A-VL: Adaptive Attention for Large Vision-Language Models

Junyang Zhang[1], Mu Yuan[1], Ruiguang Zhong[2], Puhan Luo[1], Huiyou Zhan[1],
Ningkang Zhang[1], Chengchen Hu[2], Xiang-Yang Li[1]

[1]University of Science and Technology of China, Hefei, China

[2]NIO Inc., Shanghai, China

中国科学技术大学
University of Science and Technology of China

Lab for Intelligent Networking
and Knowledge Engineering

Outline

- **Introduction**
- Observations and Insights
- Design of A-VL
- Evaluation
- Conclusion

Introduction

Large vision-language model (LVLM) has great application potential.

Intelligent Cockpit

Industrial Detection

Robot Control

Auxiliary Medicine

Introduction

The substantial computational requirements of LVLMs limit their practical application.

For example, intelligent vehicles use embedded computing devices (like AGX Orin) to provide intelligent cockpit services[1].

However, limited computing resources and the requirement for real-time computing may pose challenges.

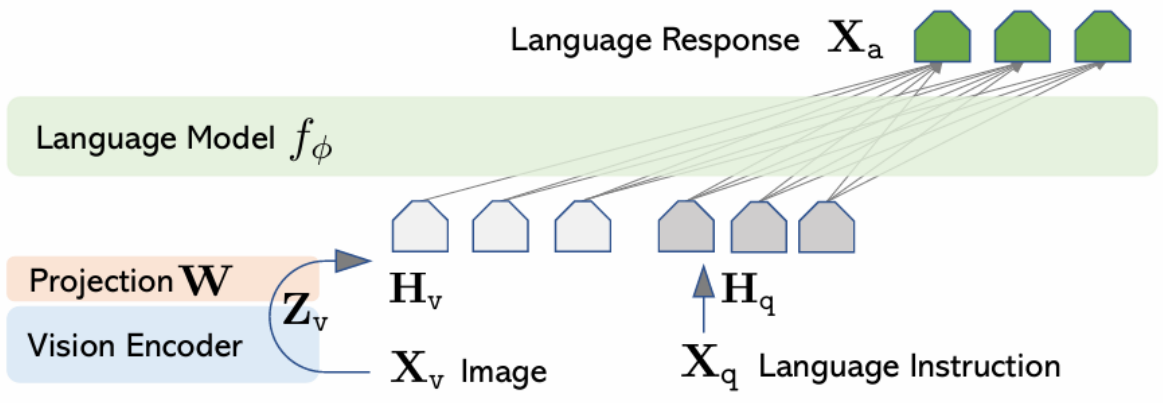
We propose A-VL to reduce memory usage and computational load without compromising performance for LVLMs.

1. https://www.nvidia.cn/self-driving-cars/partners/nio/

Structure of Large Vision-Language Modal

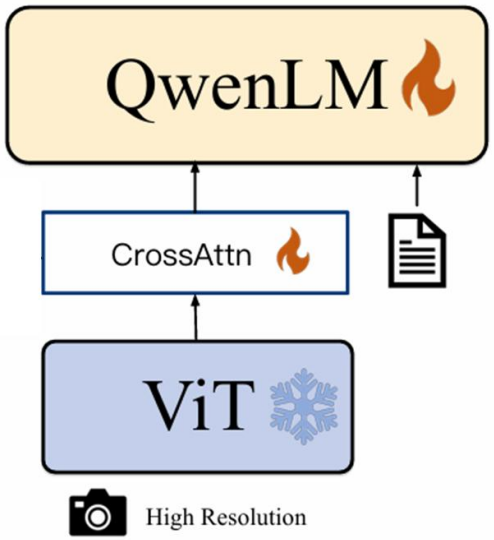
Large Vision-Language Modal Workflow:

1. Encode the multimodal input into hidden features
2. Use backbone LLM to generate answers



Vision Encoder[1]

Take Qwen-VL[2] as an example



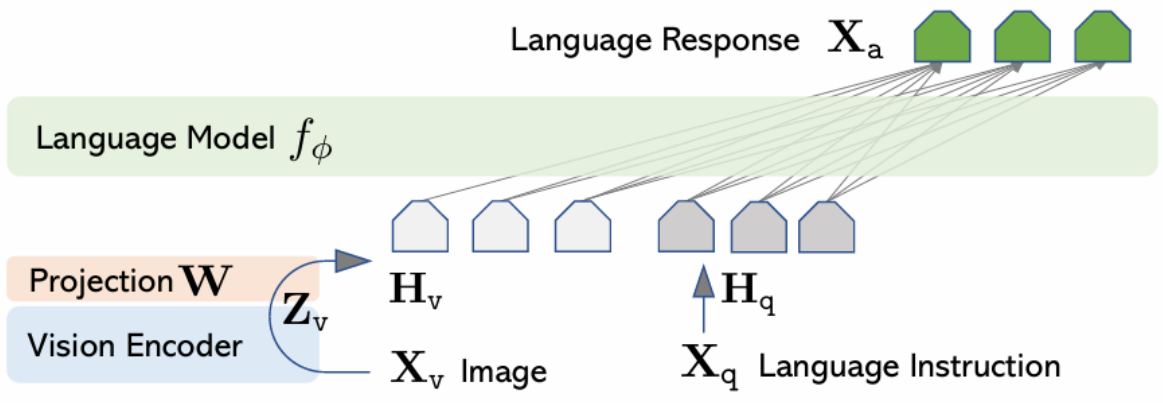
Vision Encoder	VL Adapter	LLM	Total
1.9B	0.08B	7.7B	9.6B

1. Liu, Haotian, et al. "Visual instruction tuning." Advances in neural information processing systems 36 (2024).
2. Bai, Jinze, et al. "Qwen-vl: A frontier large vision-language model with versatile abilities." *arXiv preprint arXiv:2308.12966* (2023).

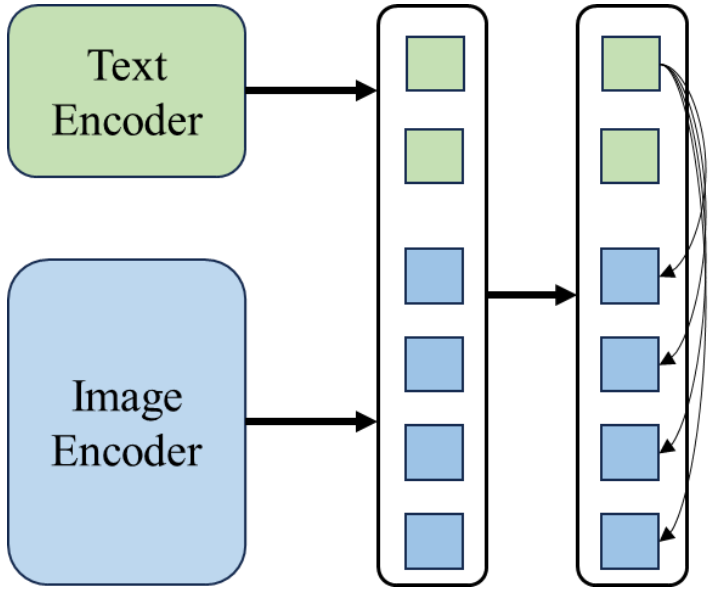
Structure of Large Vision-Language Modal

Large Vision-Language Modal Workflow:

1. Encode the multimodal input into hidden features
2. Use backbone LLM to generate answers



Vision Encoder[1]



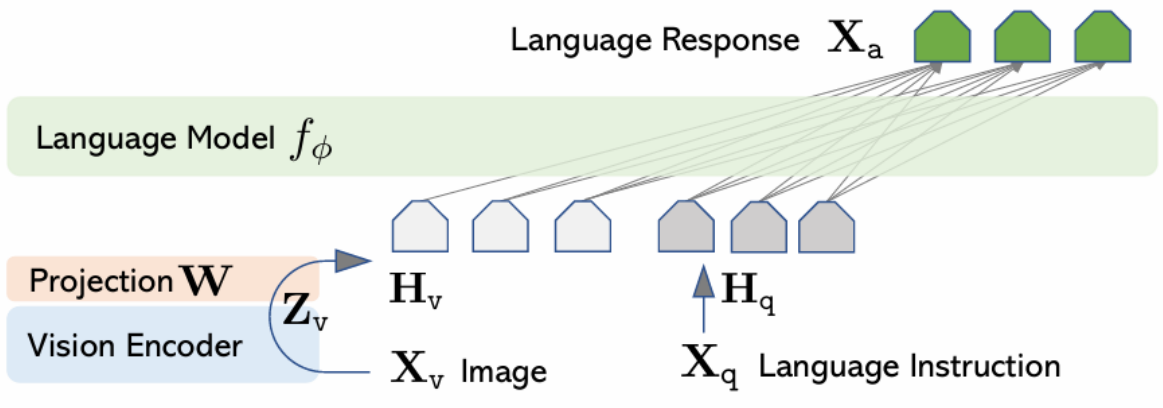
Tokens of different modalities are calculated for Attention relevance in the backbone LLM.

1. Liu, Haotian, et al. "Visual instruction tuning." Advances in neural information processing systems 36 (2024).
2. Bai, Jinze, et al. "Qwen-vl: A frontier large vision-language model with versatile abilities." *arXiv preprint arXiv:2308.12966* (2023).

Structure of Large Vision-Language Modal

Large Vision-Language Modal Workflow:

1. Encode the multimodal input into hidden features
2. Use backbone LLM to generate answers



Vision Encoder[1]

Performance bottleneck:

- The number of image tokens is large. LLaVA 1.6 model can often exceed **2000 tokens** for one image.
- The large number of tokens brings large KV cache usage.

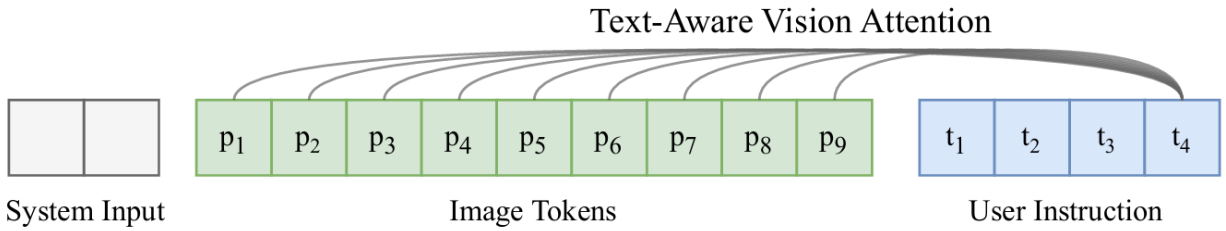
1. Liu, Haotian, et al. "Visual instruction tuning." Advances in neural information processing systems 36 (2024).
2. Bai, Jinze, et al. "Qwen-vl: A frontier large vision-language model with versatile abilities." *arXiv preprint arXiv:2308.12966* (2023).

