Re-thinking Temporal Search for Long-Form Video Understanding

Jinhui Ye[1*†], Zihan Wang[2*], Haosen Sun[2], Keshigeyan Chandrasegaran[1], Zane Durante[1], Cristobal Eyzaguirre[1], Yonatan Bisk[3], Juan Carlos Niebles[1], Ehsan Adeli[1], Li Fei-Fei[1], Jiajun Wu[1], Manling Li[1,2]

[1]Stanford University [2]Northwestern University [3]Carnegie Mellon University

https://longvideohaystack.github.io

Data Code Live Demo Video Results

Abstract

Efficiently understanding long-form videos remains a significant challenge in computer vision. In this work, we revisit temporal search paradigms for long-form video understanding and address a fundamental issue pertaining to all state-of-the-art (SOTA) long-context vision-language models (VLMs). Our contributions are twofold: ***First****, we frame temporal search as a* ***Long Video Haystack*** *problem – finding a minimal set of relevant frames (e.g., one to five) from tens of thousands based on specific queries. Upon this formulation, we introduce* **LV-HAYSTACK***, the first dataset with 480 hours of videos,* 15,092 *human-annotated instances for both training and evaluation aiming to improve temporal search quality and efficiency. Results on* LV-HAYSTACK *highlight a significant research gap in temporal search capabilities, with current SOTA search methods only achieving* 2.1% *temporal* F_1 *score on the* LONGVIDEOBENCH *subset.*

Next*, inspired by visual search in images, we propose a lightweight temporal search framework,* ***T**** *that reframes costly temporal search as spatial search. T* leverages powerful visual localization techniques commonly used in images and introduces an adaptive zooming-in mechanism that operates across both temporal and spatial dimensions. Extensive experiments show that integrating T* with existing methods significantly improves SOTA long-form video understanding. Under an inference budget of 32 frames,* ***T**** *improves GPT-4o's performance from 50.5% to* ***53.1%*** *and LLaVA-OneVision-OV-72B's performance from 56.5% to* ***62.4%*** *on the* LONGVIDEOBENCH *XL subset. Our code, benchmark, and models are provided in the Supplementary material.*

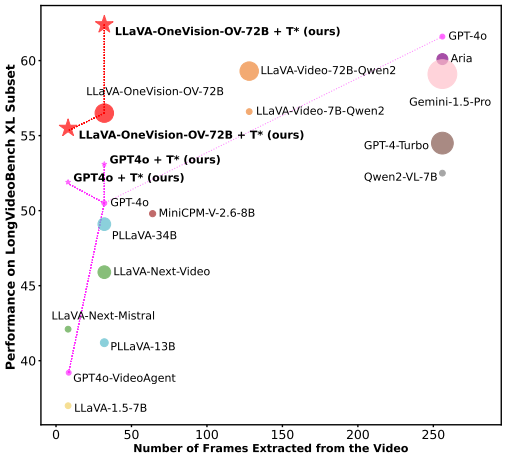

Figure 1. **Long-form video understanding performance comparison on LongVideoBench [72] XL subset (900-3600s)**. Open-sourced model size is indicated by marker size. Our lightweight temporal search algorithm *T** (§3) improve SOTA models significantly: GPT-4o (50.5% → **53.1%** and LLaVA-OneVision-OV-72B (56.5% → **62.4%**), both with 32 frames.

1. Introduction

As video understanding research expands from seconds-long to hour-long videos, [6, 18, 72], video understanding tasks face fundamental challenges in quickly and accurately locating relevant frames in long-form videos [12, 26, 36]. Current large vision-language models (VLMs) often require a large number of tokens for frame processing, e.g., 576 tokens per image for LLaVA [41] and Tarsier [64]. This makes frame-by-frame analysis of long videos, which contain thousands of frames, computationally challenging to all state-of-the-art VLMs. To overcome this challenge, temporal search [71, 76] has emerged as a fundamental paradigm, which is

*Equal contribution.
†Work done during internship at Stanford.

framed as a **Long Video Needle-in-a-Haystack** [34, 35]: locating a minimal set of frames (needles) within thousands of frames from a long video (haystack) which is essential to answer the question. Unlike traditional temporal localization [1, 17, 27, 55, 61, 78, 80, 82, 87, 91] which identifies continuous temporal segments, temporal search focuses on selecting relevant frames across the entire video.

To this end, we introduce benchmark **LV-HAYSTACK** specifically designed for temporal search on real-world long-form video. Unlike needle-in-a-haystack benchmarks [4, 25, 45, 46, 63, 66, 89] using randomly inserted synthetic frames as "needles", LV-HAYSTACK is built from real-world scenarios where humans answer questions by identifying few essential frames. We compile LV-HAYSTACK using videos and questions from Ego4D (Egocentric videos, Grauman et al. [18]) and LongVideoBench (Allocentric videos, Wu et al. [72]), ensuring each question has an answer and a set of keyframes. For LongVideoBench, we use keyframes and answers from the original dataset. For Ego4D, we annotate **15,092** QA instances from 988 videos spanning 423 hours with **45.7 million** frames, where each video lasts around 25 minutes with about 15 questions. Furthermore, previous long-form video evaluations [32, 53, 65, 72, 93] primarily focus on task performance and overlook the evaluation of temporal search capabilities. We propose frame-centered temporal and visual metrics and derive frame-set similarity metrics like temporal and visual F_1 to compare model-selected and reference keyframes to evaluate search capabilities.

Building upon the proposed benchmark, we examine the fundamental nature of temporal search in VLMs. Existing cluster- [22, 49, 56, 70, 86] or agent-based [14, 31, 68, 75, 83] methods rely on costly frame-by-frame processing with VLM to identify keyframes. We draw inspiration from *visual* search techniques like *V** [73], which effectively conduct spatial search in Vision Transformers [44] in a coarse-to-fine manner, suggesting that *temporal* search could be performed similarly. To unify temporal and spatial dimensions for video temporal search, we leverage the superior performance of image-language models over video-language models [20], effectively recasting temporal search as a spatial search task.

Specifically, we propose ***T****, a temporal search framework reframed as a spatial search task by transforming frame sequences into a single large image, gradually refining temporal resolution by discarding irrelevant frames and inserting frames around key temporal regions. Acting like an agent, *T** dynamically balances the spatial-temporal trade-offs by determining what spatial details to sacrifice, enhancing the temporal sampling probability in the promising time regions, and zooming-in images to achieve higher spatial resolution and recover details. This approach reduces search costs through multi-step zooming-in refinement, enabling more efficient and effective temporal search. With this unified approach, *T** seamlessly integrates both temporal and spatial dimensions within a single image space to achieve efficient long-form video understanding.

Subset	HAYSTACK-EGO4D	HAYSTACK-LVBENCH
Video Type	Egocentric	Allocentric
# video	988	114
# length	423 h - 25.7 min per video	26.7 h - 14.1 min per video
# frame	45,700,000 - 46,300 per video	2,200,000 - 19,100 per video
# QA pair	15,092 - 15.3 per video	342 - 3.0 per video
# keyframe	28,300 - 1.9 per question	496 - 1.5 per question

Table 1. Data Statistics of LV-HAYSTACK.

Empirically, the iterative sampling-scoring-reweighting paradigm of *T** results in 3x computational efficiency in terms of FLOPs compared to frame-by-frame search. On long-form video understanding tasks, *T** applied GPT-4o [47] and LLaVA-Onevision-OV-72B [28] achieves compatible performance while using 4x fewer frames, outperforming pervious search and non-search methods. Furthermore, our fine-grained evaluation framework provides interpretable metrics for different components of video understanding, and our findings reveal that temporal search capabilities closely aligns with downstream performance.

2. Temporal Search in Video Understanding

To explore efficient temporal search with long-context VLMs, we formulate Temporal Search (TS) similar to the needle-in-a-haystack [63] task, i.e., selecting few keyframes from the video to answer questions, which is critical for VLMs in processing long videos [37, 49, 59, 67, 70, 84].

2.1. Task Formulation

Given a video $V = \{f_1, f_2, ..., f_N\}$ with N frames and a question Q, temporal search tries to find a minimal subset of k keyframes $V^K = \{f_1^K, f_2^K, \ldots, f_k^K\} \subseteq V$ that contains all critical information required to answer Q. Specifically, the identified keyframe set requires two features:

- **Completeness**: V^K should be a complete frame set to answer questions. If the answer to Q based on V is A, then the answer derived from V^K should also be A.
- **Minimality**: V^K should contain only essential frames, with no redundant or irrelevant frames while maintaining completeness.

2.2. The LV-HAYSTACK Benchmark

Based on this task formulation, we construct a benchmark specifically designed for Temporal Search. Each search instance in our dataset is represented as a tuple comprising four elements $\langle V, Q, V^K, A \rangle$, with a video $V = \{f_i\}_{i=0}^{N}$, a question Q, annotated keyframes $V^K = \{f_j^K\}_{j=0}^{k}$ and

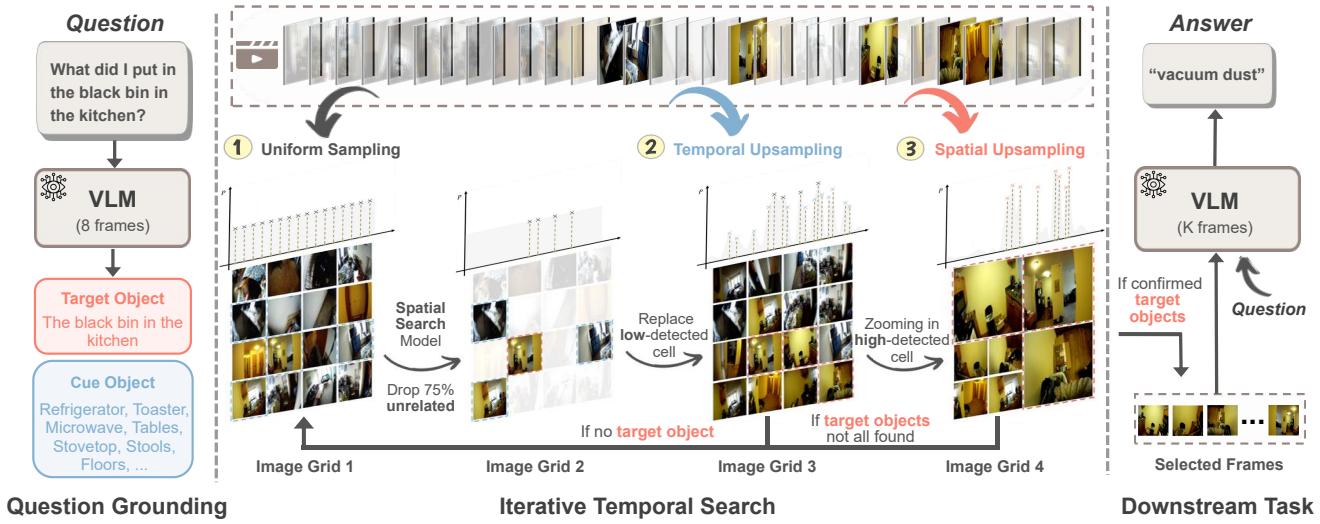

Figure 2. **The *T** framework that employs efficient temporal search for long-form video understanding.** *T** employs an iterative temporal search approach to search keyframes essential to answer questions. Left: Question Grounding, where a visual language model identifies visual cues (target and cue object) from the textual question. Center: Iterative Temporal Search, formulated as Spatial Search where a spatial search model iteratively detects visual cues and upsamples relevant temporal/visual regions. Right: Downstream Task, where the visual language model answer questions using K keyframes sampled from the final temporal search distribution as visual input.

