
Supplemental Material

Video-RAG: Visually-aligned Retrieval-Augmented Long Video Comprehension

Anonymous Author(s)

Affiliation

Address

email

1 Open-source Codes

2 Our code is available at <https://anonymous.4open.science/r/Video-RAG-97DA> by links to
3 Anonymous GitHub.

4 Decouple Query

5 In the initial phase of the proposed Video-RAG, we employ a decouple prompt, denoted as **P**, to
6 guide the LVLm in generating retrieval requests. In this section, we present one example of a prompt
7 designed for multiple-choice questions, as illustrated in Figure 3.

8 Sub-set of Video-MME

9 As outlined in the implementation details, we randomly sampled a subset of the Video-MME [2]
10 dataset to evaluate a computationally resource-intensive, agent-based method with long-context
11 LVLms. Specifically, we selected 10% of the full dataset, comprising 30 short, 30 medium-length,
12 and 30 long videos. Each video contains three multiple-choice questions. Importantly, we ensured
13 that the performance ranking of the methods on the subset mirrored that of the full dataset. As shown
14 in Tables 1 and 2, we evaluated four distinct 7B models Chat-Univi-v1.5 [4], LLaVA-NeXT-Video
15 [10], LongVA [9], and Long-LLaVA [8] using a frame sampling rate of 16 for both the subset and
16 the full set. Our results indicate that the performance rankings remained consistent across both
17 evaluations.

Table 1: Performance of Video-MME sub-set.

Method	Short	Medium	Long	Overall
Chat-Univi-v1.5 [4]	50.0	33.3	17.8	33.7
LLaVA-NeXT-Video [10]	54.4	33.3	23.3	37.0
LongVA [9]	56.7	50.0	38.9	48.5
Long-LLaVA [8]	58.9	52.2	40.0	50.4

18 4 Results on Video-MME Sub-Set

19 We examine Video-RAG against two representative methods in terms of inference time, GPU resource
20 requirements, and overall performance. Given that GPT-based Agent methods are resource-intensive,
21 we randomly sampled a sub-set of the Video-MME [2] for evaluation, as described in Section

Table 2: Performance of Video-MME full-set.

Method	Short	Medium	Long	Overall
Chat-Univi-v1.5 [4]	45.7	39.0	35.7	40.1
LLaVA-NeXT-Video [10]	51.1	41.8	36.8	43.2
LongVA [9]	60.8	45.2	41.4	49.1
Long-LLaVA [8]	59.3	49.3	44.4	51.0

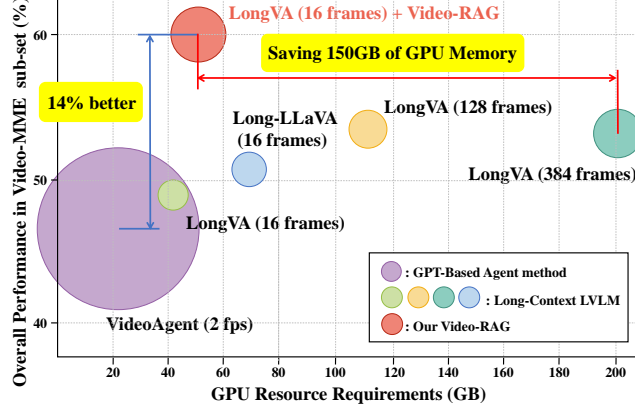


Figure 1: The comparison of our Video-RAG with two common approaches. The size of the bubbles represents the total time consumed for completing inference on the Video-MME [2] sub-set.

3. As demonstrated in Figure 1, VideoAgent [1], a typically GPT-based Agent method, requires significant time to process video and deliver suboptimal performance. Meanwhile, LongVA [9], a representative long-context LVLM, shows limited improvement from increasing the frame rate and even experiences performance degradation. Integrating our Video-RAG into the 16-frame LongVA results in substantial performance improvements while reducing GPU resource consumption. Specifically, with only increasing 8GB GPU memory compared to the base (16-frames LongVA), we achieve 11.5% overall performance improvement, while outperforming another long-context LVLM Long-LLaVA-7B [8] in 16-frames setting by 9.6% with less GPU memory requirements and compatible total inference time. These results demonstrated that our Video-RAG is lightweight with lower computing overhead than the other typical methods. Moreover, we provide detailed time consuming to construct three types of databases (which can be built in parallel) and inference per query, as shown in Table 3.

