Iterative Zoom-In: Temporal Interval Exploration for Long-Form VideoQA

Anonymous ACL submission

Abstract

Multimodal Large Language Models (MLLMs) have shown strong performance in video understanding tasks. However, they continue to struggle with long-form videos because of an inefficient perception of temporal intervals. Unlike humans, who can dynamically adjust their temporal focus to locate query-relevant moments, current MLLMs often rely on dense, uniform sampling across the video timeline, leading to high memory consumption and a risk of missing crucial information. To address this challenge, we introduce Temporal Search (TS), a self-refine framework that enables MLLMs to explore temporal regions for improved long-video understanding iteratively. TS operates through two main iterative stages. First, the MLLM proposes a temporal interval that is likely to contain task-relevant information. Then, it samples frames from the interval, regardless of length, and feeds them into the model to produce a refined response and confidence score. TS refines the focus of the model by iteratively shifting attention to more fine-grained temporal intervals, improving its understanding of long videos. Additionally, keyframe-level descriptions are collected to facilitate cross-interval perception throughout the video. To further improve efficiency, we introduce TS-BFS, a best-first search strategy over a tree. Each node represents a candidate interval and is expanded via two methods: self-driven proposals and uniform partitioning. Nodes are scored based on confidence and self-evaluation, and the most promising one is selected for continued exploration. Extensive experiments on long-form video QA benchmarks demonstrate that TS consistently improves the performance of MLLMs on long videos. Specifically, TS-BFS boosts Qwen-VL-2.5's accuracy from 51.5% to 57.9% on LongVideoBench, and from 48.5% to 55.1% on VideoMME.

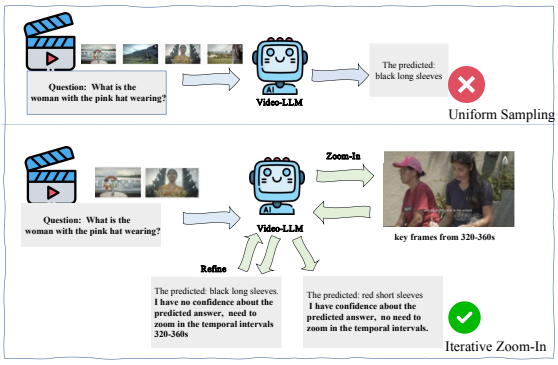

Figure 1: Iterative Zoom-In vs. Single-Pass Inference. Our method iteratively proposes new temporal intervals and samples frames from each. Shorter intervals enable finer temporal perception, effectively zooming in on relevant intervals for more accurate understanding.

1 Introduction

Multimodal Large Language Models (MLLMs) (Li et al., 2023; Zhang et al., 2023; Lin et al., 2023; Li et al., 2024a; Bai et al., 2025; Zhang et al., 2024c) have shown strong capabilities in video understanding tasks (Yu et al., 2019; Ning et al., 2023; Chen et al., 2023; Fang et al., 2024; Li et al., 2024c,b). However, they still face challenges when dealing with long-form videos (Fu et al., 2024; Wu et al., 2024a), largely due to the computational cost associated with processing a large number of frames. A common practice is to apply uniform dense sampling in the video timeline (Tang et al., 2025; Nguyen et al., 2024; Xiao et al., 2025; Huang et al., 2024). While simple, this approach is memory-intensive and often fails to capture the fine-grained temporal cues essential for long-form understanding (Cheng et al., 2025). In contrast, humans tend to analyze videos more selectively. Rather than inspecting all frames, they intuitively focus on salient temporal regions (Zacks and Swallow, 2007). When visual information is not clear, humans tend to iteratively focus on

