Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081534

ARTICLE

Intra-Video Temporal-Aware RAG: A Self-Contained Framework for Video-Based Question Answering

Sumaira Shafiq[1], **Naveed Ejaz**[2], **Munam Ali Shah**[3,*], **Rashid Kamal**[2], **Adnan Sohail**[1] and **Sheraz Aslam**[4,5,6]

[1]Department of Computing and Technology, Islamabad Campus, Iqra University, Islamabad, Pakistan
[2]School of Computing, Ulster University, Belfast, UK
[3]Department of Computer Networks and Communication, College of Computer Science and Information Technology, King Faisal University, Al-Ahsa, Saudi Arabia
[4]Department of Computer Science, CTL Eurocollege, Limassol, Cyprus
[5]Department of Computer Science, American University of Cyprus, Larnaca, Cyprus
[6]International Digital Economy College, Minjiang University, Fuzhou, China

*Corresponding Author: Munam Ali Shah. Email: mashah@kfu.edu.sa

Received: 09 March 2026; Accepted: 21 April 2026; Published: 15 June 2026

ABSTRACT: Lecture videos are widely used in modern education, yet answering questions from them remains challenging. Relevant information is often distributed across time and expressed through multiple modalities, including speech, slides, and visual content. Existing VideoQA approaches, including recent retrieval-augmented generation (RAG) methods, typically rely on static text representations or global video features. Consequently, they may retrieve evidence that is semantically relevant but temporally misaligned, leading to inaccurate or weakly grounded responses. In addition, dependence on external knowledge sources can introduce hallucinations and reduce reliability in educational settings. To address these limitations, we propose a temporally aware, intra-video RAG framework tailored for lecture videos. The approach aligns automatic speech transcripts and visual captions into timestamped segments and performs retrieval constrained by temporal boundaries. Retrieved segments are further refined using a cross-encoder before answer generation, ensuring that responses are grounded in the correct portions of the video. We evaluate the proposed method on the LectQA-Vid dataset, consisting of 100 lecture videos and 3000 temporally annotated questions. Experimental results demonstrate improved factual alignment and robustness over non-temporal baselines, highlighting the importance of temporal grounding in lecture VideoQA.

KEYWORDS: Video question answering; retrieval-augmented generation; temporal grounding; multimodal retrieval; educational videos; whisper ASR; visual captioning; large language models; explainable AI; timestamped evidence

1 Introduction

In recent years, Video Question Answering (VideoQA) has become a popular research area in multimedia artificial intelligence. The goal of VideoQA is to generate responses to diverse questions from video content using visual, spoken, and temporal information [1]. The VideoQA problem differs from visual question answering (VQA), which focuses on a single image [2]. VideoQA requires reasoning over temporal sequences in which relevant cues may be partially observable, appear briefly, or emerge at different times across different modalities [3].

Video lectures have become a popular way to learn. In lecture videos, the required information to answer a particular question is usually distributed over time and may also rely on different communication modalities, including spoken explanations, slides, diagrams, and on-screen text [4]. Therefore, effective VideoQA in lecture videos demands robust multimodal understanding, fine-grained temporal grounding, and the retrieval of relevant evidence from specific moments in the video.

Retrieval-Augmented Generation (RAG) pipelines [5,6] have recently been used for open-domain VideoQA by combining dense retrieval with the generative reasoning capabilities of large language models (LLMs). Despite their success, conventional RAG approaches typically assume the availability of structured, self-contained textual passages and provide no mechanisms for temporal alignment or multimodal grounding [7]. Most of these methods do not incorporate temporal grounding and thus do not guarantee that the retrieved segments cover the appropriate moments in the video timeline. Moreover, VideoQA systems that use LLMs often rely heavily on external text corpora, which may lead to hallucinations. These issues demand a temporally aware, multimodal, and intra-video retrieval-and-reasoning framework for VideoQA in educational videos.

To address these challenges, we propose a temporal-aware retrieval-augmented generation (RAG) framework designed specifically for lecture videos. The main idea is to use the video's temporal structure to generate answers d from the correct parts of the lecture. To achieve this, we first combine transcripts and visual captions into meaningful segments that are aligned with the video timeline. When a question is asked, the system searches for relevant segments within the appropriate time ranges and then refines the results to retain only the most useful ones. Finally, a language model generates the answer using only this selected evidence. This ensures that the responses are accurate, based on the lecture, and easy to understand.

To evaluate the proposed framework, we developed a dataset called LectQA-Vid. This dataset consists of 100 computer science lecture videos collected from YouTube, each ranging from 2 to 5 min long. Three annotators created a set of 3000 temporally aligned question–answer pairs. Each question was assigned one of three difficulty levels: "Simple", "Hard", or "Very Hard". On the LectQA-Vid dataset, the model achieves semantic similarity of 0.71–0.77 and F1 scores of 0.16–0.29 for open-ended questions, and 50%–57% accuracy for multiple-choice questions. Moreover, ablation results highlight the importance of temporal filtering, multimodal fusion, and timestamp grounding.

The major contributions of this work are as follows:

- We propose a **temporally aware, intra-video** retrieval-augmented generation framework for lecture VideoQA that enforces **timestamp-constrained evidence selection**, ensuring that generated answers are grounded exclusively in the source video.
- We introduce a unified **multimodal pre-processing pipeline** that integrates Whisper ASR transcripts and Gemini-based visual captions into **temporally aligned semantic units** suitable for dense retrieval.
- We develop a **temporally aware retrieval strategy** that combines FAISS-based dense retrieval with explicit timestamp filtering and cross-encoder re-ranking, enabling **verifiable and interpretable grounding**.

The remainder of this paper is arranged as follows. Section 2 provides a brief overview of the existing research in VideoQA. Section 3 discusses the LectQA-Vid dataset. Section 4 discusses the proposed Temporal-Aware RAG framework. Section 5 describes the experimental setup and detailed experimental results. Finally, Section 6 concludes the work.

2 Related Work

The goal of VideoQA is to answer natural-language questions using visual and audio cues from videos. VideoQA methods involve the usage of temporal dynamics, motion cues, and long-range semantic dependencies, which make the task extremely challenging [1,8,9]. This section provides a brief review of the existing methods in VideoQA. It also discusses methods for temporal reasoning and grounding. The section also briefly discusses RAG-based approaches for VideoQA.

Typical VideoQA pipelines involve steps including video feature extraction, question encoding, multi-modal fusion, and answer prediction [1]. Spatio-temporal reasoning is a fundamental step of video question answering, as many questions require the joint capture of spatial relationships and temporal dependencies across frames. Early VideoQA approaches model spatio-temporal information using appearance and motion features extracted at the frame or clip level [10,11]. Graph-based methods model temporal, spatial, and visual-linguistic relations by representing video units as nodes and enabling interactions with question structures [12–14]. However, their use of nodes with mixed semantics limits the explicit alignment of video and questions.

Transformer-based architectures can model complex dependencies across multimodal inputs. Transformer-based video–language models [15–17] adopt frame- or clip-level features as visual inputs. Yang et al. [18] used subtitles and visual concepts using BERT. Furthermore, Yang et al. [19] conducted a comparative study of text-language Transformers, including BERT, XLNet, RoBERTa, and ALBERT. They demonstrated that Transformer-based architectures can better capture complex multimodal semantics than recurrent models. Garcia et al. proposed a knowledge-driven framework called KnowIT VQA [20], which incorporated external knowledge sources by retrieving and integrating knowledge representations with video and textual features. Wu et al. [21] investigated knowledge-oriented transfer learning in VideoQA by distinguishing between domain-specific and domain-agnostic knowledge, and transferring the latter across datasets to improve generalization. However, existing Transformer-based, knowledge-driven, and transfer learning approaches primarily rely on global video representations, external knowledge, or dataset-level knowledge transfer. They do not explicitly enforce temporal grounding during retrieval or reasoning. As a result, retrieved or attended evidence can be semantically relevant but temporally misaligned. In contrast, our work explicitly models intra-video temporal structure via timestamp-constrained retrieval, ensuring that evidence aligns with the correct temporal segments.

Temporal moment localization has been incorporated in VideoQA, as answering many questions requires first localizing the relevant temporal moment before reasoning over its visual content. Related research on temporal moment localization aims to align a natural-language query with its corresponding temporal segment in a video, i.e., identifying the start and end timestamps at which the described event occurs, typically using cross-modal attention [22] and structured temporal modeling [23,24]. Grounded VideoQA frameworks further extend this idea by highlighting supporting frames or regions during answer inference [25,26]. However, these approaches are typically embedded in end-to-end architectures that lack explicit retrieval mechanisms or timestamp constraints, leading models to attend to semantically relevant yet temporally misaligned moments. This is especially critical in lecture videos, where information is dense and highly structured over time.

RAG improves factual grounding by using retrieved evidence during generation. The original RAG framework retrieves text prior to LLM decoding [27], whereas later multimodal extensions incorporate images and cross-modal features [28]. However, most RAG systems are built for static data and do not model temporal structure. In VideoQA, existing approaches rely mainly on semantic similarity and do not enforce retrieval from the correct time segments. As a result, evidence may be relevant but temporally misaligned.

Across the above areas, three key gaps remain. First, existing VideoQA and grounding methods rarely incorporate explicit timestamp constraints, leading to semantically relevant but temporally misaligned evidence. Second, multimodal RAG systems lack mechanisms to constrain retrieval to a specific video or temporal window, which is essential for factual alignment in lecture settings. Third, current educational VideoQA datasets generally lack timestamped evidence and do not enforce Intra-video grounding, limiting transparency and verifiability. This work addresses these gaps through a temporally aware Intra-video RAG framework that retrieves multimodal, timestamped evidence directly from the lecture video. By combining Whisper transcriptions, Gemini keyframe captions, BGE embeddings, FAISS retrieval, temporal filtering, and cross-encoder re-ranking, the proposed method produces verifiable, grounded answers tailored to the structure of educational lecture videos.