the answer A. Our benchmark consists of both egocentric and allocentric videos, sourced from Ego4D [18] and LongVideoBench [72], respectively. For HAYSTACK-EGO4D, we select video segments from the Ego4D NLQ validation set, with an average duration of 8.3 minutes per segment. These segments capture diverse scenarios such as object finding and shopping activities. We hire crowdworkers to identify the minimal set of keyframes required to answer task-specific questions and provide corresponding answers. For HAYSTACK-LVBENCH, we repurpose the LongVideoBench dataset for the temporal search task, where annotators verify and refine the original reference timestamps to ensure the frames contain minimal sufficient information to answer each question. Statistics of our dataset can be found in Table 1 and more data annotation details are listed in the Appendix E.

2.3. Evaluation Metrics for Search Utility

Our evaluation framework focuses on both search utility and efficiency. For search utility, we develop metrics comparing model-predicted keyframes with human annotations at both frame and set levels, addressing the challenge that multiple valid keyframe sets may exist for the same question.

Frame-to-Frame Metrics. To evaluate alignment between a model-predicted frame f_{pt} and a human-annotated frame f_{gt}, we consider two dimensions. **1) Temporal Similarity** measures the timestamp difference between f_{pt} and f_{gt}, using a binary threshold to mitigate outlier effects. Two frames are considered similar if their temporal difference falls within this threshold. **2) Visual Similarity** adopts the Structural Similarity Index Measure (SSIM) [5] to identify the visual similarity between the frames f_{pt} and f_{gt} based on structural details, luminance, and contrast.

Set-to-Set Metrics. The major challenge in extending frame-to-frame metrics to frame set evaluation is defining what makes two sets *similar*. We introduce **Precision** and **Recall** as two complementary metrics. Precision measures whether each model-selected frame aligns with at least one reference frame, while Recall evaluates whether reference frames are represented in the model's selection.

Let $F_{\text{gt}} = \{f_{\text{gt}}^j\}_{j=1}^N$ denote the reference frame set and $F_{\text{pt}} = \{f_{\text{pt}}^i\}_{i=1}^M$ represent the model-predicted frame set. We define precision and recall as follows:

$$\text{Precision}(F_{\text{pt}}, F_{\text{gt}}) = \frac{1}{|F_{\text{pt}}|} \sum_{f_{\text{pt}}^i \in F_{\text{pt}}} \text{sim}(f_{\text{pt}}^i, F_{\text{gt}}), \quad (1)$$

$$\text{Recall}(F_{\text{pt}}, F_{\text{gt}}) = \frac{1}{|F_{\text{gt}}|} \sum_{f_{\text{gt}}^j \in F_{\text{gt}}} \text{sim}(f_{\text{gt}}^j, F_{\text{pt}}), \quad (2)$$

where $\text{sim}(f^i, F') = \max_{f^j \in F'} \text{sim}(f^i, f^j)$ defines the frame-to-set similarity for any frame and set. The sim function can measure either temporal or visual similarity. To balance search relevance (Precision) and coverage (Recall), we compute the F_1 score as the harmonic mean of them.

Algorithm 1: Efficient Temporal Search with Dynamic Sampling

Input : Video V, target/cue objects $\{T, C\}$, keyframe count K, search budget B, threshold θ
Output : Keyframes F with timestamps τ

1 Initialize: $S, N \leftarrow \mathbf{0}^L, \mathbf{1}^L$, $P \leftarrow \frac{1}{L}\mathbf{1}^L$, $R \leftarrow T$, $F, \tau \leftarrow \emptyset$; // $L = |V|$
2 **while** $R \neq \emptyset$ ***and*** $B > 0$ **do**
3 $I \leftarrow \text{Sample}(P \odot N, g^2)$, $G \leftarrow \text{Grid}(V[I])$, $B \leftarrow B - g^2$; // Sample and grid
4 $(C, O) \leftarrow \text{Detect}(G)$; // Get confidence maps and objects
5 **for** $i \in [1..|I|]$ ***where*** $O_i \cap R \neq \emptyset$ **do**
6 $S[I_i], N[I_i] \leftarrow C_i, 0$; // Update scores and mark visited
7 **if** *Verify*$(V[I_i]) > \theta$ **then**
8 $F, \tau \leftarrow F \cup \{V[I_i]\}, \tau \cup \{I_i/fps\}, R \leftarrow R \setminus (O_i \cap R)$
9 $P \leftarrow \text{Normalize}(\text{Spline}(S, N))$; // Update distribution
10 **return** $\text{Sample}(V, S, K)$

2.4. Evaluation Metrics for Search Efficiency

Previous research [14, 49, 67, 70, 73] have primarily focused on downstream task performance and overlook the temporal search computational efficiency. We evaluate search efficiency with three key metrics: **1) Frame Cost**, which measures the total number of frames processed, **2) FLOPs**, which quantifies the computational complexity, and **3) Latency**, which captures the total search time.

3. *T**: Efficient Temporal Search

*T** facilitates long-form video understanding through temporal search, reformulated as spatial search with spatial search models. The framework (Figure 2) comprises three phases: question grounding (§3.1), iterative temporal search (§3.2), and downstream task completion (§3.3). The first two phases conduct temporal search to identify keyframes, and the last phase forward these frames to a vision language model to answer questions. The temporal search process is shown in Algorithm 1, and explained in detail as follows.

3.1. Question Grounding

The question grounding phase aims to obtain target objects T and cue objects C essential for temporal search with spatial search models. We sample N frames at fixed intervals from video V, denoted as $\overline{V_N}$ for the VLM to scan. The VLM processes these frames with question Q to identify two types of elements: (1) **Target Objects** T, visual elements directly relevant to answering the question, (2) **Cue Objects** C, contextual elements indicating potential regions of interest. These objects are formally represented as:

$$\{T, C\} = \texttt{VLM}(\overline{V_N}, Q). \tag{3}$$

This query grounding phase identifies both primary targets and contextual cues helpful to answer the question, which are then used to guide the search process (§3.2). As shown in Figure 2, for the question "What did I put in the black dustbin?", the VLM identifies both target object (dustbin) and cue objects (room corners, furniture placement).

3.2. Iterative Temporal Search

Initialization The temporal search begins by initializing a uniform probability distribution P over frames, and a confidence threshold θ for object detection. The initial score distribution S and non-visiting indicator N_v are initialized as zero and one vectors over the total frame count L (Algorithm 1, line 1). Each object $o \in \mathcal{O}$ is weighted at 1.0 for targets and 0.5 for cues to reflect search importance. The remaining target set R starts with all targets T, while two empty sets F and τ are created for storing keyframes and timestamps, respectively.

Frame Sampling and Grid Construction In each iteration, the algorithm samples frames according to the current probability distribution P. We arrange sampled frames into a grid layout G sized $g \times g$, where indices I are first sampled and then used to construct the grid (Algorithm 1, line 3). The sampling process is defined as:

$$I = \text{Sample}(P \odot N_v, g^2) \tag{4}$$

where g^2 is the number of frames to sample and $\odot$ denotes element-wise multiplication. The search budget B is then reduced by g^2 after grid construction.

Object Detection and Scoring For each grid image, we perform object detection using a pre-trained model to identify both target and cue objects (line 4). The detection confidence for each grid cell (i, j) is computed as:

$$C_{i,j} = \max_{o \in \mathcal{D}_{i,j}} (c_o \cdot w_o) \tag{5}$$

where $\mathcal{D}_{i,j}$ represents the detected objects in cell (i, j), c_o is the detection confidence, and w_o is the object weight. When target objects are detected with sufficient confidence, they are added to the keyframe set and removed from the remaining targets (lines 5-8).

Distribution Update The score distribution is updated by spline-based interpolation (line 9). For each sampled frame $f \in F_s$, we update its score and mark it as visited:

$$S_f = C_f, \quad N_{v,f} = 0 \tag{6}$$

Method	Frames↓	HAYSTACK-EGO4D						HAYSTACK-LVBENCH					
		Temporal			Visual			Temporal			Visual		
		Precision ↑	Recall ↑	F_1 ↑	Precision ↑	Recall ↑	F_1↑	Precision ↑	Recall ↑	F_1 ↑	Precision ↑	Recall ↑	F_1↑
Baselines: Static Frame Sampling													
Uniform [72]	8	1.0	3.4	1.6	58.0	63.0	60.2	1.4	6.3	2.2	56.0	72.0	62.7
Uniform [72]	32	1.1	14.8	2.0	58.5	65.6	61.5	1.4	24.9	2.7	58.7	81.6	67.3
Baselines: Adaptive Temporal Search													
VideoAgent [68]	10.1	1.7	5.8	2.7	58.0	62.4	59.9	1.2	8.5	2.1	58.8	73.2	64.7
Retrieval-based	8	1.2	4.2	1.9	58.5	61.7	59.9	1.5	6.3	2.3	63.1	65.5	64.1
Retrieval-based	32	1.0	13.8	1.9	58.5	65.4	61.4	1.3	21.8	2.4	**59.9**	80.8	**67.8**
Ours: T^* for Zooming In Temporal Search													
Attention-based	8	2.2	7.5	3.3	58.4	62.5	60.2	1.5	6.6	2.4	63.6	68.6	65.7
Training-based	8	1.4	4.9	2.1	58.0	61.5	59.6	1.5	6.6	2.3	59.8	71.1	64.5
Detector-based	8	1.7	5.8	2.7	63.8	70.1	66.8	1.6	7.1	2.5	58.4	72.7	64.3
Detector-based	32	**1.8**	**26.3**	**3.4**	**62.9**	**76.2**	**68.9**	**1.7**	**28.2**	**3.1**	58.3	**83.2**	**67.8**

Table 2. Search utility results on LV-HAYSTACK. 8-frame setting bests are underlined, 32-frame setting bests are in **bold**. We show that more searched frames consistently improves recall but reduces precision in retrieval methods. Detector-based *T** achieves best performance in 32-frame setting across metrics, demonstrating the effectiveness of visual grounding and iterative temporal search. Attention-based *T** performs well in 8-frame setting but requires larger foundation models, thereby reducing efficiency.

