VideoEvent: Leveraging Relevance and LLMs for Video Question Answering

Chen-Chen Lin, Ming-Han Lee, Kun-Ru Wu, Yu-Chee Tseng
Department of Computer Science
National Yang Ming Chiao Tung University
{cclin.cs12, mhlee.cs09, wufish, yctseng}@nycu.edu.tw

Abstract

We propose VideoEvent, a lightweight and efficient training-free framework for Video Question Answering (VQA) with large language models (LLMs). Although several training-free VQA methods have been proposed, they often neglect the temporal dependencies between frames or clips, treating them as isolated units and relying on complex or resource-intensive components. To address this limitation while maintaining performance and simplicity, we propose VideoEvent, a framework that segments an input video into question-relevant temporal events and selectively supplements them with low-level visual cues such as background and object layout. Our method selects semantically relevant time spans and retrieves one representative background frame to enrich the LLM prompt. This design minimizes reliance on additional tools and reduces inference cost, making it highly suitable for practical deployment. Experimental results on EgoSchema and NExT-QA show that VideoEvent reduces inference cost by up to 30% while maintaining state-of-the-art accuracy, and its background module improves accuracy by 1–3% across multiple frameworks.

Keywords: Large Language Model (LLM), Multimodal Model, Training-Free Model, Visual-Language Model (VLM), Video Question Answering

1. Introduction

Large language models (LLMs) have recently been applied in training-free settings, i.e., without pretraining or fine-tuning, across domains such as tool use (Lu et al., 2023; Shen et al., 2023), multi-agent systems (Chen et al., 2024; Hong et al., 2024), IoT reasoning (Xu et al., 2024; An et al., 2024), and table QA (Ye et al., 2023). These works highlight the potential of LLMs as general-purpose reasoning engines, inspiring training-free frameworks for downstream tasks.

In video understanding, Video Question Answering (VideoQA) is particularly challenging, as it requires reasoning over long video content and answering questions in natural language. Videos may last several minutes or longer, and questions can range from factual (e.g., recognizing an object or action) to abstract (e.g., inferring intent). Solving such tasks demands not only accurate perception of individual frames but also temporal integration and semantic grounding across the sequence.

Recent training-free approaches such as LLoVi (Zhang et al., 2024) show that converting video content into captions with pretrained visual-language models and leveraging LLM reasoning yields strong results, sometimes surpassing supervised methods. Subsequent works (Wang et al., 2024, 2025; Kahatapitiya et al., 2025) refine frame selection or integrate additional pretrained modules.

Despite these advances, training-free VideoQA methods still face two limitations: (i) neglect of temporal continuity and semantic coherence across frames, treating captions in isolation rather than as contextually related events; and (ii) omission of critical background details, since captioners emphasize foreground objects and actions while discarding contextual cues (e.g., scene layout, supporting objects) that may be decisive for reasoning. As a result, current pipelines pass redundant or incomplete information into the LLM, increasing inference cost and sometimes misleading reasoning.

To address these challenges, we present **VideoEvent**, a training-free framework based on semantics-guided event segmentation and background extraction. VideoEvent dynamically selects question-relevant, temporally coherent spans and enriches them with contextual background often missing from captions. The background extraction module provides this context at minimal cost, reducing redundancy before LLM reasoning. On EgoSchema (Mangalam et al., 2023), VideoEvent achieves accuracy comparable to state-of-the-art methods while reducing prompt-level inference cost by up to 30%.

Our main contributions are as follows:

- We propose a semantics-guided event segmentation method that captures temporal coherence and semantic relevance, enabling efficient long-video reasoning.
- A plug-and-play background extraction module is designed to enrich captions with contextual visual details, improving reasoning without any additional training.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 5024–5034
11–16 May 2026.

- The proposed background module generalizes effectively across different frameworks, as integrating it into LLoVi, VideoAgent, and VideoTree consistently boosts accuracy across datasets.

- Our approach improves inference efficiency by decoupling caption filtering from reasoning, reducing input size (e.g., from 56 to 9.2 captions on NExT-QA (Xiao et al., 2021)) and cutting cost by up to 30% on EgoSchema.

2. Related Work

2.1. Training-Free LLM-based VQA

LLoVi (Zhang et al., 2024) is one of the earliest training-free VideoQA frameworks. It first divides a video into short clips (0.5–8s), generates captions with a pretrained visual-language model (e.g., BLIP2 (Li et al., 2023), LaViLa (Zhao et al., 2023), LLaVA (Liu et al., 2023)), and then concatenates these captions as input to an LLM (e.g., GPT-3.5/4) for reasoning. This simple pipeline laid the foundation for later works that refine segment selection or incorporate additional modules.

2.2. Modular/Tool-based Methods

Another line of work treats LLMs as controllers that call external tools to complete sub-tasks. MoReVQA (Min et al., 2024) decomposes the task into event parsing, grounding, and reasoning, with API calls and an external memory for intermediate results. DoraemonGPT (Yang et al., 2024) builds symbolic memories for spatial and temporal information, invoking tools (e.g., object detection, OCR, ASR) and planning with MCTS. VideoMindPalace (Huang et al., 2025) organizes video content into a structured semantic graph derived from multiple perception models, enabling more complex long-video reasoning.