3 Dataset: LectQA-Vid

For the evaluation of temporally grounded retrieval and reasoning in educational VideoQA, we constructed a benchmark dataset called "LectQA-Vid" with explicit temporal supervision. This section describes the LectQA-Vid dataset, including its source videos, annotation pipeline, question design, and scope, independent of the modeling and retrieval techniques introduced later.

3.1 Corpus Construction and Video Selection

Videos in LectQA-Vid were selected from YouTube via keyword searches, including queries such as "short computer science lecture" and topic-specific terms, including operating systems, algorithms, artificial intelligence, computer architecture, and machine learning. From the retrieved results, we sampled 100 videos subject to quality and format constraints.

We retained videos that (i) are in English, (ii) have clear and intelligible audio narration, (iii) are short (2–5 min), (iv) feature a single speaker, and (v) contain minimal background music. We focused on slide-intensive, lecture-style videos, in which the instructor's narration accompanies on-screen slides. To encourage diversity and reduce redundancy, videos were drawn from different channels. The resulting corpus consists of $N = 100$ videos. Each video focuses on a computer science topic, including (but not limited to) operating systems, algorithms, databases, networking, and programming languages, etc.

LectQA-Vid was intentionally constructed from computer science lectures featuring a single speaker and a slide-based format to provide a controlled and consistent experimental setting. This design isolates the impact of temporal alignment from additional variability introduced by multi-speaker interactions, diverse domains, or heterogeneous presentation styles.

3.2 Modalities and Derived Annotations

For each video, we derived two complementary textual modalities:

- **Transcript (Whisper ASR).** The audio track was transcribed using Whisper (large-v3), producing a time-aligned transcript that served as the primary textual source for question answering.
- **Visual captions (Gemini).** A set of keyframes was extracted from each video, and captions were generated using a Gemini-based vision–language model. These captions summarize salient on-screen visual information. While OCR-based approaches can extract textual content from slides, they are limited in capturing semantic and contextual information beyond visible text. Therefore, we adopted a vision-language model to generate richer visual captions that better support downstream reasoning.

3.3 Data Representation

The dataset is defined as:

$$\mathcal{D} = \{(v_i, \mathcal{Q}_i)\}_{i=1}^{N},$$

where v_i denotes the i^{th} video (represented by its YouTube link and metadata), and $\mathcal{Q}_i$ denotes the associated set of question–answer pairs. For each video v_i, we also provide metadata such as title, topic, duration, and the derived transcript and visual captions.

For each video v_i, we define:

$$\mathcal{Q}_i = \{(q_{ij}, a_{ij}^{*}, \tau_{ij}, d_{ij}, y_{ij})\}_{j=1}^{M_i},$$

where:

- q_{ij} is the j^{th} question associated with video v_i;
- a_{ij}^{*} is the corresponding **ground-truth answer**, manually created and verified from the video content;
- $\tau_{ij} = [t_{ij}^{\text{start}}, t_{ij}^{\text{end}}] \subseteq [0, T_i]$ is the annotated temporal interval within v_i supporting a_{ij}^{*} (with T_i the video duration);
- $d_{ij} \in \{\text{“Simple”}, \text{“Hard”}, \text{“Very Hard”}\}$ denotes question difficulty;
- $y_{ij} \in \{\text{MCQ}, \text{Open}\}$ denotes question type (multiple-choice or open-ended).

The total number of question–answer pairs across the corpus is:

$$M = \sum_{i=1}^{N} M_i.$$

3.4 Question Design and Grounding Constraint

From each video, 15 multiple-choice questions (MCQs) and 15 open-ended questions were designed. Each MCQ presents four possible answers, with only one correct option. The questions in the dataset were categorized into three levels: “Simple”, “Hard”, and “Very Hard”, based on their difficulty. The difficulty labels are inherently subjective; thus, annotators were provided with clear guidelines, and a consensus on difficulty labels was achieved during the annotation review stage.

3.5 Annotation Pipeline and Quality Control

Three computer science graduates annotated the videos in the dataset. All annotators possessed subject matter expertise and adhered to standardized guidelines. They reviewed the complete video, Whisper-generated transcript, and Gemini-generated keyframe captions.

For every question–answer pair, annotators labeled a ground-truth temporal interval as:

$$\tau = [t^{\text{start}}, t^{\text{end}}],$$

representing a *single contiguous supporting span* that contains sufficient evidence to answer the question. Temporal boundaries were defined using transcript-aligned timestamps to ensure consistency between textual and temporal annotations. Annotators were instructed to mark a *reasonable supporting span* rather than the minimal possible interval; boundaries are intended to be approximate (i.e., looser but safe) rather than frame-precise. When the necessary evidence was distributed across multiple portions of the lecture (e.g., a concept introduced earlier and applied later, or an audio explanation paired with a slide containing a key formula), τ was chosen to cover the whole region encompassing all required supporting content.

All annotated questions were subsequently reviewed by the first author, who verified the correctness of answer pairs, the alignment between the annotated temporal interval and the supporting video content, and consistency with the assigned difficulty labels. Ambiguous or weakly grounded questions were revised or removed during this review. To assess annotation consistency, a random subset of the dataset was independently inspected by a second annotator. Agreement was evaluated qualitatively with respect to answer correctness and temporal-span alignment, indicating high consistency in question interpretation and the localization of supporting temporal intervals, with only minor boundary-level differences that did not affect answerability.

3.6 Data Splits

We construct an 80/10/10 train–validation–test split over the 100 videos, while maintaining a balanced distribution of difficulty levels across splits. Unless otherwise stated, experiments are reported on a held-out evaluation subset of 1000 QA pairs (500 MCQ and 500 open-ended) sampled from the complete set of 3000 QA pairs.

3.7 Illustrative Examples and Dataset Statistics

Table 1 summarises the LectQA-Vid dataset. The dataset consists of 100 short computer science lecture videos (2–5 min each). The dataset includes 3000 question–answer pairs, which are evenly split between MCQs and open-ended questions. The questions are of three difficulty levels ("Simple", "Hard", "Very Hard"). Each video contributes 30 questions (15 MCQs and 15 open-ended). Additionally, the dataset includes time-aligned ASR transcripts (Whisper), Gemini-based visual captions, and precise temporal spans for each question. Table 2 shows examples of questions in the LectQA-Vid dataset along with the groundtruth answers.

Table 1: Dataset statistics for the LectQA-Vid lecture VideoQA corpus.

Category	Value
Total Number of Videos	100
Video Type	Computer science micro-lectures (YouTube)
Topics Covered	OS, Networking, Databases, Algorithms, AI/ML
Video Duration	2–5 min
Total Number of QA Pairs	3000 (1500 MCQ + 1500 open-ended)
Question Types	MCQ (4 options) + open-ended
QA Difficulty Levels	Simple, Hard, Very Hard
Distribution Across Difficulty Levels	1000 Simple, 1000 Hard, 1000 Very Hard
Questions per Video	30 (15 MCQ + 15 open-ended)
MCQ Difficulty Split (per video)	5 Simple, 5 Hard, 5 Very Hard
Open-ended Difficulty Split (per video)	5 Simple, 5 Hard, 5 Very Hard
Average Transcript Length	8.3k characters/video
ASR Transcript	Whisper (large-v3), time-aligned
Visual Captions	Gemini-generated keyframe captions (timestamped)
Temporal Supervision	$\tau = [t^{\text{start}}, t^{\text{end}}]$ per QA (timestamp-aligned)
Provided Artifacts	YouTube links + transcripts + keyframe captions + QA pairs
Knowledge Constraint	Intra-video only; no external knowledge

Table 2: Examples from the LectQA-Vid dataset.

Question	Reference Answer	Difficulty	Primary Grounding
What assumption is made when defining the empirical risk minimization objective in the lecture?	The objective assumes that training samples are independently and identically distributed and that the expected risk can be approximated by the empirical average shown on the slide.	Simple	Transcript + Slide
Why does the instructor prefer stochastic gradient descent over full-batch gradient descent in the optimization discussion?	Because stochastic gradient descent enables scalable learning on large datasets and provides faster convergence in practice, as explained using the optimization slides.	Hard	Transcript
What limitation of the baseline algorithm is highlighted by the convergence plot shown in the lecture?	The convergence plot shows that the baseline algorithm converges slowly due to a fixed learning rate, motivating the adaptive method introduced later.	Very Hard	Visual (Plot)

4 Materials and Methods

This section provides a detailed description of the proposed Temporal-Aware RAG framework. Fig. 1 provides an overview of the proposed framework.

