Supplementary Materials for *AirCache: Activating Inter-modal Relevancy KV Cache Compression for Efficient Large Vision-Language Model Inference*

Kai Huang, Hao Zou, Bochen Wang, Ye Xi, Zhen Xie*, Hao Wang
Alibaba Group
zhouwan.hk, zh372956, bochen.wbc, yx150449, xiezhen.xz, qiao.wh@alibaba-inc.com

In this supplementary material, we first state the limitations of the proposed method and potential future work in Section A. Next, we provide more details on the method's application in Section B. After that, additional main comparative experiments with more models are discussed in Section C. Furthermore, we present additional ablation experiments in Section D. Finally, the visualization of chat generation is shown in Section E.

A. Limitations and Future Works

In performing dynamic allocation of the layer-wise compression budget, the proposed method requires obtaining the strength and skewness of the importance distribution of visual tokens for all layers before determining the allocable budget for each layer. This necessitates storing the complete KV cache after the prefill stage and executing the reduction only once the final compression budget is determined. Consequently, the proposed method is at a disadvantage in terms of peak memory consumption, a challenge also faced by most hierarchical budget allocation methods. Addressing how to maintain a peak memory advantage while supporting dynamic allocation of budgets across layers will be a key focus of our future work. In parallel, we will continue to explore the information flow mechanisms of different modalities in the inference process of LVLMs to further optimize the proposed method.

B. Implementation Details

For most methods, we adhere to their initial setup and perform the reduction of the visual KV cache based on the obtained importance ranking of visual tokens and the specified compression ratio. Our findings indicate that merging the dropped KV cache into the KV cache that needs to be retained works effectively on the LLaVA-v1.5 [5]. However, this approach tends to cause repetition issues in the LLaVA-OV series [3], InternVL2 series [1], and Qwen2-VL series [12], which results in a decline in model performance. Consequently, for Elastic Cache [6], we omitted the merge operation in the main experiments to achieve optimal performance results. In practical applications, and as observed in most existing multimodal evaluation datasets, visual tokens constitute the majority, while text tokens remain concise and short. The redundancy in the KV cache primarily resides in the visual part. As demonstrated in Table 8, a comparison of the actual number of visual tokens and text tokens in these multimodal VQA datasets shows that the visual component accounts for more than 97%. Thus, compressing only the visual KV cache eliminates redundant cache without affecting the complete expression of text instructions. Unless otherwise specified, all methods and experiments perform KV cache compression solely on the visual part.

Table 8. The number of visual tokens and text tokens across different models and evaluation sets.

Models	ChatQA [7]		InfoVQA [8]		DocVQA [9]		TextVQA [10]	
	N_v	N_t	N_v	N_t	N_v	N_t	N_v	N_t
LLaVA-OV-7B [3]	4763	47	6382	45	7224	42	5183	39
InternVL2-8B [1]	1828	32	3740	31	3230	28	1668	25
Qwen2-VL-7B [12]	1302	36	4450	34	4669	31	1325	28

C. Additional Main Results

Comparison with various model parameter sizes. Table 9 displays the comparison results of different parameter-sized InternVL2 [1] series models across various VQA datasets as the compression ratio varies. Similar to the conclusions drawn from experiments with different model architectures, the proposed method achieves superior results on models with different parameter sizes compared to existing methods. For instance, when retaining only 1% of the visual KV cache, the proposed method outperforms the SnapKV [4] method by an average of approximately 4.3% to 6.0% across four VQA evaluation datasets as the model parameter size varies. By synthesizing experiments on different architectures and base models with varying parameter sizes, we observe that the proposed method not only achieves better compression results but also demonstrates good general applicability. Furthermore, a comparison of

*Corresponding authors.

Table 9. The comparison of the KV cache compression methods on multimodal VQA benchmarks. The best result is highlighted in bold.