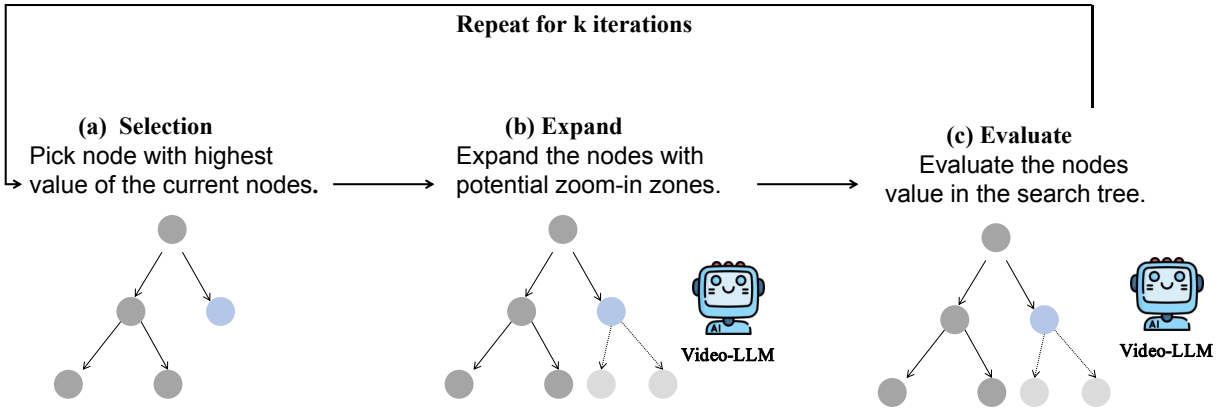

Figure 2: Nodes are selected based on a combined score of the model's prediction confidence and its self-evaluation of answer correctness. Each node is expanded using two strategies: model-guided proposal and uniform splitting of the current interval. In each iteration, the VideoLLM proposes new temporal intervals, generates responses with confidence scores, evaluates its answers, and produces keyframe-level descriptions incorporated into the input to support global video understanding.

fine-grained temporal regions until a satisfactory answer is found (Frederick, 2005; Paternoster and Pogarsky, 2009). Inspired by this behaviour, we develop a temporal search strategy that iteratively explores diverse regions of the video and refines predictions until a confident conclusion is reached. In particular, we observe that the confidence in the model generation over a temporal interval is highly correlated with its prediction accuracy. Building on this insight, we propose Temporal Search (TS), a search-based framework for efficient and accurate long-form video understanding as shown in Figure 1.

TS iteratively proposes temporal intervals that are likely to contain task-relevant content and uniformly samples frames to feed into the MLLM regardless of the length of the interval. As intervals become shorter, the input captures finer-grained temporal details, enabling the model to zoom in on critical temporal regions iteratively. To further improve the iterative process, we introduce TS-BFS, a best-first search strategy over a temporal interval tree (TS-BFS). Each node represents a temporal segment defined by its start and end frames. The search involves three key operations: selection, expansion, and evaluation, as shown in Figure 2. The root node is initialised with frames sampled from the entire video. At each step, the most promising node is selected based on a combined score of confidence in the model and self-evaluation. Expansion is conducted by either the model proposing new intervals or by uniformly splitting the current interval. During evaluation, the model estimates the likelihood that its response is correct, based on the available information. In addition, it generates keyframe-level descriptions relevant to the query, which are stored in a global memory for cross-segment awareness. During the search, a priority queue ranks the candidate nodes, and the one with the highest score is selected for expansion. The search terminates either when the maximum number of iterations is reached or when the confidence score exceeds a predefined threshold. Unlike existing methods that process many uniformly sampled frames in a single pass, TS adopts an efficient iterative strategy that uses only a minimal number of frames per step. We evaluate TS on multiple long-form video understanding benchmarks, including LongVideoBench (Wu et al., 2024a) and VideoMME (Fu et al., 2024), achieving consistent accuracy gains over single-pass video perception with models such as Qwen-VL-2.5 (Bai et al., 2025), LLaVA-OV (Li et al., 2024a), and LLaVA-Video (Zhang et al., 2024d). Specifically, TS improves Qwen-VL-2.5's accuracy from 51.5% to 57.9% on LongVideoBench, and from 48.5% to 55.1% on VideoMME.

