Shrinking Vision Language Models for Edge Deployment via Distillation and Pruning

Denis Gudovskiy

Distinguished Engineer

Panasonic AI Lab, Mountain View, CA

PSVL

Agenda

Applications using Vision-Language Models (VLMs)

- Our UI agent running on the phone
- LLMs vs. VLMs for physical AI agents

Part 1. Visual token pruning for VLMs

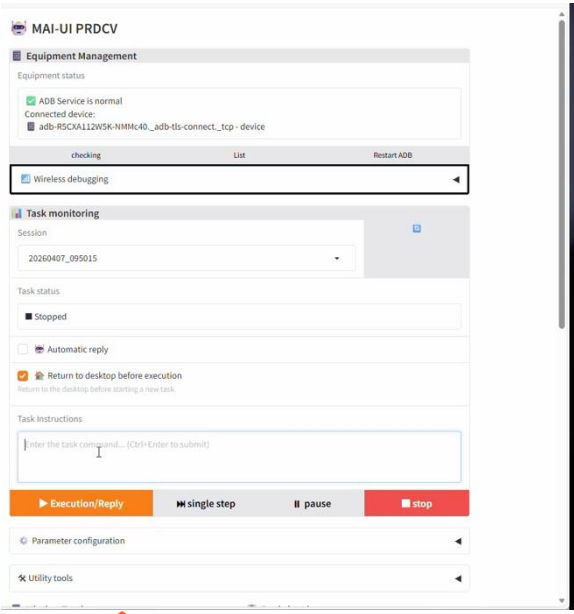
- Overview of recent training-free methods
- Our SparseVLM @ ICML 2025

Part 2. Distilling representations in 3D VLMs

- 3D spatial reasoning using VLMs
- Our Proxy3D @ CVPR 2026

Applications: Our UI Agent

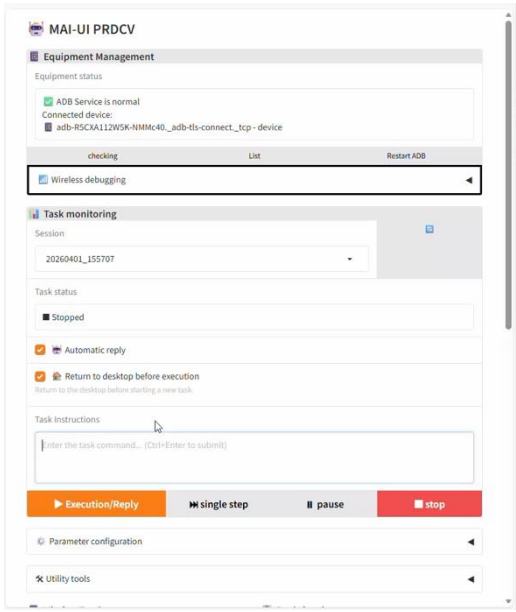
- API for LLM agents is not always available
- Screenshot-based agents are more general

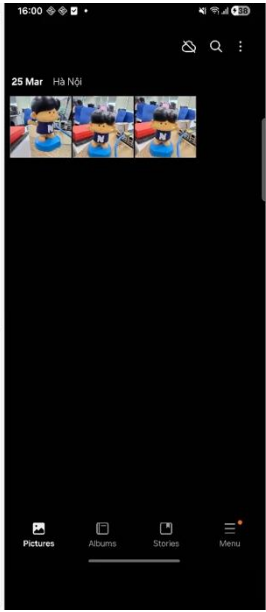


Android Phone
@ ADB

←GitHub actions

Gallery actions→

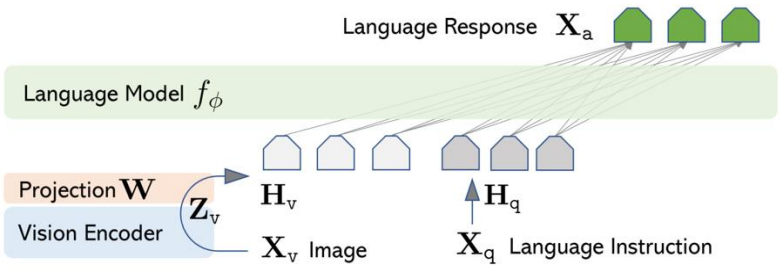




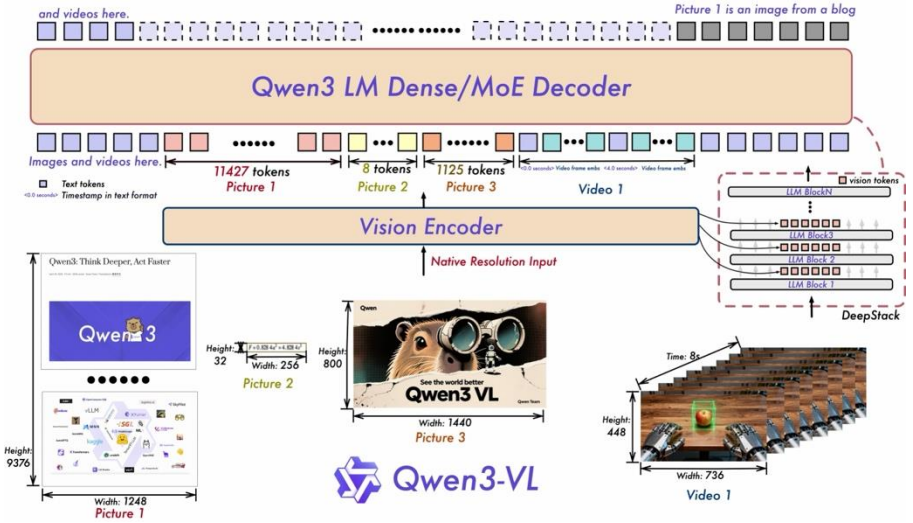
Overview

Vision-Language Models (VLMs)