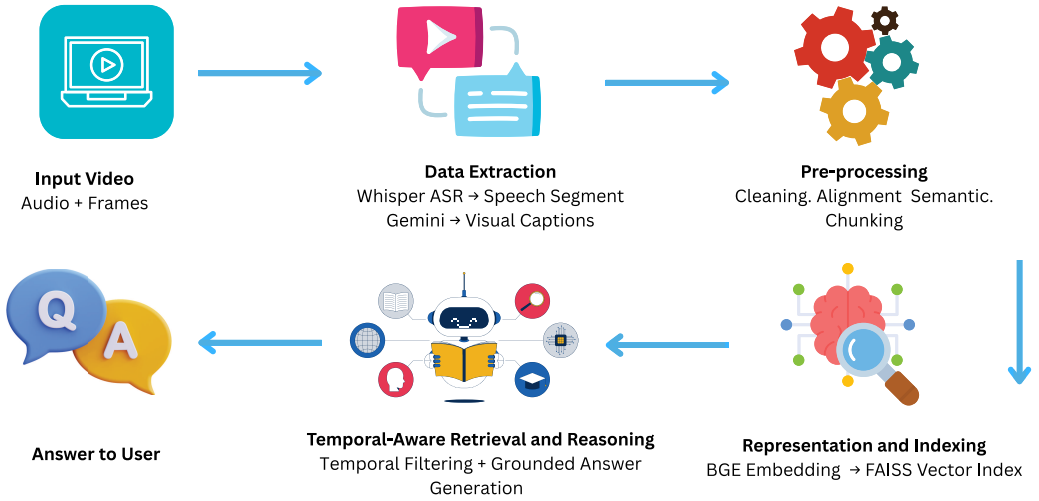


Figure 1: Overview of the proposed temporal-aware, Intra-video RAG pipeline for lecture VideoQA.

4.1 Data Extraction

Let a video dataset be represented as a collection $\mathcal{V} = \{v_i\}_{i=1}^{N}$, where each video v_i is characterized by its visual frame sequence and corresponding audio track, $v_i = \{\mathbf{I}_i(t), \mathbf{a}_i(t) \mid t \in [0, T_i]\}$, with $\mathbf{I}_i(t)$ denoting the RGB frame at time t and $\mathbf{a}_i(t)$ the corresponding audio signal, and T_i the total duration of v_i.

The goal of the data extraction phase is to convert each raw video v_i into two temporally aligned, semantically meaningful textual modalities: (1) a sequence of linguistic segments obtained from automatic speech recognition (ASR), and (2) a sequence of visual sentences obtained from keyframe captioning. Formally, this phase produces the intermediate representation:

$$\mathcal{S}_i = \mathcal{S}_i^{(\mathrm{a})} \cup \mathcal{S}_i^{(\mathrm{v})} \tag{1}$$

where $\mathcal{S}_i^{(\mathrm{a})}$ and $\mathcal{S}_i^{(\mathrm{v})}$ denote audio-derived and vision-derived textual items, respectively.

4.1.1 Audio Transcription

The audio component $\mathbf{a}_i(t)$ is processed by an automatic speech recognition model $f_{\mathrm{ASR}}(\cdot)$, implemented using Whisper. The model generates a set of $N_i^{(\mathrm{a})}$ speech segments:

$$\mathcal{S}_i^{(\mathrm{a})} = \left\{\left(s_{ij}^{(\mathrm{a})}, \tau_{ij}^{\mathrm{start}}, \tau_{ij}^{\mathrm{end}}\right)\right\}_{j=1}^{N_i^{(\mathrm{a})}} \tag{2}$$

where $s_{ij}^{(\mathrm{a})}$ denotes the transcribed text and $(\tau_{ij}^{\mathrm{start}}, \tau_{ij}^{\mathrm{end}})$ are the corresponding temporal boundaries. Equivalently, each segment is obtained via the mapping:

$$f_{\mathrm{ASR}} : \mathbf{a}_i(t) \mapsto \left(s_{ij}^{(\mathrm{a})}, \tau_{ij}^{\mathrm{start}}, \tau_{ij}^{\mathrm{end}}\right) \tag{3}$$

providing temporally localized linguistic evidence aligned with the spoken narrative of the video.

4.1.2 Keyframe Extraction and Captioning

The visual stream $\mathbf{I}_i(t)$ is sampled to obtain a set of representative keyframes:

$$\mathcal{K}_i = \{\mathbf{I}_i(t_{ik})\}_{k=1}^{N_i^{(\mathrm{v})}} \tag{4}$$

where the sampling instants $\{t_{ik}\}$ are determined either by scene-change detection or by uniform temporal intervals Δt. Each keyframe is converted into a natural-language description using a vision–language captioning model $f_{\mathrm{VLM}}(\cdot)$, instantiated here as Gemini. Video transcripts are generated using Whisper, while visual captions are produced using Gemini. The resulting set of visual sentences is:

$$\mathcal{S}_i^{(\mathrm{v})} = \left\{\left(s_{ik}^{(\mathrm{v})}, t_{ik}\right)\right\}_{k=1}^{N_i^{(\mathrm{v})}} \tag{5}$$

where $s_{ik}^{(\mathrm{v})} = f_{\mathrm{VLM}}(\mathbf{I}_i(t_{ik}))$ encodes the semantic content of frame $\mathbf{I}_i(t_{ik})$ at time t_{ik}.

4.1.3 Temporal Unification

In the next step, the two modalities are temporally synchronized into a unified sequence ordered by time:

$$\mathcal{S}_i = \mathrm{sort}_t\left(\mathcal{S}_i^{(\mathrm{a})} \cup \mathcal{S}_i^{(\mathrm{v})}\right) = \{(s_{im}, \tau_{im}^{\mathrm{start}}, \tau_{im}^{\mathrm{end}})\}_{m=1}^{L_i} \tag{6}$$

where sorting ensures chronological coherence. This unified representation $\mathcal{S}_i$ is then used for subsequent preprocessing, semantic chunking, and embedding steps.

4.2 Data Pre-Processing

The unified sequence $\mathcal{S}_i = \{(s_{im}, \tau_{im}^{\text{start}}, \tau_{im}^{\text{end}})\}_{m=1}^{L_i}$ obtained from the data extraction phase may be noisy and redundant. Therefore, preprocessing is performed to transform $\mathcal{S}_i$ into a temporally consistent set of semantically meaningful textual units. Formally, this transformation is represented as:

$$g_{\text{prep}} : \mathcal{S}_i \longrightarrow \mathcal{C}_i = \{(c_{in}, t_{in}^{\text{start}}, t_{in}^{\text{end}})\}_{n=1}^{N_i} \tag{7}$$

where each c_{in} denotes a normalized, semantically coherent chunk, temporally bounded by $(t_{in}^{\text{start}}, t_{in}^{\text{end}})$. The rest of the section describes the pre-processing steps.

4.2.1 Text Normalization and Noise Removal

Each segment s_{im} is normalized using the function $\tilde{s}_{im} = f_{\text{norm}}(s_{im})$, which removes non-linguistic tokens (e.g., [music], [applause]), repeated punctuation, and capitalization inconsistencies. Segments with $\|\tilde{s}_{im}\| < \epsilon_{\text{len}}$ (shorter than a minimum token threshold) are discarded. Moreover, duplicate or near-duplicate sentences are eliminated using a cosine-similarity filter.

4.2.2 Sentence Boundary Refinement

Long ASR sentences are split into smaller parts using punctuation and grammar rules. These parts are then combined again when they are temporally close and syntactically relevant.

If $\mathcal{B}(\tilde{s}_{im}) = \{\tilde{s}_{im}^{(1)}, \ldots, \tilde{s}_{im}^{(r)}\}$ denote the boundary refinement operator producing r refined sub-sentences, the updated sequence after boundary refinement is

$$\mathcal{S}_i' = \bigcup_{m=1}^{L_i} \mathcal{B}(\tilde{s}_{im}) \tag{8}$$

4.2.3 Temporal Alignment across Modalities

In Eq. (8), $\mathcal{S}_i'$ combines both audio and visual textual items. Next, a temporal alignment function f_{align} is used to ensure consistent ordering and gap filling. The function is defined as:

$$\mathcal{S}_i'' = f_{\text{align}}(\mathcal{S}_i') = \{(s_{ip}, t_{ip}^{\text{start}}, t_{ip}^{\text{end}}, \eta_{ip})\}_{p=1}^{P_i} \tag{9}$$

where $\eta_{ip} \in \{\text{audio}, \text{visual}\}$ indicates the source modality.

In this step, audio transcripts and visual captions are fused at the text level by aligning modality-specific segments based on timestamps and combining them into a single sequence while preserving temporal order. We define a temporal overlap threshold θ_t to resolve any potential conflicts between modalities. The temporal overlap between two segments is defined as:

$$\text{overlap}(s_{ip}, s_{iq}) = \max\left(0,\ \min(t_{ip}^{\text{end}}, t_{iq}^{\text{end}}) - \max(t_{ip}^{\text{start}}, t_{iq}^{\text{start}})\right) \tag{10}$$

If the overlap between an audio and visual segment exceeds θ_t, the segments are considered temporally redundant. In such cases, the visual segment is either suppressed or merged into the corresponding audio segment, with priority given to the audio transcript.

Alternative fusion strategies, such as early feature-level fusion or late fusion of independently processed modalities, were considered. However, early fusion increases model complexity and requires cross-modal embedding alignment, while late fusion may lead to fragmented or inconsistent context across modalities. In contrast, the proposed temporally aligned text-level fusion provides a simple, interpretable, and computationally efficient mechanism that preserves both semantic coherence and temporal structure.

4.2.4 Semantic Chunking

Consecutive sentences in $\mathcal{S}_i''$ that are semantically related and temporally contiguous are merged into higher-level units, termed *semantic chunks*. Let $\phi_{\text{sem}}(\cdot)$ denote a semantic affinity function between two adjacent sentences:

$$\phi_{\text{sem}}(s_{ip}, s_{i(p+1)}) = \frac{\mathbf{h}_{ip}^{\top}\mathbf{h}_{i(p+1)}}{\|\mathbf{h}_{ip}\|\|\mathbf{h}_{i(p+1)}\|} \tag{11}$$

where $\mathbf{h}_{ip}$ are contextual embeddings from a pretrained sentence encoder.