2.3. Frame-Selection-Based Methods

Frame-selection approaches aim to identify the most relevant frames without external tools, making them the main comparison targets of our study. LLoVi (Zhang et al., 2024) adopts uniform sampling, which is efficient but ignores semantic differences. VideoAgent (Wang et al., 2024) improves selection via multi-round retrieval with LLMs and CLIP (Sun et al., 2024), but remains shallow in temporal modeling. VideoTree (Wang et al., 2025) builds a hierarchical tree of frames using CLIP embeddings and selects keyframes at multiple levels, reducing redundancy but still lacking explicit modeling of temporally coherent events.

In summary, existing training-free approaches either rely on modular tool integration, which increases complexity and inference cost, or on frame selection strategies that neglect temporal coherence and contextual details. Our work addresses these issues with a semantics-guided event segmentation framework and lightweight background extraction, enabling more efficient, context-aware reasoning for long-video QA.

3. Proposed Method

VideoEvent is a training-free method designed for VQA. Its objective is to enable an LLM to understand long-form video content and answer natural language questions without additional model training or fine-tuning. Formally, given a video $\mathcal{V}$, a question $\mathcal{Q}$, and a candidate answer set $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, the task is to select the most likely answer from $\mathcal{A}$ with respect to the visual semantics of $\mathcal{V}$.

Figure 1 illustrates the VideoEvent pipeline. Our method dynamically segments video events based on question relevance and incorporates a plug-and-play background extraction module, ensuring that only informative content is passed to the reasoning stage. This design reduces computational cost while preserving accuracy, addressing both inefficiency and interpretability limitations of prior training-free frameworks. VideoEvent operates in three stages: (i) video caption generation, where clips are transcribed into textual descriptions; (ii) relevance-based event segmentation with optional background extraction; and (iii) answer reasoning using the selected captions and background context.

Compared to prior methods, such as LLoVi's (Zhang et al., 2024) uniform frame sampling and VideoAgent's (Wang et al., 2024) query-based caption retrieval, VideoEvent takes a semantics-driven approach by combining event segmentation and background augmentation. This not only enhances semantic understanding, but also effectively reduces the interference cost.

3.1. Stage 1: Video Caption Generation

The input video $\mathcal{V}$ is segmented into K fixed-length clips $\{v_1, v_2, \ldots, v_K\}$. Each clip $v_i, i = 1..K$, may be directly cropped from $\mathcal{V}$ or further down-sampled, if needed, at a fixed rate. Each v_i is then processed by a pretrained VLM to generate a caption c_i, serving as a textual representation and also as the basis for semantic event segmentation.

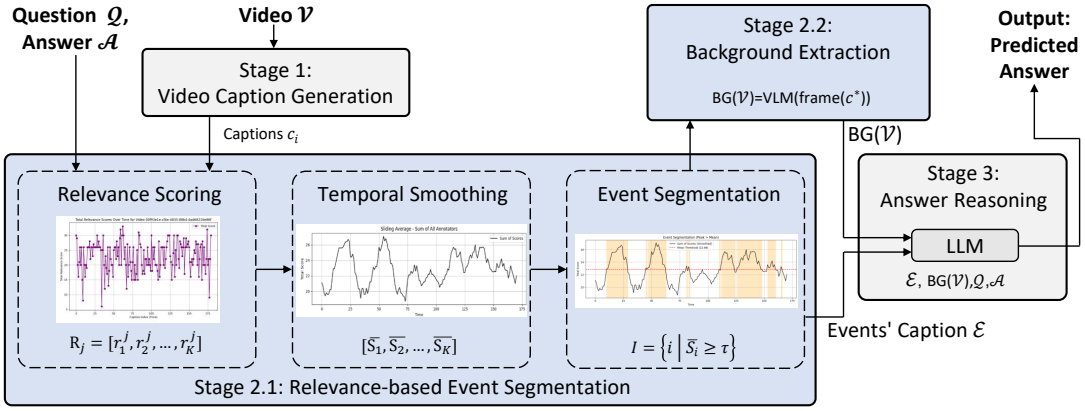

Figure 1: Overview of the proposed VideoEvent framework. It generates captions (Stage 1), performs relevance-based event segmentation and background extraction (Stage 2), and conducts answer reasoning with an LLM to produce the final answer (Stage 3).

3.2. Stage 2.1: Relevance-based Event Segmentation

This stage highlights the key drawback of the approaches of (Zhang et al., 2024; Wang et al., 2024), which often lack semantic structure and temporal coherence across clips. In contrast, our approach is both semantics- and answer-driven: it dynamically segments the video into highly relevant event intervals based on the relevance among $\mathcal{Q}$, $\mathcal{A}$, and c_is. This strategy generates segments that are better aligned with the QA task, facilitating context-aware reasoning while effectively suppressing irrelevant information.