Models	Methods	ChatQA [7]				InfoVQA [8]				DocVQA [9]				TextVQA [10]			
		50%	10%	5%	1%	50%	10%	5%	1%	50%	10%	5%	1%	50%	10%	5%	1%
InternVL2-1B [1]	Full	67.9	67.9	67.9	67.9	50.1	50.1	50.1	50.1	80.0	80.0	80.0	80.0	70.8	70.8	70.8	70.8
	H2O [13]	67.7	62.0	57.7	53.5	50.0	42.7	38.8	33.7	79.8	73.8	68.2	57.0	70.1	58.4	52.3	47.6
	Elastic [6]	67.5	61.8	57.6	54.1	50.1	42.6	39.7	32.2	79.8	74.0	68.8	57.5	70.3	59.8	54.7	47.2
	PrefixKV [11]	**67.9**	62.1	58.0	53.3	49.9	43.8	40.6	34.2	79.7	74.3	70.4	59.6	70.1	60.7	55.3	48.6
	SnapKV [4]	67.8	63.7	59.8	56.7	50.1	47.3	44.4	39.7	79.8	76.5	72.1	61.2	70.4	64.5	60.7	52.2
	Ours	67.8	**65.7**	**63.5**	**60.8**	**50.1**	**49.6**	**47.5**	**45.8**	**80.0**	**77.7**	**74.3**	**68.5**	**70.6**	**68.9**	**66.5**	**59.3**
InternVL2-4B [1]	Full	81.1	81.1	81.1	81.1	65.9	65.9	65.9	65.9	88.1	88.1	88.1	88.1	74.7	74.7	74.7	74.7
	H2O [13]	81.1	79.2	77.6	72.1	65.9	61.1	57.4	51.8	79.9	80.1	75.4	69.2	74.2	66.4	58.8	51.3
	Elastic [6]	81.1	79.4	77.9	73.6	65.8	61.8	59.2	53.3	79.7	80.7	75.9	69.6	74.0	67.0	60.4	52.6
	PrefixKV [11]	81.0	79.5	77.8	73.2	65.9	62.6	59.1	53.4	88.0	81.4	76.4	70.3	74.4	67.5	61.4	53.7
	SnapKV [4]	81.1	79.3	78.5	74.6	65.9	64.3	61.8	56.7	88.0	84.3	79.7	73.2	74.4	70.3	65.8	60.3
	Ours	**81.1**	**80.4**	**79.6**	**77.7**	**66.0**	**65.5**	**64.2**	**62.1**	**88.1**	**86.8**	**84.3**	**81.5**	**74.5**	**73.6**	**70.7**	**67.4**
InternVL2-26B [1]	Full	85.4	85.4	85.4	85.4	75.4	75.4	75.4	75.4	92.1	92.1	92.1	92.1	82.5	82.5	82.5	82.5
	H2O [13]	84.9	82.4	80.4	78.6	75.0	71.8	65.1	62.5	91.9	84.4	81.6	75.1	82.3	75.2	70.3	65.2
	Elastic [6]	84.6	82.8	81.6	78.3	74.8	73.6	65.5	62.7	91.8	83.8	81.2	74.3	82.4	75.6	70.7	65.7
	PrefixKV [11]	84.8	82.2	81.5	78.8	75.2	73.8	66.4	63.2	92.0	84.2	81.5	74.7	82.4	75.5	70.6	65.4
	SnapKV [4]	85.3	83.5	83.0	80.1	75.4	74.1	69.5	65.6	91.9	86.6	86.3	82.5	**82.5**	78.6	74.3	71.1
	Ours	**85.5**	**84.7**	**84.2**	**82.3**	**75.4**	**74.8**	**72.8**	**70.7**	**92.1**	**91.4**	**89.0**	**86.8**	82.4	**81.7**	**78.2**	**76.6**

Table 10. Quantitative results on inference latency and throughput. The number of tokens output is consistently set to 512.

