Logic-in-Frames: Dynamic Keyframe Search via Visual Semantic-Logical Verification for Long Video Understanding

Weiyu Guo[1†] **Ziyang Chen**[1†] **Shaoguang Wang**[1] **Jianxiang He**[1]
Yijie Xu[1] **Jinhui Ye**[2] **Ying Sun**[1,2*] **Hui Xiong**[1,2*]
[1]Thrust of Artificial Intelligence, HKUST (Guangzhou), China
[2]Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
wguo395@connect.hkust-gz.edu.cn; yings@hkust-gz.edu.cn; xionghui@ust.hk

Abstract

Understanding long video content is a complex endeavor that often relies on densely sampled frame captions or end-to-end feature selectors, yet these techniques commonly overlook the logical relationships between textual queries and visual elements. In practice, computational constraints necessitate coarse frame subsampling, a challenge analogous to "finding a needle in a haystack." To address this issue, we introduce a semantics-driven search framework that reformulates keyframe selection under the paradigm of `Visual Semantic-Logical Search`. Specifically, we systematically define four fundamental logical dependencies: 1) spatial co-occurrence, 2) temporal proximity, 3) attribute dependency, and 4) causal order. These relations dynamically update frame sampling distributions through an iterative refinement process, enabling context-aware identification of semantically critical frames tailored to specific query requirements. Our method establishes new SOTA performance on the manually annotated benchmark in key-frame selection metrics. Furthermore, when applied to downstream video question-answering tasks, the proposed approach demonstrates the best performance gains over existing methods on LongVideoBench and Video-MME, validating its effectiveness in bridging the logical gap between textual queries and visual-temporal reasoning. The code is available at `https://github.com/guoweiyu/Logic-in-Frames`.

1 Introduction

Vision-Language Models (VLMs) Yin et al. (2024) have achieved remarkable progress in video understanding Zou et al. (2024); Tang et al. (2023), particularly in video question answering Wang et al. (2024d); Zhang et al. (2023), demonstrating potential for modeling real-world scenarios. However, existing methods can only simultaneously process a limited number of frames due to the inherent token limit and extremely high dimension of spatio-temporal video data, especially for long videos. Furthermore, uniformly sampled keyframes are query-agnostic and insufficient to represent query-related contents. To tackle these challenges, this paper addresses a pivotal research question:

> *How can we efficiently and accurately select keyframes that are semantically critical for answering video-based queries?*

We hypothesize that deconstructing visual semantic and logical cues (e.g., target objects, logical relations including *temporal*, *spatial*, *attribute*, and *causal* relationships between visual entities) from textual queries enables effective identification of task-relevant frames through heuristic sampling and search. Building on this insight, we propose `Visual Semantic-Logical Search` (`VSLS`), a novel keyframe search framework that incorporates target object confidence estimation and joint verification of visual semantic logic into the iterative update of frame sampling distribution and

*Corresponding authors.
†Equal Contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

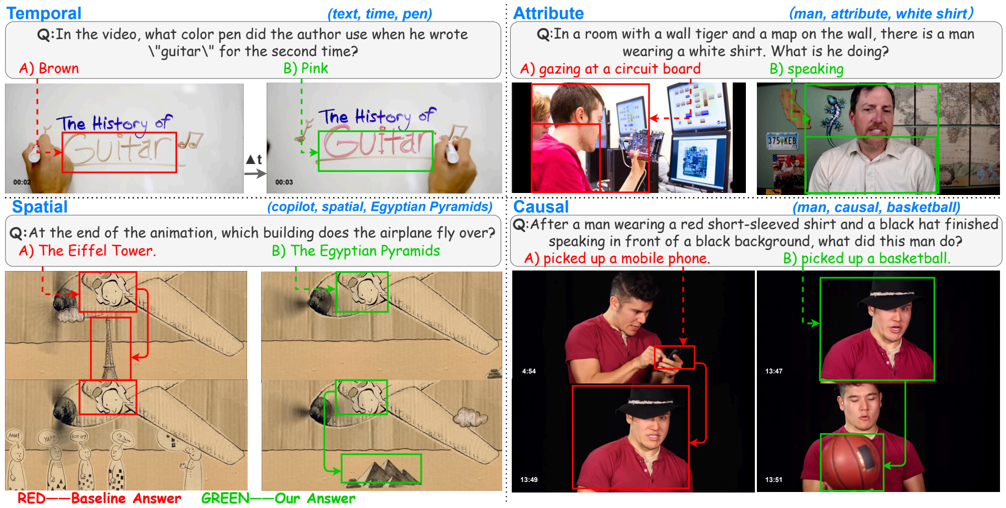

Figure 1: Examples of four types of visual semantic-logical relationships in video QA detected by our VSLS framework: **Temporal** (text, time, pen), **Attribute** (man, attribute, white shirt), **Spatial** (copilot, spatial, Egyptian Pyramids), and **Causal** (man, causal, basketball). Green boxes indicate correct answers, while red boxes show baseline errors.

selects the most informative frames with the highest confidence. Experimental results show that our approach requires only sparse sampling (1.4% of frames per video on average) to identify critical frames, significantly reducing computational complexity compared to conventional dense sampling strategies while maintaining performance on downstream video understanding tasks.

Compared to conventional methods, VSLS shows three distinct advantages. First, the framework is training-free and highly efficient in comparison with dense captioning Chen et al. (2024c); Kim et al. (2024); Wang et al. (2024c) or video clustering Wang et al. (2024f); Rajan and Parameswaran (2025) strategies, sampling only 1.4% of frames on average in LVHAYSTACK. Second, it explicitly models logical binary relations (namely spatial, temporal, attribute and causal) in the query beyond simple target detection Ye et al. (2025b), utilizing additional visual semantic feature and enhancing logical consistency throughout the reasoning process. Third, VSLS is a plug-and-play module, which can be seamlessly integrated into existing VLM pipelines without cross-component dependencies.

We further examine VSLS on several public datasets, including LONGVIDEOBENCH Ye et al. (2025a), a comprehensive benchmark for long video understanding; VIDEO-MME Fu et al. (2024), a widely adopted multimodal video question answering dataset; and HAYSTACK-LVBENCH Ye et al. (2025a) with meticulously annotated keyframes based on human feedback for more precise analysis. Extensive experiments demonstrate significant improvements in both the semantic similarity and temporal coverage between the retrieved keyframes and the ground truth labels, as well as the accuracy in downstream video question answering tasks. More importantly, with only **1.4%** of video frames (EGO4D Grauman et al. (2022)) sampled in the search iteration, our method achieve an **8.7%** improvement in GPT-4O Hurst et al. (2024)'s long video QA accuracy. This performance gain is attributed to our simple yet powerful observation: query-guided visual semantic logic retrieval can mitigate the gap between potential visual logic in video frames and the logic expressed in the query. To be specific, constructing ternary logic triplets with visual elements (e.g., `object1, logic type, object2`) can enhance downstream reasoning capabilities when performing textual-visual retrieval.

To the best of our knowledge, we are arguably the first to search for keyframes in long videos by detecting visual semantic logic, with potential extensions to other textual-visual retrieval tasks. Our main contributions are as follows:

- We define four fundamental types of semantic logic relations in video QA tasks, including *temporal*, *causal*, *attribute*, and *spatial* relations, which can be accurately detected across various datasets.
- We sample only 1.4% of frames on average of frames on average during keyframe search through heuristic sampling and distribution updating by different visual semantics and logical relations.
- We comprehensively evaluate retrieval efficiency, semantic similarity, temporal coverage, and video question answering accuracy across several widely used video understanding datasets, demonstrating significant improvements in downstream tasks.

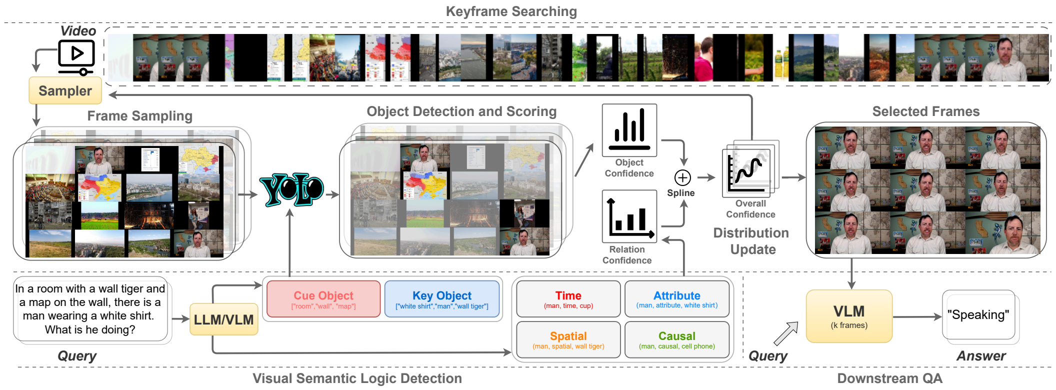

Figure 2: **Our `VSLS` Framework for Efficient Keyframe Selection.** `VSLS` sparsely samples frames and selects key ones via object detection and logic verification: 1) Use LLM&VLM to extract cue/target objects and four logic types (*spatial*, *temporal*, *attribute*, *causal*); 2) Adaptive sampling with evolving confidence; 3) Detect objects via YOLO-WORLD; 4) Fuse scores with a spline to identify frames for downstream tasks.

2 Method

Although existing long-context VLM frameworks implement keyframe search for video QA tasks Liang et al. (2024); Park et al. (2024); Tan et al. (2024); Wang et al. (2024b,e); Yu et al. (2024), their computational efficiency and search accuracy remain suboptimal. To address this *needle-in-a-haystack* challenge Wang et al. (2025); Zhao et al. (2024), we propose a novel method `VSLS` that aligns the semantic relations between the text modality and video modality, improving the plausibility of logical reasoning and the performance of downstream tasks.

2.1 Task Formulation

Given a video sequence $V = \{f_t\}_{t=1}^{N_v}$ with N_v frames and a query Q, the ideal temporal search framework aims to retrieve the minimal keyframe subset $V^K = \{f_{m_i}\}_{i=1}^{K} \subseteq V$ with K keyframes that satisfies the following:

- **Conservation**: The keyframe subset $V^K \subseteq V$ must satisfy the answer consistency condition: $\mathcal{A}(V^K, Q) = \mathcal{A}(V, Q)$, where $\mathcal{A}(\cdot)$ denotes the video QA function.
- **Compactness**: V^K must be a minimal subset that preserves completeness, which means that no frame in V^K can be removed without hindering the accuracy and efficiency of video QA.

2.2 Visual Semantic Logic Extraction

Starting from a question Q and uniformly sampled frames $\overline{V}_N$ from video V, our goal is to extract key visual elements to answer Q. We first classify the detected objects in Q and $\overline{V}_N$ into two categories:

- **Key Objects**: The main participants or references in the scene that the question explicitly or implicitly focuses on (e.g., "*person*", "*microphone*").
- **Cue Objects**: Secondary or contextual entities that help locate or disambiguate the Key Objects (e.g., "*book*", "*tiger painting*").

