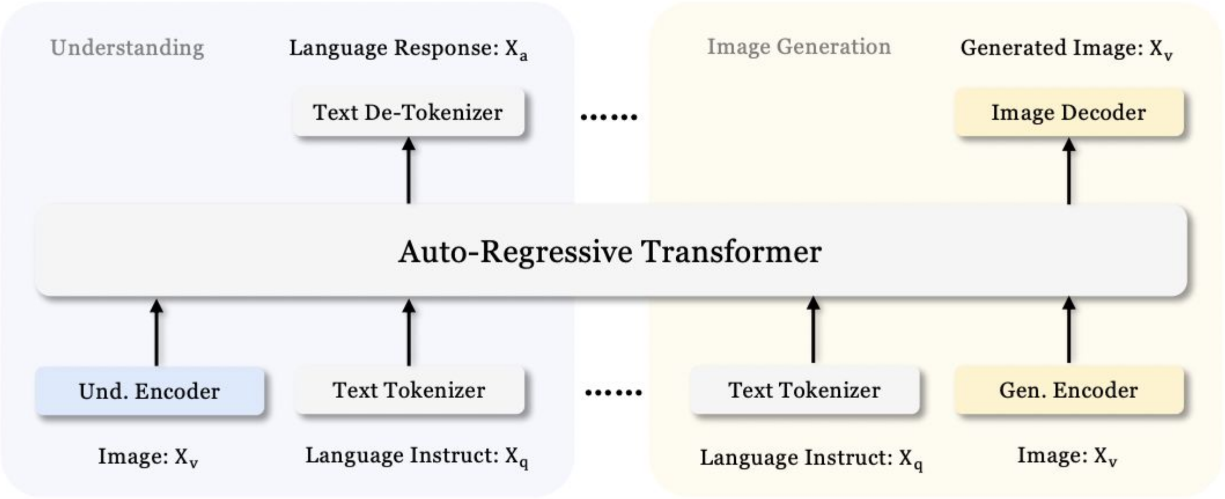
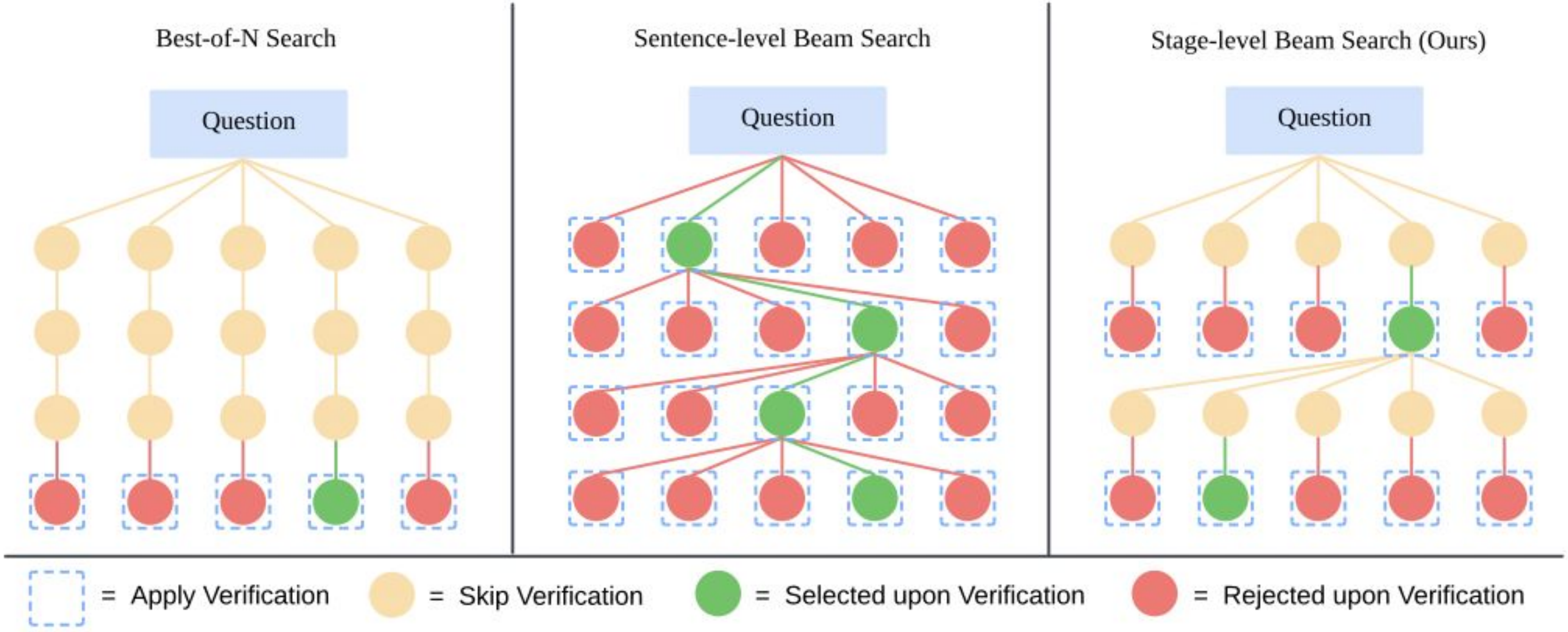


Figure 3 | **Architecture of our Janus-Pro.** We decouple visual encoding for multimodal understanding and visual generation. “Und. Encoder” and “Gen. Encoder” are abbreviations for “Understanding Encoder” and “Generation Encoder”, respectively. Best viewed on screen.

39. LLaVA-CoT (arxiv 2024)

- **Multi-stage reasoning**을 하는 VLM


Best-of-N Search
Sentence-level Beam Search
Stage-level Beam Search (Ours)
Question
Question
Question
= Apply Verification
= Skip Verification
= Selected upon Verification
= Rejected upon Verification

40. LLM2CLIP (arxiv 2024)

- Fine-tuning 방법론
    - **LLM & CLIP visual encoder를 연결**하기 위해!
- 2-stage approach
    - **(1) Caption Contrastive FT**
        - LLM을 FT한다
        - Causal ATT -> Bidirectional ATT (목적: 생성 X, 이해 O 이므로)
    - **(2) CLIP Vision Encoder FT**
        - Vision Encoder를 FT한다

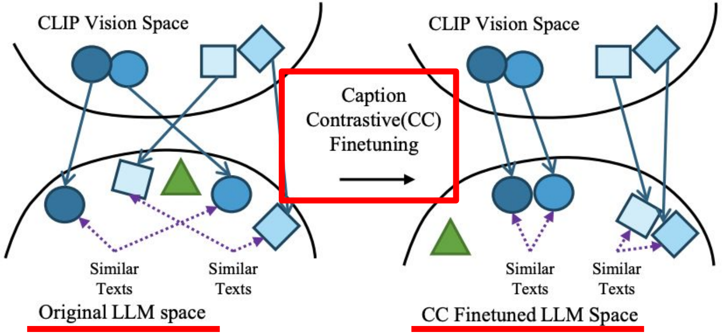
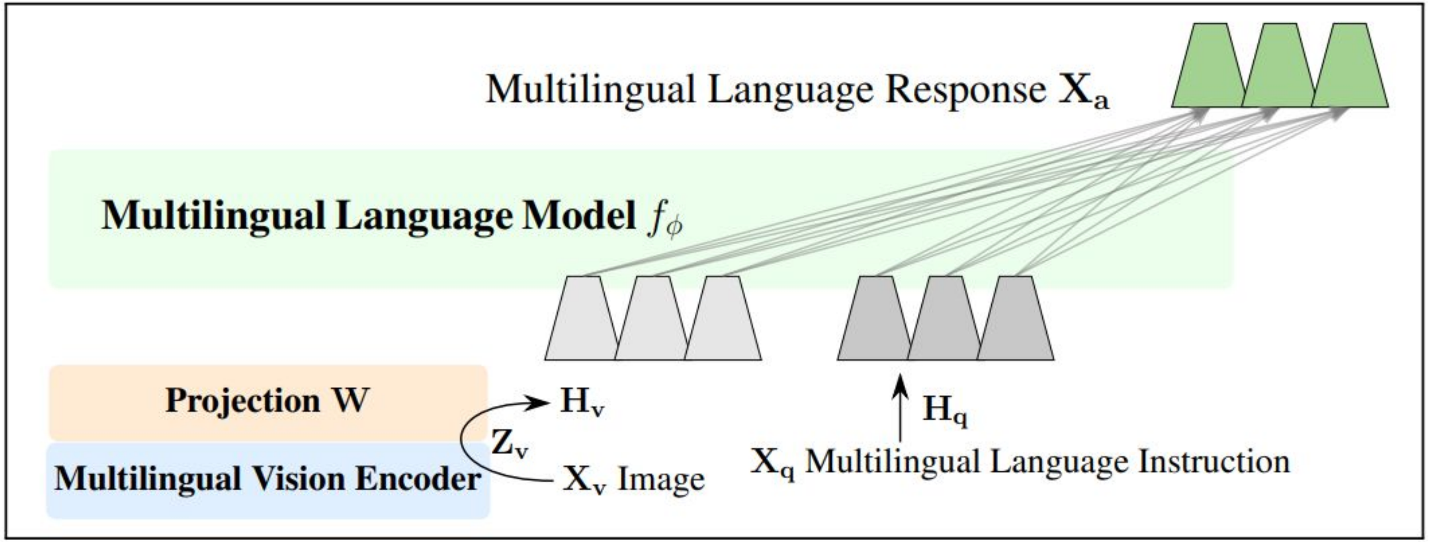


Figure 2: By adjusting the output space of the LLM, we enable the LLM to more effectively act as a private tutor for CLIP, allowing the fine-tuning process of *LLM2CLIP* to efficiently adjust the pretrained CLIP's cross-modal space.

41. Maya (arxiv 2024)

- Multimodal **Multilingual** VLM (mVLM)
- Framework: **LLaVA 1.5**

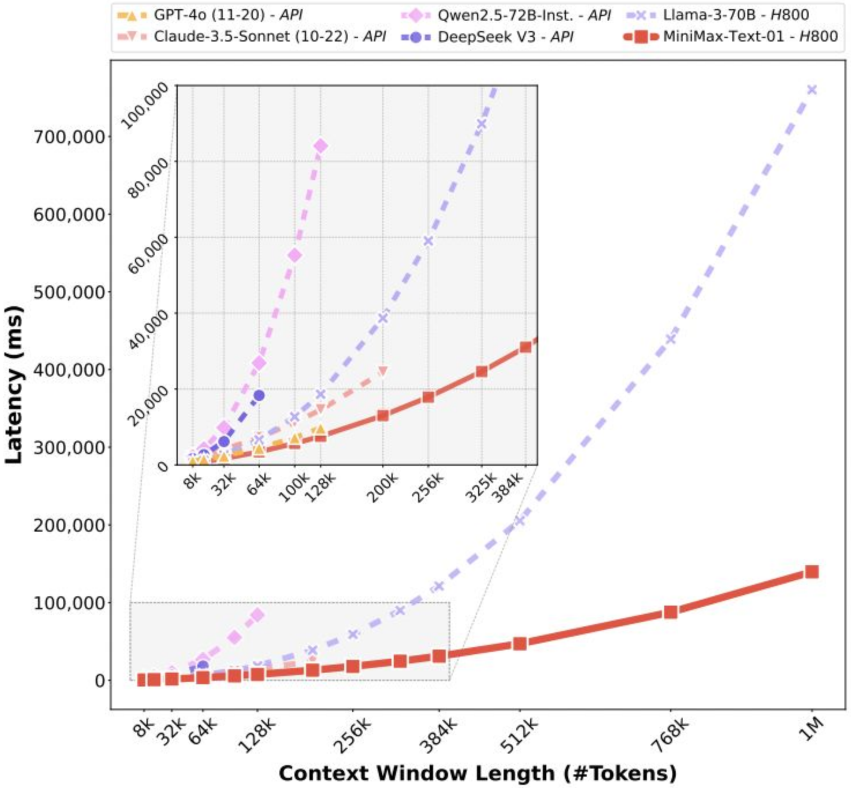

Multilingual Language Response X_a
Multilingual Language Model f_ϕ
Projection W
Multilingual Vision Encoder
Z_v
H_v
X_v Image
H_q
X_q Multilingual Language Instruction

MiniMax-01: Scaling Foundation Models with Lightning Attention

42. MiniMax-01 (arxiv 2025)

- MiniMax-01
    - MiniMax-Text-01
    - MiniMax-VL-01
- **Longer context window (~ 4M)**

  => How? Novel architecture!
    - **Lightning attention**
    - MoE



MiniMax-01: Scaling Foundation Models with Lightning Attention

42. MiniMax-01 (arxiv 2025)

- MiniMax-01
  - MiniMax-Text-01
  - MiniMax-VL-01
- **Longer context window (~ 4M)**

  => How? Novel architecture!
  - Lightning attention
  - **MoE**

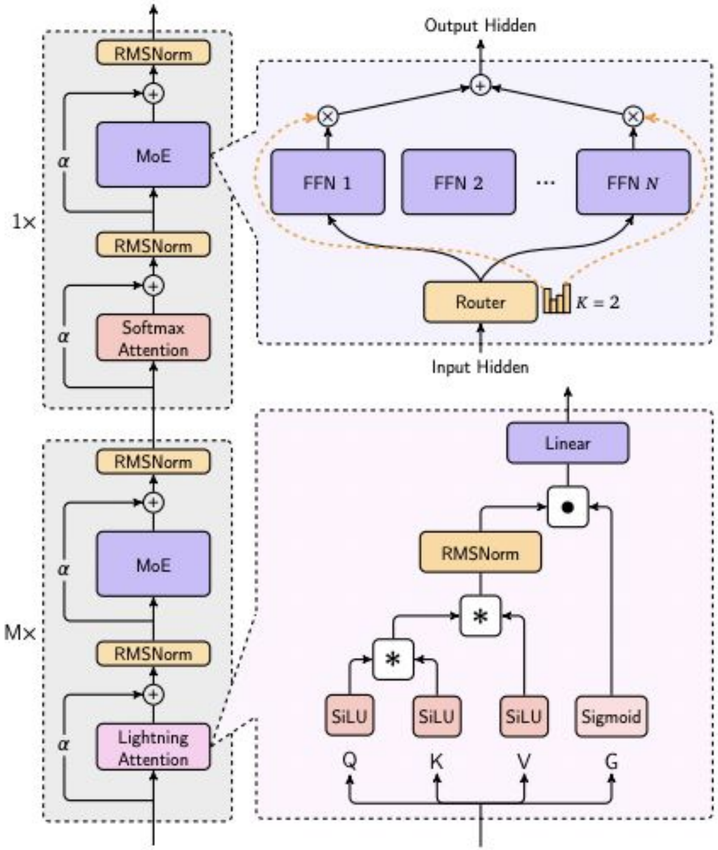
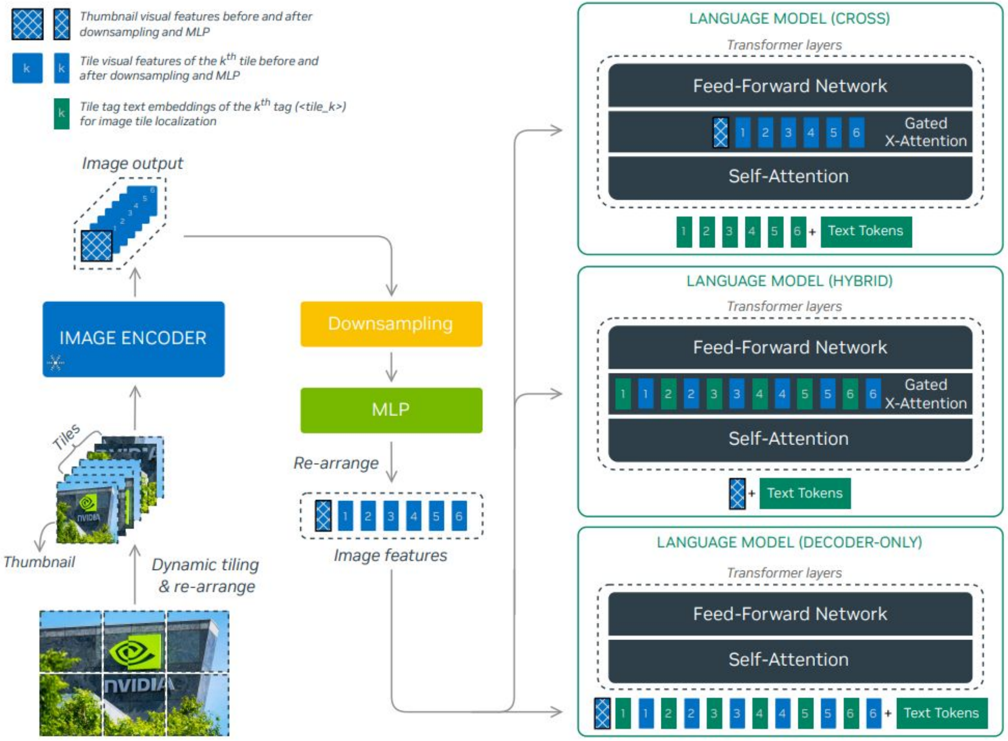


Figure 3 | **The architecture of MiniMax-Text-01.**

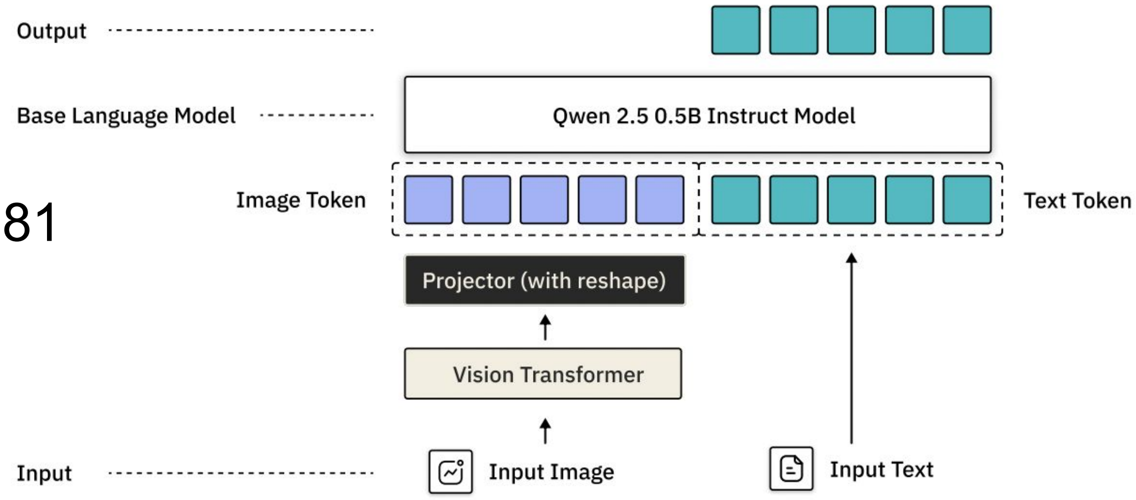
43. NVLM (arxiv 2024)

- NVLM = NVIDIA VLM
  - NVLM-**D** (Decoder-only)
  - NVLM-**X** (Cross-attention)
  - NVLM-**H** (Hybrid)



44. OmniVLM (arxiv 2024)

- VLM with sub-billion # params
- 목적: **On-device** inference
- **Token compression**
  - Convolution-based
  - # Visual token: 729 -> 81
- Architecture
  - Vision: SigLIP-400M
  - LLM: Qwen2.5-0.5B



45. Pixtral 12B (arxiv 2024)

- Mistral AI에서 공개한 모델
- Architecture
  - **(1) Vision encoder = Pixtral-ViT**
    - Trained with a new vision encoder from scratch
    - Support **variable** image sizes & aspect ratios
    - New **ROPE-2D** implementation
  - **(2) Multimodal decoder: NTP task**
    - Built on top of Mistral Nemo 12B
  - MLP: Connect (1) to (2)
- New benchmark: MM-MT-Bench

$$\text{RoPE-2D}\left(x^{(i,j)}, \Theta\right) = M_{\Theta}^{(i,j)} x^{(i,j)},$$

$$\text{where} \quad M_{\Theta}^{(i,j)} = \begin{pmatrix} \cos i\theta_1 & -\sin i\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin i\theta_1 & \cos i\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos j\theta_2 & -\sin j\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin j\theta_2 & \cos j\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos j\theta_{\frac{d}{2}} & -\sin j\theta_{\frac{d}{2}} \\ 0 & 0 & 0 & 0 & \cdots & \sin j\theta_{\frac{d}{2}} & \cos j\theta_{\frac{d}{2}} \end{pmatrix}$$

45. Pixtral 12B (arxiv 2024)

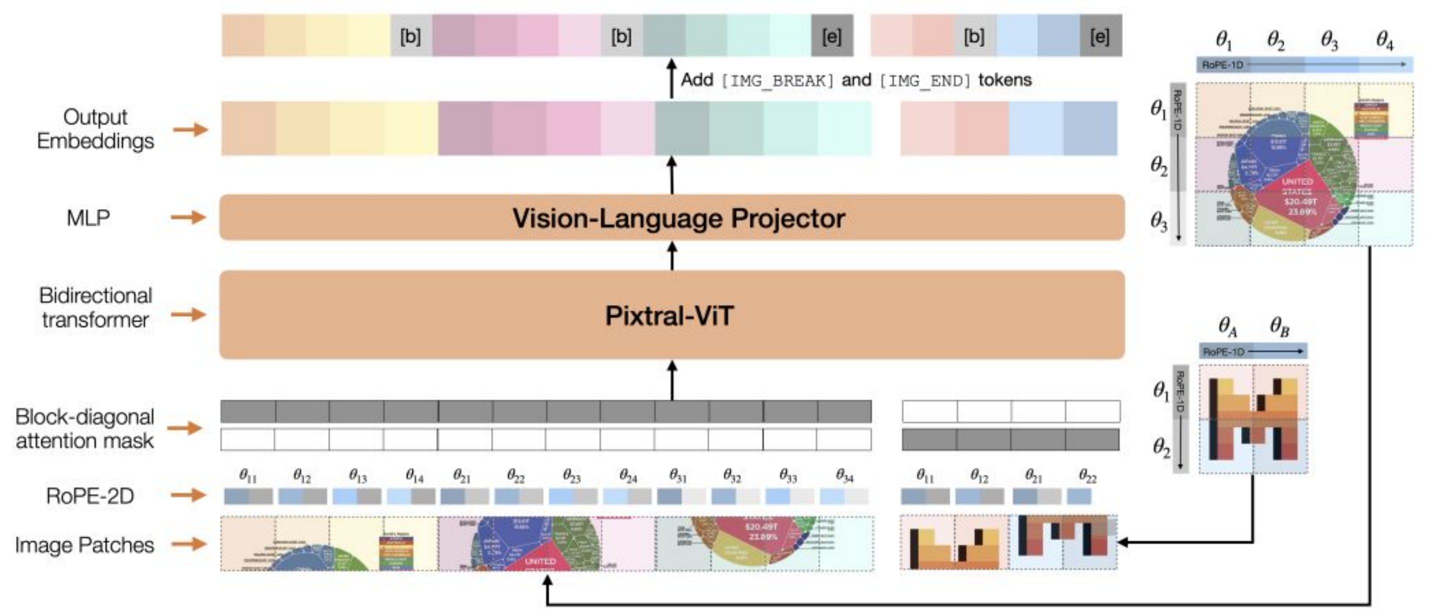

[b]
[b]
[e]
[b]
[e]
Add [IMG_BREAK] and [IMG_END] tokens
Output Embeddings
MLP
Vision-Language Projector
Bidirectional transformer
Pixtral-ViT
Block-diagonal attention mask
RoPE-2D
Image Patches
θ_{11} θ_{12} θ_{13} θ_{14} θ_{21} θ_{22} θ_{23} θ_{24} θ_{31} θ_{32} θ_{33} θ_{34}
θ_{11} θ_{12} θ_{21} θ_{22}
θ_1 θ_2 θ_3 θ_4
RoPE-1D
θ_1 θ_2 θ_3
RoPE-1D
UNITED STATES $20.49T 23.89%
θ_A θ_B
RoPE-1D
θ_1 θ_2
RoPE-1D

Figure 2: Pixtral Vision Encoder. Pixtral uses a new vision encoder, which is trained from scratch to natively support variable image sizes and aspect ratios. Block-diagonal attention masks enable sequence packing for batching, while ROPE-2D encodings facilitate variable image sizes. Note that the attention mask and position encodings are fed to the vision transformer as additional input, and utilized only in the self-attention layers.