We introduce a semantic similarity threshold θ_{sem} to control chunk formation, and a temporal gap threshold δ_{gap} to enforce temporal continuity. The threshold δ_{gap} defines the maximum allowable time difference between two consecutive sentences for them to be considered temporally adjacent. Two sentences are merged if

$$\phi_{\text{sem}}(s_{ip}, s_{i(p+1)}) > \theta_{\text{sem}} \quad \text{and} \quad t_{i(p+1)}^{\text{start}} - t_{ip}^{\text{end}} < \delta_{\text{gap}},$$

ensuring both semantic coherence and temporal continuity.

The merged chunk c_{in} is defined as

$$c_{in} = \bigoplus_{p=p_s}^{p_e} s_{ip}, \qquad t_{in}^{\text{start}} = t_{ip_s}^{\text{start}}, \qquad t_{in}^{\text{end}} = t_{ip_e}^{\text{end}} \tag{12}$$

where $\oplus$ denotes string concatenation preserving sentence order. The resulting set of chunks $\mathcal{C}_i$ forms the temporally grounded, semantically consolidated representation of video v_i.

4.2.5 Output Representation

After pre-processing, each video v_i is represented by $\mathcal{C}_i = \{(c_{in}, t_{in}^{\text{start}}, t_{in}^{\text{end}})\}_{n=1}^{N_i}$, where each c_{in} corresponds to a normalized, semantically coherent textual unit which is associated with a temporal interval within the video. This structured representation serves as input to the embedding and indexing stage, as discussed in Section 4.3.

4.3 Representation and Indexing

The pre-processed video representation $\mathcal{C}_i = \{(c_{in}, t_{in}^{\text{start}}, t_{in}^{\text{end}})\}_{n=1}^{N_i}$ encapsulates semantically coherent textual units. Each textual unit is associated with a temporal interval in video v_i. Each chunk c_{in} is then mapped to a dense vector representation in a continuous embedding space $\mathbb{R}^d$. This transformation is achieved by a sentence-level embedding model $f_{\text{emb}}(\cdot)$:

$$\mathbf{z}_{in} = f_{\text{emb}}(c_{in}) \in \mathbb{R}^d \tag{13}$$

where $\mathbf{z}_{in}$ is a d-dimensional vector encoding the semantic and contextual meaning of chunk c_{in}.

4.3.1 Embedding Function

The embedding model $f_{\text{emb}}(\cdot)$ is instantiated using a pre-trained Bidirectional Generalized Embedding (BGE) encoder, denoted as BGE_{base}. For each textual input c_{in} composed of L_{in} tokens, the encoder produces token-level hidden states $\mathbf{H}_{in} = [\mathbf{h}_{in}^{(1)}, \ldots, \mathbf{h}_{in}^{(L_{in})}]$, which are mean-pooled to obtain the chunk-level embedding:

$$\mathbf{z}_{in} = \frac{1}{L_{in}} \sum_{\ell=1}^{L_{in}} \mathbf{h}_{in}^{(\ell)} \tag{14}$$

The embeddings are then ℓ_2-normalized:

$$\hat{\mathbf{z}}_{in} = \frac{\mathbf{z}_{in}}{\|\mathbf{z}_{in}\|_2} \tag{15}$$

The embedding dimension d is fixed ($d = 768$ for BGE_{base}).

4.3.2 FAISS Index Construction

All normalized embeddings from the dataset are aggregated into a single matrix: $\mathbf{Z} = [\hat{\mathbf{z}}_{11}, \hat{\mathbf{z}}_{12}, \ldots, \hat{\mathbf{z}}_{in}, \ldots]^\top \in \mathbb{R}^{M \times d}$, where $M = \sum_{i=1}^{N} N_i$ is the total number of chunks across all videos. Each embedding $\hat{\mathbf{z}}_{in}$ is associated with metadata $\mu_{in} = \{\text{video\_id} = i, \text{chunk\_id} = n, t_{in}^{\text{start}}, t_{in}^{\text{end}}\}$, linking the vector to its source and temporal span.

For efficient approximate nearest-neighbor (ANN) search, we employ the FAISS (Facebook AI Similarity Search) library to construct an index $\mathcal{I}$:

$$\mathcal{I} = \text{FAISS.build}(\mathbf{Z}, \text{metric} = \text{cosine}) \tag{16}$$

where FAISS organizes the embeddings using an inverted-file index with product quantization (IVF-PQ) for sub-linear retrieval. This yields a mapping: $f_{\text{index}} : \hat{\mathbf{z}}_{in} \mapsto \mathcal{I}$, allowing rapid lookup of semantically similar vectors.

4.3.3 Similarity Formulation

Given a query embedding $\hat{\mathbf{z}}_q = f_{\text{emb}}(q)$ derived from a textual question q, semantic similarity between the query and a candidate chunk $\hat{\mathbf{z}}_{in}$ is computed using cosine similarity:

$$\text{sim}(\hat{\mathbf{z}}_q, \hat{\mathbf{z}}_{in}) = \hat{\mathbf{z}}_q^\top \hat{\mathbf{z}}_{in} \tag{17}$$

We define a retrieval similarity threshold θ_r to characterize relevance, and only those chunks with similarity greater than θ_r are selected as semantically relevant candidates. For large-scale retrieval, FAISS efficiently estimates the top-k nearest neighbors:

$$\mathcal{R}_q^{(k)} = \text{TopK}_{(\hat{\mathbf{z}}_{in})} \left[\text{sim}(\hat{\mathbf{z}}_q, \hat{\mathbf{z}}_{in}) \right],$$

where k denotes the number of retrieved candidates.

The resulting set is a ranked list of chunk identifiers and corresponding similarity scores:

$$\mathcal{R}_q^{(k)} = \left\{ (i, n, \text{sim}_{in}) \mid \text{sim}_{in} = \hat{\mathbf{z}}_q^\top \hat{\mathbf{z}}_{in} \right\}_{\text{top-}k} \tag{18}$$

4.3.4 Temporal Metadata Preservation

Each retrieved embedding in $\mathcal{R}_q^{(k)}$ retains its temporal and source metadata μ_{in}. This structure allows downstream modules to apply temporal constraints during the temporal-aware retrieval phase. The representation and indexing phase produces a searchable, semantically rich vector space, given by:

$$\mathcal{D}_{\text{index}} = \{(\hat{\mathbf{z}}_{in}, \mu_{in})\}_{i,n=1}^{N_i} \tag{19}$$

This vector space $\mathcal{D}_{\text{index}}$ serves as the knowledge base for efficient, timestamp-aware video question answering.

4.4 Temporal-Aware Retrieval and Reasoning

Given the indexed representation $\mathcal{D}_{\text{index}}$, the goal of this stage is to find video segments that are both relevant to the query q and correctly aligned in time. Unlike standard RAG systems that search over static text, this framework retrieves information directly from the video. It uses a temporally aware, intra-video retrieval process to ensure that all selected evidence comes from the correct parts of the same video. This design helps maintain factual accuracy and enables the system to produce answers with clear, timestamped evidence that is easy to interpret.

4.4.1 Query Embedding and Initial Retrieval

Each user question q is first encoded into a dense query vector, given by:

$$\hat{\mathbf{z}}_q = f_{\text{emb}}(q), \qquad \|\hat{\mathbf{z}}_q\|_2 = 1 \tag{20}$$

using the same embedding function $f_{\text{emb}}(\cdot)$ defined in Section 4.3. An initial candidate set is obtained from the FAISS index by retrieving the top-K semantically nearest neighbors:

$$\mathcal{R}_q^{(K)} = \text{TopK}_{(\hat{\mathbf{z}}_{in})}\left[\text{sim}(\hat{\mathbf{z}}_q, \hat{\mathbf{z}}_{in})\right], \quad \text{sim}(\hat{\mathbf{z}}_q, \hat{\mathbf{z}}_{in}) = \hat{\mathbf{z}}_q^{\top}\hat{\mathbf{z}}_{in} \tag{21}$$

The resulting retrieval set $\mathcal{R}_q^{(K)} = \{(i_k, n_k, \text{sim}_{i_k n_k})\}_{k=1}^{K}$ contains both the similarity scores and corresponding temporal metadata $\mu_{i_k n_k}$.

4.4.2 Temporal-Aware Filtering

To exploit temporal structure and maintain coherence, a timestamp-based filtering function $f_{\text{temp}}(\cdot)$ is applied to $\mathcal{R}_q^{(K)}$. The parameter Δt represents a temporal tolerance window that expands the query-specific interval to account for minor misalignment between annotated timestamps and retrieved segments.

$$\mathcal{R}_q^{\text{temp}} = \left\{(i_k, n_k) \in \mathcal{R}_q^{(K)} \,\middle|\, i_k = j,\ [t_{i_k n_k}^{\text{start}}, t_{i_k n_k}^{\text{end}}] \cap [t_q^{\text{start}} - \Delta t,\ t_q^{\text{end}} + \Delta t] \neq \varnothing\right\} \tag{22}$$

The system then applies local temporal smoothing to group adjacent chunks belonging to the same video:

$$\mathcal{R}_q^{\text{smooth}} = \text{merge}\left(\mathcal{R}_q^{(K)};\ \delta_t\right) \tag{23}$$

where $\text{merge}(\cdot)$ is used to aggregate consecutive results whose time gaps are below δ_t. This temporal-aware filtering ensures that retrieved evidence is not only semantically relevant but also temporally contiguous and contextually stable.