To further leverage semantic and logical links among these objects, we define a set of relations $\mathcal{R} \subseteq \mathcal{O} \times \Delta \times \mathcal{O}$, where each relation $r = (o_i, \delta, o_j) \in \mathcal{R}$, with $o_i, o_j \in \mathcal{O}$ denoting detected objects in the key and cue objects dataset and $\delta \in \Delta$ representing one of the following types of relations:

Spatial Co-occurrence	**Attribute Dependency**
o_i and o_j appear in the same frame, indicating co-occurrence or proximity. ***Example:*** "A person is standing beside a vase." ⇒ `(person, spatial, vase)`	o_i and o_j share visual properties, e.g., color or size. ***Example:*** "A person wears a black shirt." ⇒ `(person, attribute, black shirt)`
Temporal Proximity	**Causal Order**
o_i and o_j occur in close frames, linking sequences or transitions. ***Example:*** "After a dog entered the room, a cat entered." ⇒ `(dog, temporal, cat)`	o_i and o_j follow a cause-effect or prerequisite order. ***Example:*** "A little girl broke the vase." ⇒ `(little girl, causal, pieces)`

Algorithm 1: `Visual Semantic-Logical Search`

```
Function SemanticLogicalTemporalSearch(V, Q, K, Δ_t, τ, α, γ)
    O, R ← ParseQuestion(Q)                                  // Extract key/cue objects and relations
    P ← Uniform, B ← |V|, S ← ∅, N_v ← |V|                   // Initialize distribution and state
    while B > 0 and |O| > 0 do
        k ← ⌊√B⌋, G ← Grid(Sample(P, k^2))                   // Adaptive grid sampling
        Ω ← DetectObjects(G)                                 // Detect objects in sampled frames
        foreach t ∈ G do
            C_t ← CalculateBaseScore(Ω_t)                    // Base detection confidence
            foreach r_type ∈ R do
                δ ← Processrelation(r_type, Ω, Δ_t, τ, α, γ) // relations require distinct processing
                C_t ← C_t + δ
            UpdateScores(S, t, C_t)                          // Update global score registry
        DiffuseScores(S, w)                                  // Temporal context propagation
        P ← NormalizeDistribution(S), B ← B − k^2            // Update sampling distribution
        foreach g ∈ TopK(S, K) do
            if Ω[g] ∩ O ≠ ∅ then
                O ← O \ Ω[g]                                 // Remove identified key objects
    return TopK(S, K)                                        // Return top-K keyframes
```

The choice of these four relations draws on core concepts in linguistics and logic Cohen (1968); Sowa (2000); Talmy (2000), which identify spatial, temporal, attributive, and causal aspects as fundamental for structuring, perceiving, and communicating information about events and states. For more details on this selection, please see appendix A for reference. As shown in Figure 1, we construct semantic-logical relations that support a broad range of question-answering tasks. Specifically, questions involving temporal queries (*when does X happen?*"), causal reasoning (*why did Y occur?*"), attribute dependence (*What is the person wearing sunglasses doing?*"), or spatial constraints (*Who is standing next to the red car?*") can be answered more reliably by incorporating these structured relations and contextual cues.

2.3 Iterative Semantic-Logical Temporal Search

Based on the extracted key and cue objects and their logic relations, our algorithm iteratively searches for keyframes through semantic and logical reasoning, including four main stages: **Frame Sampling** (Sec. 2.3.1), **Object Detection and Scoring** (Sec. 2.3.2), **Visual Semantic Logic Detection** (Sec. 2.3.3), and **Distribution Update** (Sec. 2.3.4). The pseudocode is shown in Algorithm 1, and Algorithm 2 provides a more detailed version.

2.3.1 Frame Sampling

Given a video with N_v frames, we employ a probability sampling strategy instead of exhaustively scanning all frames to improve searching efficiency. Let P denote a uniform distribution over all frames, then the sampling process is defined as:

$$I_s = \mathrm{Sample}(N_v, N_s, P), \tag{1}$$

where $\mathrm{Sample}(\cdot,\cdot,\cdot)$ is a function that draws N_s indices from the population N_v based on the probability distribution P. To further leverage the detection capability of YOLO, we stack the sampled frames on a $k \times k$ grid, which requires that the sample size N_s be a square number. The benefits of such practice are analyzed in detail in Appendix C.1. Although P is initially uniform, it can be adapted over multiple rounds of sampling to focus on frames of greater interest in the video.

2.3.2 Object Detection and Scoring

In this stage, we construct the detection search space by taking the union of both key objects and cue objects. For each iteration, we detect objects in the N_s sampled frames using a lightweight model like YOLO-WORLD Cheng et al. (2024a) for high efficiency and score the frames based on detection confidence. Specifically, let Ω_t be the set of detected objects in the frame at time t, c_o the confidence of each detected object, and w_o the corresponding weight. We define the frame score as:

$$C_t = \max_{o \in \Omega_t} \big(c_o \cdot w_o\big). \tag{2}$$

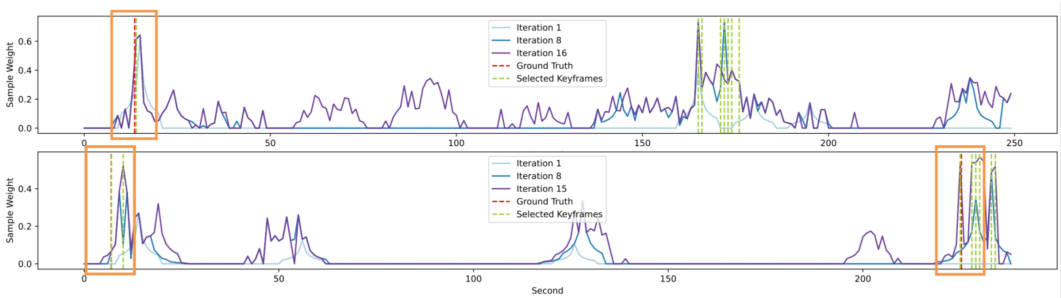

Figure 3: **Sample weight evolution under VSLS optimization for keyframe selection.** Top: 16 iterations show progressive convergence toward Ground Truth (red). Bottom: 15 iterations demonstrate similar alignment. Yellow highlights indicate precise matches between algorithm outputs (green) and manual annotations.

If the confidence score of any key object exceeds a predefined threshold, we will add it to a list to maintain a record of frames where crucial targets have been identified for subsequent processing.

2.3.3 Visual Semantic Logic Detection

Beyond individual object detection and frame-level scoring, we refine each frame's confidence score by modeling higher-level object relations. Let $\mathcal{R}$ be the set of relations, where each $r \in \mathcal{R}$ involves a pair (o_1, o_2) and is labeled by a type r_{type}. Denote C_t as the confidence score at time t, with a global scaling factor α and a relation-specific weight $\gamma_{r_{\text{type}}}$. The confidence refined $C_t^{(r)}$ considering the relation r is defined as:

$$C_t^{(r)} = C_t + \alpha \cdot \gamma_{r_{\text{type}}}. \tag{3}$$

Spatial Relation. A *spatial* relation enforces that two objects o_1 and o_2 must co-occur in the same frame. Let Ω_t be the set of detected objects in frame t. If both $o_1 \in \Omega_t$ and $o_2 \in \Omega_t$, then the corresponding frame confidence is updated by:

$$C_t \leftarrow C_t + \alpha \cdot \gamma_{\text{spatial}}. \tag{4}$$

Attribute Relation. An *attribute* relation is satisfied when the bounding box of o_1 and o_2 share significant overlap in the same frame. Let overlap be the ratio of their intersection area to the minimum of their individual areas of bounding box. If the overlap ratio exceeds a predefined threshold τ ($\tau = 0.5$ in our experimental setting), we increase the frame confidence by:

$$C_t \leftarrow C_t + \alpha \cdot \gamma_{\text{attribute}}. \tag{5}$$

Time Relation. A *time* relation checks whether two objects appear in temporally close frames. Suppose t_i and t_j ($t_i \leq t_j$) are sampled such that $|\,t_j - t_i| < \Delta_t$, where Δ_t is a threshold (e.g. 5 frames in our experimental setting), if o_1 occurs in frame t_i and o_2 in frame t_j, then the confidences of both frames are updated by:

$$C_{t_i} \leftarrow C_{t_i} + \alpha \cdot \gamma_{\text{time}}, \quad C_{t_j} \leftarrow C_{t_j} + \alpha \cdot \gamma_{\text{time}}. \tag{6}$$

Causal Relation. A *causal* relation models an ordering constraint, enforcing that o_1 must appear earlier than o_2. Specifically, if $o_1 \in \Omega_{t_i}$ and $o_2 \in \Omega_{t_j}$ with $t_i < t_j$, we update the confidences of the frames t_i and t_j by:

$$C_{t_i} \leftarrow C_{t_i} + \alpha \cdot \gamma_{\text{causal}}, \quad C_{t_j} \leftarrow C_{t_j} + \alpha \cdot \gamma_{\text{causal}}. \tag{7}$$

Through this scoring mechanism, frames with detected relations will have greater confidence and are more likely to be retrieved as keyframes for the given query and video. We have also performed hyperparameter search experiments and find that $\alpha = 0.3$ (from 0.3, 0.5, 0.7, 1.0) and $\gamma_{r_{\text{type}}} = 0.5$ achieve the best results across different datasets.

2.3.4 Distribution Update

After each iteration of frame sampling, we merge the newly obtained frame confidences into the global score distribution $\{S_f\}$ spanning all frames $f = 1, 2, \ldots, N_v$. When a frame f is selected for detection, its score is set as the confidence value C_f and marked as visited. To incorporate temporal context, we diffuse this updated score to neighboring frames within a window of size w. Denoting each nearby index by $f \pm \delta$ (for $\delta \in [-w, w]$), we apply the following:

$$S_{f\pm\delta} \leftarrow \max\left(S_{f\pm\delta}, \frac{S_f}{1 + |\delta|}\right). \tag{8}$$

In this way, high-confidence frames will increase the scores of close-by frames and ensure temporal continuity. Following these local updates, the sampling probability distribution P is refined using spline interpolation and then normalized. This iteration continues until either the search budget B is reached or all key objects have been successfully identified. The visualization of the probability distribution in different iterations can be seen in Figure 3. Finally, the method outputs the top K frames according to their integrated scores.

3 Experiment

3.1 Benchmark Datasets

The proposed VSLS is systematically evaluated across four benchmark datasets: a) LongVideoBench Ye et al. (2025a) for assessing long-context video-language comprehension capabilities; b) Video-MME Fu et al. (2024) as the first comprehensive benchmark for multimodal video analytics; c) Haystack-LVBench, extended from LongVideoBench with human-annotated frame index answers; and d) Haystack-Ego4D, derived from Ego4D with similar annotations. While LongVideoBench and Video-MME measure performance enhancement in QA accuracy, Haystack-Ego4D and Haystack-LVBench quantitatively evaluate keyframe selection accuracy through recall and precision metrics. Further details of datasets are provided in Appendix E.