Method	Search Efficiency				Overall Task Efficiency		
	Grounding	Matching	TFLOPs ↓	Latency (sec) ↓	TFLOPs ↓	Latency (sec) ↓	Acc ↑
Baselines: Static Frame Sampling							
Uniform-8 [72]	N/A	N/A	N/A	0.2	139.3	3.8	45.9
Baselines: Adaptive Temporal Search							
VideoAgent [68]	GPT4×4	CLIP-1B×840	536.5†	30.2	690.7†	34.9	49.2
Retrieval-based	N/A	YOLO-world-110M×840	216.1	28.6	355.4	32.2	50.3
Ours: T^* for Efficient Temporal Search							
Attention-based	LLaVA-72B×3	N/A	88.9	13.7	228.2	17.3	49.6
Detector-based	LLaVA-7B×1	YOLO-world-110M×49	33.3	7.5	172.6	11.1	50.8
Training-based	LLaVA-7B×1	YOLO-world-110M×38	**30.3**	**6.8**	**169.6**	**10.4**	**51.0**

Table 3. Efficiency results on the full LV-HAYSTACK, including search efficiency and overall (search+downstream) efficiency. We report the search models used and their avg. call frequency (e.g., VideoAgent calls GPT-4 four times for grounding). *T** achieves high performance with significantly less computation and lower latency. VideoAgent's FLOPs (†) exclude GPT-4 costs due to its closed-source nature. All training and inference operations are carried out on a cluster of 8*H800 Nvidia GPUs.

To capture temporal locality, we employ a window-based update for high-confidence frames:

$$S_{f\pm\delta} = \max(S_{f\pm\delta}, \frac{S_f}{|\delta|+1}), \quad \delta \in [-w, w] \tag{7}$$

where w is the window size. The probability distribution P is then updated using spline interpolation and normalized.

The search process continues iteratively until either all target objects are found or the search budget B is exhausted. Finally, the algorithm returns the top K frames based on their final scores (line 10).

3.3. Downstream Task Completion

The final keyframes are selected using TopK operation on the score distribution (Algorithm 1, line 10), which returns K frames with timestamps for downstream tasks, ensuring both relevance and temporal coverage.

4. Experimental Setup

4.1. Evaluations on Search Utility and Efficiency

Datasets and models. We evaluate on LV-HAYSTACK (Sec. 2.2). For downstream task efficiency evaluation, 8 searched frames are passed to LLaVA-OneVision-72B for all methods. We implement the spatial search model H with three complementary ways: (1) attention-based using VLM's attention matrix, (2) detector-based using object detector like YOLO-world [8], (3) training-based using custom trained models. More details can be found in the codebase.

Evaluation Metrics. We report search performance metrics from §2.3 and §2.4 with a 5-second temporal threshold.

Baselines. We include three representative sampling or search strategies: **1) Uniform Sampling** following [28, 72]; **2) Temporal Search methods like VideoAgent** [68] which leverages LLM-based video keyframe selection; **3)**

LongVideoBench						Video-MME					
Model and Size	#Frame	Video Length XLong 15-60min	Long 2-10min	Medium 15-60s	Short 8-15s	Model and Size	#Frame	Video Length Long 41min	Medium 9min	Short 1.3min	Total 17min
GPT4o	8	47.1	49.4	67.3	69.7	GPT4o	8	51.4	54.3	55.7	53.8
GPT4o + *T**	8	**51.9**	**52.4**	**72.7**	**70.0**	GPT4o + *T**	8	**55.9**	**57.3**	**56.4**	**56.5**
LLaVA-OneVision-72B	8	53.7	57.4	74.1	73.0	LLaVA-OneVision-72B	8	52.6	55.5	59.6	55.9
LLaVA-OneVision-72B + *T**	8	**55.5**	**63.7**	**76.3**	**73.5**	LLaVA-OneVision-72B + *T**	8	**57.7**	**57.5**	**61.7**	**59.0**
GPT4o	32	50.5	57.3	73.5	71.4	GPT4o	32	56.3	60.7	68.3	61.8
GPT4o + *T**	32	**53.1**	**59.4**	**74.3**	**71.4**	GPT4o + *T**	32	**59.3**	**63.5**	**69.5**	**64.1**
LLaVA-OneVision-72B	32	56.5	61.6	77.4	74.3	LLaVA-OneVision-72B	32	60.0	62.2	76.7	66.3
LLaVA-OneVision-72B + *T**	32	**62.4**	**64.1**	**79.3**	**74.6**	LLaVA-OneVision-72B + *T**	32	**61.0**	**66.6**	**77.5**	**68.3**
GPT-4o (0513)	256	61.6	66.7	76.8	71.6	Gemini-1.5-Pro (0615)	1/0.5 fps[1*]	67.4	74.3	81.7	75.0
Aria-8x3.5B	256	60.1	64.6	76.6	69.4	Qwen2-VL-72B	768[3*]	62.2	71.3	80.1	71.2
LLaVA-Video-72B-Qwen2	128	59.3	63.9	77.4	72.4	GPT-4o (0615)	384[2*]	65.3	70.3	80.0	71.9
Gemini-1.5-Pro (0514)	256	59.1	65.0	75.3	70.2	LLaVA-Video-72B	64	61.5	68.9	81.4	70.6
Qwen2-VL-7B	256	52.5	56.7	67.6	60.1	Aria-8x3.5B	256	58.8	67.0	76.9	67.6

Table 4. Downstream task evaluation results show T* effectiveness as a temporal search module for VLMs on LongVideoBench and Video-MME (without subtitles for fair comparison). Using QA accuracy (%) as the metric, we compare with top leaderboard models (shown in gray), noting these typically use substantially more frames, making direct comparisons challenging. Models are ranked by XLong video performance on LongVideoBench and total score on Video-MME, with frame counts indicated. All baseline figures are directly cited from their original publications. Standard deviations and more detailed analysis are available in Appendix C.

Retrieval-based methods that score and rank all frames instead of *T** search methods with iterative frame sampling.

4.2. Evaluations on Downstream Tasks: Video QA

Datasets. We evaluate QA performance on a diverse set of video understanding tasks: LongVideoBench [72], Video-MME [15], EgoSchema [43], NExT-QA [77] and Ego4D LongVideo QA, which we extended from Ego4D NLQ tast [19].

The videos range from brief clips (15 seconds) to extensive narratives (up to 60 minutes), covering tasks like temporal action reasoning, causal inference, and egocentric understanding.

Evaluation Metrics. We adopt accuracy on downstream QA tasks, following [32, 53, 65, 72, 93].

Baselines. We test various open/closed-source VLMs, comparing *T** and uniform frame selection with 8/32 frames. Implementation details can be found in Appendix D.

5. Experimental Results

5.1. Results on LV-Haystack Search Performance

For search utility results (Table 2), attention-based *T** achieves the best temporal metrics with higher precision, recall, and F1 scores with 8 frames. For 32 frames, detector-based *T** has best performance across both temporal and visual metrics. Results show more frames improve recall but reduce precision in retrieval methods, demonstrating the effectiveness of visual grounding and iterative temporal search over retrieval-based methods or uniform sampling.

Model	Frames	NExT-QA 0.7min	EgoSchema 3min
Baselines using Static Uniform Sampling			
InternVideo [69]	90	49.1	32.1
MVU [52]	16	55.2	60.3
LLoVi [86]	90	67.7	57.6
LangRepo [23]	180	60.9	66.2
LLaVA-OneVision-7B [28]	32	79.4	65.4
Baselines using Adaptive Frame Selecting			
SeViLA [83]	32	63.6	25.7
VideoAgent [67]	8.4	71.3	60.2
LVNet [49]	12	72.9	66.0
VideoTree [70]	63.2	73.5	66.2
VidF4 [37]	8	74.1	-
Ours: Plug in *T** for Efficient Temporal Search			
LLaVA-OneVision-7B	8	76.4	63.6
+Detector-based *T**	8	**80.4**	**66.6**

Table 5. Downstream task evaluation results by plugging in *T** as an additional temporal search method for VLMs on NExT-QA and EgoSchema. The video length is shorter than Table 4. Baseline results are directly cited from respective publications.

For search efficiency results, as shown in Table 3, *T** achieve competitive accuracy with significantly fewer TFLOPs and lower latency than baselines. Training-based T* is particularly efficient (169.6 TFLOPs, 10.4s latency, 60.3% accuracy) compared to VideoAgent (690.7 TFLOPs, 34.9s latency, 49.2% accuracy). While uniform sampling has no search cost, it requires more frames to achieve similar performance, leading to more computational costs.

Model	Frames	Tiny		Test	
		Clip	Video	Clip	Video
Baselines using Static Uniform Sampling					
GPT4o	8	45.5	41.5	45.9	45.3
GPT4o + *T**	8	49.5	45.0	49.4	46.7
GPT4o	32	49.0	45.5	52.3	51.0
GPT4o + *T**	32	**51.0**	46.5	54.9	52.5
QWen2.5VL 7B	8	37.0	32.0	40.0	38.8
QWen2.5VL 7B + *T**	8	38.5	37.0	42.7	40.3
QWen2.5VL 7B	16	37.5	35.0	40.9	38.8
QWen2.5VL 7B + *T**	16	39.5	38.5	43.8	42.8
QWen2.5VL 72B	8	45.0	45.0	51.0	50.1
QWen2.5VL 72B + *T**	8	45.5	46.0	53.5	52.8
QWen2.5VL 72B	16	49.0	49.5	53.6	50.6
QWen2.5VL 72B + *T**	16	50.0	**50.0**	**55.1**	**52.8**

Table 6. Downstream evaluation results on the Ego4D Longvideo QA dataset. We extend the Ego4D NLQ task by including answer options and responses, and report performance for both clip-level and full video inputs using vision-language models.