- Multimodal LLMs (MLLMs) with visual inputs
- Early VLMs had fixed-resolution tokenization
- Recent ones have dynamic variable-lengths



Early VLMs: LLaVA network architecture

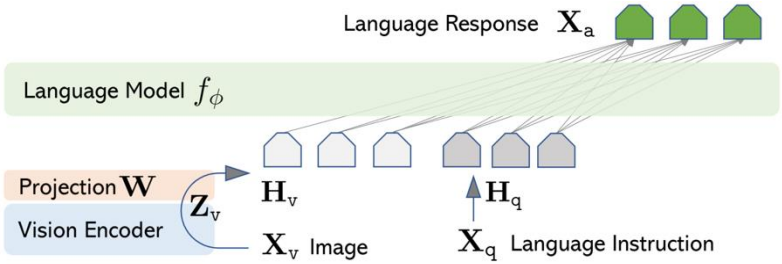


Recent VLMs: Qwen3-VL tokenization scheme

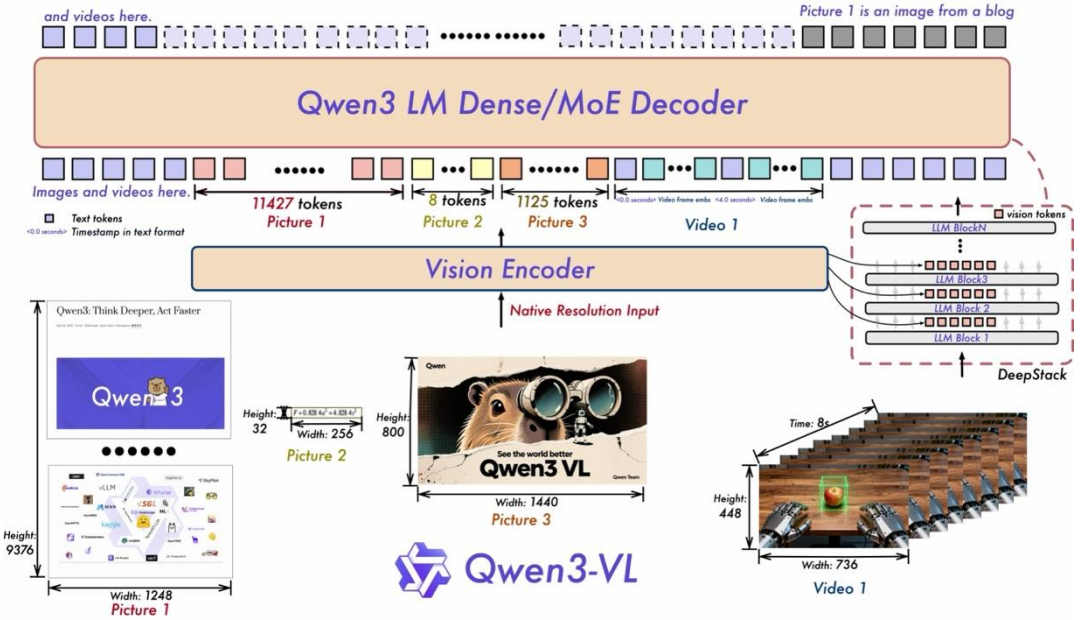
Problem Statement

Exponential growth in number of tokens

- 3D/video >> 2D images >> 1D text
- How to reduce number of tokens?
- How to encode visual modality?



Early VLMs: LLaVA network architecture



Recent VLMs: Qwen3-VL tokenization scheme

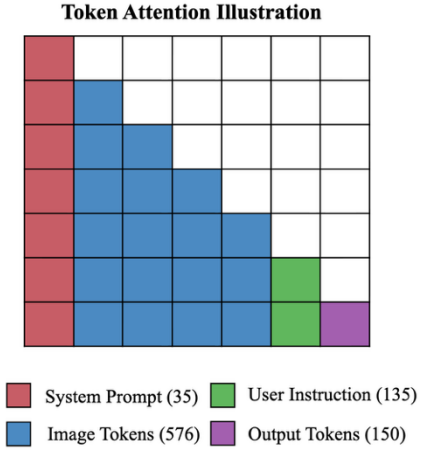
Part 1. Training-free Token Pruning

Visual token pruning

- Self-attention complexity is quadratic with the sequence
- Training-based pruning is hard due to complex pipelines
- Training-free visual token pruning is significantly easier
- Hardware support by edge devices and GPUs

Spatial and temporal redundancy

- Static visual / video inputs are often sparse
- Examples: security cameras, user interfaces
- How to prune tokens in a training-free manner?