Outline

- Introduction
- **Observations and Insights**
- Design of A-VL
- Evaluation
- Conclusion

Observations and Insights

Token Importance:



We use the attention score of the latest token to the previous token to evaluate the token importance[1].

Attention score for each token in each layer:

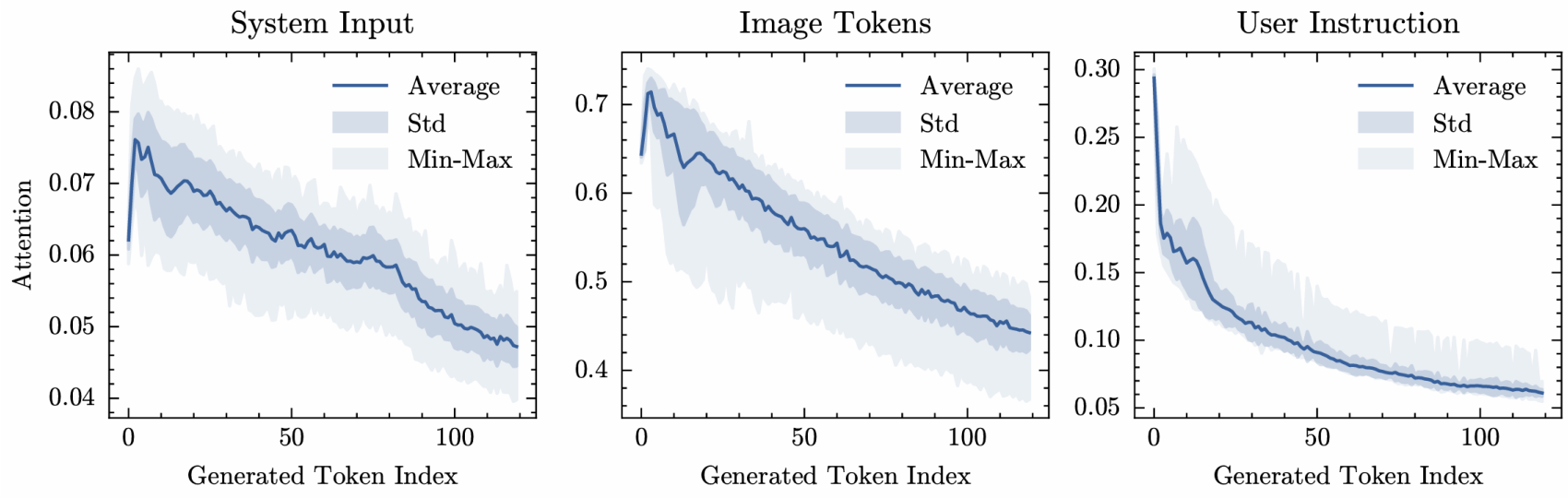
$$s_t^l = \sum_{h=1}^{H} A_t^{l,h} / H$$

- A : attention weights assigned from the last token;
- l : the index of the transformer decoder layers;
- h :indicates the attention head.

1. Chen, Liang, et al. "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.

Observations and Insights

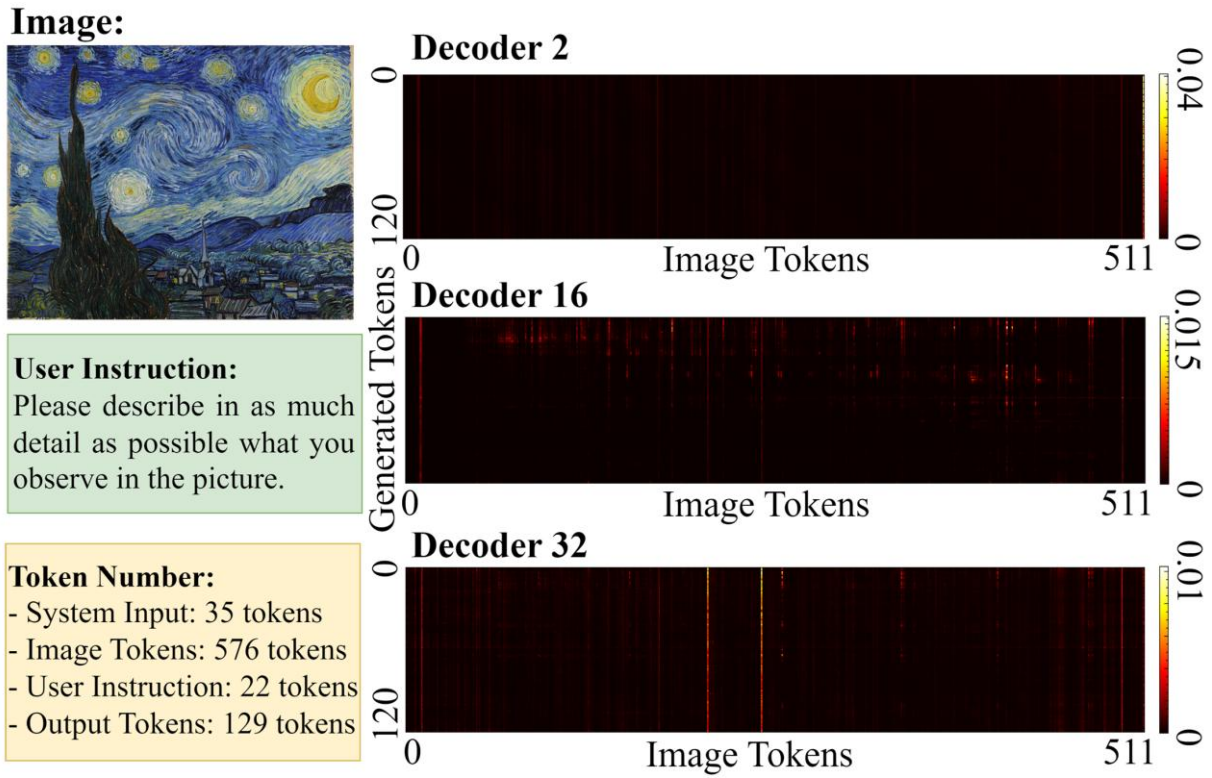
Different modalities have different attention patterns.



Different modalities have different attention patterns.

Observations and Insights

Image Token Attention Characteristics



Characteristic 1: High Sparsity

Most image tokens are allocated little attention.

22.88% image tokens receive $> 80\%$ attention

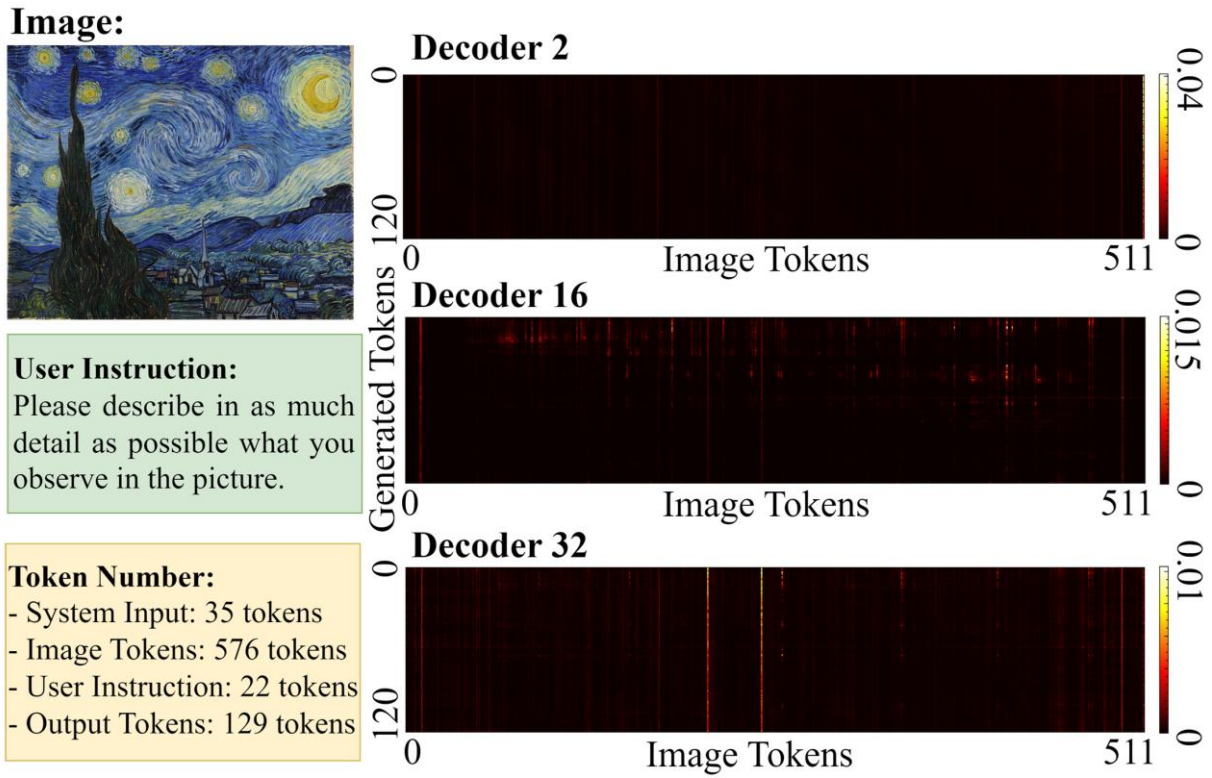
Setting:

- LLaVA-1.5 7B model[1]
- OCRVQA dataset[2]

1. Liu, Haotian, et al. "Visual instruction tuning." Advances in neural information processing systems 36 (2024).
2. Mishra, Anand, et al. "Ocr-vqa: Visual question answering by reading text in images." 2019 international conference on document analysis and recognition (ICDAR). IEEE, 2019.