2 Related Works

2.1 Large Video-Language Models

Most Video-LLMs extend image-based multimodal models (Liu et al., 2023) by incorporating multiple frames to capture temporal dynamics (Li et al., 2023; Zhang et al., 2023; Lin et al., 2023; Li et al., 2024a). These models typically consist of a frame-wise visual encoder, followed by a projection mod-

ule (e.g., MLP or Q-Former) to align visual features with the language model (Nie et al.). This pipeline has proven effective on short video clips. However, as the number of frames increases, the memory and computational cost of frame-by-frame encoding grows rapidly (Xu et al., 2024). Additionally, dense and uniform frame sampling results in excessive redundancy, hindering video understanding performance. To alleviate this, recent studies address the challenge of limited input frame numbers by extending the context length of models (Chen et al., 2024b) or explore frame-level and token-level compression techniques such as adaptive pooling and token pruning (Liu et al., 2024; Wu et al., 2024b; Wang et al., 2024a; Alvar et al., 2025), thus significantly expanding the effective input frame number. Although these approaches reduce resource usage, they often degrade performance by discarding critical temporal details. In contrast, we propose Temporal Search (TS), a training-free framework that formulates temporal perception as a search problem. Instead of processing the entire video at once, TS performs inference at the segment level, iteratively selecting and reasoning over diverse temporal regions. Each iteration operates on a small, memory-efficient set of frames, avoiding the need for dense input and improving both efficiency and accuracy.

2.2 Tree Search with Large Language Models

Tree search algorithms have long been effective for navigating structured decision spaces, with strategies such as depth-first, breadth-first (Korf, 1985), best-first (Dechter and Pearl, 1985) widely used in planning and game-playing tasks. Recent research has explored their integration with Large Language Models (LLMs) to enhance reasoning and generation. By iteratively refining outputs through search, LLMs can better explore alternative answers and self-correct over time. This paradigm often referred to as self-refinement or tree-of-thoughts (Yao et al., 2023), has been shown to improve performance on complex question answering and decision-making tasks. In the multimodal domain, (Li et al., 2025; Shen et al., 2024) apply visual search techniques for image understanding tasks. Similarly, tree search methods have also been applied to video understanding. Existing studies like VideoTree (Wang et al., 2024d) and VideoAgent (Wang et al., 2024c) combine LLMs with external tools such as captioners or retrievers modules to process long videos. However, these pipelines often limit LLMs to controller or post-processing roles, rarely enabling direct reasoning over a video's temporal structure. In contrast, we treat temporal understanding as an active, internal search process. We propose a lightweight, self-refine tree search framework that empowers VideoLLMs to direct their temporal search. This approach extends recent LLM-based self-refine techniques (Yao et al., 2023; Hao et al., 2023; Zhang et al., 2024a) to long-form video reasoning with minimal external dependencies.

Algorithm 1 Vanilla Temporal Search (TS)

Input: Video $V = \{f_1, \ldots, f_T\}$; query Q; steps k; thresholds c_1, c_2; number of frames per inference n_f; model p_θ
Initialization: $I_0 = [0, T]$, keyframes $\mathcal{K} \leftarrow \emptyset$
Sample $\{f_i\} \sim \text{Uniform}(I_0, n_f)$
$(\hat{y}, \gamma) \leftarrow p_\theta(\{f_i\}, Q, \mathcal{K})$
if $\gamma > c_1$ **then**
  return $\hat{y}$
end if
for $i = 1$ to k **do**
  $I_i \leftarrow p_\theta(I_{i-1}, Q, \mathcal{K}, \mathit{Prompt}_{\text{expand}})$
  Sample $\{f_i\} \sim \text{Uniform}(I_i, n_f)$
  $(\hat{y}, \gamma) \leftarrow p_\theta(\{f_i\}, Q, \mathcal{K})$
  if $\gamma > c_1$ **then**
    return $\hat{y}$
  else if $\gamma > c_2$ **then**
    $d \leftarrow p_\theta(I_i, Q, \mathit{Prompt}_{\text{keyinfo}})$
    $\mathcal{K} \leftarrow \mathcal{K} \cup \{d\}$
  end if
end for
return $\hat{y}$