4.4.3 Cross-Encoder Re-Ranking

We refine the top-M candidates ($M < K$) of the retrieval using a cross-encoder scoring model $f_{\text{ce}}(\cdot,\cdot)$. This cross-encoder scoring model evaluates the pairwise relevance between the query q and each candidate chunk c_{in}:

$$r_{in} = f_{\text{ce}}(q, c_{in}) \tag{24}$$

where r_{in} is a scalar relevance score in $[0,1]$. The final ranked set is:

$$\mathcal{R}_q^* = \text{sort}_{r_{in}}\left(\{(i,n,r_{in}) \mid (i,n) \in \mathcal{R}_q^{\text{temp}}\}\right) \tag{25}$$

This set is ordered by descending r_{in}. f_{ce} is implemented using a transformer-based cross-encoder fine-tuned on QA relevance pairs. This re-ranking step yields high-precision evidence selection, improving over embedding-only retrieval.

4.4.4 Context Assembly and Reasoning

From the top-L elements of $\mathcal{R}_q^*$, a compact context window $\mathcal{E}_q$ is constructed:

$$\mathcal{E}_q = \left\{(\tilde{c}_\ell, \tilde{t}_\ell^{\text{start}}, \tilde{t}_\ell^{\text{end}})\right\}_{\ell=1}^{L}, \quad (\tilde{c}_\ell, \tilde{t}_\ell^{\text{start}}, \tilde{t}_\ell^{\text{end}}) \in \mathcal{R}_q^* \tag{26}$$

The corresponding text snippets are concatenated into a temporally ordered context:

$$C_q = \bigoplus_{\ell=1}^{L}\left[[\tilde{t}_\ell^{\text{start}}, \tilde{t}_\ell^{\text{end}}] : \tilde{c}_\ell\right] \tag{27}$$

An instruction-tuned LLM $f_{\text{LLM}}(\cdot,\cdot)$ then generates a grounded natural-language answer:

$$a_q = f_{\text{LLM}}(q, C_q) \tag{28}$$

subject to the grounding constraint that all information used in a_q must originate from C_q.

This constraint instructs the LLM to act primarily as a reasoning engine by restricting answer generation to the retrieved video-derived context.

Fig. 2 **summarizes the structured input formulation used in the proposed framework.**

5 Experiments and Results

This section presents the experimental evaluation of the proposed framework on the LectQA-Vid dataset. We first describe the evaluation setup, implementation details, and assessment metrics. We then report both quantitative and qualitative results. Finally, we discuss performance over different difficulty levels and the approach's strengths and limitations.

All experiments were conducted using **Google Colab Pro+**, configured with an NVIDIA Tesla T4 GPU (16 GB VRAM), 32 GB RAM, and 4 vCPUs. The software environment included Python 3.10, PyTorch 2.2.0, HuggingFace Transformers 4.39.3, FAISS-gpu 1.7.4, and SentenceTransformers 2.6.0. All runs were made deterministic by setting the random seed to 42 and enabling deterministic CUDA kernels.

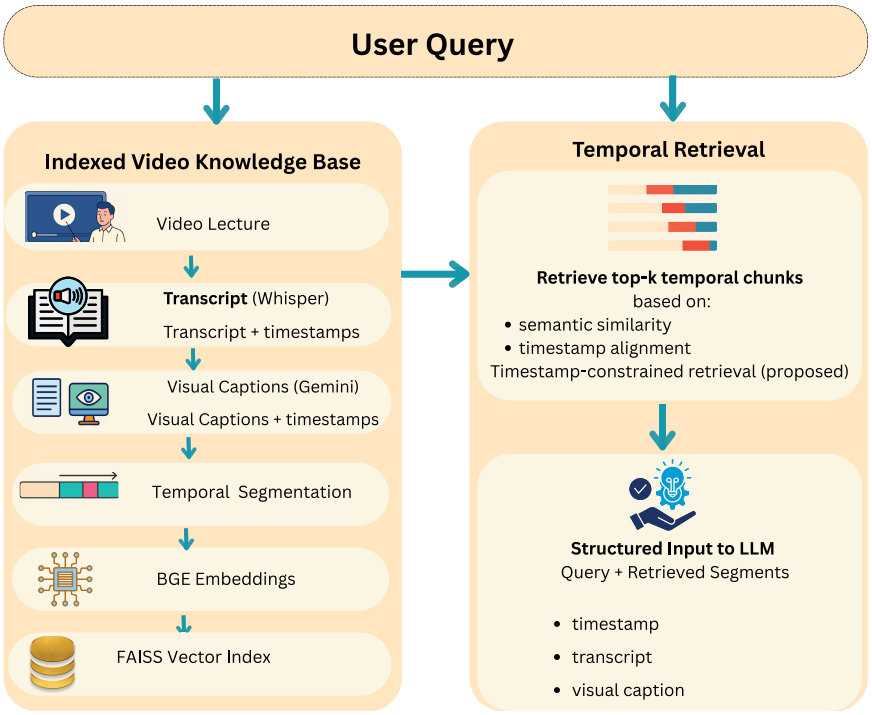


Figure 2: Preparation of structured, temporally grounded input for the proposed VideoQA framework.

The experiments were performed using the LectQA-Vid dataset (Section 3). The dataset is partitioned at the video level into 80% training, 10% validation, and 10% test splits. For computational efficiency, evaluation is conducted on a stratified subset of 1000 question–answer pairs sampled from the test videos (500 MCQs and 500 open-ended questions).

5.1 Threshold Selection

The proposed framework uses several thresholds to control semantic chunking, temporal alignment, and retrieval relevance. These thresholds include the semantic similarity threshold θ_{sem}, temporal overlap threshold θ_t, temporal gap threshold δ_{gap}, duplicate similarity threshold δ_{dup}, retrieval similarity threshold θ_r, and the temporal tolerance parameter Δt used in timestamp-based filtering, along with the retrieval parameter k. All similarity-based thresholds are derived from cosine similarity. Temporal thresholds are defined in seconds.

We used validation data to find the right values for these parameters that balance semantic coherence, temporal consistency, and retrieval precision. Table 3 contains the selected thresholds.

Table 3: Thresholds used in the proposed framework.

Parameter	Symbol	Value	Range
Semantic similarity threshold	θ_{sem}	0.75	$[0, 1]$
Temporal overlap threshold	θ_t	2 s	$[0, \infty)$
Temporal gap threshold	δ_{gap}	5 s	$[0, \infty)$
Duplicate similarity threshold	δ_{dup}	0.90	$[0, 1]$

(Continued)

Table 3 (continued)

Parameter	Symbol	Value	Range
Retrieval similarity threshold	θ_r	0.60	[0, 1]
Temporal tolerance window	Δt	3 s	$[0, \infty)$
Top-k retrieval	k	10	$\mathbb{Z}^+$

Validation across different threshold values shows that the system remains stable near the selected configuration. Lowering θ_{sem} yields overly long representations, whereas increasing it yields fragmented outputs and reduced contextual continuity. Increasing θ_r improves precision but reduces recall by admitting less relevant evidence into the retrieved context.

Temporal thresholds (θ_t, δ_{gap}, and Δt) balance alignment and coverage. The tolerance parameter Δt provides flexibility when matching retrieved segments to the query's temporal window. Smaller Δt values enforce tighter alignment but may miss relevant evidence, while larger values increase recall but can include temporally distant segments. Similarly, very small θ_t and δ_{gap} values may exclude useful multimodal information, while larger values can introduce temporal drift.

5.2 Evaluation Metrics

This section provides details of the evaluation metrics used to assess the proposed framework. We assume that y denotes the ground-truth answer and $\hat{y}$ denotes the predicted answer.

5.2.1 Metrics for Open-Ended Questions

For Open-ended questions, we used standard text-based and semantic evaluation metrics. Token-level precision, recall, and F1 are computed as:

$$\text{Precision} = \frac{|T_y \cap T_{\hat{y}}|}{|T_{\hat{y}}|}, \qquad \text{Recall} = \frac{|T_y \cap T_{\hat{y}}|}{|T_y|} \tag{29}$$

$$\text{F1} = \frac{2PR}{P + R} \tag{30}$$

where T_y and $T_{\hat{y}}$ denote the sets of tokens in the ground-truth and generated answers, respectively.

The next used metric is BLEU score which measures n-gram overlap between y and $\hat{y}$ and is defined as:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{4} w_n \log p_n\right), \quad w_n = \frac{1}{4} \tag{31}$$

where BP is the brevity penalty and p_n denotes modified n-gram precision.

The metric METEOR emphasizes recall and incorporates a penalty for fragmented matches:

$$\text{METEOR} = F_\alpha(1 - P_{\text{pen}}), \qquad F_\alpha = \frac{10PR}{R + 9P} \tag{32}$$

where P and R denote precision and recall, respectively. The penalty term P_{pen} accounts for fragmented alignments and is defined as:

$$P_{\text{pen}} = \gamma\left(\frac{ch}{m}\right)^{\beta} \tag{33}$$

where m is the number of matched unigrams, ch is the number of contiguous chunks in the alignment, and γ and β are tunable parameters ($\gamma = 0.5$, $\beta = 3$).

Next, ROUGE-1 is used, which measures unigram recall:

$$\text{ROUGE-1} = \frac{|T_y \cap T_{\hat{y}}|}{|T_y|} \tag{34}$$

where T_y and $T_{\hat{y}}$ denote the reference and generated unigram sets, respectively.

To account for paraphrasing and semantic equivalence beyond surface-form overlap, we compute cosine similarity between sentence embeddings:

$$\text{Sim} = \frac{\mathbf{e}_y \cdot \mathbf{e}_{\hat{y}}}{\|\mathbf{e}_y\|_2 \|\mathbf{e}_{\hat{y}}\|_2} \tag{35}$$

where embeddings $\mathbf{e}_y$ and $\mathbf{e}_{\hat{y}}$ are generated using all-MiniLM-L6-v2.