Models	Batch Size	Prompt Length	Prefill Latency (s)		Decoding Latency (s)			Throughput (token/s)		
			Full	Ours	Full	50%	10%	Full	50%	10%
InternVL2-8B [1]	8	2k	1.2	$1.4_{+16.7\%}$	9.3	$7.0_{+24.7\%}$	$5.9_{+36.6\%}$	440	$585_{+33.0\%}$	$694_{+36.6\%}$
		8k	4.6	$5.0_{+8.7\%}$	13.3	$10.5_{+21.1\%}$	$8.8_{+33.8\%}$	308	$390_{+26.6\%}$	$465_{+51.0\%}$
		16k	9.8	$10.5_{+7.1\%}$	24.8	$15.7_{+36.7\%}$	$11.9_{+52.0\%}$	165	$261_{+58.2\%}$	$344_{+52.0\%}$
		32k	23.8	$24.8_{+4.2\%}$	46.2	$28.1_{+39.2\%}$	$18.2_{+60.6\%}$	89	$146_{+64.0\%}$	$225_{+60.4\%}$
	16	2k	2.5	$2.7_{+8.0\%}$	12.2	$7.4_{+39.3\%}$	$6.0_{+50.8\%}$	671	$1107_{+65.0\%}$	$1365_{+50.8\%}$
		8k	9.9	$10.6_{+7.1\%}$	21.6	$12.9_{+40.3\%}$	$8.9_{+58.8\%}$	379	$635_{+67.5\%}$	$920_{+58.8\%}$
		16k	21.6	$22.4_{+3.7\%}$	28.5	$16.2_{+43.2\%}$	$10.0_{+64.9\%}$	287	$506_{+76.3\%}$	$819_{+65.0\%}$
Qwen2-VL-7B [12]	8	2k	1.1	$1.3_{+18.2\%}$	8.4	$6.6_{+21.4\%}$	$5.2_{+38.1\%}$	488	$621_{+27.3\%}$	$788_{+38.1\%}$
		8k	4.3	$4.7_{+27.3\%}$	12.6	$9.7_{+23.0\%}$	$8.1_{+35.7\%}$	325	$422_{+29.8\%}$	$506_{+55.7\%}$
		16k	9.4	$10.2_{+8.5\%}$	23.7	$14.9_{+37.1\%}$	$11.2_{+52.7\%}$	173	$275_{+59.0\%}$	$366_{+112.9\%}$
		32k	22.7	$24.0_{+5.7\%}$	45.0	$26.2_{+41.8\%}$	$14.6_{+67.6\%}$	91	$156_{+71.4\%}$	$214_{+135.2\%}$
	16	2k	2.1	$2.3_{+9.5\%}$	10.1	$6.8_{+32.7\%}$	$5.2_{+48.5\%}$	811	$1205_{+48.6\%}$	$1575_{+94.2\%}$
		8k	8.6	$9.2_{+7.0\%}$	19.7	$12.1_{+38.6\%}$	$8.4_{+57.4\%}$	426	$677_{+37.1\%}$	$975_{+128.9\%}$
		16k	19.2	$20.1_{+4.7\%}$	27.0	$15.6_{+42.2\%}$	$9.7_{+64.1\%}$	303	$525_{+73.3\%}$	$845_{+179.0\%}$

models with different parameter sizes reveals that as the model parameter size decreases, the impact of KV cache compression on model performance becomes more significant. This trend indicates that smaller parameter-sized models are less effective at integrating information within tokens, thereby placing a greater emphasis on the KV cache compression method's ability to select important visual tokens. The proposed method demonstrates a superior capability in assessing the importance of visual tokens, thereby more effectively reducing model performance loss.

Inference efficiency on more LVLMs. Inference efficiency on more LVLMs. Table 10 further presents the comparison of inference latency between the proposed method and the full cache on InternVL2-8B [1] and Qwen2-VL-7B [12]. From the table, it can be observed that when the input demand is relatively low, the model's need for the KV cache is reduced, thus limiting the gains from the KV cache compression method. Nevertheless, there is at least a 21% reduction in decoding latency and a 27% increase in throughput. As the input demand continues to rise, the benefits from KV cache compression become more significant. For example, in the case of the Qwen2-VL-7B [12] with a batch size of 16 and a prompt length of 16k, the proposed method can reduce decoding latency by 42% and increase throughput by 73% with almost no impact on model performance, while only adding 5% to prefill latency.