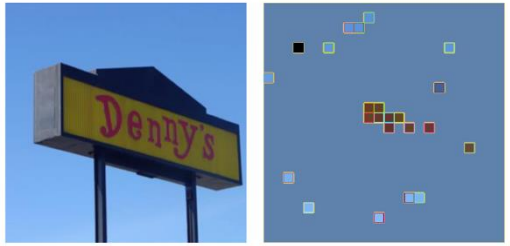
LLaVA self-attention example from FastV

Existing Methods Training-free Methods

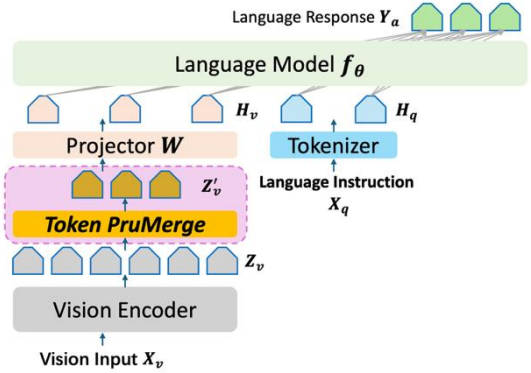
Localization: input vs. layer-wise

Scoring: relevance vs. attention

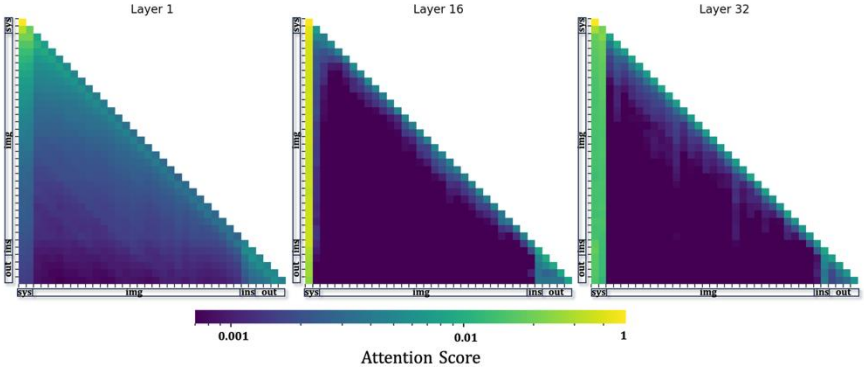
Selection: pruning vs. merging

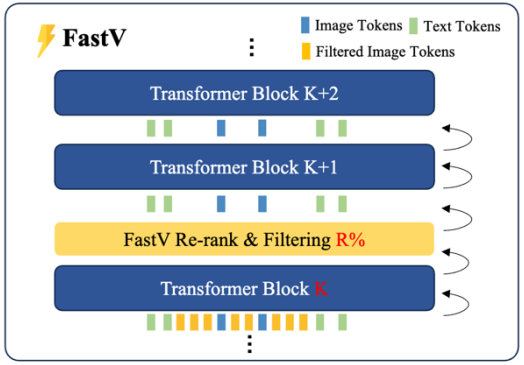


(a) Original Image (b) *PruMerge*



Input pruning in PruMerge



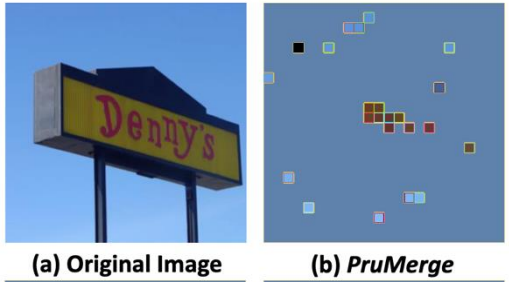


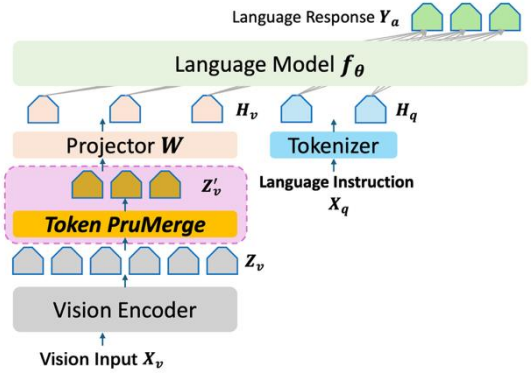
Block-wise pruning in FastV

Existing Methods Training-free Methods

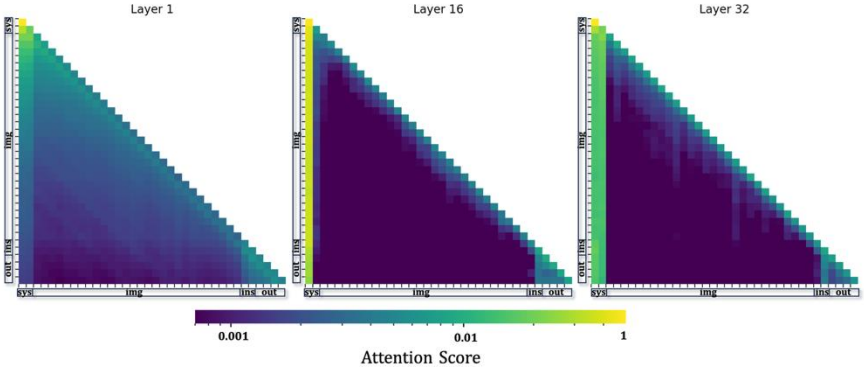
Drawbacks

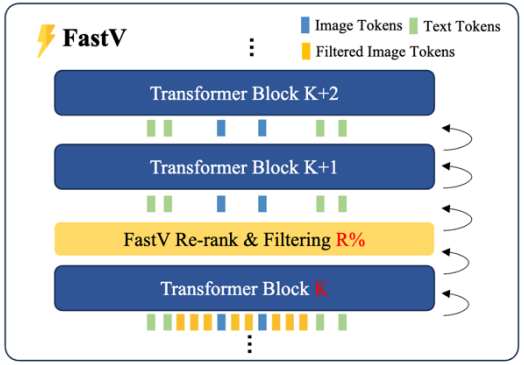
- Text-agnostic methods
- Suboptimal sparsification