5.2. Results on Downstream Tasks: Long Video QA

We evaluate *T** on four QA datasets by integrating it as a lightweight plugin into proprietary (GPT-4o) and open-source (LLaVA-OnVision-72B) vision-language models. For long-form videos, as shown in Table 4, *T** enhances VLM performance on LongVideoBench and VideoMME consistently across various frame budgets, video lengths, and VLMs. For short videos, Table 5 demonstrates that on NExT-QA and EgoSchema, *T** outperforms other frame selection methods while using the least number of frames.

Notably, *T** is particularly effective for longer videos and smaller frame budgets. On the LongVideoBench XLong subset with an 8-frame constraint, *T** increases the SOTA models with a large margin, boosting GPT-4o performance from 47.1% to 51.9% with 8 frames, and LLaVA- OneVision-OV-72B from 56.5% to 62.4% with 32 frames.

5.3. Results on Ego4D LongVideo QA

In the original Ego4D [19] NLQ task, each sample consists of a text query, an hours-long video, and a recommended video clip (approximately 10 minutes) that provides context for the query. In LVHaystack, we recruited seven annotators to answer the queries from the Ego4D NLQ test set and to generate corresponding answer options, resulting in a new long-video QA dataset from an ego-centric perspective. The dataset is partitioned into three subsets: tiny, dev, and test.

We report the performance of two mainstream open-source models, QWen2.5-VL [2] and GPT4o. To facilitate future research, we provide results using both the full-length videos (hours-long video) and shorter clips (minutes-long video). The results are shown in Table 6.

6. Analysis

6.1. Time and Cost Complexity of *T**

*T** can be viewed as a quaternary search guided by a heuristic informed by an object detector. **In the worst case,** where the heuristic provides no useful information, *T** randoms upsampling frames into the image grid with a complexity of $\mathcal{O}(\log L)$. **In the best case,** the heuristic always identifies the most relevant top cell in the grid. So *T** operates as a quaternary search with a complexity of $\mathcal{O}(\log L)$.

Therefore, *T** offers better efficiency compared to other methods that perform linear processing and examine all frames. And, the complexity for processing images in this scenario lies between $\mathcal{O}(L)$ and $\mathcal{O}\left(\frac{\log L}{P}\right)$, where P is the probability that the object-detector-based heuristic selects the target frame from among image cells.

The computational overhead C of *T** is:

$$C = \underbrace{N_{\text{VLM}} \cdot C_{\text{VLM}}}_{\text{Grounding Overhead}} + \underbrace{N_{\text{YOLO-world}} \cdot C_{\text{YOLO-world}}}_{\text{Matching Overhead}}, \quad (8)$$

Where, the overhead includes the frequency of reasoning and processing by the VLM and the cost associated with YOLO-world processing for each image grid. Empirical measurements of C_{VLM} and $C_{\text{YOLO-world}}$ are provided in Section 4.1.

6.2. Sampling Iteration Dynamics

We show the dynamics of temporal focus over multiple iterations of temporal search for three example videos in Figure 3. As one can observe, the results show that our method progressively aligns sampling weights with ground truth frames across iterations, enhancing the model's ability to focus on relevant frames. Notably, even for distantly separated frames (e.g., around 50s and 100s in the top video), the model can simultaneously increase the sampling weights, demonstrating its ability to capture multiple critical frames in videos. This iterative refinement allows the model to identify and emphasize key frames accurately, improving overall long-form video understanding performance.

6.3. Effect of Search Frame Count on Accuracy

This section explores how the number of search frames influences the performance of our Visual Language Models (VLMs) on LongVideoBench.

Figure 4 illustrates the impact of different numbers of search frames on the performance of VLMs on LongVideoBench. The results show that *T** consistently outperforms the baseline model across varying frame counts and closely approaches human selection (oracle) accuracy as the frame count increases. Notably, with 64 frames, *T** achieves performance on par with human-selected frames, indicating that our method effectively captures the essential information with fewer frames.

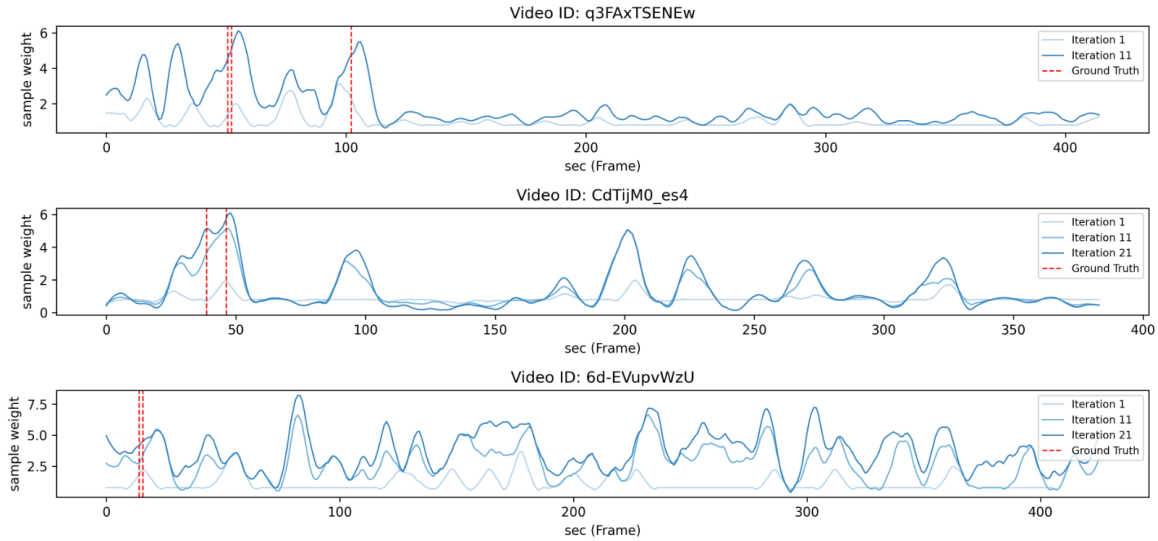

Figure 3. **Sampling weight dynamics over iterations for example videos.** Ground truth frames are marked in red. Sampling weights progressively focus on ground truth frames across iterations (1, 11, and 21), indicating improved model alignment with keyframes over time. Notably, due to the efficient sampling in temporal search, our model can simultaneously zoom in and focus on distantly located key frames (e.g., around 50s and 100s in the top plot).

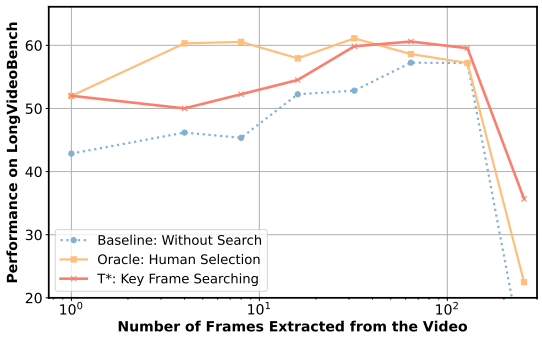

Figure 4. Performance improvement with increasing search frames. *T** consistently enhances accuracy and reaches near-human oracle performance at 64 frames.

7. Related Work

Long-form Video Understanding. Recent attention mechanisms [3, 10, 11, 29, 40, 42, 58, 62, 85, 90, 92] and video transformers [38, 88] improve temporal processing [21, 39, 48, 50] but struggle with long-range dependencies [60]. Current solutions use compression [22, 57] or frame selection [16, 33, 54, 56, 79, 83], while existing benchmarks [65, 93] focus on long videos, but are only evlauated on downstream QA while we focus on temporal search evaluation.

Temporal Localization and Temporal Search. While temporal localization [1, 17, 27, 55, 61, 78, 80, 81, 87] struggles with boundary detection, recent keyframe selection advances from "glance annotation" [9] to caption-based [24] and fine-grained approaches [7, 27]. Our work focus on a more challenging problem with longer videos.

Needle in a Haystack. Needle in a Haystack approaches span text [25, 45] and multimodal [4, 13, 63, 74, 89] domains but rely on synthetic data, while our Long Video Haystack focuses on real-world natural video contexts.

8. Conclusion

In this work, we revisit temporal search paradigms for long-form video understanding, studying a fundamental issue pertaining to all SOTA long-context vision-language models (VLMs). First, we formulate temporal search as a Long Video Haystack problem, i.e., finding a minimal set of relevant frames among tens of thousands of frames from real-world long videos given specific queries. To validate our formulation, we create **LV-HAYSTACK**, the first benchmark containing 15,092 human-annotated instances with a set of fine-grained evaluation metrics for assessing temporal search quality and computational efficiency. Empirical results on **LV-HAYSTACK** under SOTA temporal search methods reveal a significant gap in temporal search capabilties. Next, we re-think temporal search in long-form videos and propose a lightweight temporal search framework, ***T****, which casts the expensive temporal search as a spatial search problem. Extensive experiments show that when integrated with video understanding models, ***T**** significantly improves SOTA performance. We hope that **LV-HAYSTACK** and *T** framework will drive meaningful advancements in developing efficient long-form video understanding systems.

Limitations

A potential limitation of our work lies in the assumption that most problems can be addressed with a few keyframes, which may not fully extend to more complex tasks requiring a broader context or dense reasoning. Additionally, our approach focuses primarily on visual cues, without leveraging other modalities such as audio or subtitles, which can be explored in future work to enhance multi-modal understanding

Acknowledgments. This work is in part supported by ONR N00014-23-1-2355 and ONR MURI N00014-22-1-2740. We thank Anabella Isaro for their valuable contributions in helping organizing open-source code.