Observations and Insights

Image Token Attention Characteristics



Characteristic 1: High Sparsity

Most image tokens are allocated little attention.

22.88% image tokens receive $> 80\%$ attention

Setting:

- LLaVA-1.5 7B model[1]
- OCRVQA dataset[2]

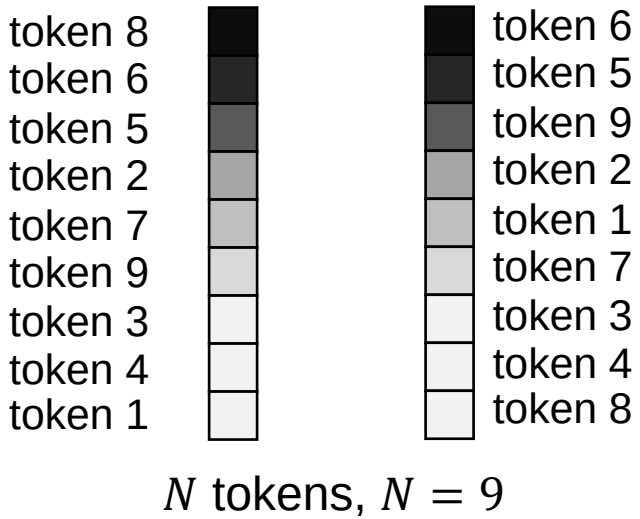
Considering the **high sparsity** of visual attention, we designed an index for measuring the similarity of attention weights that focuses on tokens with high attention scores, named ***p*-percentile concordance index (PPCI)**

Observations and Insights

Image Token Attention Characteristics

Characteristic 1: High Sparsity

Suppose there are two attention scores in different layers:

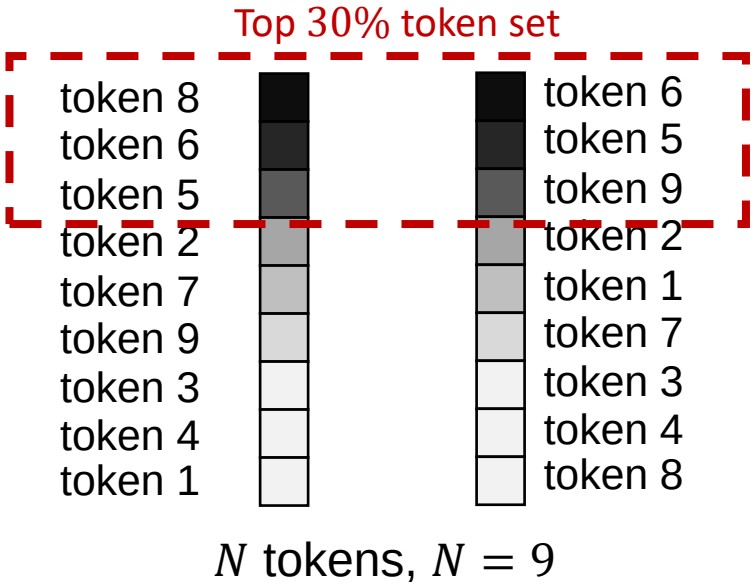


Considering the **high sparsity** of visual attention, we designed an index for measuring the similarity of attention weights that focuses on tokens with high attention scores, named ***p*-percentile concordance index (PPCI)**

Observations and Insights

Image Token Attention Characteristics

Suppose there are two attention scores in different layers:



Characteristic 1: High Sparsity

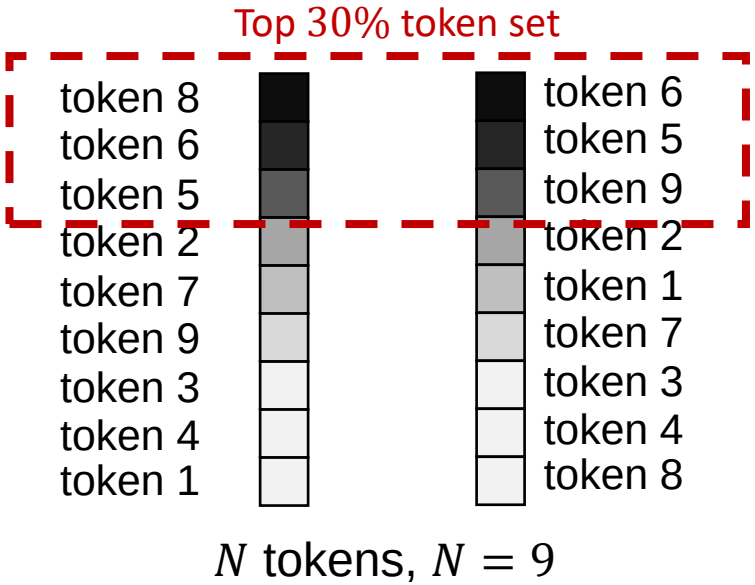
We focus on the top 30% of tokens that allocated the majority of attention.

Considering the **high sparsity** of visual attention, we designed an index for measuring the similarity of attention weights that focuses on tokens with high attention scores, named ***p*-percentile concordance index (PPCI)**

Observations and Insights

Image Token Attention Characteristics

Suppose there are two attention scores in different layers:



Characteristic 1: High Sparsity

We focus on the top 30% of tokens that allocated the majority of attention.

Top 30% token set in two sequence

$$30\%\ PPCI = \frac{|\{8,6,5\} \cap \{6,5,9\}|}{30\% \times N}$$

$$= \frac{|\{6,5\}|}{30\% \times 9} = \frac{2}{3} = 0.66$$

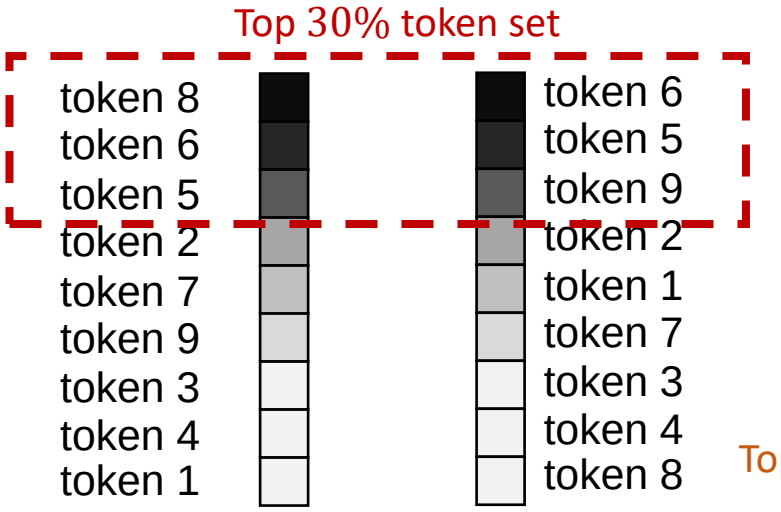
PPCI measures the similarity of the attention scores of two token sequences.

Considering the **high sparsity** of visual attention, we designed an index for measuring the similarity of attention weights that focuses on tokens with high attention scores, named ***p*-percentile concordance index (PPCI)**

Observations and Insights

Image Token Attention Characteristics

Suppose there are two attention scores in different layers:



Characteristic 1: High Sparsity

We focus on the top 30% of tokens that allocated the majority of attention.

Top 30% token set in two sequence

$$30\%\ PPCI = \frac{|\{8,6,5\} \cap \{6,5,9\}|}{30\% \times N}$$

$$= \frac{|\{6,5\}|}{30\% \times 9} = \frac{2}{3} = 0.66$$

Top $p\%$ token set in two sequence

$$p\%\ PPCI = \frac{|T_1 \cap T_2|}{p\% \cdot N}$$

PPCI measures the similarity of the attention scores of two token sequences.

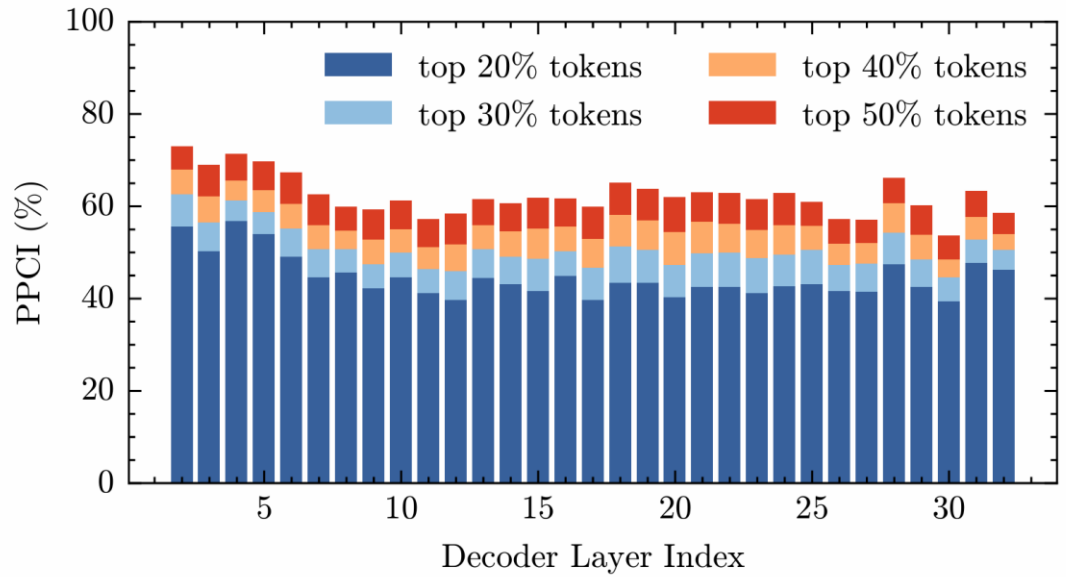
Considering the **high sparsity** of visual attention, we designed an index for measuring the similarity of attention weights that focuses on tokens with high attention scores, named ***p*-percentile concordance index (PPCI)**

Observations and Insights

Image Token Attention Characteristics

Characteristic 2: Heterogeneity among Layers

Different Layer



The correlation of attention between the first decoder layer and each subsequent decoder layer.

Less than 70% of the tokens in the top 50% of the attention scores between each layer and the first layer are the same.