3 Method

3.1 Problem Setting

A VideoLLM, denoted as as p_θ takes as input a long-form video $V = \{f_1, f_2, \ldots, f_T\}$, represented as a sequence of frames, and a natural language query Q. The model generates a textual response $A = [a_1, \ldots, a_m]$ in an autoregressive manner, where each token a_t is sampled from the conditional distribution:

$$p_\theta(a_t \mid V, Q, a_{<t}) \propto \exp\left(\text{logit}_\theta(a_t \mid V, Q, a_{<t})\right), \tag{1}$$

where logit_θ denotes the unnormalized logit (i.e., pre-softmax score) assigned to token a_t given the context. Prior approaches often densely and uniformly sample frames from the entire video V, leading to memory-intensive computation that is prohibitively expensive under GPU memory constraints and the sparsity of relevant content over time. To this end, we reformulate long-video understanding as a temporal search problem, where the objective is to locate key temporal intervals in

Algorithm 2 Temporal Search with Best-First Tree-Based Search (TS-BFS)

Input: Video $V = \{f_1, \ldots, f_T\}$; query Q; steps k; thresholds c_1, c_2; intervals per expand n; number of frames per inference n_f; model p_θ
Initialization: $\mathcal{Q} \leftarrow \{\text{Node}(0, T)\}$, $\mathcal{K} \leftarrow \emptyset$
Sample $\{f_i\} \sim \text{Uniform}(0, T, n_f)$
$(\hat{y}, \gamma) \leftarrow p_\theta(\{f_i\}, Q, \mathcal{K})$
$v \leftarrow p_\theta(\{f_i\}, Prompt_{\text{evaluate}}, Q, \hat{y}, \mathcal{K})$
Root.update$(\hat{y}, v + \gamma)$
if $\gamma > c_1$ **then**
  return $\hat{y}$
end if
for $t = 1$ to k **do**
  $(s^*, e^*) \leftarrow \mathcal{Q}$.pop()
  Sample $\{f_i\} \sim \text{Uniform}(s^*, e^*, n_f)$
  $interval_{\text{heur}} \leftarrow p_\theta(\{f_i\}, Prompt_{\text{expand}}, Q, \mathcal{K})$
  $interval_{\text{uniform}} \leftarrow \text{UniformSplit}(s^*, e^*, n)$
  $\mathcal{C} \leftarrow interval_{\text{heur}} \cup interval_{\text{uniform}}$
  $\mathcal{S} \leftarrow \text{Sample}(\mathcal{C}, n)$
  for $(s', e') \in \mathcal{S}$ **do**
    Sample $\{f'_i\} \sim \text{Uniform}(s', e', n_f)$
    $(\hat{y}', \gamma') \leftarrow p_\theta(\{f'_i\}, Q, \mathcal{K})$
    $v' \leftarrow p_\theta(\{f'_i\}, Prompt_{\text{evaluate}}, Q, \hat{y}', \mathcal{K})$
    $\mathcal{Q}$.push(Node$(s', e', \hat{y}', v' + \gamma')$)
    if $\gamma' > c_2$ **then**
      $d \leftarrow p_\theta(\{f'_i\}, Prompt_{\text{keyinfo}}, Q, \mathcal{K})$
      $\mathcal{K} \leftarrow \mathcal{K} \cup \{d\}$
    else if $\gamma' > c_1$ **then**
      return $\hat{y}'$
    end if
  end for
end for
$\hat{n} \leftarrow \arg\max_{node \in \mathcal{Q}} node.\text{value}$
return $\hat{n}$.pred

I, ultimately enabling accurate answer generation. Let $I = \{f_i, \ldots, f_j\}$ denote a candidate interval. Given I and Q, the model generates a response $\hat{A} = [\hat{a}_1, \ldots, \hat{a}_m]$ along with a confidence score that reflects the model's certainty in the generated answer. This score is computed as the average log-probability of the generated tokens:

$$\text{Conf}(I, Q) = \frac{1}{m} \sum_{t=1}^{m} \log p_\theta(\hat{a}_t \mid I, Q, \hat{a}_{<t}) \quad (2)$$

We empirically observe that higher generation confidence over a temporal interval correlates with greater prediction accuracy, making it a reliable signal for guiding segment-level search in long videos. We also design three prompts through prompt engineering, `Expand_Prompt`, `Evaluate_Prompt`, and `Key_Frame_Prompt` to guide node expansion, evaluation, and keyframe description generation as shown in Figure 3. Keyframe descriptions will be integrated into the input prompt, enabling the VideoLLM to perceive the video globally.