45. Pixtral 12B (arxiv 2024)

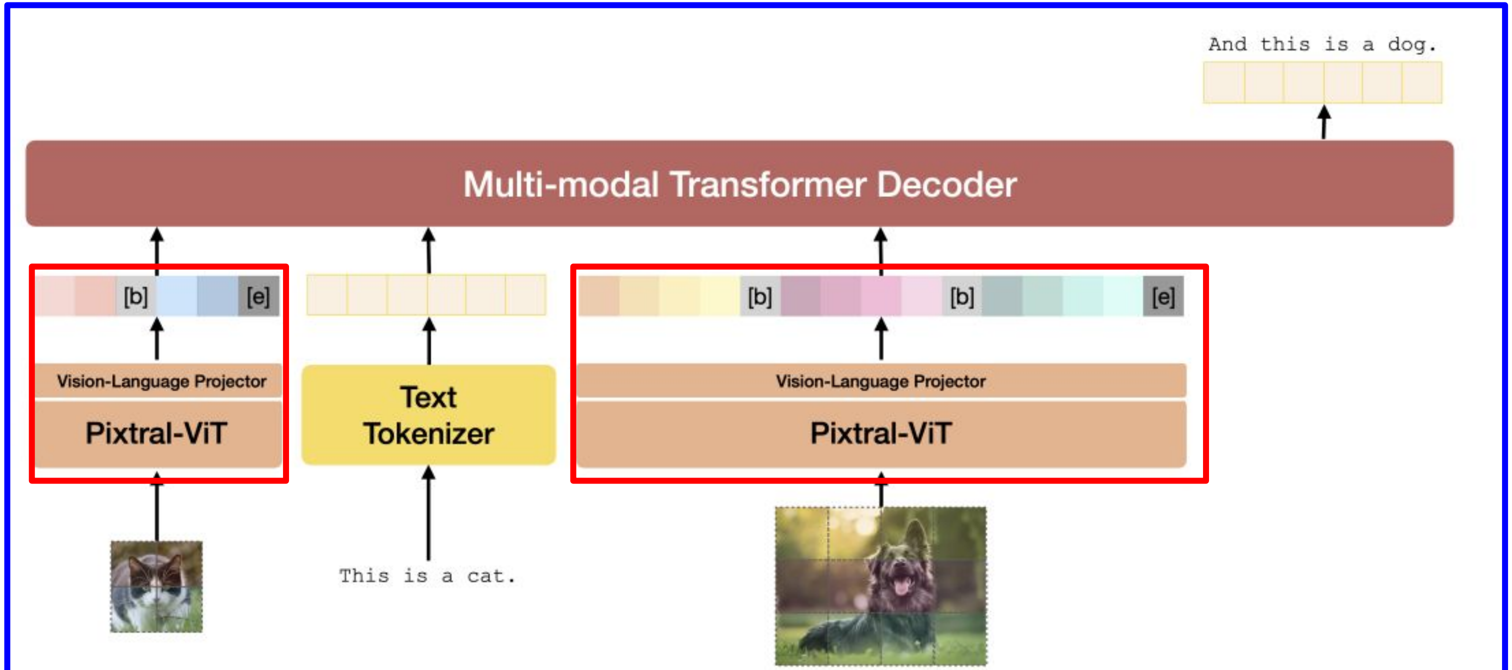


Figure 3: Complete Pixtral Architecture. Pixtral has two components: a *vision encoder*, which tokenizes images, and a *multimodal decoder*, which predicts the next text token given a sequence of text and images. Pixtral can take an arbitrary number of images as input, provided they fit within its 128K context window.

Sa2VA: Marrying SAM2 with LLaVA for Dense Grounded Understanding of Images and Videos

46. Sa2VA (arxiv 2025)

- “Unified” model for “**dense**” grounded understanding of “**image & video**”
- Treat **all inputs (text, images, videos)** as tokens in a **shared LLM**

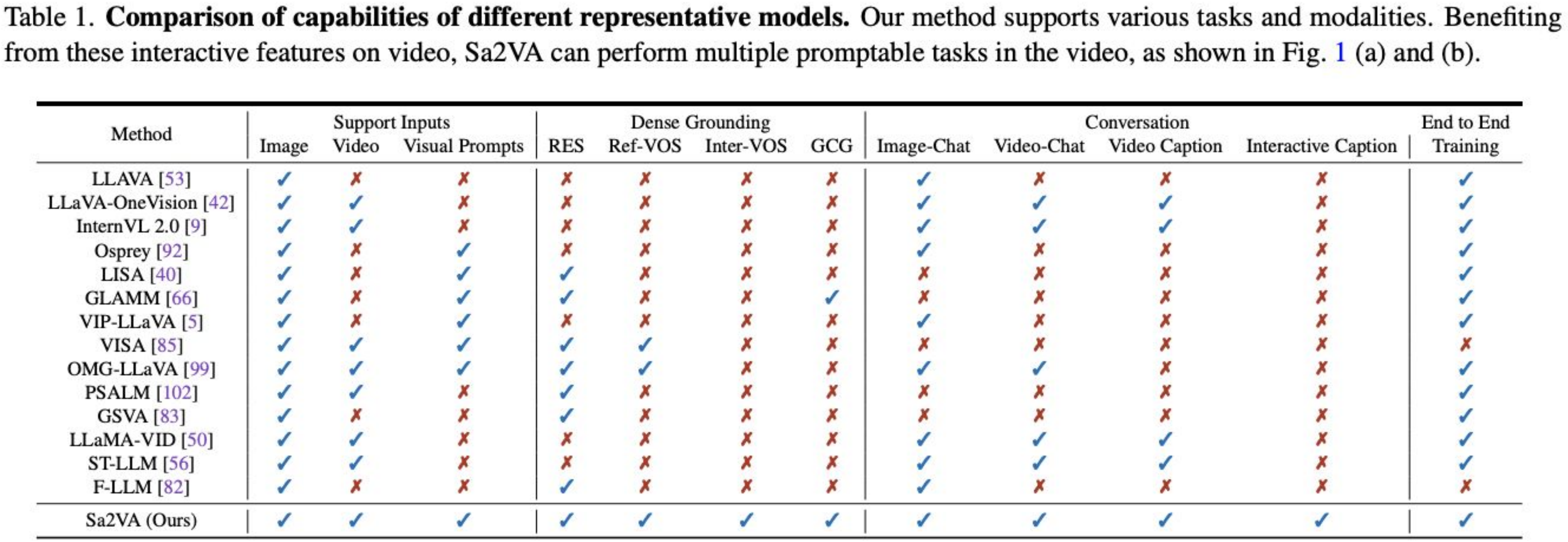


Table 1. **Comparison of capabilities of different representative models.** Our method supports various tasks and modalities. Benefiting from these interactive features on video, Sa2VA can perform multiple promptable tasks in the video, as shown in Fig. 1 (a) and (b).

Method	Support Inputs			Dense Grounding				Conversation				End to End
	Image	Video	Visual Prompts	RES	Ref-VOS	Inter-VOS	GCG	Image-Chat	Video-Chat	Video Caption	Interactive Caption	Training
LLAVA [53]	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
LLaVA-OneVision [42]	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓
InternVL 2.0 [9]	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓
Osprey [92]	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
LISA [40]	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓
GLAMM [66]	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓
VIP-LLaVA [5]	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
VISA [85]	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
OMG-LLaVA [99]	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓
PSALM [102]	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
GSVA [83]	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
LLaMA-VID [50]	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓
ST-LLM [56]	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓
F-LLM [82]	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Sa2VA (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

46. Sa2VA (arxiv 2025)

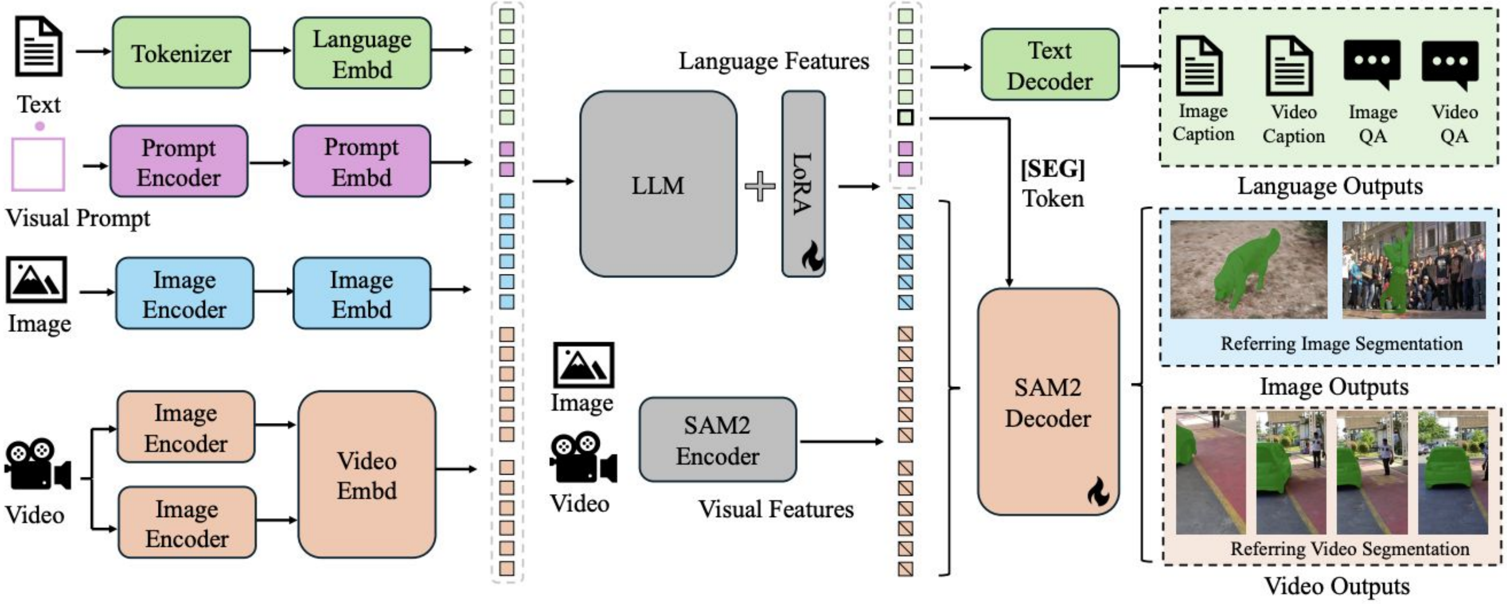
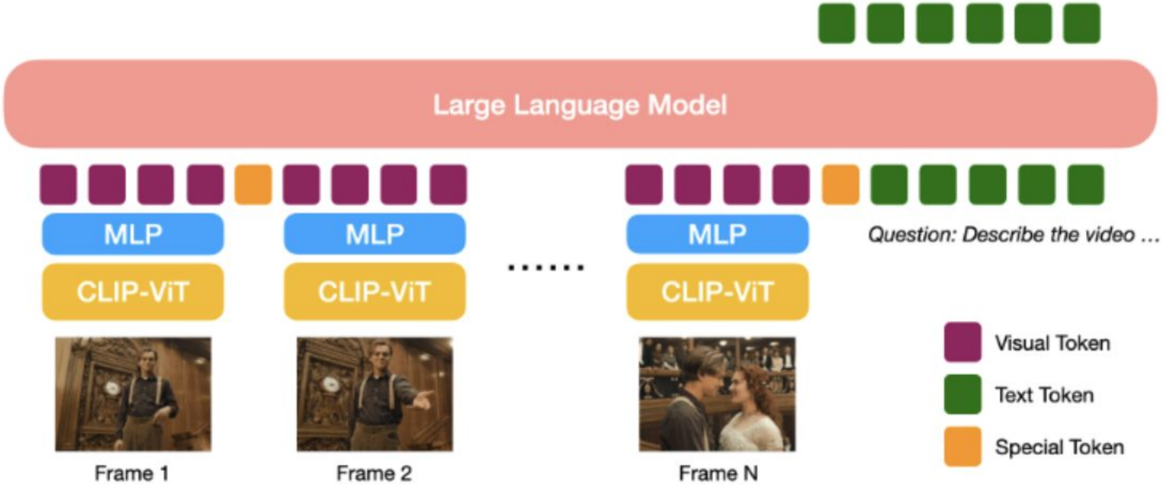


Figure 2. **Our proposed Sa2VA model.** The model first encodes the input texts, visual prompts, images, and videos into token embeddings. These tokens are then processed through a large language model (LLM). The output text tokens are used to generate the "[SEG]" token and associated language outputs. The SAM-2 decoder receives the image and video features from the SAM-2 encoder, along with the "[SEG]" token, to generate corresponding image and video masks.

Tarsier2: Advancing Large Vision-Language Models from Detailed Video Description to Comprehensive Video Understanding

47. Tarsier2 (arxiv 2025)

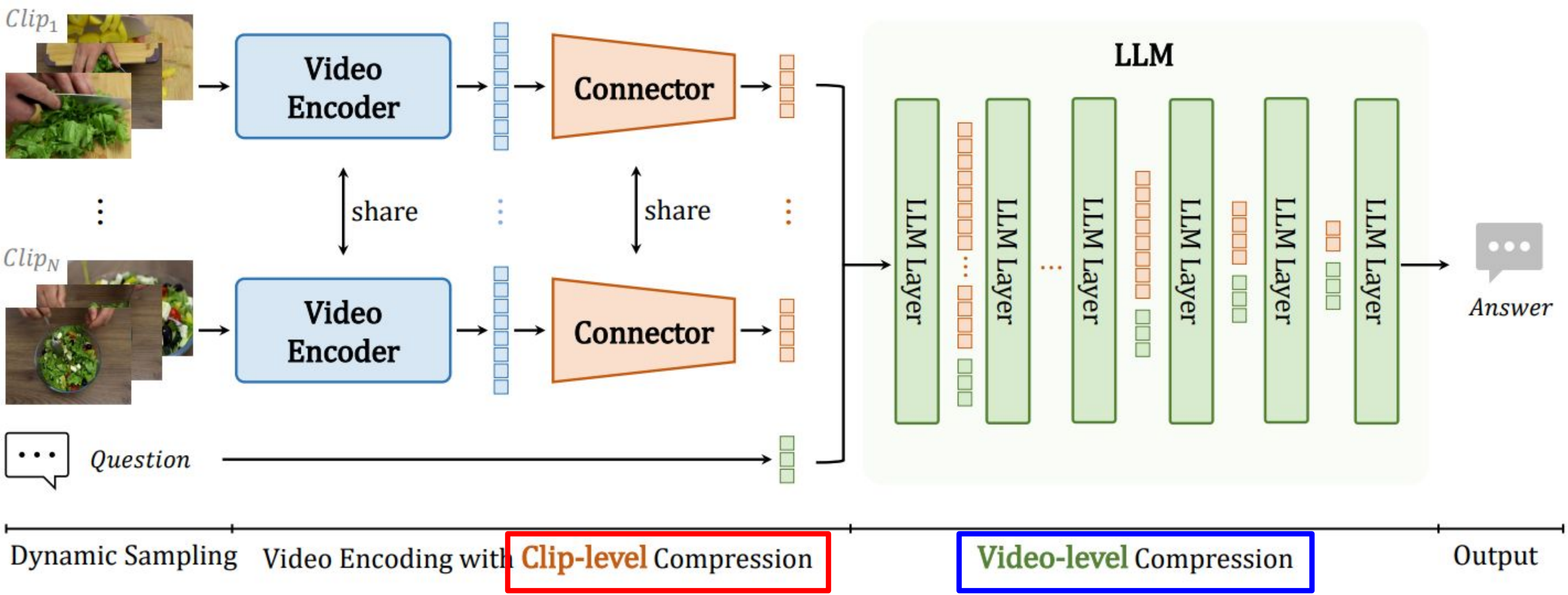
- VLM that **generates video descriptions**
- Superior video understanding capabilities
- 3 stage: Pretraining - SFT - DPO



48. VideoChat-Flash (arxiv 2024)

- Handling **long-form video** content in MLLMs
- **HiCo = Hierarchical visual token Compression**
    - To reduce computational load!
- Two compression
    - **(1) Clip-level**
        - Video = Clips x N
        - Visual encoder processes each clip
        - Compressor reduces # of visual tokens (to exploit inter-frame redundancy)
    - **(2) Video-level**
        - Visual tokens are further reduced via dropout

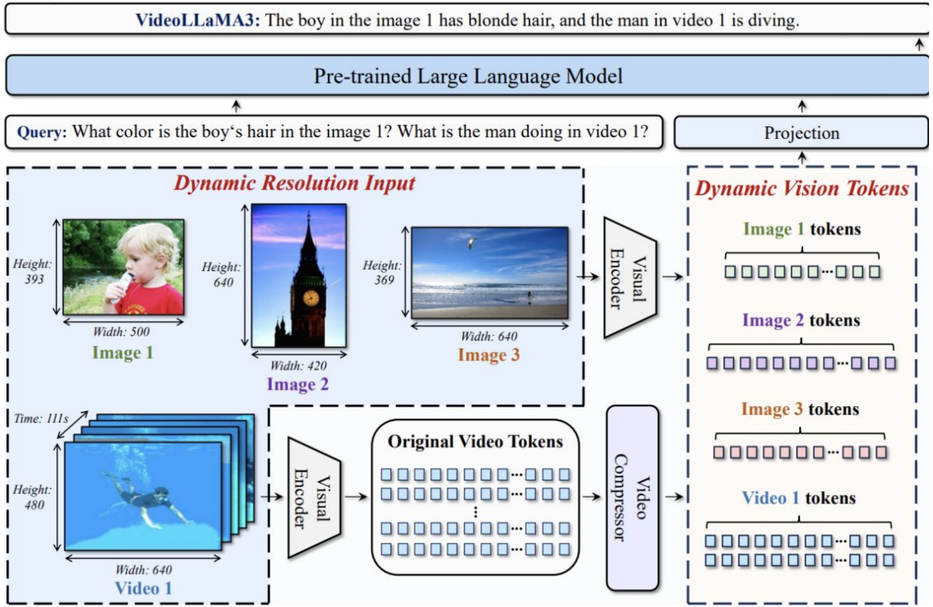
48. VideoChat-Flash (arxiv 2024)



VideoLLaMA 3: Frontier Multimodal Foundation Models for Image and Video Understanding

49. VideoLLaMA 3 (arxiv 2025)

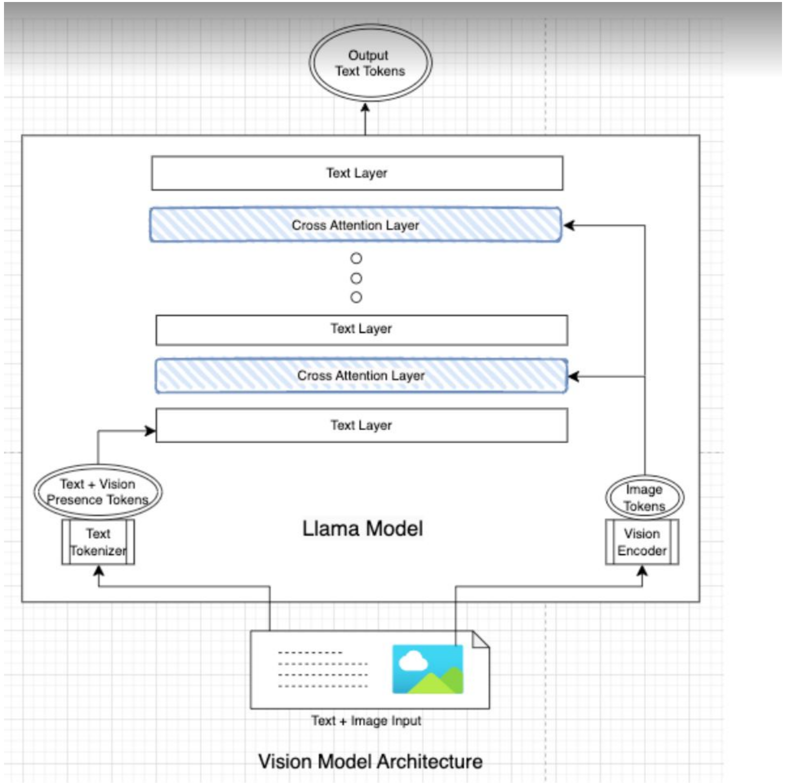
- For both image & video understanding
- Architecture
  - (1) Vision: Pretrained SigLIP
  - (2) Video: DiffFP (compressor)
  - Projector: for (1)+(2) to LLM
  - (3) LLM: Qwen 2.5
- Four training stages