3.2.1. Relevance Scoring

For each caption $c_i, i = 1..K$, and answer $a_j, j = 1..n$, we construct a prompt with c_i, a_j, and $\mathcal{Q}$, and use a lightweight LLM (e.g., GPT-4o-mini) to score their semantic relevance on a scale of 1–10, denoted as r_i^j. This produces a relevance sequence for each answer:

$$R_j = [r_1^j, r_2^j, \ldots, r_K^j].$$

Unlike VideoAgent and VideoTree, which rely on embeddings for retrieval or clustering, our method directly operates on raw captions. This avoids extra feature modules, preserves human-readable semantics for interpretability, and reduces computational cost, aligning with our goal of an efficient, training-free VQA system.

3.2.2. Temporal Smoothing

R_j may contain unreliable relevance scores since the correct answer is still unknown during inference. Next, we aggregate information along the dimension of "candidate answers." This allows incorrect options, which may contribute misleading semantic signals, to be suppressed during the subsequent reasoning. For each clip $c_i, i = 1..K$, we compute the sum:

$$S_i = \sum_{j=1}^{n} r_i^j$$

From these aggregated scores, the sequence $[S_1, S_2, \ldots, S_K]$ is formed. Then we apply a moving average with a window size W, leading to a smoothed sequence $[\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_K]$. Temporal smoothing maintains coherent event segments by reducing sudden spikes, drops and local noise, preventing unnecessary fragmentation and improving model stability.

3.2.3. Event Segmentation

Since only a subset of video clips are semantically relevant to the question $\mathcal{Q}$, it is necessary to group these clips into coherent segments while filtering out irrelevant ones.

To identify which parts of the video warrant further analysis, we define a mean threshold $\tau = (\sum_{i=1}^{K} \bar{S}_i)/K$. The following set $\mathcal{I}$ contains all i whose corresponding score $\bar{S}_i$ exceeds τ:

$$\mathcal{I} = \{i \mid \bar{S}_i \geq \tau\}$$

In practice, the indices in $\mathcal{I}$ naturally cluster along the timeline, forming contiguous *semantic events*. This process suppresses irrelevant captions without relying on hand-crafted rules, thereby improving robustness across diverse video–question pairs. Figure 2 illustrates a qualitative example, where the highlighted regions correspond to key actions relevant to $\mathcal{Q}$, effectively guiding the reasoning process without manual intervention.

3.3. Stage 2.2: Background Extraction

Captions generated by visual-language models (VLMs), whether at the clip or frame level, pro-

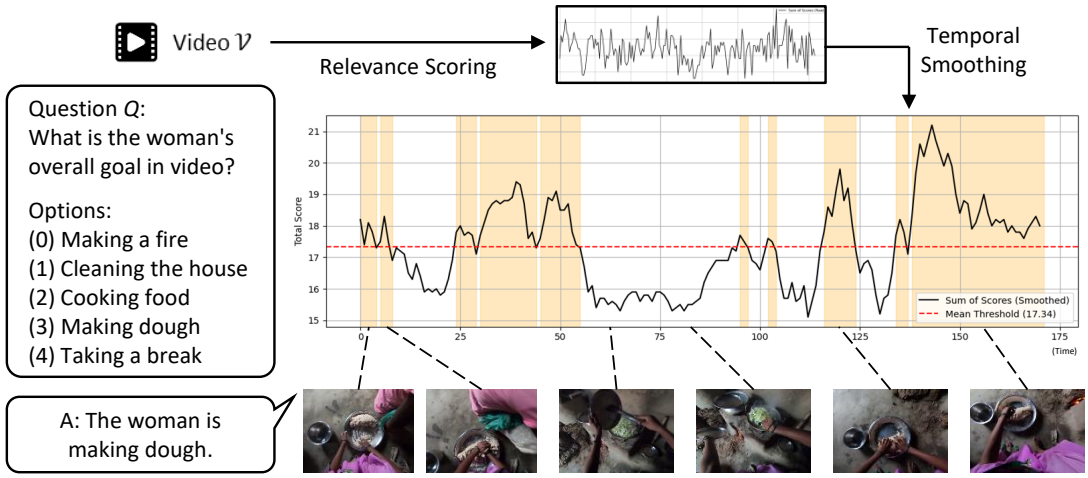

Figure 2: Example of event segmentation. Highlighted regions correspond to semantically meaningful segments that align with key actions (e.g., making dough), directly supporting question answering.