Table 3: Overall performance, databases construct and average inference time (include building databases) per query (#Time) in Video-MME-mini.

Model	ASR	OCR	DET	Total Time	w/o subs #Time Overall	w/ Video-RAG #Time Overall
VideoAgent	-	-	-	-	14min 47.7	- -
LongVA-16fs	21min	2min	3min	max(21, 2, 3)=21min	1s 48.5	1s + 5s 60.0
LongVA-128fs	21min	16min	16min	max(21, 16, 16)=21min	8s 54.1	8s + 5s 63.3
LongVA-384fs	42min	48min	24min	max(42, 48, 24)=48min	20s 53.7	20s + 11s 63.6

5 Details of Similarity Score Calculation

In the process of using the RAG system to retrieve auxiliary texts extracted from videos, we define a similarity threshold t to ensure the selection of relevant texts. Specifically, we employ FAISS-based [5] similarity to select OCR and ASR texts, while CLIP [6] similarity is used for keyframe selection. In our implementation, the similarity threshold t is set to 0.3. As for OCR and ASR selection, For any given list of the retrieve request \mathbf{R} and auxiliary texts \mathbf{A} , the Contriever [3] framework maps the text to a text embedding as:

$$\mathbf{E}_{a_i} = \text{Contriever}(\mathbf{A}_i), \quad i = 1, 2, \dots, n$$

$$\mathbf{E}_{r_i} = \text{Contriever}(\mathbf{R}_i), \quad i = 1, 2, \dots, n$$

41 The average embedding of the retrieve request is then computed as:

$$\mathbf{E}_r = \frac{1}{n} \sum_{i=1}^n \mathbf{E}_{r_i}$$

42 After that, the embedding of the request and the list of auxiliary texts is normalized:

$$\mathbf{E}_{a_i} = \frac{\mathbf{E}_{a_i}}{\|\mathbf{E}_{a_i}\|}, \quad \mathbf{E}_r = \frac{\mathbf{E}_r}{\|\mathbf{E}_r\|}$$

43 The similarity between the query embedding \mathbf{E}_r and the document vector \mathbf{E}_a is computed using the
44 inner product, the FAISS library is employed to efficiently perform this search and return the indices
45 of the auxiliary texts meeting the criterion:

$$S(\mathbf{E}_r, \mathbf{E}_{a_i}) = \mathbf{E}_r \cdot \mathbf{E}_{a_i} > t$$

46 As for object detection, we use CLIP to select the video keyframe. During this process, we first
47 filter the object detection request \mathbf{R}_{det} to ensure they correspond to CLIP-sensitive physical entities,
48 avoiding the inclusion of abstract concepts. Specifically, if it is a single word, direct part-of-speech
49 filtering is applied; if it is a compound word, certain rules are followed to check for compliance, such
50 as whether it is an adjective plus a noun, or a noun plus a noun. We use the Spacy library to achieve
51 this. After this, we put the text ‘‘A picture of’’ before each object detection request.

52 Then, we extracting embedding from both the video frames \mathbf{F} and the detection request \mathbf{R}_{det} :

$$\begin{aligned} \mathbf{E}_{\mathbf{F}_j} &= \text{CLIP}(\mathbf{F}_j), \quad j = 1, 2, \dots, m \\ \mathbf{E}_{\mathbf{R}_i} &= \text{CLIP}(\mathbf{R}_{det_i}), \quad i = 1, 2, \dots, n \end{aligned}$$