5.2.2 Metrics for Multiple-Choice Questions

Multiple-choice questions (MCQs) are treated as a *classification task*, since each question contains four answer options with exactly one correct choice. Accordingly, we report objective classification metrics. The accuracy measures the proportion of correctly selected options as:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \tag{36}$$

For MCQs, precision, recall, and F1 are computed over the predicted and ground-truth option labels, providing a balanced view of classification performance, particularly under varying difficulty levels.

In summary, text-generation and semantic metrics (F1, BLEU, METEOR, ROUGE, and semantic similarity) are used exclusively for *open-ended* questions. In contrast, classification metrics (Accuracy, Precision, Recall, and F1) are used exclusively for *multiple-choice* questions.

5.3 Experimental Results

This section presents a quantitative evaluation of the proposed framework on both open-ended and multiple-choice (MCQ) questions. We compare the proposed Temporal-Aware RAG framework with LLaVA-1.6, a strong open-source multimodal foundation model. This comparison assesses whether a large vision-language model, when provided with both visual and textual context, can implicitly perform the temporal localization and evidence grounding required for lecture-style VideoQA without explicit retrieval or alignment mechanisms.

LLaVA-1.6 combines a CLIP-based visual encoder with a large language model trained on a broad multimodal instruction-following corpus. The model has demonstrated strong performance on general image reasoning, visual question answering, and OCR-related tasks. As such, it serves as a representative and competitive baseline for evaluating whether general-purpose multimodal reasoning alone is sufficient for answering questions grounded in long-form instructional videos. However, LLaVA-1.6 does not incorporate explicit mechanisms for temporal modeling, retrieval, or evidence re-ranking across extended video content.

In our evaluation setting, LLaVA-1.6 is provided with 4–6 uniformly sampled keyframes from each lecture video together with the full Whisper ASR transcript. The model must answer questions using only this information. The following prompt is used:

Prompt: You are given (i) a set of keyframes extracted from a lecture video and (ii) the full ASR transcript of the same video. Answer the question using *only* the provided keyframes and transcript. Do not use any external knowledge. If the evidence is insufficient, respond with "Insufficient evidence from the provided transcript/keyframes."

Question: {QUESTION}

Transcript: {WHISPER_TRANSCRIPT}

Keyframes: (images provided above)

Output format: Provide a short answer only. For MCQs, output exactly one letter from {A,B,C,D}.

For multiple-choice questions, the four answer options (A–D) are appended to the question text. For open-ended questions, the model is instructed to generate concise answers grounded strictly in the provided transcript and keyframes.

5.3.1 Open-Ended Question Performance

Table 4 shows the results of open-ended QA for LLaVA-1.6 and the proposed framework. It can be seen that the performance of both systems decreases as the difficulty level of the questions increases. However, the proposed framework consistently outperforms LLaVA-1.6 on all metrics and difficulty levels. The proposed Temporal-Aware RAG improves F1 from 15.30% to 23.52% and ROUGE-1 from 21.70% to 29.76%, indicating more faithful overlap with the reference answers and better coverage of the key facts required by the questions.

Table 4: Open-ended QA performance comparison (in%).

Model	Difficulty	F1	Sim	BLEU	METEOR	R1
LLaVA-1.6 (Keyframes + Transcript)	Simple	21.29	70.34	5.18	30.73	28.41
	Hard	15.62	66.11	3.49	24.28	21.95
	Very Hard	10.47	62.85	2.67	18.39	16.12
	Overall	15.30	66.88	3.30	24.57	21.70
Temporal-Aware RAG (Ours)	Simple	29.47	77.23	8.61	38.15	36.82
	Hard	24.38	74.56	5.29	34.71	30.94
	Very Hard	16.72	71.48	2.83	24.36	22.51
	Overall	23.52	74.42	5.58	32.41	29.76

LLaVA-1.6 exhibits moderate semantic similarity and a relatively low F1 and ROUGE-1. This suggests that LLaVA-1.6 can often infer the question's general topic and produce broadly plausible answers. However, it frequently fails to respond to the specific lecture segment where the correct information is stated. This behavior is consistent with the absence of an explicit evidence-selection mechanism. The model implicitly searches over long transcripts and sparsely sampled visual context, which increases the likelihood of generating generic, underspecified, or partially correct answers.

In contrast, the proposed framework demonstrates higher similarity and significantly improved lexical and coverage-based scores. These results suggest that temporally filtered retrieval and re-ranking not only identify conceptually relevant evidence but also enable the generator to produce answers that more closely align with the reference content.

5.3.2 Multiple-Choice Question Performance

Table 5 reports MCQ performance. Since each MCQ contains four answer options with exactly one correct choice, we treat MCQs as a classification task and report Accuracy (ACC), Precision, Recall, and F1-score.

Table 5: MCQ performance comparison (in%).

Model	Difficulty	ACC	Precision	Recall	F1
LLaVA-1.6 (Keyframes + Transcript)	Simple	45.37	46.58	46.21	46.84
	Hard	40.62	41.13	41.79	41.45
	Very Hard	35.48	36.92	36.27	36.64
	Overall	40.00	41.56	41.34	41.89
Temporal-Aware RAG (Ours)	Simple	57.29	62.15	62.08	62.11
	Hard	52.34	53.18	53.25	53.21
	Very Hard	50.67	51.42	51.39	51.40
	Overall	53.43	55.58	55.57	55.57

For MCQs, the proposed Temporal-Aware RAG framework achieves consistently higher performance. The accuracy, precision, recall, and F1 values were higher for our method than for LLaVA-1.6. The performance of LLaVA-1.6 decreases steadily from "Simple" to "Very Hard" questions. This shows that it struggles when questions require careful distinction between similar options or when information must be combined from different parts of the lecture. This happens because LLaVA-1.6 uses only a few keyframes and a full, unstructured transcript. As a result, it is easily affected by irrelevant information and by details that appear far apart in time.

On the other hand, the proposed framework shows a gradual decline as difficulty rises, but it still performs better than LLaVA-1.6. These results show that explicit retrieval, temporal filtering, and re-ranking make systems more robust, even when questions require detailed reasoning or distributed evidence.

Overall, the results across both open-ended and MCQ settings demonstrate that explicit, timestamp-aware evidence selection is essential for lecture-style VideoQA. While LLaVA-1.6 provides a strong general-purpose multimodal baseline, the proposed Temporal-Aware RAG framework yields more accurate and reliable answers by combining multimodal cues with structured retrieval, temporal constraints, and re-ranking.

5.3.3 Comparison with Standard RAG Baselines

We next compare the proposed Temporal-Aware RAG framework against standard retrieval-based baselines that progressively introduce multimodal evidence while explicitly excluding temporal reasoning. This controlled comparison isolates the contribution of temporal grounding beyond retrieval and multimodality alone.

Text-only RAG (no temporal awareness): A FAISS index is constructed over semantically chunked Whisper ASR transcript segments, and the top-k chunks are retrieved using embedding similarity. The generator produces answers using only the retrieved transcript evidence, without visual captions or timestamp constraints.

Multimodal RAG (no timestamps): ASR transcript chunks and Gemini-generated visual caption chunks are jointly indexed in a single retrieval space. Top-k chunks are retrieved by similarity without enforcing any timestamp constraints, allowing retrieved evidence to be semantically relevant but temporally misaligned.

Table 6 reports results for open-ended question answering. The results show that retrieval based on ASR transcripts alone is not robust as question difficulty increases. While "Simple" questions achieve 24.37% F1, performance drops sharply for "Hard" (9.01%) and "Very Hard" (3.89%) questions, with a corresponding decline in semantic similarity. This indicates that similarity-based retrieval over transcripts struggles to retrieve precise, question-specific evidence for complex queries.

Table 6: Standard RAG baselines for open-ended QA without temporal awareness (in%).

Difficulty	F1	Semantic Similarity	BLEU	METEOR
Text-only RAG (ASR only)				
Simple	24.37	60.06	8.05	41.05
Hard	9.01	50.59	1.23	18.18
Very Hard	3.89	34.06	1.58	10.53
Overall	12.42	48.24	3.62	23.25
Multimodal RAG (ASR + captions, no timestamps)				
Simple	27.90	71.00	6.12	38.89
Hard	11.30	62.00	1.21	16.34
Very Hard	5.30	52.00	1.51	9.73
Overall	14.80	61.00	2.95	21.65

Incorporating visual captions substantially improves semantic alignment across all difficulty levels, increasing overall similarity from 48.24% to 61.00%. This highlights the complementary value of slide content and on-screen information. However, without timestamp constraints, multimodal RAG remains vulnerable to temporal drift and can retrieve conceptually related but contextually incorrect segments.

5.3.4 Ablation Study

To assess the contribution of each individual component in the proposed framework, an ablation study was conducted on open-ended questions. The ablation study evaluates four model variants, each created by removing one key component: visual captions, temporal filtering, the cross-encoder re-ranker, or normalization.

Table 7 presents the ablation results. It can be seen that removing the temporal filter from the framework had the greatest impact on performance, as the overall F1 score drops to 11.40 and the semantic similarity to 36.67. This performance decline is especially more evident for "Very Hard" questions. The decline in performance is due to the retriever becoming less effective at localizing relevant segments without temporal constraints.

The ablation study after removing the cross-encoder re-ranker shows that the performance in the simple category remains relatively stable (F1 = 31.15). However, for the "Hard" and "Very Hard" questions, a relatively sharp decline is observed. This indicates that, without re-ranking, the framework struggles with hard questions due to its inability to prioritize the most relevant evidence.