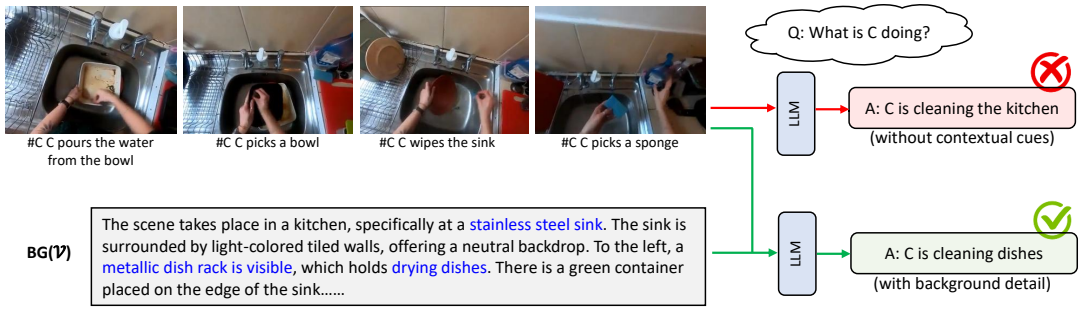

Figure 3: Illustration of the background extraction module. Contextual cues clarify the correct answer (cleaning dishes) rather than the more ambiguous interpretation (cleaning the kitchen).

vide useful descriptions but tend to emphasize foreground actions while omitting background details such as scene layout, surrounding objects, or environment. Consequently, short captions alone often lack sufficient context, even though such cues can be crucial for accurate reasoning.

Figure 3 illustrates this limitation. Although the action sequence includes wiping the sink and adjusting the tap, the absence of spatial and object-level cues makes it unclear whether the subject is cleaning the kitchen or the dishes. Incorporating background information extracted by a VLM clarifies the scene: a stainless steel sink, drying rack, and dishes indicates that the correct action is dish cleaning.

To address this, we propose a plug-and-play background extraction module. We first identify the most relevant caption:

$$c^* = \arg\max_{c_i} \bar{S}_i$$

From the corresponding clip, we sample a single frame and use a promptable VLM to generate a background description that includes contextual elements such as scene, objects, and actions:

$$\text{BG}(\mathcal{V}) = \text{VLM}(\text{frame}(c^*))$$

Because it relies on a single frame, this module is lightweight, avoids redundancy, and is architecture-agnostic. It can be integrated seamlessly into frameworks such as *LLoVi* (Zhang et al., 2024), *VideoAgent* (Wang et al., 2024), and *VideoTree* (Wang et al., 2025) without altering their reasoning pipelines. Empirically, BG($\mathcal{V}$) consistently improves accuracy across all three frameworks.

3.4. Stage 3: Answer Reasoning

We organize the following into a structured prompt for answer reasoning: (i) the selected captions

$$\mathcal{E} = \{c_i \mid i \in \mathcal{I}\},$$

(ii) the background description BG($\mathcal{V}$), (iii) question $\mathcal{Q}$, and (iv) answer $\mathcal{A}$. With the prompt, an LLM is instructed to reason about the inputs and generate a final prediction. At this stage, a stronger model, such as GPT-4, is employed.

3.5. Prompt Templates

Our framework uses the following three prompt templates for relevance scoring, background extraction, and final answer reasoning, as shown in Figure 4.

Relevance Score Prompt

You are presented with a single caption from a first view video clip (#C means the first person view, and #O indicates another). The ultimate goal is to answer a question related to this video, choosing the correct option out of five possible answers. Please provide the answer with a single letter (A, B, C, D, E).
It is crucial that you imagine the visual scene as vividly as possible to enhance the accuracy of your response. After selecting your answer, rate your confidence level in this choice on a scale from 1 to 100, where 1 indicates low confidence and 100 signifies high confidence.
Please provide a concise one-sentence explanation for your chosen answer. If you are uncertain about the correct option, select the one that seems closest to being correct.
Meanwhile, evaluate how relevant each of the five options (A, B, C, D, E) is to the given caption. Provide five relevance scores (one for each option), where each score is between 1 and 10, with 1 indicating low relevance and 10 signifying high relevance. Return the relevance scores in the format of a list of five scores.

Caption: `{caption}`

###

Question: `{question}`
Options:
A: `{option0}`
B: `{option1}`
C: `{option2}`
D: `{option3}`
E: `{option4}`

###

The prediction, explanation, confidence, and relevance scores are (please respond in the format of 'prediction: \n explanation: \n confidence: \n option relevance: \n'):

Background Extraction Prompt

The following image is a frame extracted from a video. Your task is to generate a clear and factual description of both the environment and the human actions within the frame. Follow these rules:

1. Start by describing the overall scene—including the setting, key objects, and any static elements that provide context. This should be a structured overview that helps establish where the actions take place.

2. Then, describe the human actions and interactions with objects in a continuous, natural flow. Do not list frames individually. Instead, explain how movements and interactions unfold over time.

3. Only state what is visually present. Avoid assumptions, emotions, or unnecessary interpretations.

4. Ensure clarity and conciseness. The description should be straightforward but detailed enough to capture key actions and their relationship to the environment.