Llama 3.2-Vision: Enhanced Multimodal Capabilities Built on Llama 3

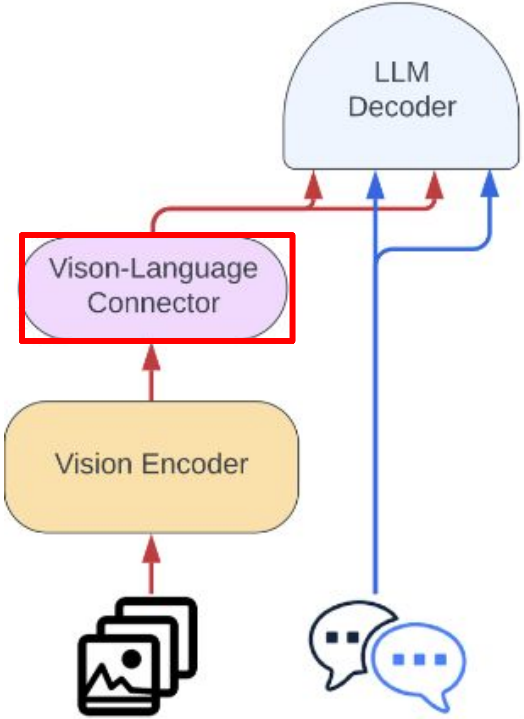
50. LLaMA 3.2-Vision (blog 2024)

- **LLaMA 3** with “**multimodal** capabilities”
- # params: 11B, 90B
- Architecture:
    - (1) Visual encoder
    - (2) LLM: LLaMA 3
    - **Vision adapter**
        - with cross-attention layers
        - for (1) -> (2)



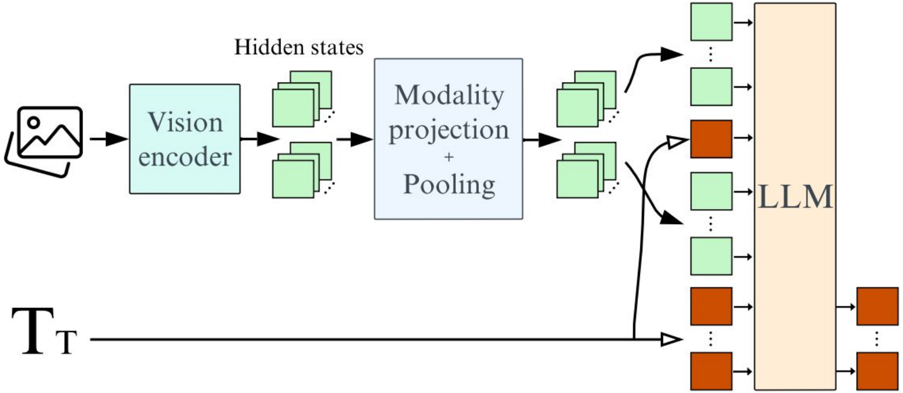
51. Idefics2-8B (blog 2024)

- Efficiently process **interleaved image & text**
- Architecture
  - (1) Vision: SigLIP
  - (2) LLM: Mistral-7B
  - **Perceiver pooling layer + MLP**: for (1) -> (2)



52. Idefics3-8B (arxiv 2024)

- Idefics2-8B보다 뛰어남
- Architecture
  - Vision: SigLIP
  - LLM: LLaMA 3.1 instruct
- Idefics2의 **perceiver resampler를 simple pixel shuffle strategy로 대체** (similar to InternVL-1.5)



53. InternLM-XComposer2 (arxiv 2024)

- Free-form text-image composition & comprehension
- Architecture
  - (1) Vision: CLIP
  - (2) LLM: InternLM-2
  - **Partial LoRA**: apply to visual tokens

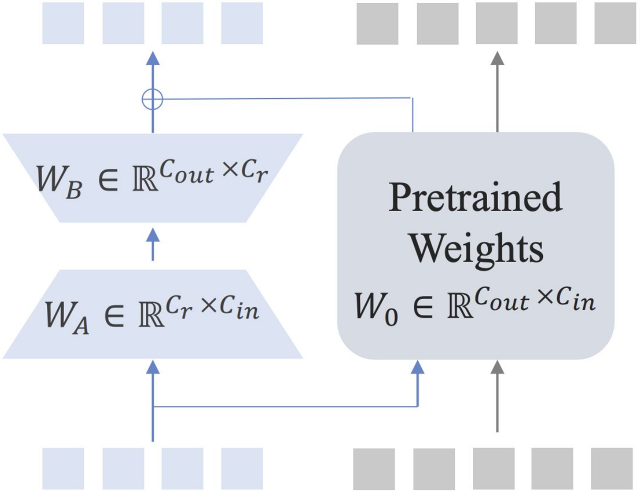

$W_B \in \mathbb{R}^{C_{out} \times C_r}$
$W_A \in \mathbb{R}^{C_r \times C_{in}}$
Pretrained Weights
$W_0 \in \mathbb{R}^{C_{out} \times C_{in}}$

Figure 2. **The illustration of the Partial-LoRA.** The blue tokens represent the visual tokens and the gray tokens are the language tokens. Our Partial-LoRA is only applied to the visual tokens.

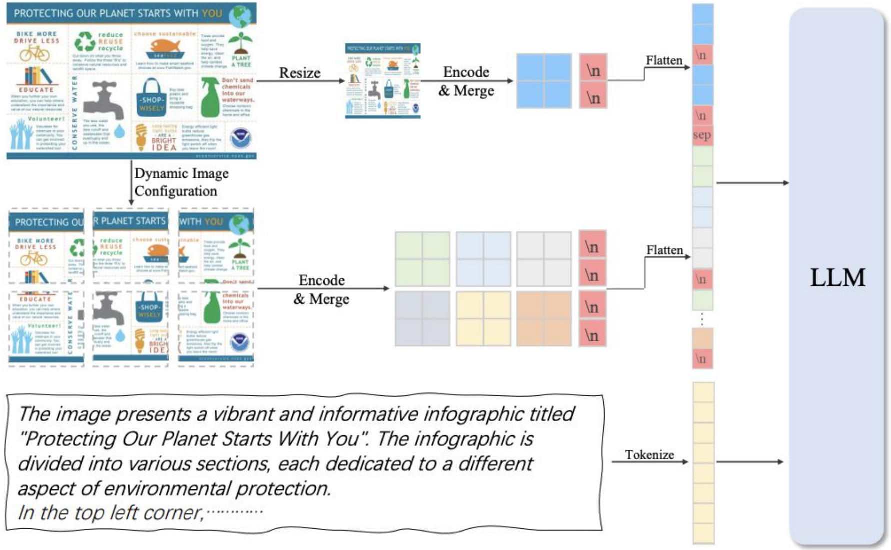
InternLM-XComposer2-4KHD: A Pioneering Large Vision-Language Model Handling Resolutions from 336 Pixels to 4K HD

54. InternLM-XComposer2-4KHD (NeurIPS 2024)

- InternLM-XComposer를 기반으로
- Handle high-resolution image
- How? **Dynamic resolution**

  + Automatic patch configuration

  - Resolution 336 ~ 4K HD



The image presents a vibrant and informative infographic titled "Protecting Our Planet Starts With You". The infographic is divided into various sections, each dedicated to a different aspect of environmental protection.
In the top left corner,…………

Figure 3: The framework of InternLM-XComposer2-4KHD. Our model processes the high-resolution image with a Dynamic Image Partition strategy and concatenates the image tokens with text tokens as LLM input.

InternLM-XComposer-2.5: A Versatile Large Vision Language Model Supporting Long-Contextual Input and Output

55. InternLM-XComposer-2.5 (arxiv 2024)

- Handle "**LONG**" contextual input & output

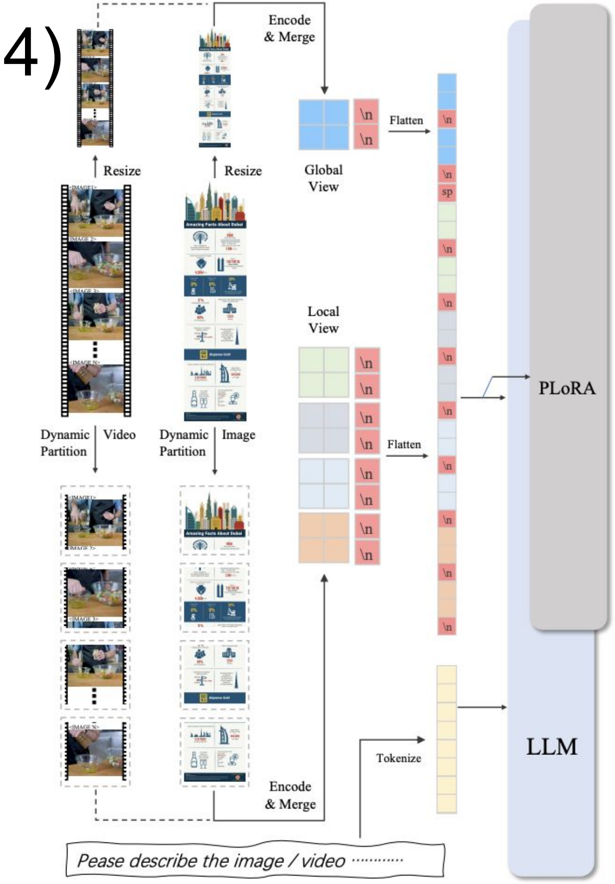
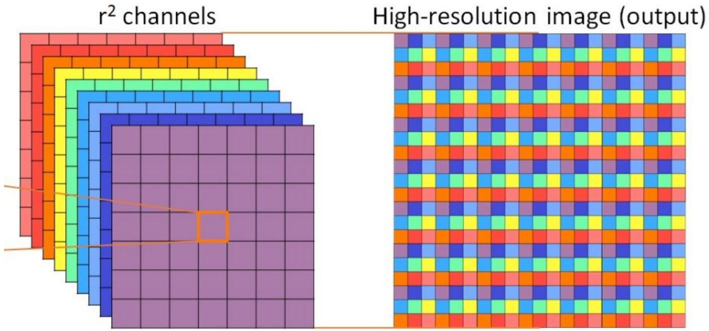


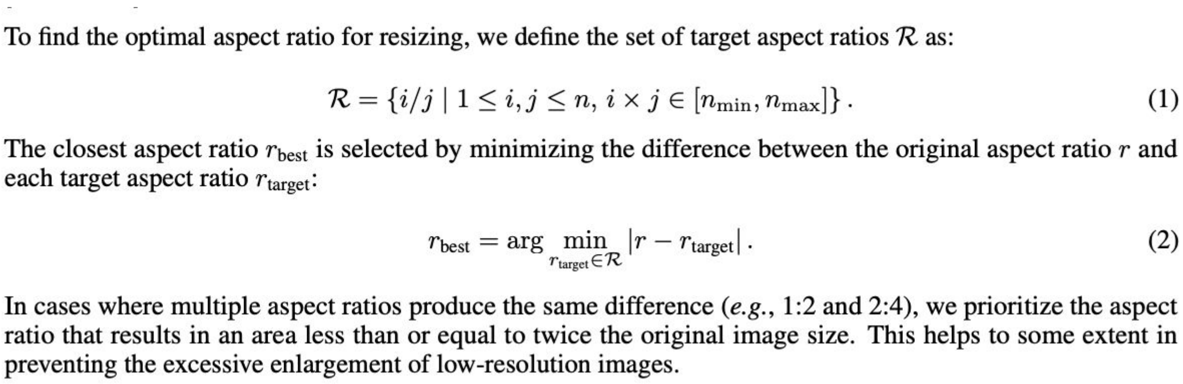
Figure 5. **Framework** of IXC-2.5 that supports the multi-modal inputs, including text, single/multiple images, and videos.

InternVL 2.5: Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling

56. InternVL 2.5 (arxiv 2024)

- InternVL 2.0를 이음
- 448x448 image tile로 나눔
- 핵심: **Pixel unshuffle operation**
    - Visual token: 1024 -> 256로 줄임
- **Dynamic resolution**





To find the optimal aspect ratio for resizing, we define the set of target aspect ratios $\mathcal{R}$ as:

$$\mathcal{R} = \{i/j \mid 1 \leq i, j \leq n,\ i \times j \in [n_{\min}, n_{\max}]\}\,. \tag{1}$$

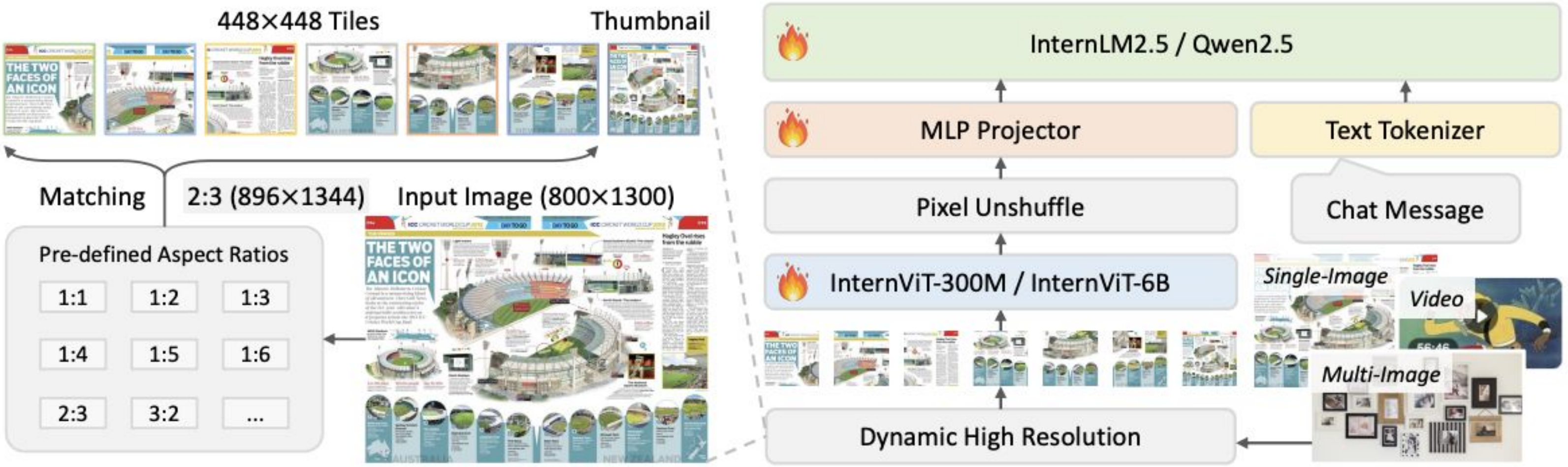
The closest aspect ratio r_{best} is selected by minimizing the difference between the original aspect ratio r and each target aspect ratio r_{target}:

$$r_{\text{best}} = \arg \min_{r_{\text{target}} \in \mathcal{R}} |r - r_{\text{target}}|\,. \tag{2}$$

In cases where multiple aspect ratios produce the same difference (*e.g.*, 1:2 and 2:4), we prioritize the aspect ratio that results in an area less than or equal to twice the original image size. This helps to some extent in preventing the excessive enlargement of low-resolution images.

InternVL 2.5: Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling

56. InternVL 2.5 (arxiv 2024)



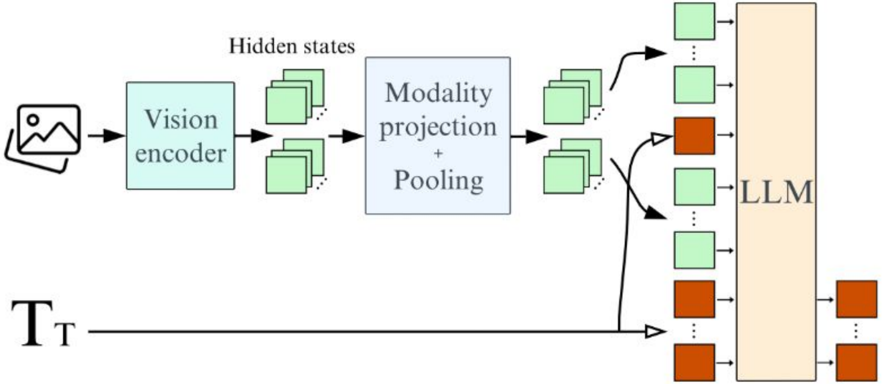
(a) Data Preprocessing

(b) Model Architecture

Figure 2: **Overall architecture.** InternVL 2.5 retains the same model architecture as InternVL 1.5 [35] and InternVL 2.0, *i.e.* the widely-used "ViT-MLP-LLM" paradigm, which combines a pre-trained InternViT-300M or InternViT-6B with LLMs [19, 229] of various sizes via an MLP projector. Consistent with previous versions, we apply a pixel unshuffle operation to reduce the 1024 visual tokens produced by each 448×448 image tile to 256 tokens. Moreover, compared to InternVL 1.5, InternVL 2.0 and 2.5 introduced additional data types, incorporating multi-image and video data alongside the existing single-image and text-only data.

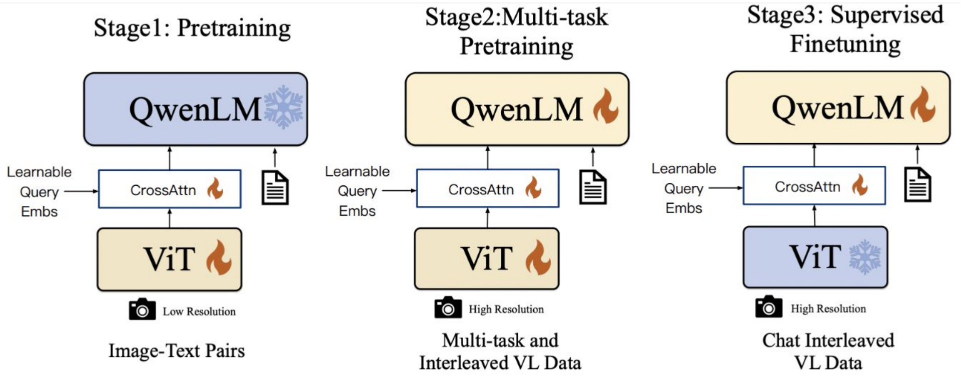
57. SmolVLM (blog 2024)

- **2B** params VLM
- SOTA for its memory footprint
- **Small, fast, memory-efficient**
- Built upon Idefics3
  - LLaMA 3.18B -> SmoLM2 1.7B
- Aggressive image compression



58. Qwen-VL (arxiv 2023)

- LLM인 **QWEN에 visual 능력**을 갖춰줌.
- (1) **Visual receptor**
  - Visual Encoder = ViT
  - Position-aware VL Adapter = 1개의 cross-attention
- (2) Input-output
  - Input = Image & **Bbox**
  - Output = **Bbox**
- (3) Three-stage training pipeline



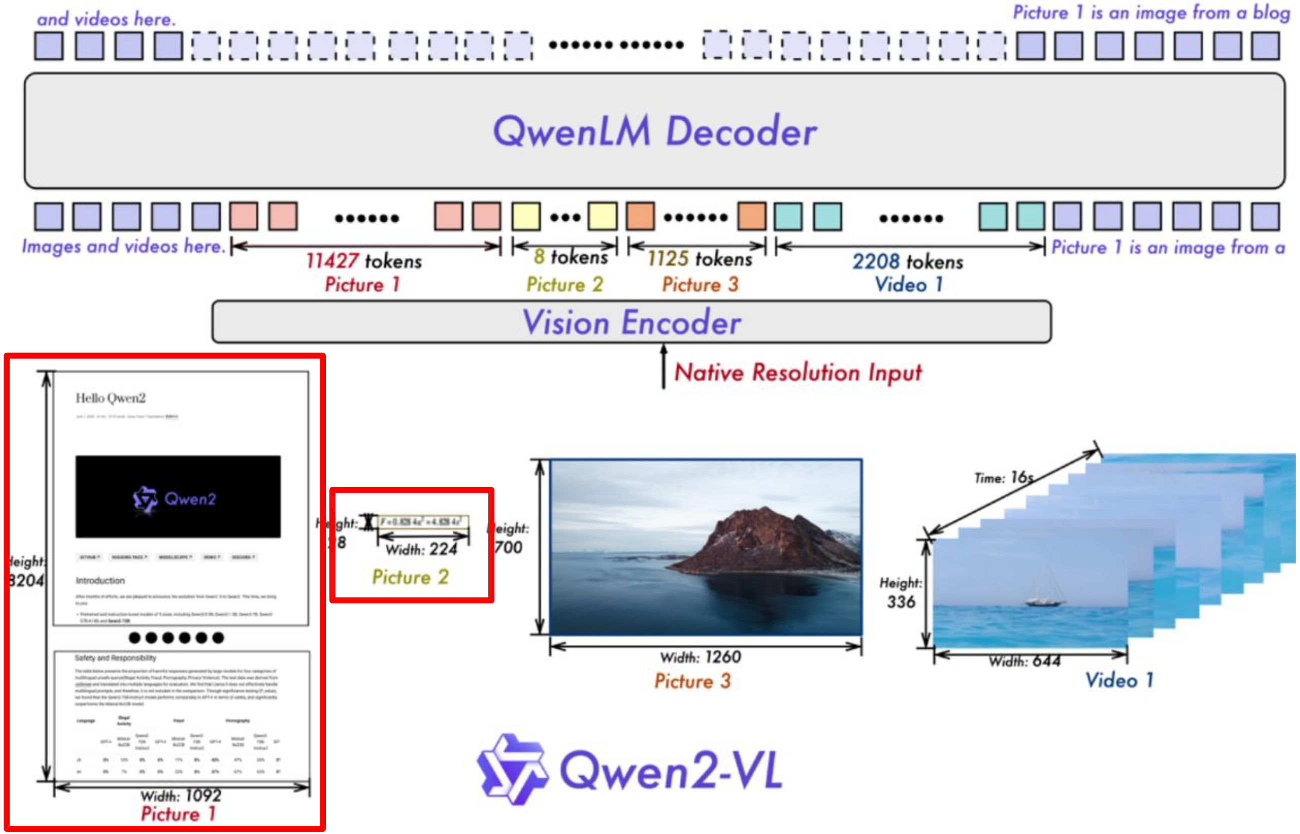
59. Qwen2-VL (arxiv 2024)

