Efficient Visual Understanding and Interaction with VLMs

Wentong LI (李文通)

Homepage: https://cslwt.github.io/

Associate Professor@NUAA; PhD@ZJU

2025/08/09

Content

1. Fine-grained Object & Region Understanding
   - Image/Video
2. Efficient VLMs via Visual Token Compression
   - Model-driven/Data-driven
3. Streaming Understanding & Interaction for AI Assistant
   - Training/Training-free

Content

1. Fine-grained Object & Region Understanding

- Image/Video

2. Efficient VLMs via Visual Token Compression

- Model-driven/Data-driven

Fine-grained Object/Region Understanding

Osprey

Object Category: person

Part Taxonomy: body

Attribute: color, position …

Caption: region short / detailed description

SAM "Segment Everything" Predictions

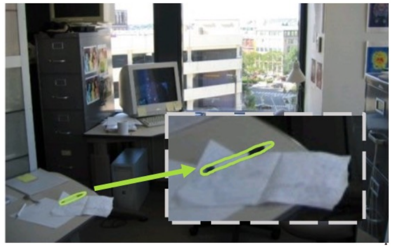
Fine-grained Region/Pixel Understanding

No semantic information

Rich semantic information containing different granularities

- Integrate images, target regions (masks), and textural data;
- Enable fine-grained semantic description of arbitrary regions or objects within images;
- Strong robustness and generalization.

General scene

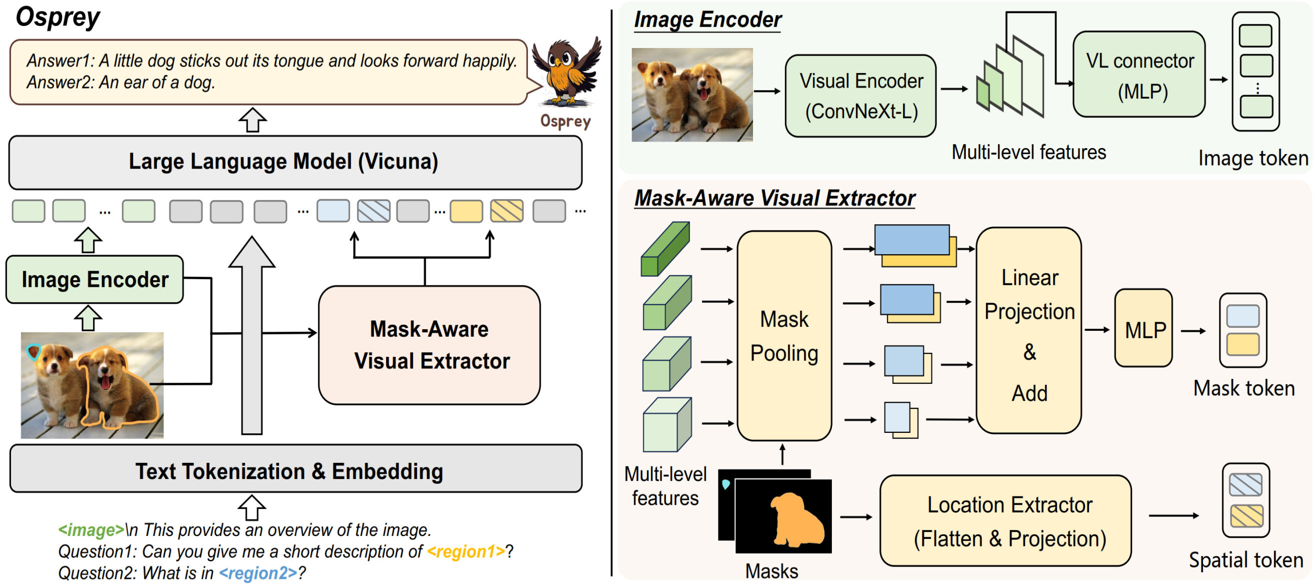


Out-of-domain Scene

Osprey: Pixel Understanding with Visual Instruction Tuning, in *CVPR 2024*.

2023.9-2023.11

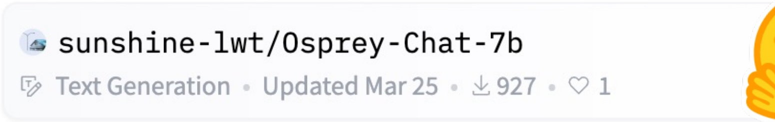
Fine-grained Object/Region Understanding



- Support high-resolution image
  - ConvNeXt (512x521@training, 800x800@inference)
- Pixel-level region feature extraction
  - Mask-Aware visual extractor (multi-level)

- 720K region-text pairs.
- Six types of object region-text data.

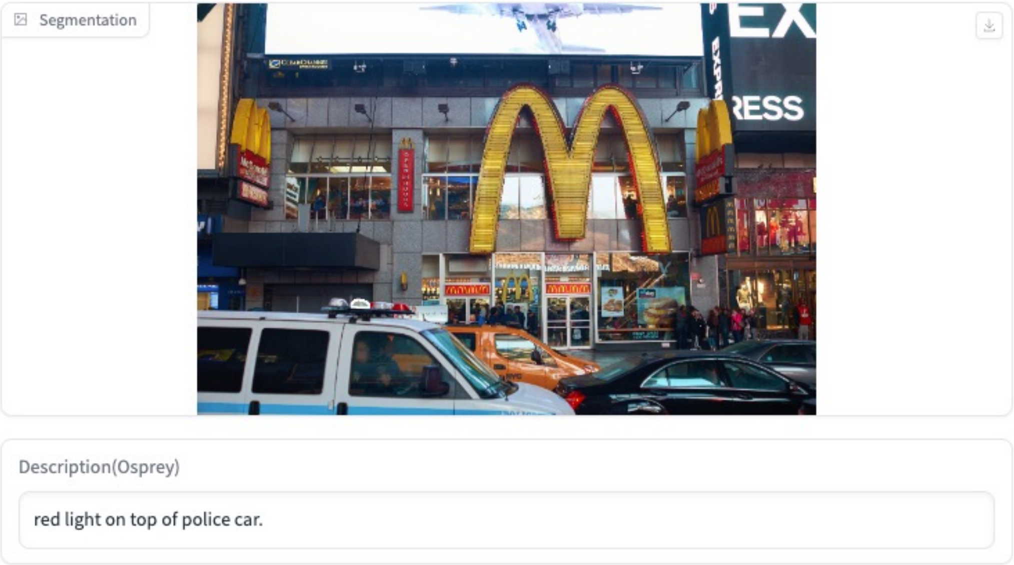
Open-source: https://huggingface.co/sunshine-lwt

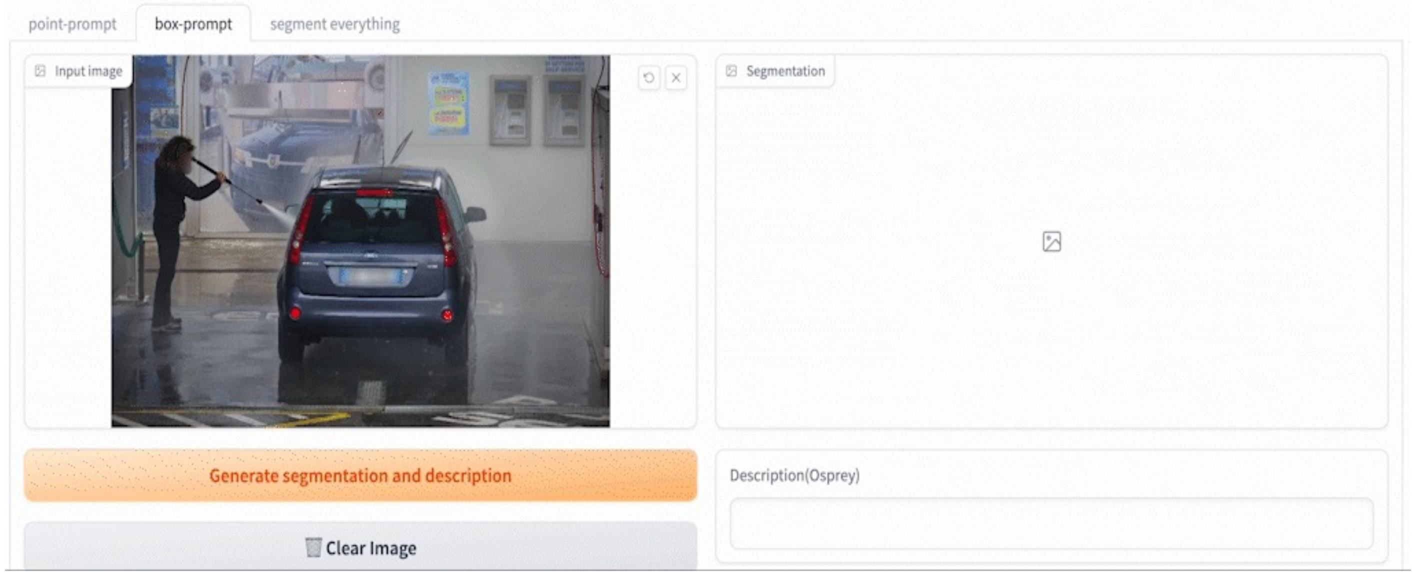


[1] ConvLLaVA: Hierarchical Backbones as Visual Encoder for Large Multimodal Models, *arXiv:2405.15738.* **Tsinghua Uni.**
[2] Mini-gemini: Mining the potential of multi-modality vision language models, *arXiv:2403.18814.* **CUHK**

Fine-grained Object/Region Understanding

Online Demo





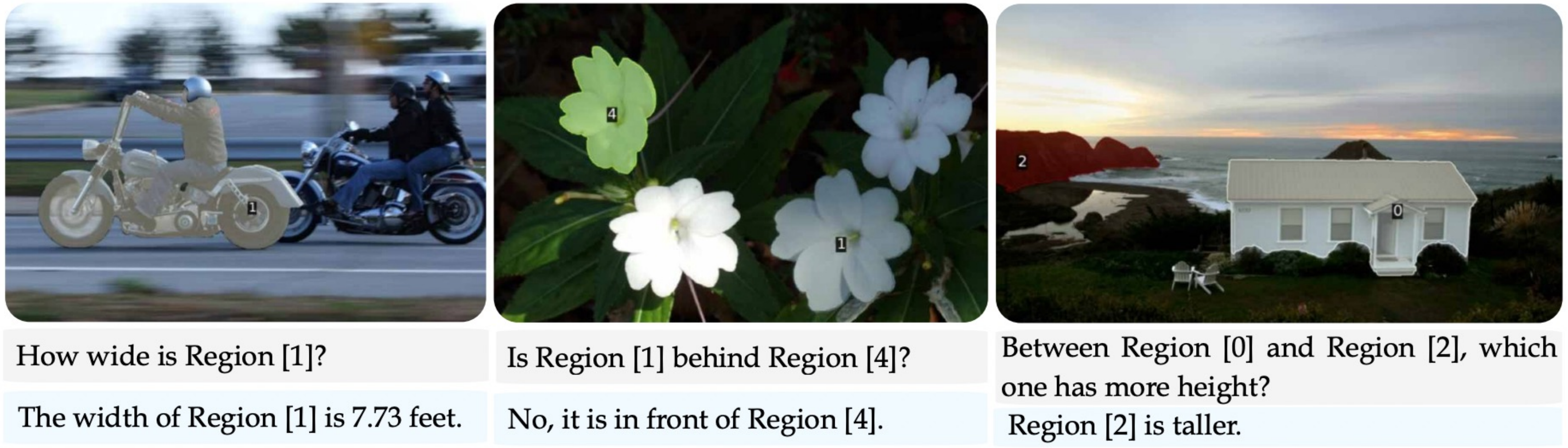
Video Visualization

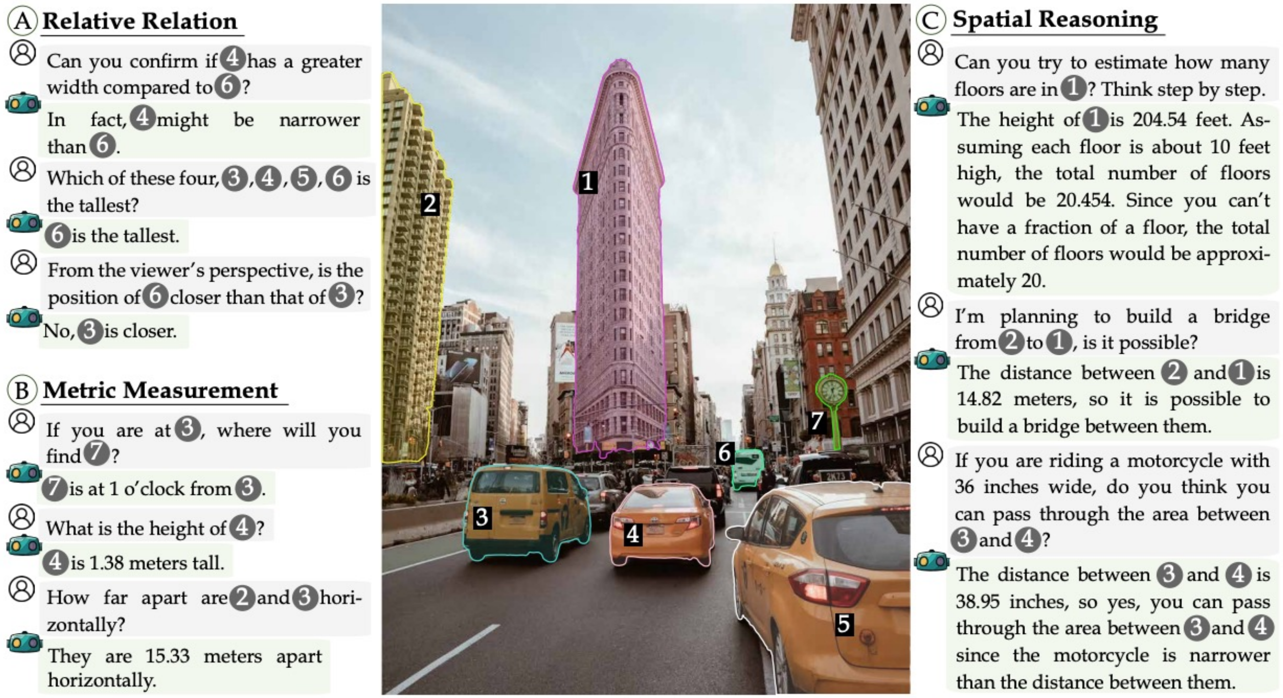
Fine-grained Object/Region Understanding

Spatial RGPT

Accurate Spatial Understanding & Reasoning on Objects in Image.

- Relative Relation
- Metric Measurement
- Spatial Reasoning



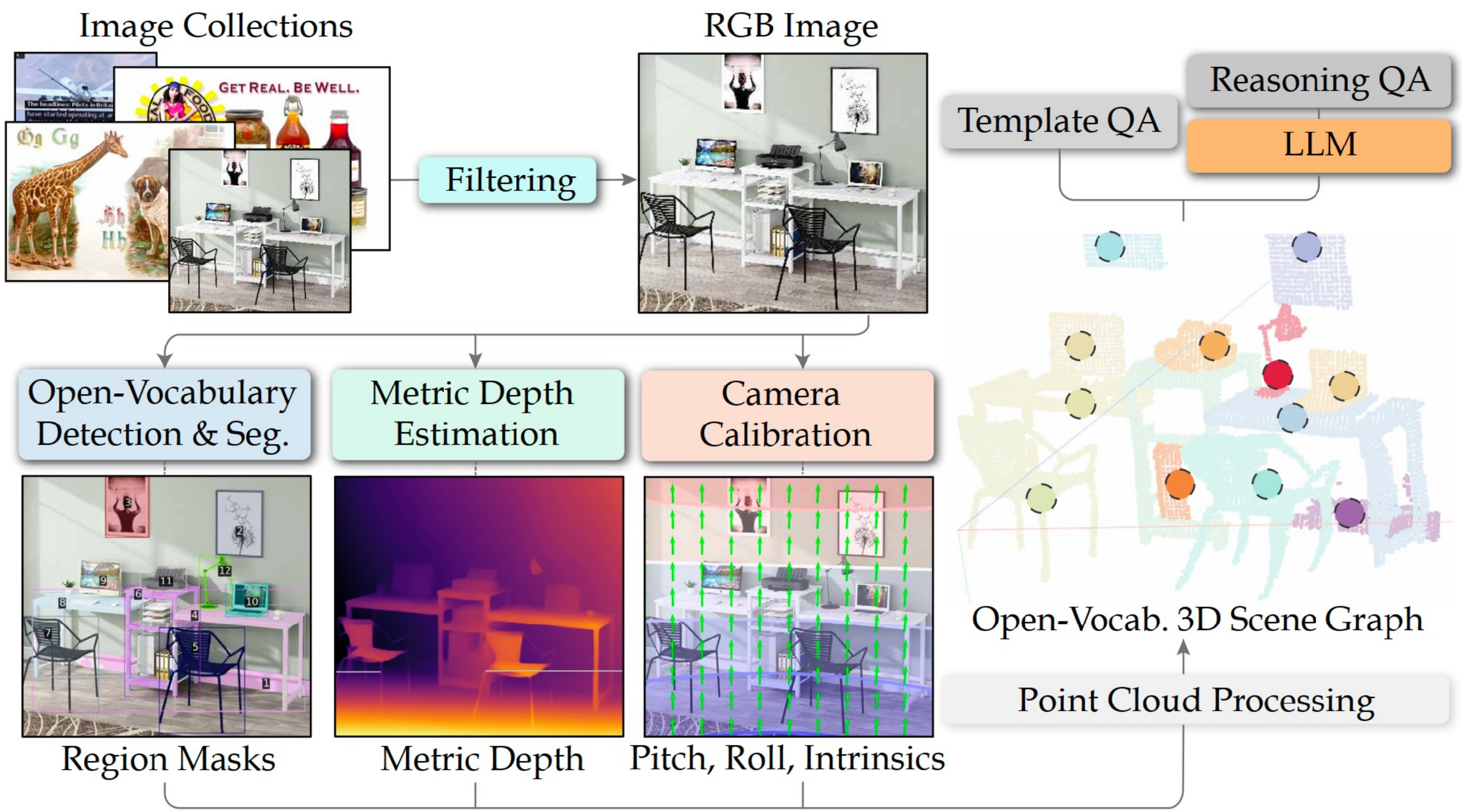


SpatialRGPT: Grounded Spatial Reasoning in Vision-Language Models, in *NeurIPS2024*.

UCSD&NVIDA

Fine-grained Object/Region Understanding

3D scene graph construction from Single 2D image



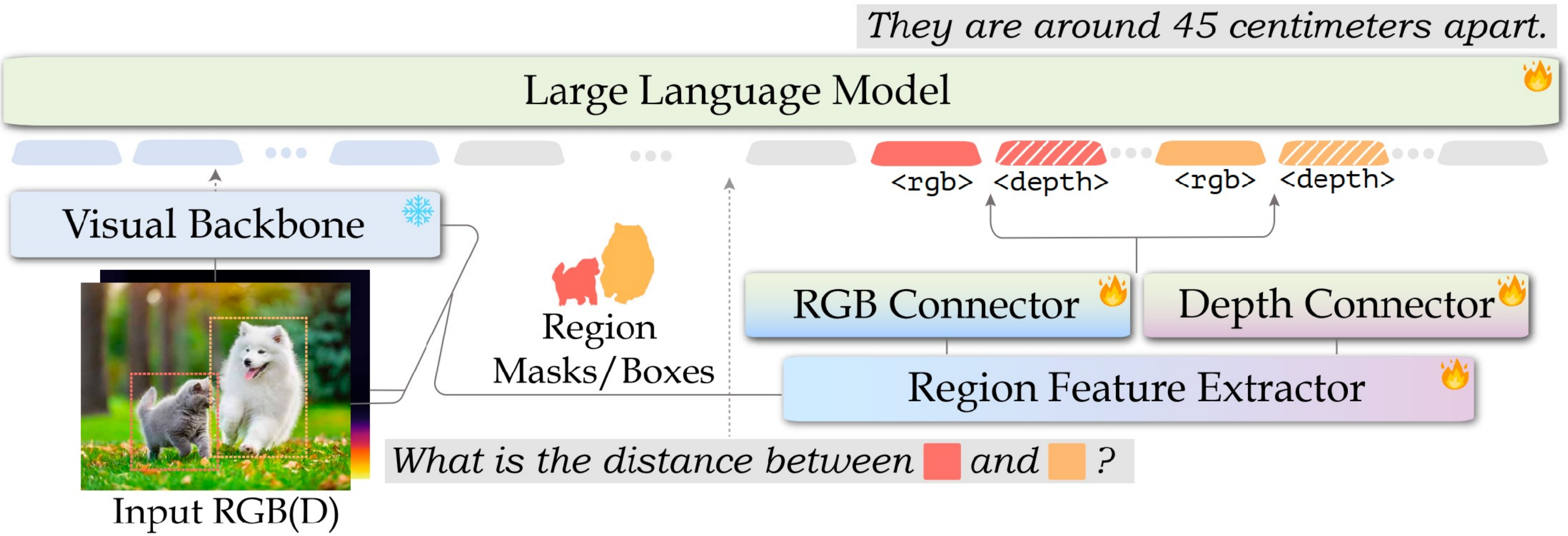
- Open-Voc. Detection &Segmentation: Tagging model, Grounding DINO,SAM-HQ
- Metric Depth Estimation: Metric3Dv2
- Camera Calibration: WildCamera: camera intrinsic PerspectiveFields: camera extrinsics
- 3D Scene Graph Construction

SpatialRGPT: Grounded Spatial Reasoning in Vision-Language Models, in *NeurIPS2024*.