Answer Reasoning Prompt

You are presented with a textual description of a video clip, which has been segmented into several meaningful events. Each event contains captions sampled from different frames in the video. Your task is to answer a question related to this video, choosing the correct option out of five possible answers. Please provide the answer with a single letter (A, B, C, D, E).
It is crucial that you imagine the visual scene as vividly and coherently as possible based on the provided event-based descriptions. After selecting your answer, rate your confidence level in this choice on a scale from 1 to 100, where 1 indicates low confidence and 100 signifies high confidence.
Please provide a concise one-sentence explanation for your chosen answer. If you are uncertain about the correct option, select the one that seems closest to being correct.

Description: `{captions}`

###

Question: `{question}`
Options:
A: `{option0}`
B: `{option1}`
C: `{option2}`
D: `{option3}`
E: `{option4}`

###

The prediction, explanation, and confidence are (please respond in the format of 'prediction: \n explanation: \n confidence: \n'):

Figure 4: Prompt templates used in our framework for relevance scoring, background extraction, and final answer reasoning.

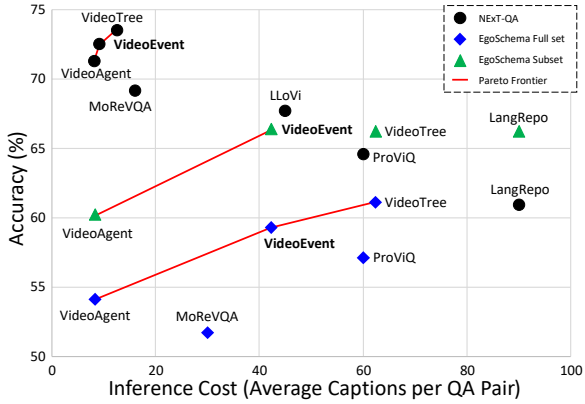

Figure 5: Trade-off between accuracy and inference cost. Each marker denotes a dataset–method pair (NextQA, EgoSchema full set, or subset). Lower cost and higher accuracy are preferred, with the red line indicating the Pareto frontier.

4. Experiments

4.1. Datasets, Metrics, and Experimental Setup

We evaluate VideoEvent on two representative benchmarks: EgoSchema (Mangalam et al., 2023) and NExT-QA (Xiao et al., 2021). EgoSchema consists of long egocentric videos (3 minutes) paired with multiple-choice QA, offered in a subset (500 QA) with public labels and a full set (5,031 QA) requiring leaderboard submission. NExT-QA focuses on daily human activities (average length 44 seconds) with 8,564 QA pairs, covering temporal, action, and causal reasoning questions. We report accuracy as the primary evaluation metric, and also measure the number of captions included in the reasoning prompt as a proxy for inference cost.

For implementation, we follow prior work (Zhang et al., 2024) and use LaViLa (Zhao et al., 2023) as the visual captioner (1 Hz for EgoSchema, 0.5 Hz for NExT-QA). GPT-4o-mini is employed for event relevance scoring and background extraction, while GPT-4 is used for final reasoning to ensure comparability with state-of-the-art methods.

4.2. Results and Analyses

We compare VideoEvent with existing training-free VQA approaches, focusing on accuracy and inference cost to demonstrate its balanced performance. To illustrate this trade-off, we adopt the concept of Pareto optimality, where ideal solutions lie on or near the Pareto frontier. Our method achieves competitive accuracy with an efficient accuracy–cost balance for long-video QA. It consistently positions near the Pareto frontier across datasets, highlighting its strong cost-efficiency and practical scalability.

4.2.1. Comparison on EgoSchema and NextQA

We observe a trade-off between accuracy and inference cost. For our task, the goal is to maximize accuracy while minimizing inference cost. As shown in Figure 5 and Table 1, our approach maintains a strong balance between efficiency and accuracy, consistently near the Pareto frontier, demonstrating high cost-efficiency.

EgoSchema. On EgoSchema, our method requires approximately 42 captions per question on average, which is substantially fewer than most other methods, except for VideoAgent (Wang et al., 2024). In comparison, both LLoVi (Zhang et al., 2024) and LangRepo (Kahatapitiya et al., 2025) each require 180 captions. As shown in Table 1, our flat prompt variant achieves 64.2% subset accuracy, while our event-based prompt variant reaches 66.4% subset accuracy, a performance comparable to or even surpassing other top-performing methods.

For the full-set evaluation, VideoTree (Wang et al., 2025) reports a higher accuracy (61.1% vs. our 59.3%). However, this result is obtained using a few-shot prompting strategy on EgoSchema. When we reevaluate it under a fairer zero-shot setting by removing the exemplars, its subset accuracy drops from 66.2% to 63.4%. This performance gap suggests that its reported full-set accuracy may also benefit significantly from its few-shot setup. However, the few-shot configuration introduces a substantial increase in cost, requiring over 1,100 captions per question due to its 6-shot format. In contrast, VideoEvent achieves competitive performance with less than 45 captions on average, offering a more practical and scalable alternative for resource-constrained settings.