54 The similarity between each video frame and the detection retrieve requests is computed using the
55 dot product between the image and text feature embeddings. For each frame \mathbf{F}_j , and for each retrieve
56 request $\mathbf{E}_{\mathbf{R}_i}$, the similarity score is given by:

$$S_{ij} = \mathbf{E}_{\mathbf{F}_j} \cdot \mathbf{E}_{\mathbf{R}_i}$$

57 where \cdot denotes the dot product. The final similarity score for each frame is the average similarity
58 across all requests:

$$S_j = \frac{1}{n} \sum_{i=1}^n S_{ij}$$

59 This computes the mean similarity for each frame across all text descriptions, resulting in a similarity
60 vector $\mathbf{S} = [S_1, S_2, \dots, S_m]$. The similarity scores are adjusted by a scaling factor α , which is
61 computed based on the number of frames m and a base frame number b (which is set to 16 and 4.0,
62 respectively) to adapted different video sampling rate of LVLMS:

$$\alpha = \beta \times \frac{m}{b}$$

63 where β is a predefined scaling parameter.

64 Next, the similarity scores are scaled and normalized to ensure that they sum to 1:

$$S_j^{\text{norm}} = \frac{\alpha \times S_j}{\sum_{k=1}^m S_k}$$

65 where S_j^{norm} represents the normalized similarity score for frame \mathbf{F}_j .

66 The final step is to select the keyframes based on the normalized similarity scores. A threshold t is
67 applied to the normalized similarities, such that frames with similarity scores above the threshold are
68 selected as keyframes:

$$\text{Keyframe: } \mathbf{F}_j \quad \text{if } S_j^{\text{norm}} > t$$

69 Thus, the set of selected keyframes is given by:

$$\mathbf{F}_{key} = \{\mathbf{F}_j \mid S_j^{\text{norm}} > t, j = 1, 2, \dots, m\}$$

6 More Ablation Studies

Effect of different components of Video-RAG. We evaluate the performance across sub-tasks within Video-MME [2], as shown in Figure 2. The results reveal that object detection auxiliary texts significantly enhance spatial perception and object counting, while OCR auxiliary texts specifically improve performance on text recognition tasks. Additionally, ASR auxiliary texts contribute to a general improvement in inference tasks, underscoring the critical role of audio transcription in video understanding. Given that audio transcription is considerably more time-consuming than character recognition or object detection, these texts should be selected based on the requirements of the application.

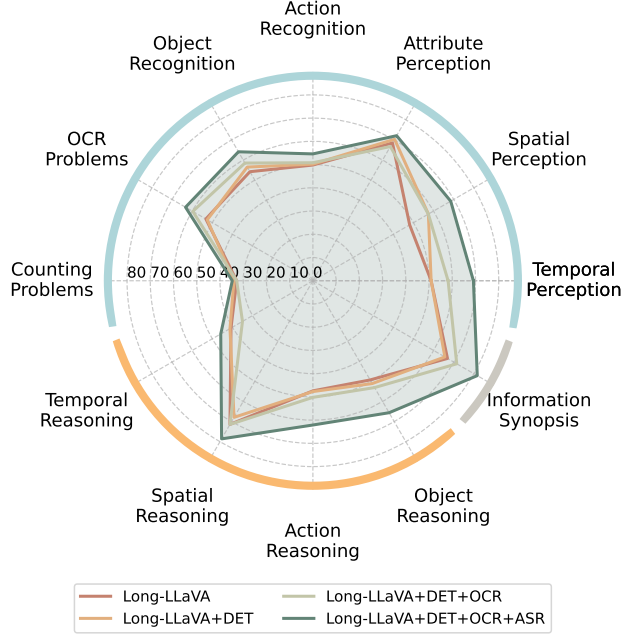


Figure 2: Performance on 12 sub-tasks in Video-MME [2] benchmark after applying different components in Long-LLaVA.

Besides studying the inference of different components of Video-RAG in the Video-MME [2] benchmark, we also experiment with a different type of video benchmark. We first evaluate LLaVA-Video in MLVU [12] and LongVideoBench [7] in both 7B and 72B scale with the 64-frame setting, results are shown in Table 4. As demonstrated, when all components are combined, we get optimal performance in both datasets, including 7B and 72B scales. Specifically, the performance in MLVU [12] even declined when the RAG system was not implemented.

Table 4: Ablation study in MLVU and LongVideoBench.