UCSD&NVIDA

Fine-grained Object/Region Understanding

Framework



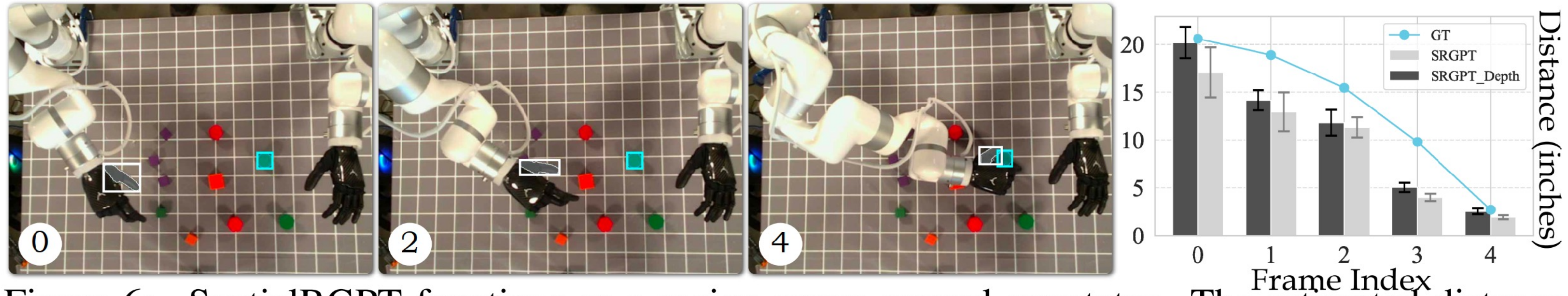
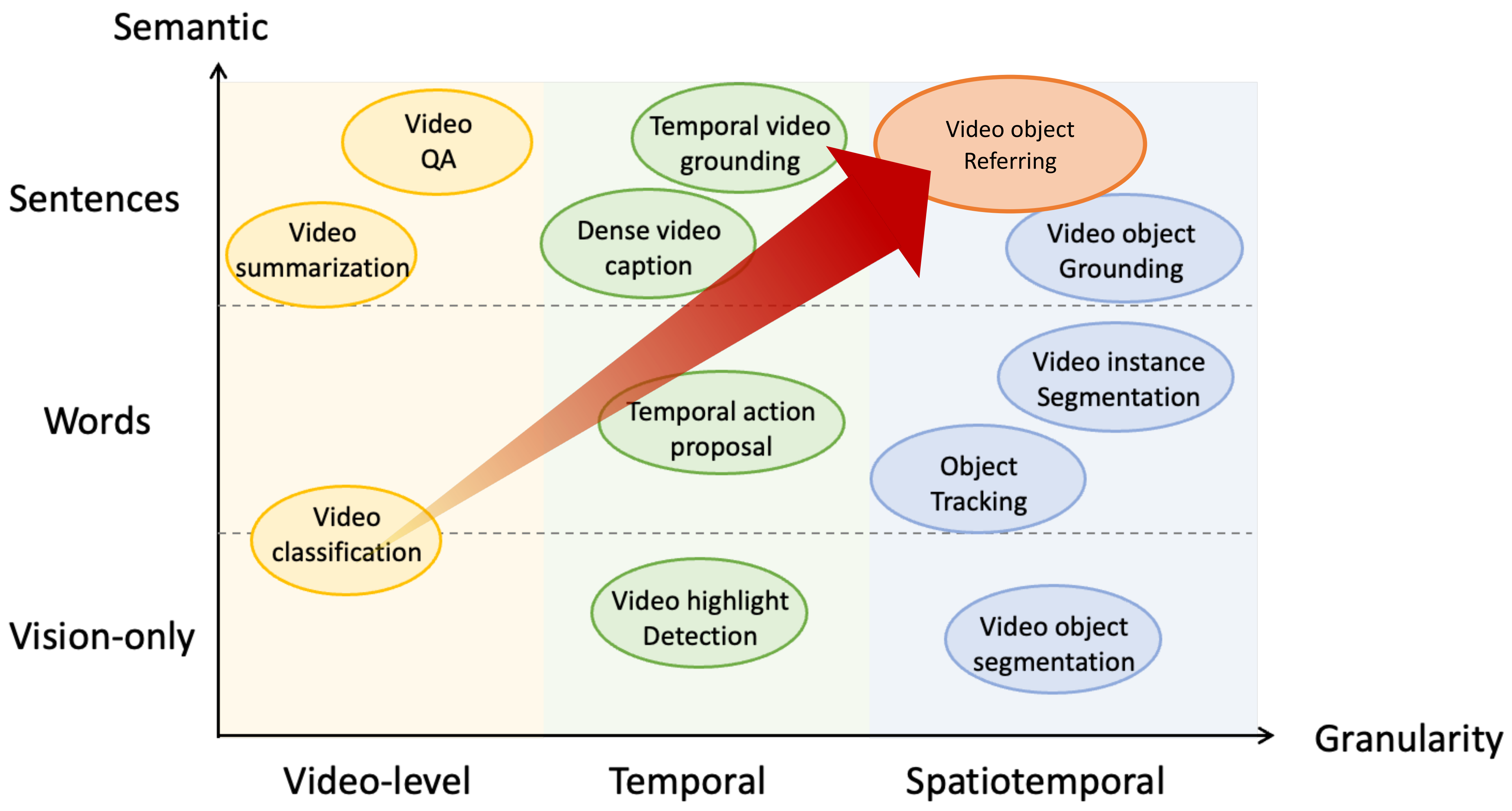


Figure 6: SpatialRGPT functions as a region-aware reward annotator. The estimated distance decreased monotonically as the fingertip moves towards the target.

SpatialRGPT: Grounded Spatial Reasoning in Vision-Language Models, in *NeurIPS2024*.

UCSD&NVIDA

Fine-grained Object/Region Understanding



Fine-grained Object/Region Understanding

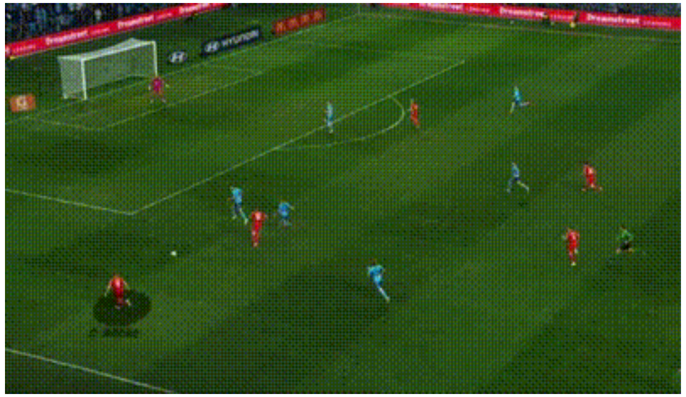
Video Object Referring

A man with a cocked hat and green robes, riding a horse, slowly riding from the left to the right.

Video Objects Relationship

The knife <object1> moves the spring onions from the chopping board <object2> to the pan.

Future Reasoning



Q: What will <object1> probably do next?

A: <object1> will probably have to shoot or pass the ball to a teammate.

Video Object Retrieval

Input image

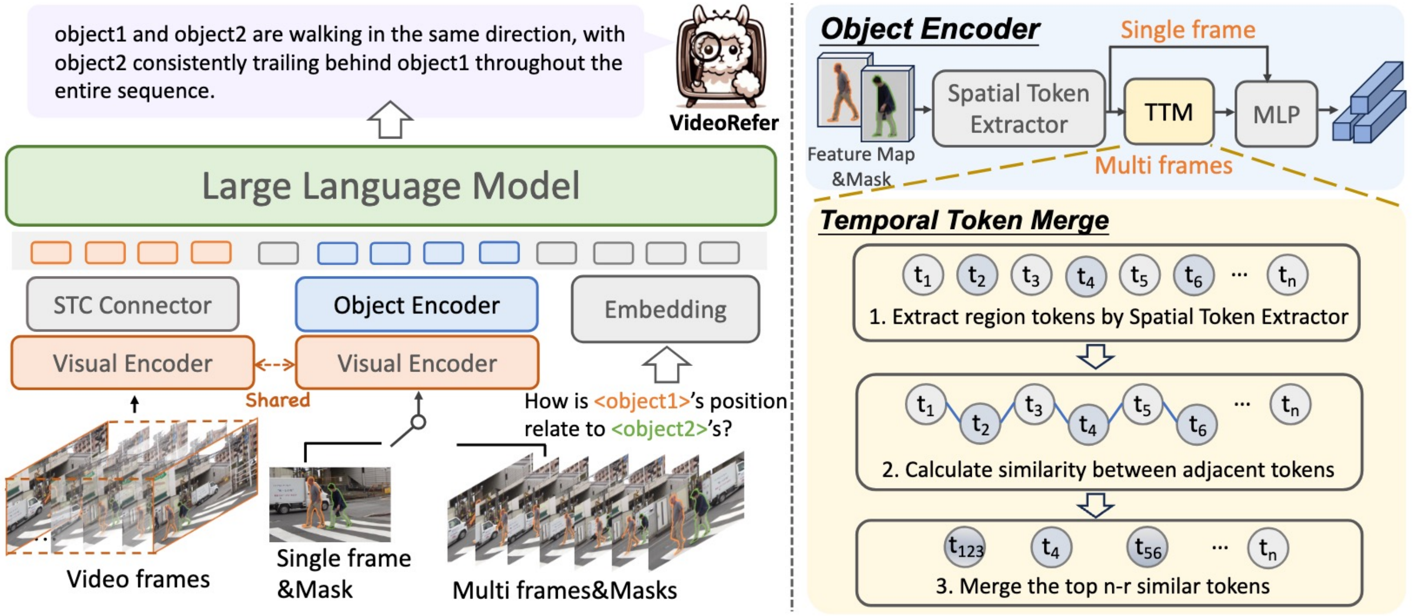
The man was Trump, who stood in the crowd waving and waving his fist to the left and right.

VideoRefer Suite: Advancing spatial-temporal object understanding with video LLM, in *CVPR 2025*.

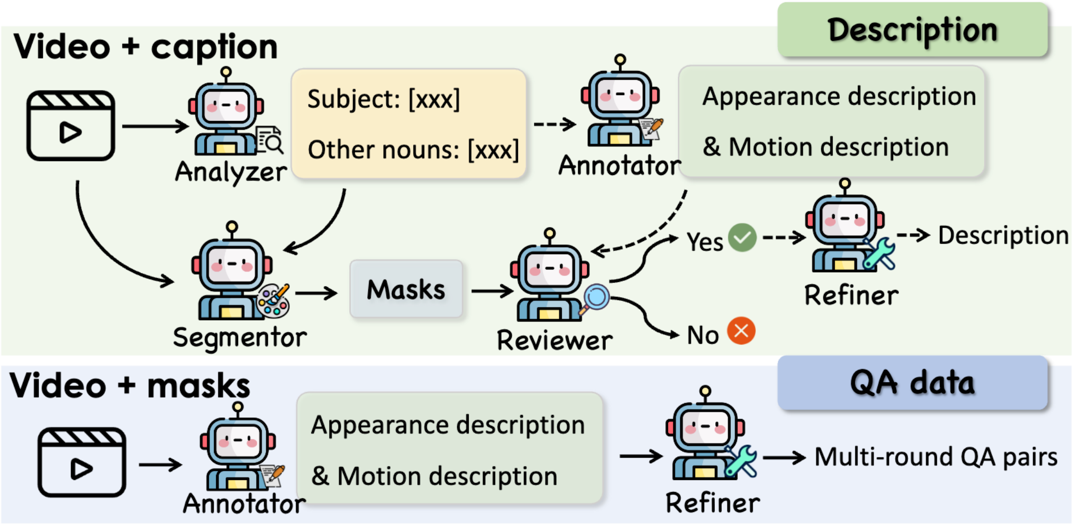
2024.6-2024.11

Fine-grained Object/Region Understanding

VideoRefer Suite

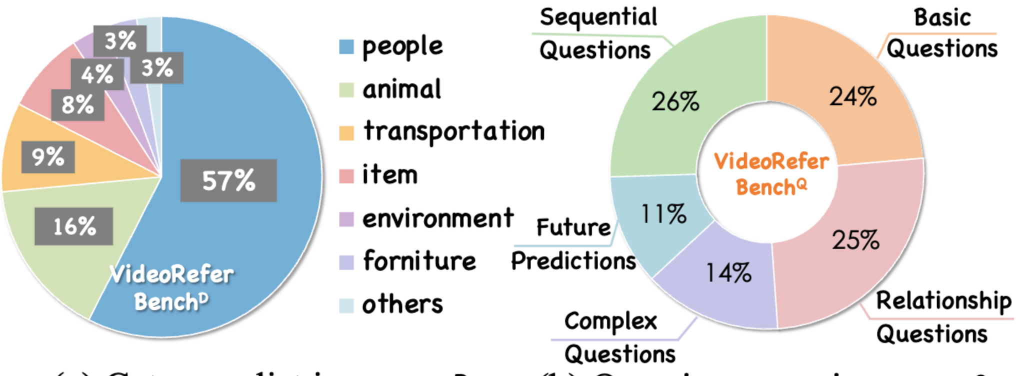


VideoRefer Model



VideoRefer-700K—Multi-agent Data Engine

- Spatiotemporal Region-level understanding Architecture;
- Constructing Large-scale Video Region Dataset;
- Evaluation Benchmarks for Video-based Object Understanding.



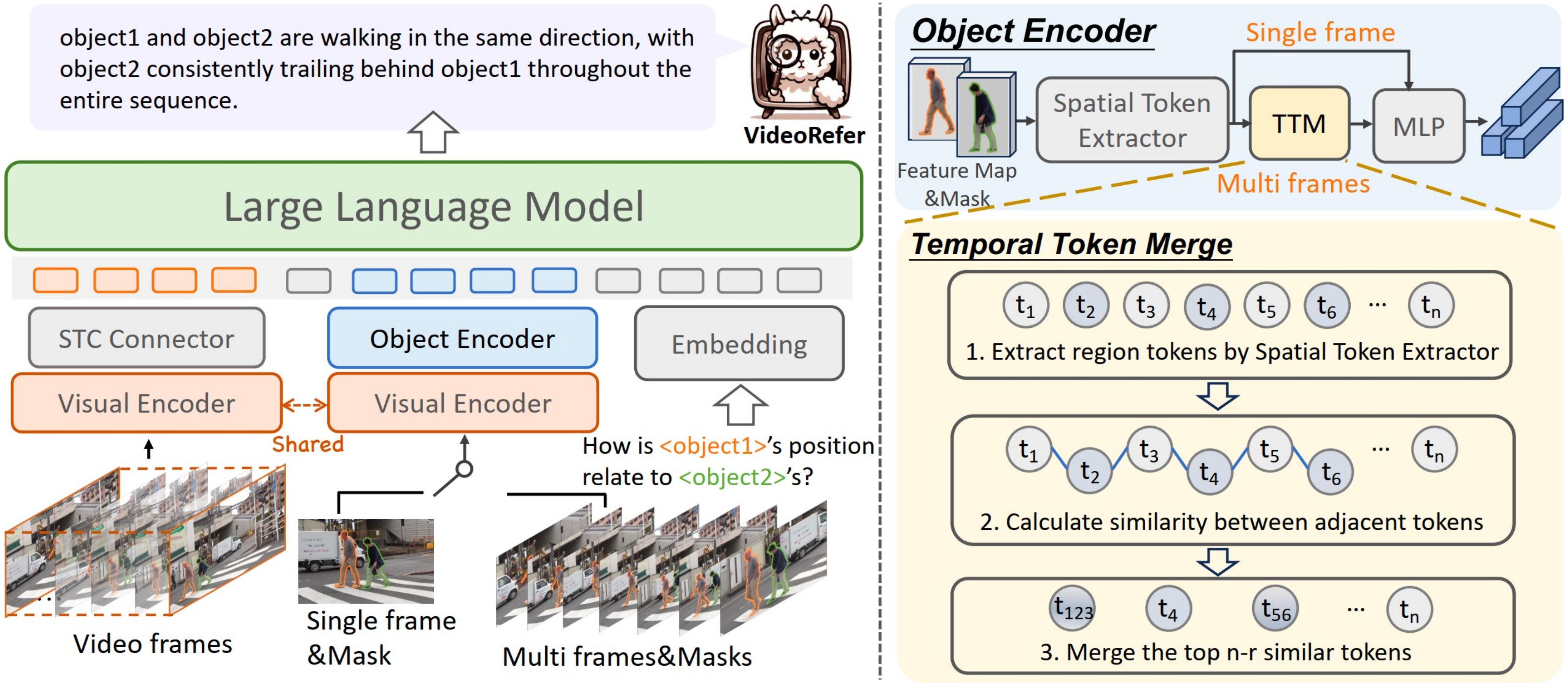
(a) Category list in Bench^D (b) Question types in Bench^Q

VideoRefer-Bench

VideoRefer Suite: Advancing spatial-temporal object understanding with video LLM, in *CVPR 2025*. **2024.6-2024.11**

Fine-grained Object/Region Understanding

VideoRefer Model



Base Model: VideoLLaMA2

A plug-and-play Spatial-Temporal Object Encoder:

- Spatial Token Extractor (*Single-frame*)
- Temporal Token Merge Module (*Multi-frame*)
- Free-from input region (*Mask*)

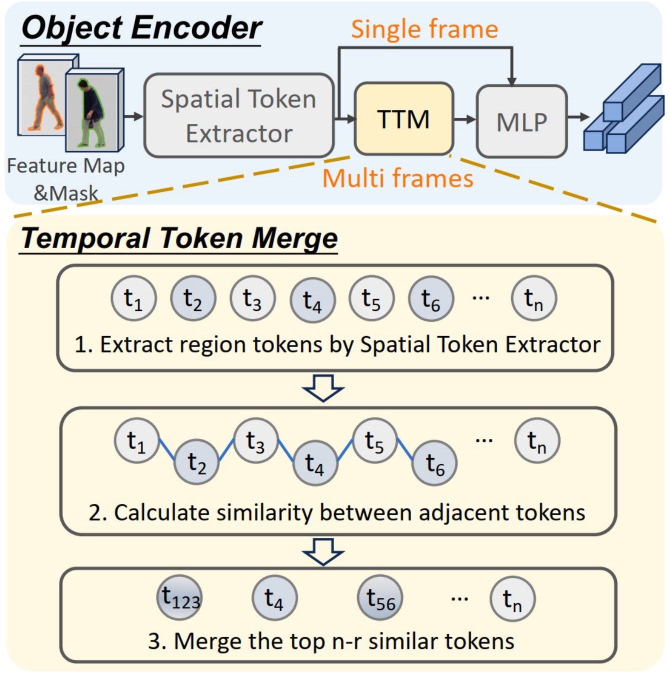
Optimization Loss:

$$\mathcal{L} = \sum_{(V,\boldsymbol{R},x,y)} \log P(y \mid V, R_1, ..., R_n, x)$$

VideoRefer Suite: Advancing spatial-temporal object understanding with video LLM, in *CVPR 2025*. **2024.6-2024.11**

Fine-grained Object/Region Understanding

VideoRefer Model



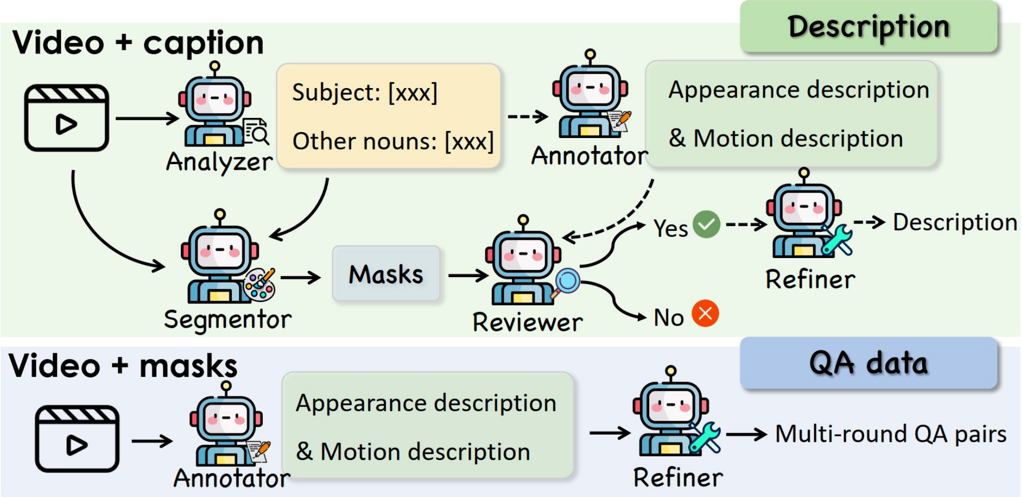
Compute the cosine similarity between each pair of adjacent tokens:

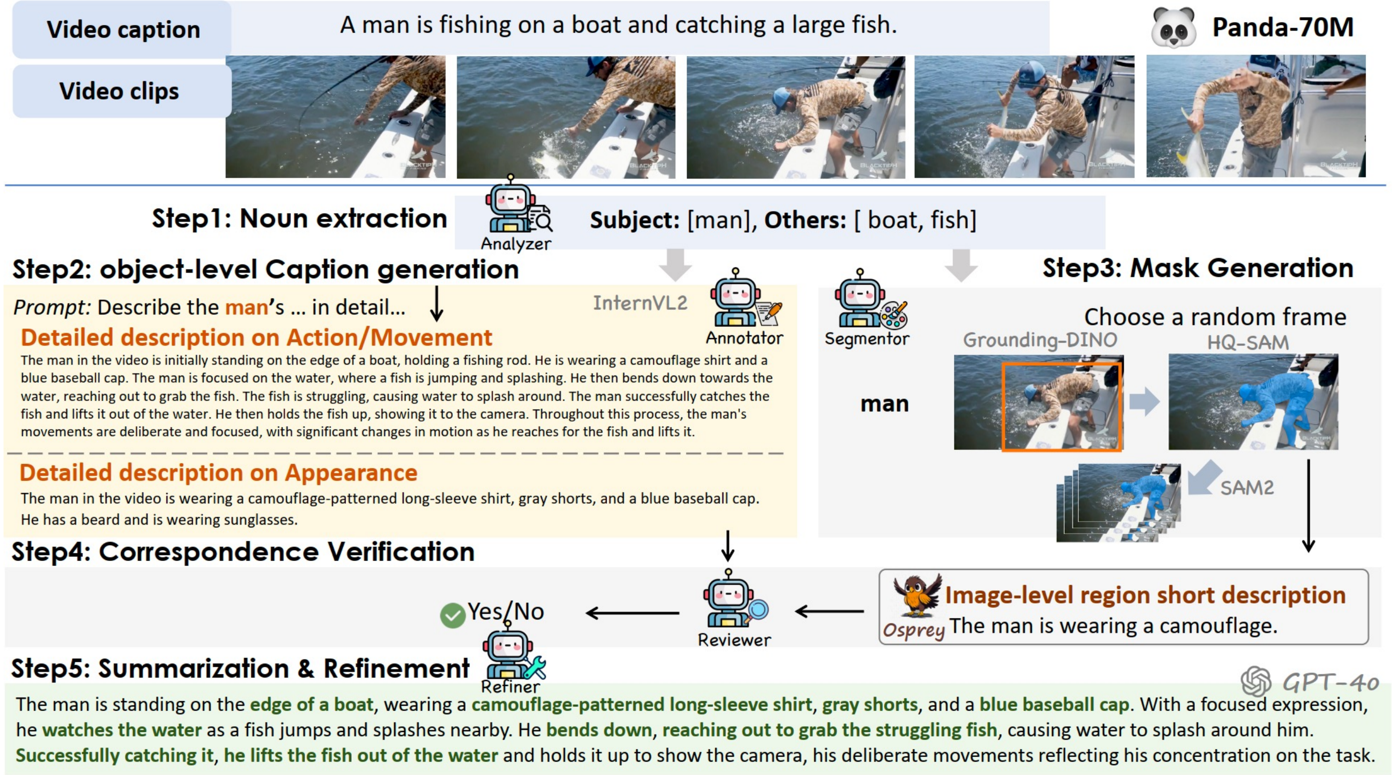
$$\mathbf{S}_{m,m+1} = \frac{\mathbf{O}_m \cdot \mathbf{O}_{m+1}}{\|\mathbf{O}_m\| \cdot \|\mathbf{O}_{m+1}\|}, 0 \leq m < k$$

VideoRefer Suite: Advancing spatial-temporal object understanding with video LLM, in *CVPR 2025*. **2024.6-2024.11**

Fine-grained Object/Region Understanding

VideoRefer-700K





Multi-agent Data Engine

Step1- Analyzer: Qwen2-Instruct-7B
Step2-Annotator: InternVL2-26B
Step3-Segmentor:Grounding DINO&SAM 2
Step4-Reviewer: Osprey&Qwen2-Instruct-7B
Step5-Refiner:GPT-4o

Three types:
- Object-level Detailed Caption
- Object-level Short Capton
- Object-level QA

	Manually True	Manually False
Reviewer True	88 (TP)	12 (FP)
Reviewer False	36 (FN)	64 (TN)

Table 8. Confusion matrix of the randomly sampled 100 items in the Reviewer evaluation.