NextQA. Our method uses only 9.2 captions per video and achieves 72.5% accuracy, comparable to VideoTree's 73.5%, and better than VideoAgent (71.3%) and LLoVi (67.7%). Notably, VideoTree reports its best performance using an average of 12.6 captions per video, which is still higher than ours. This highlights the efficiency of our method.

In summary, our method reduces caption input by approximately 30% compared to VideoTree across both datasets, while maintaining competitive accuracy. Its consistent performance near the Pareto frontier underscores its strong practical viability in real-world scenarios that demand both accuracy and efficiency. In contrast, VideoAgent and VideoTree may be better suited for applications that prioritize either minimal input or maximum accuracy alone.

Method	EgoSchema			NextQA	
	Avg. Capt.	Subset Acc.	Full set Acc.	Avg. Capt.	Acc.
Modular/Tool-based Methods					
VideoMindPalace (Huang et al., 2025)	-	**68.6**	-	-	**75.8**
DoraemonGPT (Yang et al., 2024)	-	-	-	-	54.7
ViperGPT (Surís et al., 2023)	-	-	-	-	60.0
MoReVQA (Min et al., 2024)	30	-	51.7	16	69.2
ProViQ (Choudhury et al., 2024)	60	-	57.1	60	64.6
Frame-Selection-Based Methods					
VideoAgent (Wang et al., 2024)	8.4	60.2	54.1	8.2	71.3
VideoTree (Wang et al., 2025) (few-shot)	1142.4	66.2	**61.1**	-	-
VideoTree (zero-shot)	62.4	63.4	-	12.6	**73.5**
LLoVi (Zhang et al., 2024)	180	57.6	50.3	45	67.7
LangRepo (Kahatapitiya et al., 2025)	180	66.2	41.2	90	60.9
Frame-Selection-Based Methods with BG Extraction					
VideoEvent (Flat Prompt)	43.1	64.2	58.1	9.2	72.5
VideoEvent (Event-based Prompt)	42.3	66.4	59.3	9.2	71.7
VideoAgent + **BG_Extract**	8.4	61.2	-	-	-
LLoVi + **BG_Extract**	180	63.8	-	-	-
VideoTree + **BG_Extract**	1142.4	**67.4**	-	-	-

Table 1: Comparison of training-free VQA methods on EgoSchema and NextQA. Higher accuracy and lower caption count are better. "Avg. Capt." denotes the average number of captions included in the final reasoning prompt per QA sample after event segmentation and sampling, including any few-shot QA examples when used. For instance, VideoTree (few-shot) includes 6×180 additional captions. Methods are grouped by video processing paradigm.

4.2.2. Caption Count as a Cost Proxy

Most training-free VideoQA methods report only the number of captions included in the reasoning prompt and do not provide token-level statistics. As a result, direct comparison based on token consumption is often infeasible.

To enable fair comparison across methods, we adopt the number of captions in the final reasoning prompt ("Avg. Capt.") as a practical proxy for input token cost. Since the length of individual captions is relatively stable across methods, total token consumption scales approximately linearly with the number of captions.

We empirically validate this approximation on the EgoSchema subset by comparing both caption counts and token counts. The relative ratios are nearly identical across methods. For example, VideoTree requires 535.5 tokens and 62.4 captions on average, corresponding to 1.47× and 1.46× our cost, respectively. LLoVi uses 1541.8 tokens and 180 captions, which are 4.24× and 4.21× higher than ours. In contrast, our method requires only 363.9 tokens and 42.8 captions.

These consistent ratios confirm that caption count serves as a reliable and reproducible proxy for prompt-level inference cost.

4.2.3. Definition of Inference Cost

Inference cost in training-free LLM-based VideoQA frameworks arises from two primary components: (1) the length of the reasoning prompt, and (2) the number of model invocations.

Prompt-Level Cost. This corresponds to the captions included in the final reasoning prompt. As established in Section 4.2.2, caption count closely reflects token consumption and therefore provides a consistent measure of prompt-level cost.

Model Invocation Cost. For each QA sample, our framework performs K lightweight LLM inferences (GPT-4o-mini) for relevance scoring, one additional inference for background extraction, and one final inference using a stronger LLM (GPT-4) for answer reasoning.

Each relevance-scoring inference processes a single caption and returns relevance scores for all answer options simultaneously. Thus, the number of lightweight LLM inferences scales linearly with the number of captions K, rather than with $K \times n$, where n denotes the number of answer options.

Importantly, the final reasoning stage dominates overall token consumption due to its substantially longer prompt. Therefore, reducing the number of captions included in this stage directly lowers the dominant component of inference cost.

4.2.4. Computational Latency Analysis

The computational complexity of our framework is linear in the number of captions K.

Relevance scoring requires $O(K)$ lightweight LLM inferences. These inferences are independent across captions and can be executed in parallel. Moreover, lightweight model inference typically incurs significantly lower latency compared to the final reasoning stage.

The dominant latency arises from the single GPT-4 inference used for answer reasoning. The latency of this stage primarily depends on prompt length, which scales proportionally with the number of captions included in the prompt. Since our event segmentation strategy substantially reduces the retained captions, it proportionally reduces the dominant reasoning latency.