Input pruning in PruMerge

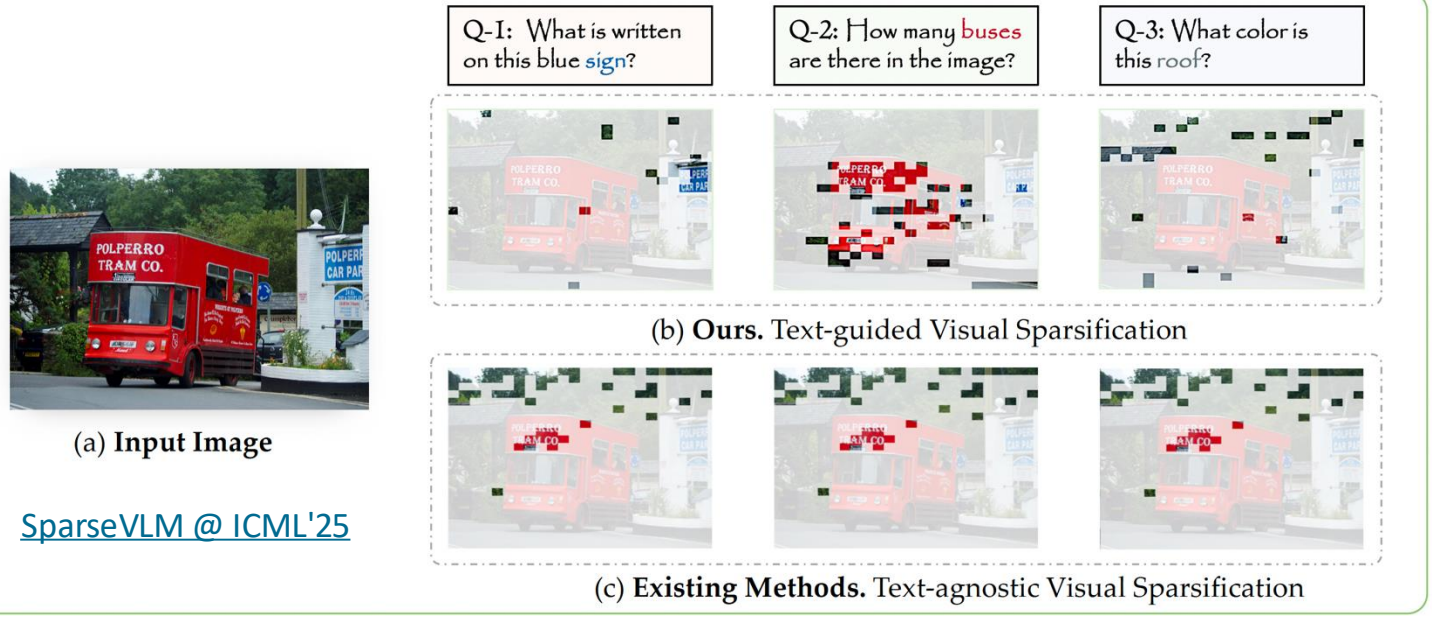




Block-wise pruning in FastV

SparseVLM: Text Prompt Guidance for Pruning

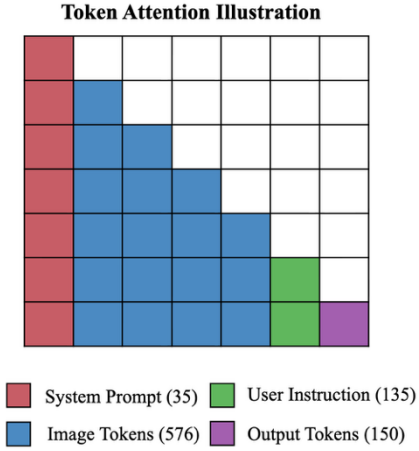
- Queries typically contain questions about specific objects
- Text prompts can be used to guide a VLM sparsification method



SparseVLM @ ICML'25

SparseVLM: Attention Scores for Sparsification

- Main idea is to use transformer's self-attention scores
- Then, we can estimate significance of visual tokens w.r.t. text query
- This is computationally-free and compatible with e.g., FlashAttention



$$P = A_{i,j}, and\ (i,j) \in \{\mathbb{L}, \mathbb{I}\},$$

$$\tilde{p} = [\tilde{p}_1, \tilde{p}_2, \cdots \tilde{p}_{L_v}] = \frac{1}{L_t}\sum_{i=1}^{L_t} P_i\ .$$

How to get the priority matrix?

SparseVLM: Token Selection and Sparsification Level

- Not all text tokens are relevant to the sparsification objective
- Hence, we use “raters” to estimate relevance

$$R = \frac{1}{L_v}\sum_{j=1}^{L_v} Softmax\left(H_v {H_q}^T\right),$$

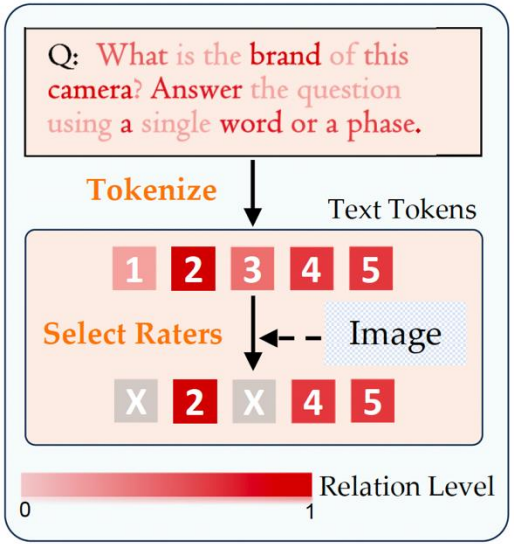
$$S = \{i | R[i] \geq Mean(R), i \in \{1,2,\cdots,L_t\}\}.$$