References

[1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3173–3183, 2021. 2, 8

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 7

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 8

[4] Eleftheria Briakou, Colin Cherry, and George Foster. Searching for needles in a haystack: On the role of incidental bilingualism in palm's translation capability. *arXiv preprint arXiv:2305.10266*, 2023. 2, 8

[5] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2011. 3

[6] Keshigeyan Chandrasegaran, Agrim Gupta, Lea M. Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. Hourvideo: 1-hour video-language understanding, 2024. 1

[7] Houlun Chen, Xin Wang, Hong Chen, Zeyang Zhang, Wei Feng, Bin Huang, Jia Jia, and Wenwu Zhu. Verified: A video corpus moment retrieval benchmark for fine-grained video understanding. *arXiv preprint arXiv:2410.08593*, 2024. 8

[8] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911, 2024. 5, 18, 19

[9] Ran Cui, Tianwen Qian, Pai Peng, Elena Daskalaki, Jingjing Chen, Xiaowei Guo, Huyang Sun, and Yu-Gang Jiang. Video moment retrieval from text queries via single frame annotation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1033–1043, 2022. 8

[10] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. 8

[11] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 8

[12] Shangzhe Di and Yahong Han. Grounded question-answering in long egocentric videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):14898–14912, 2023. 1

[13] Nigel Joseph Bandeira Dias, Gustavo Teodoro Laureano, and Ronaldo Martins Da Costa. Keyframe selection for visual localization and mapping tasks: A systematic literature review. *Robotics*, 12(3):88, 2023. 8

[14] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2025. 2, 4

[15] Chaoyou Fu, Yuhan Dai, Yongdong 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, 2024. 6

[16] Difei Gao, Luowei Zhou, Lei Ji, Linchao Zhu, Yi Yang, and Mike Zheng Shou. Mist: Multi-modal iterative spatial-temporal transformer for long-form video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14773–14783, 2023. 8

[17] Junyu Gao and Changsheng Xu. Fast video moment retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1523–1532, 2021. 2, 8

[18] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem,

Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022. 1, 2, 3, 19

[19] 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. 6, 7

[20] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, Lei Zhao, Zhuoyi Yang, Xiaotao Gu, Xiaohan Zhang, Guanyu Feng, Da Yin, Zihan Wang, Ji Qi, Xixuan Song, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Yuxiao Dong, and Jie Tang. Cogvlm2: Visual language models for image and video understanding, 2024. 2

[21] Minyoung Hwang, Joey Hejna, Dorsa Sadigh, and Yonatan Bisk. Motif: Motion instruction fine-tuning, 2024. 8

[22] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024. 2, 8

[23] Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. *arXiv preprint arXiv:2403.14622*, 2024. 6

[24] Keito Kudo, Haruki Nagasawa, Jun Suzuki, and Nobuyuki Shimizu. A challenging multimodal video summary: Simultaneously extracting and generating keyframe-caption pairs from video. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7380–7402, 2023. 8

[25] Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. In search of needles in a 11m haystack: Recurrent memory finds what llms miss, 2024. 2, 8

[26] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. 1

[27] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858, 2021. 2, 8

[28] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2, 5, 6, 19

[29] Dacheng Li, Rulin Shao, Anze Xie, Eric P Xing, Xuezhe Ma, Ion Stoica, Joseph E Gonzalez, and Hao Zhang. Distflashattn: Distributed memory-efficient attention for long-context llms training. In *First Conference on Language Modeling*, 2024. 8

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 18

[31] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2

[32] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 2, 6

[33] Yunxin Li, Xinyu Chen, Baotain Hu, and Min Zhang. Llms meet long video: Advancing long video question answering with an interactive visual adapter in llms. *arXiv preprint arXiv:2402.13546*, 2024. 8

[34] Yang Li, Yuxin Wen, Michael S. Ryoo, et al. End-to-end video question answering with frame scoring mechanisms and adaptive sampling. *arXiv preprint arXiv:2407.15047*, 2024. 2

[35] Yang Li, Yuxin Wen, Michael S. Ryoo, et al. Needle in a video haystack: A scalable synthetic evaluator for video mllms. *arXiv preprint arXiv:2406.09367*, 2024. 2

[36] 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. In *arXiv preprint arXiv:2407.15047*, 2024. 1

[37] 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 preprint arXiv:2407.15047*, 2024. 2, 6

[38] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 8

[39] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024. 8

[40] Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023. 8

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1

[42] Hao Liu, Matei Zaharia, and Pieter Abbeel. Ringattention with blockwise transformers for near-infinite context. In *The Twelfth International Conference on Learning Representations*, 2024. 8

[43] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 6

[44] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 2

[45] Elliot Nelson, Georgios Kollias, Payel Das, Subhajit Chaudhury, and Soham Dan. Needle in the haystack for memory based large language models, 2024. 2, 8

[46] Elliot Nelson, Georgios Kollias, Payel Das, Subhajit Chaudhury, and Soham Dan. Needle in the haystack for memory based large language models. *arXiv preprint arXiv:2407.01437*, 2024. 2

[47] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2023. 2

[48] Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. Quality: Question answering with long input texts, yes! *arXiv preprint arXiv:2112.08608*, 2021. 8

[49] Jongwoo 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 preprint arXiv:2406.09396*, 2024. 2, 4, 6

[50] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023. 8

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 18

[52] Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S Ryoo. Understanding long videos in one multimodal language model pass. *arXiv preprint arXiv:2403.16998*, 2024. 6

[53] Ruchit Rawal, Khalid Saifullah, Miquel Farré, Ronen Basri, David Jacobs, Gowthami Somepalli, and Tom Goldstein. Cinepile: A long video question answering dataset and benchmark. *arXiv preprint arXiv:2405.08813*, 2024. 2, 6

[54] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14313–14323, 2024. 8

[55] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2464–2473, 2020. 2, 8

[56] David Romero and Thamar Solorio. Question-instructed visual descriptions for zero-shot video question answering. *arXiv preprint arXiv:2402.10698*, 2024. 2, 8

[57] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 8

[58] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021. 8

[59] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13581–13591, 2024. 2

[60] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020. 8

[61] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 247–263, 2018. 2, 8

[62] Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling, 2023. 8

[63] 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. *arXiv preprint arXiv:2406.11230*, 2024. 2, 8

[64] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. 1

[65] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024. 2, 6, 8

[66] Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *arXiv preprint arXiv:2406.07230*, 2024. 2

[67] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *European Conference on Computer Vision (ECCV)*, 2024. 2, 4, 6

[68] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*, 2024. 2, 5, 16, 18

[69] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 6

[70] 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*, 2024. 2, 4, 6

[71] Yuxin Wen et al. Too many frames, not all useful: Efficient strategies for long-form video qa. *arXiv preprint arXiv:2406.09396*, 2024. 1

[72] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. 1, 2, 3, 5, 6, 14, 15, 16, 19

[73] Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024. 2, 4

[74] Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E. Gonzalez, Trevor Darrell, and David M. Chan. Visual haystacks: A vision-centric needle-in-a-haystack benchmark, 2024. 8

[75] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019. 2

[76] Zhenyu Wu et al. Discovering spatio-temporal rationales for video question answering. *arXiv preprint arXiv:2307.12058*, 2023. 1

[77] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9777–9786. Computer Vision Foundation / IEEE, 2021. 6

[78] Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. Boundary proposal network for two-stage natural language video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2986–2994, 2021. 2, 8

[79] 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. 8

[80] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13623–13633, 2023. 2, 8

[81] Jinhui Ye, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Hui Xiong. Cross-modality data augmentation for end-to-end sign language translation. *arXiv preprint arXiv:2305.11096*, 2023. 8

[82] Jinhui Ye, Xing Wang, Wenxiang Jiao, Junwei Liang, and Hui Xiong. Improving gloss-free sign language translation by reducing representation density. *arXiv preprint arXiv:2405.14312*, 2024. 2

[83] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6, 8

[84] Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xioalei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. *arXiv preprint arXiv:2410.03226*, 2024. 2

[85] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020. 8

[86] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. *arXiv preprint arXiv:2312.17235*, 2023. 2, 6

[87] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020. 2, 8

[88] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 8

[89] Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic framework for benchmarking video mllms. *arXiv preprint arXiv:2406.09367*, 2024. 2, 8

[90] Jiaming Zhou, Kun-Yu Lin, Haoxin Li, and Wei-Shi Zheng. Graph-based high-order relation modeling for long-term action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8984–8993, 2021. 8

[91] Jiaming Zhou, Hanjun Li, Kun-Yu Lin, and Junwei Liang. Adafocus: Towards end-to-end weakly supervised learning for long-video action understanding. *arXiv preprint arXiv:2311.17118*, 2023. 2

[92] Jiaming Zhou, Kun-Yu Lin, Yu-Kun Qiu, and Wei-Shi Zheng. Twinformer: Fine-to-coarse temporal modeling for long-term action recognition. *IEEE Transactions on Multimedia*, 2024. 8

[93] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024. 2, 6, 8

Appendix

Table of Contents

A. Ablation Study

This section investigates the sensitivity of different parameters in our proposed *T** framework.

A.1. Ablation on Question Grounding

Question Grounding transforms the original question into spatially queryable targets, as detailed in Section 4. In this study, we examine how increasing computational resources for Question grounding affects the efficiency and quality of the search process. Specifically, we analyze the impact of scaling Vision-Language Models (VLMs) from 7B (LLaVA) to 72B (LLaVA) parameters, as well as varying the number of initial frames, from 8 to 32.

The results, summarized in Table 7, indicate that increasing the VLM size and the number of initial frames marginally enhances both the effectiveness of the search process and downstream task performance. These results suggest that Question Grounding can be effectively achieved with modest resource allocations, offering a favorable balance between performance and resource usage.

Grounding VLM	Frames	TFLOPs	Visual F1	QA Acc
LLaVA-OneVision-7B	8	26.9	59.9	59.8
LLaVA-OneVision-72B	8	148.5	60.6	59.9
LLaVA-OneVision-7B	32	108.2	60.7	60.3

Table 7. Impact of VLM size and initial frame count on question grounding and search effectiveness on LV-HAYSTACK. Experimental settings are aligned with the baseline setup reported in Section 4 (main paper). Our results indicate that increasing the resources for question grounding results in marginal improvements in search effectiveness ($< 1\%$ gain in QA Acc.).

A.2. Ablation on Searching Algorithm

*T** aims to reduce computational overhead by partially representing the video as an $n \times n$ image grid. This approach leverages well-trained image models to systematically replace irrelevant grid cells until the target is found, based on a specified threshold θ.

Impact of Grid Size (n): We investigate how the configuration of the concatenated image grid affects both the search cost and the efficacy of the search process. Figure 5 displays the impact of varying grid sizes n (represented on the X-axis) on the average number of search steps and the corresponding average accuracy on LongVideoBench [72] XL subset.