Fine-grained Object/Region Understanding

VideoRefer-Bench

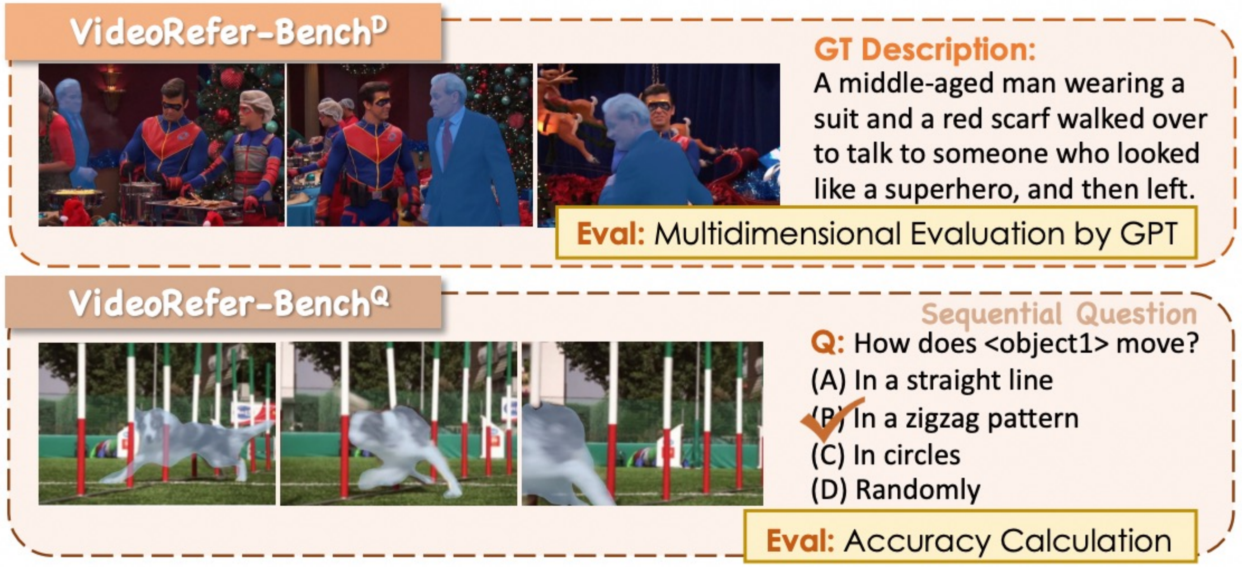
VideoRefer-Bench^D *(Descripion Generation)*

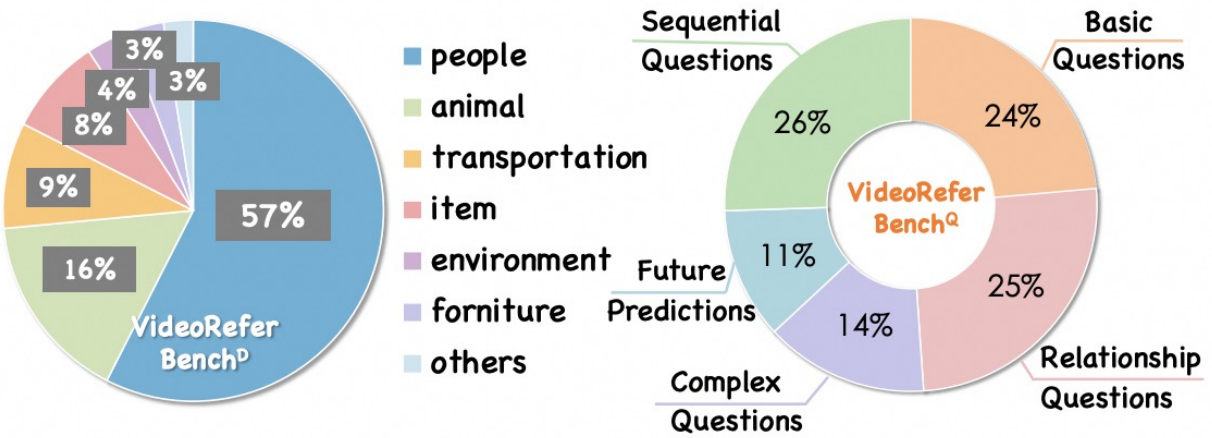
GPT assign scores from 0 to 5 across:

- Subject Correspondence
- Appearance Description
- Temporal Description
- Hallucination Detection

VideoRefer-Bench^Q *(Multi-choice QA)*

- Basic Questions
- Sequential Questions
- Relationship Questions
- Reasoning Questions
- Future Predictions





(a) Category list in Bench^D (b) Question types in Bench^Q

Fine-grained Object/Region Understanding

Experiments

Method	Single-Frame					Multi-Frame				
	SC	AD	TD	HD	Avg.	SC	AD	TD	HD	Avg.
Generalist Models										
LongVU-7B [38]	2.02	1.45	1.98	1.12	1.64	2.33	1.80	2.39	1.68	2.05
LongVA-7B [54]	2.63	1.59	2.12	2.10	2.11	3.02	2.30	1.92	2.51	2.44
LLaVA-OV-7B [15]	2.62	1.58	2.19	2.07	2.12	3.09	1.94	2.50	2.41	2.48
Qwen2-VL-7B [45]	2.97	2.24	2.03	2.31	2.39	3.30	2.54	2.22	2.12	2.55
InternVL2-26B [8]	3.55	2.99	2.57	2.25	2.84	4.08	3.35	3.08	2.28	3.20
GPT-4o-mini [29]	3.56	2.85	2.87	2.38	2.92	3.89	3.18	2.62	2.50	3.05
GPT-4o [29]	3.34	2.96	3.01	2.50	2.95	4.15	**3.31**	**3.11**	2.43	3.25
Specialist Models										
Image-level models										
Ferret-7B [46]	3.08	2.01	1.54	2.14	2.19	3.20	2.38	1.97	1.38	2.23
Osprey-7B [48]	3.19	2.16	1.54	2.45	2.34	3.30	2.66	2.10	1.58	2.41
Video-level models										
Elysium-7B [43]	2.35	0.30	0.02	**3.59**	1.57	–	–	–	–	–
Artemis-7B [33]	–	–	–	–	–	3.42	1.34	1.39	2.90	2.26
VideoRefer-7B	**4.41**	**3.27**	**3.03**	2.97	**3.42**	**4.44**	3.27	3.10	**3.04**	**3.46**

Method	Basic Questions	Sequential Questions	Relationship Questions	Reasoning Questions	Future Predictions	Average
Generalist Models						
LongVU-7B [38]	47.2	61.3	57.5	85.3	65.8	61.0
LongVA-7B [54]	56.2	62.5	52.0	83.9	65.8	61.8
InternVL2-26B [8]	58.5	63.5	53.4	88.0	**78.9**	65.0
GPT-4o-mini [29]	57.6	67.1	56.5	85.9	75.4	65.8
Qwen2-VL-7B [45]	62.0	69.6	54.9	87.3	74.6	66.0
LLaVA-OV-7B [15]	58.7	62.9	64.7	87.4	76.3	67.4
GPT-4o [29]	62.3	**74.5**	**66.0**	88.0	73.7	71.3
Specialist Models						
Osprey-7B [48]	45.9	47.1	30.0	48.6	23.7	39.9
Ferret-7B [46]	35.2	44.7	41.9	70.4	74.6	48.8
VideoRefer-7B	**75.4**	68.6	59.3	**89.4**	78.1	**71.9**

Method	Perception-Test	MVBench	VideoMME
VideoLLaMA2 [9]	51.4	54.6	47.9/50.3
VideoLLaMA2.1 [9]	54.9	57.3	54.9/56.4
Artemis [33]	47.1	34.1	28.8/35.3
VideoRefer	**56.3**	**59.6**	**55.9/57.6**

Mode	VideoRefer-Bench^{D}			VideoRefer-Bench^{Q}		
	TD	HD	Avg.	SQ	RQ	Avg.
Single-frame	3.03	2.97	3.42	68.3	59.1	71.9
Multi-frame	**3.10**	**3.04**	**3.46**	**70.6**	**60.5**	**72.1**

VideoRefer Suite: Advancing spatial-temporal object understanding with video LLM, in *CVPR 2025*. **2024.6-2024.11**

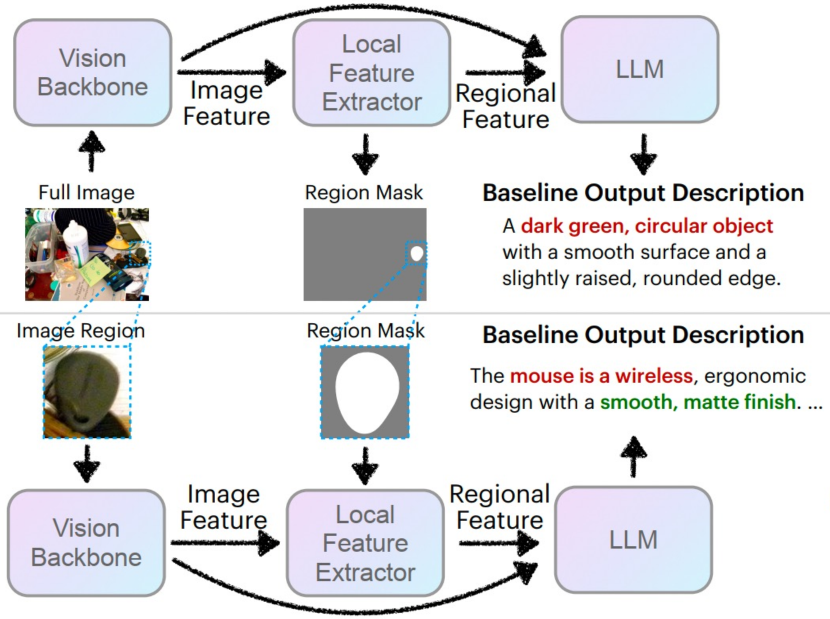
Fine-grained Object/Region Understanding

NVIDIA Describe Anything Model（DAM）

Describe Anything: Detailed Localized Image and Video Captioning

Long Lian[1,2] Yifan Ding[1] Yunhao Ge[1] Sifei Liu[1] Hanzi Mao[1] Boyi Li[1,2] Marco Pavone[1]
Ming-Yu Liu[1] Trevor Darrell[2] Adam Yala[2,3] Yin Cui[1]
[1]NVIDIA [2]UC Berkeley [3]UCSF

Figure 1: **Describe Anything Model (DAM)** generates **detailed localized captions** for user-specified regions within **images** (top) and **videos** (bottom). DAM accepts various region specifications, including clicks, scribbles, boxes, and masks. For videos, specifying the region in *any frame* suffices.



Lack of details 😢

Reason: region details already lost in image feature extraction

Degraded understanding 😢

Reason: regions are given without image context

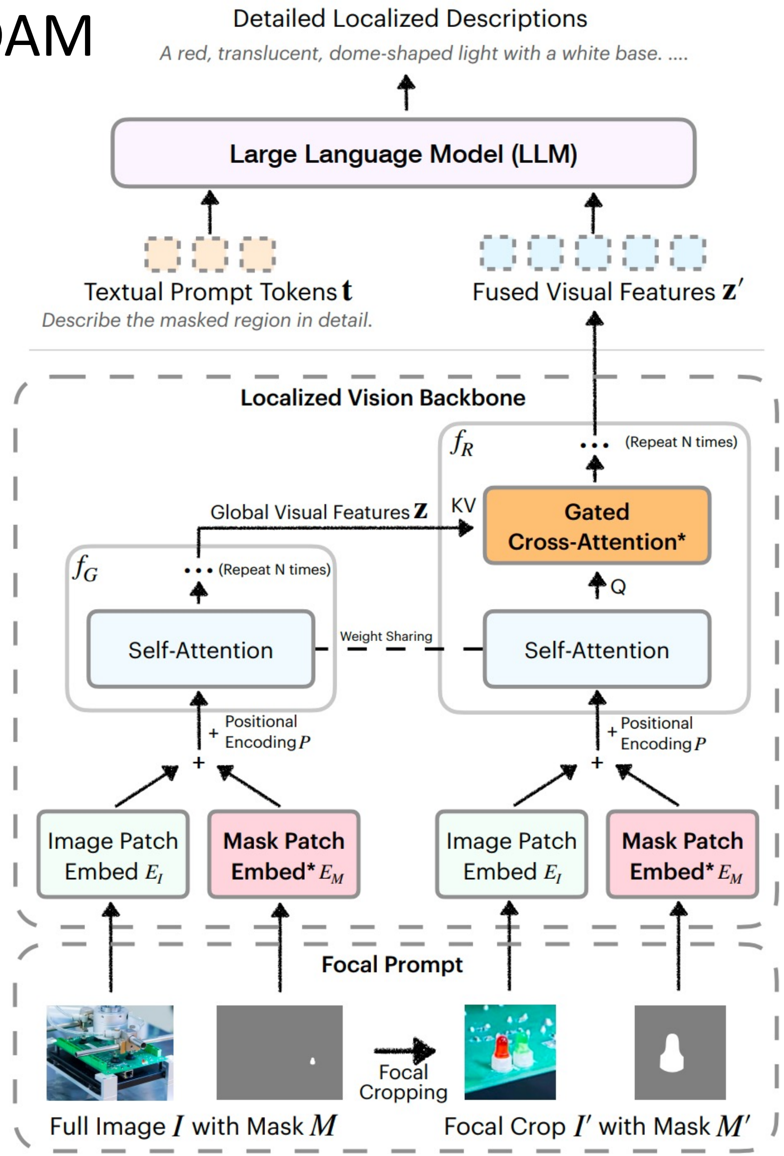
Typical two regional frameworks

Adopting VideoRefer-Bench & Osprey Evaluation.

Describe Anything: Detailed Localized Image and Video Captioning, in *ICCV 2025*. **NVIDIA&UC Berkely**

Fine-grained Object/Region Understanding

DAM



- Focal Prompt

  Full image and a **zoomed-in region** with corresponding mask

$$x = E_I(I) + E_M(M) + P, \quad z = f_G(x)$$

$$x' = E_I(I') + E_M(M') + P, \quad z' = f_R(x', z)$$

- Localized Vision Backbone

  Inject global features into the encoding of local regions using

Gated Cross-Attention Adaptor

$$\mathbf{h}^{(l)\prime} = \mathbf{h}^{(l)} + \tanh\left(\gamma^{(l)}\right) \cdot \text{CrossAttn}\left(\mathbf{h}^{(l)}, \mathbf{z}\right),$$

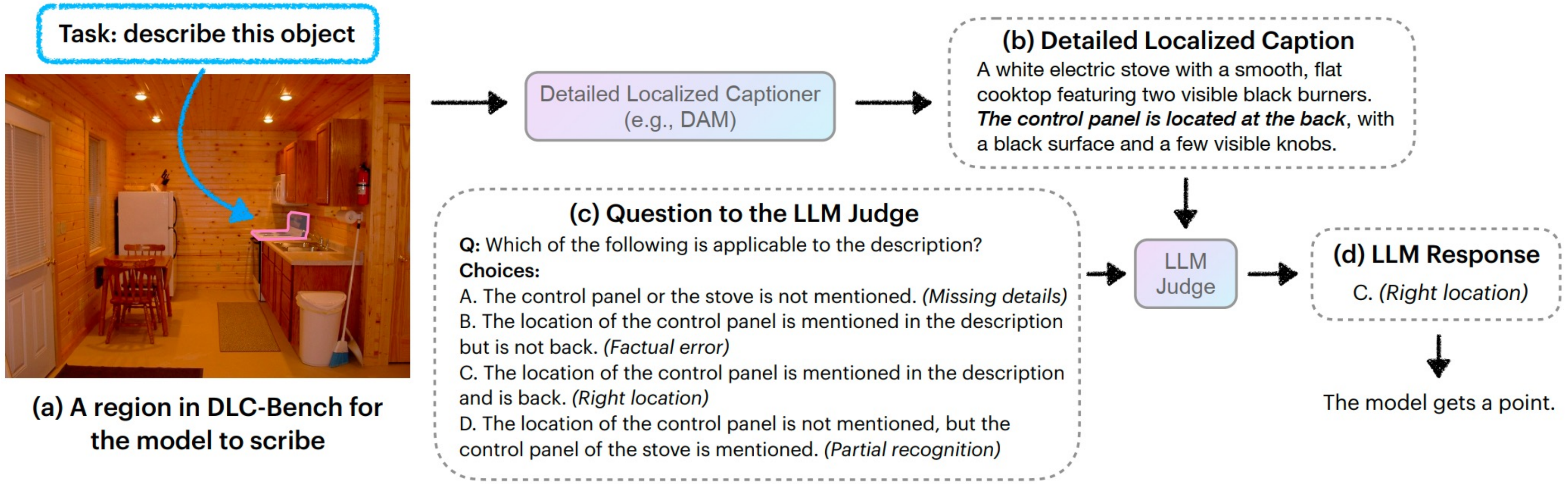
$$\mathbf{h}^{(l)}_{\text{Adapter}} = \mathbf{h}^{(l)\prime} + \tanh\left(\beta^{(l)}\right) \cdot \text{FFN}\left(\mathbf{h}^{(l)\prime}\right),$$

Describe Anything: Detailed localized image and video captioning [C], in *ICCV 2025*.

NVIDIA&UC Berkely

Fine-grained Object/Region Understanding

- **Simple Extension to Video Frames**
  - All frames are naïvely concatenated along the temporal axis, **without considering inter-frame correlations;**
  - Each object per frame is represented by **196 tokens;**
  - Limited to captioning tasks only.
- Using LLM as Judge for Performance Evaluation & Dataset



Dataset	# Images	# Regions
Stage 1:		
LVIS [29]	90,613	373,551
Mapillary Vistas v2.0 [53]	17,762	100,538
COCO Stuff [11]	28,365	32,474
OpenImages v7 [33, 35]	64,874	96,006
PACO [60]	24,599	81,325
Stage 2:		
SA-1B (10%)	592,822	774,309
Total	**819,035**	**1,458,203**

Describe Anything: Detailed localized image and video captioning [C], in *ICCV 2025*.

NVIDIA&UC Berkely

Fine-grained Object/Region Understanding

Experiments

The Evaluation setting of Osprey

Method	LVIS (%)		PACO (%)	
	Sem. Sim. (↑)	Sem. IoU (↑)	Sem. Sim. (↑)	Sem. IoU (↑)
LLaVA-7B [48]	49.0	19.8	42.2	14.6
Shikra-7B [15]	49.7	19.8	43.6	11.4
GPT4RoI-7B [99]	51.3	12.0	48.0	12.1
Osprey-7B [95]	65.2	38.2	73.1	52.7
Ferret-13B [93]	65.0	37.8	-	-
VP-SPHINX-7B [45]	86.0	61.2	74.2	49.9
VP-LLAVA-8B [45]	86.7	61.5	75.7	50.0
DAM-8B (Ours)	**89.0**	**77.7**	**84.2**	**73.2**

VideoRefer-Bench

Method	SC	AD	TD	HD†	Avg.
Zero-shot:					
Qwen2-VL-7B [81]	3.30	2.54	2.22	2.12	2.55
InternVL2-26B [20]	4.08	**3.35**	3.08	2.28	3.20
GPT-4o-mini [54]	3.89	3.18	2.62	2.50	3.05
GPT-4o [54]	4.15	3.31	**3.11**	2.43	3.25
Osprey-7B [95]	3.30	2.66	2.10	1.58	2.41
Ferret-7B [93]	3.20	2.38	1.97	1.38	2.23
Elysium-7B [80]	2.35	0.30	0.02	**3.59**	1.57
Artemis-7B [59]	3.42	1.34	1.39	2.90	2.26
DAM-8B (Ours)	**4.45**	3.30	3.03	2.58	**3.34**
In-domain:*					
VideoRefer-7B [96]	4.44	3.27	3.10	3.04	3.46
DAM-8B (Ours)	**4.69**	**3.61**	**3.34**	**3.09**	**3.68**

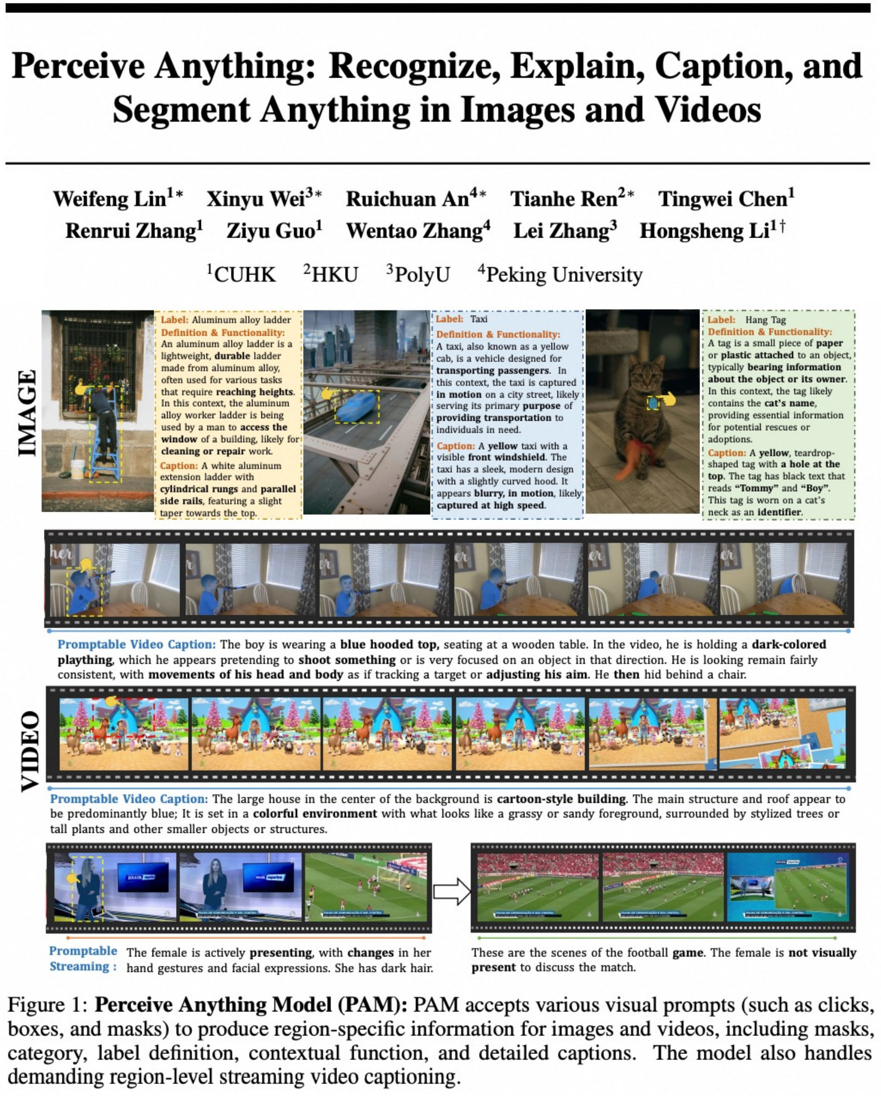
Prompting	XAttn	#IT	Pos (%)	Neg (%)	Avg (%)
Full Image Only	No	196	32.1	65.4	48.7
Local Crop Only	No	196	43.5	76.6	60.1 (+11.4)
Full + Local Crop	No*	392	26.3	58.6	42.4 (-6.3)
Full + Local Crop	Yes	196	45.7	80.6	63.2 (+14.5)
Focal Crop Only	No	196	47.3	**83.6**	65.4 (+16.7)
Full + Focal Crop	Yes	196	**52.3**	82.2	**67.3** (+18.6)

Method	#Params	Pos (%)	Neg (%)	Avg (%)
API-only General VLMs:				
GPT-4o (SOM) [54]	-	5.0	29.2	17.1
o1 (SOM) [55]†	-	0.8	28.0	14.4
Claude 3.7 Sonnet (SOM) [73]†	-	0.5	40.2	20.4
Gemini 2.5 Pro (SOM) [74, 75]†	-	13.2	65.0	39.1
Open-source General VLMs:				
Llama-3.2 Vision (SOM) [25]	11B	16.8	40.4	28.6
Llama-3 VILA1.5 (SOM) [44]	8B	0.6	0.6	0.6
InternVL2.5 (SOM) [20, 21, 84]	8B	8.6	28.6	18.6
LLaVA v1.6 (SOM) [46–48]	7B	2.2	3.8	3.0
Qwen2.5-VL (SOM) [77, 81]	7B	8.5	27.2	17.8
VILA1.5 (SOM) [44]	3B	-0.4	15.4	7.5
DAM (Ours)	3B	**52.3**	**82.2**	**67.3**

Describe Anything: Detailed localized image and video captioning [C], in *ICCV 2025*.