Detailed results of MMBench-Video. Table 11 presents the breakdown scores of different methods applied to the LLaVA-OV-7B [3] on the MMBench-Video [2] evaluation dataset. The proposed method outperforms existing methods in most subcategories, demonstrating its superior performance and stability. Notably, as the compression ratio increases, the advantages of the proposed method become more pronounced, especially for perceptual items that are more sensitive to visual information. By more accurately assessing the importance of visual tokens, the proposed

Table 11. The detailed comparison of the KV cache compression methods on MMBench-Video [2]. CP (coarse perception), FP-S (single-instance fine-grained perception), FP-C (cross-instance fine-grained perception), HL (Hallucination), LR (logic reasoning), AR (attribute reasoning), RR (relation reasoning), CSR (commonsense reasoning), TR (temporal reasoning).

Ratio	Methods	Overall	Perception	Reasoning	CP	FP-S	FP-C	HL	LR	AR	RR	CSR	TR
	Full	1.81	1.86	1.70	1.90	1.94	1.70	0.81	1.63	1.84	1.64	1.85	1.57
50%	H2O [13]	1.66	1.68	1.56	1.76	1.74	1.66	0.71	1.56	1.76	1.65	1.63	1.42
	Elastic [6]	1.68	1.71	1.60	1.79	1.75	1.66	0.73	1.58	1.77	1.67	1.66	1.44
	PrefixKV [11]	1.72	1.75	1.63	1.84	1.79	1.68	0.82	1.58	1.79	1.68	1.68	1.48
	SnapKV [4]	1.75	1.80	1.67	1.88	1.88	1.59	0.77	1.58	1.83	1.73	1.85	1.47
	Ours	1.80	1.84	1.69	1.89	1.94	1.66	0.81	1.61	1.84	1.77	1.79	1.53
10%	H2O [13]	1.63	1.64	1.61	1.72	1.71	1.52	0.74	1.54	1.81	1.64	1.73	1.43
	Elastic [6]	1.62	1.64	1.58	1.77	1.67	1.51	0.79	1.46	1.71	1.61	1.68	1.48
	PrefixKV [11]	1.68	1.74	1.58	1.81	1.73	1.63	0.80	1.56	1.79	1.68	1.69	1.47
	SnapKV [4]	1.70	1.76	1.60	1.83	1.83	1.62	0.82	1.55	1.70	1.67	1.78	1.44
	Ours	1.78	1.80	1.65	1.85	1.91	1.64	0.82	1.58	1.80	1.75	1.79	1.48
1%	H2O [13]	1.47	1.45	1.48	1.62	1.51	1.40	0.72	1.20	1.51	1.45	1.55	1.42
	Elastic [6]	1.51	1.52	1.51	1.65	1.54	1.40	0.75	1.22	1.54	1.50	1.60	1.45
	PrefixKV [11]	1.50	1.48	1.50	1.63	1.52	1.39	0.74	1.20	1.51	1.48	1.59	1.44
	SnapKV [4]	1.55	1.56	1.54	1.70	1.60	1.45	0.82	1.30	1.63	1.57	1.69	1.50
	Ours	1.67	1.70	1.58	1.78	1.76	1.65	0.72	1.58	1.76	1.65	1.64	1.52

Table 12. The results that the dropped KV cache is merged with the nearest preserved KV cache at different proportions. 100% means complete merging is used, while 0% means complete dropping is used.

Datasets	Ratio	100%	80%	60%	40%	20%	0%
ChatQA [7]	10%	76.4	77.0	77.4	78.6	79.2	79.9
	1%	72.2	73.6	74.2	75.5	76.1	76.4
InfoVQA [8]	10%	61.1	63.7	64.5	64.9	65.5	65.7
	1%	56.4	57.6	58.8	60.4	61.8	62.5
DocVQA [9]	10%	80.3	81.6	82.7	83.8	84.3	85.5
	1%	67.0	68.9	70.5	71.8	72.6	73.2
TextVQA [10]	10%	68.8	70.6	72.2	73.7	74.6	75.3
	1%	59.7	61.8	63.2	65.3	66.6	67.1

Table 13. Comparison of results across different evaluation sets and compression ratios with varying relevance thresholds.