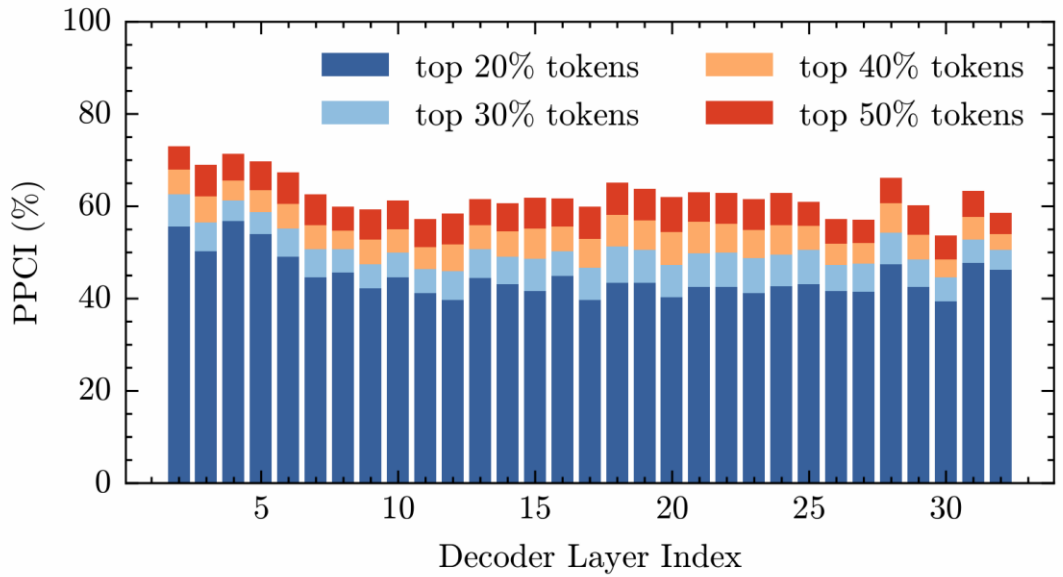
The attention distribution of each layer is not similar.

Observations and Insights

Image Token Attention Characteristics

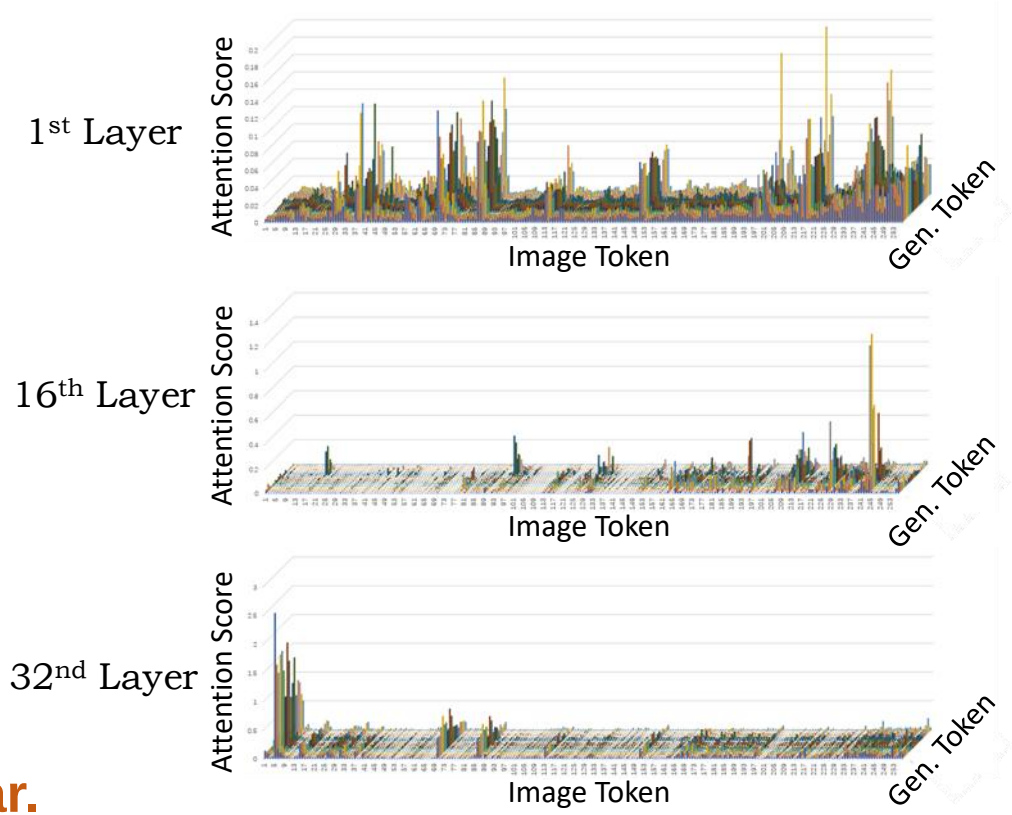
Characteristic 2: Heterogeneity among Layers

Different Layer



The correlation of attention between the first decoder layer and each subsequent decoder layer.

Attention Distribution in Diff. Layer



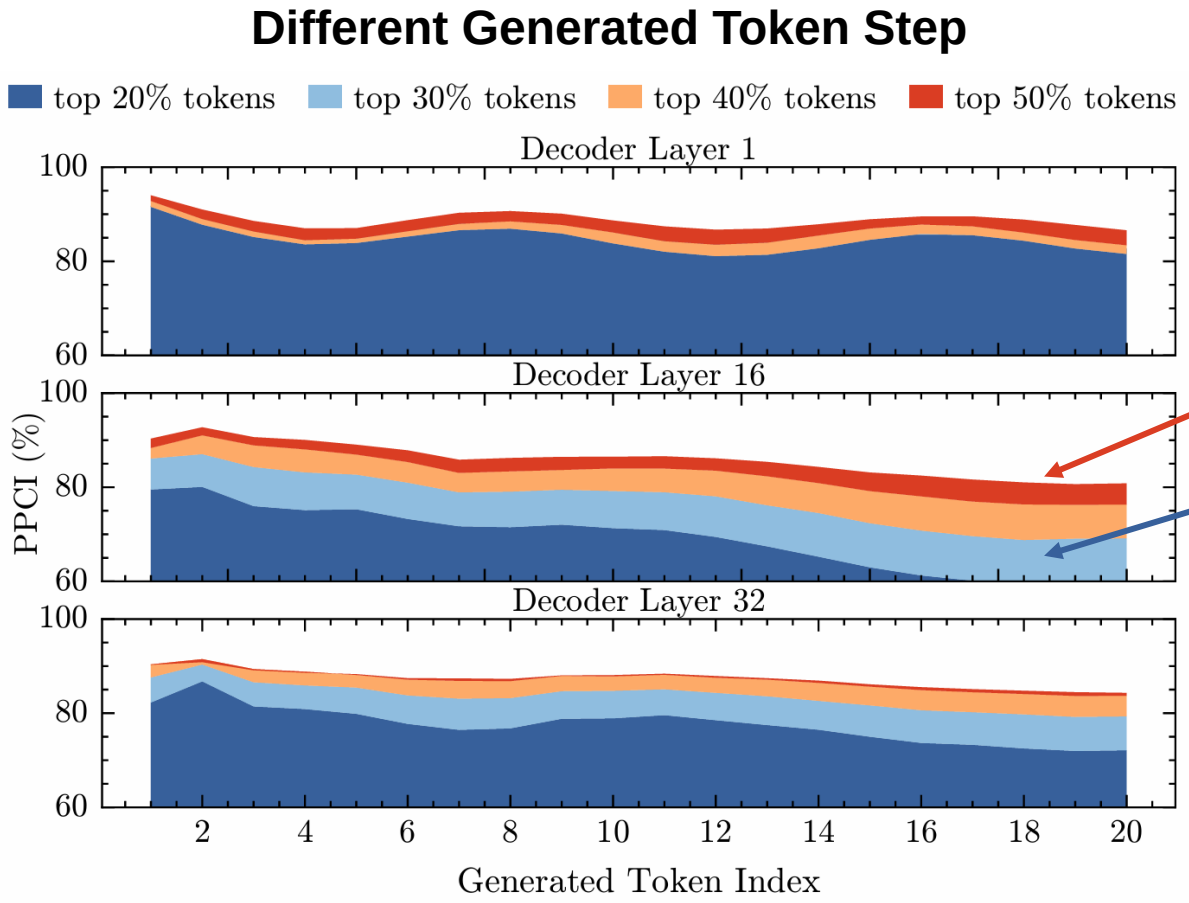
Visualization

The attention distribution of each layer is not similar.
We need to compute important tokens for different layers.

Observations and Insights

Image Token Attention Characteristics

Characteristic 3: Continuity in Sequence



Top 50% tokens always be focused

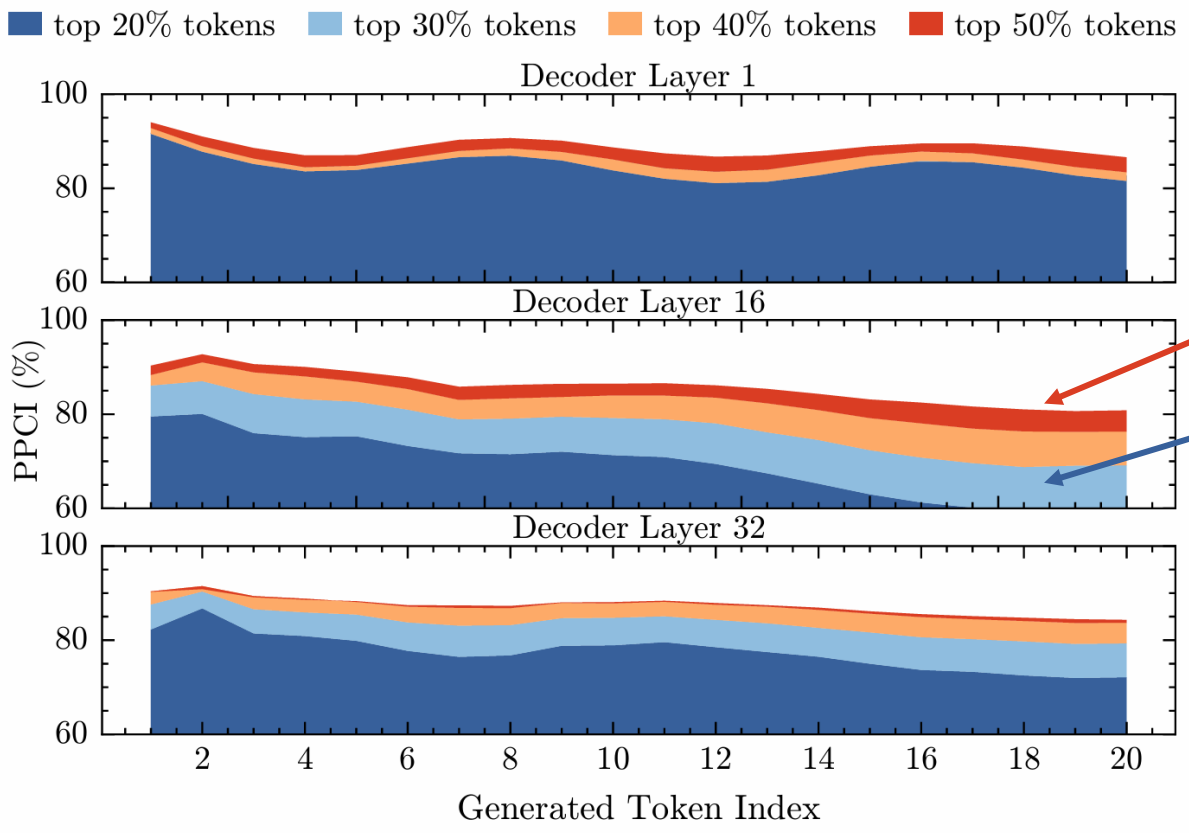
Top 30% tokens attention is slowly shifting