- Qwen-VL를 개선함
- 핵심 개선: Improved understanding of images & videos
- Details:
  - (1) **Naive Dynamic Resolution**

    - arbitrary resolution 받아서 dynamic # of visual tokens 반환
  - (2) Multimodal Rotary Position Embedding (**M-ROPE**)
    - temporal + height + width information

Qwen2-VL: A Powerful Open-Source Vision-Language Model for Image and Video Understanding

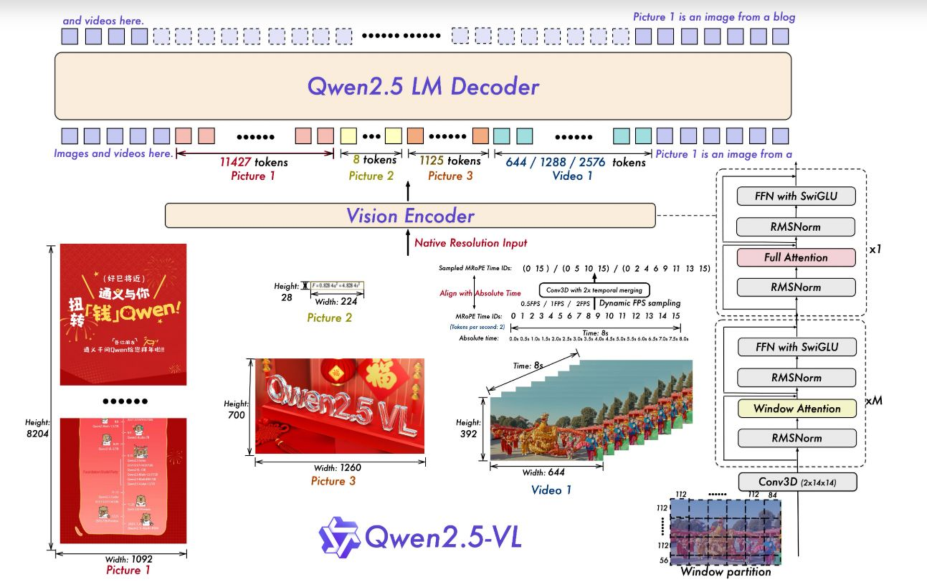
59. Qwen2-VL (arxiv 2024)



60. Qwen2.5-VL (arxiv 2025)

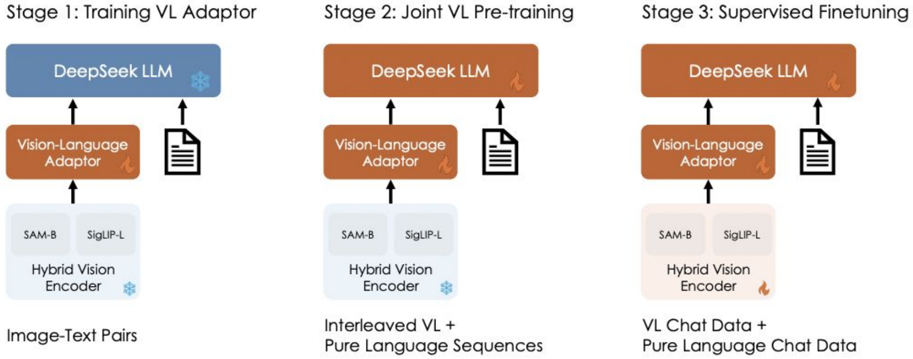
- Qwen2-VL를 개선함
  - Perception of temporal & spatial scales
  - Simplified network structure

    -> increased efficiency



61. DeepSeek-VL (arxiv 2024)

- (1) Data: Diverse & Large-scale
- (2) Model: **Hybrid** vision encoder
  - LLM: Deepseek LLM (feat. LLaMA)
  - Vision:
    - a) **SigLIP** (low res)
    - b) **SAM-B** (high res)
    - VL adaptor (for a+b)
- (3) Train: Focus on **text -> img**



61. DeepSeek-VL2 (arxiv 2024)

- DeepSeek-VL + Two upgrades
- (1) [Vision] **Dynamic tiling** vision encoding strategy
- (2) [LLM] DeepSeekMoE + MLA

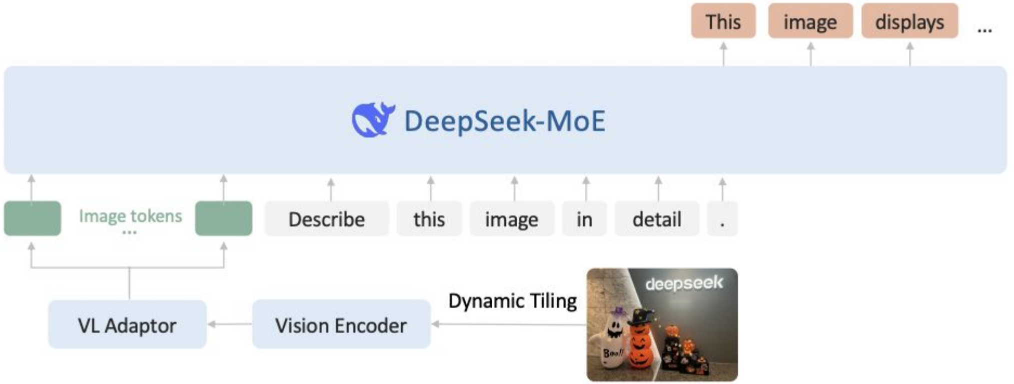


Figure 2 | **Overview of DeepSeek-VL2**. The overall structure is a llava-style architecture, which includes a vision encoder, a VL adaptor, and a MoE-based LLM.

61. DeepSeek-VL2 (arxiv 2024)

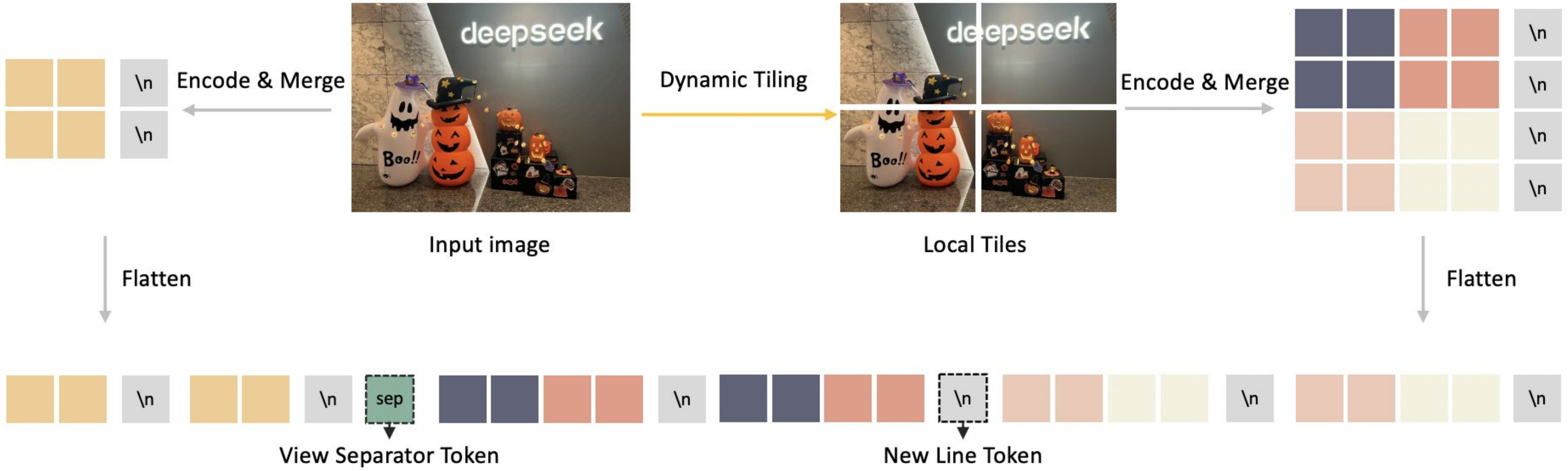
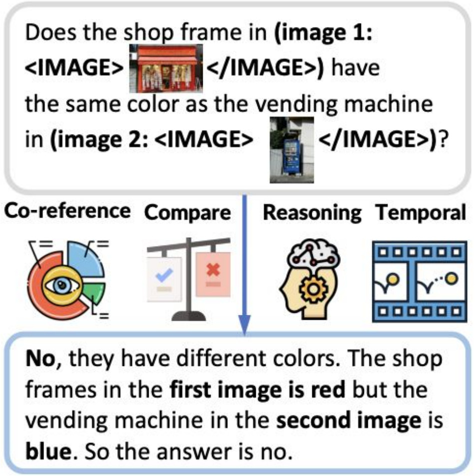


Figure 3 | **Illustration of dynamic tiling strategy in DeepSeek-VL2.** By dividing images into multiple tiles, DeepSeek-VL2 achieves stronger fine-grained understanding capabilities compared to DeepSeek-VL.

Mantis: Interleaved Multi-Image Instruction Tuning

62. MANTIS (TMLR 2024)

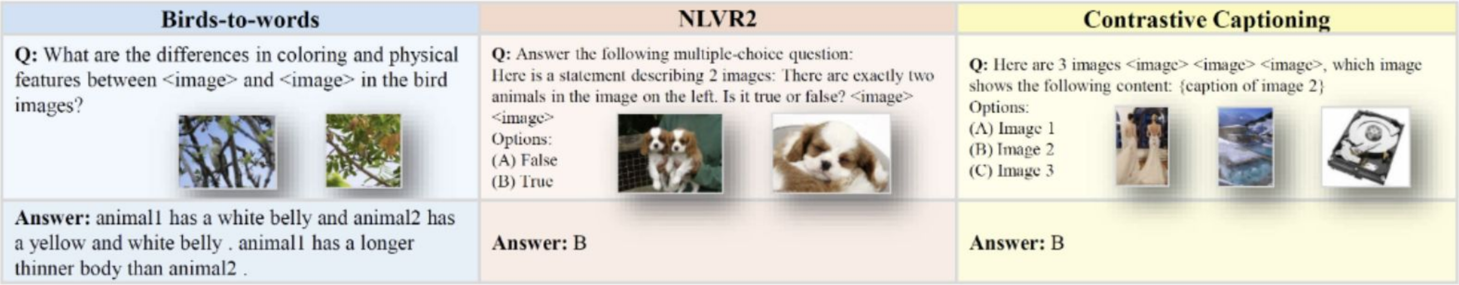
- **Multi-image** VL tasks
  - (e.g., coherence, comparison …)
- **Interleaved** text-image
- Trained with significantly **less datasets**!



Does the shop frame in **(image 1: <IMAGE>** **</IMAGE>)** have the same color as the vending machine in **(image 2: <IMAGE>** **</IMAGE>)**?

No, they have different colors. The shop frames in the **first image is red** but the vending machine in the **second image** is **blue**. So the answer is no.

Training: Mantis-Instruct dataset



Birds-to-words	NLVR2	Contrastive Captioning
Q: What are the differences in coloring and physical features between <image> and <image> in the bird images?	**Q:** Answer the following multiple-choice question: Here is a statement describing 2 images: There are exactly two animals in the image on the left. Is it true or false? <image> <image> Options: (A) False (B) True	**Q:** Here are 3 images <image> <image> <image>, which image shows the following content: {caption of image 2} Options: (A) Image 1 (B) Image 2 (C) Image 3
Answer: animal1 has a white belly and animal2 has a yellow and white belly . animal1 has a longer thinner body than animal2 .	**Answer:** B	**Answer:** B

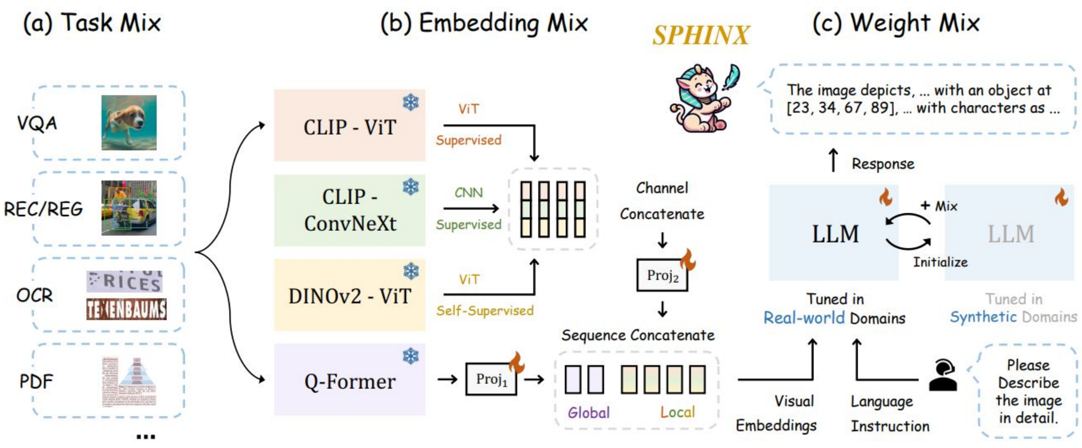
63. moondream (blog 2024)

For **practical usage**!

- moondream1: 1.6B
  - Model: SigLIP + Phi-1.5
  - Dataset: LLaVA
- moondream2: 1.86B
  - moondream1 + MLP projector
- moondream-next: 1.9B
  - structured output (JSON, XML, MD,CSV) …

64. SPHINX (arxiv 2024)

- Jointly **mixing**
  - **(1) Model weight**
  - **(2) Tasks**
  - **(3) Visual embeddings**
- 2-stage approach
  - Step 1. Pretraining) Unfreeze LLaMA-2
    - To foster more effective cross-modal learning
  - Step 2. Fine-tuning



65. SPHINX-X (ICML 2024)

- Use two visual encoders: **CLIP-ConvNeXt & DINOv2**
- Learnable skip tokens (to bypass fully-padded sub-images)
- Efficient single-stage training process

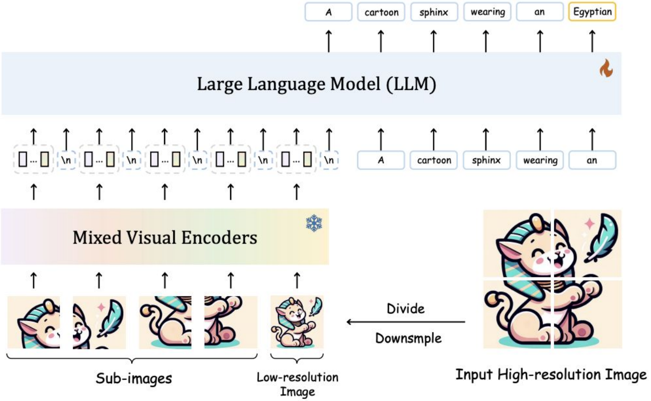


Figure 4: **Pipeline of *SPHINX* for high-resolution images.** We propose to further mix different scales and sub-images to better capture fine-grained semantics on high-resolution images.

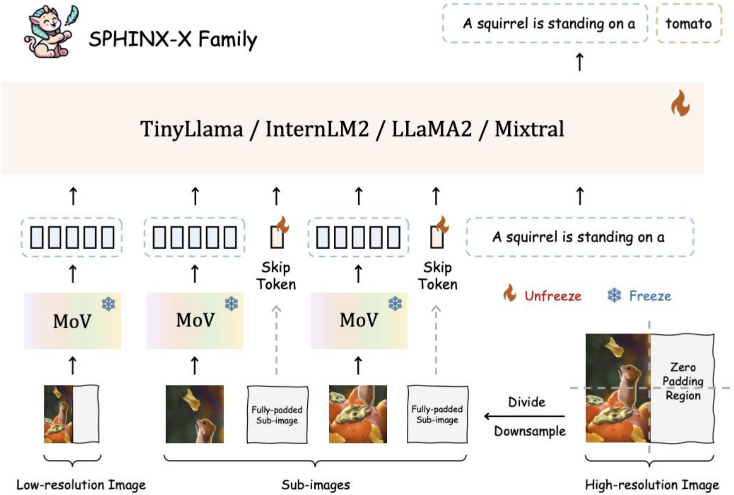
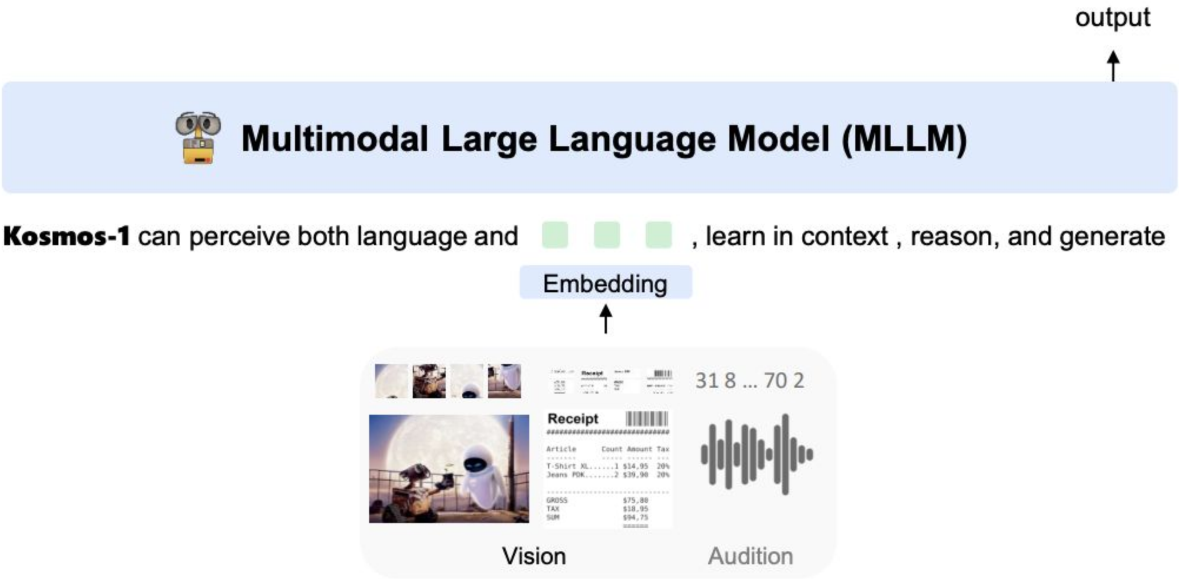


Figure 3. **Overall paradigm of SPHINX-X family.** On top of SPHINX (Lin et al., 2023), we adopt three modifications for a more general and concise architecture: removing redundant visual encoders in Mixture of Visual Experts (MoV), bypassing fully-padded sub-images with skip tokens, and simplifying multi-stage training into a one-stage all-in-one approach.