- Then, the matrix rank can be used to find visual token redundancy

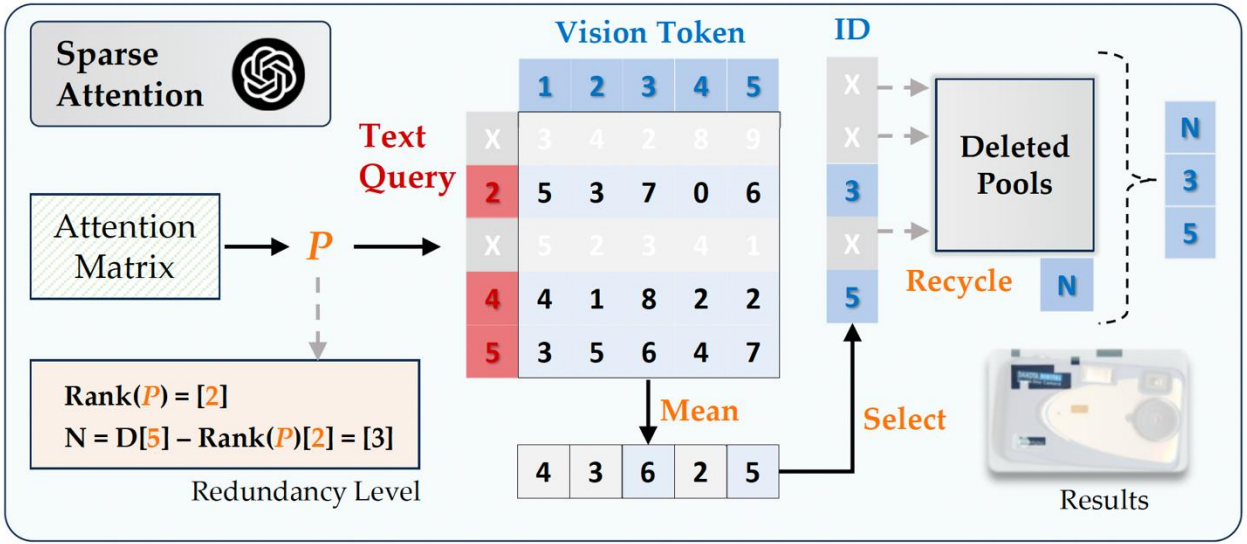
$$N = \lambda \times \left(L_v - Rank(P)\right).$$

SparseVLM: Overall Diagram

- Visual tokens are being pruned at each decoder layer with high sparsification levels
- Therefore, we recycle certain visual tokens by aggregation and reconstruction