Observations and Insights

Image Token Attention Characteristics

Characteristic 3: Continuity in Sequence

Different Generated Token Step



During sequence generation,

attention is continuous in the short term (focusing on similar tokens), but it can shift.

Top 50% tokens always be focused

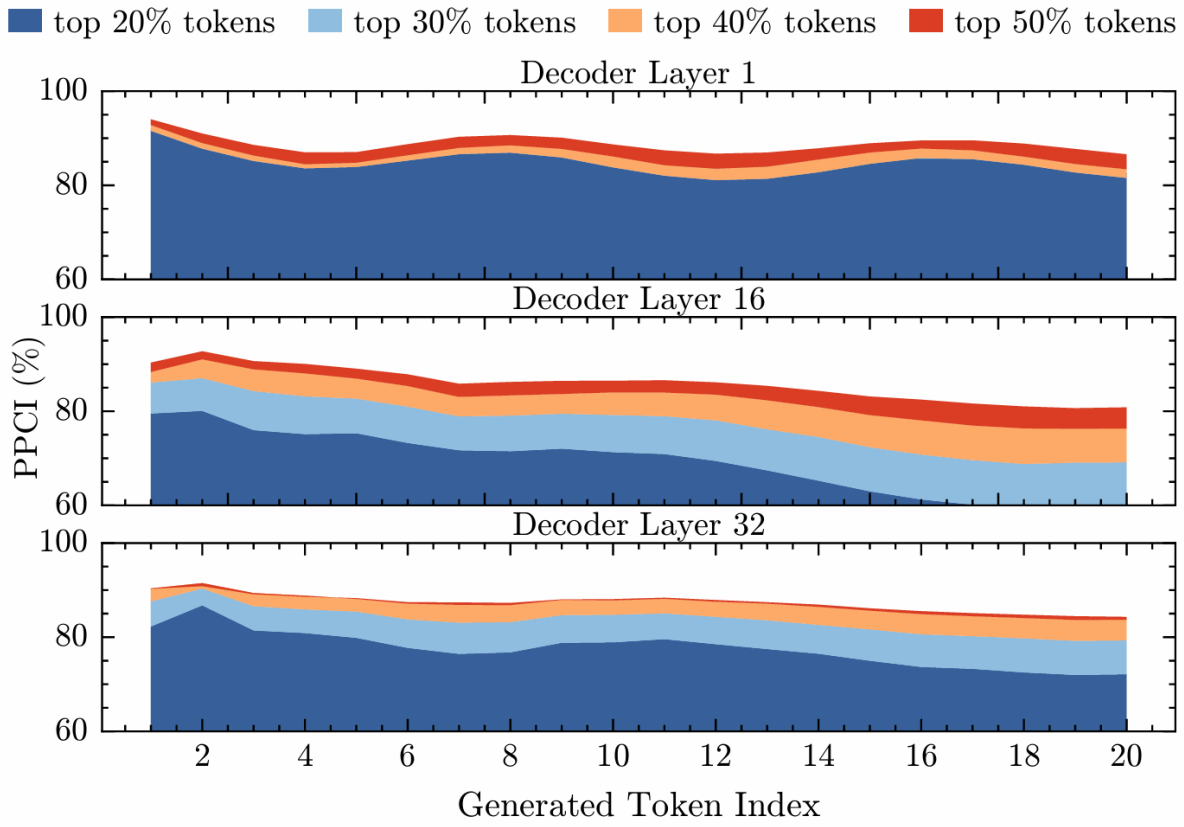
Top 30% tokens attention is slowly shifting

Observations and Insights

Image Token Attention Characteristics

Characteristic 3: Continuity in Sequence

Different Generated Token Step



During sequence generation,

attention is continuous in the short term (focusing on similar tokens), but it can shift.

Using **only 30%** top attention image tokens ensure **99%** samples generate 3 tokens that identical to those of the raw model.

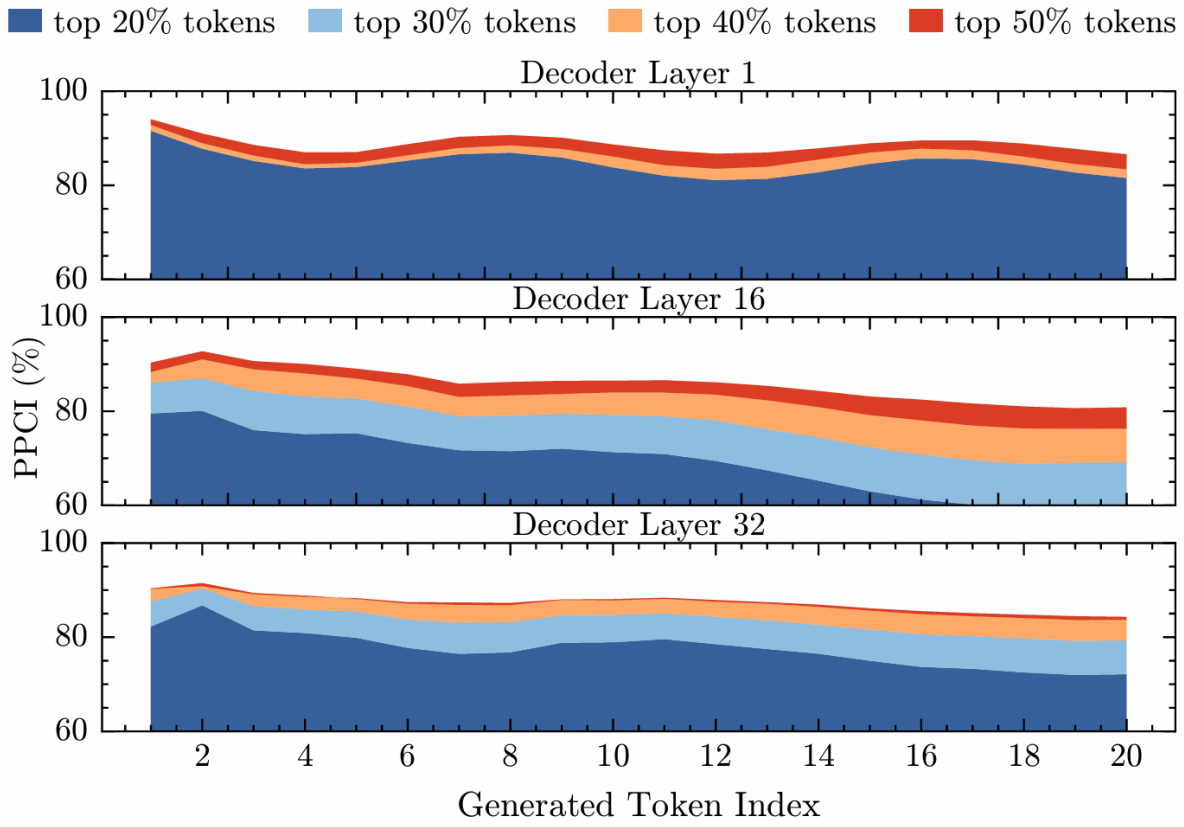
- LLaVA-1.5 7B model
- OCRVQA dataset

Observations and Insights

Image Token Attention Characteristics

Characteristic 3: Continuity in Sequence

Different Generated Token Step



During sequence generation,

attention is continuous in the short term (focusing on similar tokens), but it can shift.

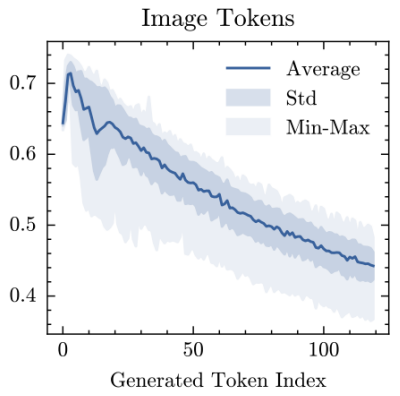
Using **only 30%** top attention image tokens ensure **99%** samples generate 3 tokens that identical to those of the raw model.

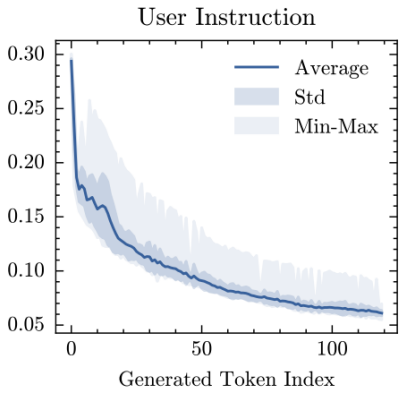
- LLaVA-1.5 7B model
- OCRVQA dataset

Used token in each step is few but may shift.