Language Is Not All You Need: Aligning Perception with Language Models

66. KOSMOS-1 (NeurIPS 2023)

- Can perceive **general modalities**
- Align various modalities with LLM


output
Multimodal Large Language Model (MLLM)
Kosmos-1 can perceive both language and 🟩 🟩 🟩 , learn in context , reason, and generate
Embedding
Receipt
Article Count Amount Tax
T-Shirt XL.......1 $14,95 20%
Jeans PDK........2 $39,90 20%
GROSS $75,80
TAX $18,95
SUM $94,75
31 8 ... 70 2
Vision
Audition

67. KOSMOS-2 (ICLR 2024)

- KOSMOS-1에 새로운 capability
  - Perceiving **object description (Bbox)**
  - **Grounding text**

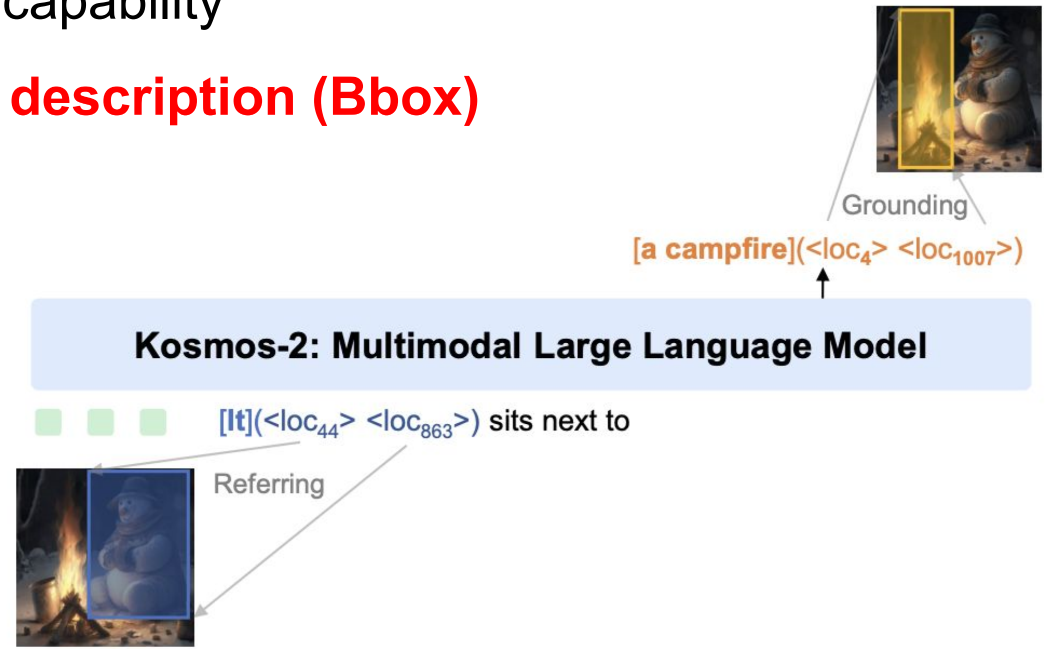
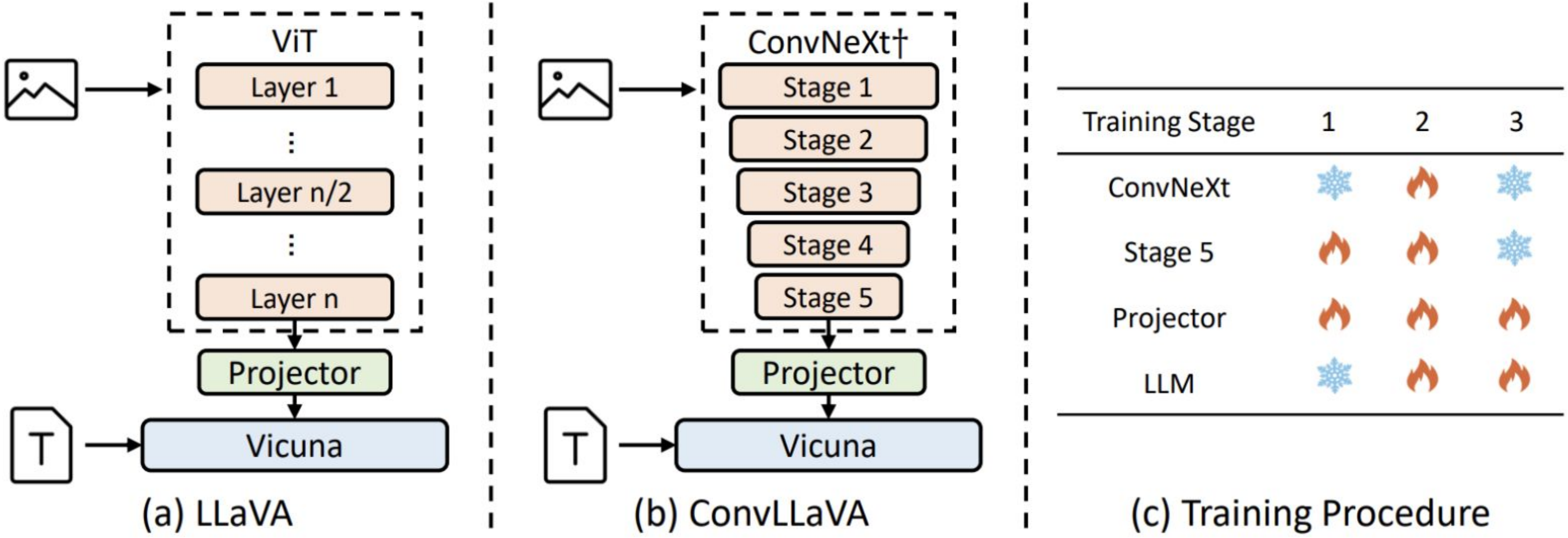

Grounding
[a campfire](<loc$_4$> <loc$_{1007}$>)
Kosmos-2: Multimodal Large Language Model
[It](<loc$_{44}$> <loc$_{863}$>) sits next to
Referring

Figure 1: KOSMOS-2 is a multimodal large language model that has new capabilities of multimodal grounding and referring. KOSMOS-2 can understand multimodal input, follow instructions, perceive object descriptions (*e.g.*, bounding boxes), and ground language to the visual world.

ConvLLaVA: Hierarchical Backbones as Visual Encoder for Large Multimodal Models

68. ConvLLAVA (arxiv 2024)

- High-resolution LMM에서 **ViT의 한계점**을 극복하기 위해!
- Hierarchical backbone “**ConvNeXT**”를 대신 사용함



(a) LLaVA

(b) ConvLLaVA

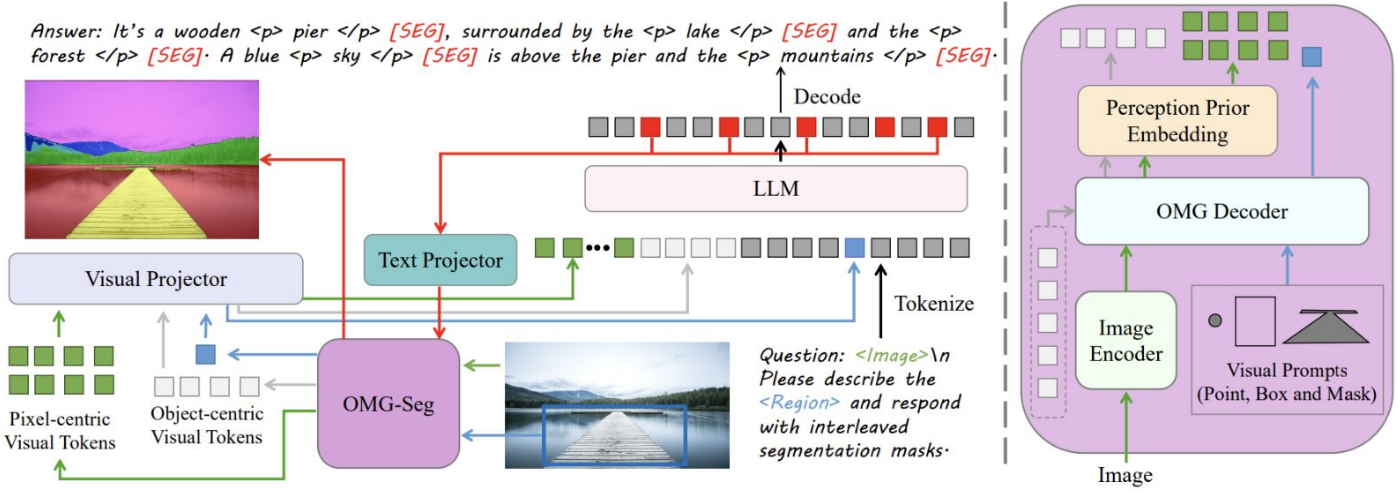
Training Stage	1	2	3
ConvNeXt	❄	🔥	❄
Stage 5	🔥	🔥	❄
Projector	🔥	🔥	🔥
LLM	❄	🔥	🔥

(c) Training Procedure

OMG-LLaVA: Bridging Image-level, Object-level, Pixel-level Reasoning and Understanding

69. OMG-LLaVA (arxiv 2024)

- Unifies (1) **Image**-level + (2) **Object**-level + (3) **Pixel**-level reasoning
- Architecture:
  - (1) Vision: **OMG-seg (segmentation model)**
  - (3) LLM



Answer: It's a wooden <p> pier </p> [SEG], surrounded by the <p> lake </p> [SEG] and the <p> forest </p> [SEG]. A blue <p> sky </p> [SEG] is above the pier and the <p> mountains </p> [SEG].

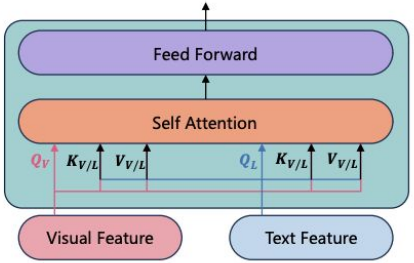
Question: <Image>\n Please describe the <Region> and respond with interleaved segmentation masks.

Decode
LLM
Text Projector
Visual Projector
Tokenize
OMG-Seg
Pixel-centric Visual Tokens
Object-centric Visual Tokens
Perception Prior Embedding
OMG Decoder
Image Encoder
Visual Prompts (Point, Box and Mask)
Image

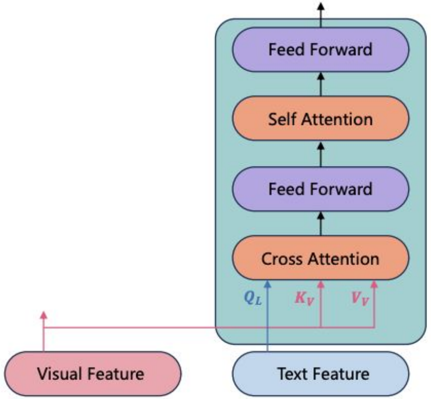
An Efficient Vision-Language Model for Visual Understanding

70. EVLM (arxiv 2024)

- Architecture: **Flamingo**
- Goal: Minimize computational cost!
- Challenges of handling long sequences of visual signals
- How?
  - **Cross-attention**
  - **Hierarchical ViT features**
- **(Q-former처럼) learnable tokens**

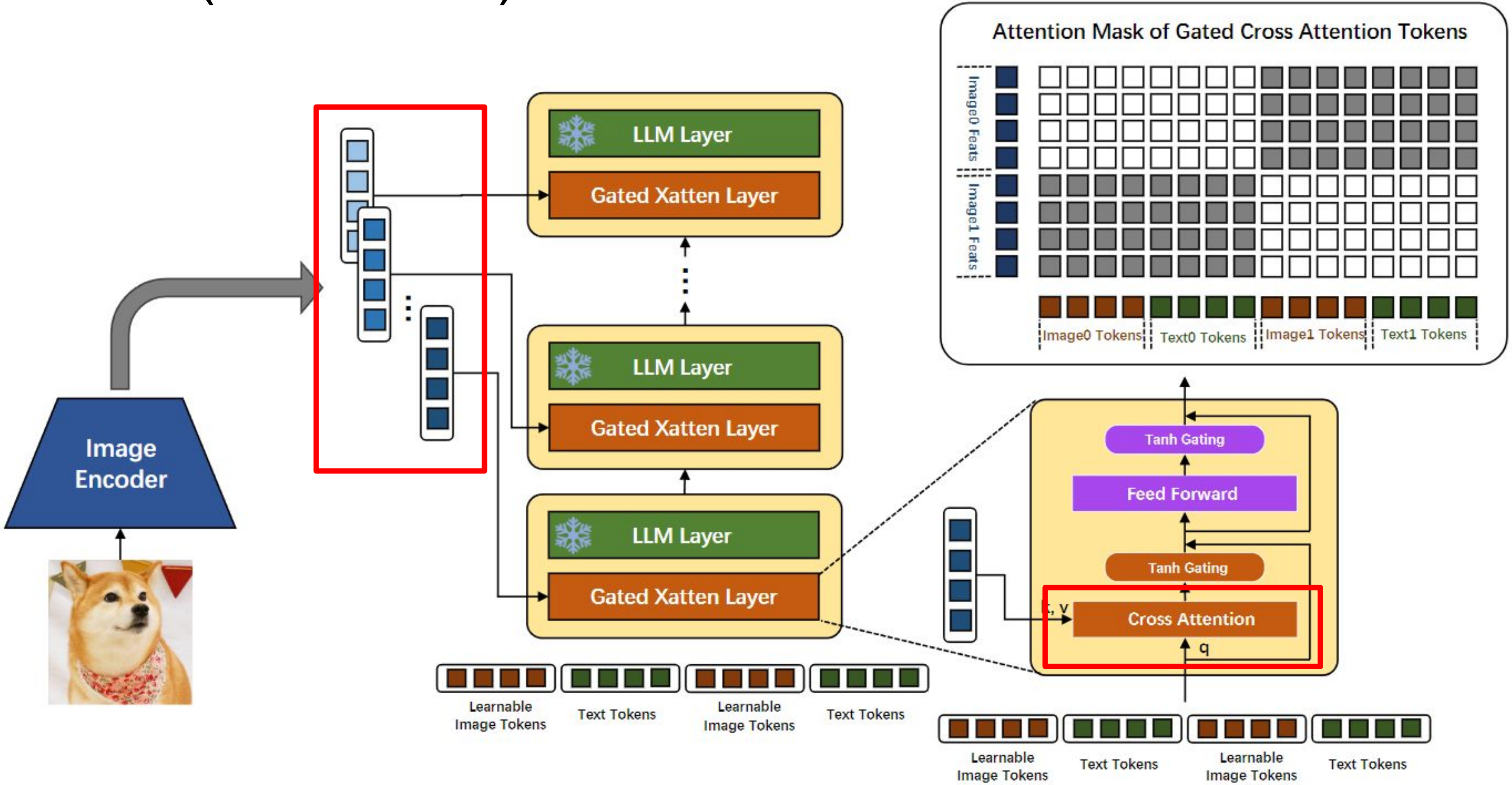


Full Attention (LLaVA)



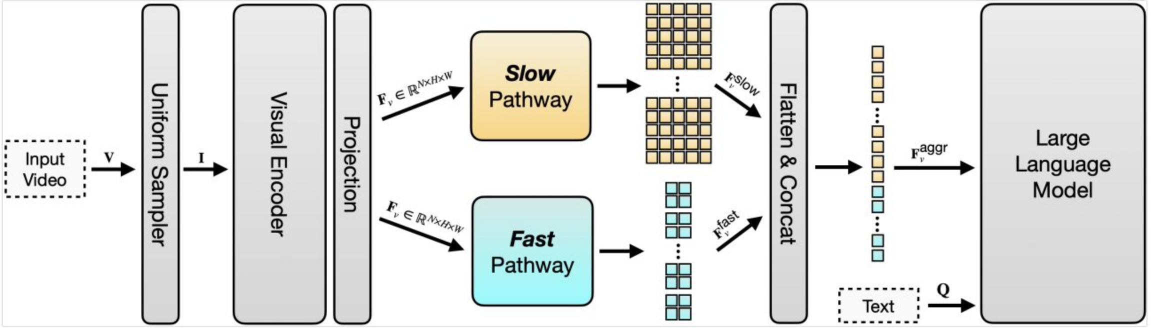
Cross Attention (Flamingo)

70. EVLM (arxiv 2024)



71. SlowFast-LLaVA (arxiv 2024)

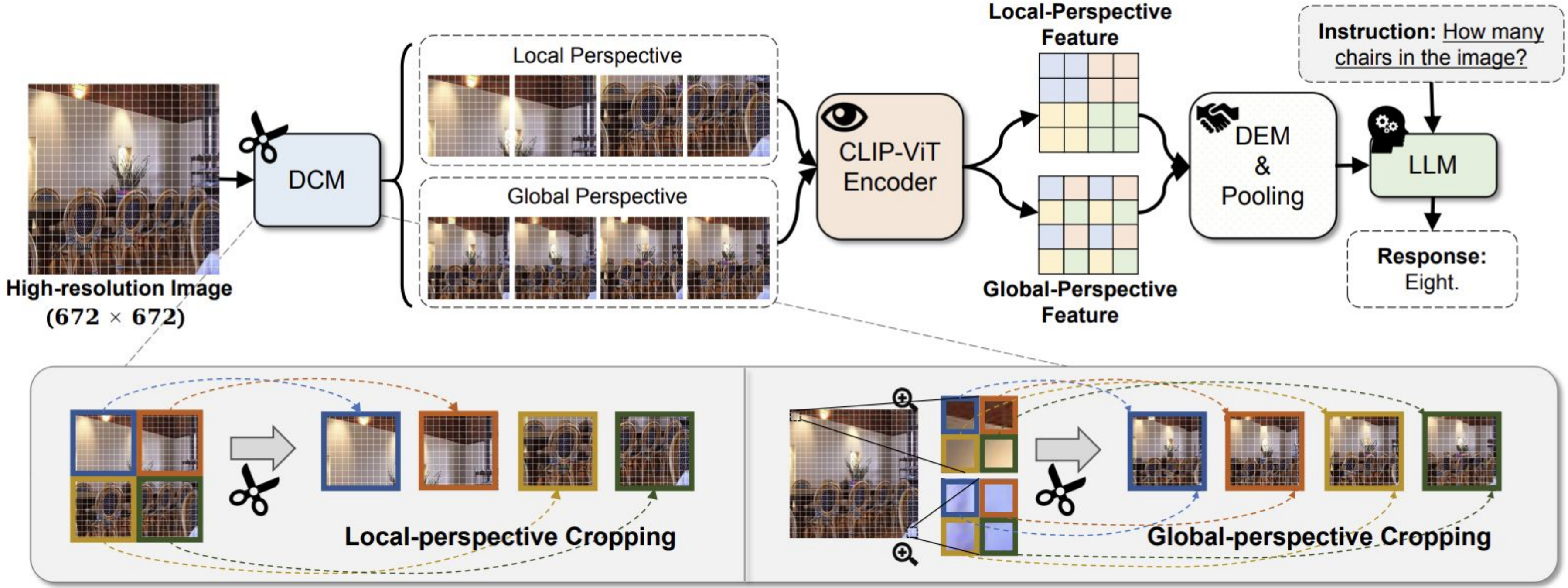
- **Training-free** Video LLM
- (1) Uniform sampler: Sample N frames from video
- (2) Pathways
  - **Slow** pathway: Detailed spatial semantics
  - **Fast** pathway: Long-range temporal context
- (3) LLM: LLaVA-Next


Input Video
$\mathbf{V}$
Uniform Sampler
$\mathbf{I}$
Visual Encoder
Projection
$\mathbf{F}_v \in \mathbb{R}^{N \times H \times W}$
Slow Pathway
$\mathbf{F}_v \in \mathbb{R}^{N \times H \times W}$
Fast Pathway
$\mathbf{F}_v^{\text{slow}}$
$\mathbf{F}_v^{\text{fast}}$
Flatten & Concat
$\mathbf{F}_v^{\text{aggr}}$
Text
$\mathbf{Q}$
Large Language Model

72. INF-LLaVA (arxiv 2024)

- MLLM to efficiently process **high-resolution images**
- 기존 연구: cropping-based & dual-encoder
- Proposal
    - **Dual-perspective** cropping module (DCM)
    - **Dual-perspective** enhancement module (DEM)

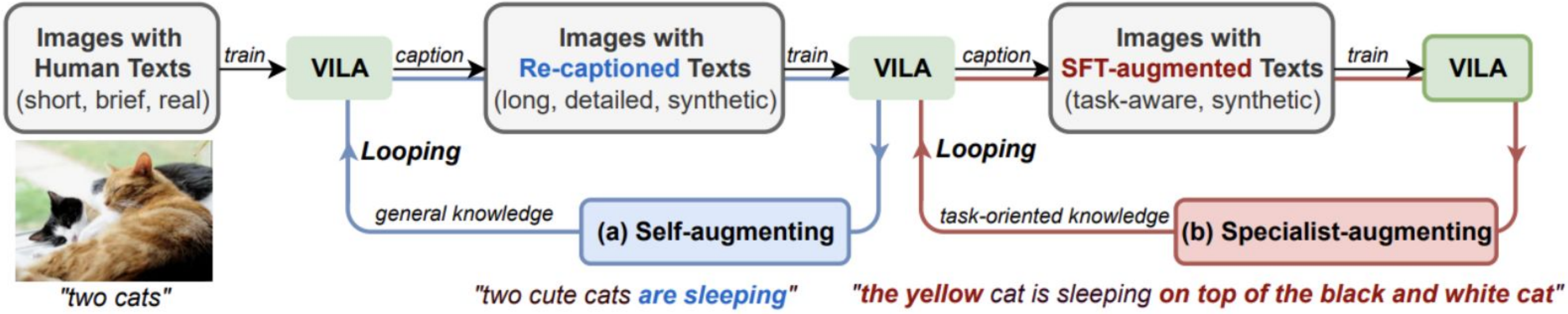
72. INF-LLaVA (arxiv 2024)


High-resolution Image
(672 × 672)
DCM
Local Perspective
Global Perspective
CLIP-ViT Encoder
Local-Perspective Feature
Global-Perspective Feature
DEM & Pooling
Instruction: How many chairs in the image?
LLM
Response: Eight.
Local-perspective Cropping
Global-perspective Cropping

73. **VILA²** (arxiv 2024)

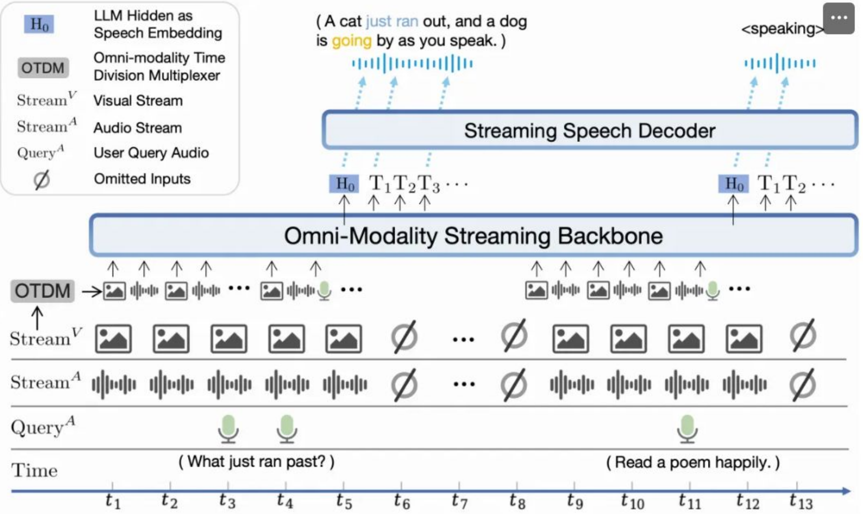
- Limitation of data quantity & quality in VLMs
- Costly human annotation, distillation 대신…

=> Leverage VLM **itself** to **refine & augment** pretraining data!