NVIDIA&UC Berkely

Fine-grained Object/Region Understanding



Perceive Anything: Recognize, Explain, Caption, and Segment Anything in Images and Videos

Weifeng Lin[1*] Xinyu Wei[3*] Ruichuan An[4*] Tianhe Ren[2*] Tingwei Chen[1]
Renrui Zhang[1] Ziyu Guo[1] Wentao Zhang[4] Lei Zhang[3] Hongsheng Li[1†]

[1]CUHK [2]HKU [3]PolyU [4]Peking University

IMAGE

VIDEO

Promptable Video Caption: The boy is wearing a blue hooded top, seating at a wooden table. In the video, he is holding a dark-colored plaything, which he appears pretending to shoot something or is very focused on an object in that direction. He is looking remain fairly consistent, with movements of his head and body as if tracking a target or adjusting his aim. He then hid behind a chair.

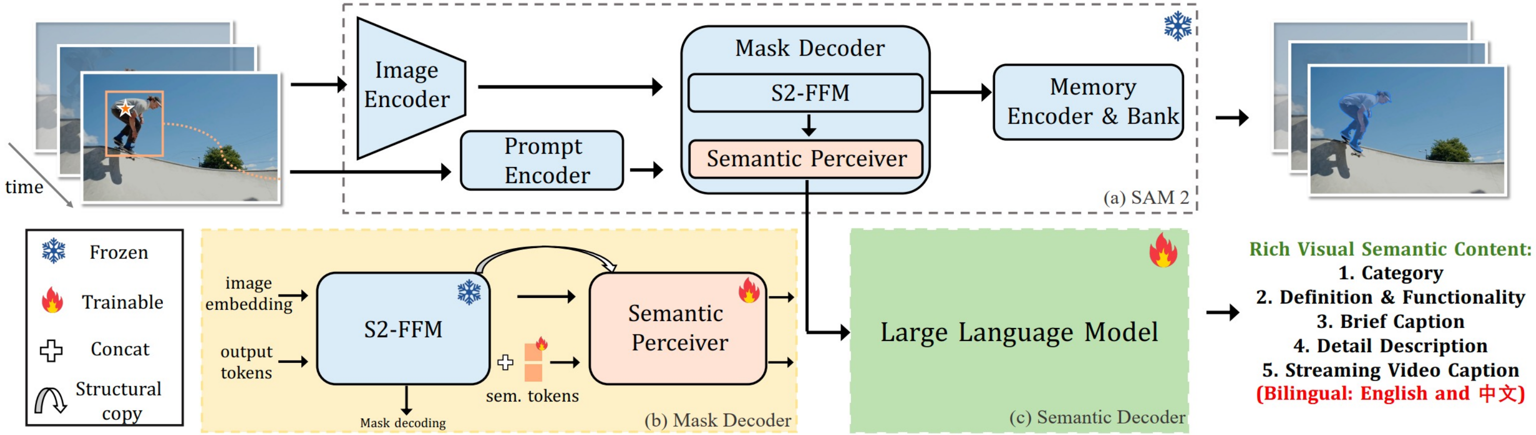
Promptable Video Caption: The large house in the center of the background is cartoon-style building. The main structure and roof appear to be predominantly blue; It is set in a colorful environment with what looks like a grassy or sandy foreground, surrounded by stylized trees or tall plants and other smaller objects or structures.

Promptable Streaming: The female is actively presenting, with changes in her hand gestures and facial expressions. She has dark hair.

These are the scenes of the football game. The female is not visually present to discuss the match.

Figure 1: **Perceive Anything Model (PAM):** PAM accepts various visual prompts (such as clicks, boxes, and masks) to produce region-specific information for images and videos, including masks, category, label definition, contextual function, and detailed captions. The model also handles demanding region-level streaming video captioning.

Perceive Anything Model（PAM）



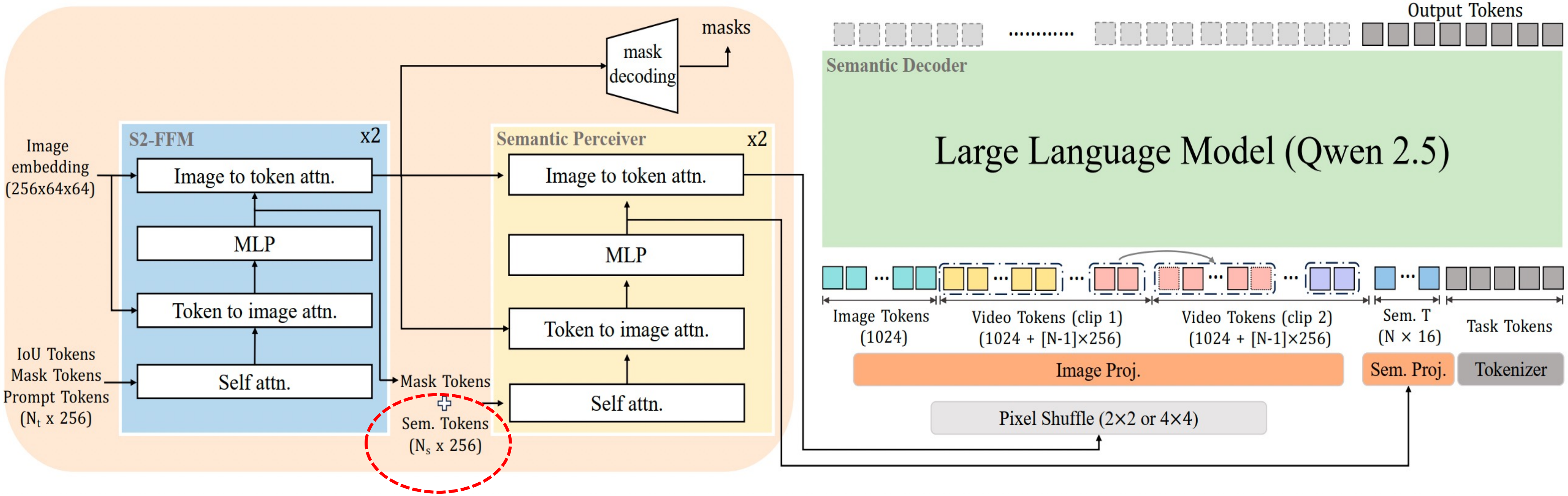
- Extends **SAM 2** by extracting its intermediate visual features and transforming them into **LLM-compatible tokens**.
- Enables segmentation mask decoding and semantic content decoding simultaneously.

CUHK & HK PloyU

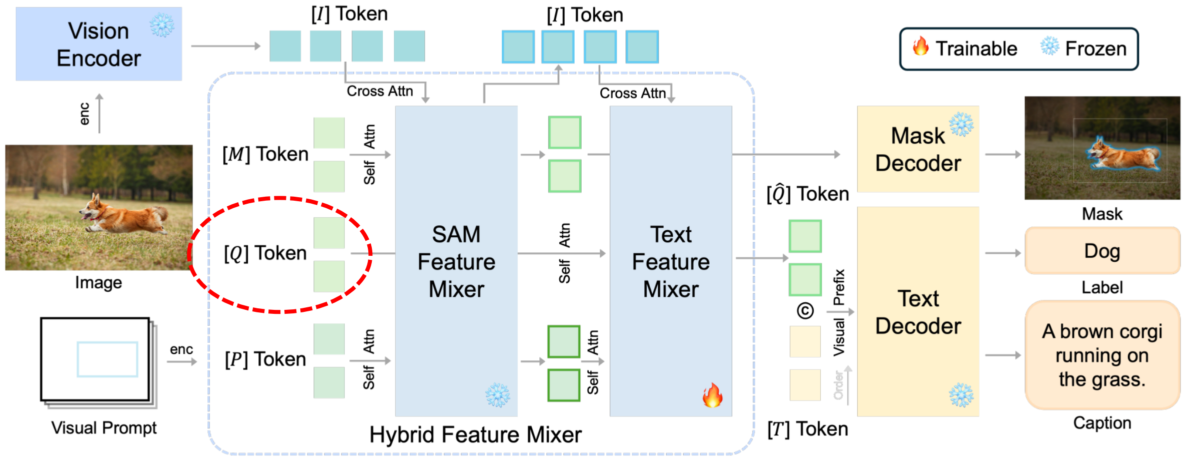
Perceive Anything: Recognize, Explain, Caption, and Segment Anything in Images and Videos, *arXiv:2506.05302*.

Fine-grained Object/Region Understanding

S2-FFM **Semantic Perceiver** **64x64 × N visual tokens & Ns × N semantic tokens**



SCA Model



Segment and Caption Anything, in *CVPR2024*.

Fine-grained Object/Region Understanding

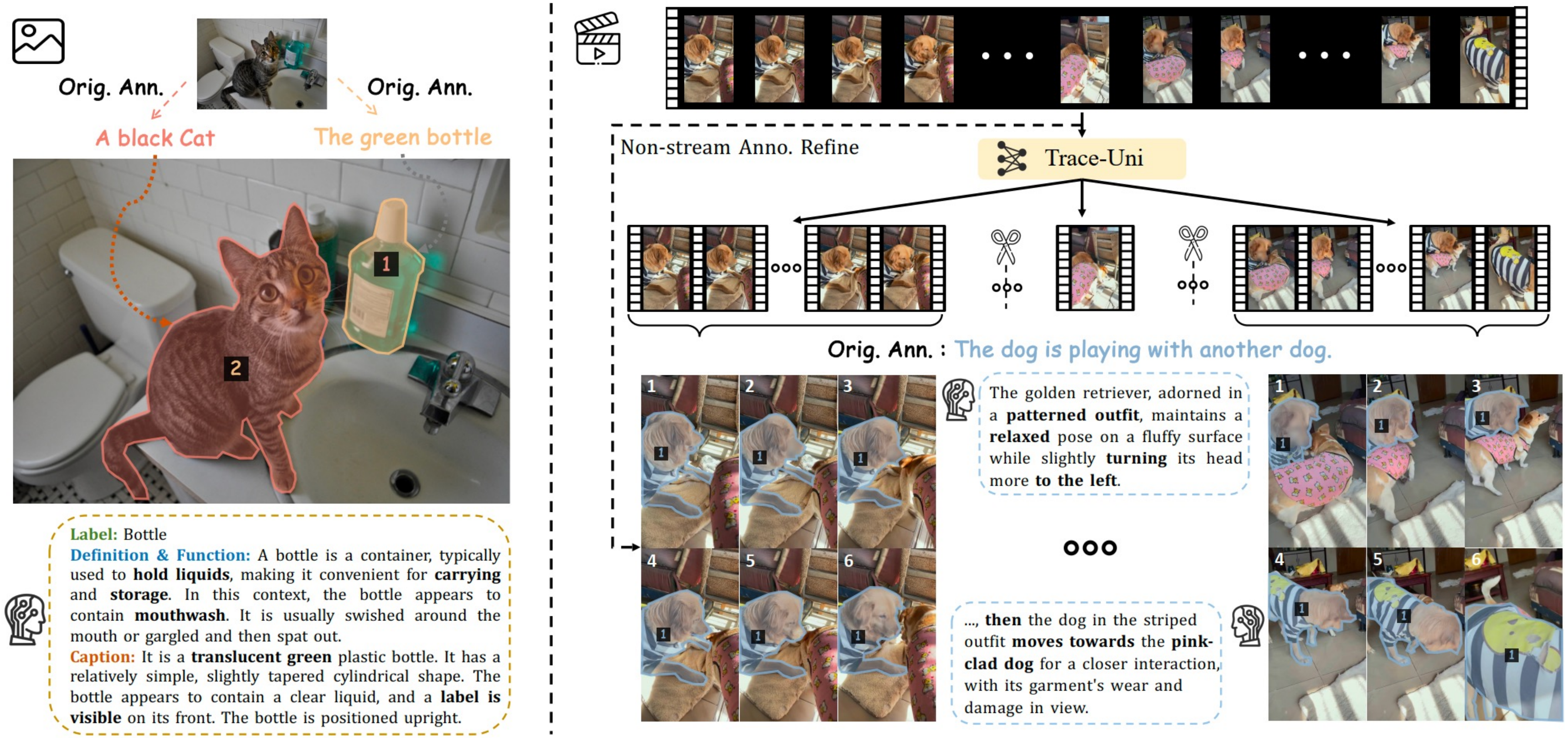


Image Data:

1.5M image region-text pairs

Video Data:

- Storyboard-based expansion
- Event-aware segmentation

600K video region-text pairs

Supporting both **English and Chinese**.

A Large-Scale, Multi-Granular Region-Text Dataset

Perceive Anything: Recognize, Explain, Caption, and Segment Anything in Images and Videos, *arXiv:2506.05302*.

Fine-grained Object/Region Understanding

Streaming Object Caption

Limitations：

- **Fixed window** size without long-term memory;
- Limited to object captioning without multi-round, multi-object interaction.

Perceive Anything: Recognize, Explain, Caption, and Segment Anything in Images and Videos, *arXiv:2506.05302*.

Content

1. Fine-grained Object/Region Understanding

    Image/Video

2. Efficient VLMs with Visual Token Compression

Model-driven：TokenPacker, FastV, VisionZip, VisionTrim & LongVU

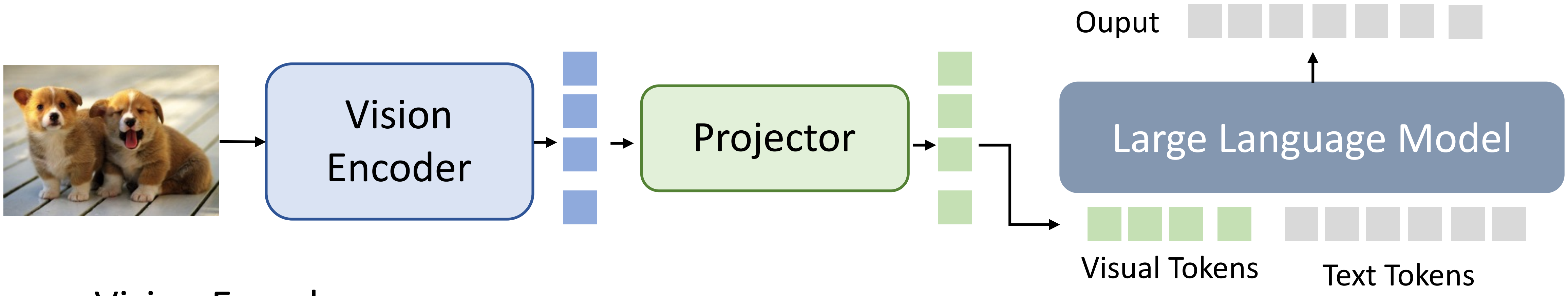
Data-driven：VocoLLAMA, Video-XL & DTR

Other Paradigm: mPLUG-Owl3, Lavi

3. Streaming Understanding & Interaction for AI Assistant

    Training/Training-free

Efficient VLMs with Visual Token Compression



- Vision Encoder
  - **CLIP-VIT-L： ~0.3B**
- Large Language Model
  - **LLaMA/Vicuna: 7B/13B**
- Visual Projector
  - **MLP： 336x336 input -> 576 tokens**

☹ **LLM dominates the main computational and memory demands.**

☺ **Reducing the number of visual tokens is a pivotal approach to bolster the efficiency.**

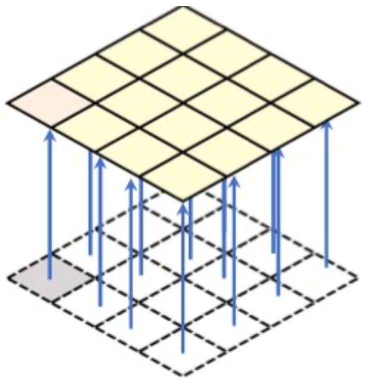
2024.2-2024.6

Tokenpacker: Efficient visual projector for multimodal LLM [J], *IJCV 2025*.

Efficient VLMs with Visual Token Compression

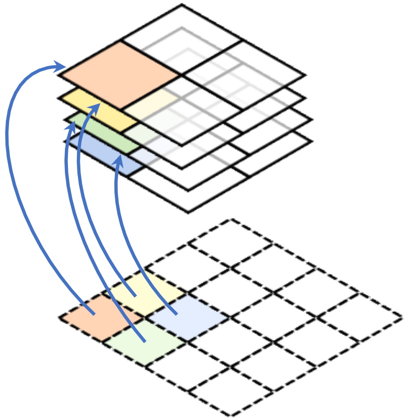
Linear Projector

- One-to-one transformation
- 336x336 ->576token
- Retaining the detailed information with **redundant tokens**



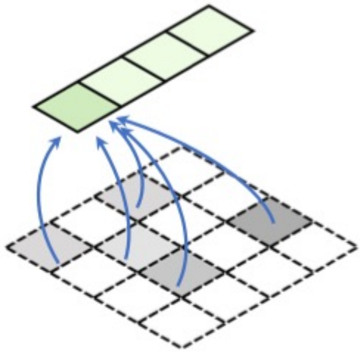
Pixel shuffle

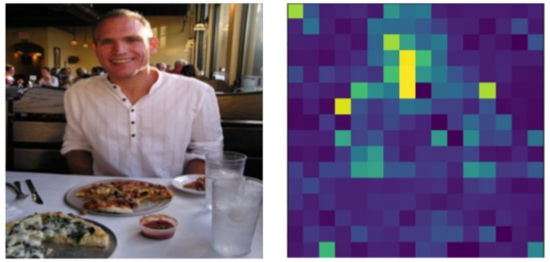
- Token reduction:144
- Nearby concatenation
- **Destroying intrinsic characteristics**



Resampler/Q-Former

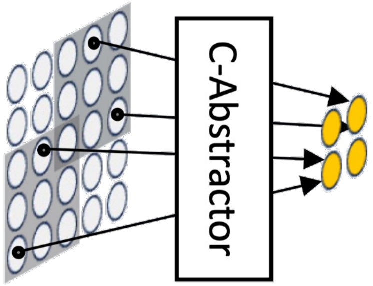
- Learnable queries （64/144）
- Extracting the most relevant visual tokens, **ignoring other objects.**





Abstracter

- Local interaction
- Convolution layers
- **Omitting fine detailed information**



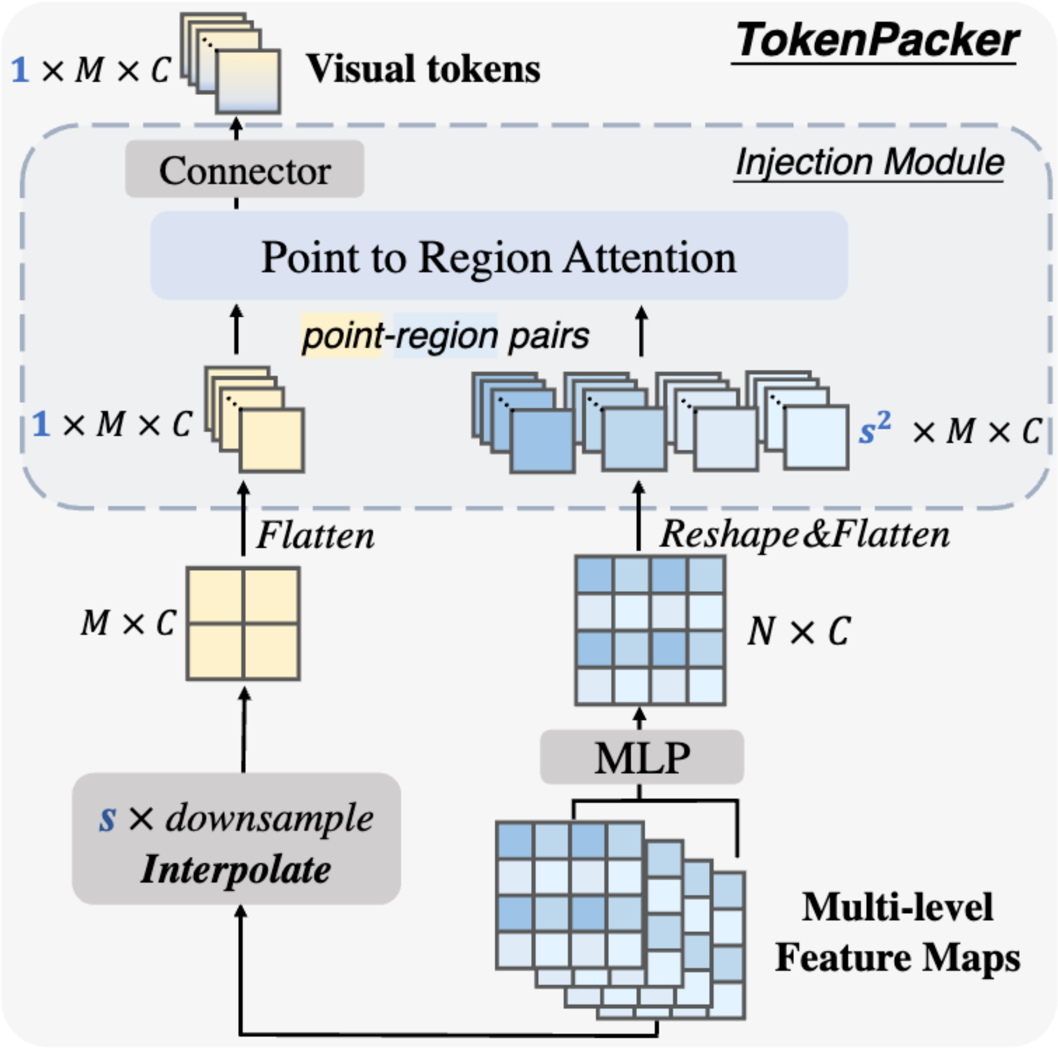
[1] Improved baselines with visual instruction tuning, in *NeurIPS2024*
[2] Qwen-vl: A frontier large vision-language model with versatile abilities, *Arxiv 2023*
[3] How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, *Arxiv 2024*
[4] Honeybee: Locality-enhanced projector for multimodal llm, in *CVPR2024*

Efficient VLMs with Visual Token Compression

TokenPacker

- **Coarse-to-fine**

  Down-sampling features as coarse foundation
- **Point to Region Attention**, injecting the finer region feature to point query
- Multi-level visual features: 12-16-22-33
- **Scale factor**： S ∈ {2,3,4} to control the **reduction rate** {4, 9, 16}, even less.