Datasets	Ratio	0.99	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6
ChatQA [7]	10%	78.4	79.8	79.9	79.6	78.7	78.6	78.1	77.5	77.7
	1%	75.8	76.5	76.4	76.2	75.5	74.1	73.3	72.9	71.5
InfoVQA [8]	10%	65.1	65.5	65.7	65.7	65.0	63.7	62.2	61.6	59.4
	1%	61.9	62.3	62.5	62.4	61.2	58.6	55.4	53.7	52.2
DocVQA [9]	10%	82.4	85.6	85.5	84.7	81.3	78.8	76.1	75.5	74.8
	1%	71.2	72.8	73.2	72.9	70.0	65.5	61.8	58.9	55.6
TextVQA [10]	10%	73.4	75.0	75.3	75.4	74.6	74.0	73.3	72.5	71.4
	1%	65.5	67.2	67.1	66.8	65.2	64.7	63.5	62.4	61.2

method retains the critical visual KV cache, thereby minimizing the loss of model performance.

D. Additional Ablation Studies

KV Cache Merge *vs.* KV Cache Drop. Table 12 compares the model performance using merge and drop strategies for KV cache compression. The drop strategy clearly outperforms the merge strategy, with performance decline becoming more significant as the merge proportion increases. This phenomenon persists across different VQA evaluation sets and various compression ratios, indicating that directly dropping less important visual KV cache in LVLMs is a wiser choice. While the drop operation results in loss of visual information in the corresponding KV cache, the full token attention interaction during the prefill stage allows the remaining important visual tokens and text tokens to potentially absorb this missing information. This absorption helps mitigate the information loss caused by dropping KV cache during subsequent decoding. Conversely, although

Table 14. Layer-wise budget Jensen-Shannon divergence across different datasets and compression ratios, where Avg. corresponds to using the average allocation strategy.

Ratio	Setting	ChatQA [7]	InfoVQA [8]	DocVQA [9]	TextVQA [10]
10%	Avg. & s_t	0.46	0.54	0.61	0.57
	Avg. & s_k	0.53	0.49	0.58	0.55
	s_t & s_k	0.62	0.58	0.63	0.52
1%	Avg. & s_t	0.51	0.58	0.63	0.59
	Avg. & s_k	0.57	0.52	0.60	0.58
	s_t & s_k	0.68	0.61	0.59	0.57

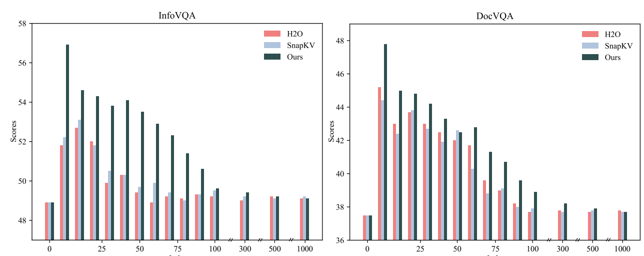


Figure 6. Comparison on InfoVQA [8] and DocVQA [9] by retaining only one visual token, which is selected based on the sorting of visual token importance scores using different methods.

the merge operation appears to preserve all visual information, the model lacks the ability to decode the original visual information from the merged visual KV cache. This operation may disrupt the representation of important visual information, ultimately leading to a decline in model performance.

Relevance Threshold α. Table 13 compares model performance under different relevance threshold. A relevance threshold around 0.9 achieves the best overall performance across various evaluation sets and compression ratios. If the relevance threshold is set too high or too low, it can lead to incomplete expression of instruction information or the inclusion of noise, respectively. This degrades the quality of visual KV cache importance assessment, thereby affecting the model's performance after KV cache compression. Comparing a higher relevance threshold with a lower one reveals that introducing more noisy text significantly affects model performance. This emphasizes the importance of filtering out irrelevant text tokens within the observation window when compressing KV cache in LVLMs.