3.2 Evaluation Metrics

3.2.1 Evaluation Metrics for Search Utility

Our assessment framework emphasizes both effectiveness and efficiency. For search effectiveness, we use three metrics to compare model-predicted keyframes with human annotations, considering both individual frames and full sets—addressing the possibility of multiple valid keyframe sets per query. For frame-level comparison, we evaluate the alignment between a predicted frame f_{pt} and a human-annotated frame f_{gt} from two perspectives:

Temporal coverage evaluates the coverage of ground truth frames by predicted frames in the temporal perspective, which can be described as:

$$T_{\text{cover}}(T_{\text{pt}}, T_{\text{gt}}) = \frac{\sum_{i=1}^{|N_{\text{gt}}|} \mathbb{I}\left[\min_j \left|t_{\text{gt}}^i - t_{\text{pt}}^j\right| \le \delta\right]}{|N_{\text{gt}}|}, \tag{9}$$

where T_{pt} and T_{gt} denote the sets of predicted and ground truth timestamps, respectively. Here, $|N_{\text{gt}}|$ is the number of ground truth frames, t_{gt}^i and t_{pt}^j are the i-th ground truth and j-th predicted timestamps, respectively. δ is the temporal similarity threshold defining the maximum allowed time deviation, and $\mathbb{I}[\cdot]$ is the indicator function, returning 1 if the condition holds and 0 otherwise.

Visual Similarity is measured by the Structural Similarity Index (SSIM) Brunet et al. (2012), capturing structural detail, luminance, and contrast between f_{pt} and f_{gt}. For set-to-set comparison, the key challenge is defining inter-set similarity. We adopt **Precision** P and **Recall** R as complementary metrics: Precision checks whether each predicted frame matches any reference frame, while Recall ensures that all reference frames are represented. Given the ground truth set $F_{\text{gt}} = f^j \text{gt}^n j = 1$ and the predicted set $F_{\text{pt}} = f^i \text{pt}^m i = 1$, we define the multimodal retrieval quality metrics as follows:

$$\begin{cases} P(F_{\text{pt}}, F_{\text{gt}}) = \dfrac{1}{|F_{\text{pt}}|} \displaystyle\sum_{f_{\text{pt}}^i \in F_{\text{pt}}} \max_{f_{\text{gt}}^j \in F_{\text{gt}}} \phi(f_{\text{pt}}^i, f_{\text{gt}}^j), & \text{(10a)} \\ R(F_{\text{pt}}, F_{\text{gt}}) = \dfrac{1}{|F_{\text{gt}}|} \displaystyle\sum_{f_{\text{gt}}^j \in F_{\text{gt}}} \max_{f_{\text{pt}}^i \in F_{\text{pt}}} \phi(f_{\text{gt}}^j, f_{\text{pt}}^i), & \text{(10b)} \end{cases}$$

where $\phi(\cdot, \cdot)$ represents an extensible multimodal similarity metric function.

3.2.2 Evaluation Metrics for Search efficiency

Existing studies Fan et al. (2024); Park et al. (2024); Wang et al. (2024b,e); Wu and Xie (2023) have mainly concentrated on optimizing task-specific performance metrics while neglecting computational efficiency in temporal search operations. To systematically analyze this dimension, our evaluation framework incorporates two criteria: 1) **FLOPs** representing arithmetic operation complexity, and 2) **Latency** recording real-world execution duration.

Method	Training Required	Searching Efficiency				Overall Task Efficiency	
		Matching	Iteration	TFLOPs ↓	Latency (sec) ↓	Latency (sec) ↓	Acc ↑
Static Frame Sampling							
UNIFORM-8 Ye et al. (2025a)	Training-Based	N/A	N/A	N/A	0.2	3.8	53.7
Dense Retrieval							
VIDEOAGENT Fan et al. (2024)	Training-Based	CLIP-1B Radford et al. (2021)	840	536.5	30.2	34.9	49.2
T*-RETRIEVAL Ye et al. (2025b)	Training-Based	YOLO-WORLD-110M	840	216.1	28.6	32.2	57.3
Temporal Search							
T*-ATTENTION Ye et al. (2025b)	Training-Based	N/A	N/A	88.9	13.7	17.3	59.3
T*-DETECTOR Ye et al. (2025b)	**Training-Free**	YOLO-WORLD-110M	43	31.7	7.3	11.1	59.8
VSLS (OURS)-DETECTOR	**Training-Free**	YOLO-WORLD-110M	49	33.3	7.8	11.6	**61.5**

Table 1: Evaluation of performance metrics across the LV-HAYSTACK benchmark, presenting both search efficiency and end-to-end processing overhead (combining search and inference stages).

3.3 Evaluation of Search Framework efficiency

Current approaches for keyframe selection can be broadly categorized into three paradigms: statistic-based frame sampling, dense feature retrieval-based selection, and temporal search-based methods. As shown in Table 1, while uniform sampling achieves the fastest processing speed, its ignorance of frame semantics severely limits downstream task effectiveness. Although dense feature retrieval methods attain moderate accuracy improvements (57.3%), their exhaustive frame processing demands 4.2× more TFLOPs and introduces 4.5× higher latency than our temporal search approach. Crucially, our method introduces four visual semantic logic detectors during temporal search while maintaining comparable execution time to T* methods. This strategic design elevates downstream task accuracy to 61.5%, achieving the best performance-efficiency trade-off.

3.4 Visual Semantic Logic Search Performance

As demonstrated in Table 2, we evaluate VSLS on LONGVIDEOBENCH from two critical perspectives: visual similarity (measured by precision and recall) and temporal coverage. Our method achieves state-of-the-art performance across all metrics. Specifically, under the 32-frame setting, VSLS attains a precision of 74.5% and recall of 92.5%, outperforming all baselines in visual similarity. More notably, the temporal coverage of VSLS reaches 41.4%, surpassing the second-best method (T* at 36.5%) by 13.4%—the largest margin among all comparisons. This significant improvement highlights the effectiveness of our visual semantic logic detection modules in identifying query-relevant keyframes with both semantic alignment and temporal completeness.

Table 2: Search utility results on LONGVIDEOBENCH. Best scores in the 8-frame setting are underlined, and in the 32-frame setting are **bold**. Gray indicates results from the original paper.

Method	Frame	LONGVIDEOBENCH		
		Precision ↑	Recall ↑	Time ↑
Static Frame Sampling Method				
UNIFORM Ye et al. (2025a)	8	56.0	72.0	6.3
UNIFORM	8	60.7	80.4	4.7
UNIFORM	32	58.7	81.6	24.9
UNIFORM	32	60.2	85.0	8.1
Dense Retrieval Method				
VIDEOAGENT Fan et al. (2024)	10.1	58.8	73.2	8.5
RETRIEVAL-BASED Ye et al. (2025b)	8	63.1	65.5	6.3
RETRIEVAL-BASED	32	59.9	80.8	21.8
Temporal Searching Method				
T* Ye et al. (2025b)	8	58.4	72.7	7.1
T*	8	75.3	88.2	26.2
VSLS **(ours)**	8	<u>75.6</u>	<u>88.6</u>	<u>26.3</u>
T*	32	58.3	83.2	28.2
T*	32	74.0	90.3	36.5
VSLS (ours)	32	**74.5**	**92.5**	**41.4**

These results empirically support our core hypothesis: leveraging semantic and logical cues from text queries enables precise detection of relevant video frames. Improvements in visual similarity and temporal coverage confirm that VSLS effectively captures keyframes while preserving temporal coherence through visual-logical alignment.

3.5 Downstream Video QA Performance

To demonstrate the advantages of VSLS, we evaluate downstream video QA performance on LONGVIDEOBENCH and VIDEO-MME. As shown in Table 3, videos are grouped by length into **Short**, **Medium**, and **Long** (15–3600s, up to 60 mins). VSLS consistently achieves the highest accuracy in the long-video category across different frame counts and QA models. Compared to the baseline T*, incorporating our visual semantic logic relations (Figure 1) yields substantial gains. These results confirm that modeling visual-logical relations is key to effective QA on long videos.

4 Analysis

4.1 Coverage Analysis of Semantic-Logical Relations

To ascertain the practical applicability and coverage of our defined semantic-logical relations (spatial, temporal, attribute, and causal), we conducted an analysis of their detection across all queries in the

LongVideoBench

Model and Size	Frame	Video Length: Long 900-3600s	Medium 180-600s	Short 15-60s
GPT-4o Hurst et al. (2024)	8	47.1	49.4	67.3
GPT-4o + T*	8	49.1	56.2	68.0
GPT-4o + VSLS (ours)	8	**51.2**	**58.9**	**74**
InternVL 2.5-78B Chen et al. (2024d)	8	55.7	57.3	74.0
InternVL 2.5-78B + VSLS (ours)	8	**58.0**	**61.5**	**74.0**
LLaVA-Video-7B-Qwen2 Zhang et al. (2024b)	8	42.0	46.5	50.0
LLaVA-Video-7B-Qwen2 + T*	8	39.6	**50.0**	48.0
LLaVA-Video-7B-Qwen2 + VSLS (ours)	8	**42.3**	46.9	**50.0**
Qwen2.5-VL-7B-Instruct Wang et al. (2024a)	8	41.0	43.1	**62.0**
Qwen2.5-VL-7B-Instruct + T*	8	42.0	47.7	54.0
Qwen2.5-VL-7B-Instruct + VSLS (ours)	8	**45.8**	**49.2**	54.0
GPT-4o	32	53.8	56.5	74.0
GPT-4o + T*	32	**55.3**	58.8	72.0
GPT-4o + VSLS (ours)	32	54.2	**60.0**	**76.0**
LLaVA-Video-7B-Qwen2	32	**42.3**	45.8	54.0
LLaVA-Video-7B-Qwen2 + T*	32	40.2	44.2	50.0
LLaVA-Video-7B-Qwen2 + VSLS (ours)	32	41.7	**48.1**	**54.0**
Qwen2.5-VL-7B-Instruct	32	32.7	36.5	50.0
Qwen2.5-VL-7B-Instruct + T*	32	38.7	41.9	40.0
Qwen2.5-VL-7B-Instruct + VSLS (ours)	32	**38.7**	**42.3**	**54.0**
LLaVA-OneVision-Qwen2-78B-ov	32	59.3	63.9	77.4
PLLaVA-34B	32	49.1	50.8	66.8
LLaVA-Video-78B-Qwen2	128	59.3	63.9	77.4
mPLUG-Owl3-7B	128	53.9	58.8	73.7
GPT-4o (0513)	256	61.6	66.7	76.8

Video-MME

Model and Size	Frame	Video Length: Long 30-60min	Medium 4-15min	Short 0-2min
GPT-4o	8	55.2	60.2	**69.6**
GPT-4o + T*	8	55.2	**61.2**	68.9
GPT-4o + VSLS (ours)	8	**56.9**	60.7	68.2
InternVL 2.5-78B	8	52.6	55.5	55.9
InternVL 2.5-78B + VSLS (ours)	8	**57.7**	**57.5**	**59.0**
LLaVA-Video-7B-Qwen2	8	38.0	39.7	38.2
LLaVA-Video-7B-Qwen2 + T*	8	37.5	**40.4**	37.2
LLaVA-Video-7B-Qwen2 + VSLS (ours)	8	**38.5**	38.5	**38.5**
Qwen2.5-VL-7B-Instruct	8	38.0	47.3	55.4
Qwen2.5-VL-7B-Instruct + T*	8	40.3	**50.0**	54.9
Qwen2.5-VL-7B-Instruct + VSLS (ours)	8	**43.2**	49.6	**60.8**
GPT-4o	32	55.2	61.0	71.4
GPT-4o + T*	32	55.2	61.6	72.6
GPT-4o + VSLS (ours)	32	**57.5**	**61.9**	**74.5**
LLaVA-Video-7B-Qwen2	32	35.9	36.5	37.4
LLaVA-Video-7B-Qwen2 + T*	32	35.8	**39.6**	37.8
LLaVA-Video-7B-Qwen2 + VSLS (ours)	32	**36.9**	39.0	**39.5**
Qwen2.5-VL-7B-Instruct	32	37.5	39.9	54.1
Qwen2.5-VL-7B-Instruct + T*	8	34.9	45.6	55.2
Qwen2.5-VL-7B-Instruct + VSLS (ours)	32	**37.9**	**50.0**	**55.8**
LLaVA-OneVision-78B	32	60.0	62.2	66.3
VideoLLaMA 2	32	57.6	59.9	62.4
Oryx-1.5	128	59.3	65.3	67.3
Aria-8x3.5B	256	58.8	67.0	67.6
Gemini-1.5-Pro (0615)	1/0.5 fps	67.4	74.3	75.0

Table 3: **Downstream task evaluation results on two benchmarks.** All accuracy scores (%) in black are from our replication. We also cite reported SOTA accuracy in gray (noting that their settings may differ and results may not be reproducible), along with the number of frames used for QA inference, for full transparency.