While absolute API latency may vary across deployment environments, this analysis provides a consistent and reproducible estimate of relative computational cost across methods.

4.3. Ablation Study

We conduct ablation studies to analyze the effect of event selection, temporal smoothing, caption sampling, and background integration.

4.3.1. Top-5 vs. Full Event Strategy

Our segmentation method may produce multiple relevant events. Figure 6 compares using all events versus the top five, with an average of about 11 retrieved per video. This comparison illustrates how different event selection depths affect both context preservation and efficiency. The Full strategy (Figure 6(a)) preserves more context (86 captions on average), while the Top-5 strategy (Figure 6(c)) uses 68 captions with a slight accuracy drop from 62.6% to 60.6%. This shows that accurately locating relevant segments provides a performance gain, though discarding lower-ranked ones may omit useful context.

4.3.2. Caption Sampling Rate within Events

To further reduce input cost, we examined different event selection strategies on the EgoSchema subset, as shown in Table 2. Using all retrieved events requires 86 captions and yields 62.6% accuracy, while restricting to the top five events reduces the input to 68 captions but also slightly lowers accuracy to 60.6%. Applying uniform sampling within events further decreases the input size: the Full Event strategy drops to 42 captions while maintaining the same 62.6% accuracy, and the Top-5 strategy decreases to 33 captions with a marginal reduction to 60.4%.

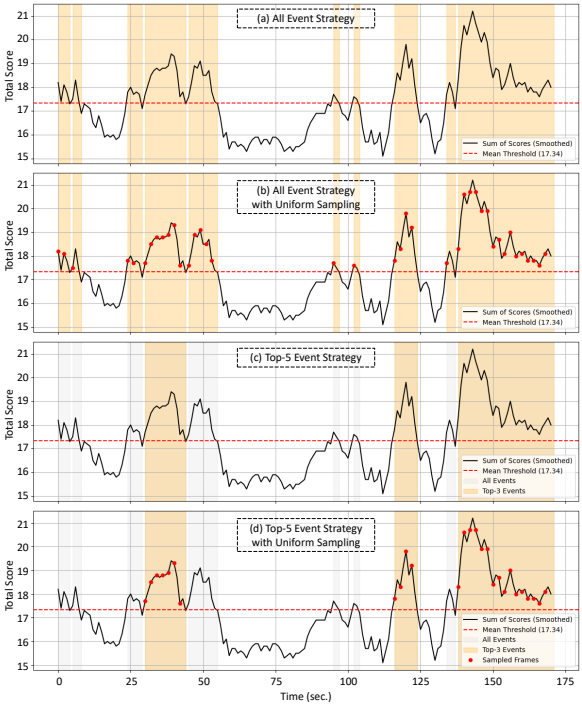

Figure 6: Comparison of event selection strategies. Sampled points are marked in red, and shaded regions indicate events.

Table 2: Ablation study of event selection strategies on EgoSchema-subset.

Method	Avg. Caption	Accuracy
Full Event	86	62.6
Top-5 Event	68	60.6
Full Event w/ Uniform Sampling	42	**62.6**
Top-5 Event w/ Uniform Sampling	33	60.4

These results suggest that (i) once relevant events are correctly identified, uniform sampling has minimal impact on performance, and (ii) the trend is consistent across both Full and Top-5 strategies. Therefore, we adopt the *Full Event Strategy with Uniform Sampling* as the default in subsequent experiments, as it strikes the best balance between efficiency and accuracy.

4.3.3. Window Size for Smoothing

We analyze the effect of the temporal smoothing window W in Stage 2.1, as shown in Table 3. Accuracy peaks at 59.3% when $W = 13$, with an average of 42 captions and 9.6 events. Smaller windows (e.g., $W = 5$) lead to over-segmentation, requiring more captions (44 on average) while reducing accuracy to 58.0%. Conversely, overly large windows (e.g., $W = 15$) over-smooth the relevance curve, lowering accuracy to 58.7% despite further reducing captions (41.9 on average). These results suggest that a moderately large window offers the

Table 3: Ablation study of window size W on EgoSchema full set.

	W=5	W=7	W=11	W=13	W=15
Accuracy (%)	58.0	58.4	58.9	**59.3**	58.7
Avg. Caption	44.21	43.85	42.88	42.35	41.87

Table 4: Ablation study of the background frame selection strategies (based on *Full Event Strategy with Uniform Sampling*).

Strategy	Accuracy (%)
No background frame	62.6
+ Background (middle index)	63.6
+ Background (max relevance)	**64.2**

best trade-off between suppressing noise and preserving informative temporal variations.

4.3.4. Effect of Background Frame Selection

We further investigate the impact of background frame selection using the *Full Event Strategy with Uniform Sampling*, as illustrated in Table 4. Without background information, the model achieves 62.6% accuracy. Selecting the middle-indexed frame improves accuracy to 63.6%, while choosing the highest-scored frame from the most relevant event segment yields the best result at 64.2%. These findings confirm that background information enhances the LLM's understanding, and that the choice of frame significantly influences performance.