RAG	DET	OCR	ASR	7B		72B	
				MLVU	LVB	MLVU	LVB
				70.8	56.6	73.1	61.9
✓	✓			71.0	56.5	73.4	63.2
✓	✓	✓		71.3	56.8	73.5	63.4
✓	✓	✓	✓	72.4	58.7	73.8	65.4
	✓	✓	✓	70.3	58.3	72.9	64.0

Then, to better point out the role of DET and OCR, we evaluate Video-RAG in VNBench [11] with Long-LLaVA-7B [8]. VNBench is a synthetic benchmark designed to evaluate models’ long-context abilities, covering tasks such as retrieval, ordering, and counting. VNBench randomly inserts stickers or text into the video that has nothing to do with the original content of the video, thus typically

Decouple Prompt of the Multiple-choice Question

To answer the question step by step, list all the physical entities related to the question you want to retrieve, you can provide your retrieve request to assist you by the following JSON format:

```
{
  "ASR": Optional[str]. The subtitles of the video that may relavent to the
  question you want to retrieve, in two sentences. If you no need for this
  information, please return null.

  "DET": Optional[list]. (The output must include only physical entities, not
  abstract concepts, less than five entities) All the physical entities and their
  location related to the question you want to retrieve, not abstract concepts. If
  you no need for this information, please return null.

  "TYPE": Optional[list]. (The output must be specified as null or a list
  containing only one or more of the following strings: 'location', 'number',
  'relation'. No other values are valid for this field) The information you want
  to obtain about the detected objects. If you need the object location in the
  video frame, output "location"; if you need the number of specific object,
  output "number"; if you need the positional relationship between objects, output
  "relation".
}
```

Example 1:

Question: How many blue balloons are over the long table in the middle of the room at the end of this video? A. 1. B. 2. C. 3. D. 4.

Your retrieve can be:

```
{
  "ASR": "The location and the color of balloons, the number of the blue
  balloons.",
  "DET": ["blue ballons", "long table"],
  "TYPE": ["relation", "number"]
}
```

Example 2:

Question: In the lower left corner of the video, what color is the woman wearing on the right side of the man in black clothes? A. Blue. B. White. C. Red. D. Yellow.

Your retrieve can be:

```
{
  "ASR": null,
  "DET": ["the man in black", "woman"],
  "TYPE": ["location", "relation"]
}
```

Example 3:

Question: In which country is the comedy featured in the video recognized worldwide? A. China. B. UK. C. Germany. D. United States.

Your retrieve can be:

```
{
  "ASR": "The country recognized worldwide for its comedy.",
  "DET": null,
  "TYPE": null
}
```

Note that you don't need to answer the question in this step, so you don't need any infomation about the video of image. You only need to provide your retrieve request (it's optional), and I will help you retrieve the infomation you want. Please provide the json format.

Figure 3: Decouple prompt of the multiple-choice question for LVLMs.

89 challenging the model’s needle-in-the-haystack capability. As shown in Table 5, we find that applying
 90 DET and OCR as auxiliary texts can significantly improve the performance in retrieval, ordering, and
 91 counting tasks. However, the ASR component will decline the performance due to the subtitles are

not ancillary to this particular task. These results demonstrated that our proposed distinct types of auxiliary texts can be selected according to the application needs to meet the requirements better.

Table 5: Results on combinations of different auxiliary texts in VNBench [11] with 1-try setting when applying 7B Long-LLaVA [8] as LVLM under the 32-frames setting. **Ret**, **Ord**, and **Cnt** represent retrieval, ordering, and counting tasks, respectively.

RAG	DET	OCR	ASR	Ret	Ord	Cnt	Overall
✓	✓			65.1	25.6	24.2	38.3
✓	✓			66.9	28.4	23.8	39.7
✓	✓	✓		68.2	31.3	28.9	42.8
✓	✓	✓	✓	66.7	31.3	29.6	42.5