LongVideoBench and VideoMME datasets. Our findings reveal a crucial insight: for every question posed within these extensive VQA benchmarks, our query analysis module successfully identified and mapped the query to at least one of the four defined logical relation types. This empirical result supports the completeness of our proposed relation set for interpreting the semantic and logical intent inherent in these VQA tasks.

4.2 Time Complexity

The proposed framework consists of two stages. First, VLMs such as LLaVA-7B and GPT-4o extract a semantic set $\mathcal{S}$ from a video V with n frames. $\mathcal{S}$ includes target objects, cue objects, and their relations, with their size constrained by prompt design. In the second stage, keyframe identification is performed via a heuristic search: k candidates are iteratively selected using a scoring function $h(\cdot, \mathcal{S})$. The score distribution scores$[n]$ is dynamically refined using outputs from the YOLO-World detector.

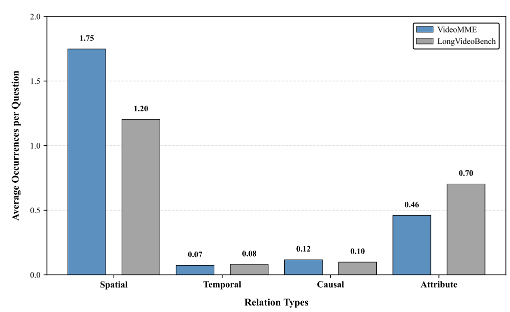

Figure 4: Average occurrences of detected semantic-logical relation types per question on VideoMME and LongVideoBench. Spatial relations are most frequent, while all queries in both datasets trigger at least one of the four relation types.

Our analysis focuses on YOLO-World detections, the main computational bottleneck due to their reliance on deep neural networks. Reducing the number of detections improves efficiency without sacrificing accuracy. At each iteration, the detector processes k selected frames to match objects and relations in $\mathcal{S}$, yielding k detections. The search stops when all targets are found or the iteration budget $\min(1000,\ 0.1 \times V_t)$ (with V_t as the video duration in seconds) is exhausted. In the worst case (e.g., videos with $>$10,000 frames and no matches), the cap is 1,000 iterations. Ideally, the evaluation function $h(\cdot, \mathcal{S})$ assigns high confidence to target frames, making the algorithm resemble top-k selection over n candidates in $\mathcal{O}(|\mathcal{S}| \log n)$ iterations Ye et al. (2025b), resulting in an average of $\mathcal{O}(|\mathcal{S}| k \log n)$ YOLO-World inferences.

Experimental results also demonstrate that integrating relational information into the search algorithm incurs negligible computational overhead compared to the baseline `T*` approach. On the LV-Haystack benchmark, the average iteration count increases from 42.94 (`T*`) to 48.82 iterations, representing a modest 13.69% rise in the time cost.

4.3 Ablation Study of Four Relations

Figure 4 illustrates the distribution of four logic relation types across LONGVIDEOBENCH and VIDEO-MME datasets, where *spatial* relations predominate, followed by *attribute* relations. In Table 4, we extract samples containing different relation types from LONGVIDEOBENCH to compare the object detection-based `T*` method with our `VSLS` approach. Experimental results demonstrate that `VSLS` achieves significant improvements across both image similarity metrics (SSIM Precision and SSIM Recall). Additionally, temporal coverage shows marked enhancement for *attribute*, *spatial*, and *causal* relations, with *spatial* relations exhibiting the most substantial improvement (21.3% increase over `T*`). For the *time* relation category, we observe a slight decrease in temporal coverage, which may be attributed to the relative scarcity of time relation samples in the dataset, limiting the opportunity to demonstrate the advantages of `VSLS`. Nevertheless, Figure 1 provides visual evidence of how effectively leveraging time relations can facilitate downstream question-answering tasks.

Logic Type	Method	LONGVIDEOBENCH		
		Precision ↑	Recall ↑	TC ↑
Spatial	`T*`	72.9	88.7	37.5
	`VSLS` **(ours)**	**73.6**	**91.4**	**45.5**
Attribute	`T*`	71.8	87.6	38.5
	`VSLS` **(ours)**	**72.7**	**90.9**	**42.1**
Time	`T*`	76.7	89.2	37.3
	`VSLS` **(ours)**	**77.5**	**92.5**	36.1
Casual	`T*`	74.7	92.4	38.6
	`VSLS` **(ours)**	74.7	**93.8**	**39.6**

Table 4: Comparison of our method (**VSLS**) with the baseline across four logic relation types on LONGVIDEOBENCH. **Precision**: SSIM Precision; **Recall**: SSIM Recall; **TC**: Temporal Coverage.

5 Related Work

Challenges in Long Video Understanding: Long video understanding is inherently more challenging than short-video or image-based tasks due to its rich temporal dynamics and massive redundancy Qian et al. (2024); Zeng et al. (2024); Yu et al. (2019). The large number of frames increases both memory and computational requirements, making straightforward dense sampling infeasible. Moreover, crucial events may span distant timestamps, demanding high-capacity models to capture long-range dependencies Ranasinghe et al. (2025); Shi et al. (2024); Chen et al. (2024b); Weng et al. (2024). Meanwhile, the diverse and continuous visual content raises noise and distractors; thus, strategies to effectively locate or distill essential parts of the video are of primary importance Zhang et al. (2023); Cheng et al. (2024b); Xu et al. (2023); Ye et al. (2025b).

Existing Solutions based on VLMs typically share three core ideas: 1) *video sampling or retrieval* for efficiency, 2) *multi-stage or interactive reasoning* to handle complex questions, and 3) *compact representation* to accommodate the VLM's limited context window. For instance, retrieval-based pipelines partition a video into segments and employ a learned or rule-based retriever to identify the relevant chunks before passing them to a VLM Pan et al. (2023); Choudhury et al. (2023, 2025). Other lines of research compress each frame into minimal tokens to reduce computational overhead Li et al. (2024); Chen et al. (2024a); Song et al. (2024), or adopt a streaming mechanism to propagate memory representations along the temporal axis Qian et al. (2024); Wu et al. (2022); Liu et al. (2024). Beyond these efficiency-oriented approaches, LLM/VLM-as-planner frameworks factorize the process into a series of perception queries, enabling an agent to fetch additional frame-level details if needed Wang et al. (2024c); Zhang et al. (2024a); Liao et al. (2024).

6 Conclusion

In this paper, we present `Visual Semantic-Logical Search` (`VSLS`), a novel framework that efficiently selects semantically keyframes for long video understanding by decomposing logical relationships between textual queries and visual elements. `VSLS` based on four defined logical dependencies (spatial co-occurrence, temporal proximity, attribute dependency, and causal order), significantly outperforms existing methods while sampling only 1.4% of video frames. The 8.7% improvement in GPT-4O's long video QA accuracy demonstrates that query-guided visual semantic logic search effectively bridges the gap between textual queries and visual content. `VSLS`'s plug-and-play nature enables seamless integration with existing pipelines, making it practical for real-world applications. Future work could consider more logical relations, learnable search methods, enhancing interpretability, and exploring more downstream tasks.

7 Acknowledgment

This work was supported in part by the National Key R & D Program of China (Grant No.2023YFF0725001), in part by the National Natural Science Foundation of China (Grant No.92370204), in part by the Guangdong Basic and Applied Basic Research Foundation (Grant No.2023B1515120057), in part by the Education Bureau of Guangzhou. Yijie Xu acknowledge the support from the modern matter laboratory, HKUST(GZ).

References

Dominique Brunet, Edward R. Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 2012.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Llavolta: Efficient multi-modal models via stage-wise visual context compression. In *arXiv preprint arXiv:2406.20092*, 2024a.

Jr-Jen Chen, Yu-Chien Liao, Hsi-Che Lin, Yu-Chu Yu, Yen-Chun Chen, and Yu-Chiang Frank Wang. ReXTime: A benchmark suite for reasoning-across-time in videos. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b.

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *NeurIPS*, 37:19472–19495, 2024c.

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

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *CVPR*, 2024a.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms, 2024b.

Rohan Choudhury, Koichiro Niinuma, Kris M Kitani, and László A Jeni. Zero-shot video question answering with procedural programs. *arXiv preprint arXiv:2312.00937*, 2023.

Rohan Choudhury, Koichiro Niinuma, Kris M. Kitani, and László A. Jeni. Video question answering with procedural programs. In *ECCV*, 2025.

David Cohen. Universals in linguistic theory, 1968.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. *ArXiv*, abs/2403.11481, 2024.

Charles J Fillmore. The case for case. *Bach and Harms (Ed.): Universals in Linguistic Theory*, 1967.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *ArXiv*, abs/2405.21075, 2024.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. Do you remember? dense video captioning with cross-modal memory retrieval. In *CVPR*, 2024.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2024.

Jianxin Liang, Xiaojun Meng, Yueqian Wang, Chang Liu, Qun Liu, and Dongyan Zhao. End-to-end video question answering with frame scoring mechanisms and adaptive sampling. *ArXiv*, abs/2407.15047, 2024.

Ruotong Liao, Max Erler, Huiyu Wang, Guangyao Zhai, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. Videoinsta: Zero-shot long video understanding via informative spatial-temporal reasoning with llms. In *EMNLP Findings*, 2024.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision*, pages 126–142. Springer, 2024.

William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

Leland Gerson Neuberg. Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory*, 19(4):675–685, 2003.

Junting Pan, Ziyi Lin, Yuying Ge, Xiatian Zhu, Renrui Zhang, Yi Wang, Yu Qiao, and Hongsheng Li. Retrieving-to-answer: Zero-shot video question answering with frozen large language models. In *ICCV Workshops*, 2023.

Jong Sung Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryoo, Donghyun Kim, and Michael S. Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. *ArXiv*, abs/2406.09396, 2024.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *NeurIPS*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Manjusha Rajan and Latha Parameswaran. Key frame extraction algorithm for surveillance videos using an evolutionary approach. *Scientific Reports*, 15(1):536, 2025.

Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S Ryoo. Understanding long videos with multimodal language models. In *ICLR*, 2025.

Yudi Shi, Shangzhe Di, Qirui Chen, and Weidi Xie. Unlocking video-llm via agent-of-thoughts distillation. *arXiv preprint arXiv:2412.01694*, 2024.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal llms. *arXiv preprint arXiv:2409.10994*, 2024.