4.3.5. Generalizability of the Background Extraction Module

The proposed background extraction module is generalizable across different frameworks. The enhancement of incorporating background information in the EgoSchema subset is illustrated in Table 5. These consistent gains demonstrate that even minimal additional visual context can enhance reasoning capabilities in training-free LLM-based VQA systems.

Table 5: Accuracy improvement from adding background information across different methods on EgoSchema-subset.

Framework	Base	+BG	Improvement
LLoVi	61.2	63.8	+2.6%
VideoAgent	60.2	61.2	+1.0%
VideoTree	66.2	67.4	+1.2%

4.3.6. Flat vs. Event-based Prompting Strategy

In Stage 3, the textual prompt to the LLM is based on the selected captions. We further compare two prompting strategies: a flat prompt that directly concatenates all captions, and an event-based prompt that organizes captions by semantic segments. The event-based strategy improves performance on EgoSchema, while maintaining comparable accuracy on NExT-QA, highlighting the benefit of structured context presentation for video QA, particularly for longer videos.

5. Conclusions

We present VideoEvent, a lightweight and training-free framework for VQA that leverages semantic relevance to guide event segmentation and background augmentation. By identifying question-relevant temporal spans and introducing a plug-and-play visual background extraction module, our method addresses the limitations of existing approaches in temporal coherence and contextual understanding without increasing inference cost or system complexity. Experimental results demonstrate that VideoEvent achieves competitive accuracy and significantly reduces inference costs, showcasing superior efficiency and generalization capability. Moreover, our method is highly extensible and can be integrated into other training-free frameworks such as LLoVi, VideoAgent, and VideoTree, consistently improving answer accuracy with minimal modification. Future work will focus on refining event segmentation, incorporating richer modalities, and exploring structured event representations to further enhance reasoning precision and generalizability.

6. Bibliographical References

Tuo An, Yunjiao Zhou, Han Zou, and Jianfei Yang. 2024. IoT-LLM: Enhancing Real-World IoT Task Reasoning with Large Language Models. *arXiv preprint arXiv:2410.02429*.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. 2024. AutoAgents: A Framework for Automatic Agent Generation. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, page 3.

Rohan Choudhury, Koichiro Niinuma, Kris M. Kitani, and László A. Jeni. 2024. Zero-Shot Video Question Answering with Procedural Programs.

In *Proc. European Conference on Computer Vision (ECCV)*, volume 14977, pages 324–342.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2024. MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. In *Proc. International Conference on Learning Representations (ICLR)*.

Zeyi Huang, Yuyang Ji, Xiaofang Wang, Nikhil Mehta, Tong Xiao, Donghyun Lee, Sigmund Vanvalkenburgh, Shengxin Zha, Bolin Lai, Licheng Yu, et al. 2025. Building a Mind Palace: Structuring Environment-Grounded Semantic Graphs for Effective Long Video Analysis with LLMs. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24169–24179.

Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. 2025. Language Repository for Long Video Understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5627–5646.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proc. International Conference on Machine Learning (ICML)*, pages 19730–19742.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 34892–34916.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 43447–43478.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. EgoSchema: A Diagnostic Benchmark for Very Long-Form Video Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 46212–46244.

Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. 2024. MoReVQA: Exploring Modular Reasoning Models for Video Question Answering. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13235–13245.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI Tasks with ChatGPT and Its Friends in Hugging Face. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 38154–38180.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. 2024. EVA-CLIP-18B: Scaling CLIP to 18 Billion Parameters. *arXiv preprint arXiv:2402.04252*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. ViperGPT: Visual Inference via Python EXecution for Reasoning. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11888–11898.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024. VideoAgent: Long-Form Video Understanding with Large Language Model as Agent. In *Proc. European Conference on Computer Vision (ECCV)*, pages 58–76.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2025. VideoTree: Adaptive Tree-Based Video Representation for LLM Reasoning on Long Videos. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786.

Huatao Xu, Liying Han, Qirui Yang, Mo Li, and Mani Srivastava. 2024. Penetrative AI: Making LLMs Comprehend the Physical World. In *Proc. International Workshop on Mobile Computing Systems and Applications (HotMobile)*, pages 1–7.

Zongxin Yang, Guikun Chen, Xiaodi Li, Wenguan Wang, and Yi Yang. 2024. DoraemonGPT: Toward Understanding Dynamic Scenes with Large Language Models (Exemplified as A Video Agent). In *Proc. International Conference on Machine Learning (ICML)*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large Language Models Are Versatile Decomposers: Decomposing Evidence and Questions for Table-Based Reasoning. In *Proc. International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–184.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas

Bertasius. 2024. A Simple LLM Framework for Long-Range Video Question-Answering. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 21715–21737.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. 2023. Learning Video Representations from Large Language Models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6586–6597.