Observations and Insights

Text Token Attention Characteristics



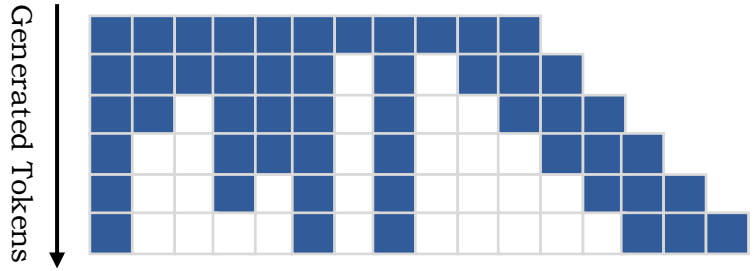


As the number of generated tokens increases,

- the attention of remote tokens decays rapidly;
- Model allocates more attention to the recent tokens;
- a small number of important remote tokens are allocated attention.

We also observed this phenomenon in LLaMa 2, which is a language model.

Therefore, we keep the recent token and the most important remote token, like the KV cache compression method in language model.



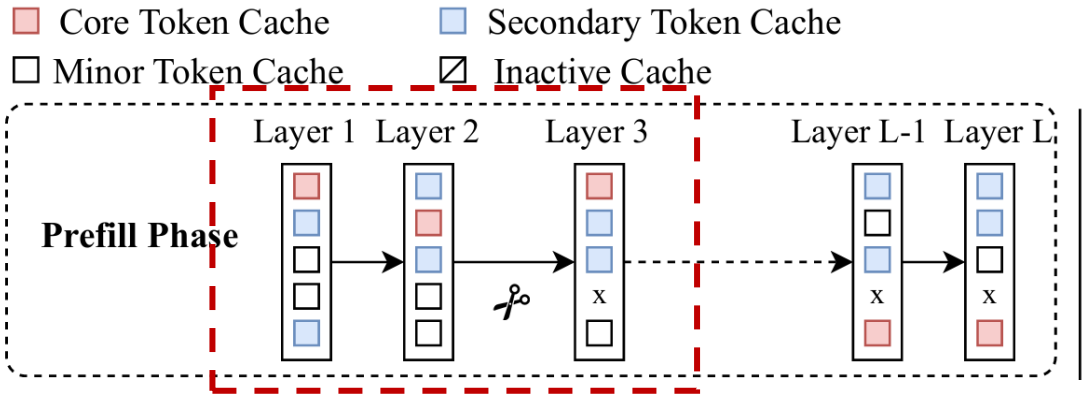
Outline

- Introduction
- Observations and Insights
- **Design of A-VL**
- Evaluation
- Conclusion

Design of A-VL

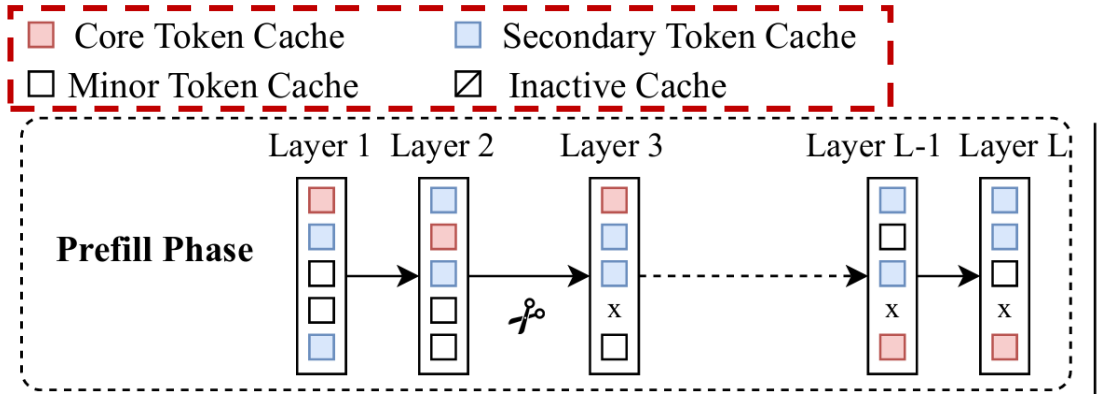
Adaptive Text-Aware Vision Attention

Prefill Phase: Delete a small amount of completely redundant image tokens in the shallow layer ($P\%$ tokens)



Design of A-VL

Adaptive Text-Aware Vision Attention



Prefill Phase: Delete a small amount of completely redundant image tokens in the shallow layer ($P\%$ tokens)

Redundancy Recognition: Tokens are divided into three types of importance based on attention scores:

- Core: Most relevant to the current query. $C\%$ tokens
- Secondary: Potentially useful for future. $S\%$ tokens
- Minor: Least relevant, evicted from cache. other tokens

Design of A-VL

Adaptive Text-Aware Vision Attention

Decode Phase: Adaptive Attention

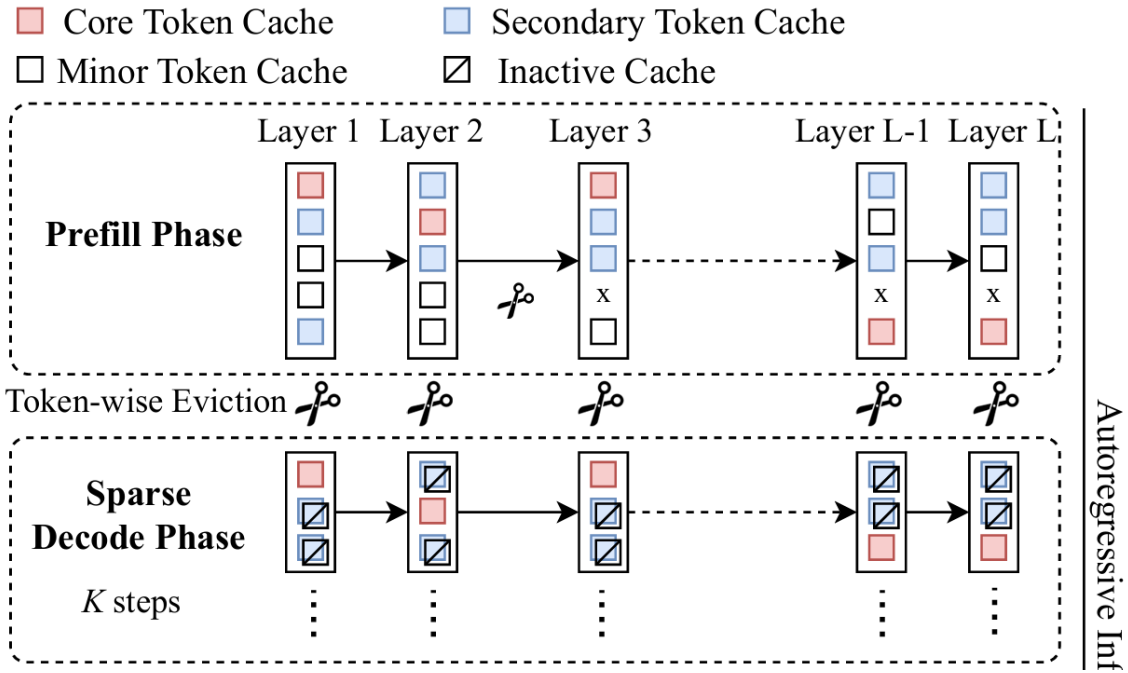
- Core: Most relevant to the current query. $C\%$ tokens
- Secondary: Potentially useful for future. $S\%$ tokens
- Minor: Least relevant, evicted from cache. other tokens

Evicted minor tokens for each layer.

Observation: Heterogeneity among Layers

Keep secondary tokens in memory, and only core tokens are computed at each step.

Observation: Sparsity & Continuity in image tokens



Design of A-VL

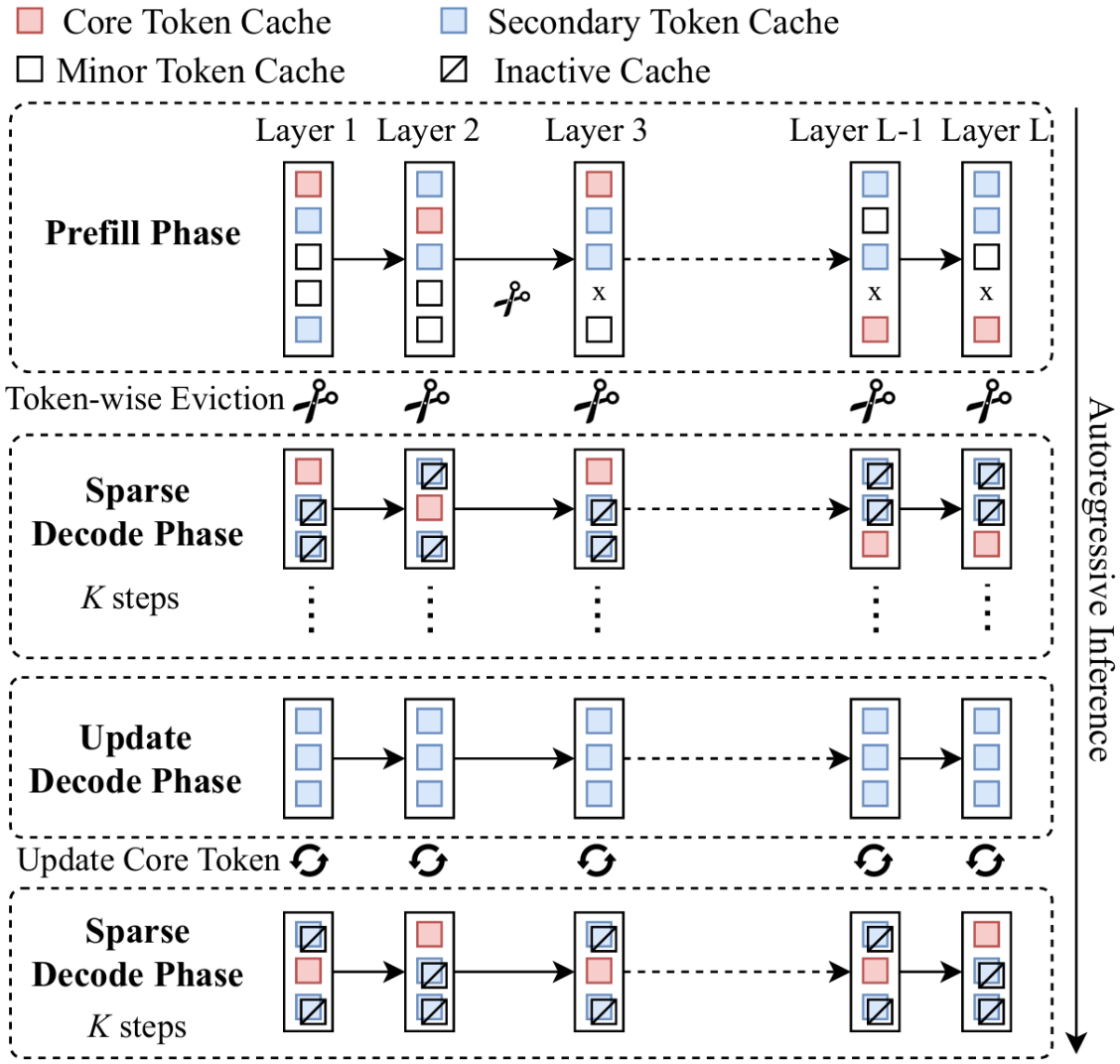
Adaptive Text-Aware Vision Attention

Decode Phase: Adaptive Attention

- Core: Most relevant to the current query. $C\%$ tokens
- Secondary: Potentially useful for future. $S\%$ tokens
- Minor: Least relevant, evicted from cache.

