

TL;DR Video RAG extracts audio transcripts, on screen text, and object cues from long videos, aligns them to frames/clips, and feeds them as auxiliary texts alongside sampled frames to an LVLM in a single turn retrieval step. This training free approach improves accuracy on long video QA/understanding while keeping compute low and remaining model agnostic.

---

## Context (Nov-Dec 2024)

The paper reports: submitted 20 Nov 2024; last revised 20 Dec 2024. It demonstrates consistent gains on long video benchmarks (Video MME, MLVU, LongVideoBench) and highlights that a strong open model (e.g., a 72B LVLM) with Video RAG can surpass some proprietary systems on these tasks.

References:

- arXiv: <https://arxiv.org/abs/2411.13093>
  - Overview: <https://learnopencv.com/video-rag-for-long-videos/>
  - Repo (reference): <https://github.com/Leon1207/Video-RAG-master>
- 

## What is Video RAG?

Large video language models struggle with hour long videos due to context limits and information dispersion. Video RAG uses a retrieval augmented approach that turns raw video into a compact, visually aligned text corpus which the LVLM can efficiently search and reason over in one pass.

Key ideas:

- Visually aligned auxiliary texts: Extract ASR (audio transcripts), OCR (on screen text), and open vocabulary object detections; timestamp and align them to frames/clips.
- Single turn retrieval: One lightweight retrieval step selects the most relevant snippets based on the user query.
- Plug and play with any LVLM: Works as an input side augmentation; no LVLM fine tuning required.

Benefits:

- Training free: No new model weights to train.
  - Low overhead: Single turn retrieval keeps compute reasonable for long videos.
  - Broad compatibility: Integrates with different LVLM backbones.
- 

## Pipeline at a glance

### 1. Pre-processing/Indexing

- Sample frames/clips from the long video.
- Run ASR over audio; OCR over frames; open vocabulary detection (objects/scenes).
- Build a temporally aligned “evidence” store mapping timestamps (text spans, object tags).

### 2. Retrieval (single-turn)

- Embed the user query and the evidence store (or use sparse retrieval).
- Select top k clips/evidence that best match the query.

### 3. LVLM reasoning

- Feed the LVLM with: a small set of frames (or keyframes) + the retrieved auxiliary texts + the user question.
  - Generate an answer grounded in both visuals and aligned texts.
- 

## Why it matters

- Scalability: Extends long video comprehension without custom training or massive context windows.
  - Robustness: Auxiliary texts capture content LVLMs might miss from frames alone (e.g., signage, slides, small objects, speech details).
  - Practicality: Training free and model agnostic makes it easier to deploy across teams and stacks.
-

# Getting started (adaptation tips)

While implementations vary, a practical setup follows:

- Frame sampling: Uniform + scene change aware sampling to cover content shifts.
- Text extraction: High quality ASR for speech; OCR tuned for subtitles/overlays; optional speaker diarization.
- Visual tags: Open vocabulary/object detection for entities/events; store labels with timestamps.
- Indexing: Build a searchable store (e.g., embeddings or BM25) with timestamp links back to clips.
- Prompting: Concise instruction templates that explain the auxiliary texts and ask for grounded answers.

Operational hints:

- Limit per query context: cap top k evidence to stay within LVLM token limits.
  - Prefer precise spans: include short, timestamped snippets instead of full transcripts.
  - Multilingual: ensure ASR/OCR support the language(s) present in the video.
- 

## Practical notes

- Single turn vs multi turn: The paper focuses on single turn retrieval to keep overhead predictable; multi turn agents add cost and latency.
  - Hallucination control: Keep evidence concise and ask for citations (timestamps) in the LVLM output.
  - Evaluation: Use long video QA sets and track both accuracy and latency.
- 

## References

- Paper: <https://arxiv.org/abs/2411.13093>
- LearnOpenCV explainer: <https://learnopencv.com/video-rag-for-long-videos/>
- Repository (reference): <https://github.com/Leon1207/Video-RAG-master>

Notes: Benchmarks and claims reflect public sources at the stated dates. Validate performance with your LVLM, retrieval stack, and video domain.