Images with Human Texts (short, brief, real)
train
VILA
caption
Images with Re-captioned Texts (long, detailed, synthetic)
train
VILA
caption
Images with SFT-augmented Texts (task-aware, synthetic)
train
VILA
Looping
general knowledge
(a) Self-augmenting
Looping
task-oriented knowledge
(b) Specialist-augmenting
"two cats"
"two cute cats are sleeping"
"the yellow cat is sleeping on top of the black and white cat"

74. MiniCPM-o-2.6 (arxiv 2024)

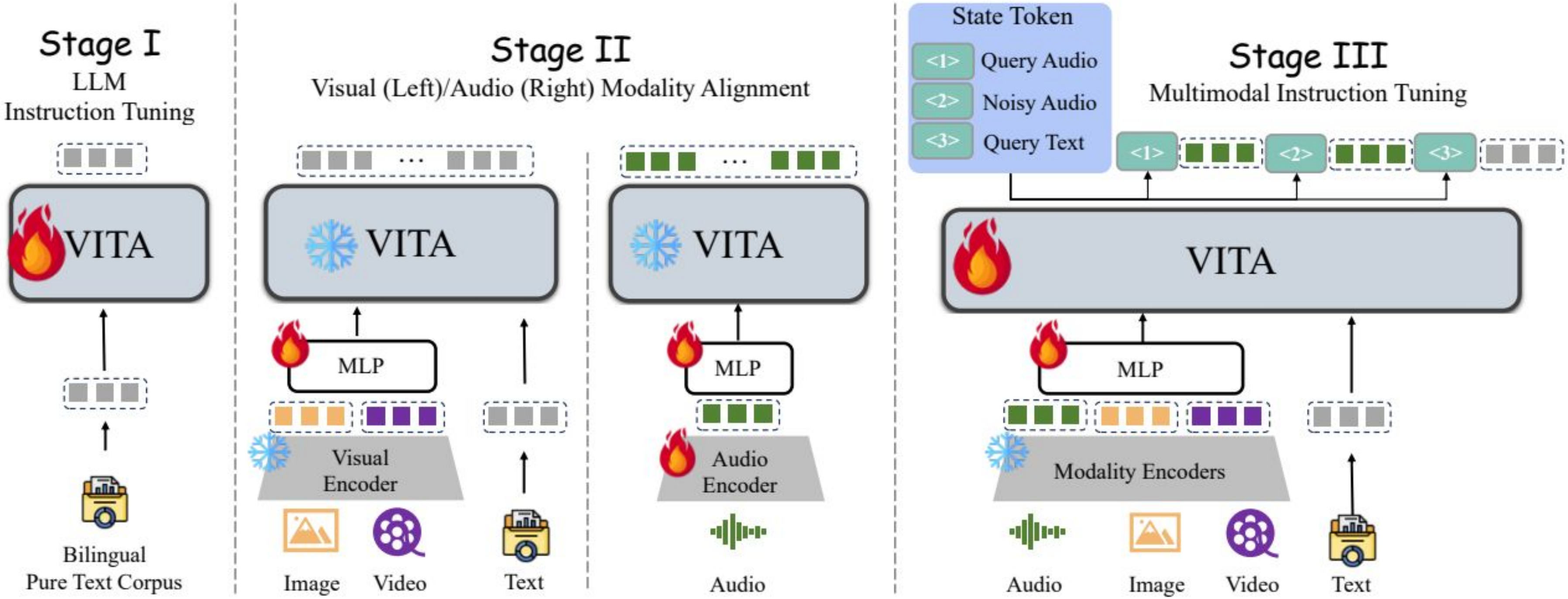
- **8B** MLLM
  - Efficient for deployment on **edge devices**
- **Vision, Speech, and Multimodal** live streaming
- Architecture
  - Vision: SigLIP-400M
  - Audio: Whisper-medium-300M
  - TTS: ChatTTS-200M
  - LLM: Qwen2.5-7B


H_0 LLM Hidden as Speech Embedding
OTDM Omni-modality Time Division Multiplexer
Stream^V Visual Stream
Stream^A Audio Stream
Query^A User Query Audio
∅ Omitted Inputs
(A cat just ran out, and a dog is going by as you speak.)
<speaking>
Streaming Speech Decoder
H_0 $T_1 T_2 T_3 \cdots$
H_0 $T_1 T_2 \cdots$
Omni-Modality Streaming Backbone
OTDM
Stream^V
Stream^A
Query^A
(What just ran past?)
(Read a poem happily.)
Time
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13}

75. VITA (arxiv 2024)

- First **open-source** MLLM capable of processing video, image, text, audio
- Architecture
  - LLM: Mixtral 8x7B
  - Vision: InternViT-300M-448px
- Three stage training
  - Stage 1) **LLM** instruction tuning
  - Stage 2) Multimodal **alignment**
  - Step 3) Multimodal **instruction tuning**

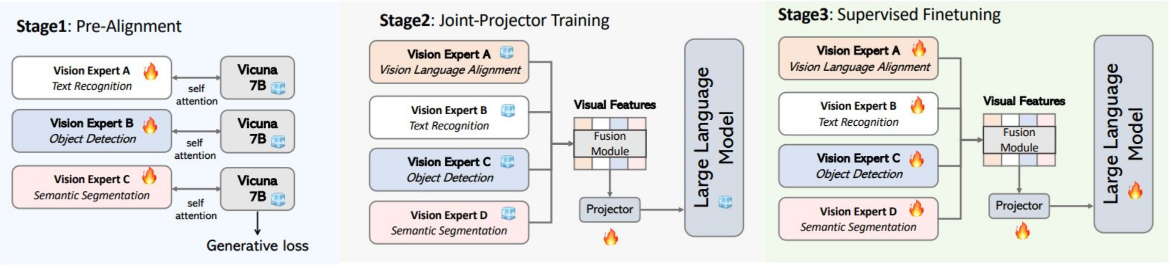
75. VITA (arxiv 2024)



EAGLE: Exploring The Design Space for Multimodal LLMs with Mixture of Encoders

76. EAGLE (ICLR 2025)

- **Open-source** MLLMs
- Mixture of vision encoders
- Particularly in OCR and document understanding
- Architecture: LLaVA
    - (1) LLM
    - (2) Vision encoder
    - Projection layer
- 3 stages of training



Eagle 2: Building Post-Training Data Strategies from Scratch for Frontier Vision-Language Models

77. EAGLE2 (arxiv 2025)

- “**Data-centric**” approach
  - Prioritize data diversity & quality
- Tiled Mixture of vision encoders (MoVE)
  - SigLIP & ConvNeXt-XXLarge
  - Image tiling to handle high resolution
- LLM: Qwen2.5
- Build upon open-source components
- Three-stage training recipe

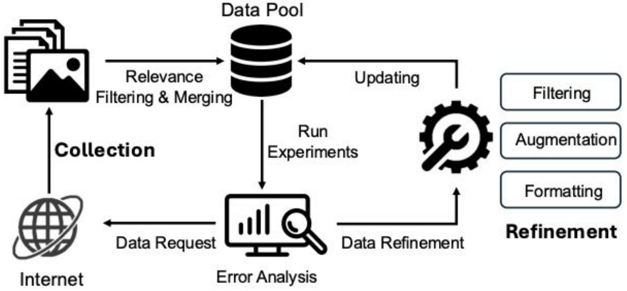


Figure 3 | An overview of our data strategy. The upper part shows the date collection pipeline and the lower part shows the data refinement pipeline.

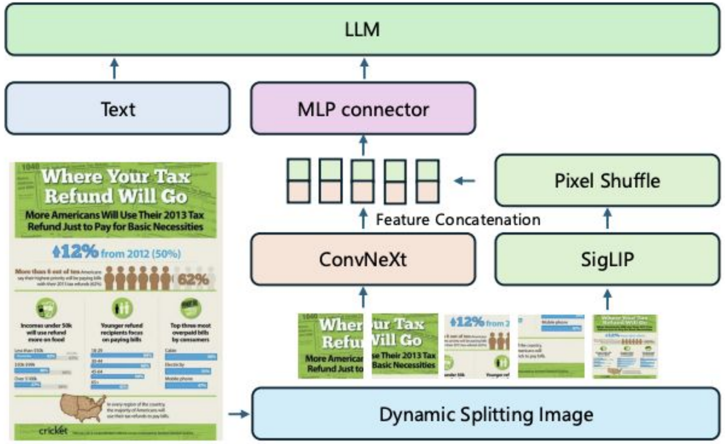


Figure 11 | Tiled Mixture of Vision Encoders.

78. Florence (arxiv 2021)

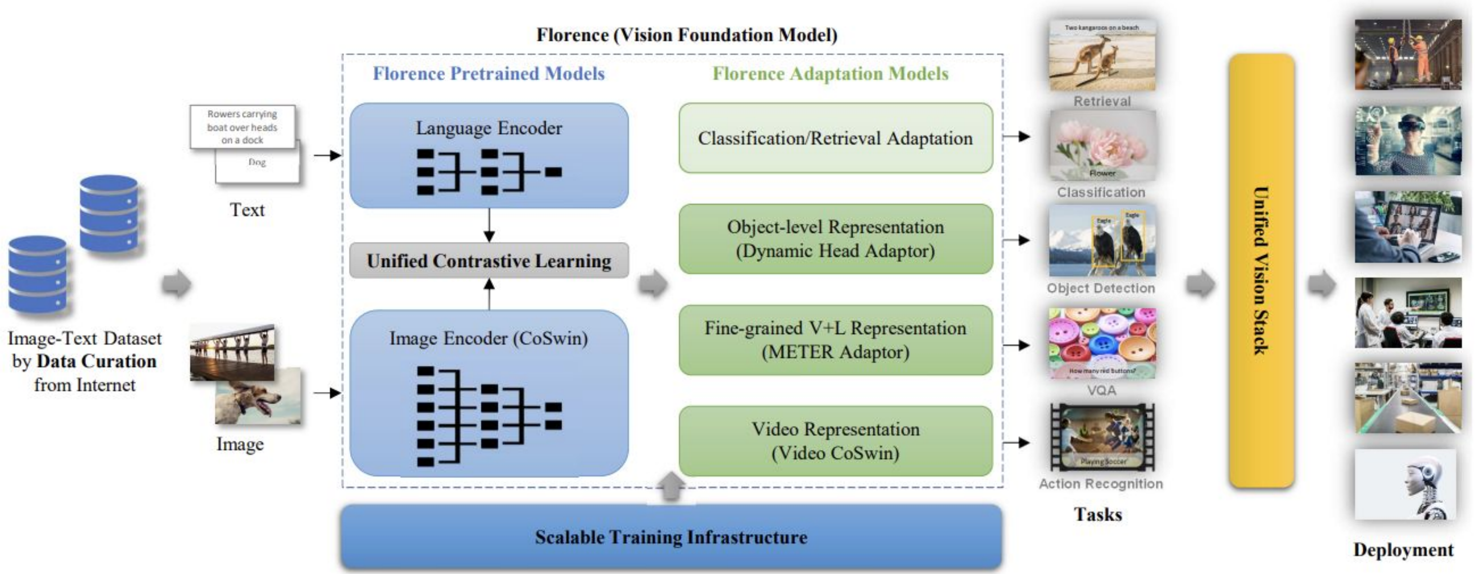

Florence (Vision Foundation Model)
Florence Pretrained Models
Florence Adaptation Models
Rowers carrying boat over heads on a dock
Dog
Text
Image
Image-Text Dataset by Data Curation from Internet
Language Encoder
Unified Contrastive Learning
Image Encoder (CoSwin)
Classification/Retrieval Adaptation
Object-level Representation (Dynamic Head Adaptor)
Fine-grained V+L Representation (METER Adaptor)
Video Representation (Video CoSwin)
Scalable Training Infrastructure
Two kangaroos on a beach
Retrieval
Flower
Classification
Eagle
Object Detection
How many red buttons?
VQA
Playing Soccer
Action Recognition
Tasks
Unified Vision Stack
Deployment

Figure 2. Overview of building Florence. Our workflow consists of data curation, unified learning, Transformer architectures and adaption. It shows the foundation model can be adapted to various downstream tasks and finally integrated into modern computer vision system to power real-world vision and multimedia applications. Compared with existing image-text pretraining models (Radford et al., 2021; Jia et al., 2021; Wud), mainly limited on cross-modal shared representation for classification and retrieval (illustrated by *light-green* adaptation module), Florence expands the representation to support object level, multiple modality, and videos respectively.

78. Florence (arxiv 2021)

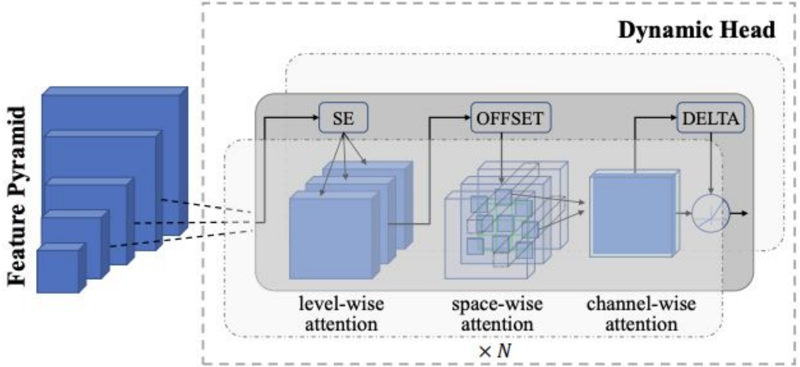


Figure 3. Dynamic Head (Dai et al., 2021a) adapter is used for object-level visual representation learning.

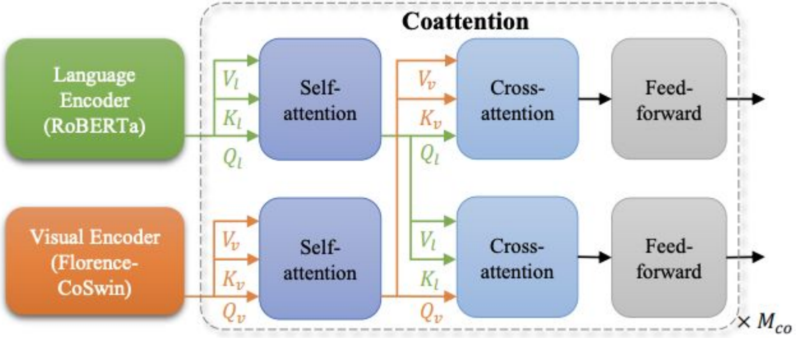
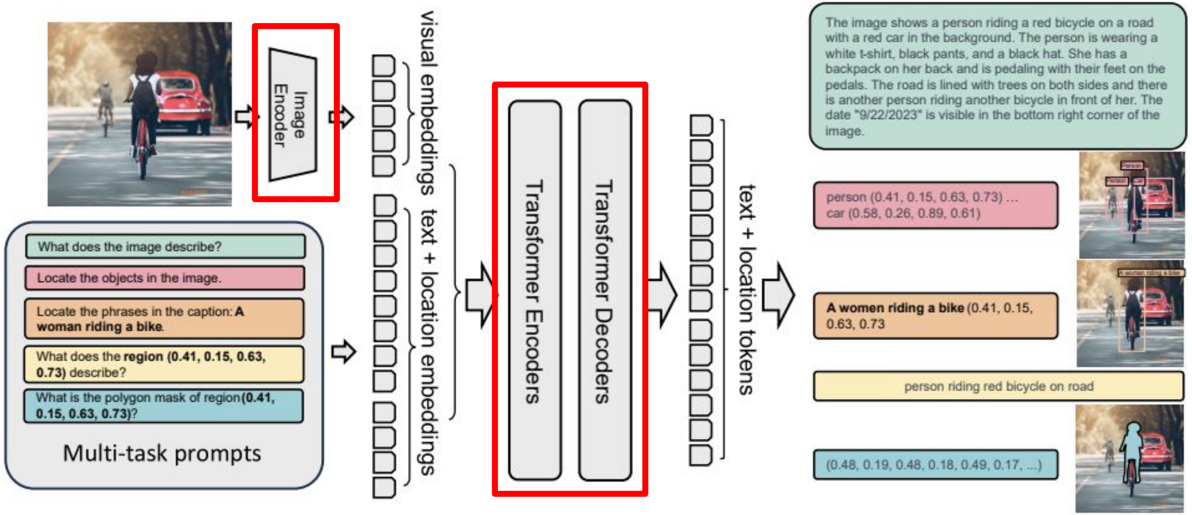


Figure 4. METER (Dou et al., 2021) is used as Florence V+L adaptation model, trained with the image-text matching (ITM) loss and the masked language modeling (MLM) loss.

Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks

79. Florence-2 (CVPR 2024)

- Two key components
  - (1) Image encoder (DaViT arch)
  - (2) Multimodality encoder-decoder.



What does the image describe?

Locate the objects in the image.

Locate the phrases in the caption: **A woman riding a bike**.

What does the **region (0.41, 0.15, 0.63, 0.73)** describe?

What is the polygon mask of region**(0.41, 0.15, 0.63, 0.73)**?

The image shows a person riding a red bicycle on a road with a red car in the background. The person is wearing a white t-shirt, black pants, and a black hat. She has a backpack on her back and is pedaling with their feet on the pedals. The road is lined with trees on both sides and there is another person riding another bicycle in front of her. The date "9/22/2023" is visible in the bottom right corner of the image.

person (0.41, 0.15, 0.63, 0.73) ...
car (0.58, 0.26, 0.89, 0.61)

A women riding a bike (0.41, 0.15, 0.63, 0.73

person riding red bicycle on road

(0.48, 0.19, 0.48, 0.18, 0.49, 0.17, ...)

80. MultiInstruct (arxiv 2022)

- To enhance multi-modal zero-shot learning
  - By **Instruction tuning**
- Built upon **OFA (One For All)** as pretrained model
  - Single Transformer-based **sequence-to-sequence** framework
  - Pre-trained on a diverse set of multimodal and unimodal tasks

80. MultiInstruct (arxiv 2022)

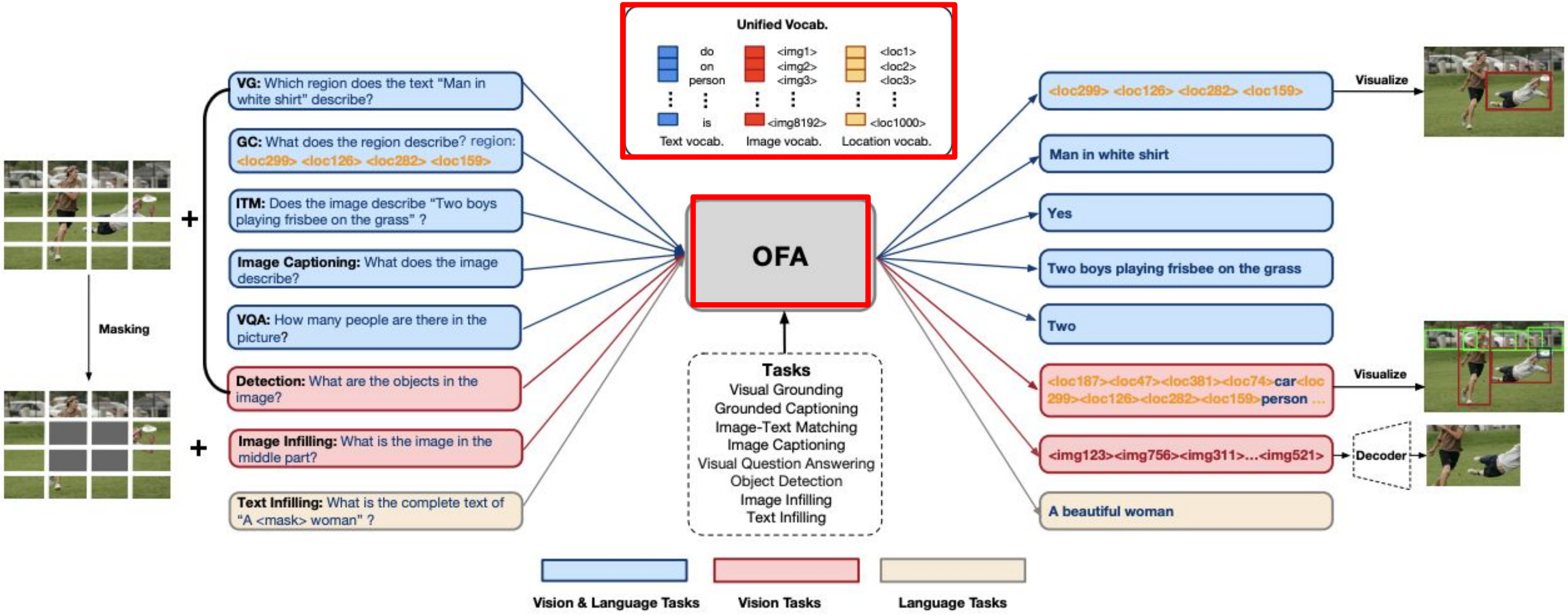
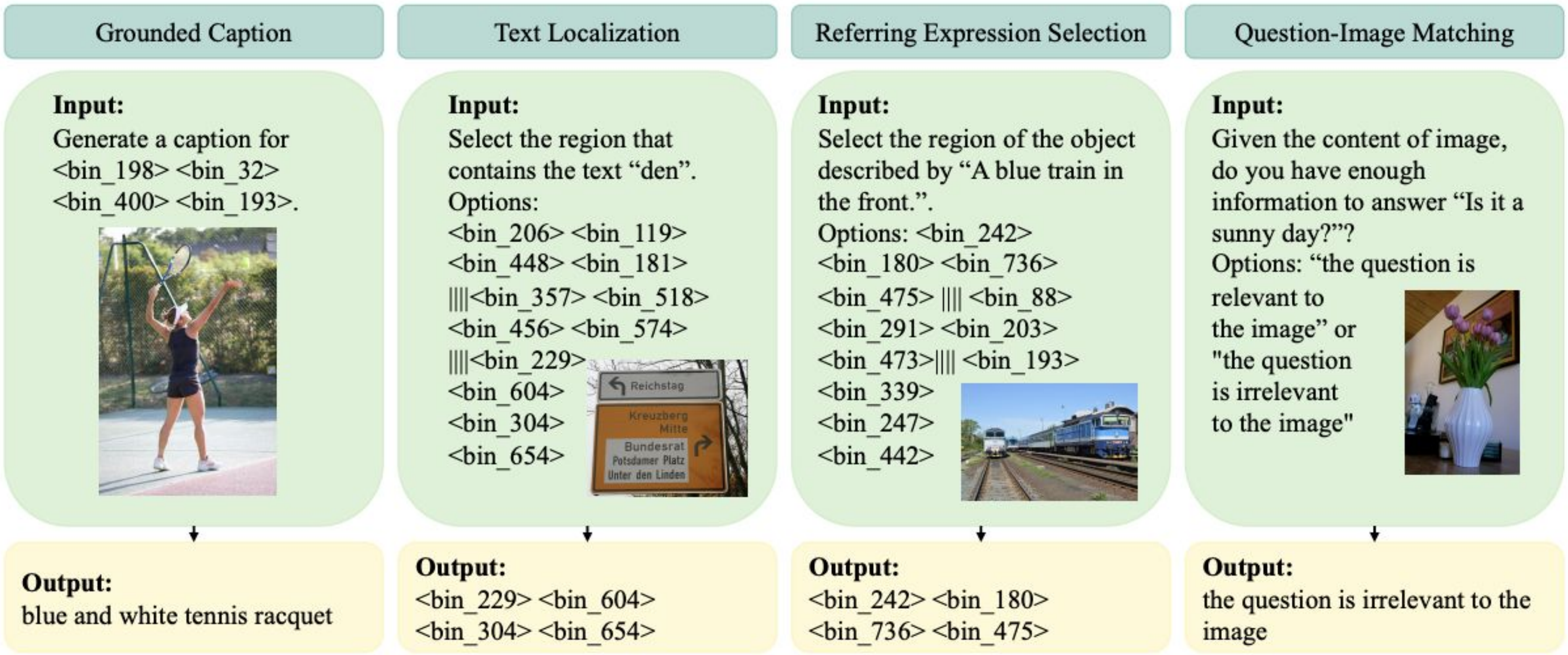

Unified Vocab.
do
on
person
is
Text vocab.
<img1>
<img2>
<img3>
<img8192>
Image vocab.
<loc1>
<loc2>
<loc3>
<loc1000>
Location vocab.
Masking
+
+
VG: Which region does the text "Man in white shirt" describe?
GC: What does the region describe? region: <loc299> <loc126> <loc282> <loc159>
ITM: Does the image describe "Two boys playing frisbee on the grass" ?
Image Captioning: What does the image describe?
VQA: How many people are there in the picture?
Detection: What are the objects in the image?
Image Infilling: What is the image in the middle part?
Text Infilling: What is the complete text of "A <mask> woman" ?
OFA
Tasks
Visual Grounding
Grounded Captioning
Image-Text Matching
Image Captioning
Visual Question Answering
Object Detection
Image Infilling
Text Infilling
<loc299> <loc126> <loc282> <loc159>
Visualize
Man in white shirt
Yes
Two boys playing frisbee on the grass
Two
<loc187><loc47><loc381><loc74>car<loc 299><loc126><loc282><loc159>person ...
Visualize
<img123><img756><img311>...<img521>
Decoder
A beautiful woman
Vision & Language Tasks
Vision Tasks
Language Tasks

Figure 2: A demonstration of the pretraining tasks, including visual grounding, grounded captioning, image-text matching, image captioning, VQA, object detection, image infilling as well as text infilling.