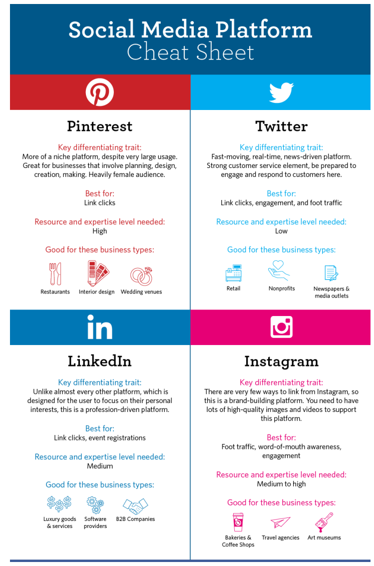
The Consistency of Strength and Skewness. To illustrate the difference between the dynamic budget and the average budget derived from the distribution strength and skewness used in this method, we recorded the budget distribution differences across various evaluation instances. The differences are quantified using Jensen-Shannon (JS) divergence, which ranges from 0 to 1. A JS divergence closer to 0 indicates smaller differences between the two distributions, while a value closer to 1 indicates larger differences. As shown in Table 14, the hierarchical budgets allocated based solely on the strength and skewness of the importance distribution are similar to those with an average allocation, indicating a complementary relationship. The former examines the layer's emphasis on visual information, while the latter focuses on the layer's ability to understand and interpret visual information. Combining both approaches can lead to better model performance.

Ablation Results of the Visual KV Cache Importance Evaluation. Figure 6 shows a comparison of selecting a single visual token based on the importance ranking of visual KV cache obtained by various methods on InfoVQA [8] and DocVQA [9]. As the importance of the selected visual token decreases, the performance of our proposed method also decreases reasonably. Additionally, for the same importance ranking, the performance of our proposed method is superior to that of existing methods.

E. Visualization of Chat Generation

Figures 7, 8, 9, and 10 illustrate a comparison of different methods applied to real chat generation while retaining only 1% of the visual KV cache. It is evident that the answers generated by the proposed method are more accurate.

User:

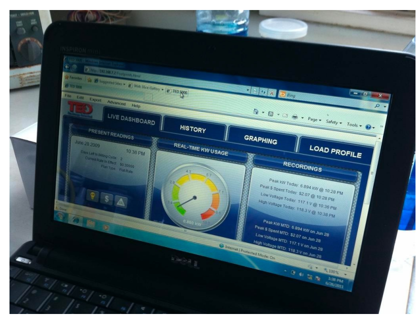


Which three business types is Pinterest good for?

LLaVA-OV w/ Full Cache:	Restaurants, Interior design, Wedding venues
LLaVA-OV w/ H2O:	Restaurants, Hotels, Retail
LLaVA-OV w/ Elastic:	Restaurants, Hotels, Retail
LLaVA-OV w/ AirCache:	Restaurants, Interior design, Wedding venues

Figure 7. Chat example applying KV cache compression methods on LLAVA-OV-7B [3].

User:



What date is under present readings?

LLaVA-OV w/ Full Cache:	june 28, 2009
LLaVA-OV w/ H2O:	june 20
LLaVA-OV w/ Elastic:	june 28
LLaVA-OV w/ AirCache:	june 28, 2009

Figure 8. Chat example applying KV cache compression methods on LLAVA-OV-7B [3].