John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2000.

Leonard Talmy. *Toward a Cognitive Semantics (Volume 1: Concept Structuring Systems; Volume 2: Typology and Process in Concept Structuring)*. MIT Press, Cambridge, MA, USA, 2000.

Reuben Tan, Ximeng Sun, Ping Hu, Jui hsien Wang, Hanieh Deilamsalehy, Bryan A. Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. *CVPR*, 2024.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. Video understanding with large language models: A survey. *arXiv preprint arXiv:2312.17432*, 2023.

Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *ECCV*, 2024b.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *ECCV*, pages 58–76. Springer, 2024c.

Zhanyu Wang, Longyue Wang, Zhen Zhao, Minghao Wu, Chenyang Lyu, Huayang Li, Deng Cai, Luping Zhou, Shuming Shi, and Zhaopeng Tu. Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation. In *ACM MM*, pages 3907–3916, 2024d.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *ArXiv*, abs/2405.19209, 2024e.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024f.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pages 453–470. Springer, 2024.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13587–13597, 2022.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *CVPR*, 2023.

Jiaqi Xu, Cuiling Lan, Wenxuan Xie, Xuejin Chen, and Yan Lu. Retrieval-based video language model for efficient long video question answering. *arXiv preprint arXiv:2312.04931*, 2023.

Jinhui Ye, Zihan Wang, and Haosen Sun. Longvideohaystack. `https://huggingface.co/datasets/LVHaystack/LongVideoHaystack`, 2025a. v1.0.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, Jiajun Wu, and Manling Li. Re-thinking temporal search for long-form video understanding. In *CVPR*, 2025b.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 2024.

Sicheng Yu, Chengkai Jin, Huan Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xioalei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. Frame-voyager: Learning to query frames for video large language models. *ArXiv*, abs/2410.03226, 2024.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. Timesuite: Improving mllms for long video understanding via grounded tuning, 2024.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023.

Lu Zhang, Tiancheng Zhao, Heting Ying, Yibo Ma, and Kyusong Lee. OmAgent: A multi-modal agent framework for complex video understanding with task divide-and-conquer. In *EMNLP*, 2024a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024b.

Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic evaluator for video mllms. *arXiv preprint arXiv:2406.09367*, 2024.

Heqing Zou, Tianze Luo, Guiyang Xie, Fengmao Lv, Guangcong Wang, Junyang Chen, Zhuochen Wang, Hansheng Zhang, Huaijian Zhang, et al. From seconds to hours: Reviewing multimodal large language models on comprehensive long video understanding. *arXiv preprint arXiv:2409.18938*, 2024.

Part I

Appendix

Table of Contents

A Theoretical Underpinnings of Relation Categories

Our choice of the four relation categories——*spatial, temporal, attribute, and causal*——is grounded in foundational concepts from linguistics and logic. While achieving absolute "completeness" in describing the infinite complexity of the real world is a formidable challenge, this selection aims to describe core aspects of events, states, and the way humans conceptualize and communicate them.

A.1 Linguistic Grounding

Semantic Roles and Case Grammar: Theories like Fillmore's Case Grammar Fillmore (1967) analyze sentences in terms of semantic roles that nominals play in relation to the verb (the event).

- **Spatial relations** directly correspond to roles like *Locative* (the location of an event or state) or *Path* (the trajectory of motion).
- **Temporal relations** align with *Temporal* roles, specifying when an event occurs or its duration.
- **Attributes** describe the properties of entities (participants) involved in these roles. While not direct case roles for verbs, they are fundamental for identifying and characterizing the "who" and "what" (e.g., Agent, Patient, Theme, Instrument) that possess these attributes during an event.
- **Causal relations** are central to understanding agency and event structure. Roles like *Agent* (the instigator of an action) or *Cause* (the non-volitional trigger of an event) highlight the importance of causality in linguistic descriptions of events.

Lexical Semantics and Event Structure: Works in lexical semantics (e.g., following Pustejovsky Cohen (1968) on the generative lexicon, or Talmy Talmy (2000) on cognitive semantics) often decompose event meaning into fundamental components. Talmy Talmy (2000), for instance, extensively discusses how language structures concepts like space, time, and force dynamics (which inherently relate to causality). Events are situated in space and time, involve entities with specific attributes, and are often linked through causal chains (e.g., one action causing another, or an agent causing a change of state).

Discourse Relations: Theories like Rhetorical Structure Theory (RST) Mann and Thompson (1988) identify relations that bind textual units together. Many of these fundamental relations are inherently temporal (e.g., *Sequence*), causal (e.g., *Cause, Result, Purpose*), or involve describing entities and their settings (which encompasses spatial and attributive information, often under relations like *Elaboration* or *Background*). This suggests that these four categories capture essential elements for constructing coherent descriptions and explanations, a core function of Video Question Answering (VQA).

A.2 Logical Grounding

Predicate Logic and Knowledge Representation: In formal logic and AI knowledge representation (e.g., Sowa Sowa (2000)), events and states are often represented using predicates with arguments that specify participants, locations, times, and properties. A typical event representation might implicitly or explicitly include `Location(event, place)`, `Time(event, time_interval)`, `HasProperty(entity, attribute_value)`, and relations like `Causes(event1, event2)`. Our four categories provide a high-level abstraction over these common predicate types.

Modal and Specialized Logics: • **Temporal Logic** is specifically designed to reason about propositions qualified in terms of time.

- **Spatial Logic** deals with reasoning about spatial properties and relations between entities.
- Logics of **Action and Causality** (e.g., situation calculus, event calculus, or Pearl's work on causality Neuberg (2003)) explicitly model how actions bring about changes and the causal dependencies between events.

A.3 Pragmatic Completeness for VQA

From a pragmatic standpoint, particularly for VQA, these four relations address the core "Wh-questions" humans often ask to understand a scene or event:

- **What/Who?** (Identifies objects/entities, often distinguished by their **attributes**)
- **Where?** (Answered by **spatial** relations)
- **When?** (Answered by **temporal** relations)

- **Why/How did it happen?** (Often answered by **causal** relations or a sequence of events linked temporally and spatially)

While more fine-grained relations (as in Action Genome) undoubtedly provide deeper semantic detail, our chosen set aims to provide a foundational, yet computationally manageable, framework for keyframe selection based on the most common semantic and logical inferences required for a broad range of video queries. They represent a level of abstraction that is both meaningful for human queries and feasible for current visual-language models to parse and verify.

In essence, these categories are not arbitrary but reflect fundamental dimensions along which events and states are structured, perceived, and communicated in language and reasoned about in logic. We believe they offer a robust and broadly applicable framework for the task at hand.

A.4 Accuracy of Extracting the Logical Relations

Evaluating the LLM's ability to accurately extract logical relations is a crucial point for validating our framework's reliability. To quantitatively address this, we conducted a verification study. We randomly sampled 500 query instances from each question category across the Video-MME and Long VideoBench datasets. We then used LLMs to perform the logical relation and object extraction as described in our paper. Subsequently, to ensure the quality of the results, we performed a rigorous manual audit of all extracted logical relations and objects. Our analysis yielded the following high accuracy rates for the extraction task:

GPT-4o: 92%
Qwen-VL 72B: 88%

These results demonstrate that state-of-the-art LLMs are highly proficient at this task, confirming that the logic extraction ability does not serve as a performance bottleneck for our framework at this stage. Due to committee guidelines, we are unable to provide a direct link to this data in the rebuttal. However, we are committed to transparency and will release the full set of our manually audited data and LLM outputs as part of our public code release.

A.5 Semantic Logics in Specialized Applications

The four logics we presented—spatial, temporal, attribute, and causal—are intended as a foundational set designed to cover a broad range of general queries. We see VSLS not as a system with a fixed set of logics, but as an extensible framework that can be adapted to various domains.

To derive new logics for different scenarios, we propose the following systematic, three-step approach:

- Domain Knowledge Elicitation: The first step is to collaborate with domain experts to identify the critical relationships and events they analyze. For a medical application like surgical video analysis, this would involve identifying key instrument-tissue interactions (e.g., 'cutting', 'suturing', 'retracting') which are far more specific than our general 'causal' or 'spatial' relations. This process translates expert knowledge into a set of target logical relations.
- Operationalization into Verifiable Rules: The conceptual relationship must then be translated into a computable, verifiable rule that the VSLS search can execute. This involves defining the specific visual and temporal evidence required. For example, a new logic like suturing could be operationalized as:
  - (i) Detecting a needle-holder instrument and suture material.
  - (ii) Observing the instrument in periodic contact with tissue edges.
  - (iii) Verifying that the tissue edges become approximated in subsequent frames.
- Modular Integration: Finally, this new, operationalized logic can be integrated as a new module into the VSLS framework. Our design allows new logic-checking functions to be added alongside the existing four. The query parser would be extended to recognize domain-specific keywords (e.g., "suture") and trigger the corresponding verification function during the iterative search process.

This process ensures that the VSLS framework can be effectively adapted to specialized fields, making it more general and powerful. We believe this discussion significantly strengthens our paper's contribution. We will add a section covering the framework's extensibility and these guidelines for deriving new logics to the final version of the paper. Thank you again for this valuable suggestion.

B Performance

Long-form video understanding presents unique challenges due to the complexity of temporal dynamics and cross-modal interactions in extended durations (900-3,600 seconds). Our comprehensive evaluation of the LVB-XL benchmark reveals significant performance gaps between existing approaches. While large-scale models like GPT-4o (32 frames) and InternVL 2.5-78B (16 frames) have demonstrated competence in short-video tasks, their direct application to long-form content (marked by circle sizes proportional to model parameters) yields suboptimal results (53.8% and 56.5% accuracy respectively).

Our `Visual Semantic-Logical Search` (`VSLS`) framework addresses these limitations. This advancement enables consistent performance improvements across different architecture scales, elevating GPT-4o to 54.2% (+0.4pp) and achieving a remarkable 62.4% (+5.9pp) for InternVL 2.5-78B on this benchmark. The comparative analysis further suggests that `VSLS`'s gains become particularly pronounced when processing longer visual sequences, highlighting its effectiveness in modeling extended temporal contexts.

C Hyperparameter Setting

While several parameters in our framework are manually set, they are designed to be interpretable and correspond to clear, logical trade-offs, allowing users to configure the system based on their specific needs for efficiency versus accuracy.

C.1 Stacking Multiple Frames

Our rationale for this design is that it offers a deliberate trade-off between search efficiency and downstream task accuracy. The grid size is a configurable parameter allowing users to balance these two competing factors based on their needs.The table below shows a supplementary experiment we conducted. It presents the required steps to complete the keyframe search and the performance of the downstream QA task of GPT4o. As demonstrated, increasing the grid size substantially decreases the search cost (from 770 steps down to as few as 48 steps). However, it simultaneously leads to a moderate reduction in performance (from 56.7 down to 52.7). Notably, increasing the number of images in grids leads to a decline in detector accuracy, which in turn increases the overall search cost.

Table 5: Effect of the number of images in grids on search cost and QA performance.

Num. of Images in Grids	Search Cost (steps)	QA Performance
1	770	56.7
4	160	55.5
8	48	53.5
12	730	53.2
16	860	52.7

Therefore, this stacking approach presents a clear efficiency-performance trade-off, offering users the flexibility to choose an optimal balance based on their specific requirements or resource constraints. Users prioritizing detection accuracy may prefer smaller grid sizes or individual frames, while those with limited computational resources or stringent runtime constraints may opt for larger grid configurations to achieve greater efficiency.