80. MultiInstruct (arxiv 2022)



Grounded Caption

Input:
Generate a caption for <bin_198> <bin_32> <bin_400> <bin_193>.

Output:
blue and white tennis racquet

Text Localization

Input:
Select the region that contains the text "den".
Options:
<bin_206> <bin_119> <bin_448> <bin_181>
||||<bin_357> <bin_518> <bin_456> <bin_574>
||||<bin_229> <bin_604> <bin_304> <bin_654>

Output:
<bin_229> <bin_604> <bin_304> <bin_654>

Referring Expression Selection

Input:
Select the region of the object described by "A blue train in the front.".
Options: <bin_242> <bin_180> <bin_736> <bin_475> |||| <bin_88> <bin_291> <bin_203> <bin_473>|||| <bin_193> <bin_339> <bin_247> <bin_442>

Output:
<bin_242> <bin_180> <bin_736> <bin_475>

Question-Image Matching

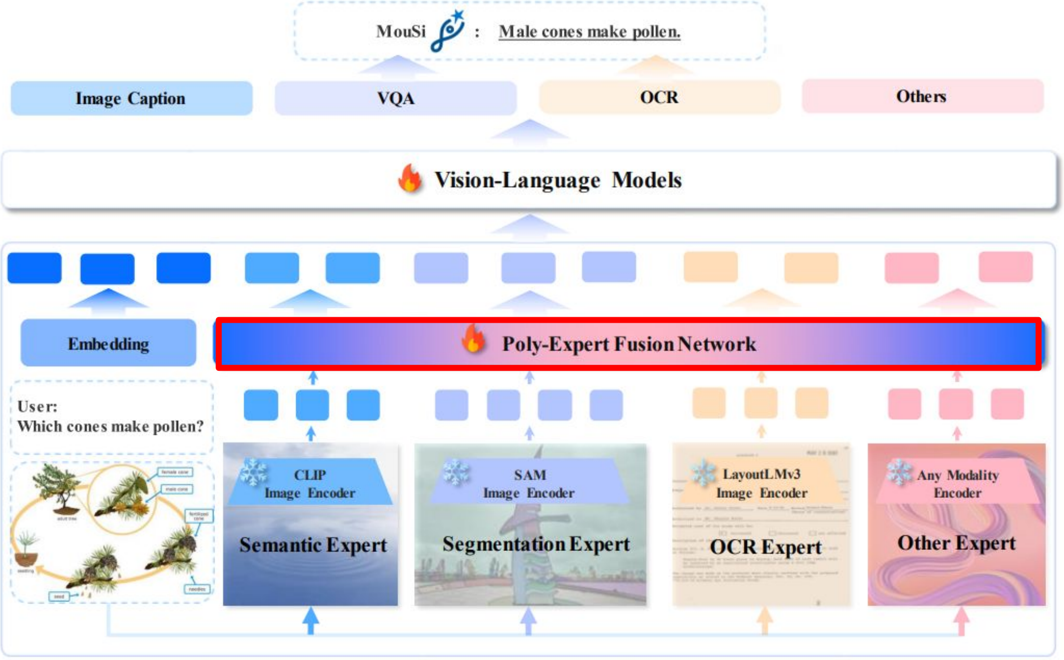
Input:
Given the content of image, do you have enough information to answer "Is it a sunny day?"?
Options: "the question is relevant to the image" or "the question is irrelevant to the image"

Output:
the question is irrelevant to the image

Figure 1: **Example Instances from MULTIINSTRUCT for Four Tasks.**

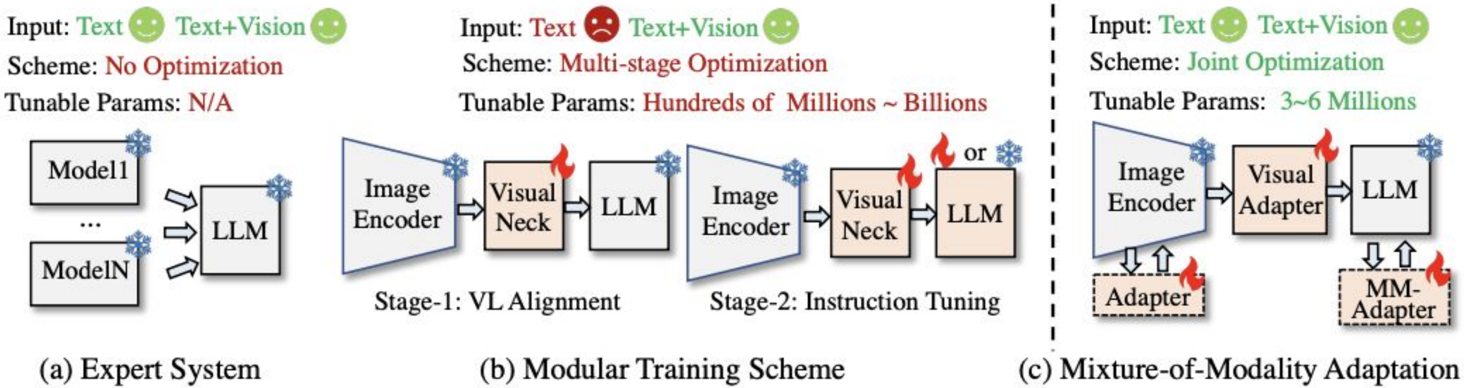
81. MouSi (COLM 2024)

- Incorporate **multiple visual experts** (e.g., CLIP and SAM) to VLMs
- **Poly-expert fusion network**
    - Combine their outputs (e.g., CLIP, SAM)
    - Interface with LLMs (e.g., Vicuna)



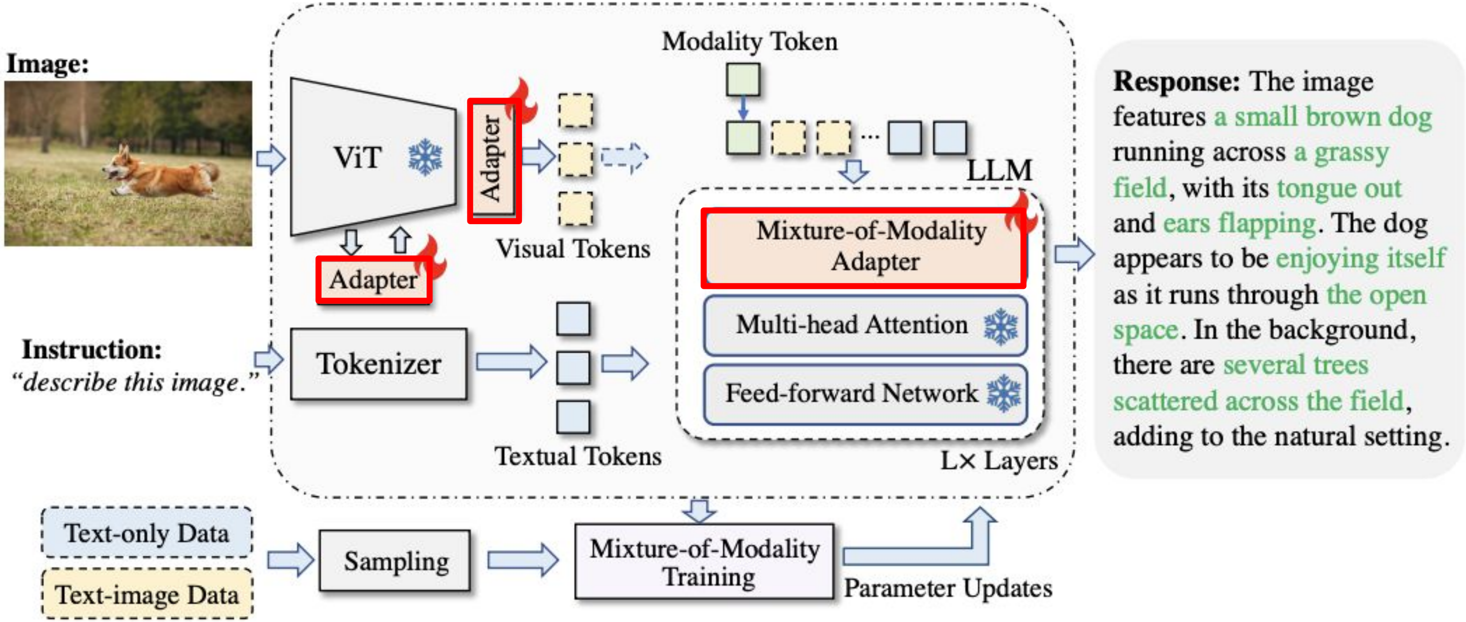
82. LaVIN (NeurIPS 2023)

- **Cost-effective** approach to VL instruction-tuning
- How? **Mixture-of-Modality Adapter (MM-Adapter)**
    - Significantly reduce # of trainable parameters
    - Adapt visual features to LLM



(a) Expert System
(b) Modular Training Scheme
(c) Mixture-of-Modality Adaptation

82. LaVIN (NeurIPS 2023)



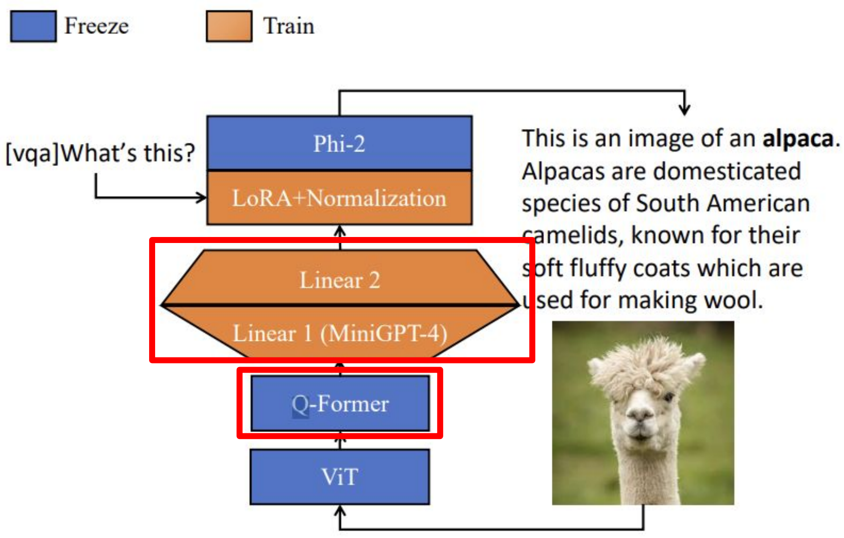
Response: The image features a small brown dog running across a grassy field, with its tongue out and ears flapping. The dog appears to be enjoying itself as it runs through the open space. In the background, there are several trees scattered across the field, adding to the natural setting.

Figure 2: The overview of the Mixture-of-Modality Adaptation (MMA) and the architecture of LaVIN. In LaVIN, the novel Mixture-of-Modality Adapters are employed to process the instructions of different modalities. During instruction tuning, LaVIN is optimized by Mixture of Modality Training (MMT) in an end-to-end manner.

83. TinyGPT-V (ICMLW 2024)

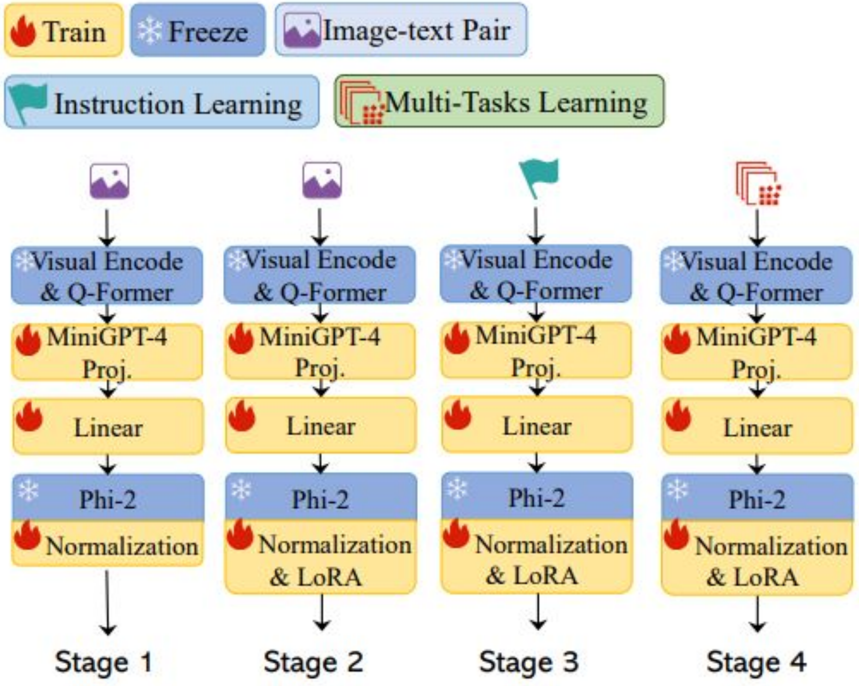
- Combining
  - (1) Vision: compact **EVA-ViT**
    - with **linear projection layers (+Q-Former layer)**
  - (2) LLM: powerful **Phi-2**

83. TinyGPT-V (ICMLW 2024)



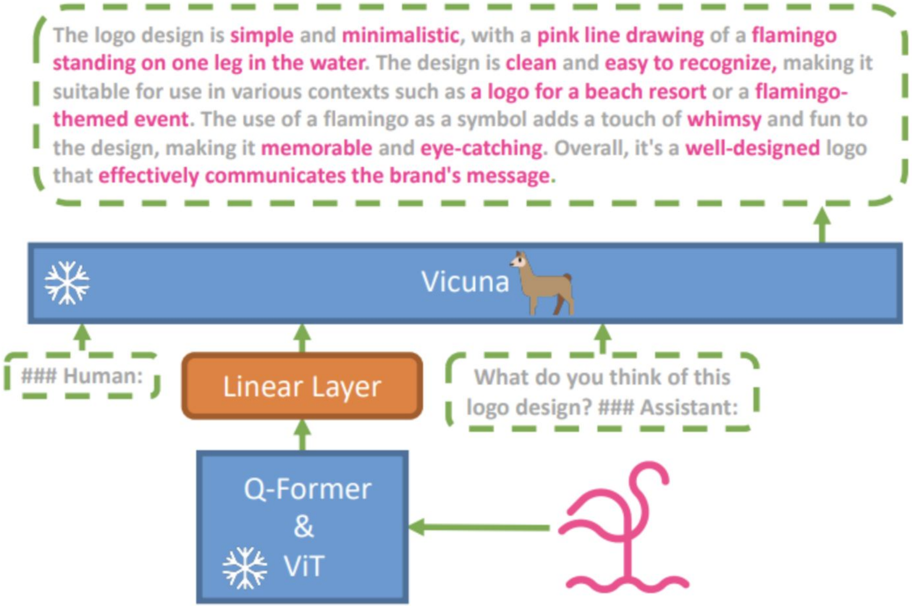
This is an image of an **alpaca**. Alpacas are domesticated species of South American camelids, known for their soft fluffy coats which are used for making wool.

Figure 4: Architecture of TinyGPT-V. The model takes a visual backbone, which remains frozen during all training phases. We concatenate Q-Former layer visual output tokens from ViT backbone and project them into Phi-2 language model space via two linear projection layers.



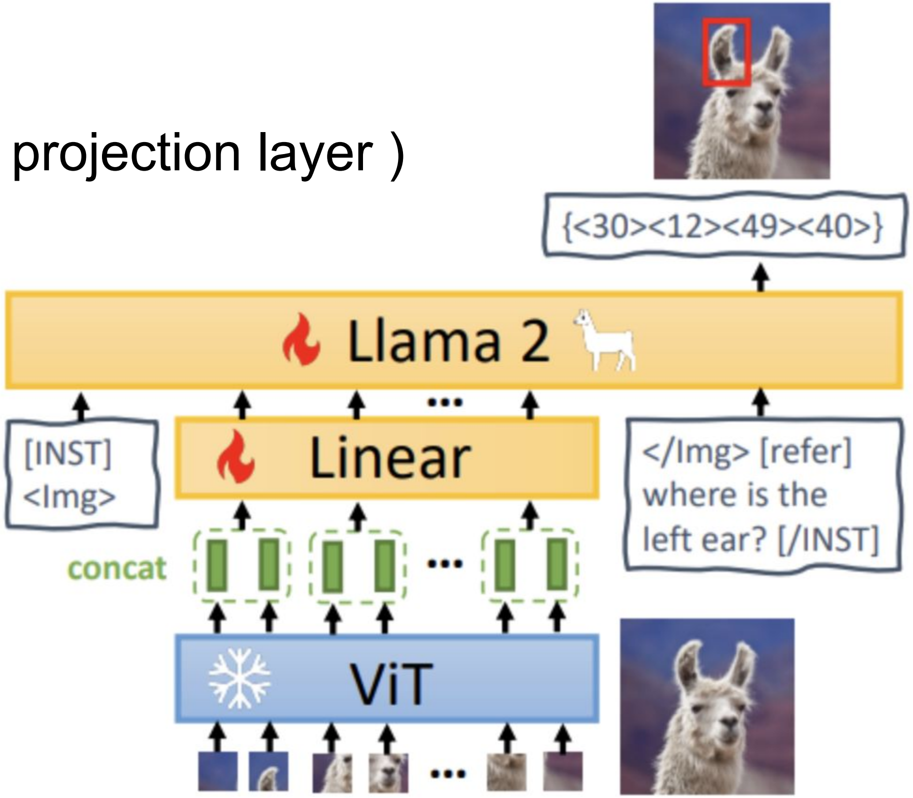
84. MiniGPT-4 (ICLR 2024)

- Architecture
  - (1) Vision: **Pretrained ViT + Q-Former (+ linear proj)**
  - (2) LLM: Vicuna


The logo design is simple and minimalistic, with a pink line drawing of a flamingo standing on one leg in the water. The design is clean and easy to recognize, making it suitable for use in various contexts such as a logo for a beach resort or a flamingo-themed event. The use of a flamingo as a symbol adds a touch of whimsy and fun to the design, making it memorable and eye-catching. Overall, it's a well-designed logo that effectively communicates the brand's message.
Vicuna
### Human:
Linear Layer
What do you think of this logo design? ### Assistant:
Q-Former
&
ViT

85. MiniGPT-v2 (ICLR 2024)

- Architecture
  - (1) Vision: static ViT (+ linear projection layer)
  - (2) LLM: LLaMA-2-chat



86. CoVLM (ICLR 2024)

- Propose **communication tokens**
- To enable dynamic interaction btw…
  - (1) CLIP ViT-L image encoder
  - (2) YOLOX detection network
  - (3) Pythia LLM

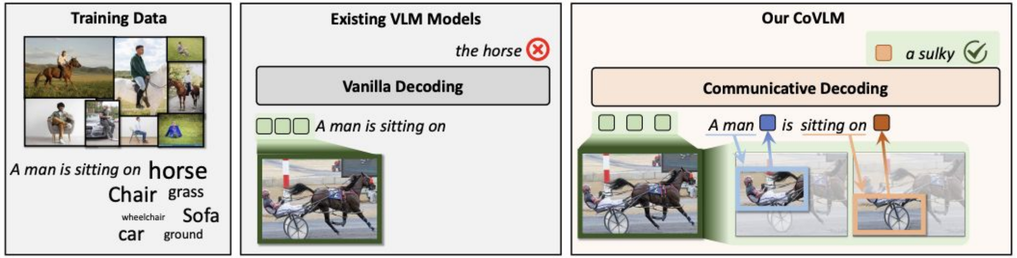


Figure 1: Comparison with existing VLMs. Previous models take in a whole image as input, impairing the compositionality of VLMs. Our CoVLM inserts communication tokens into the LLM after visual entities / relationships to enable the language-to-vision and vision-to-language communication, improving compositionality to a large extent.