Expand_Prompt: Based on the visual and contextual information provided above. your task is to simulate a human-like visual search strategy and propose {} distinct temporal interval, each in the form [start_frame, end_frame]) that are likely to provide useful visual evidence for answering the question.

Evaluate_Prompt: Based on the visual and contextual information provided above, do you think the predicted answer is correct? Respond with a single word: Yes or No.

Key_Frame_Prompt: Your task is to enrich the existing video description by adding or updating frame-level observations specifically relevant to the question.

Figure 3: Prompts for Tree Search.

3.2 Temporal Search Framework

To efficiently explore and refine relevant temporal regions in long videos, we propose Temporal Search (TS), a self-refine framework that progressively focuses attention on informative segments. TS integrates two zoom-in strategies:

1. **Sequential Temporal Search (TS):** A lightweight, step-wise approach that proposes one new temporal interval at each step and directly refines the response within it.

2. **Best-First Tree-Based Search (TS-BFS):** A tree search framework that proposes multiple candidate intervals per step, evaluates them, and selects the most promising one for response refinement and further exploration.

Sequential Temporal Search. The vanilla TS strategy starts by treating the entire video as the initial interval $I_0 = [0, T]$, where T is the total number of frames. The model p_θ generate an initial prediction $\hat{y}_0$ and a corresponding confidence score $\gamma_0 = \text{Conf}(I_0, Q)$, given a query Q. If $\gamma_0 \leq c_1$, the search is refined by proposing a new interval $I_1 = [s_1, e_1]$. The model then uniformly samples n frames from I_1, performs inference again, and outputs an updated prediction $\hat{y}_1$ along with a new confidence score $\gamma_1 = \text{Conf}(I_1, Q)$. This refinement process is repeated iteratively: at each step i, a new interval I_i is proposed based on prior predictions, and the model generates a corresponding $\hat{y}_i$ and γ_i. It also collects keyframe-level descriptions to enhance the overall comprehension. The

search terminates when either $\gamma_i > c_1$ or a maximum number of iterations k is reached as shown in Algorithm 1.

Best-First Tree-Based Search (TS-BFS). To further enhance the initial zoom-in process, we introduce a best-first tree-based search strategy (TS-BFS) as shown in Algorithm 2. Each node in TS-BFS represents a temporal interval defined by its start and end frame indices $[s, e]$, along with a predicted response and associated confidence score. The frames is uniformly sampled from each segment, regardless of its duration, which ensures that longer intervals provide a coarse summary, while shorter intervals capture finer temporal details. More promising intervals are prioritised for zoom-in, with the top-score node selected for expansion. The key steps of this process are detailed below.

Node Expansion. To progressively focus on informative content, a selected node $[s, e]$ is expanded into child intervals using two strategies:

- **Heuristic Proposal:** The VideoLLM proposes n sub-intervals based on its internal assessment of where relevant information may reside with `Expand_Prompt`.
- **Uniform Partitioning:** The interval is evenly split into n sub-segments.

Each resulting sub-interval becomes a new node and is evaluated in the same manner.

Node Evaluation. For each node, frames are uniformly sampled and input into the VideoLLM, which generates a predicted answer y and a confidence score c. Empirically, we observe that higher confidence scores are strongly correlated with greater answer accuracy. Thus, c is used as a value metric for prioritizing nodes. The confidence score is derived from the model's token-level probabilities. For a predicted answer $y = [y_1, \ldots, y_m]$, the confidence score is computed as $\mathrm{Conf}(S, Q)$, where S denotes the temporal segment associated with the node, and Q is the query. In addition, `Evaluate_Prompt` is used to instruct the VideoLLM to assess the quality of the generated answer. Specifically, the model is prompted to produce a binary judgment (e.g., `yes` or `no`), and the probability assigned to the `yes` response is taken as the evaluation score. The overall value score is as follows:

$$\mathrm{Value}(S, Q) = \cdot\,\mathrm{Conf}(S, Q) + \cdot\,\mathrm{Eval}(y)$$

Global Keyframe Memory. For each node, the confidence score from node evaluation is used to decide whether keyframe descriptions should be generated. If the score exceeds a predefined threshold, `Key_Frame_Prompt` is used to instruct the VideoLLM to produce textual descriptions of the sampled frames. These descriptions are stored in a global keyframe memory to maintain cross-segment awareness and support future expansions.