2024.2-2024.6

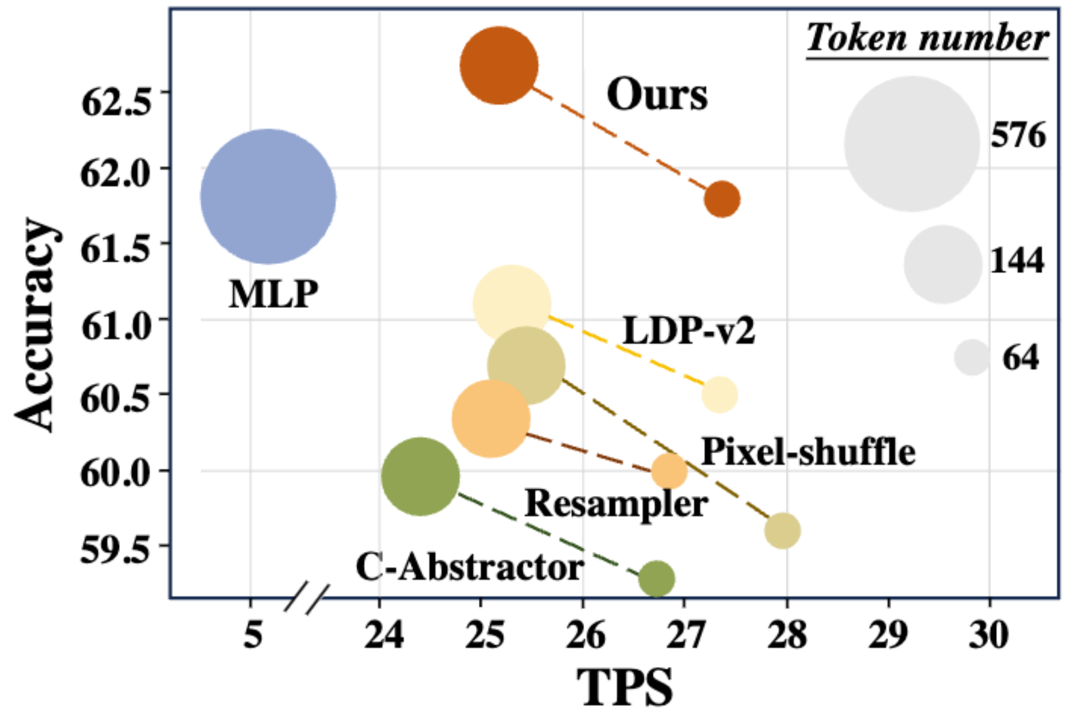
Tokenpacker: Efficient visual projector for multimodal LLM [J], *IJCV 2025*.

Efficient VLMs with Visual Token Compression

Comparisons with same setting

LLaVA-1.5 as the baseline

Projector	#Token	TPS	MMB	MM-Vet	VQA^{v2}	GQA	POPE	VizWiz	Avg.
MLP [9]	576	4.9	64.3	31.1	78.5	62.0	85.9	50.0	62.0
Resampler [11]	144	24.8	63.1	29.2	75.1	58.4	84.7	51.9	60.4
C-Abstractor [24]	144	24.1	63.1	29.4	74.6	59.2	84.6	49.2	60.0
Pixel-Shuffle [13]	144	**25.2**	64.0	29.7	76.2	60.1	85.9	48.8	60.8
LDP-v2 [26]	144	25.1	**66.2**	28.7	77.3	61.1	86.1	47.6	61.2
Ours	144	24.9	65.1	**33.0**	**77.9**	**61.8**	**87.0**	**52.0**	**62.8**
Resampler [11]	64	26.6	63.4	29.2	74.1	57.7	83.4	**53.0**	60.1
C-Abstractor [24]	64	26.5	62.5	29.0	74.4	59.3	62.5	45.6	59.3
Pixel-Shuffle [13]	64	**27.7**	63.2	28.5	74.6	59.1	85.2	47.4	59.7
LDP-v2 [26]	64	27.1	63.7	30.0	75.3	59.7	85.5	49.3	60.6
Ours	64	27.1	**64.1**	**31.7**	**77.2**	**61.1**	**86.3**	50.7	**61.9**



- TPS: token per second
- Evaluation on a NVIDIA A100 GPU

1/9 of the original results in a 5.5× acceleration, while maintaining comparable performance.

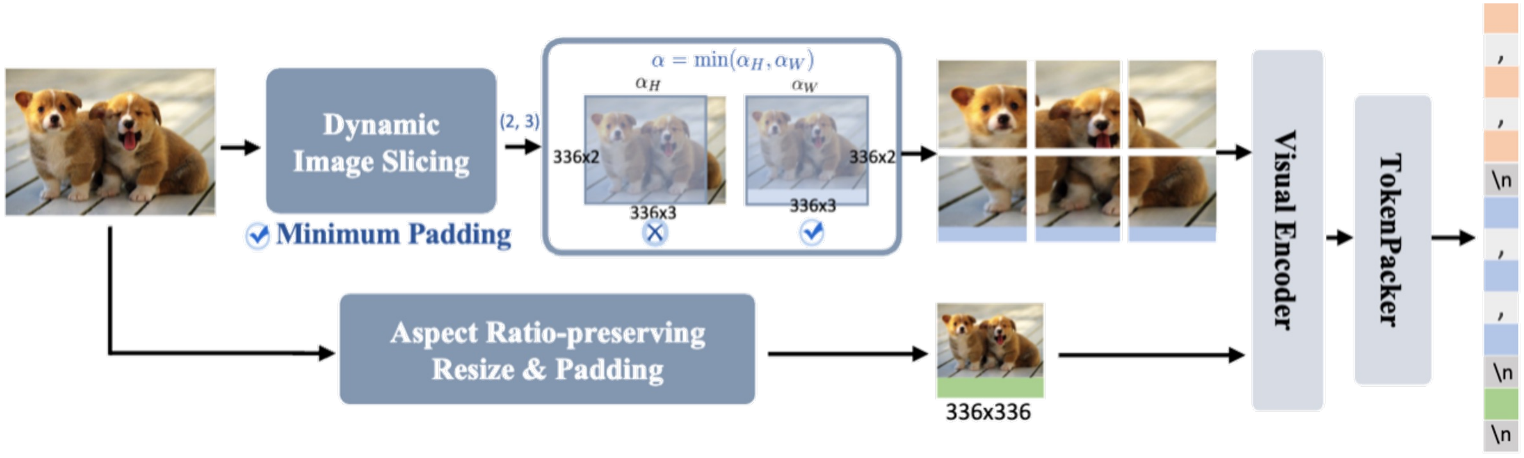
Exhibit a more favorable superiority on accuracy and efficient against other counterparts.

Tokenpacker: Efficient visual projector for multimodal LLM [J], *IJCV 2025*.

2024.2-2024.6

Efficient VLMs with Visual Token Compression

TokenPacker-HD



High-resolution Framework（TokenPacker-HD）

Method	LLM	#Data	Max Res.	#Token	VQA^T	OCRB	DocVQA	MMB	MMMU	MME	VQA^{v2}	VizWiz	POPE
OtterHD [26]	Fuyu-8B [5]	-	1024×1024	–	–	–	–	58.3	–	1294/–	–	–	86.0
SPHINX-2k [32]	LLaMA-13B	1.0B	762×762	2890	61.2	–	–	65.9	–	1471/–	80.7	44.9	87.2
UReader [56]	LLaMA-13B	86M	896×1120	–	57.6	–	65.4	–	–	–	–	–	–
Monkey [30]	QWen-7B	1.0B	896×1344	1792	–	514	–	–	–	–	80.3	**61.2**	67.6
TextHawk [59]	InternLM-7B	115M	1344×1344	–	–	–	**76.4**	**74.6**	–	1500/–	–	–	–
LLaVA-UHD [55]	Vicuna-13B	1.2M	672×1008	–	67.7	–	–	68.0	–	1535/–	81.7	56.1	**89.1**
LLaVA-NeXT [34]	Vicuna-7B	1.3M	672×672	2880	64.9	–	–	67.4	35.8	1519/332	81.8	57.6	86.5
LLaVA-NeXT [34]	Vicuna-13B	1.3M	672×672	2880	67.1	–	–	70.0	36.2	1575/326	**82.8**	60.5	86.2
Mini-Genimi-HD [28]	Vicuna-7B	2.7M	1536×1536	2880	68.4	456*	65.0*	65.8	36.8	1546/319	80.3*	54.6*	86.8*
Mini-Genimi-HD [28]	Vicuna-13B	2.7M	1536×1536	2880	70.2	501*	70.0*	68.6	37.3	1575/326	81.5*	57.2*	87.0*
LLaVA-TokenPacker-HD	Vicuna-7B	2.7M	1088×1088	~954†	68.0	452	60.2	67.4	35.4	1489/338	81.2	54.7	88.2
LLaVA-TokenPacker-HD	Vicuna-13B	2.7M	1088×1088	~954†	69.3	498	63.0	69.5	**38.8**	**1595/356**	82.0	59.2	88.1
LLaVA-TokenPacker-HD	Vicuna-13B	2.7M	1344×1344	~1393†	**70.6**	**521**	70.0	68.7	37.4	1574/350	81.7	57.0	88.0
LLaVA-TokenPacker-HD	Vicuna-13B	2.7M	1344×1344	~619♯	68.8	470	63.0	69.9	38.2	1577/353	81.7	61.0	87.6
LLaVA-TokenPacker-HD	Vicuna-13B	2.7M	1344×1344	~347§	68.4	447	58.0	68.3	36.9	1577/332	81.2	58.1	88.0

Adopt the same training data Mini-Gemini[1]

Method	Res.	LVIS		PACO		Cityscapes	ADE20K
		SS	S-IoU	SS	S-IoU	AP	AP
Osprey [60]	512	65.2	38.2	73.1	52.7	29.2	31.8
FixedSplit [36]	672	69.4	45.6	79.3	63.5	33.7	39.5
AdaptiveSplit-1 [37]	Any	69.7	45.9	79.3	63.9	38.0	40.6
AdaptiveSplit-2 [56]	Any	70.0	46.3	79.3	63.9	42.3	41.0
Ours	Any	**71.6**	**47.5**	**79.8**	**64.1**	**43.8**	**42.0**

Employing Osprey on TokenPakcer-HD framework

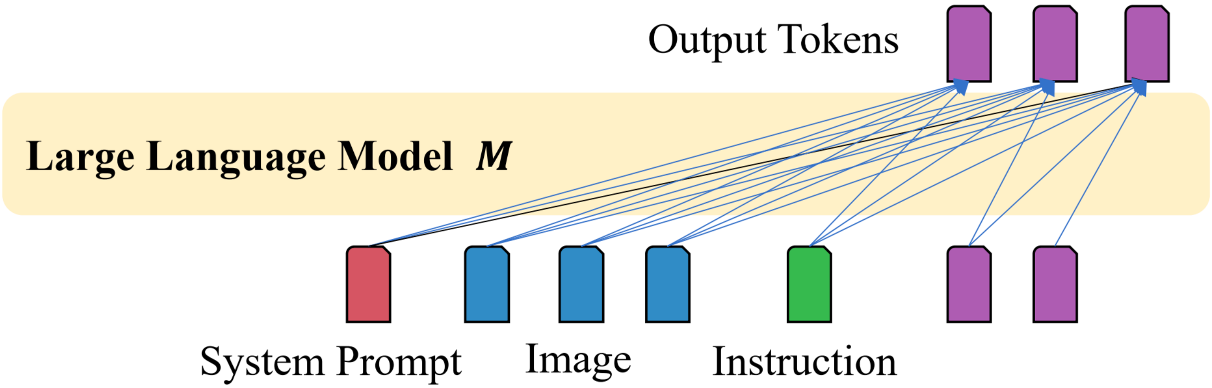
Mini-gemini: Mining the potential of multi-modality vision language models, *arXiv:2403.18814*.

Efficient VLMs with Visual Token Compression

Training-free

FastV

(Within **LLM Decoding**)

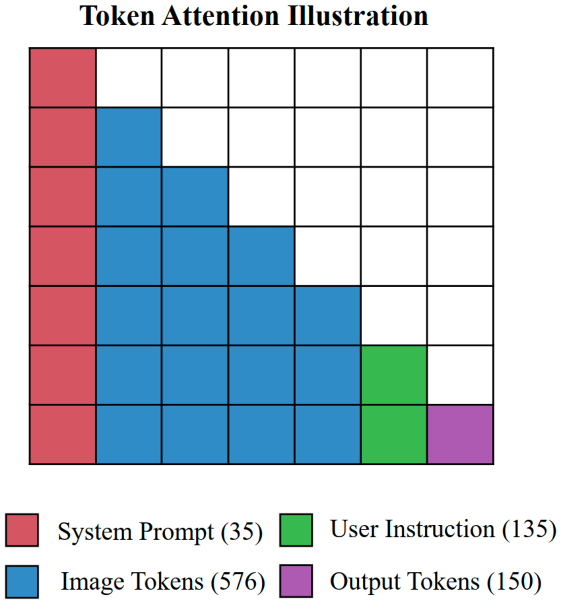


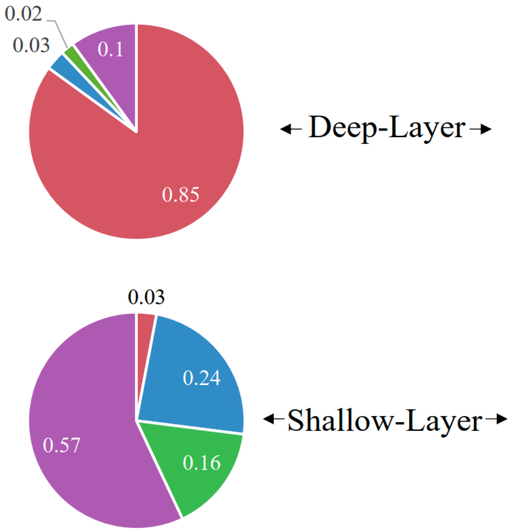
Total attention score of **current token**:

$$\alpha_{sys}^{i,j} + \alpha_{img}^{i,j} + \alpha_{ins}^{i,j} + \alpha_{out}^{i,j} = 1$$

Attention Allocation:

$$\lambda_{sys}^{j} = \sum_{i=1}^{n} \alpha_{sys}^{i,j}$$





Attention Allocation (Decoding Output Tokens)

- System Prompt : **472x**
- Image Tokens : 1x
- User Instruction: 3x
- Output Tokens : 12.8x

- System Prompt : 2x
- Image Tokens : 1x
- User Instruction : 3x
- Output Tokens : **9x**

Attention Efficiency (Attention Allocation / Token Number)

An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, *ECCV2024*.

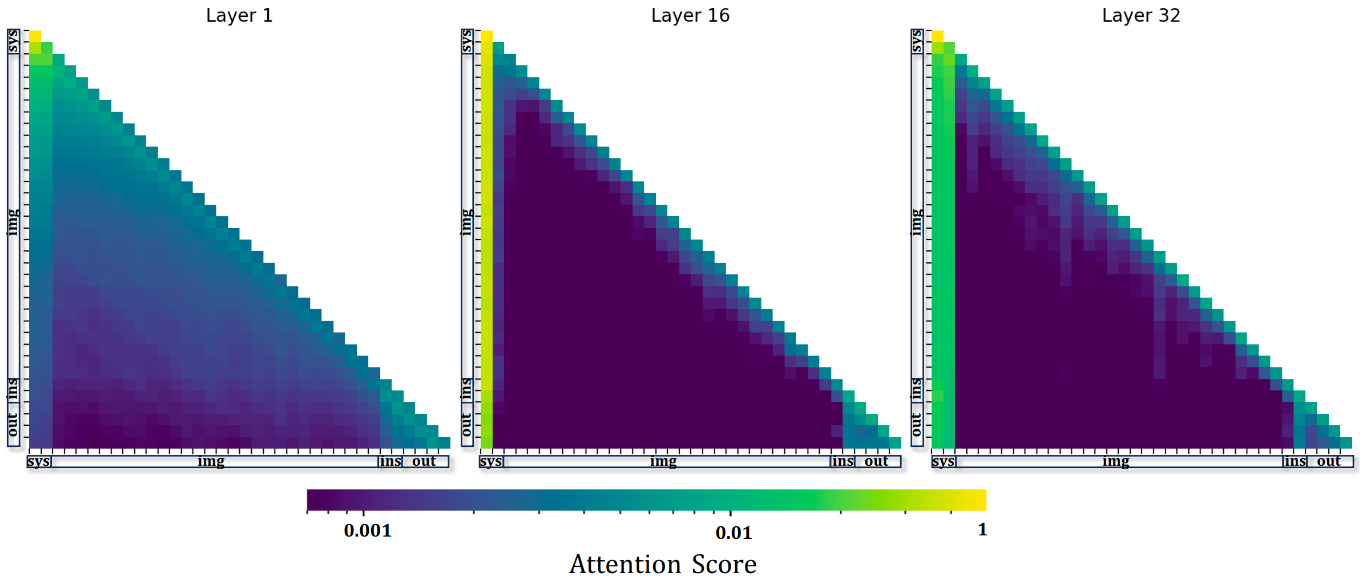
PKU

Efficient VLMs with Visual Token Compression

Training-free

FastV

(Within **LLM Decoding**)



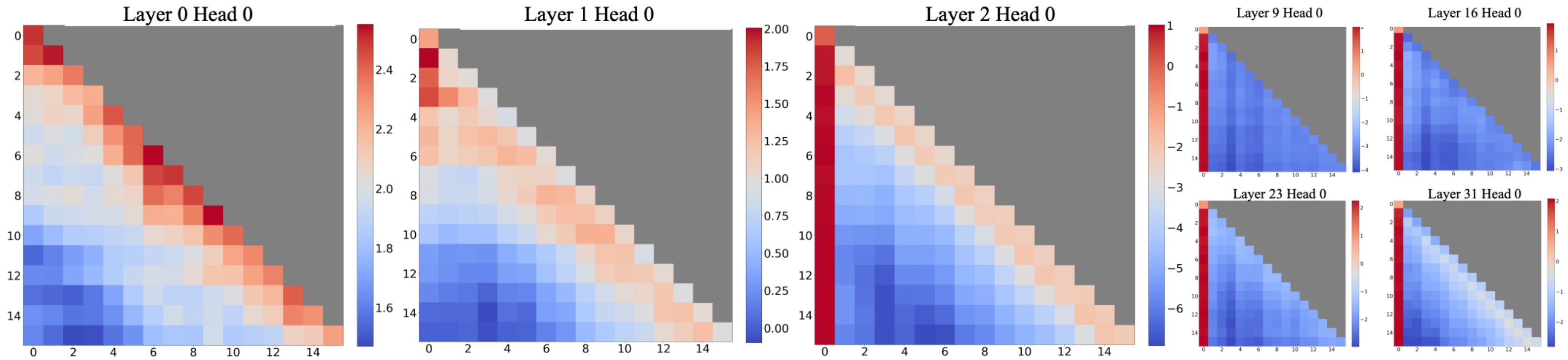


Figure 2: Visualization of the *average* attention logits in Llama-2-7B over 256 sentences, each with a length of 16. Observations include: (1) The attention maps in the first two layers (layers 0 and 1) exhibit the "local" pattern, with recent tokens receiving more attention. (2) Beyond the bottom two layers, the model heavily attends to the initial token across all layers and heads.