(a) Before LLM

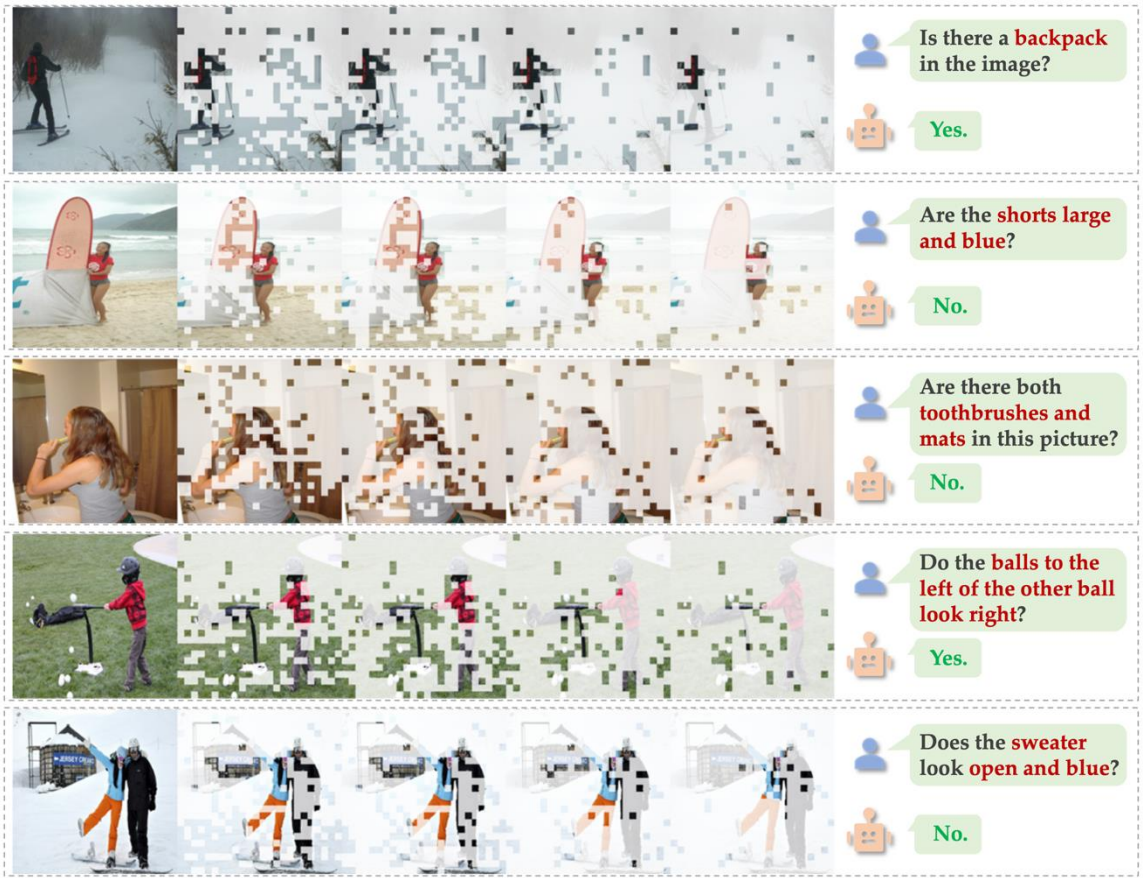


(b) During LLM

SparseVLM: Pruning Results

Method	GQA	MMB	MME	POPE	SQA	SEED	VQA^{Text}	MMVet	Acc. (%)	FLOPs (T)	Latency (ms)
Upper Bound, 576 Tokens **(100%)**											
Vanilla	61.9	64.6	1864	85.9	69.5	60.3	58.3	30.9	100	4.62	57.82
	100%	100%	100%	100%	100%	100%	100%	100%			
Retain 128 Tokens (↓ **77.8%**)											
ToMe (ICLR23)	52.4	53.3	1343	62.8	59.6	50.9	49.1	27.2	81.9 (↓ 18.1)	1.62	30.00
	84.7%	82.4%	72.1%	73.1%	85.8%	84.4%	84.4%	88.0%			
FastV (ECCV24)	49.6	56.1	1490	53.4	68.6	48.1	50.5	26.3	82.4 (↓ 17.6)	1.70	30.70
	80.1%	86.8%	79.9%	62.2%	98.7%	79.8%	86.6%	85.1%			
PDrop (CVPR25)	56.0	61.1	1664	82.3	69.9	53.3	55.1	30.8	94.3 (↓ 5.7)	1.62	37.77
	90.5%	95.4%	89.3%	95.8%	100.6%	88.4%	94.5%	99.7%			
SparseVLM	58.4	64.5	1746	85.0	68.6	58.2	56.7	29.0	96.7 (↓ **3.3**)	1.72	33.28
	94.3%	99.8%	93.7%	99.0%	98.7%	96.5%	97.3%	93.9%			
Retain 64 Tokens (↓ **88.9%**)											
ToMe (ICLR23)	48.6	43.7	1138	52.5	50.0	44.0	45.3	24.1	71.1 (↓ 28.9)	1.19	26.52
	78.5%	67.5%	61.1%	61.1%	71.9%	73.0%	77.8%	78.0%			
FastV (ECCV24)	46.1	47.2	1255	38.2	68.7	43.7	47.8	19.6	72.0 (↓ 28.0)	1.29	27.30
	74.5%	73.1%	67.3%	44.5%	98.8%	72.5%	82.0%	63.4%			
PDrop (CVPR25)	41.9	33.3	1092	55.9	69.2	40.0	45.9	30.7	73.4 (↓ 26.6)	1.18	43.41
	67.7%	51.6%	58.6%	65.1%	99.6%	66.3%	78.7%	99.4%			
SparseVLM	53.8	60.1	1589	77.5	69.8	52.2	53.4	24.9	89.3 (↓ **10.7**)	1.30	29.89
	86.9%	93.0%	85.2%	90.2%	100.4%	86.6%	91.6%	80.6%			