Impact of Return Threshold θ: We examine how varying the return threshold θ impacts the efficiency and efficacy of the search process. As demonstrated in Figure 6, increasing the threshold tends to improve the accuracy of the search results but at the cost of increased computational effort. This trade-off is critical; thus, we have selected a default threshold of $\theta = 0.6$ for a balanced approach.

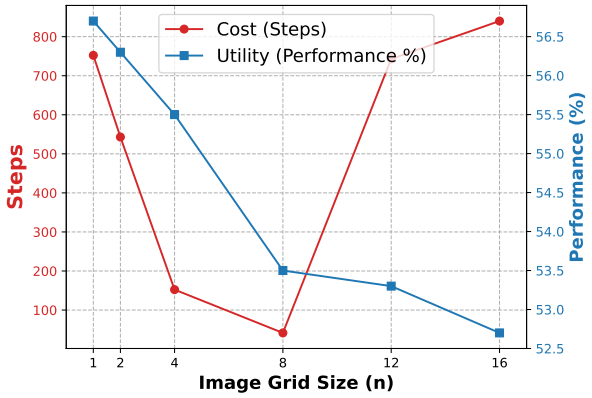

Figure 5. **Grid Size Impact on Search Performance**. The red line represents the average number of search iterations for different image grid configurations, while the blue line shows the performance on the LongVideoBench [72] XL subset using 8 frames and the LLaVA-72B as the downstream QA model.

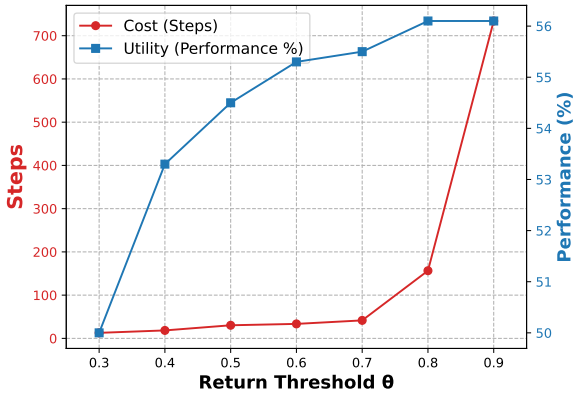

Figure 6. **Impact of Return Threshold** θ. The graph clearly illustrates the trade-off between threshold settings and search performance: lower thresholds result in quicker searches but may reduce accuracy, while higher thresholds enhance accuracy at the expense of increased search steps and computational cost.

A.3. Ablation on Searching Utility Metrics

In our primary evaluation framework, we adopt Temporal Similarity and Visual Similarity as core metrics for measuring search utility. To further investigate the robustness of our framework, we include semantic distance as an additional metric for ablation studies. Semantic distance measures the alignment of high-level features between predicted and annotated frames, as encoded in pretrained models such as `openai/clip-vit-large-patch14`.

The results on HAYSTACK-EGO4D are shown Appendix Table 8. While this metric provides insights into the semantic relevance of frames, our results reveal that its scores are closely clustered across methods, ranging from 87.9 to 89.2. Therefore, semantic distance, while informative, does not significantly discriminate between methods due to the high-level feature similarities shared across retrieved keyframes. We exclude it as an evaluation metric to maintain focus on the more distinctive temporal and visual search utility.

A.4. Correlation between Search Utility and Video Understanding

As discussed in Section 2.3, we propose multiple temporal and visual metrics to evaluate search utility. To identify the metrics most correlated with long-form video understanding, we analyzed the Pearson and Spearman correlation coefficients between utility scores and downstream task accuracy. Table 9 shows that Temporal $F1$ has the highest Pearson correlation, while Temporal Precision has the highest Spearman correlation with downstream performance, highlighting these metrics as strong predictors of effective video understanding.

B. Complexity Analysis on *T**

In this section, we analyze the time and cost complexity of the *T** search algorithm. *T** leverages adaptive temporal and spatial upsampling to efficiently collect partial information from the video and progressively determine the keyframe distribution. Based on previous observations, *T** prioritizes high-probability regions for efficient keyframe localization, similar to an A* search algorithm. By retaining only a portion of the video grid cells in each iteration (up to $1/b$ of total cells), *T** effectively performs a multi-branch search guided by a heuristic scoring function, forming a b-way search tree.

As illustrated in Figure 7, *T** is a quaternary search algorithm operating on a b-ary search tree. At each step, video frames are sampled on a grid with $b = n \times n$ cells. The top 25% of regions based on their scores are retained, and the algorithm prioritizes sampling frames around these high-scoring regions. Similar to the A* algorithm, *T** uses a heuristic scoring function to select branches, thereby shortening the search path. Ultimately, it performs a quaternary search with a heuristic function on a b-ary tree.

To simplify the discussion, assume a video of length L, containing only one frame that satisfies the target condition f_t. The grid size is $b = 2 \times 2$, and the probability that the scoring function selects the correct branch is P. The complexity analysis is conducted for the worst-case, best-case, and average-case scenarios.

Worst Case ($P \leq \frac{1}{b}$): In the worst case, when $P \leq \frac{1}{b}$, the scoring function provides no effective guidance, effectively selecting branches at random. The algorithm degrades to a linear search, sequentially checking each frame until the target frame f_t is found. The time complexity can be expressed as:

$$T_{\text{worst}} = \mathcal{O}(L), \tag{9}$$

Method	Frames↓	HAYSTACK-EGO4D								
		Temporal			Visual			Semantic		
		Precision ↑	Recall ↑	F_1 ↑	Precision ↑	Recall ↑	F_1 ↑	Precision ↑	Recall ↑	F_1 ↑
Baselines: Static Frame Sampling										
Uniform [72]	8	1.0	3.4	1.6	58.0	63.0	60.2	87.2	89.3	88.2
Uniform [72]	32	1.1	14.8	2.0	58.5	65.6	61.5	**87.3**	90.4	88.8
Baselines: Adaptive Frame Selection										
VideoAgent [68]	10.1	1.7	5.8	2.7	58.0	62.4	59.9	87.0	88.9	87.9
Retrieval-based	8	1.2	4.2	1.9	58.5	61.7	59.9	87.3	88.7	88.0
Retrieval-based	32	1.0	13.8	1.9	58.5	65.4	61.4	87.3	90.5	88.9
Ours: T^* for Zooming In Temporal Search										
Attention-based	8	2.2	7.5	3.3	58.4	62.5	60.2	87.3	89.1	88.1
Training-based	8	1.4	4.9	2.1	58.0	61.5	59.6	87.2	89.0	88.0
Detector-based	8	1.7	5.8	2.7	63.8	70.1	66.8	87.2	88.9	87.9
Detector-based	32	**1.8**	**26.3**	**3.4**	**62.9**	**76.2**	**68.9**	87.2	**91.4**	**89.2**

Table 8. Results of searching utility on LV-HAYSTACK. Best results for the 8-frame setting are underlined, and best results for the 32-frame setting are in **bold**. We include semantic metric (detailed in Appendix A.3) for ablation. Scores range closely from 87.9 to 89.2, showing limited differentiation across methods compared to temporal and visual metrics.

Metric	Pearson Correlation	Pearson p-value	Spearman Correlation	Spearman p-value
Temporal F_1	**0.901**	0.037	0.700	0.188
Temporal Precision	0.828	0.084	**0.975**	0.005
Visual F_1	0.829	0.083	0.600	0.285
Temporal Recall	0.655	0.231	0.700	0.188
Visual Recall	0.568	0.317	0.500	0.391
Visual Precision	0.327	0.591	0.100	0.873

Table 9. Pearson and Spearman correlations (with p-values) between search utility metrics and downstream task accuracy. The highest correlations are highlighted in **bold** for Temporal F_1 and Temporal Precision, suggesting they are strong predictors of effective video understanding performance.

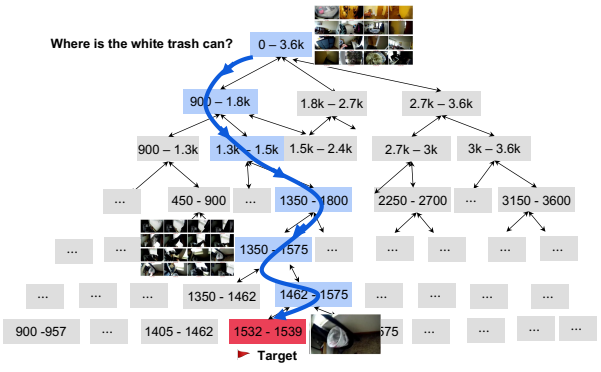

Figure 7. **Illustration of the *T** search process on a b-ary tree.** The video duration is 3.6k seconds, and the target white trash appears between 1532 and 1539 seconds. Numbers in the figure indicate the visited intervals of nodes, while the lines indicate the visited nodes and the search trajectory.

where L is the total number of video frames.

Best Case ($P = 1$): In the best case, when $P = 1$, the scoring function always selects the correct branch leading towards the target frame. The algorithm approaches the target frame directly at each step, similar to a b-ary search. The time complexity is given by:

$$T_{\text{best}} = \mathcal{O}(\log_b L), \tag{10}$$

where $b = n \times n$ is the branching factor determined by the grid size.

General Case ($\frac{1}{b} < P < 1$): In the general case, the scoring function improves branch selection accuracy based on scene correlations (e.g., a kitchen scene is more likely to contain a refrigerator than a bed). The search process can be modeled as a tree with depth:

$$m = \log_b L, \tag{11}$$

where m represents the depth of the tree. At each level, the expected number of attempts to correctly select the branch is $\frac{1}{P}$. Therefore, the total expected number of nodes visited is:

$$E[N] = m \times \frac{1}{P}, \tag{12}$$

where $E[N]$ represents the expected number of nodes visited.

The average time complexity is then given by:

$$T_{\text{avg}} = \mathcal{O}\left(\frac{\log_b L}{P}\right), \tag{13}$$

where the efficiency of the algorithm is inversely proportional to the scoring function's accuracy P. A higher P

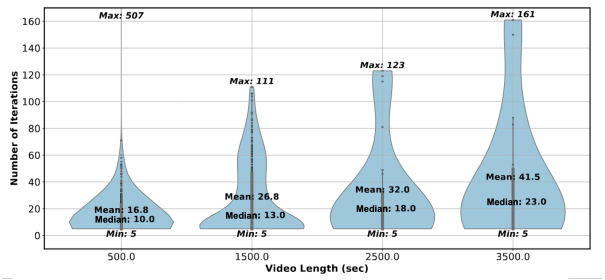

Figure 8. **Empirical results on search complexity of *T**.** We show the number of iterations of T^* on the LongVideoBench dataset, dividing videos with lengths between 0 and 4000 seconds into four groups. The figure shows the generally required intensity of iterative search as the length of videos vary.

value reduces the exploration of incorrect branches, significantly improving efficiency.