Termination Criteria. The search continues until a predefined budget is exhausted, such as a maximum number of iterations or when a segment's prediction confidence exceeds a specified threshold. The final output with their corresponding responses, which can be used for downstream tasks such as video question answering. By allocating computation to high-value temporal regions, TS enables fine-grained understanding of long videos in a resource-efficient manner.

Model	Video-Level		Interval-Level	
	Cor.	Err.	Cor.	Err.
Qwen-VL-2.5	0.74	0.53	0.72	0.53
LLaVA-OV	0.79	0.63	0.77	0.63
LLaVA-Video	0.75	0.55	0.73	0.56

Table 1: Correctness and error confidence scores at video-level and interval-level predictions on VideoMME.

4 Experiments

Experimental Setup Temporal Search is configured with default settings: 5 iterations ($k = 5$), 6 expansions per step ($n = 6$), stopping thresholds $c_1 = 0.9$ and $c_2 = 0.7$, and 8 frames ($n_f = 8$) uniformly sampled per segment. We evaluate on two long-form video QA benchmarks: LongVideoBench (Wu et al., 2024a) and VideoMME (Fu et al., 2024), both featuring multi-choice questions over videos up to one hour long, posing significant challenges for efficient temporal understanding. Following the LMMEval protocol (Zhang et al., 2024b), we disable subtitles during inference to isolate visual understanding and avoid reliance on textual priors. We apply TS to three representative video-based MLLMs: Qwen-VL-2.5 (Bai et al., 2025), LLaVA-OV (Li et al.,

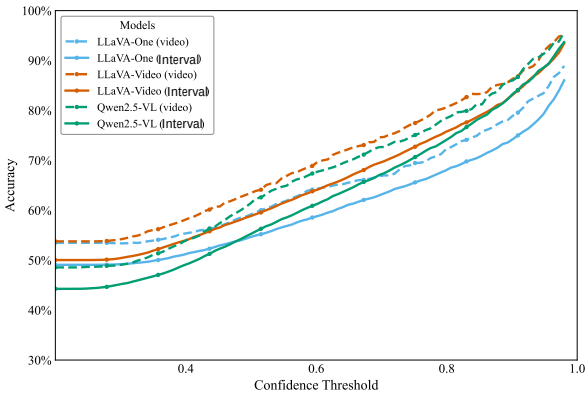

Figure 4: Relationship between model confidence and accuracy at the video and interval levels. Higher confidence thresholds consistently correspond to higher accuracy, indicating that model confidence serves as a reliable signal for guiding temporal interval selection. Each point represents the accuracy of predictions whose confidence exceeds the corresponding threshold.

Method	Frames / Inf.	LVB	V-MME
Closed-source Models			
GPT-4V (Achiam et al., 2023)	256	61.3	59.9
GPT-4o (Achiam et al., 2023)	256	66.7	71.9
Gemini-1.5-Pro (Team et al., 2024)	256	64.0	75.0
Gemini-1.5-Flash (Team et al., 2024)	256	61.6	70.3
Open-source Models			
TimeMarker-8B (Chen et al., 2024a)	128	56.3	57.3
LongVILA-7B (Chen et al., 2024b)	256	57.7	52.1
PLLaVA-34B (Xu et al., 2024)	32	53.2	–
Video-R1-7B (Feng et al., 2025)	64	–	52.2
Qwen2-VL-7B (Wang et al., 2024b)	256	55.6	–
Qwen2-VL-7B w/ TS	16	56.6	–
LLaVA-Video (7B)			
Uniform Sampling	8	53.5	53.7
Uniform Temporal Voting	8	54.4	54.7
TS	8	56.3	55.6
TS-BFS	8	**57.9**	**57.5**
LLaVA-OV (7B)			
Uniform Sampling	8	51.6	53.4
Uniform Temporal Voting	8	51.6	53.2
TS	8	53.9	55.4
TS-BFS	8	**54.4**	**56.9**
Qwen2.5-VL (7B)			
Uniform Sampling	8	51.5	48.5
Uniform Temporal Voting	8	50.8	49.3
TS	8	56.4	53.6
TS-BFS	8	**57.9**	**55.1**