An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, *ECCV2024*. **PKU**

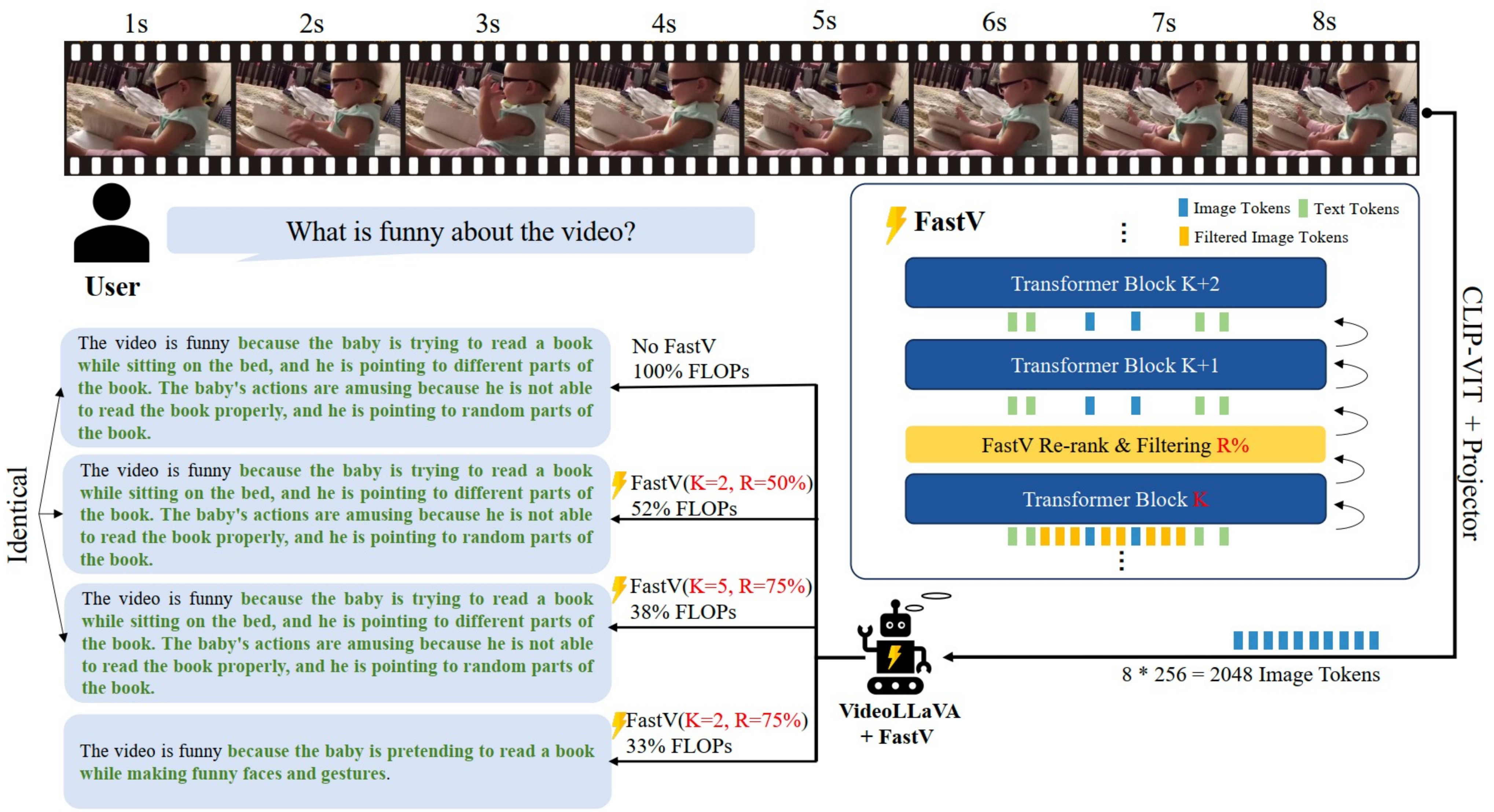
Efficient Streaming Language Models with Attention Sinks, *Arxiv 2309.17453* **MIT, Meta & CMU**

Efficient VLMs with Visual Token Compression

Training-free

FastV

(Within **LLM Decoding**)



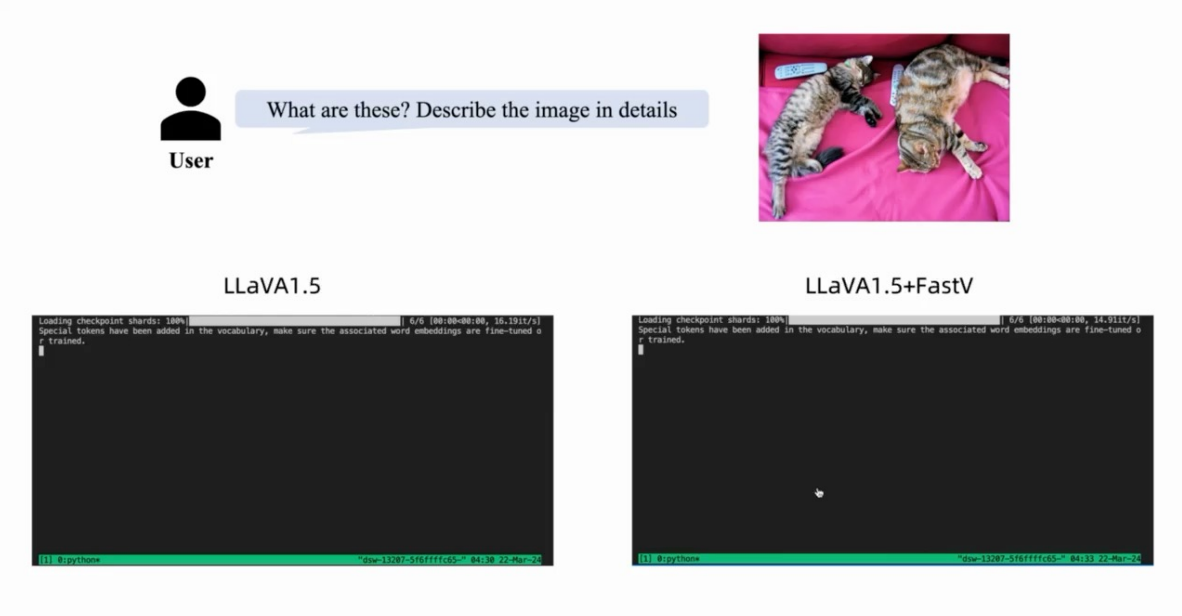
An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, *ECCV2024*.

Efficient VLMs with Visual Token Compression

Training-free

Comparisons with same setting

Model	K	R	FastV Settings Flops(B)	Flops Ratio	Nocaps *CIDEr*	Flickr30k *CIDEr*	A-OKVQA *Accuracy*	MMMU *Accuracy*	Avg
LLaVA-1.5-7B	Baseline		99.3	100%	99.8	67.9	76.7	34.8	**69.8**
	2	90%	19.9	20%	72.1	43.7	70.1	**35**	55.2
	2	75%	32.8	33%	94.6	63.6	75.5	34.8	67.1
	2	50%	54.6	55%	99.7	67.5	**77**	34.4	69.7
	3	90%	22.8	23%	87.2	55.8	71.9	34.8	62.4
	3	75%	34.8	35%	98	65	74.7	34.1	68.0
	3	50%	56.6	57%	99.7	**68.3**	76.7	34.3	**69.8**
	5	90%	27.8	28%	88.6	59.3	70.6	33.9	63.1
	5	75%	39.7	40%	98.5	66.3	74.8	34.3	68.5
	5	50%	59.6	60%	99.2	67.9	76.8	34.3	69.6
	0	90%	18.9	19%	7	53.2	66.8	34.7	40.4
	0	75%	28.8	29%	27.2	61.4	72.8	35.1	49.1
	0	50%	51.6	52%	**100.9**	65.5	75.3	34.3	69.0
LLaVA-1.5-13B	Baseline		154.6	100%	102.8	73	**82**	36.4	**73.6**
	2	90%	29.7	19%	87.9	62	75	36.3	65.3
	2	75%	50.2	32%	100.5	72.5	80.9	**38.1**	73.0
	2	50%	84.6	55%	**103.1**	73.4	81	36.7	**73.6**
	3	90%	33.0	21%	90.2	63.6	75.2	34.9	66.0
	3	75%	52.9	34%	100.9	72.1	79.5	36.4	72.2
	3	50%	86.4	56%	102.7	73.4	81.3	36.4	73.5
	5	90%	39.6	26%	93.5	67.4	75.8	35.4	68.0
	5	75%	58.4	38%	101.4	72.5	80	36.2	72.5
	5	50%	90.1	58%	102.5	**73.5**	81.2	36.6	73.5
QwenVL-Chat-7B	Baseline		71.9	100%	**94.9**	**72.5**	**75.6**	**35.8**	**69.7**
	2	90%	15.8	22%	81.9	61.5	68.5	35.3	61.7
	2	75%	24.4	34%	90.5	67.0	75.1	35.3	67.0
	2	50%	39.5	55%	94.4	71.4	75.3	35.6	69.2



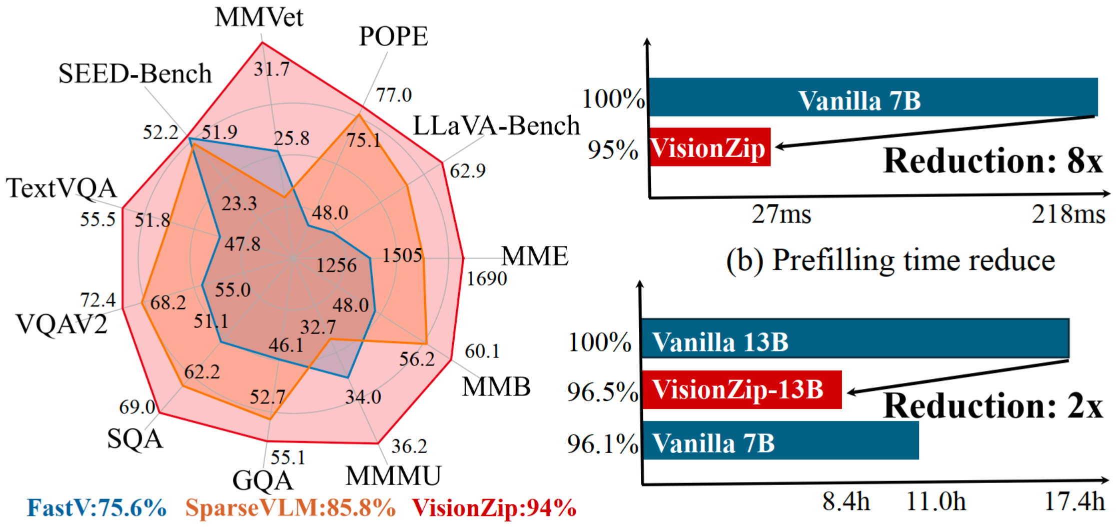
An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, *ECCV2024*.

Efficient VLMs with Visual Token Compression

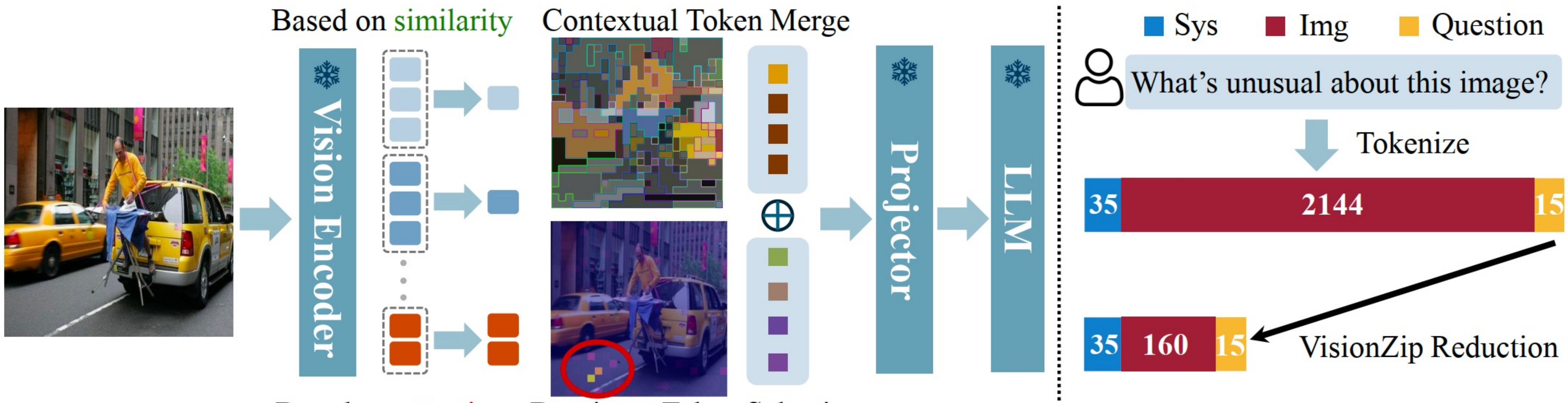
Training-free

VisionZip

(Within **Visual Encoding**)



(a) VisionZip outperforms sota EfficientVLM (c) Boost 13B faster and better than 7B



Visionzip: Longer is better but not necessary in vision language models, in *CVPR2025*.

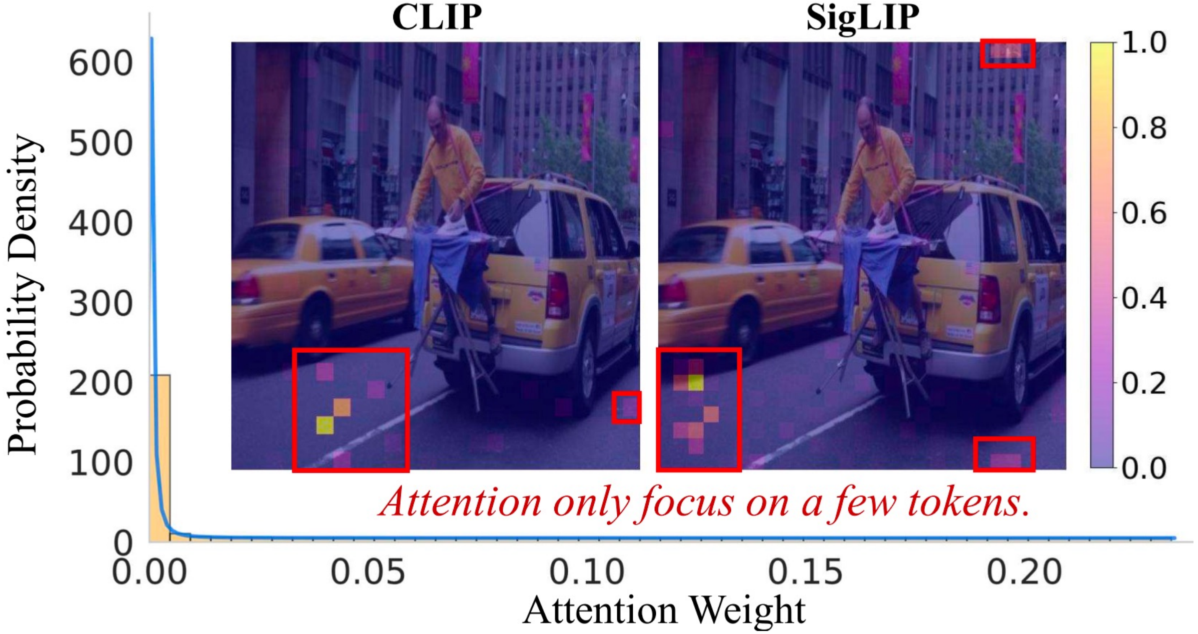
CUHK & HKUST

Efficient VLMs with Visual Token Compression

Training-free

Dominant Token Selection

- Using **[CLS] Tokens** attention scores to identify key visual tokens (CLIP)
- Average attention each token receives from all others (SigLIP)



Algorithm 1 Pseudocode for Dominant Token Selection

```
# B: batch size; S: sequence length
# H: number of attention heads;
# K: number of target dominant tokens
# CLS_IDX: Index of the CLS token
# SELECT_LAYER: Selected layer for Visual Token

# set the output_attentions=True to get the attention
output = vision_tower(images, output_hidden_states=
    True, output_attentions=True)

#attn in shape (B, H, S, S)
attn = output.attentions[SELECT_LAYER]

#attn in shape (B, H, S, S)
vanilla_tokens = output.hidden_states[SELECT_LAYER]

#The attention received by each token
#If no CLS, use mean calculate received attention
attn_rec = attn[:, :, cls_idx, cls_idx+1:].sum(dim=1)

# Select K Dominant Tokens
_, topk_idx = attn_rec.topk(K, dim=1)

# Concat with the CLS token
dominant_idx = cat(CLS_IDX, topk_idx+1)

# filter the Dominant Tokens
dominant_tokens = vanilla_tokens.filter(dominant_idx)
```

`cat`: concatenation; `filter`: select the tokens based on the index.

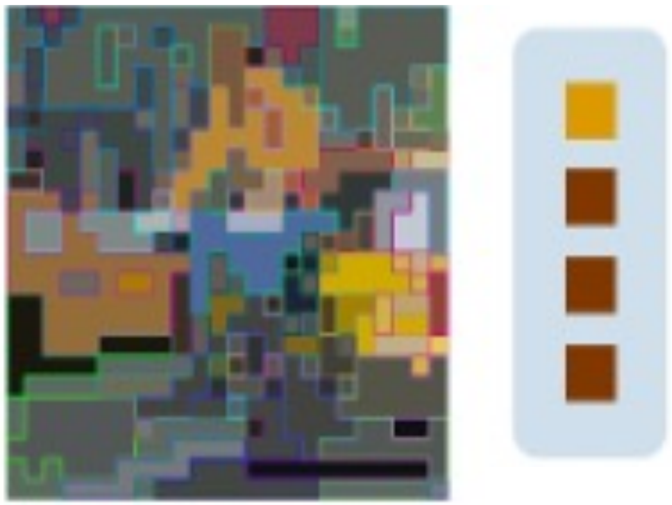
Visionzip: Longer is better but not necessary in vision language models, in *CVPR2025*.

Efficient VLMs with Visual Token Compression

Training-free

Contextual Tokens Merging

- Merge the remaining tokens to avoid losing any small but potentially important information.



Algorithm 2 Pseudocode for Contextual Tokens Merging.

```
# Remove dominant tokens
remaining = vanilla_tokens.mask(dominant_tokens)

# Split into target and merge tokens
# M represents the desired number of contextual tokens
targets, merge = uniform_split(remaining, M)

# Compute similarity based on the key values
simlarity = bmm(to_merge.K, targets.K.transpose(1, 2))

# Assign each merge token to the most similar target
assign_idx = simlarity.argmax(dim=2)

# Merge by averaging
context_tokens = avg_merge(assign_idx, targets, merge)
```

`uniform_split`: Uniformly sample the target tokens, and the rest are the merge tokens; `avg_merge`: Average merge the tokens based on the assigned indices.

Visionzip: Longer is better but not necessary in vision language models, in *CVPR2025*.

Efficient VLMs with Visual Token Compression

Training-free

Experiments

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED	MMVet	LLaVA-B	Avg.
Upper Bound, 576 Tokens **(100%)**												
Vanilla (CVPR24)	61.9	64.7	1862	85.9	69.5	78.5	58.2	36.3	58.6	31.1	66.8	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	
Retain 192 Tokens (↓ **66.7%**)												
FastV (ECCV24)	52.7	61.2	1612	64.8	67.3	67.1	52.5	34.3	57.1	27.7	49.4	88.2%
	85.1%	94.6%	86.6%	75.4%	96.8%	85.5%	90.2%	94.5%	97.4%	89.7%	74.0%	
SparseVLM (2024.10)	57.6	62.5	1721	83.6	69.1	75.6	56.1	33.8	55.8	31.5	66.1	96.4%
	93.1%	96.6%	92.4%	97.3%	99.4%	96.3%	96.4%	93.1%	95.2%	101.3%	99.0%	
VisionZip	59.3	63.0	1782.6	85.3	68.9	76.8	57.3	36.6	56.4	31.7	67.7	**98.5%**
	95.8%	97.4%	95.7%	99.3%	99.1%	97.8%	98.5%	100.8%	96.2%	101.9%	101.3%	
VisionZip ‡	60.1	63.4	1834	84.9	68.2	77.4	57.8	36.2	57.1	32.6	66.7	99.1%
	97.1%	98.0%	98.5%	98.8%	98.1%	98.6%	99.3%	99.7%	97.4%	104.8%	99.9%	
Retain 128 Tokens (↓ **77.8%**)												
FastV (ECCV24)	49.6	56.1	1490	59.6	60.2	61.8	50.6	34.9	55.9	28.1	52.0	83.5%
	80.1%	86.7%	80.0%	69.4%	86.6%	78.7%	86.9%	96.1%	95.4%	90.9%	77.8%	
SparseVLM (2024.10)	56.0	60.0	1696	80.5	67.1	73.8	54.9	33.8	53.4	30	62.7	93.4%
	90.5%	92.7%	91.1%	93.7%	96.5%	94.0%	94.3%	93.1%	91.1%	96.5%	93.9%	
VisionZip	57.6	62.0	1761.7	83.2	68.9	75.6	56.8	37.9	54.9	32.6	64.8	**97.6%**
	93.1%	95.8%	94.6%	96.9%	99.1%	96.3%	97.6%	104.4%	93.7%	104.8%	97.6%	
VisionZip ‡	58.9	62.6	1823	83.7	68.3	76.6	57.0	37.3	55.8	32.9	64.8	98.4%
	95.2%	96.8%	97.9%	97.4%	98.3%	97.6%	97.9%	102.8%	95.2%	105.8%	97.0%	
Retain 64 Tokens (↓ **88.9%**)												
FastV (ECCV24)	46.1	48.0	1256	48.0	51.1	55.0	47.8	34.0	51.9	25.8	46.1	75.6%
	74.5%	74.2%	67.5%	55.9%	73.5%	70.1%	82.1%	93.7%	88.6%	83.0%	69.0%	
SparseVLM (2024.10)	52.7	56.2	1505	75.1	62.2	68.2	51.8	32.7	51.1	23.3	57.5	85.8%
	85.1%	86.9%	80.8%	87.4%	89.4%	86.9%	89.0%	90.1%	87.2%	74.5%	86.1%	
VisionZip	55.1	60.1	1690	77.0	69.0	72.4	55.5	36.2	52.2	31.7	62.9	**94.0%**
	89.0%	92.9%	90.8%	89.6%	99.3%	92.2%	95.4%	99.7%	89.1%	101.9%	94.2%	
VisionZip ‡	57.0	61.5	1756	80.9	68.8	74.2	56.0	35.6	53.4	30.2	63.6	95.2%
	92.1%	95.1%	94.3%	94.2%	99.0%	94.5%	96.2%	98.1%	91.1%	97.1%	95.2%	

‡ Fine-tuning visual projector; other frozen

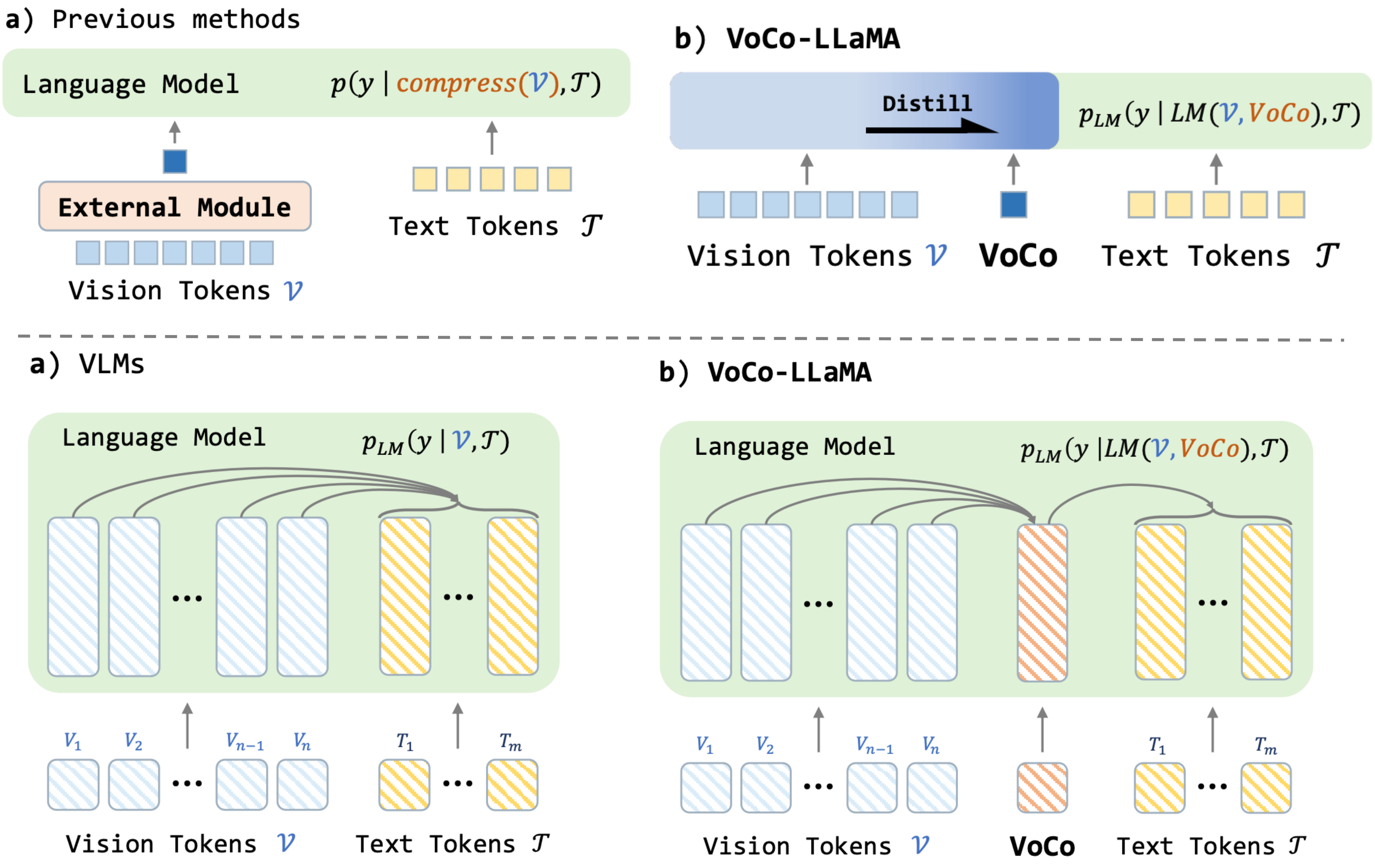
Visionzip: Longer is better but not necessary in vision language models, in *CVPR2025*.