C.2 Confidence Threshold

For example, two key parameters are the confidence threshold and the number of images in grids. The confidence threshold allows a user to balance search cost against performance; as shown in our analysis, increasing the threshold from 0.3 to 0.8 improves performance from 50.2 to 56.4, while the required search cost increases from 18 to 162 steps. Similarly, the number of images in grids parameter controls the granularity of the search. A smaller grid size (e.g., 1) yields the highest performance (56.7) at a high search cost (770 steps), whereas a medium grid size (e.g., 8) minimizes

Table 6: Effect of confidence threshold on search cost and performance.

Confidence Threshold	Search Cost (steps)	Performance
0.3	18	50.2
0.5	42	54.1
0.7	49	55.3
0.8	162	56.4

the search cost to just 48 steps, offering a more efficient search. This demonstrates that the parameters offer predictable control over the search behavior rather than being arbitrary settings.

D Analysis of the Impact of Search Frame Count

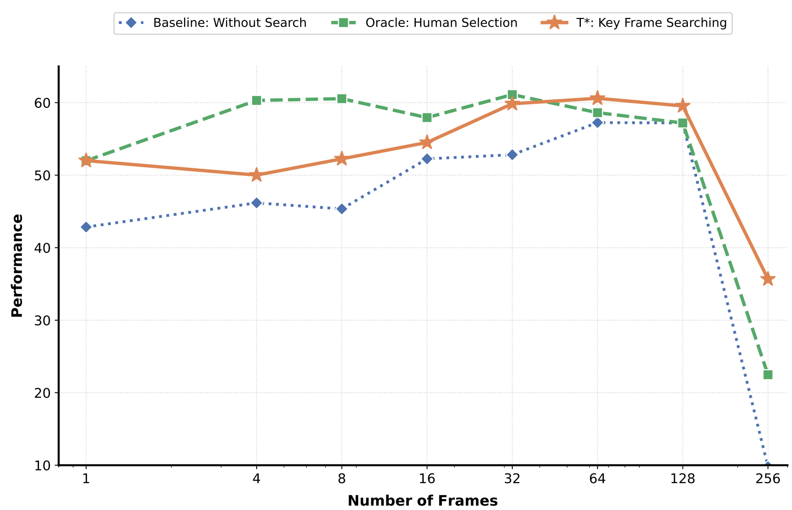

Figure 5: Performance improvement with increasing search frames. VSLS consistently enhances accuracy and reaches near-human oracle performance at 64 frames.

This section investigates the impact of the number of search frames on the performance of our Visual Language Models (VLMs) in the context of LongVideoBench.

Figure 5 in the T* framework study empirically demonstrates the non-monotonic relationship between input frame quantity and model accuracy on the LongVideoBench XL benchmark. Through systematic experimentation across 18 state-of-the-art VLMs, this visualization reveals a critical phenomenon: excessive frame inputs degrade performance for models lacking temporal redundancy mitigation mechanisms.

E Details of Datasets

E.1 Details of Video-MME

The Video-MME (Video Multi-Modal Evaluation) dataset represents the first comprehensive benchmark tailored to assess the capabilities of Vision-Language Models (VLMs) in video understanding. Aiming to address limitations in existing benchmarks, it emphasizes diversity, temporal complexity, and multi-modal integration while ensuring high-quality human annotations. The dataset contains 900 carefully curated videos across six primary domains—Knowledge, Film and Television, Sports Competition, Artistic Performance, Life Record, and Multilingual—with 30 fine-grained subcategories such as astronomy, esports, and documentaries. These videos vary significantly in duration, ranging from short clips (11 seconds) to long-form content (up to 1 hour), enabling robust evaluation across temporal scales.

Each video is paired with expert-annotated multiple-choice questions (2,700 QA pairs in total), rigorously validated to ensure clarity and reliance on visual or multi-modal context. Questions span 12 task types, including action recognition, temporal reasoning, and domain-specific knowledge, with a focus on scenarios where answers cannot be inferred from text alone. To quantify temporal complexity, the dataset introduces certificate length analysis, revealing that answering questions often requires understanding extended video segments (e.g., median lengths of 26 seconds for short videos and 890.7 seconds for long videos), surpassing the demands of prior benchmarks like EGOSCHEMA.

VIDEO-MME serves as a universal benchmark, applicable to both image- and video-focused MLLMs, and exposes key challenges for future research. These include improving architectures for long-sequence processing, developing datasets for complex temporal reasoning, and enhancing cross-modal alignment. By providing a rigorous evaluation framework, VIDEO-MME aims to drive progress toward MLLMs capable of understanding dynamic, real-world scenarios.

E.2 Details of LONGVIDEOBENCH

The LONGVIDEOBENCH benchmark pioneers the evaluation of long-context interleaved video-language understanding in VLMs, addressing critical gaps in existing benchmarks through its focus on detailed retrieval and temporal reasoning over hour-long multimodal inputs. Designed to overcome the "single-frame bias" prevalent in prior video benchmarks, the novel referring reasoning paradigm enables models to locate and analyze specific contexts within extended sequences. The data set comprises 3,763 web-sourced videos that span various themes - movies, news, life vlogs, and knowledge domains (including art, history, and STEM) - with durations progressively grouped into four levels: 8-15 seconds, 15-60 seconds, 3-10 minutes, and 15-60 minutes. Each video is paired with aligned subtitles, forming interleaved multimodal inputs that mimic real-world viewing scenarios.

The benchmark features 6,678 human-annotated multiple-choice questions categorized into 17 fine-grained task types across two levels: Perception (requiring object/attribute recognition in single scenes) and Relation (demanding temporal/causal reasoning across multiple scenes). Questions incorporate explicit referring queries (e.g., "When the woman descends the rocky hill...") that anchor reasoning to specific video moments, with an average question length of 43.5 words to ensure precision. Temporal complexity is quantified through duration-grouped analysis, where models must process up to 256 frames (at 1 fps) for hour-long videos, significantly exceeding the demands of predecessors like EGOSCHEMA (180s videos).

E.3 Details of LV-HAYSTACK

The LV-HAYSTACK benchmark establishes the first comprehensive evaluation framework for temporal search in long-form video understanding, addressing critical limitations in existing synthetic needle-in-haystack benchmarks through real-world video annotations and multi-dimensional evaluation metrics. Designed to assess models' ability to locate minimal keyframe sets (typically 1-5 frames) from hour-long videos containing tens of thousands of frames, the dataset comprises 3,874 human-annotated instances spanning 150 hours of video content across two distinct categories: egocentric videos from EGO4D (101 hours) and allocentric videos from LONGVIDEOBENCH (57.7 hours).

Organized into HAYSTACK-EGO4D and HAYSTACK-LVBENCH subsets, the benchmark features videos averaging 24.8 minutes in length (max 60 minutes) with 44,717 frames per video. Each instance contains:

- Expert-curated multi-choice questions requiring temporal reasoning (15.9 questions/video);
- Human-annotated keyframe sets (4.7 frames/question for egocentric, 1.8 frames/question for allocentric);
- Temporal and visual similarity metrics for precise search evaluation.

E.4 Details of EGO-4D

The EGO4D (Egocentric Computer Vision Benchmark) dataset establishes a transformative foundation for advancing research in first-person visual perception through unprecedented scale, diversity, and multi-modal integration. Designed to overcome limitations in existing egocentric datasets, it captures 3,670 hours of unscripted daily activities from 931 participants across 74 global locations and 9 countries, spanning household, workplace, leisure, and outdoor scenarios. The dataset features

30+ fine-grained activity categories including carpentry, social gaming, and meal preparation, with videos ranging from brief interactions (8-minute clips) to extended continuous recordings (up to 10 hours), enabling comprehensive analysis of long-term behavioral patterns.

Each video is enriched with multi-modal annotations totaling 3.85 million dense textual narrations (13.2 sentences/minute), coupled with 3D environment meshes, eye gaze tracking, stereo vision, and synchronized multi-camera views. Rigorous privacy protocols ensure ethical data collection, with 612 hours containing unblurred faces/audio for social interaction studies. The benchmark suite introduces five core tasks organized across temporal dimensions:

- **Episodic Memory**: Temporal localization of natural language queries (74K instances) and 3D object tracking using Matterport scans;
- **Hand-Object Interaction**: State change detection (1.3M annotations) with PNR (point-of-no-return) temporal localization;
- **Social Understanding**: Audio-visual diarisation (2,535h audio) and gaze-directed communication analysis;
- **Action Forecasting**: Anticipation of locomotion trajectories and object interactions.

Quantitative analysis reveals the dataset's complexity: hand-object interactions involve 1,772 unique verbs and 4,336 nouns, while social scenarios contain 6.8 participant interactions per minute on average. Multi-modal fusion experiments demonstrate performance gains, with 3D environment context improving object localization accuracy by 18.7% compared to RGB-only baselines. State-of-the-art models achieve 68.9% accuracy in action anticipation tasks, yet struggle with long-term forecasting (41.2% accuracy for 5s predictions), highlighting critical challenges in temporal reasoning.

EGO4D's unique integration of egocentric video with complementary modalities (IMU data in 836h, gaze tracking in 45h) enables novel research directions in embodied AI and augmented reality. The dataset exposes fundamental limitations in current architectures, particularly in processing hour-long video contexts and synthesizing cross-modal signals—only 23% of tested models effectively utilized audio-visual synchronization cues. By providing standardized evaluation protocols and curated challenge subsets, EGO4D serves as a universal testbed for developing perceptive systems capable of understanding persistent 3D environments and complex human behaviors.

F Detailed Algorithm

The detailed `VSLS` algorithm is represented in Algorithm 2.

F.1 Algorithm Overview and Core Components

The algorithm operates as an adaptive search framework that intelligently explores video content (represented as set V) to locate frames matching semantic-logical query requirements (Q). Unlike traditional linear search methods, it employs a probabilistic sampling strategy that dynamically adjusts based on confidence scores from multiple relationship types.

Initialization Phase The process begins by parsing the input query Q into two fundamental components:

- $\mathcal{O}$: A set of key objects or entities to identify
- $\mathcal{R}$: A collection of relationships (spatial, temporal, causal, and attribute) that must be satisfied

The algorithm initializes with a uniform probability distribution (P) across all video frames, establishing a budget (B) equivalent to the total number of frames ($|V|$), and creating an empty score registry (S) to track confidence values. This approach ensures unbiased initial exploration before evidence-guided refinement.

Adaptive Sampling Strategy Rather than exhaustively processing every frame, the algorithm employs a square-root scaling sampling strategy where $k = \lfloor\sqrt{B}\rfloor$ determines the sampling density. This provides a mathematical balance between exploration breadth and computational efficiency. The Grid function organizes sampled frames into a structured representation that preserves spatial-temporal relationships, facilitating subsequent relationship analysis.