86. CoVLM (ICLR 2024)

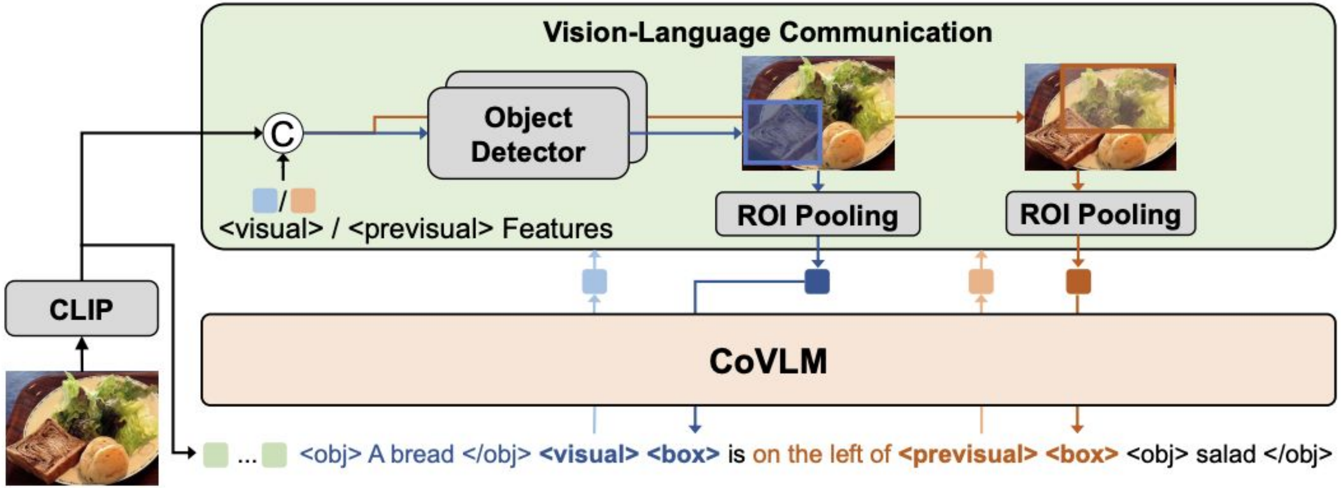
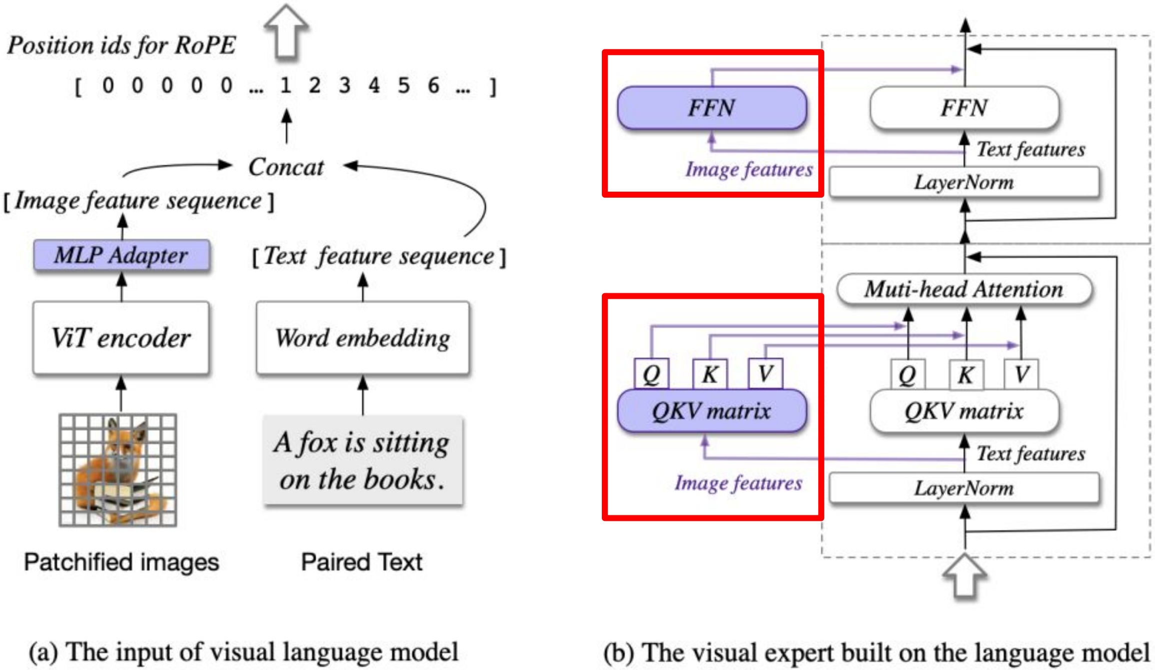

Vision-Language Communication
C
Object Detector
<visual> / <previsual> Features
ROI Pooling
ROI Pooling
CLIP
CoVLM
... <obj> A bread </obj> <visual> <box> is on the left of <previsual> <box> <obj> salad </obj>

Figure 2: Overview of our CoVLM framework. Our vision module consists of a CLIP encoder to encode the image, and an object detector which takes in the image together with language inputs to generate relevant regions. For language modelling, we insert a set of communication tokens into the LLM, which can appear after a visual entity with a `<visual>` token or after a relationship with a `<previsual>` token. The last hidden layer of the LLM is then sent to the object detector to propose regions relevant to the language inputs so far. This is termed as top down language-to-vision communication. Next, in vision-to-language communication, the features of the proposed regions are fed back to LLM via `<box>` or `<prebox>` token for further language generation.

87. CogVLM (NeurIPS 2024)

- Enhance pretrained LM with **Visual Expert** module
    - Employ a visual expert module in each layer that leverages **QKV matrix & MLP**
- Components
    - (1) ViT encoder
    - (2) LLM
    - MLP adapter to (1)->(2)

87. CogVLM (NeurIPS 2024)



(a) The input of visual language model

(b) The visual expert built on the language model

Figure 3: **The architecture of CogVLM.** (a) The illustration about the input, where an image is processed by a pretrained ViT and mapped into the same space as the text features. (b) The Transformer block in the language model. The image features have a different QKV matrix and FFN. Only the purple parts are trainable.

88. CogVLM2 (arxiv 2024)

- VLM for **image and video**
- Adapter: Incorporates …
  - Conv for downsampling
  - SwiGLU module

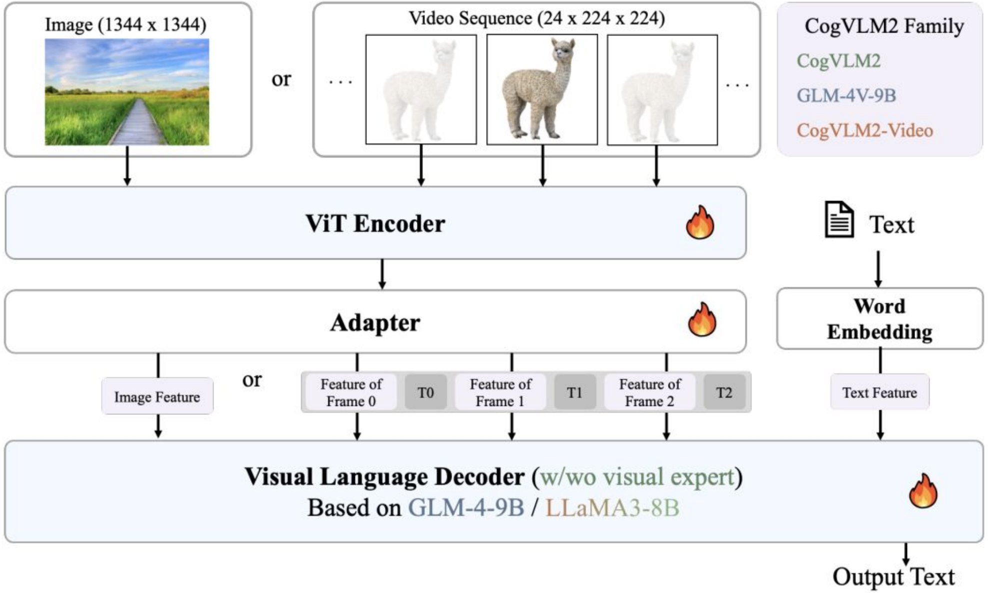
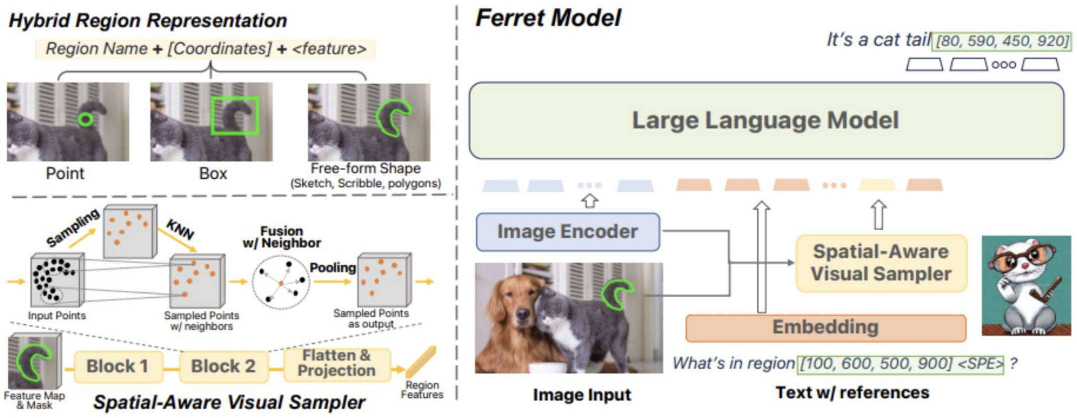


Figure 2: The architecture of the CogVLM Family. Taking a high resolution image or the extracted frames from a given video, CogVLM models embed visual information with a pre-trained ViT Encoder and an Adapter. The embedded visual features are sent to a Visual Language Decoder. CogVLM2-Video is capable of answering image-related and video-related queries.

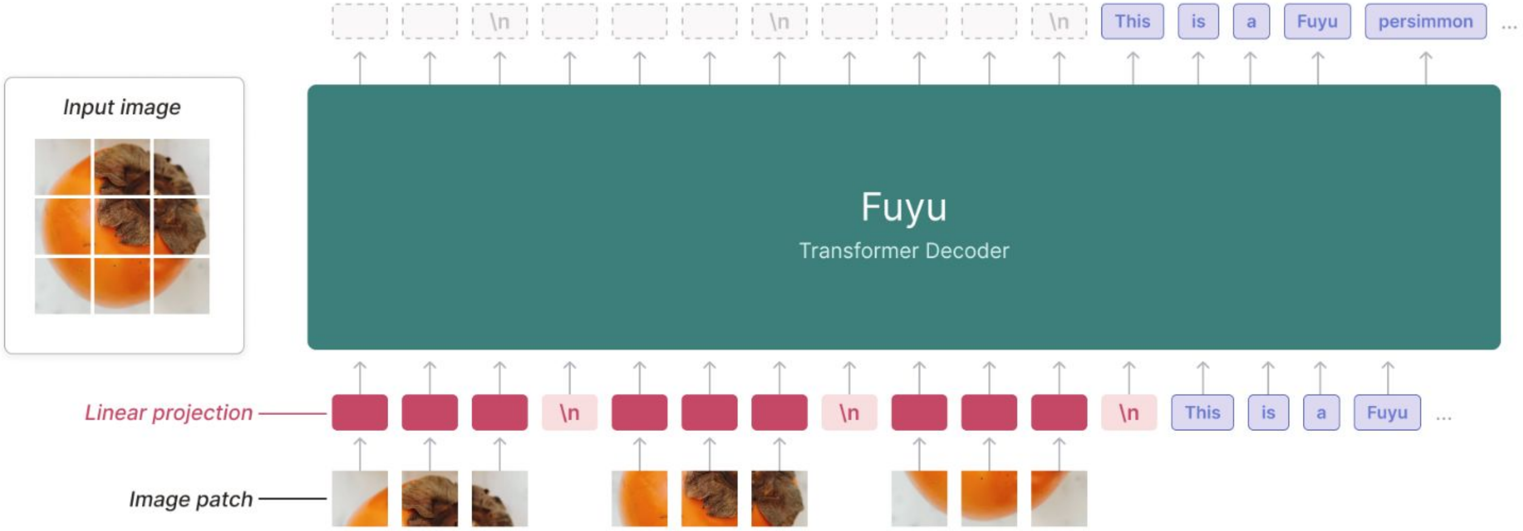
89. Ferret (ICLR 2024)

- **Hybrid** region representation
    - Discrete coordinates
    - Continuous features
- Allow it to precisely pinpoint objects & regions within images



90. Fuyu-8B (blog 2023)

- **Directly** projecting images patches into (decoder-only) LLM
- Treat **image & text tokens uniformly**
- Efficient!



91. OtterHD (arxiv 2023)

- Inspired by Fuyu-8B
- **Directly integrates pixel-level** info into LLM using position embedding
  - Separate vision encoding X
  - Varying image sizes O
  - High-resolution images up to 1024x1024

92. GLaMM (CVPR 2024)

- Pixel-level grounding
- 5 component architecture
    - Global & Regional image encoders
    - LLM
    - Grounding Image encoder
    - Pixel decoder

COSMO: COntrastive Streamlined MultimOdal Model with Interleaved Pre-Training

93. COSMO (arxiv 2024)

- ViT + (partitioned) LLM
  - LLM half: **unimodal**
  - LLM half: **mulitmodal**
- Interleaved data sequences
- Pretraining: LM loss + CL loss

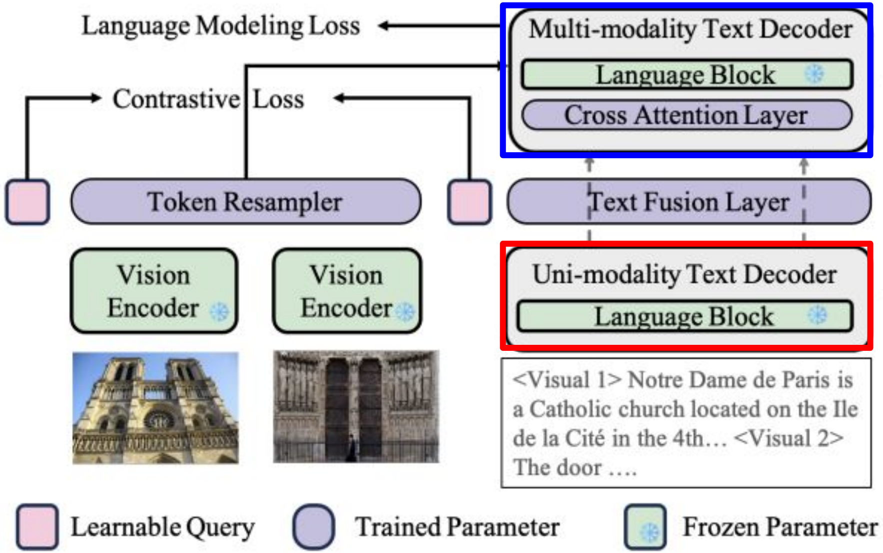
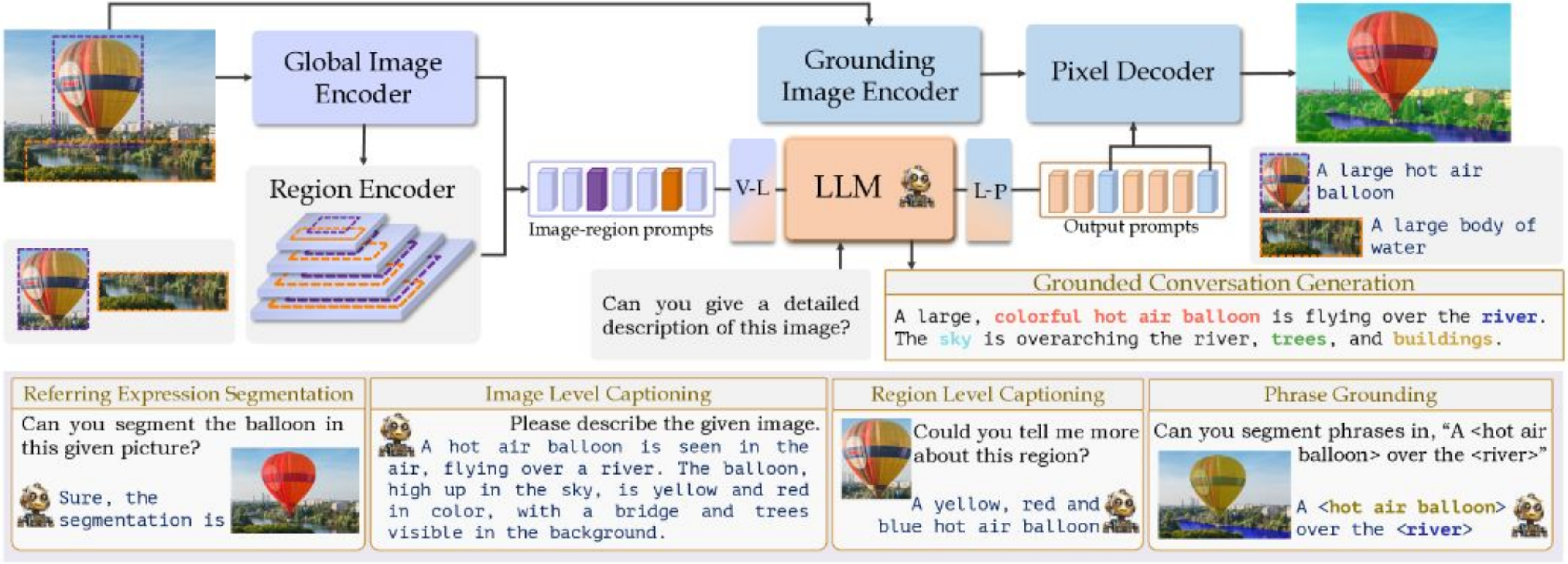


Figure 3. **An introduction to CosMo**: This model handles both image/video text pairs and inter-level image/video text pairs. The Large Language Model is divided into two parts to compute contrastive loss and language modeling loss.

COSMO: COntrastive Streamlined MultimOdal Model with Interleaved Pre-Training

93. COSMO (arxiv 2024)



Can you give a detailed description of this image?

A large, colorful hot air balloon is flying over the river. The sky is overarching the river, trees, and buildings.

Can you segment the balloon in this given picture?

Sure, the segmentation is

Please describe the given image.

A hot air balloon is seen in the air, flying over a river. The balloon, high up in the sky, is yellow and red in color, with a bridge and trees visible in the background.

Could you tell me more about this region?

A yellow, red and blue hot air balloon

Can you segment phrases in, "A <hot air balloon> over the <river>"

A <hot air balloon> over the <river>

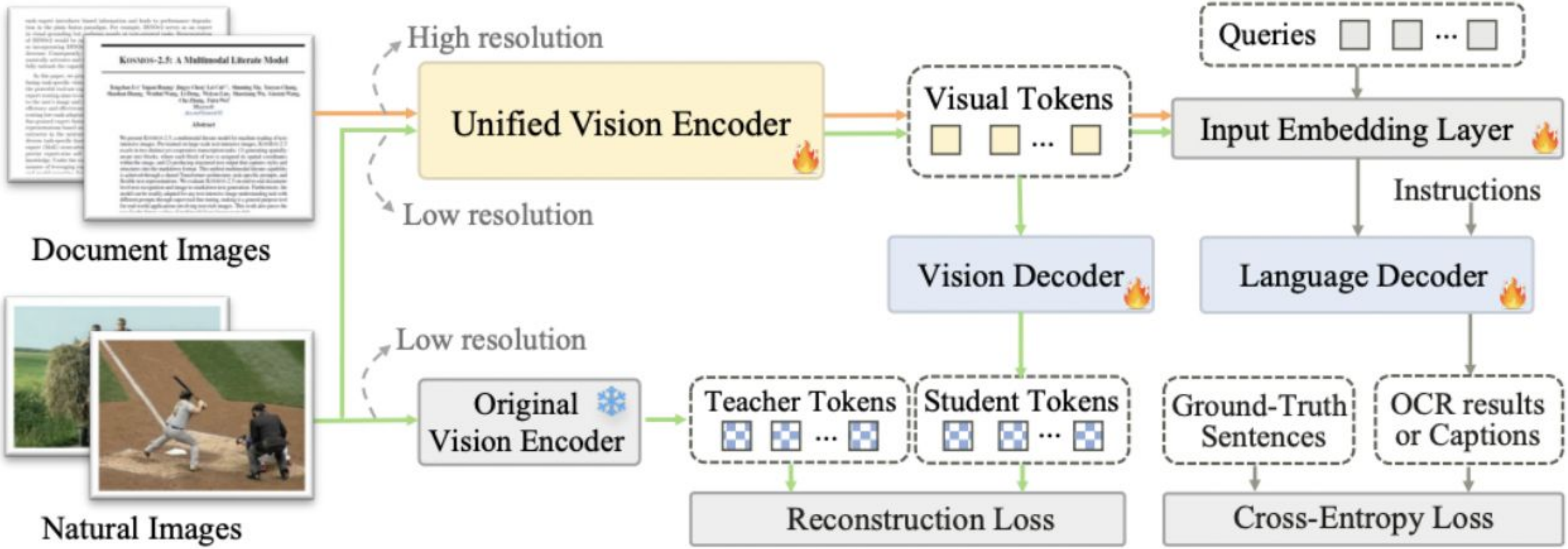
94. UNIT (NeurIPS 2024)

- Vision Encoder (ViT)에 **Text Recognition 능력**을 심어주자!

  => **Single** (Vision) Model로써 전부 처리 가능!

- Two components (both lightweight)
  - (1) **Language** decoder: Text 생성 위해
  - (2) **Vision** decoder: Visual 능력 손실 방지

94. UNIT (NeurIPS 2024)


Document Images
Natural Images
High resolution
Low resolution
Low resolution
Unified Vision Encoder
Original Vision Encoder
Visual Tokens
Queries
Input Embedding Layer
Instructions
Vision Decoder
Language Decoder
Teacher Tokens
Student Tokens
Ground-Truth Sentences
OCR results or Captions
Reconstruction Loss
Cross-Entropy Loss