Figure 8 shows the behavior of *T** across various video lengths, presenting empirical statistics of search steps from our LV-HAYSTACK dataset. The statistical results demonstrate that for videos ranging from 100 to 3600 seconds, the average number of search steps is categorized into four equidistant groups. The average number of steps required by *T** increases gradually with video length. Notably, for videos longer than 3000 seconds, the maximum number of search steps recorded is 161, the minimum is 5 steps, and the average is 41.5 steps to complete the search. These variations are attributable to the differing intrinsic correlations within the content of each video, which informs the heuristic-based object detection process.

C. Detail Analysis on LongVideoBench

Table 10 highlights the impact of incorporating *T** as a frame selection module on QA accuracy across different video lengths in the LongVideoBench dataset.

Overall Effectiveness of *T:** Incorporating *T** consistently improves QA accuracy for both GPT4o and LLaVA-OneVision-72B across all video lengths, demonstrating the effectiveness of the keyframe selection module in enhancing video understanding. For instance, in XLong videos (15-60 minutes), GPT4o's accuracy improves from 47.1 to 51.55 $\pm$ 0.35, while LLaVA-OneVision-72B's accuracy increases from 53.7 to 55.25 $\pm$ 0.25.

Impact of Video Length: The improvements are more pronounced for longer videos (XLong and Long), where information density is higher, suggesting that *T** is particularly effective in identifying and prioritizing relevant frames in complex scenarios. For shorter videos (Medium and Short), while the improvements are relatively smaller, *T** still contributes to stabilizing performance across multiple runs.

Effect of Model Size: Larger models, such as LLaVA-OneVision-72B, benefit slightly more from *T** compared to smaller models like GPT4o, especially for longer videos. This indicates that larger models can better utilize the high-quality keyframes selected by *T**.

In conclusion, *T** consistently enhances QA accuracy across various video lengths, with greater impact on longer videos and larger models. These results demonstrate the potential of *T** in improving video-language understanding and reasoning in long-form videos.

D. Implementation Details

D.1. Implementation of Training-free *T**

Question Grounding: For Question Grounding, we primarily use the LLaVA-OneVision 7B model, applying it to 8 uniformly sampled frames. The prompt adheres to the official release guidelines, and the specific template used is listed in Table 12.

Iterative Temporal Search: The default configuration for the image grid size is $b = 8 \times 8$. We set the return threshold θ at 0.6 for object-based and training-based scoring functions as trade-off in Figure 6. For the attention-based method, we typically use the sum of the attention scores from the target object in the last layer of each frame. This approach was chosen because using smaller models or shallower layers resulted in performance below the baseline. Additionally, the process terminates after three iterations to manage the high computational costs associated with using the 72B model.

Downstream Question Answering: For downstream task evaluations, we experiment with the most prominent state-of-the-art (SOTA) models, both open- and closed-source, namely GPT4o and LLaVA-OneVision 72B. For GPT4o, we use the official API. For LLaVA, we employ the official code. The prompt template for this testing is listed in Table 13.

D.2. Implementation of Trainable T^*

In our framework, both object-based and attention-based *T** methods score each cell within the image grid and guide zooming based on straightforward rules. The training-based *T** approach, however, renders this iterative search process learnable.

To learn the search policy, we employ a reinforcement learning approach. Its action space, reward function, and loss function are described as follows:

Action Space To implement trainable scoring, we replace YOLO's detection header with a single-layer Multilayer Perceptron (MLP). This MLP maps high-level detection features into a score, indicating the likelihood that a specific area within a frame contains the visual context necessary to answer the question. For an image grid with $b = n \times n$ cells, each cell is assigned a predicted score, represented as

Model and Size	#Frame	LongVideoBench			
		Video Length			
		XLong 15-60min	Long 2-10min	Medium 15-60s	Short 8-15s
GPT4o	8	47.1	49.4	67.3	69.7
GPT4o + *T**	8	**51.6 ± 1.4**	**51.7 ± 1.7**	**72.9 ± 1.2**	**70.2 ± 0.2**
LLaVA-OneVision-72B	8	53.7	57.4	74.1	73.0
LLaVA-OneVision-72B + *T**	8	**55.3 ± 1.3**	**63.5 ± 1.2**	**76.6 ± 1.3**	**73.7 ± 0.2**
GPT4o	32	50.5	57.3	73.5	71.4
GPT4o + *T**	32	**53.3 ± 1.2**	**59.2 ± 1.2**	**74.3 ± 0.0**	**71.4 ± 0.0**
LLaVA-OneVision-72B	32	56.5	61.6	77.4	74.3
LLaVA-OneVision-72B + *T**	32	**62.6 ± 1.2**	**63.9 ± 1.2**	**79.3 ± 0.0**	**74.6 ± 0.0**

Table 10. Detailed downstream task evaluation results for T^* as an additional frame selection module for VLMs on LongVideoBench. The metric is QA accuracy (%). We run *T** two times and report the average accuracy and standard deviation (±).

$\mathcal{C} \in \mathbb{R}^{n \times n}$, which serves as the action space:

$$\mathcal{C}, \mathcal{B} \leftarrow \text{ScoreFunction}(G, \mathcal{T}) = \text{MLP}(\text{YOLO}(G, \mathcal{T})). \tag{14}$$

Reward Function To evaluate the quality of selected frames, we define a reward function based on their effectiveness in answering the question. Using the predicted scores $\mathcal{C} \in \mathbb{R}^{n \times n}$, we select K frames and pass them to a VLM for question answering. The reward is calculated as the difference in accuracy between selected frames and a uniform baseline:

$$\text{reward} = \text{VLM}(K_{\text{selected}}, Q) - \text{VLM}(K_{\text{uniform}}, Q), \tag{15}$$

where $\text{VLM}(K_{\text{uniform}}, Q)$ represents the baseline accuracy using uniform sampled frames, and $\text{VLM}(K_{\text{selected}}, Q)$ represents the accuracy using frames sampled based on the predicted relevance scores $\mathcal{C}$.

Loss Function To optimize the trainable scoring mechanism, we employ a reinforcement learning-inspired approach using Monte Carlo estimation. We sample K frames M times based on the predicted scores $\mathcal{C}$. These sampled frames are passed into a Visual Language Model (e.g., `llava-OneVision-7B`) to answer the question. The average accuracy across these attempts is used as the reward signal. The loss function for training is defined as:

$$\text{loss} = \sum_{i=1}^{M} \left(\text{reward}_i \times \text{CrossEntropy}(\mathcal{C}, \mathcal{C}_{\text{i}})\right), \tag{16}$$

where $\mathcal{C}_{\text{i}}$ represents binary labels for the i-th Monte Carlo sample, with selected cells for K as 1 and others as 0, and reward_i is the reward for the i-th sample, according to Eqn. 15.

This formulation ensures that the model is reinforced to predict scores $\mathcal{C}$ aligning with the sampling labels $\mathcal{C}_{\text{i}}$ when the reward is positive. Conversely, when the reward is negative, the model adjusts its predictions to reduce the similarity between $\mathcal{C}$ and $\mathcal{C}_{\text{i}}$, penalizing incorrect sampling patterns.

Training and Inference The search policy is trained on existing short videos and tested on unseen long videos. During the inference phase, *T** uses the output of the trained YOLO model as a heuristic score. All training and inference operations are carried out on a cluster of 8*H800 Nvidia GPUs.

We observed that models trained on the NExT-QA dataset can also effectively identify better frames for long video tasks, such as those in LongVideoBench. This suggests that unifying video representation as an $n \times n$ grid—whether for short or long videos—enables a consistent approach. Furthermore, identifying better frames should be considered a foundational task, facilitating cross-dataset generalization.

D.3. Implementation of the baseline VideoAgent

VideoAgent [68], the state-of-the-art temporal search baseline, leverages LLM-based video keyframe selection to optimize VLM input. It generates captions to describe video content and incrementally aggregates relevant information for question answering. We adapt the original public code to make it runnable for long video haystack setting and benchmark. While the original VideoAgent implementation uses BLIP-Large [30] for caption generation, which is significantly larger than our YOLO-based approach [8] (110M parameters), we adapted the implementation to use CLIP-1B [51] for fair comparison. Specifically, we employed `clip-vit-large-patch14` and `blip-image-captioning-large`* for our experiments.

*Available at CLIP and BLIP

D.4. Video QA Implementation

For downstream video question answering experiments, we uniformly use LLaVA-OneVision 72B [28] as our QA model. This open-source VLM excels in processing multimodal inputs, integrating text, image, and video analysis. We selected this model for its ability to handle arbitrary video frames and its demonstrated superior performance across diverse VQA benchmarks.

D.5. Implementation of Different Search Strategies

The core of *T** leverages a well-trained open-world YOLO model for rapid object verification based on question questions. We evaluate *T**'s effectiveness through three distinct search strategies:

- **Retrieval-based Search**: Utilizes the YOLO model [8] to exhaustively scan and rank video frames based on target object detection confidence. The top 8 frames (by default) are selected as final outputs.
- **Zooming In Search**: Implements a hierarchical approach starting with an $N \times N$ image grid matrix at low fps and resolution. The search progressively refines both fps and resolution in promising segments identified through object detection and visual cues, ultimately returning 8 frames.
- **Trainable Search**: Adapts frame processing dynamically through YOLO model fine-tuning. Beginning with uniform sampling on an $N \times N$ image grid, it predicts correlation coefficients to guide subsequent grid sampling distributions. This process iterates three times by default, maintaining an 8-frame output. The model is trained on NExT-QA dataset and evaluated across multiple datasets.

E. Data Annotation Details

To curate data for our benchmark, we repurpose established long-video understanding datasets that focused on question answering. To represent different visual scenes, we cover both egocentric and allocentric views [18, 72], resulting in two diverse subsets HAYSTACK-EGO4D and HAYSTACK-LVBENCH. This approach not only allows direct comparison with past results but also saves time and resources for extensive data curation. We ask crowd-source annotators to identify keyframes and answers for HAYSTACK-EGO4D, while directly borrowing the keyframes and answers annotated from LONGVIDEOBENCH.