Regularly update the core token set to ensure that attention is always accurate. Update every K steps

Observation: Attention is slowly shifting

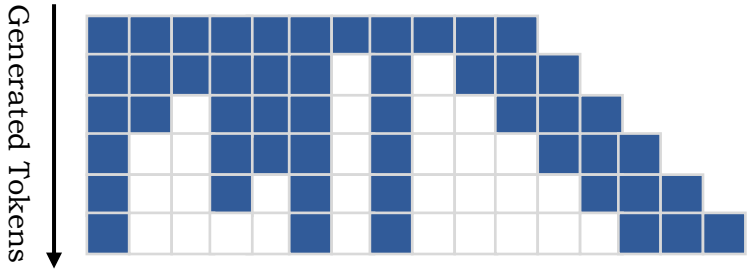


Adaptive Text Attention

Since the text attention pattern is similar to language models,

we use KV cache compression method in LLM, H_2O[1], to evict history tokens.

We only keep a cache of length $T\%$ of the original text sequence



Description of parameter

Param.	Description
S%	The proportion of secondary image tokens
C%	The proportion of core image tokens
K	Update the core token set every K steps
P%	Image tokens retained in prefill phase
T%	The proportion of retained text cache

1. Zhang, Zhenyu, et al. "H2o: Heavy-hitter oracle for efficient generative inference of large language models." Advances in Neural Information Processing Systems 36 (2023): 34661-34710.

Implementation of A-VL

To achieve adaptive attention, we need

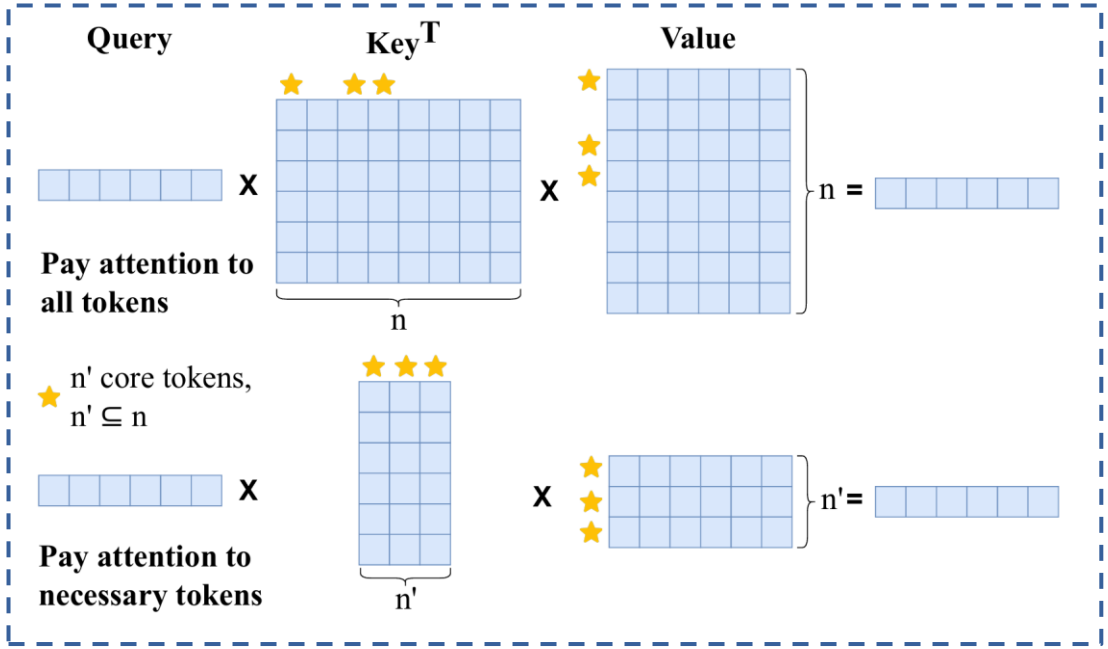
- **Cache selection and eviction:** execute concurrently with inference, ensuring efficiency without waiting.
- **Selective calculation:** Only the core tokens are calculated at each step.

Design of A-VL

Implementation of A-VL

To achieve adaptive attention, we need

- **Cache selection and eviction:** execute concurrently with inference, ensuring efficiency without waiting.
- **Selective calculation:** Only the core tokens are calculated at each step.

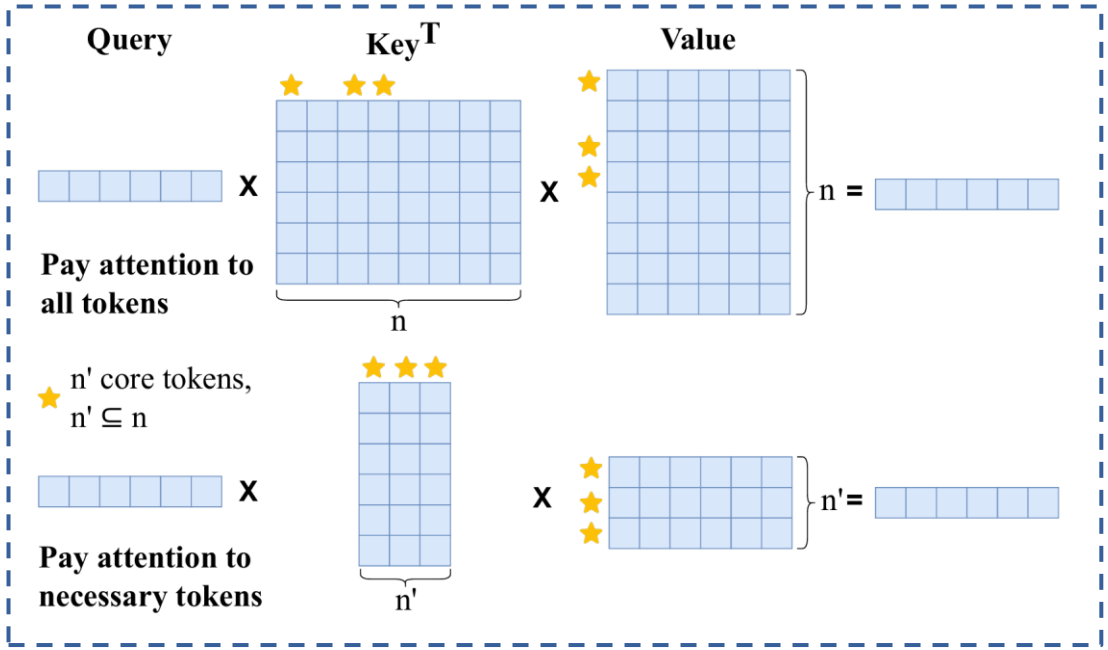


Design of A-VL

Implementation of A-VL

To achieve adaptive attention, we need

- **Cache selection and eviction:** execute concurrently with inference, ensuring efficiency without waiting.
- **Selective calculation:** Only the core tokens are calculated at each step.



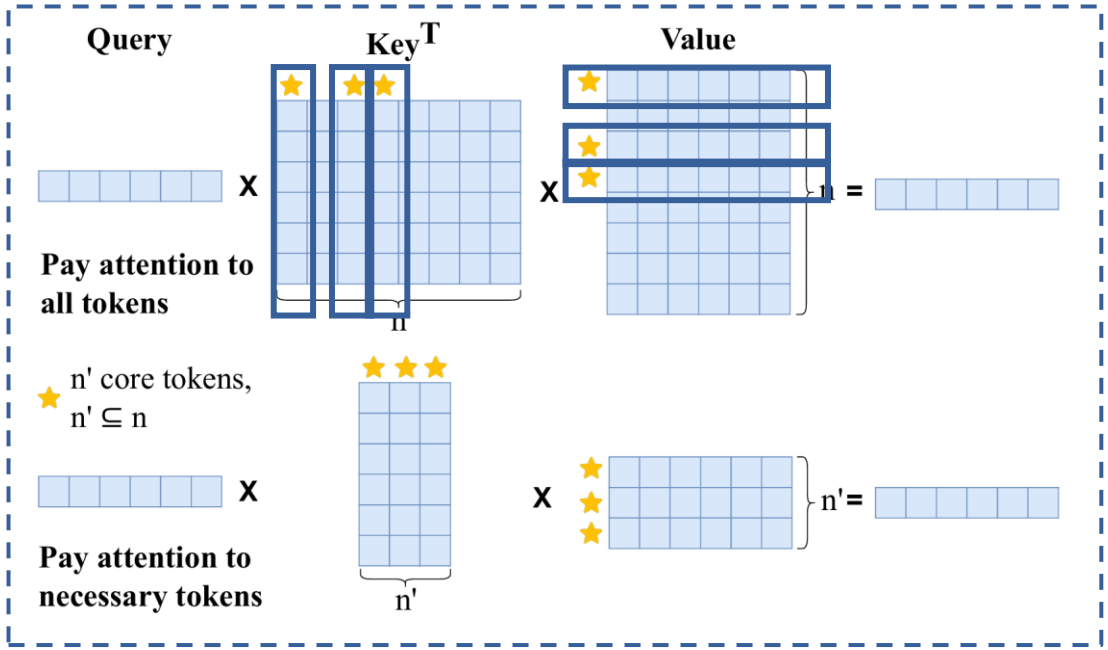
Slicing the cache before matrix multiplication intro duces a performance bottleneck, because the core caches are not stored contiguously in memory.