Efficient VLMs with Visual Token Compression

Data-Driven Method

VoCo-LLaMA

This work introduces a **learnable** Vision Compression **(VoCo) token** between visual and text tokens.



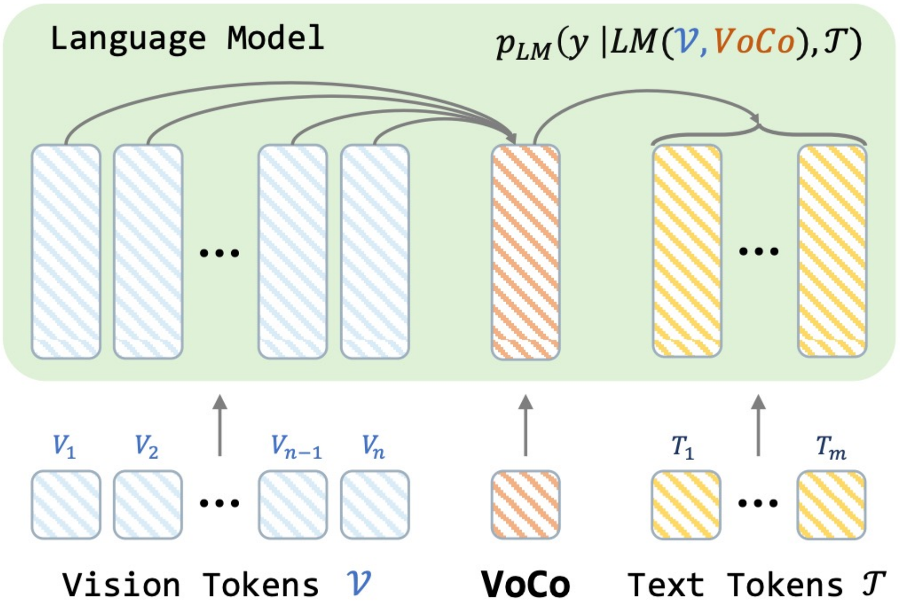
VoCo-LLaMA: Towards Vision Compression with Large Language Models, in *CVPR2025*. **Tsinghua Uni. & Tencent**

Learning to Compress Prompts with Gist Tokens, in *NeurIPS2023*. **Stanford University**

Efficient VLMs with Visual Token Compression

Data-Driven Method

VoCo-LLaMA



576× compression rate while maintaining 83.7% performance.

Modifying the attention mechanism, text tokens attend **solely** to VoCo tokens:

$$M_{ij} = \begin{cases} True, & \text{if } i \in \mathcal{T} \text{ and } j \in VoCo, \\ False, & \text{if } i \in \mathcal{T} \text{ and } j \in \mathcal{V}, \\ True, & \text{otherwise.} \end{cases}$$

Distillation objective:

$$E_{\mathcal{V},\mathcal{T}}[D_{KL}(p_{LM_o}(y \mid \mathcal{V}, \mathcal{T}) \parallel p_{LM_c}(y \mid \mathcal{C}, \mathcal{T}))]$$

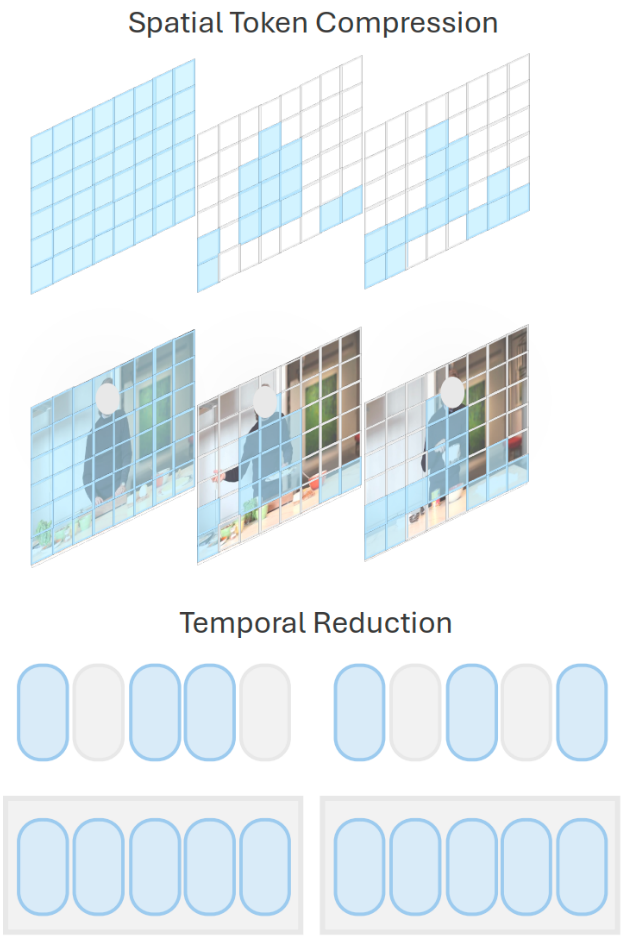
Token	MMB	GQA	VQA^{v2}	SEED	Avg.
576	64.0	61.1	77.7	57.9	100%
128	**61.0**	59.8	**76.9**	**59.1**	**97.7%**
64	60.5	**60.4**	75.4	56.3	93.7%
32	59.4	60.2	75.3	56.2	92.6%
16	58.6	59.4	75.4	56.2	91.3%
8	58.7	59.2	75.3	56.3	91.3%
4	60.4	58.4	74.5	56.0	90.4%
2	60.1	57.7	73.5	55.0	87.8%
1	58.8	57.0	72.3	53.7	83.8%
1	22.3	37.7	41.2	36.9	0%

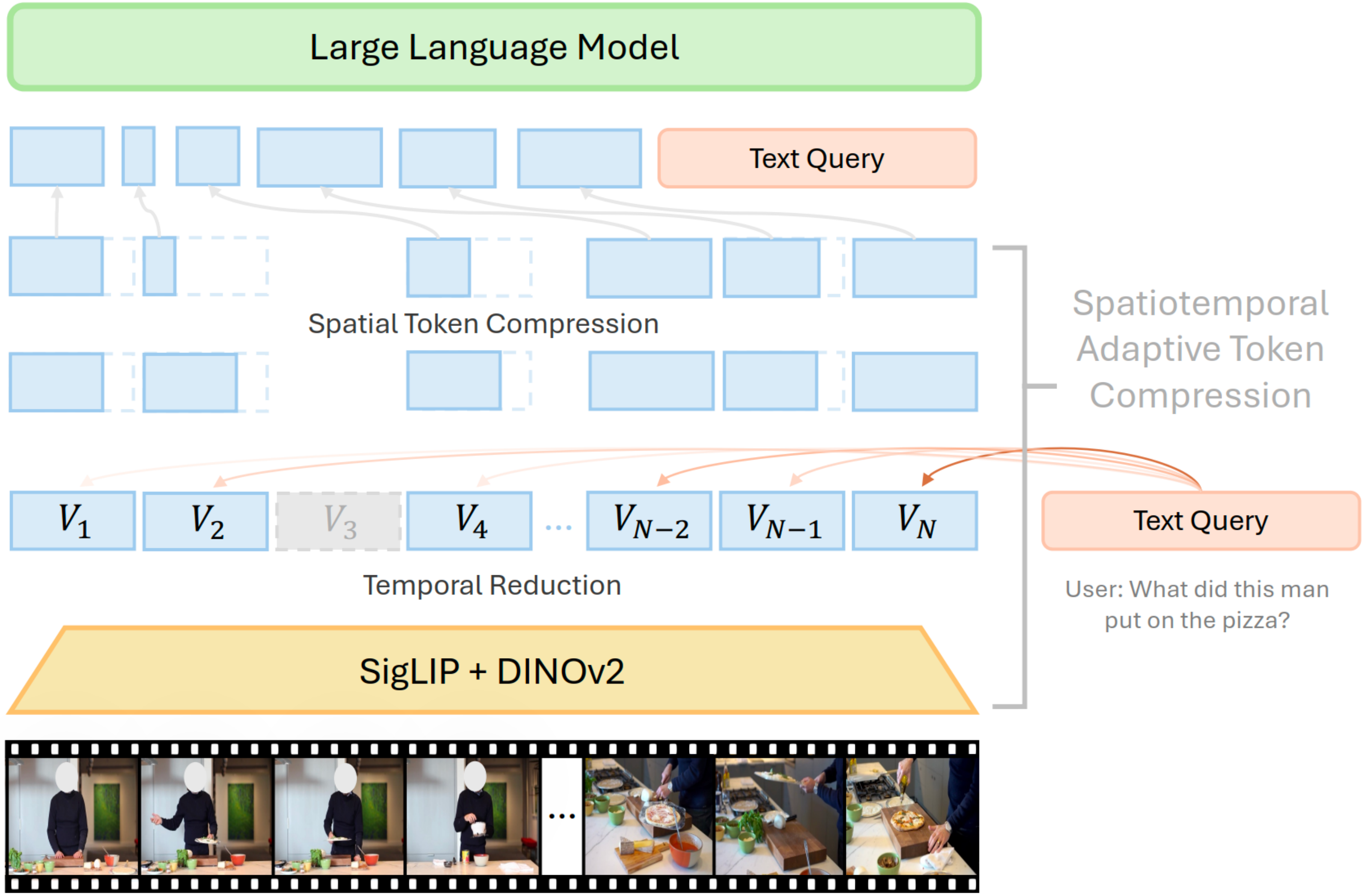
Ye et al., VoCo-LLaMA: Towards Vision Compression with Large Language Models, in *CVPR2025*.

Efficient VLMs with Visual Token Compression

Model-driven Video method

LongVU





Step1: Temporal Reduction: DINOv2

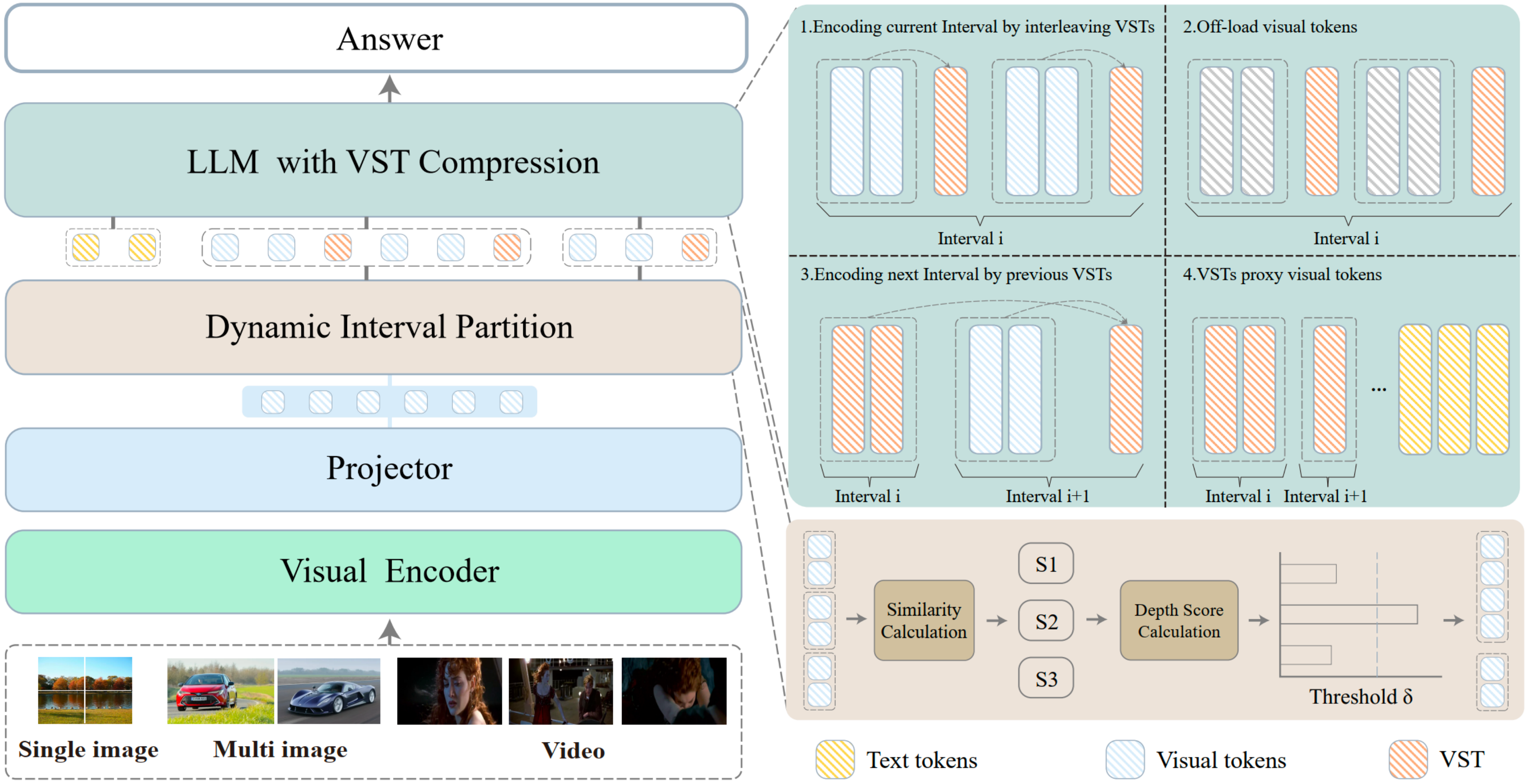
Step2: Selective Feature Reduction via Cross-modal Query

Step3: Spatial Token Compression (STC): pixel-level

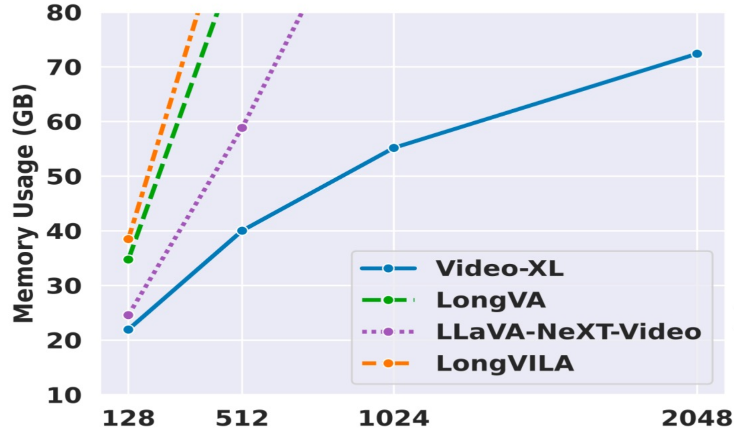
LongVU: Spatiotemporal Adaptive Compression for Long Video-Language Understanding, in *ICML2025*. **Meta & KAUST**

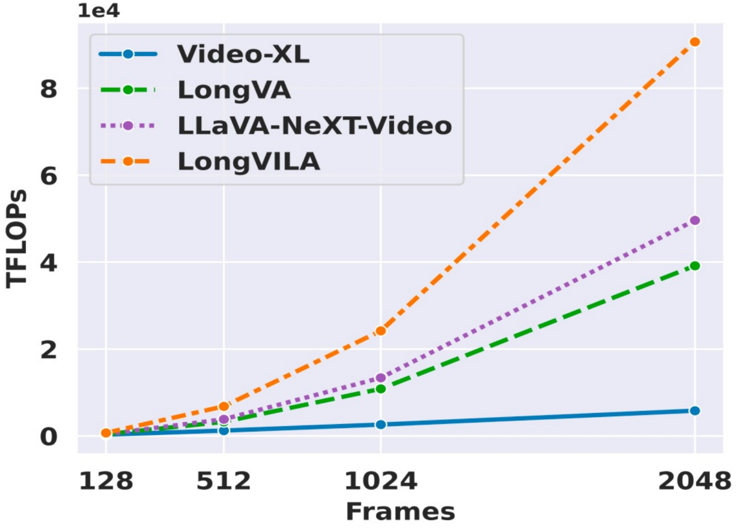
Efficient VLMs with Visual Token Compression

VideoXL



Data-Driven Video Method





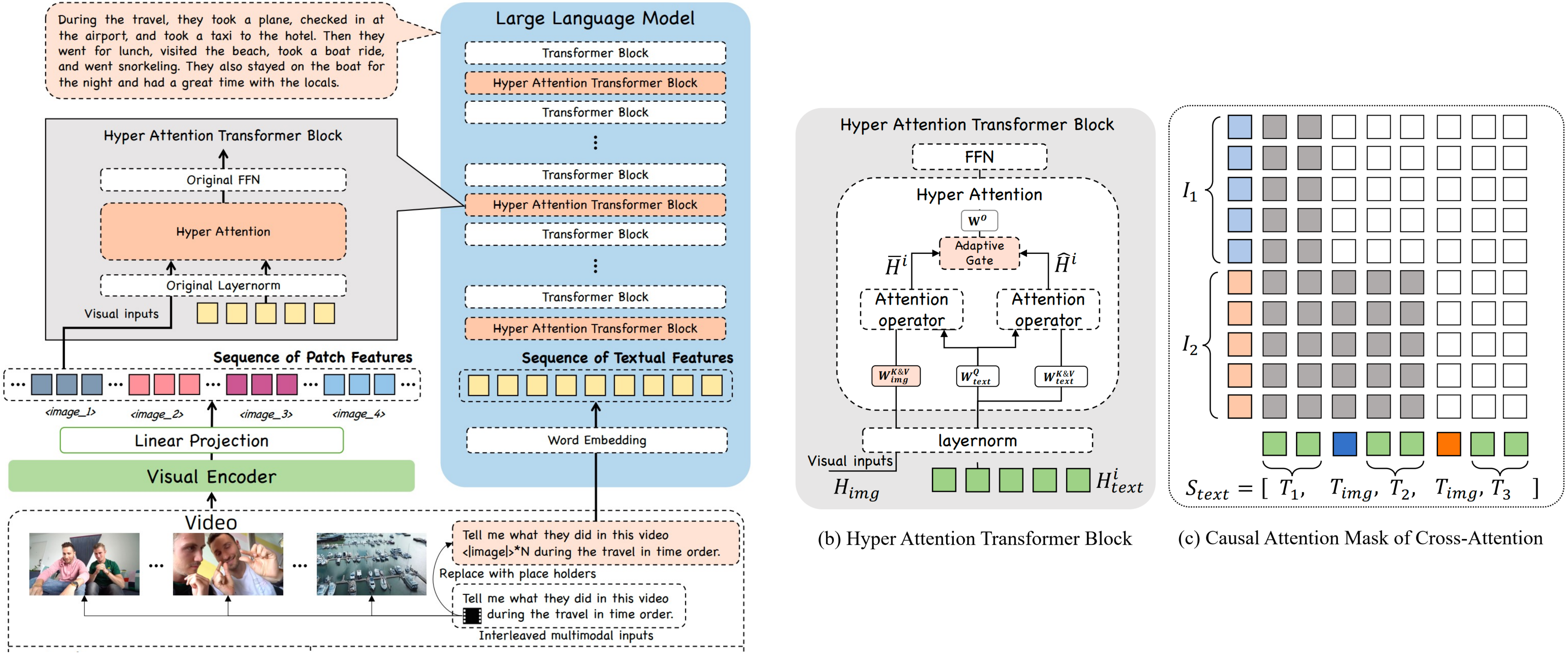
Video-XL: Extra-Long Vision Language Model for Hour-Scale Video Understanding, in *CVPR2025*.