Table 7: Ablation study for open-ended QA (in%). Each block removes one component from the full temporal-aware RAG pipeline.

Difficulty	F1	Semantic Similarity	BLEU	METEOR
No Visual Captions				
Simple	26.23	63.56	7.50	35.62
Hard	14.50	55.21	3.50	24.12
Very Hard	7.80	42.13	1.20	12.32
Overall	16.40	55.80	4.10	24.00
No Temporal Filter				
Simple	21.80	44.23	9.05	31.05
Hard	8.70	37.57	2.23	17.18
Very Hard	3.80	28.23	2.58	9.53
Overall	11.40	36.67	4.52	22.85
No Cross-Encoder Re-Rank				
Simple	31.15	60.31	8.19	33.57
Hard	24.07	48.72	4.79	26.19
Very Hard	11.30	42.48	0.56	7.31
Overall	22.21	52.58	4.53	26.20
No Normalization				
Simple	30.15	58.31	8.19	33.57
Hard	23.07	47.72	4.79	26.19
Very Hard	10.30	28.48	0.56	7.31
Overall	20.21	44.83	4.53	26.20

When visual captions were removed, a moderate decline in performance was observed as the overall F1 score decreased to 16.40, and semantic similarity fell to 55.80. Again, the decline is more evident in the more difficult categories. This indicates that visual captions are an important clue to answer complex questions.

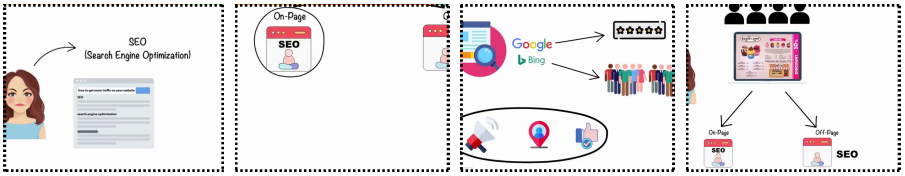
Finally, the ablation study after removing the normalization component also resulted in a moderate performance decrease. The "Very Hard" category shows a more substantial decrease in semantic similarity (to 28.48). These results suggest that normalization supports consistency in more challenging scenarios.

5.4 Qualitative Analysis of Success and Failure Cases

This section shows examples of both failure and success from the LectQA-Vid dataset. Fig. 3 shows a case of failure with a multiple choice question about when the narrator talks about off-page SEO. The ground truth shows a section where the speaker discusses topics such as backlinks, guest blogging, and influencer marketing. The model, on the other hand, retrieves content from an earlier section on on-page SEO, indicating that the system chooses an option that is relevant to the topic but not to the time.

Fig. 4 shows an example of correct detection. The question asks about the hierarchical structure of Internet Service Providers and their roles in connectivity. The corresponding video segment contains clear slide diagrams and explicit verbal references to each tier, which resulted in highly distinctive lexical cues

(e.g., “Tier 1 backbone”, “Tier 2 regional providers”, “Tier 3 access networks”). In this case, both embedding-based retrieval and cross-encoder re-ranking were able to discriminate the relevant chunk from the rest of the lecture.



Q: When does the narrator explain the benefits of off-page SEO, and what activities are mentioned?

Options:
A: Between 154 and 175 seconds, mentioning backlinks, guest blogging, influencer marketing, and social media promotion.
B: At the start during on-page SEO.
C: During the meta tag optimization.
D: In the certification section.

GT: Between 154 and 175 seconds, mentioning backlinks, guest blogging, influencer marketing, and social media promotion.

Pred: At the start during on-page SEO.

Figure 3: An example MCQ where the proposed method failed.

Q: How does the explanation of Tier 1, 2, and 3 ISPs illustrate the hierarchical structure of internet distribution?

Options
A: By showing how Tier 1 handles global connectivity, Tier 2 covers regional distribution, and Tier 3 provides local access.
B: By describing only local providers.
C: By focusing solely on bandwidth metrics.
D: By repeating examples of dial-up connections.

GT: By showing how Tier 1 handles global connectivity, Tier 2 covers regional distribution, and Tier 3 provides local access.

Pred: By showing how Tier 1 handles global connectivity, Tier 2 covers regional distribution, and Tier 3 provides local access.

Figure 4: An example MCQ where the proposed method gave the correct result.

The example in Fig. 5 shows a failure case of an open-ended question. The question asks which algorithms are discussed in the lecture, with the correct answer being linear and logistic regression, decision trees, and convolutional neural networks. Instead, the model outputs an answer related to Prim’s algorithm. This lecture introduces different algorithmic examples in different segments. This error was because of the incomplete coverage of the slide text by visual captioning. The retrieval step picks up only a limited number

of top-ranked chunks, and thus those chunks may emphasize “algorithmic thinking” in general and thus the LLM generates an answer that is thematically compatible yet factually incorrect.

Q: What examples of algorithms does the video mention, and what do they do?"

GT: The video mentions linear and logistic regression, decision trees, and convolutional neural networks, which assign weights to features to make predictions.

Pred: "The video mentions Prim's algorithm as an example. Prim's algorithm is used to find a minimum spanning tree in a graph, which is a minimum weight connected graph with no cycles. The algorithm involves selecting the smallest edge that connects to an unvisited node, progressively adding nodes to the minimum spanning tree until all nodes are connected."

Metrics: F1: 13%, Sim: 51%, BLEU: 3%, METEOR: 24, ROUGE-1: 20%

Figure 5: An example open-ended question where the proposed method failed.

Lastly, Fig. 6 shows an open-ended question which was answered correctly. The question asks when the instructor highlights a particular selling point of the program. The correct answer requires identifying a segment where small class sizes and close student–faculty relationships are discussed. In this case, the ASR transcript contains unambiguous phrases (e.g., “small class sizes”, “personalized attention”) that appear within a narrow time window, and the corresponding frames show slides that emphasize the same message. Temporal-aware retrieval, therefore, focuses on the initial seconds of the lecture, and the cross-encoder re-ranker promotes the segment that contains the correct emphasis.

Q: "At what point does the lecturer highlight the uniqueness of St. Mary’s computer science program, and what is the first distinctive factor mentioned?

GT: Between 0 and 8 seconds, the lecturer describes St. Mary’s computer science program as unique, beginning with its small class sizes and close student relationships.

Pred: The lecturer highlights the uniqueness of St. Mary’s computer science program at the beginning of the context. The first distinctive factor mentioned is the small class size, which allows students to get to know their fellow computer science students well.

Metrics : F1: 31%, Sim: 90%, BLEU: 16%, METEOR: 47%, ROUGE-1: 42%

Figure 6: An example open-ended question where the proposed method succeeded.

These visual examples support our quantitative results. The model works well when key concepts are clear, occur within a short time window, and are consistently captured in both ASR and visual captions. In such cases, the temporal-aware retriever and re-ranker can select a small set of highly relevant chunks. Failures usually occur due to temporal drift between similar segments when the question requires information from multiple distant parts of the video.

5.5 Efficiency Analysis

Next, we evaluated the computational efficiency of the proposed framework. We measure index construction time, retrieval latency, temporal filtering cost, re-ranking time, and answer generation time under varying video lengths and retrieval scales.

The results in Table 8 show that retrieval latency increases gradually with the number of indexed chunks. Temporal filtering incurs negligible overhead, and the cost of re-ranking increases with larger top-k values because more candidate segments are evaluated. Across all settings, answer generation is the dominant component of the total response time. Importantly, total query time remains stable (around 2 s) even as video length and retrieval scale increase.

Table 8: Efficiency analysis under varying video lengths and retrieval scales.

Setting	Video Length	Chunks	Index (s)	Retr. (ms)	Temp. (ms)	Re-Rank (ms)	Gen. (s)	Total (s)
Short (small)	2 min	36	0.3	11	2	34	1.86	1.91
Medium (medium)	3 min	54	0.5	14	3	47	2.01	2.07
Long (large)	4 min	72	0.7	17	4	63	2.18	2.26
Medium, $k = 20$	3 min	54	0.5	20	4	86	2.09	2.20
Medium, $k = 30$	3 min	54	0.5	26	5	121	2.19	2.34

5.6 Discussions

Experimental results demonstrate that a range of answers related to lecture videos can be answered by the proposed framework. Lecture videos are characterized by a high degree of structure, conceptual density, and sequential organization. Therefore, accurate answers frequently require grounding evidence in the specific temporal segment in which a concept is introduced or further developed. The proposed framework uses timestamp-based filtering to reduce temporal drift by restricting retrieval to the relevant time window, thereby enhancing semantic faithfulness.

The experiments also show that lecture VideoQA is not purely a speech-understanding problem. The inclusion of Gemini-generated visual captions improves semantic alignment, indicating that on-screen slide content provides complementary evidence that is usually essential for reliable question answering. The ablation study further confirms this dependency: removing visual captions reduces robustness as question difficulty increases.

There are, however, certain limitations of the proposed method:

- The evaluation is conducted in a micro-lecture setting in which the videos are 2–5 min long. Thus, the proposed method has not been tested on long lecture videos where long-range dependencies and increased topic drift are likely to make the task of video question answering more challenging.