Design of A-VL

Implementation of A-VL

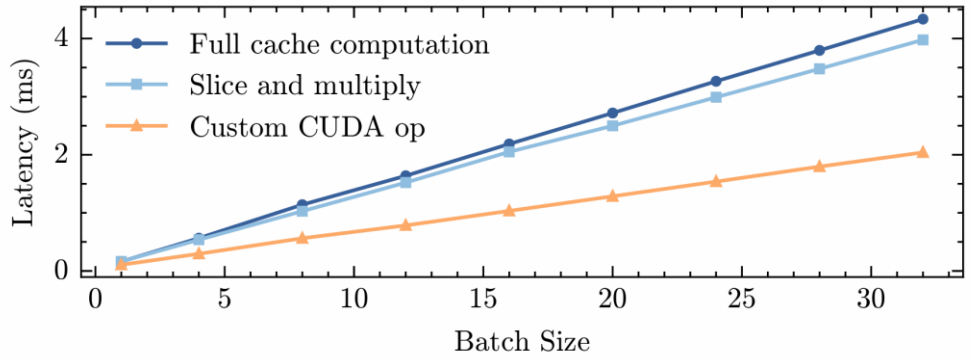
To achieve adaptive attention, we need

- **Cache selection and eviction:** execute concurrently with inference, ensuring efficiency without waiting.
- **Selective calculation:** Only the core tokens are calculated at each step.



Slicing the cache before matrix multiplication intro duces a performance bottleneck, because the core caches are not stored contiguously in memory.

we develop a **CUDA operator** that allows multiplication with selected rows or columns of the second matrix, eliminating the need for slicing.



Outline

- Introduction
- Observations and Insights
- Design of A-VL
- **Evaluation**
- Conclusion

Evaluation

Performance Guarantee

Model	Method	DocVQA	TextVQA	OCRBench	Nocaps	Flickr30k	VQAv2	Avg.
LLaVA 1.5 7B	Original	31.20	48.67	20.60	101.88	73.81	76.52	**58.78**
	FastV	27.35	47.16	19.30	101.06	72.99	74.86	57.12
	H2O	30.34	48.22	20.40	102.23	73.53	76.49	58.54
	Our	31.26	48.44	20.90	102.67	73.81	76.52	**58.93**
LLaVA 1.5 13B	Original	35.39	52.86	22.50	107.53	79.69	77.97	**62.66**
	FastV	30.27	50.64	20.60	106.41	78.88	76.65	60.57
	H2O	33.86	52.72	22.30	107.11	79.15	77.96	62.18
	Our	35.22	52.97	22.40	107.52	79.79	77.97	**62.65**
LLaVA 1.6 7B	Original	74.43	64.65	53.10	88.18	68.45	80.01	**71.47**
	FastV	67.69	63.15	48.30	86.21	66.73	79.58	68.61
	H2O	72.01	64.15	51.10	88.63	68.61	79.97	70.75
	Our	74.46	64.69	52.00	88.14	68.38	80.02	**71.28**
LLaVA 1.6 13B	Original	77.23	67.06	53.90	88.12	66.67	80.93	**72.32**
	FastV	70.26	65.14	49.10	87.78	65.96	80.48	69.79
	H2O	75.85	66.58	53.00	88.33	66.71	80.91	71.90
	Our	77.30	66.97	53.90	87.68	66.52	80.94	**72.22**
Qwen-VL	Original	68.65	61.10	49.10	58.92	59.01	79.17	**62.66**
	FastV	51.45	52.85	38.60	58.73	57.84	76.05	55.92
	H2O	65.84	60.59	46.60	59.06	59.39	79.16	61.77
	Our	68.09	60.66	49.10	58.46	58.88	79.16	**62.39**

Our method achieves nearly lossless performance and our average performance is better than the other two baselines.

Memory Usage of KV Cache

Description of parameter

Param.	Description
S%	The proportion of secondary image tokens
C%	The proportion of core image tokens
K	Update the core token set every K steps
P%	Image tokens retained in prefill phase
T%	The proportion of retained text cache

Model	Settings					Stored Cache	Used Cache	KV Cache Memory	DocVQA	TextVQA	OCRBench	Nocaps
	S%	C%	K	P%	T%							
LLaVA 1.5 7B	-	-	-	-	-	100%	100%	339 MB	31.20	48.67	20.60	101.88
	40	-	-	-	-	46%	46%	156 MB	31.26	48.44	20.90	102.67
	40	30	3	-	-	46%	39%	156 MB	31.02	48.44	21.00	102.67
	45	30	3	90	-	50%	40%	170 MB	31.02	48.46	20.70	102.34
	45	30	3	90	50	48%	38%	163 MB	30.92	48.46	20.50	102.22
LLaVA 1.5 13B	-	-	-	-	-	100%	100%	529 MB	35.39	52.86	22.50	107.53
	40	-	-	-	-	46%	46%	243 MB	35.37	52.86	22.20	107.61
	45	30	3	-	-	50%	40%	265 MB	35.22	52.97	22.40	107.52
	45	30	3	90	-	50%	40%	265 MB	35.46	52.83	22.30	107.60
	45	30	3	90	60	48%	38%	254 MB	35.43	52.84	22.30	107.59
LLaVA 1.6 7B	-	-	-	-	-	100%	100%	1179 MB	74.43	64.65	53.10	88.18
	40	-	-	-	-	43%	43%	507 MB	74.29	64.76	51.20	88.14
	45	30	3	90	-	47%	36%	554 MB	74.46	64.71	51.90	87.43
	45	30	3	90	70	45%	35%	531 MB	74.44	64.69	52.00	87.45
LLaVA 1.6 13B	-	-	-	-	-	100%	100%	1841 MB	77.23	67.06	53.90	88.12
	40	-	-	-	-	43%	43%	792 MB	77.30	66.97	53.90	87.68
	45	30	3	90	-	47%	36%	865 MB	76.97	67.06	54.40	87.59
	45	30	3	90	70	45%	35%	828 MB	76.97	67.06	54.50	87.59

Our method maintains lossless performance with less than 50% cache stored and only 35% used in computations.

The decode latency is only 50.5% of the original.

Notably, our method does not utilize the entire stored cache for computations, thus reducing the computational load compared to other methods.

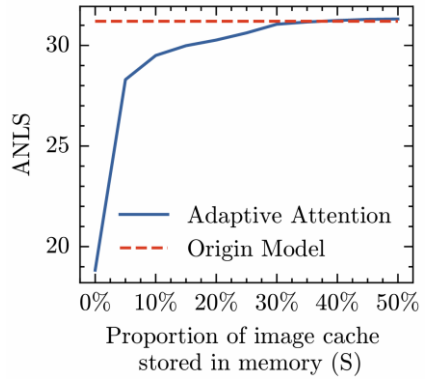
Evaluation

Memory Usage of KVCache

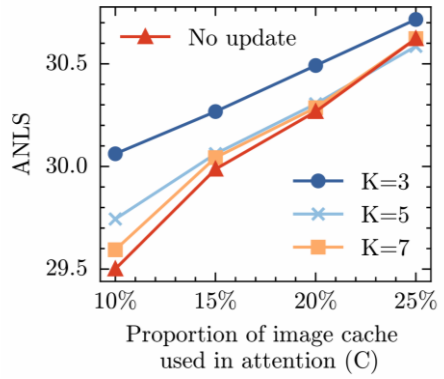
Description of parameter

Param.	Description
S%	The proportion of secondary image tokens
C%	The proportion of core image tokens
K	Update the core token set every K steps
P%	Image tokens retained in prefill phase
T%	The proportion of retained text cache

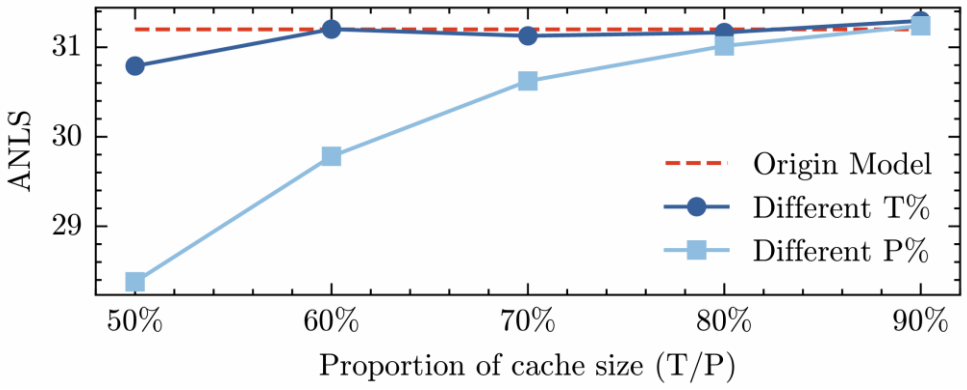
Different Param. **S**



Different Param. **C**



Different Param. **T & P**



As the amount of cache allocated and utilized for calculations decreases, the quality of the model's output degrades.

Users can make their selection based on the trade-off between efficiency and accuracy requirements.

Outline

- Introduction
- Observations and Insights
- Design of A-VL
- Evaluation
- **Conclusion**

Conclusion

The Take-Home Message

1. Large vision-language models exhibit significant redundancy, particularly in the vast number of visual tokens.
2. Different modalities have different attention patterns. The image token will be continuously allocated attention.
3. The number of tokens used at each step is small and gradually shifts. We can compute only the most critical tokens at each step.
4. Slicing matrix may pose a performance bottleneck due to memory non-contiguous, making it crucial to optimize this process at the CUDA level for efficiency.

A-VL: Adaptive Attention for Large Vision-Language Models

Lab for Intelligent Networking and Knowledge Engineering

Junyang Zhang
University of Science and Technology of China
School of Computer Science and Technology
zhangjunyang@mail.ustc.edu.cn

Xiang-Yang Li
University of Science and Technology of China
School of Computer Science and Technology
xiangyangli@ustc.edu.cn