E.1. HAYSTACK-LVBENCH

To curate data for HAYSTACK-LVBENCH, we utilize the frame positions from LONGVIDEOBENCH [72] as ground-truth human-recommended frame indices. This decision is based on LONGVIDEOBENCH's annotation process, where annotators were required to propose questions based on given frame positions. Since LONGVIDEOBENCH only retained frame position records in the validation set, we exclusively constructed HAYSTACK-LVBENCH using data from the validation set. Furthermore, considering our focus on long-video understanding, we only included cases from the 3600-second duration group. To ensure broader applicability, we also excluded cases that referenced subtitles in their questions. As a result, we obtained a final set of 114 videos and 342 question pairs, none of our selected cases relied on text subtitles. You can use our script *'Longvideobench2LVHaystackFormat.py'* to obtain HAYSTACK-LVBENCH and check more detailed statistics.

E.2. HAYSTACK-EGO4D

To create HAYSTACK-EGO4D, we conducted data annotation on a dataset comprising 1,324 video clips, which were extracted from the original 988 videos containing a total of 15,092 questions. The video clips were pre-segmented by the Ego4D dataset to ensure that each clip contained sufficient context for answering the associated questions. This segmentation also simplified the annotation process, as shorter videos allowed annotators to better comprehend the content and efficiently identify keyframes.

The detailed instructions and interface (see Figure 9) provided to the annotators are described in the next section, *Data Annotation Interface*. Annotators were instructed to watch each video clip and answer a predefined set of questions. For every question, they were required to identify and mark several keyframes within the video that were relevant to their answers. Subsequently, they answered the questions based on these selected frames.

To assist annotators, we provided a recommended time interval to help them quickly identify relevant frames. However, we also instructed them to watch the entire video before answering the questions, as the recommended intervals identified by the Ego4D dataset may not always be accurate. Watching the full video is crucial for ensuring logical correctness in keyframe identification. For example, some questions involve events such as *"What is the second time that somebody does something?"*, requiring the annotator to identify both the first occurrence and the second occurrence to answer accurately.

In cases where a video did not provide sufficient clues to answer certain questions (potentially due to mistakes in the original dataset), annotators were instructed to respond with *"Not able to answer the question"* and provide corresponding reasons (e.g., *"The object does not occur in this video"*).

Since the annotators were not native English speakers, we utilized the `googletrans` package to translate the original questions and the interface into their native language (Chinese). Similarly, their answers were translated back into English for consistency.

To ensure the quality of the annotations, we randomly sampled 100 question-answer pairs from the annotated dataset. Only 2 obvious mistakes were identified in the

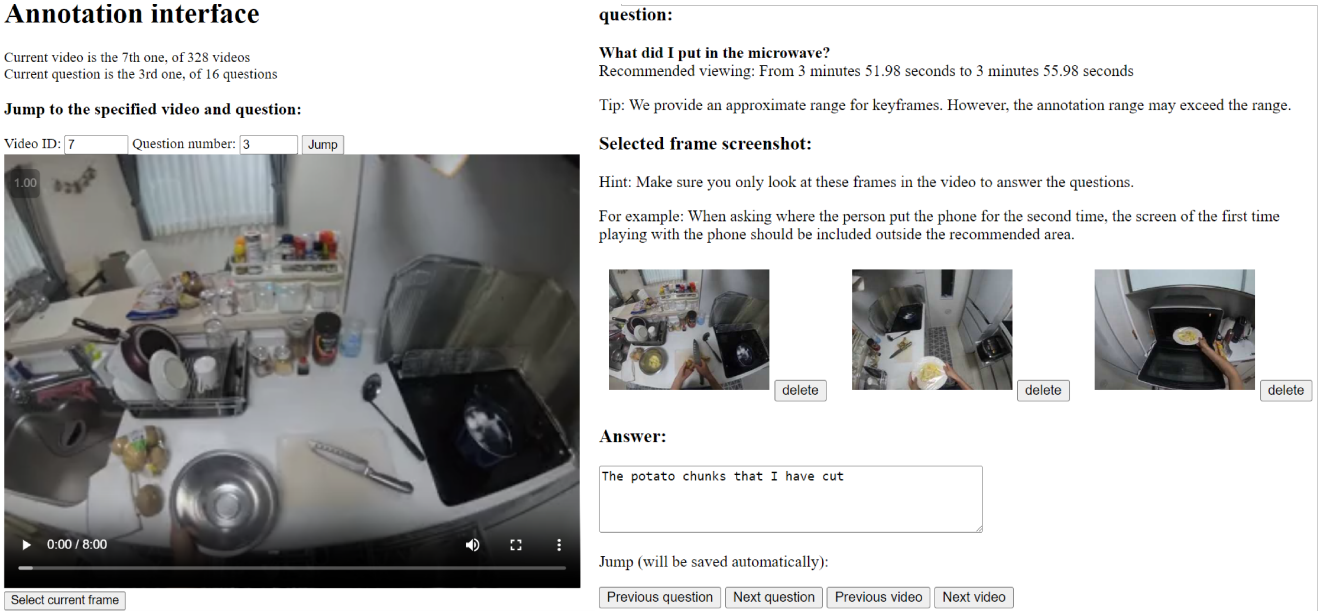

Figure 9. **Annotation Interface for Videos.** This interface allows annotators to answer questions based on video clips by keyframe annotations. Annotators can navigate to specific videos and questions using the provided controls (Video / Question ID). The current question is displayed with the recommended time range identified by the Ego4D dataset. Annotators can select key video frames and delete or modify annotations. A text box is available for entering answers based on observed video content.

Figure 10. The visualization of frame selection results demonstrates the effectiveness of our approach compared to baseline methods. Our method consistently identifies more relevant and temporally diverse keyframes, capturing important frames that directly address the question. In contrast to baseline approaches which may select redundant or less informative frames, our strategy achieves better coverage of key events while maintaining temporal coherence across the video sequence.

sample, indicating a decent overall annotation quality. Consequently, we retained this annotation set for further analysis.

F. Data Annotation Interface

Our data annotation interface facilitates annotators to provide precise answers to questions based on video clips. Key features include:

- **Video Navigation:** Annotators can jump to a specific video and question using the provided controls (e.g., Video ID and Question Number).
- **Question Display:** The current question is displayed prominently, along with the recommended time range for viewing relevant keyframes in the video.
- **Frame Selection:** Annotators can select specific video

frames for reference and delete or adjust their selection as needed to support their answer.

- **Answer Input:** A dedicated text box allows annotators to provide their responses based on the observed content in the selected video frames.
- **Navigation Controls:** Quick navigation buttons enable moving between videos or questions efficiently.

This tool ensures accurate, contextual, and streamlined annotations for video content analysis tasks.

G. Qualitative Analysis

Figure 11 compares uniform sampling with T* sampling for long-format video understanding. The task involves identifying a "metal cylinder" in an hour-long video. Uniform sampling, which selects 8 frames randomly, misses key frames containing the metal cylinder.

In contrast, T* sampling focuses on semantically relevant frames, successfully capturing those featuring the metal cylinder. This highlights T* sampling's ability to prioritize critical visual information

H. Prompt Design

In this section, we include the prompts designed for environment representation, focusing on question grounding and question answering tasks.

H.1. Prompt for Question Grounding

The following is the prompt used by our system for question grounding:

Prompt Template for Question Grounding

```
<system prompt>
Here is a video:
<image>
<image>
<image>
...
Here is a question about the video:
Question: <Question>
Options: <Options>

When answering this question about the video:
1. What key objects to locate the answer?
   - List potential key objects (short sentences,
     separated by commas).
2. What cue objects might be near the key objects
     and might appear in the scenes?
   - List potential cue objects (short sentences,
     separated by commas).

Please provide your answer in two lines, directly
     listing the key and cue objects, separated by
     commas.
```

Figure 12. The template of a question grounding prompt *T**. `<system prompt>` is the default system instruction from LLaVA and the `<image>` are PIL.Image objects for each frame and other text elements are strings.

This prompt is designed to generate a representation of the environment that facilitates the grounding of queries in a structured, object-centric manner.

H.2. Prompt for Question Answering

The following prompt is used to answer questions based on the embodied environment representation. This design is adapted from LLaVA:

Prompt Template for Question Answering

```
<system prompt>
Select the best answer to the following multiple-
     choice question based on the video.
<image>
<image>
<image>
...
Question:  <Question>
Options: <Options>

Answer with the option letter from the given choices
      directly.
```

Figure 13. The template of a question answering prompt.

H.3. Prompt for Distractor Generation

Prompt Template for Distractor Options

```
<system prompt>
You are an expert in creating challenging multiple-
     choice questions.
For the question: <question> with the correct answer
      <answer> generate 4 plausible but incorrect
     distractors that are closely related to the
     correct answer in context, category, or
     characteristics, making the question more
     challenging.
Ensure that the distractors could reasonably seem
     correct to someone who is unsure of the answer.
The output format should be:
"1. \"<Distractor 1>\"",
"2. \"<Distractor 2>\"",
"3. \"<Distractor 3>\"",
"4. \"<Distractor 4>\"".
```

Broad Impact

The *T** framework provides an efficient keyframe extraction solution compatible with any model or task, with applications in video summarization, healthcare training, entertainment indexing, and real-time surveillance. Its computational efficiency reduces energy consumption, aligning with sustainability goals. Additionally, the LV-HAYSTACK bench-

Figure 11. **Comparison of uniform sampling and T* sampling for long-format video understanding.** In this example, the task involves identifying a "metal cylinder" in an hour-long video. Uniform sampling fails to include relevant frames, as it randomly selects 8 frames across the video. In contrast, T* sampling dynamically selects frames containing the metal cylinder, providing the necessary visual context for effective understanding.

mark advances standardized evaluation practices, encouraging innovation in long-form video understanding.

Our proposed dataset is also applicable to foundation models that process entire videos. With our LV-HAYSTACK dataset which consists of both training and test sets, we aim to show that keyframe supervision can act as a guiding mechanism, enabling models to first identify the most relevant keyframes from video, then use them to produce a contextually grounded answer. This approach can be more structured, efficient, and effective than directly predicting an answer from a long-form video.