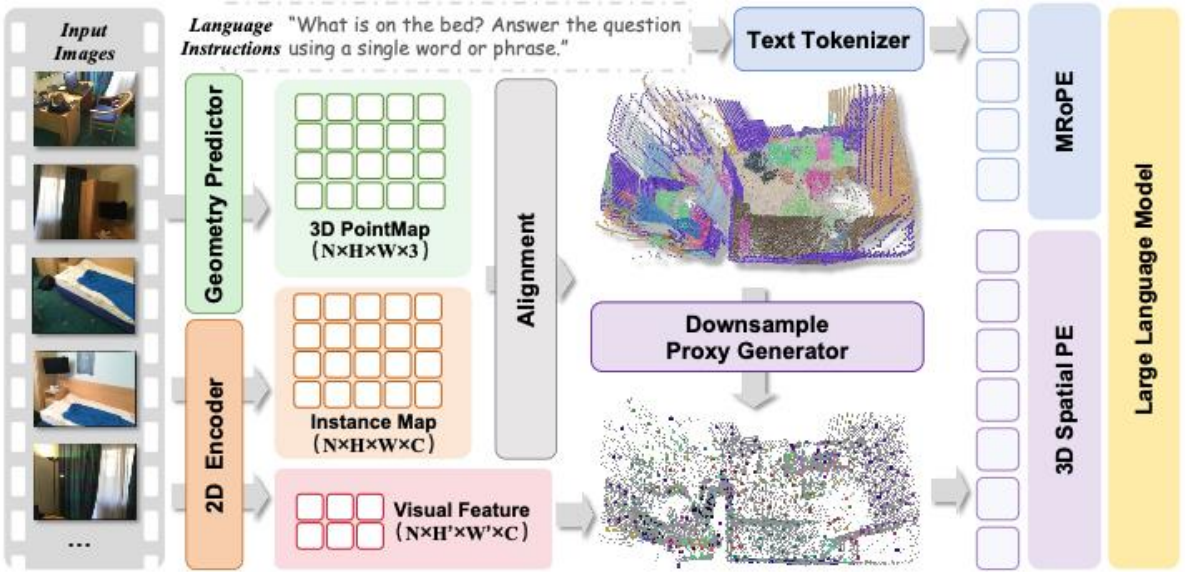
SparseVLM: Pruning Results



Part 2. VLMs for 3D Spatial Reasoning

3D VLMs are an extension of conventional VLMs

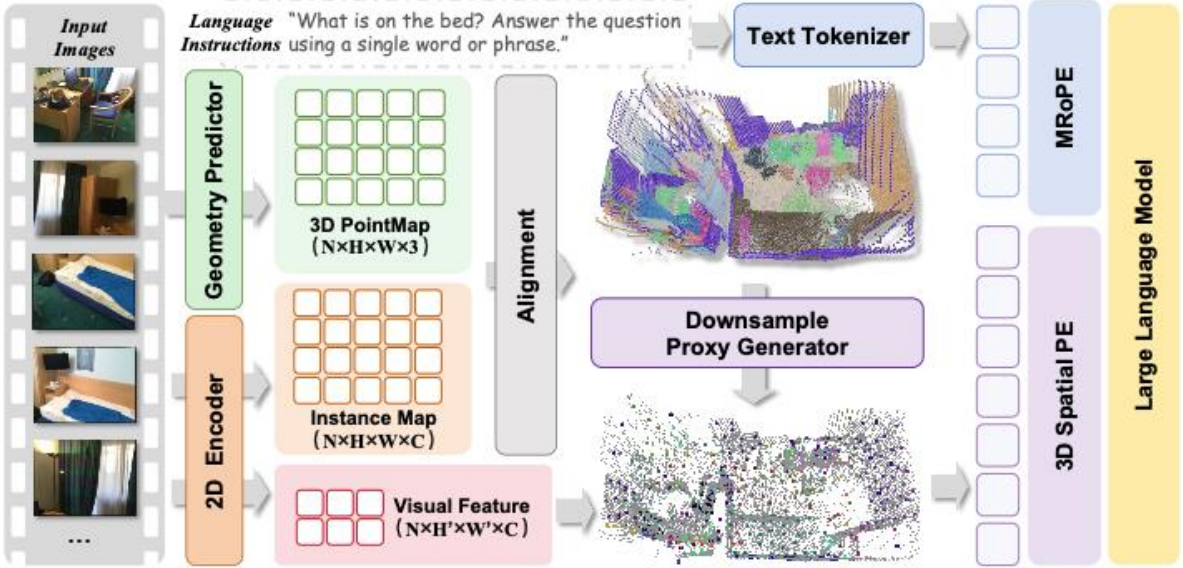
- A set of dense input frames is encoded into a latent token sequence
- How to reduce the number of 3D tokens for the edge deployment?



Comparison to Prior Methods

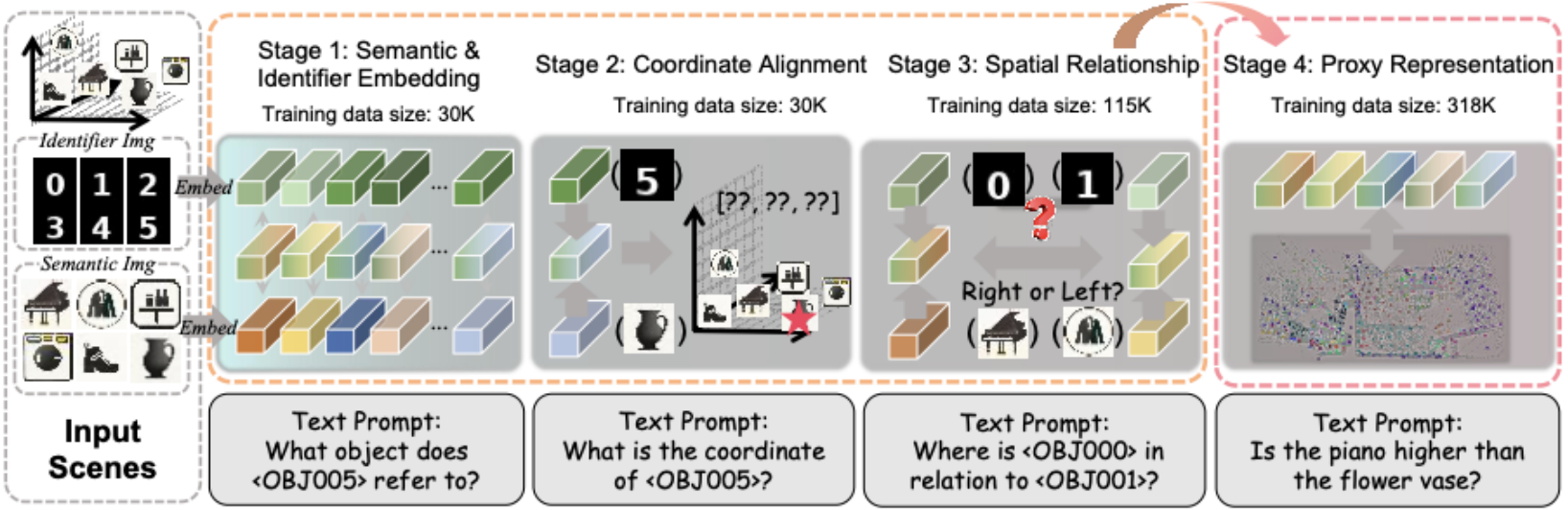
Training-free methods underperform and a distillation training is required

- Representation vs. weight distillation: quadratic vs. linear, KV cache
- Our Proxy3D with the representation distillation via 3D clustering



Proxy3D: Training Pipeline

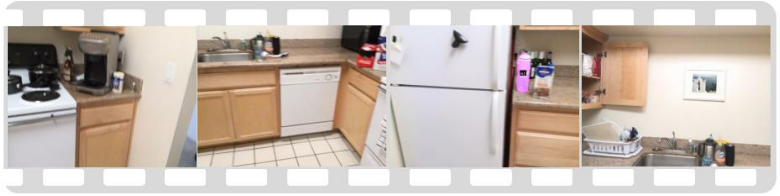
- Distill the pretrained Qwen2.5-VL knowledge to Proxy3D with token clustering
- Main idea: grow 3D spatial reasoning in a sequential manner
- Data scarcity: overcome it using multi-stage training pipeline



Proxy3D: Tasks and Results

3D scene understanding

- Visual grounding, dense scene captioning, question answering



Dense Captioning

Question: Provide a detailed description of the appearance of <OBJ008> before analyzing its spatial connections with other elements in the scene.

Answer: This is a white toilet paper roll. It is to the left of the toilet.

Ground Truth: There is a roll of toilet paper. It is on the wall to the left of the toilet.