SJTU

Efficient VLMs with Visual Token Compression

Other paradigm

mPLUG-Owl3: Only **input text token** and fuse visual tokens within attention block



(b) Hyper Attention Transformer Block

(c) Causal Attention Mask of Cross-Attention

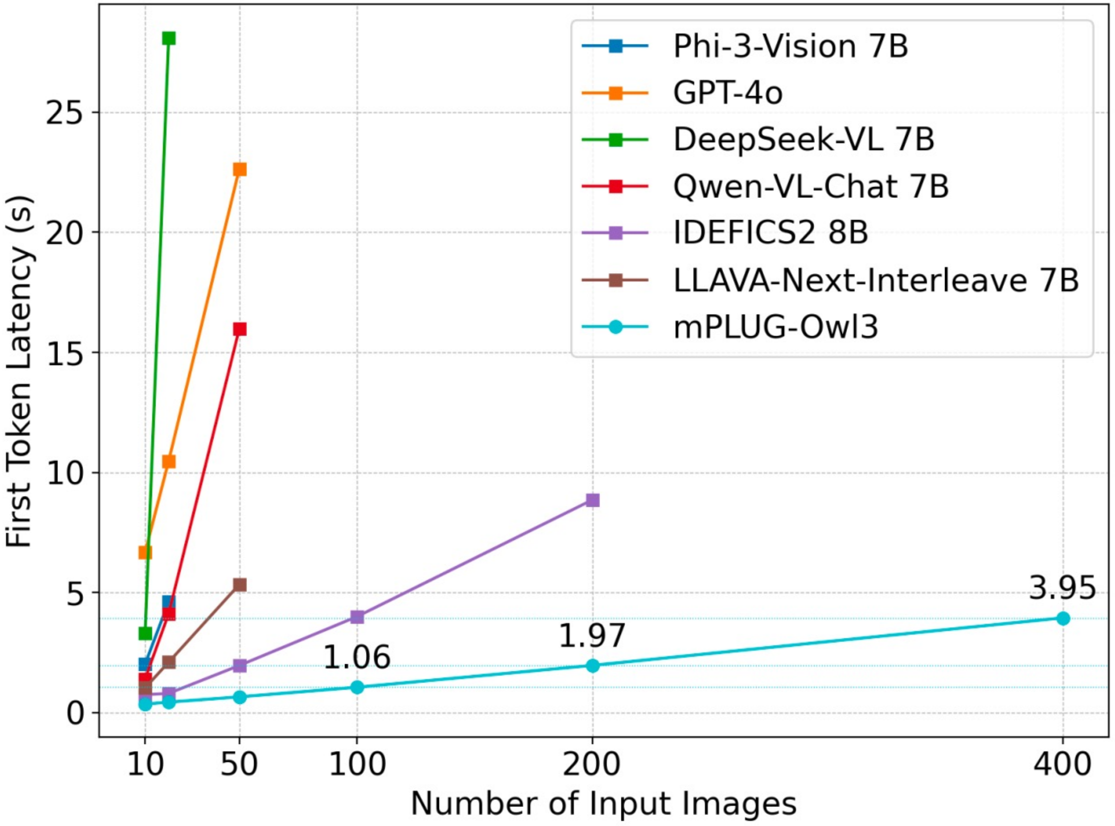
Alibaba Tongyi

mPLUG-Owl3: Towards Long Image-Sequence Understanding in Multi-Modal Large Language Models, in *ICLR2025*.

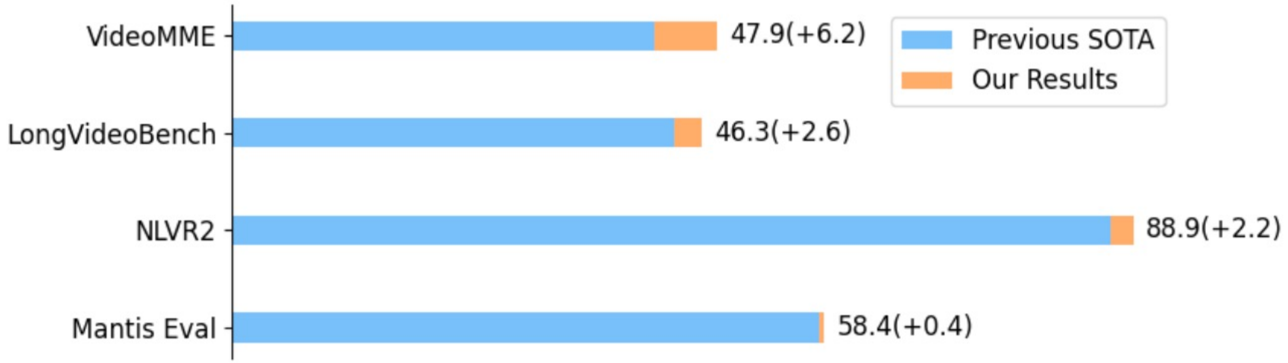
Efficient VLMs with Visual Token Compression

Other paradigm

Efficiency Comparisons



Performance Comparisons



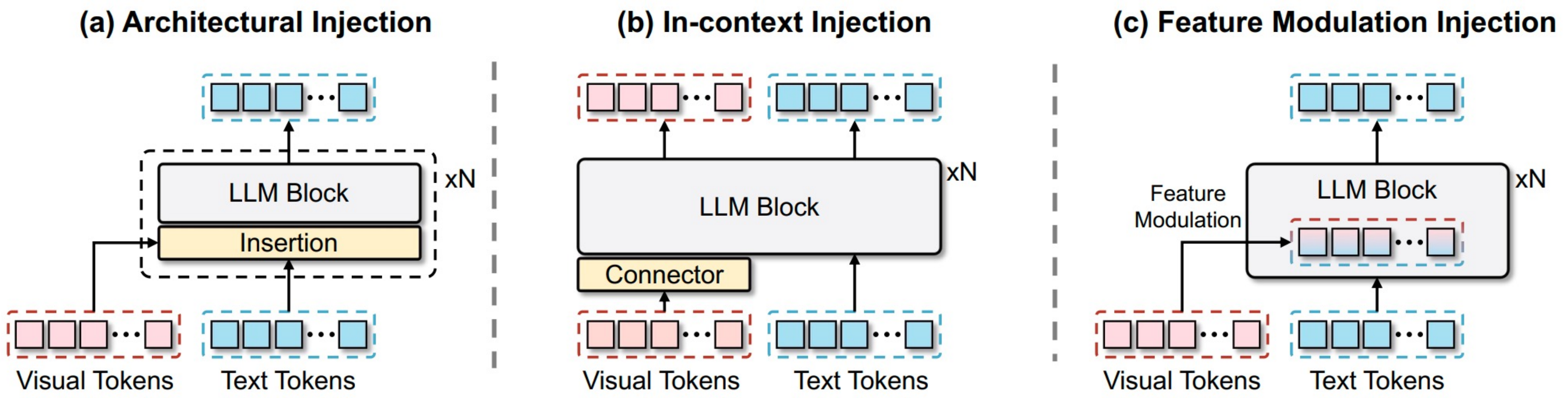
mPLUG-Owl3: Towards Long Image-Sequence Understanding in Multi-Modal Large Language Models, in *ICLR2025*. **Alibaba Tongyi**

Efficient VLMs with Visual Token Compression

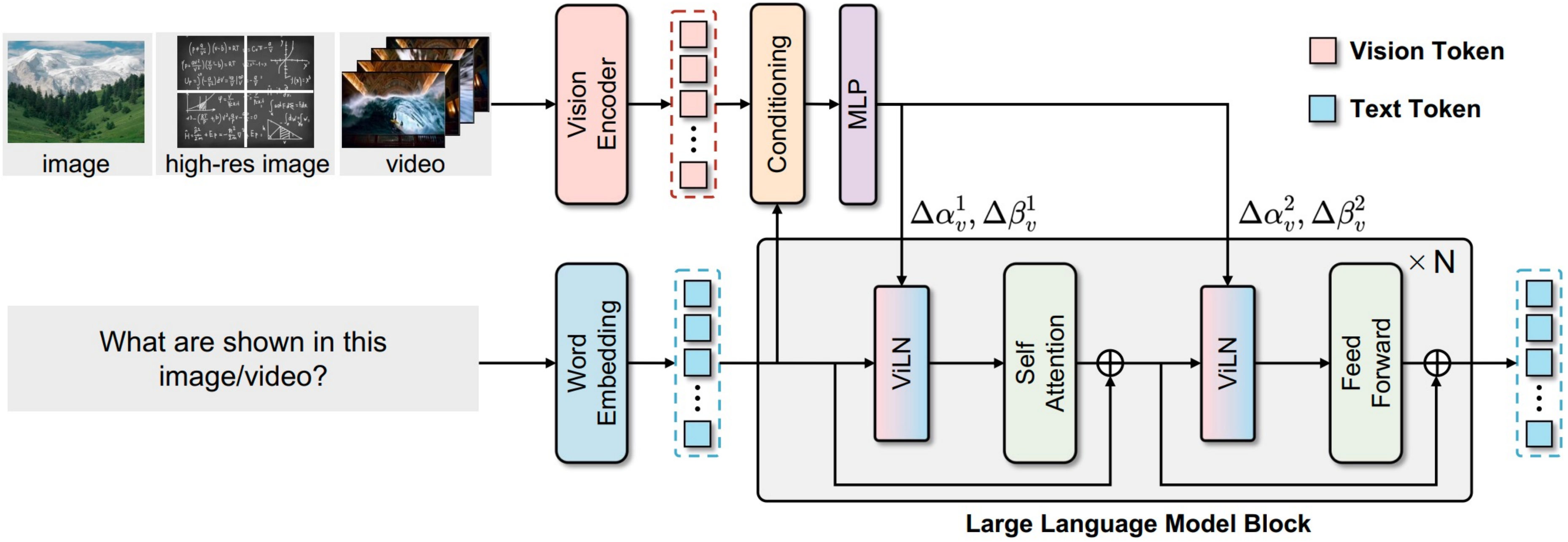
Other paradigm

LaVi

Comparison with
Current methods



LaVi Framework



LaVi: Efficient Large Vision-Language Models via Internal Feature Modulation, *arXiv:2506.16691*.

Efficient VLMs with Visual Token Compression

Other paradigm

Feature Modulation Injection

Core insight: Vision-Infused Layer Normalization

Standard LN: $$\mathrm{LN}(t) = \alpha \odot \hat{t} + \beta$$ α and β are learnable affine parameters

ViLN: $$\mathrm{ViLN}(t, v) = (\alpha + \Delta\alpha_v) \odot \hat{t} + (\beta + \Delta\beta_v)$$

Δα_v and Δβ_v are *vision-conditioned deltas* generated from visual input v.

One before self-attention and One before FFN:

$$[\Delta\alpha_v^1, \Delta\beta_v^1, \Delta\alpha_v^2, \Delta\beta_v^2] = \mathrm{Swish}(\mathrm{Cond}(t, v)) \cdot W + b$$

Three Types of Conditioning Modules:

$$\mathrm{Cond}_{mlp}(t_i, \boldsymbol{v}) = \Big[\mathbf{MLP}_{channel}\big((\mathbf{MLP}_{token}([\,t_i; \boldsymbol{v}\,]^\top))^\top\big)\Big]_{t_i}$$

$$\mathrm{Cond}_{conv}(t_i, \boldsymbol{v}) = \Big[\mathbf{Conv}_{point}\big(\sigma(\mathbf{Conv}_{depth}([\,t_i; \boldsymbol{v}\,]))\big)\Big]_{t_i}$$

$$\mathrm{Cond}_{attn}(t_i, \boldsymbol{v}) = \mathrm{Attention}(t_i\mathbf{W}_Q, \boldsymbol{v}\mathbf{W}_K, \boldsymbol{v}\mathbf{W}_V)$$

LaVi: Efficient Large Vision-Language Models via Internal Feature Modulation, *arXiv:2506.16691*.

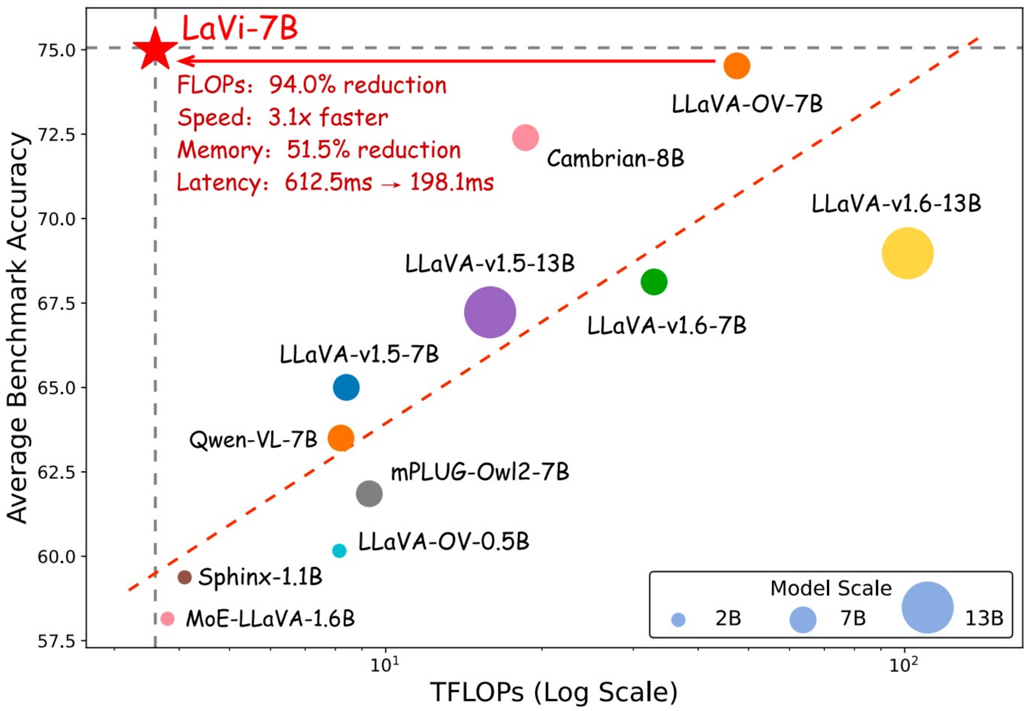
Efficient VLMs with Visual Token Compression

Other paradigm

Performance Comparisons

Method	LLM	Efficiency		Performance									
		FLOPs	Latency	VQA^{v2}	GQA	VisWiz	SciQA	VQA^{T}	POPE	MME^{P}	MMB	$SEED^{I}$	Avg.
Baselines with ≤ 2B parameters scale													
MoE-LLaVA [36]	StableLM-1.6B	3.8	206.4	76.0	60.4	37.2	62.6	47.8	84.3	65.0	59.4	–	–
MobileVLM-V2 [18]	MLLaMA-1.4B	4.3	214.9	–	59.3	–	66.7	52.1	84.3	65.1	57.7	–	–
SPHINX-tiny [38]	TLLaMA-1.1B	4.1	212.3	74.7	58.0	49.2	21.5	57.8	82.2	63.1	56.6	25.2	54.3
LLaVA-OV [27]	Qwen2-0.5B	7.8	228.0	78.5	58.0	51.4	67.2	65.9	86.0	61.9	52.1	65.5	65.2
Baselines with ≤ 8B parameters scale													
Qwen-VL-Chat [8]	Qwen-7B	8.2	239.4	78.2	57.5	38.9	68.2	61.5	–	74.4	60.6	65.4	–
mPLUG-Owl2 [65]	LLaMA2-7B	9.3	278.6	79.4	56.1	54.5	68.7	54.3	–	72.5	64.5	57.8	–
Cambrian-1 [57]	LLaMA3-8B	18.6	393.7	–	64.6	–	80.4	71.7	–	77.4	75.9	74.7	–
LLaVA-v1.5 [39]	Vicuna-7B	8.4	254.4	78.5	62.0	50.0	66.8	58.2	85.9	75.5	64.3	66.1	67.5
LLaVA-v1.6 [40]	Vicuna-7B	32.9	502.4	81.8	64.2	57.6	70.1	64.9	86.5	76.0	67.4	70.2	71.0
LLaVA-OV [27]	Qwen2-7B	60.4	612.5	84.5	62.2	53.0	96.0	76.1	87.4	79.0	80.8	75.4	77.2
Ours													
LaVi-Image	Vicuna-7B	0.6	110.8	79.6	63.0	52.9	67.8	58.4	86.9	75.2	64.8	67.5	68.5
Δ compare to LLaVA-v1.5		7.1%	43.6%	+1.1	+1.0	+2.9	+1.0	+0.2	+1.0	-0.3	+0.5	+1.4	+1.0
LaVi-Image (HD)	Vicuna-7B	1.7	148.6	81.4	63.7	57.8	71.7	64.3	87.0	77.5	68.1	71.6	71.5
Δ compare to LLaVA-v1.6		5.2%	29.6%	-0.4	-0.5	+0.2	+1.6	-0.6	+0.5	+1.5	+0.7	+1.4	+0.5
LaVi	Qwen2-7B	3.6	198.1	84.0	65.0	53.8	95.4	77.0	87.1	80.9	79.3	76.9	77.7
Δ compare to LLaVA-OV		6.0%	32.3%	-0.5	+2.8	+0.8	-0.6	+0.9	-0.3	+1.9	-1.5	+1.5	+0.5

Efficiency Comparisons



Without comparison with mPLUG-Owl3

IACAS

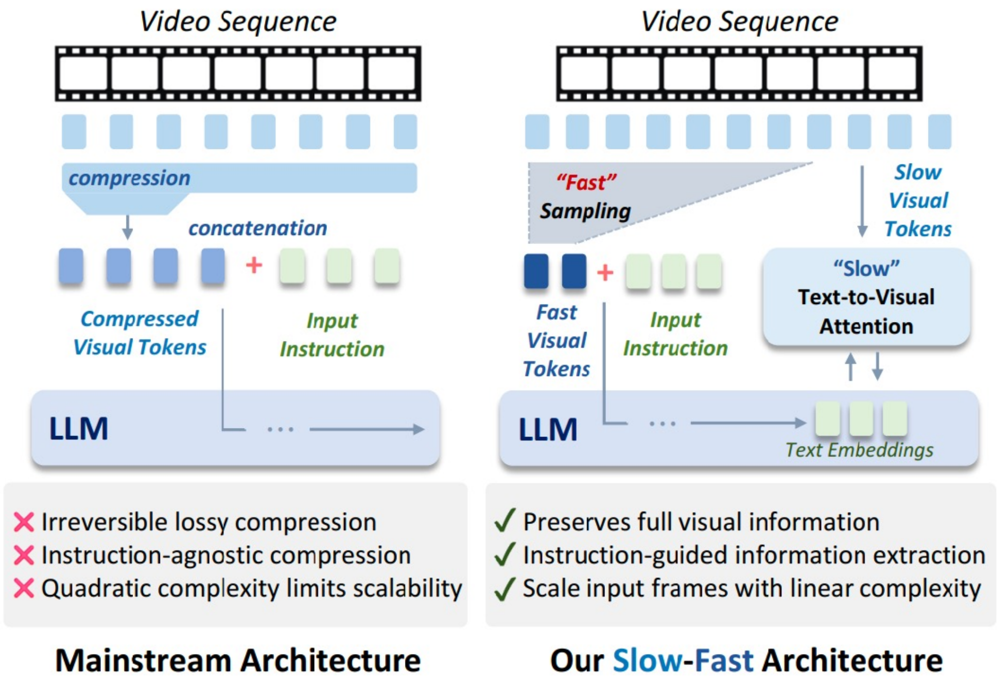
Yue et al., LaVi: Efficient Large Vision-Language Models via Internal Feature Modulation, *arXiv:2506.16691*. (NOT Open-soured)

Efficient VLMs with Visual Token Compression

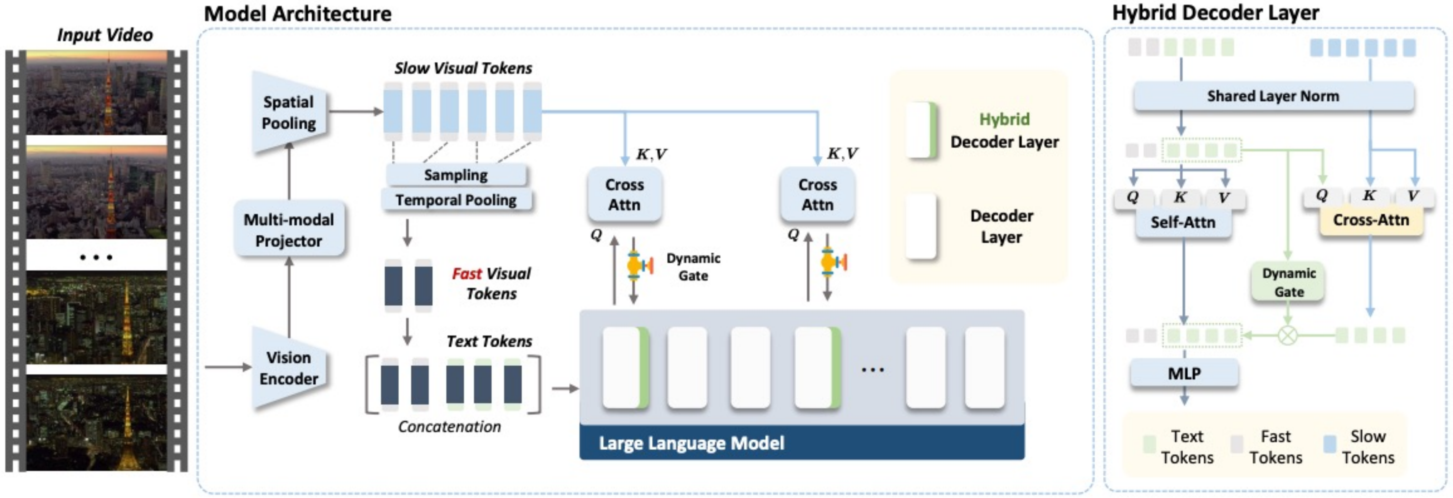
Slow-fast MLLLM

Other paradigm Video method

Comparison with Current method



Framework



Slow-Fast Architecture for Video Multi-Modal Large Language Models, *arXiv:2504.01328*.

SHI Labs &NVIDIA

Efficient VLMs with Visual Token Compression

Conclusion and Future direction

- **Model-driven Approaches**

  Numerous recent studies have emerged, though the potential for further improvement is becoming limited—particularly for image-based VLMs.

- **Data-driven Approaches**

  Demonstrate significant advantages when dealing with extremely fewer visual tokens;

  Develop large-scale token ranking datasets;

  Propose methods with strong generalization capabilities.

- **Other Paradigms**

  Develop more effective **Vision-Infused Modules**;

  Research in this area remains limited, especially for Video-LLMs.

Thanks！

Wentong LI (李文通)

Homepage: https://cslwt.github.io/

Wechat: 17795837723