Algorithm 2: The completed Visual Semantic-Logical Search

```
Function SemanticLogicalTemporalSearch(V, Q, K, Δ_t, τ, α, γ):
    O, R ← ParseQuestion(Q) ;              // Extract key/cue objects and relationships
    P ← Uniform, B ← |V|, S ← ∅, N_v ← |V| ;     // Initialize distribution and state
    while B > 0 and |O| > 0 do
        k ← ⌊√B⌋, G ← Grid(Sample(P, k^2)) ;           // Adaptive grid sampling
        Ω ← DetectObjects(G) ;                  // Detect objects in sampled frames
        foreach g ∈ G do
            C_g ← CalculateBaseScore(Ω[g]) ;           // Base detection confidence
            foreach r ∈ R do
                if r.type = Spatial then
                    C_g ← C_g + αγ_spatial · CheckSpatialRelationship(r, Ω[g])
                else if r.type = Temporal then
                    C_g ← C_g + αγ_time · CheckTemporalRelationship(r, Ω, Δ_t)
                else if r.type = Causal then
                    C_g ← C_g + αγ_causal · CheckCausalRelationship(r, Ω)
                else if r.type = Attribute then
                    C_g ← C_g + αγ_attr · CheckAttributeRelationship(r, Ω[g], τ)
            UpdateScores(S, g, C_g) ;                 // Update global score registry
        DiffuseScores(S, w) ;                         // Temporal context propagation
        P ← NormalizeDistribution(S), B ← B − k^2 ;   // Update sampling distribution
        foreach g ∈ TopK(S, K) do
            if Ω[g] ∩ O ≠ ∅ then
                O ← O \ Ω[g];                        // Remove identified key objects
    return TopK(S, K) ;                                // Return top-K keyframes
```

Multi-modal Object Detection The DetectObjects function applies state-of-the-art computer vision techniques to identify objects within each sampled frame. This step leverages deep neural networks pre-trained on diverse visual datasets, enabling recognition of a wide range of entities with their corresponding confidence scores and spatial locations within frames.

Score Propagation and Distribution Update The DiffuseScores function implements a temporal context propagation mechanism that spreads confidence values to neighboring frames, acknowledging that relevant content likely extends beyond individual frames. This diffusion creates a smoothed confidence landscape that guides subsequent sampling.

After each iteration, the algorithm normalizes the accumulated scores to form an updated probability distribution, focusing future sampling on promising regions while maintaining exploration potential in unexamined areas.

Convergence Criteria and Termination The search continues until either:

- The sampling budget (B) is exhausted, indicating comprehensive coverage of the video content
- All target objects ($\mathcal{O}$) have been successfully identified at satisfactory confidence levels

This dual-termination approach balances thoroughness with efficiency, preventing unnecessary computation once objectives are met.

Result Generation The algorithm concludes by returning the top-K frames with the highest confidence scores, representing the most relevant video segments that satisfy the semantic-logical query requirements. These keyframes provide a concise summary of the content matching the user's information needs.

F.2 Implementation Considerations

The algorithm's performance depends on several configurable parameters:

- Δ_t: Temporal window size for relationship analysis
- τ: Confidence threshold for attribute matching

- α: Global relationship influence factor
- γ: Type-specific relationship weights

These parameters can be tuned based on application requirements, video characteristics, and computational constraints. The algorithm's modular design allows for straightforward substitution of specific component implementations (e.g., different object detectors or relationship checkers) without altering the overall framework.

F.3 Computational Complexity Analysis

The time complexity scales with $O(\sqrt{N})$ where N is the total number of frames, significantly improving upon linear approaches. Space complexity remains $O(N)$ to maintain the probability distribution and score registry. The algorithm intelligently balances exploration and exploitation through its adaptive sampling approach, making it particularly suitable for large-scale video analysis tasks where exhaustive processing would be prohibitive.

F.4 Technical Implementation Details

Object Detection and Feature Extraction To achieve real-time performance, the object detection module utilizes pre-trained deep convolutional neural network architectures, particularly variants based on FAST R-CNN and YOLO series. The system employs a two-stage detection strategy:

- **Preliminary Detection**: Using lightweight models to rapidly identify potential regions;
- **Fine-grained Classification**: Applying more sophisticated models for detailed classification on high-confidence regions.

The feature extraction process leverages self-attention mechanisms from Visual Transformers (ViT), generating rich semantic embeddings robust to various visual variations such as scale, rotation, and illumination. Each identified object is associated with a feature vector $f_i \in \mathbb{R}^d$, where $d = 512$ represents the dimensionality of the embedding space.

Mathematical Formulations for Relationship Assessment The evaluation of various relationship types is based on precise mathematical definitions:

Spatial Relationships Given bounding boxes $B_i = (x_i, y_i, w_i, h_i)$ and $B_j = (x_j, y_j, w_j, h_j)$ for two objects, the confidence for a spatial relationship $r_{spatial}$ is calculated as:

$$C_{\text{spatial}}(B_i, B_j, r) = \phi_r(B_i, B_j) \cdot \psi(B_i) \cdot \psi(B_j), \tag{11}$$

where ϕ_r is a relationship-specific compatibility function and ψ is the object detection confidence. For example, the compatibility for a "contains" relationship is defined as:

$$\phi_{\text{contains}}(B_i, B_j) = \frac{\text{IoU}(B_i, B_j)}{\text{Area}(B_j)}. \tag{12}$$

Temporal Relationships Temporal relationships are calculated by evaluating object behavior patterns across a sequence of frames $\{F_t, F_{t+1}, ..., F_{t+\Delta_t}\}$:

$$C_{\text{temporal}}(O_i, O_j, r, \Delta_t) = \prod_{k=0}^{\Delta_t - 1} T_r(O_i^{t+k}, O_j^{t+k+1}), \tag{13}$$

where T_r is a relationship-specific temporal transition matrix and O_i^t represents the state of object i at time t.

Causal Relationships Causal relationships utilize a Bayesian network framework to compute conditional probabilities:

$$C_{\text{causal}}(E_i, E_j) = P(E_j|E_i) \cdot \log \frac{P(E_j|E_i)}{P(E_j)}, \tag{14}$$

where E_i and E_j represent the presumed cause event and effect event, respectively.

Attribute Relationships Attribute evaluation employs cosine similarity metrics between feature vectors and attribute prototypes:

$$C_{\text{attr}}(O_i, a) = \max(0, \cos(f_i, p_a) - \tau), \tag{15}$$

where p_a is the prototype vector for attribute a and τ is the minimum similarity threshold.

Score Propagation Algorithm Temporal score propagation is implemented through a weighted diffusion process, analogous to heat diffusion on a graph structure:

$$S'(t) = S(t) + \sum_{k \in \mathcal{N}(t)} w_{k,t} \cdot S(k), \tag{16}$$

where $\mathcal{N}(t)$ represents the temporal neighborhood of frame t, and $w_{k,t}$ is a weight based on temporal distance, defined as:

$$w_{k,t} = \exp\left(-\frac{|k-t|^2}{2\sigma^2}\right), \tag{17}$$

where σ controls the diffusion range.

Adaptive Sampling Optimization The sampling strategy is further improved through a dynamically adjusted Thompson sampling method, modeling the probability distribution P as a Beta distribution with shape parameters updated through previous observations:

$$P(t) \sim \text{Beta}(\alpha_t + \sum_i S_i(t), \beta_t + n - \sum_i S_i(t)), \tag{18}$$

where α_t and β_t are prior hyperparameters and n is the total number of observations.

F.5 Practical Application Examples

In practical visual search scenarios, the algorithm processes complex queries such as "*a person wearing a blue shirt sits down at a table and then picks up a coffee cup*":

- Query parsing identifies key objects (`person, shirt, table, coffee cup`) and relationships (`blue attribute, sitting action, temporal before-after relation, spatial proximity`);
- Adaptive sampling selects representative frames from the video;
- Multi-relationship evaluation integrates various sources of evidence;
- Score propagation establishes a unified confidence landscape across related frame sets;
- Result generation provides a concise summary of the most relevant segments in the video.

This semantic-logical-temporal search framework represents a significant advancement in multimodal content retrieval, enabling natural language queries that incorporate complex relationships across objects, time, and causal chains.

F.6 System Specifications for Reproductivity

Our experiments were conducted on high-performance servers, each equipped with either an Intel(R) Xeon(R) Platinum 8378A CPU @ 3.00GHz or an Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60GHz, 1TB of RAM, and 4/6 NVIDIA A800 GPUs with 80GB memory. Machines with 4 GPUs are configured with the SXM4 version, while those with 6 GPUs use the PCIe version. The software environment included *Python* 3.11, *PyTorch* 2.4, and *NCCL* 2.21.5 for reproductivity.

G Case Study of VSLS Keyframe Selection

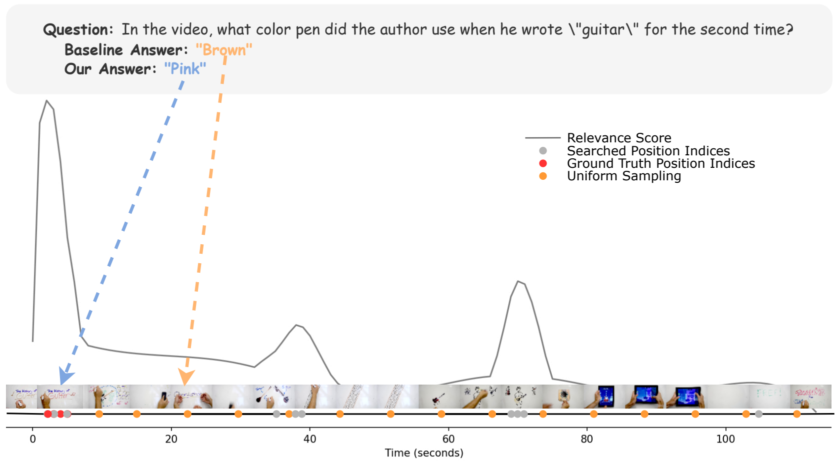

Figure 6: Qualitative comparison of frame selection strategies demonstrates VSLS's ability to pinpoint query-critical moments (e.g., the subject presenting pink objects) with temporal precision, while baseline approaches exhibit color misinterpretation (brown) due to suboptimal frame choices. VSLS maintains superior temporal diversity and content relevance, effectively avoiding the redundant selections observed in comparative methods.

As shown in Figure 6, the VSLS framework demonstrates its effectiveness through a video question-answering case study involving temporal handwriting analysis. The experiment focuses on distinguishing between two sequential events: a brown pen writing "guitar" at 2 seconds and a pink pen rewriting the same word at 3 seconds, with the query requiring identification of the second occurrence's pen color.

VSLS's analytical process unfolds through three interpretable phases:

- **Semantic Logic Extraction**: Identifies core visual entities (*handwritten text*, *pen*, *paper*) and constructs temporal relationships through triplet formulation: (*text*, *time*, *pen*), establishing the framework for tracking writing instrument changes;
- **Temporal Relevance Scoring**: The gray relevance curve reveals precise temporal localization, with peak scores aligning perfectly with ground truth positions at 2s and 3s, contrasting sharply with baseline methods' random fluctuations;
- **Search Pattern Visualization**: Demonstrates VSLS's focused inspection near critical moments versus uniform sampling's scattered temporal coverage, explaining the baseline's failure to detect the pink pen.

This case study yields two critical insights about VSLS's temporal reasoning:

- **Sequential Event Disambiguation**: The system successfully differentiates between near-identical visual events through:
  - First writing instance: Brown pen detection(false positive);
  - Second writing instance: Pink pen detection(true positive).
- **Explanation of answer generation disparity**: VSLS produces the correct answer ("*Pink*") versus uniform sampling's erroneous baseline ("*Brown*") due to temporal reasoning failures.