Visual Grounding

Question: Are there any objects that correspond to the description "a water bottle is situated on the counter to the right of a refrigerator"? If yes, could you share the IDs for those objects?

Answer: Yes. <OBJ063>.

Ground Truth: <OBJ063>.

Proxy3D: Tasks and Results

Visual-spatial intelligence

- Route planning, relational reasoning, physical agents in the 3D space

Route Plan

Question: You are a robot beginning at the blue chair and facing the fan beside the boxes. You want to navigate to the trash bin. You will perform the following actions (Note: for each [please fill in], choose either 'turn back,' 'turn left,' or 'turn right.'): 1. Go forward until passing the tables on the right. 2. [please fill in] 3. Go forward until the trash bin. You have reached the final destination.

Answer: A. Turn Left B. Turn Back C. Turn Right.

Object Relative Direction (Hard)

Question: If I am standing by the tv and facing the bed, is the table to my front-left, front-right, back-left, or back-right?
The directions refer to the quadrants of a Cartesian plane (if I am standing at the origin and facing along the positive y-axis)

Answer: A. back-left B. back-right C. front-right D. front-left

Proxy3D: Tasks and Results

Models	Vision modal.	# of tokens	Scan2Cap (val) C@0.5	Scan2Cap (val) B-4@0.5	ScanRefer (val) Uni@0.5	ScanRefer (val) Acc@0.25	ScanRefer (val) Acc@0.5	Multi3DRefer (val) F_1@0.25	Multi3DRefer (val) F_1@0.5	ScanQA (val) C	ScanQA (val) EM	SQA3D (val) EM
Correspondence-based:												
Qwen2-VL-7B [40]	I	8000	0	3.8	6.3	5.4	5.1	21.1	19.9	53.9	19.0	40.7
GPT4Scene (HDM)‡ [31]	B,I	8000	86.3	40.6	83.7	62.6	57.0	64.5	59.8	90.9	28.2	57.4
Video-3D-LLM [54]	I	8000	83.8	41.4	78.3	58.1	51.7	58.0	52.7	102.1	30.1	58.6
Spatial-MLLM-4B [42]	I	3100	–	–	–	–	–	–	–	91.8	26.3	55.9
3DRS [18]	I	8000	**86.1**	**41.6**	77.9	**62.9**	**56.1**	60.4	54.9	**104.8**	**30.3**	60.6
Representation-based:												
LLaVA-3D [56]	D,I	3096	79.2	41.1	–	54.1	42.4	–	–	91.7	27.0	55.6
LEO-VL [17]	D,I	750	–	–	–	–	–	–	–	100.4	22.6	**60.8**
Proxy3D$_{\text{Qwen2.5-VL-7B}}$	I	**700**	73.3	34.7	**84.0**	59.6	54.1	**62.0**	**57.5**	93.6	25.2	57.5

Edge deployment benefits

- Latency reduction
- Lower memory usage

Table 3. **Latency** (sec) for batch size of 8 and 16. OOM in table refers to out-of-memory run.

	Vision Encoder		Cluster.		LLM		Latency Per Question	
Batch size	8	16	8	16	8	16	8	16
Qwen2.5-VL	9.4	16.2	-	-	9.3	OOM	2.25	OOM
Ours	9.4	16.2	2.9	5.6	1.1	1.5	**1.75**	**1.5**

Proxy3D: Tasks and Results

EMBEDDED VISION SUMMIT

Specialized 3D VLMs beat proprietary models with the lower complexity

Models	Numerical Answer				Multiple-Choice Answer				Avg.	Rank
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan‡	Appr. Order‡		
Human level	94.3	47.0	60.4	45.9	94.7	95.8	95.8	100.0	79.2	-
Proprietary via API:										
GPT-4o [19]	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5	34.0	8
Gemini-1.5 Pro [36]	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6	45.4	3
Open-source:										
InternVL2-40B [10]	34.9	26.9	46.5	31.8	42.1	32.2	34.0	39.6	36.0	7
LLaVA-OV-72B [21]	43.5	23.9	57.6	37.5	42.5	39.9	32.5	44.6	40.2	5
LLaVA-Video-72B [53]	48.9	22.8	57.4	35.3	42.4	36.7	**35.0**	**48.6**	40.9	4
Qwen2.5-VL-7B [2]	40.9	14.8	43.4	10.7	38.6	38.5	33.0	29.8	33.0	9
Qwen2.5-VL-72B [2]	25.1	29.3	54.5	38.8	38.2	37.0	34.0	28.9	37.0	6
Spatial-MLLM-4B [42]	**65.3**	34.8	63.1	**45.1**	41.3	46.2	33.5	46.3	**48.4**	**1**
Proxy3D$_{\text{Qwen2.5-VL-7B}}$	63.9	**41.9**	**67.2**	42.8	**50.3**	**46.5**	31.4	32.0	47.0	2

References

Part 1:

LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models

An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models

SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference

Part 2:

Spatial-MLLM: Boosting MLLM Capabilities in Visual-based Spatial Intelligence

LEO-VL: Efficient Scene Representation for Scalable 3D Vision-Language Learning

Proxy3D: Efficient 3D Representations for Vision-Language Models via Semantic Clustering and Alignment

Conclusions

Applications using Vision-Language Models

- UI agents with the universal information-sparse inputs
- 3D VLMs for spatial reasoning in dense streaming videos
- All these cases are compute/memory “expensive” for edge devices

Part 1. Visual token pruning for VLMs

- We leveraged spatial and temporal redundancies in images
- SparseVLM with the prompt guidance and attention-based scores

Part 2. Visual-spatial intelligence using 3D VLMs

- 3D VLMs implement perception for physical AI agents
- Proxy3D clustering with the distillation compresses token length by 10×

Questions?