7 More Qualitative Results

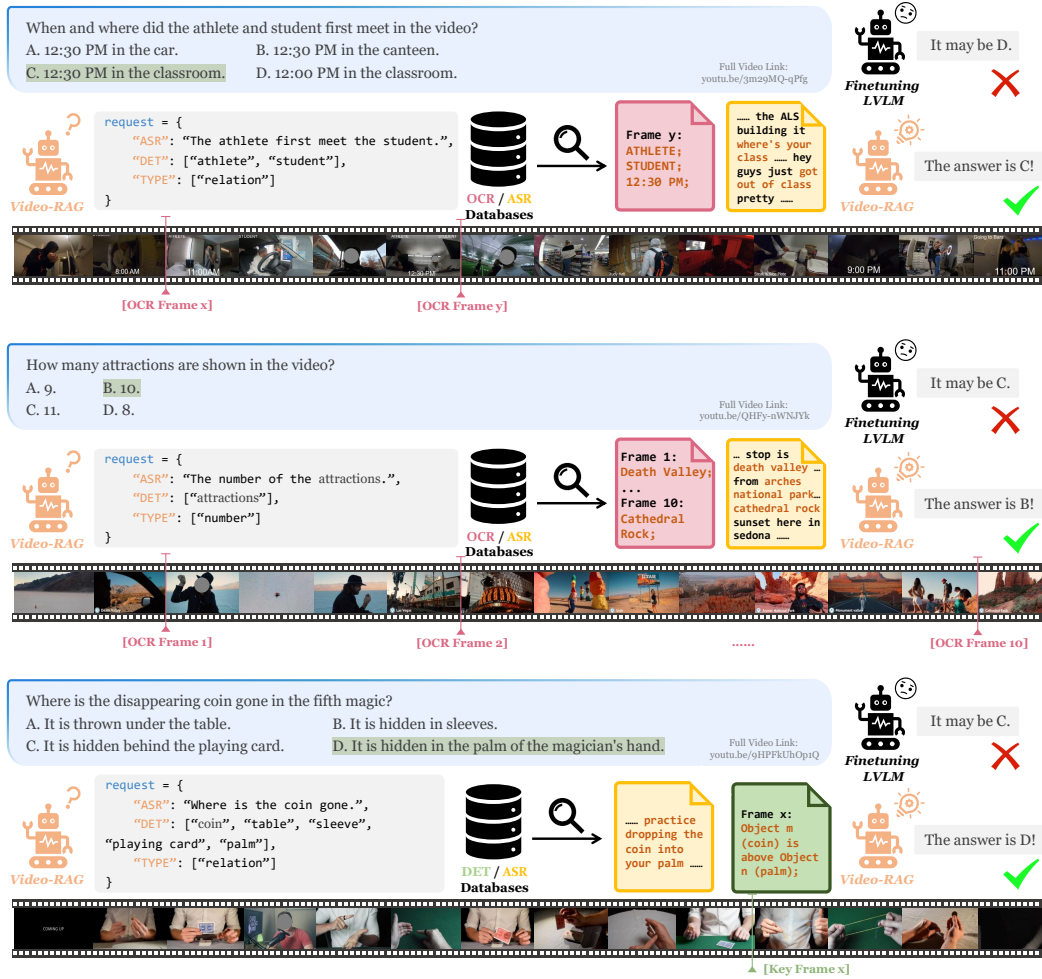


Figure 4: Qualitative results of LLaVA-Vdideo when applying Video-RAG.

In this section, we show more results of LLaVA-Vdideo-7B when applying Video-RAG in different examples in Figure 4. The figure highlights several representative cases involving detailed video comprehension from Video-MME [2]. As illustrated, augmenting LLaVA-Video with external tools to process and retrieve auxiliary texts from videos significantly enhances its ability to reduce visual hallucinations, thereby enabling more accurate and confident responses to user queries.

References

- [1] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2025.
- [2] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.
- [3] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.
- [4] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023.
- [5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.
- [6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [7] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024.
- [8] Yin Song and Chen Wu and Eden Duthie. aws-prototyping/long-llava-qwen2-7b, 2024. URL <https://huggingface.co/aws-prototyping/long-llava-qwen2-7b>.
- [9] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkan Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.
- [10] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024. URL <https://llava-vl.github.io/blog/2024-04-30-llava-next-video/>.
- [11] Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic framework for benchmarking video mllms. *arXiv preprint*, 2024.
- [12] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.