The spatial-temporal alignment between relevance peaks and ground truth positions confirms VSLS's unique capacity to synchronize semantic logic with visual evidence flow. This case particularly highlights the method's superiority in scenarios requiring precise discrimination of recurrent events with subtle visual variations.

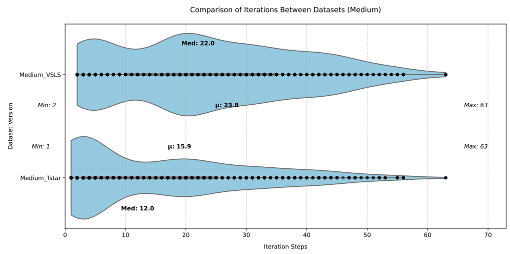

Figure 7: The comparative visualization of iteration counts on the medium-length video subset of the VIDEO-MME dataset demonstrates that our method consistently requires a higher number of iterations compared to the T* approach.

H Iteration Analysis

As shown in Fig 7, incorporating relations into the search algorithm will increase the average number of iterations for the video of medium length in the VIDEO-MME dataset from 15.9 to 23.8. The overall distribution of video iteration will not be significantly changed.

I Prompt

I.1 Prompt Template for Query Grounding

Here is the prompt we used for query grounding.

Prompt Template for Query Grounding

Analyze the following video frames and the question:
Question: `<Question>`
Options: `<Options>`
Step 1: Key Object Identification
- Extract 3-5 core objects detectable by computer vision
- Use YOLO-compatible noun phrases (e.g., "person", "mic")
- Format: Key Objects: `obj1`, `obj2`, `obj3`

Step 2: Contextual Cues
- List 2-4 scene elements that help locate key objects based on options provided
- Use detectable items (avoid abstract concepts)
- Format: Cue Objects: `cue1`, `cue2`, `cue3`

Step 3: Relationship Triplets
- Relationship types:
  - Spatial: Objects must appear in the same frame
  - Attribute: Color/size/material descriptions (e.g., "red clothes", "large")
  - Time: Appear in different frames within a few seconds
  - Causal: There is a temporal order between the objects
- Format of Relations: `(object, relation_type, object)`, `relation_type` should be exactly one of spatial/attribute/time/causal

Output Rules
1. One line each for Key Objects/Cue Objects/Rel starting with exact prefixes
2. Separate items with comma except for triplets where items are separated by semicolon
3. Never use markdown or natural language explanations
4. If you cannot identify any key objects or cue objects from the video provided, please just identify the possible key or cue objects from the question and options provided

Below is an example of the procedure:
Question: For "When does the person in red clothes appear with the dog?"
Response:
Key Objects: `person, dog, red clothes`
Cue Objects: `grassy_area, leash, fence`
Rel: `(person; attribute; red clothes)`, `(person; spatial; dog)`

Format your response EXACTLY like this in three lines:
Key Objects: `object1, object2, object`
Cue Objects: `object1, object2, object`
Rel: `(object1; relation_type1; object2), (object3; relation_type2; object4)`

I.2 Prompt Template for Question Answering

Here is the prompt we used for question answering.

Prompt Template for Question Answering

Select the best answer to the following multiple-choice question based on the video.
`<image>`
`<image>`
`...`
Question: `<Question>`
Options: `<Options>`
Answer with the option's letter from the given choices directly.
Your response format should be strictly an upper case letter A,B,C,D or E.

J Limitations

Despite the promising results of our VSLS framework, we acknowledge several limitations: First, although our approach reduces the required frame sampling to just 1.4%, the computational complexity remains a consideration for extremely long videos, with a search overhead of approximately 7.8 seconds. This may present challenges for real-time or low-latency applications. Besides, the performance of VSLS is bounded by the capabilities of the underlying object detector (YOLO-WORLD). Detection accuracy may degrade under challenging visual conditions such as poor lighting, occlusion, or unusual camera angles, potentially affecting temporal coverage.

Moreover, relying solely on bounding box overlap for attribute association is a heuristic with logical limitations. The primary role of our Visual Semantic-Logical Search (VSLS) is to function as a highly efficient candidate retrieval mechanism, not as the final reasoning engine. The goal of this heuristic is to rapidly filter a vast number of frames down to a small, manageable set that is highly likely to contain the answer. We then delegate the more complex task of logical disambiguation to a powerful downstream Vision-Language Model (VLM). For instance, given the query "What color is the shirt of the person in the car?", our search might retrieve keyframes of both a person sitting in the car and a person who happens to be walking past the car. Both sets of frames are passed to the VLM, which has the foundational reasoning capability to distinguish the correct context and provide the right answer.

Our method is therefore distinct from, and complementary to, graph-based frame relation modeling. VSLS is designed as an efficient, query-guided temporal search module that serves as a crucial preprocessing step. A more complex method, such as graph-based reasoning, would be a downstream task that operates on the concise set of keyframes our method selects. This two-stage approach allows for both efficiency over long videos and robust reasoning on the retrieved candidates.

K Broader Impacts

Our Visual Semantic-Logical Search (VSLS) framework primarily offers positive societal impacts as a foundational algorithm for efficient keyframe selection in long videos.

K.1 Positive Impacts

- **Educational Applications:** VSLS enables students and educators to quickly locate relevant segments in instructional videos, improving learning efficiency for visual content.
- **Research Enhancement:** Scientists across disciplines can benefit from more efficient analysis of video archives, particularly those studying behavioral patterns or analyzing historical footage.
- **Computational Efficiency:** By sampling only 1.4% of frames on average, our approach reduces computational requirements and energy consumption, contributing to more sustainable AI applications.
- **Accessibility:** Our framework can be integrated into assistive technologies for individuals with cognitive processing challenges, helping them identify and focus on critical moments in video content.

K.2 Potential Considerations

As a foundational algorithm, VSLS has limited direct negative impacts. However, like any computer vision technology, applications built upon it should be mindful of general considerations:

- **Underlying Model Biases:** The performance of VSLS depends partly on object detection systems (e.g., YOLO-World), so it inherits any limitations or biases present in these components. Our modular design allows for substitution with improved detection systems as they become available.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions of our work, including (1) the proposal of a semantics-driven keyframe search framework using four logical relations, (2) performance gains on multiple long video QA benchmarks, (3) efficient frame sampling (1.4%) with state-of-the-art results, and (4) plug-and-play compatibility with VLM/LLM pipelines. These claims are supported by both the method and experimental sections (see Sections "Introduction", "Method", and "Experiment"), and limitations are discussed in the main paper and Appendix J. The claims are fully aligned with the presented theoretical and empirical results.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Appendix J.

   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include formal theoretical results, theorems, or proofs. Our work is primarily methodological and experimental; all mathematical formulations are used to describe the algorithm and its components, but no formal theorems are claimed or proved. Therefore, this item is not applicable.

   Guidelines:
   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides comprehensive details required for reproducibility, including descriptions of all datasets used (see Section "Details of Datasets" and Appendix E), implementation details of the proposed algorithm (see "Method" and "Algorithm Overview"), hyperparameter choices, prompt templates (Appendix "Prompt"), and evaluation protocols for each experiment. We also specify the object detection models and baselines used, and state that the code will be publicly released. This level of detail allows other researchers to replicate the main experiments and validate our claims.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We state in the abstract and main text that the code will be publicly released. All datasets used in our experiments are from public benchmarks (LONGVIDEOBENCH, VIDEO-MME, HAYSTACK-LVBENCH, EGO4D), and details for data access are provided in Appendix E. Instructions for running our framework, data preparation, and experiment replication will be included in the released code repository. Thus, researchers will be able to access both code and data with clear instructions for full reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all relevant experimental details, including descriptions of dataset splits, hyperparameters, evaluation metrics, and prompt templates (see "Experiment," Table captions, and Appendix E). As our method is training-free, we clarify in the main text which components rely on pre-trained models and explicitly describe all parameter settings for reproducibility. This ensures that readers can fully understand and interpret the reported results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not report error bars or formal statistical significance tests for the main experimental results, as our approach is deterministic and uses fixed dataset splits and pre-trained models. Metrics are reported as single values following common practice in recent long video QA benchmarks. While this is standard in the area, we acknowledge that including error bars or additional significance analysis would further strengthen the experimental evaluation.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the computing environment in Appendix F.6, and reports both latency and FLOPs for major baselines and our method in Table 1. We also provide the number of iterations, average processing time, and model sizes in the main text and tables. This information is sufficient for others to estimate compute requirements and reproduce the experiments.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research follows the NeurIPS Code of Ethics. All datasets used are publicly available, appropriately licensed, and include human annotation with proper privacy safeguards (see Appendix E). No personally identifiable information or sensitive data is used. The proposed methods and experiments present no foreseeable risk of harm, discrimination, or privacy violation. Anonymity is preserved in all supplementary materials.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our paper discusses broader impacts in Appendix K.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work introduces a semantic-logical search framework for keyframe selection that builds upon existing object detection models and benchmarks. It does not release new datasets scraped from the internet or high-risk generative models. While our method improves video understanding capabilities, it doesn't introduce fundamentally new capabilities that would require specific safeguards beyond those already in place for the underlying technologies (such as YOLO-World) that we utilize.

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: Our work introduces a semantic-logical search framework for keyframe selection that builds upon existing object detection models and benchmarks. It does not release new datasets scraped from the internet or high-risk generative models. While our method improves video understanding capabilities, it doesn't introduce fundamentally new capabilities that would require specific safeguards beyond those already in place for the underlying technologies (such as YOLO-World) that we utilize.

    Guidelines:
    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release code for our `VSLS` framework upon publication, as mentioned in the abstract. The code will be accompanied by comprehensive documentation detailing the implementation of our four logical dependencies (spatial, temporal, attribute, and causal), the iterative refinement process, and instructions for reproducing our experimental results. Our paper does not introduce new datasets but rather evaluates our method on existing benchmarks including LONGVIDEOBENCH, VIDEO-MME, and HAYSTACK-LVBENCH, which are properly cited throughout the paper.

    Guidelines:
    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve crowdsourcing or human subject experiments. We evaluate our method using existing benchmarks (LONGVIDEOBENCH, VIDEO-MME, LONGVIDEOBENCH) that contain human-annotated ground truth data, but we did not collect new human annotations or conduct human evaluations as part of our work. Our methodology is purely algorithmic, focusing on the semantic-logical frameworks for keyframe selection and evaluation through computational metrics.

    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve human subjects. We utilize existing benchmark datasets (LONGVIDEOBENCH, VIDEO-MME, HAYSTACK-LVBENCH) without collecting new data from human participants. Our work focuses on developing and evaluating algorithmic approaches for keyframe selection based on semantic-logical relationships, which do not require IRB approval or equivalent ethical review processes.

    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our `Visual Semantic-Logical Search` framework uses LLMs (specifically mentioned in Section 3.2 and Figure 2) as part of our query decomposition process. We employ models such as LLAVA-7B and GPT-4O to extract semantic information from textual queries, including key objects, cue objects, and their logical relationships. This LLM-based decomposition is an integral component of our method, as it enables the identification of the four logical relation types (spatial, temporal, attribute, and causal) that guide our keyframe selection process. The prompt template for this query grounding is provided in Appendix I.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.