- The proposed pipeline relies on Whisper ASR transcripts and Gemini-generated visual captions. Any errors in transcription and captioning (e.g., due to pronunciation, domain-specific terminology, or noise) can propagate to retrieval and temporal grounding.
- The proposed retrieval design uses fixed thresholds for granularity. The evidence in videos is extracted using fixed-size semantic chunks and by using a single-shot top-k strategy. Therefore, the retrieval phase may miss fine-grained evidence in the videos or fail to assemble the distributed evidence needed for multi-hop instructional questions.
- The proposed framework was designed to use only intra-video evidence. However, the framework lacks an explicit mechanism to detect hallucinations or verify factual consistency.
- The proposed framework deliberately focuses on intra-video reasoning only and does not incorporate external knowledge, so the answers to the questions do not include any additional explanations which may be helpful to some students.
- Although the proposed method has been compared against a strong multimodal model and controlled RAG baselines, we do not include recent VideoQA models designed for long-video understanding. Many of these approaches operate end-to-end, without explicit retrieval or timestamp-constrained evidence selection. In contrast, our work focuses on intra-video, retrieval-based reasoning with strict temporal grounding. Due to this difference in task formulation, direct comparison is not fully aligned. Extending evaluation to recent long-video VideoQA is an important direction for future work.

6 Conclusions

This work introduced a Temporal-Aware, Intra-video Retrieval-Augmented Generation (RAG) framework for Video Question Answering (VideoQA) over short lecture videos. Unlike conventional LLM-based or non-temporal RAG approaches, the proposed system retrieves multimodal, timestamped evidence from the target video alone before generating an answer. By integrating Whisper ASR, Gemini visual captioning, semantic chunking, BGE embeddings, FAISS retrieval, cross-encoder re-ranking, and temporal filtering, the framework ensures that all predictions are explicitly grounded in verifiable video segments. The experiments on the LectVidQA dataset show that the proposed framework outperforms text-only RAG and multimodal non-temporal baselines. The extensive ablation studies also confirm the importance of multimodal cues and temporal constraints. The framework is simple and efficient as it does not rely on external knowledge and works well across different lecture topics.

In future work, we aim to improve the alignment between video frames and text by developing stronger temporal models and expanding the system to support longer instructional videos. We also plan to add uncertainty estimation to make the system more explainable. Currently, audio is converted to text using ASR, which makes it easier to combine different types of data but omits important acoustic details such as pauses and emphasis. A valuable next step would be to use raw audio features along with transcripts and visual data.

Acknowledgement: Not Applicable.

Funding Statement: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia, under Grant KFU262069.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Sumaira Shafiq and Munam Ali Shah; methodology, Sumaira Shafiq; software, Sumaira Shafiq; validation, Sumaira Shafiq, Naveed Ejaz and Munam Ali Shah; formal analysis, Sumaira Shafiq; investigation, Sumaira Shafiq; resources, Naveed Ejaz, Rashid Kamal and Sheraz Aslam; data curation, Sumaira Shafiq and Adnan Sohail; writing—original draft preparation, Sumaira Shafiq; writing—review and editing, Naveed Ejaz, Munam Ali Shah, Rashid Kamal, Adnan Sohail and Sheraz

Aslam; visualization, Sumaira Shafiq; supervision, Munam Ali Shah; project administration, Munam Ali Shah; funding acquisition, Munam Ali Shah. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset developed in this study is publicly available at https://data.mendeley.com/datasets/yt4nmz9mcv/1.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASR	Automatic Speech Recognition
CMC	Computers, Materials & Continua
FAISS	Facebook AI Similarity Search
GPT	Generative Pre-trained Transformer
LLM	Large Language Model
MCQ	Multiple-Choice Question
QA	Question Answering
RAG	Retrieval-Augmented Generation
VLM	Vision-Language Model
VideoQA	Video Question Answering

References

1. Jeshmol PJ, Kovoor BC. Video question answering: a survey of the state-of-the-art. J Vis Commun Image Represent. 2024;105(3):104320. doi:10.1016/j.jvcir.2024.104320.
2. Ishmam MF, Shovon MSH, Mridha MF, Dey N. From image to language: a critical analysis of visual question answering (VQA) approaches, challenges, and opportunities. Inf Fusion. 2024;106(6):102270. doi:10.1016/j.inffus.2024.102270.
3. Islam MM, Nagarajan T, Wang H, Bertasius G, Torresani L. BIMBA: selective-scan compression for long-range video question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2025 Jun 10–17; Nashville, TN, USA. p. 29096–107. doi:10.1109/CVPR52734.2025.02709.
4. Alemdag E. A scoping review of the literature on embodied instructional videos. Res Pract Technol Enhanc Learn. 2023;18:29. doi:10.58459/rptel.2023.18029.
5. Oche AJ, Folashade AG, Ghosal T, Biswas A. A systematic review of key retrieval-augmented generation (RAG) systems: progress, gaps, and future directions. arXiv:2507.18910. 2025. doi:10.48550/arxiv.2507.18910.
6. Han B, Susnjak T, Mathrani A. Automating systematic literature reviews with retrieval-augmented generation: a comprehensive overview. Appl Sci. 2024;14(19):9103. doi:10.3390/app14199103.
7. Raja R, Vats A. Multimedia-aware question answering: a review of retrieval and cross-modal reasoning architectures. In: Proceedings of the 2nd ACM Workshop on AI-Powered Question & Answering Systems. New York, NY, USA: The Association for Computing Machinery (ACM); 2025. p. 28–35. doi:10.1145/3746274.3760393.
8. Sun G, Liang L, Li T, Yu B, Wu M, Zhang B. Video question answering: a survey of models and datasets. Mob Netw Appl. 2021;26(5):1904–37. doi:10.1007/s11036-020-01730-0.
9. Zhong Y, Ji W, Xiao J, Li Y, Deng W, Chua TS. Video question answering: datasets, algorithms and challenges. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics; 2022. p. 6439–55. doi:10.18653/v1/2022.emnlp-main.432.
10. Gao J, Ge R, Chen K, Nevatia R. Motion-appearance co-memory networks for video question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun 18–23; Salt Lake City, UT, USA. p. 6576–85. doi:10.1109/CVPR.2018.00688.

11. Zeng P, Zhang H, Gao L, Song J, Shen HT. Video question answering with prior knowledge and object-sensitive learning. IEEE Trans Image Process. 2022;31:5936–48. doi:10.1109/TIP.2022.3205212.
12. Park J, Lee J, Sohn K. Bridge to answer: structure-aware graph interaction network for video question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 15521–30. doi:10.1109/CVPR46437.2021.01527.
13. Liu F, Liu J, Wang W, Lu H. HAIR: hierarchical visual-semantic relational reasoning for video question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 1678–87. doi:10.1109/ICCV48922.2021.00172.
14. Cherian A, Hori C, Marks TK, Le Roux J. (2.5 + 1)D spatio-temporal scene graphs for video question answering. Proc AAAI Conf Artif Intell. 2022;36(1):444–53. doi:10.1609/aaai.v36i1.19922.
15. Lei J, Berg T, Bansal M. Revealing single frame bias for video-and-language learning. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL). Toronto, ON, Canada: Association for Computational Linguistics; 2023. p. 487–507. doi:10.18653/v1/2023.acl-long.29.
16. Yang Z, Li W, Cheng G. SHMamba: structured hyperbolic state space model for audio-visual question answering. IEEE Trans Audio Speech Lang Process. 2025;33:3582–93. doi:10.1109/TASLPRO.2025.3597461.
17. Zong L, Wan J, Zhang X, Liu X, Liang W, Xu B. Video-context aligned transformer for video question answering. Proc AAAI Conf Artif Intell. 2024;38(17):19795–803. doi:10.1609/aaai.v38i17.29954.
18. Yang Z, Garcia N, Chu C, Otani M, Nakashima Y, Takemura H. Bert representations for video question answering. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2020 Mar 1–5; Snowmass Village, CO, USA. p. 1556–65.
19. Yang Z, Garcia N, Chu C, Otani M, Nakashima Y, Takemura H. A comparative study of language transformers for video question answering. Neurocomputing. 2021;445:121–33. doi:10.1016/j.neucom.2021.02.092.
20. Garcia N, Otani M, Chu C, Nakashima Y. KnowIT VQA: answering knowledge-based questions about videos. Proc AAAI Conf Artif Intell. 2020;34:10826–34.
21. Wu T, Garcia N, Otani M, Chu C, Nakashima Y, Takemura H. Transferring domain-agnostic knowledge in video question answering. arXiv:211013395. 2021.
22. Liu M, Wang X, Nie L, Tian Q, Chen B, Chua TS. Cross-modal moment localization in videos. In: Proceedings of the 26th ACM International Conference on MuLtimedia. New York, NY, USA: The Association for Computing Machinery (ACM); 2018. p. 843–51. doi:10.1145/3240508.3240549.
23. Paul S, Mithun NC, Roy-Chowdhury AK. Text-based temporal localization of novel events. In: Proceedings of the European Conference on Computer Vision (ECCV). Berlin/Heidelberg, Germany: Springer; 2022. p. 567–87. doi:10.1007/978-3-031-19781-9_33.
24. Liu M, Nie L, Wang Y, Wang M, Rui Y. A survey on video moment localization. ACM Comput Surv. 2023;55(9):1–37. doi:10.1145/3556537.
25. Xiao J, Yao A, Li Y, Chua TS. Can I trust your answer? Visually grounded video question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 13204–14. doi:10.1109/CVPR52733.2024.01254.
26. Wu J, Liu W, Liu Y, Liu M, Nie L, Lin Z, et al. A survey on video temporal grounding with multimodal large language model. IEEE Trans Pattern Anal Mach Intell. 2026;48(2):1521–41. doi:10.1109/TPAMI.2025.3615586.
27. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv:2005.11401. 2020.
28. Zheng X, Weng Z, Lyu Y, Jiang L, Xue H, Ren B, et al. Retrieval augmented generation and understanding in vision: a survey and new outlook. arXiv:2503.18016. 2025.