References

[1] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 1, 2

[2] Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao,

User:	

11:14 to 11:39 a.m. Coffee Break
Coffee will be served for men and women in the lobby adjacent to exhibit area. Please move into exhibit area. **(Exhibits Open)**

11:39 a.m. TRRF GENERAL SESSION (PART I)
Presiding: Lee A. Waller
TRRF Vice President

11:39 to 11:44 a.m. **"Introductory Remarks"**
Lee A. Waller, TRRF Vice President

11:44 a.m. to 12:25 p.m. **Individual Interviews with TRRF Public Board Members and Scientific Advisory Council Members**
Conducted by TRRF Treasurer Philip G. Kuehn to get answers which the public refrigerated warehousing industry is looking for. Plus questions from the floor.
Dr. Emil M. Mrak, University of California, Chairman, TRRF Board; Sam R. Cecil, University of Georgia College of Agriculture; Dr. Stanley Charm, Tufts University School of Medicine; Dr. Robert H. Cotton, ITT Continental Baking Company; Dr. Owen Fennema, University of Wisconsin; Dr. Robert E. Hardenburg, USDA.

12:25 to 12:58 p.m. Questions and Answers

12:58 to 4:00 p.m. **Exhibits Open**
Capt. Jack Stoney Room

2:00 to 5:00 p.m. **TRRF Scientific Advisory Council Meeting**
Ballroom Foyer

What time is 'question and answers' session?

LLaVA-OV w/ Full Cache:	12:25 to 12:58 p.m.
LLaVA-OV w/ H2O:	12:25-12:38
LLaVA-OV w/ Elastic:	12:25-12:50
LLaVA-OV w/ AirCache:	12:25 to 12:58 p.m.

Figure 9. Chat example applying KV cache compression methods on LLAVA-OV-7B [3].

User:	

What is unusual about this image?

LLaVA-OV w/ Full Cache:	The image shows a **man** ironing clothes **on the back of a moving vehicle**, which is an unusual and potentially dangerous activity. Typically, ironing is done indoors, away from traffic and moving vehicles, to ensure safety and focus on the task. The man's actions in this image are unconventional and could pose risks to himself, other road users, and the passengers in the vehicle.
LLaVA-OV w/ H2O:	The image shows a **person** ironing clothes **outside on a street**, which is an unusual and unexpected activity. Typically, ironing is done indoors, in a home or a dedicated laundry area, to protect the clothes from outdoor elements and to maintain privacy. Performing this task outdoors breaks with conventional norms and could be due to various reasons such as a lack of indoor space, a desire for fresh air, or it might be part of a public demonstration or event.
LLaVA-OV w/ Elastic:	The image shows a **person** ironing clothes in an unusual setting: it appears to be **inside a car**, which is not a typical place for doing laundry. Ironing typically takes place in a home or a dedicated laundry room, where there is space and a suitable environment for the activity. The presence of an ironing board and iron in what looks like a vehicle's interior is unexpected and humorous.
LLaVA-OV w/ AirCache:	The image shows a **man** ironing clothes **on the back of a moving vehicle**, which is an unusual and unexpected sight. Typically, ironing is done indoors, usually on a stable surface like a table or ironing board, away from traffic and outdoor elements. The man's activity here poses risks to his safety and could also be considered a disruption to traffic flow.

Figure 10. Chat example applying KV cache compression methods on LLAVA-OV-7B [3]. Important information is highlighted in red and blue.

Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *arXiv preprint arXiv:2406.14515*, 2024. 2, 3

[3] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1, 2, 4, 5

[4] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024. 1, 2, 3

[5] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 1

[6] Zuyan Liu, Benlin Liu, Jiahui Wang, Yuhao Dong, Guangyi Chen, Yongming Rao, Ranjay Krishna, and Jiwen Lu. Efficient inference of vision instruction-following models with elastic cache. In *European Conference on Computer Vision*, pages 54–69. Springer, 2024. 1, 2, 3

[7] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 1, 2, 3

[8] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 1, 2, 3, 4

[9] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 1, 2, 3, 4

[10] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 1, 2, 3

[11] Ao Wang, Hui Chen, Jianchao Tan, Kefeng Zhang, Xun-

liang Cai, Zijia Lin, Jungong Han, and Guiguang Ding. Prefixkv: Adaptive prefix kv cache is what vision instruction-following models need for efficient generation. *arXiv preprint arXiv:2412.03409*, 2024. 2, 3

[12] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1, 2

[13] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3