Table 2: Comparison of different frame selection strategies on the LongVideoBench (LVB) and VideoMME (V-MME) benchmarks. "Frames / Inf." indicates the number of frames used per inference. Our proposed **Temporal Search** consistently outperforms other baselines across different architectures, demonstrating its general applicability.

2024a), and LLaVA-Video (Zhang et al., 2024d). All models follow a similar prompting format with a textual query, multiple answer options, and a fixed number of visual frames.

Baselines We compare against leading video understanding models (Achiam et al., 2023; Team et al., 2024; Chen et al., 2024b,a; Feng et al., 2025) and two sampling-based baselines. Uniform Sampling (US) uniformly samples frames from the entire video for single-pass inference, following standard protocols (Li et al., 2024a; Wu et al., 2024a). Existing VideoLLMs widely adopt this strategy due to its simplicity. Uniform Temporal Voting (UTV) evenly divides the video into temporal intervals, samples a fixed number of frames from each, and performs independent inferences over each interval. Predictions with above-mean confidence are then aggregated via majority voting (Wang et al., 2022). This serves as a static multi-inference baseline that lacks temporal awareness and iterative refinement.

4.1 Confidence–Accuracy Correlation

We evaluate whether model-predicted confidence aligns with answer correctness under two settings, to assess its reliability as a signal for guiding temporal search. We conduct our analysis at two levels:

- **Video-level:** A fixed number of frames are uniformly sampled across the entire video. The model produces predictions per video, and we analyze the relationship between confidence and accuracy at the global level.
- **Interval-level:** The video is divided into multiple temporal intervals. For each interval, the model generates an independent prediction. We then examine how confidence correlates with accuracy at the interval level.

For each prediction, we record the model's confidence and verify whether the predictions match the ground truth. Our analysis focuses on the relationship between confidence and answer accuracy.

Confidence–Accuracy Correlation. As shown in Table 1, all models consistently assign higher average confidence to correct predictions than to incorrect ones under both the video-level and interval-level settings. For instance, Qwen-VL-2.5 achieves an average confidence of 0.74 for correct answers, compared to 0.53 for incorrect ones under video-level sampling. Figure 4 further illustrates that accuracy steadily increases with higher confidence thresholds across all models and settings. These findings confirm that model confidence is a reliable indicator of prediction correctness, justifying its use in guiding temporal search for efficient long-video inference.

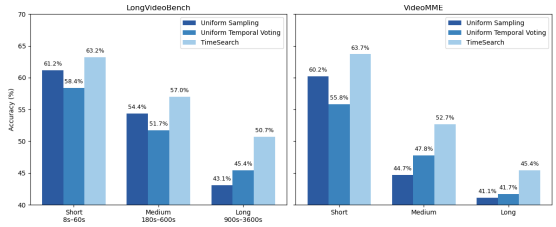

Figure 5: Accuracy on LongVideoBench and VideoMME by duration based on the Qwen2.5-Vl. TS improves accuracy across all durations, showing better generalization and robustness.

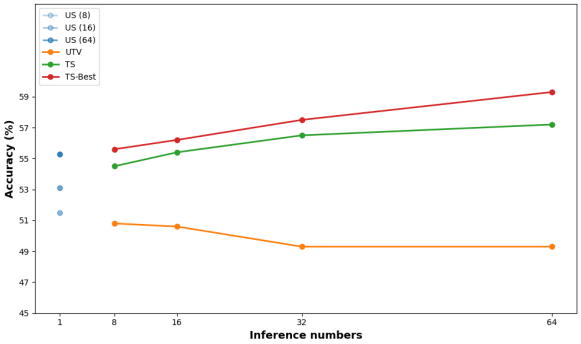

Figure 6: Model performance under different numbers of inference passes. Results are based on the Qwen2.5-VL on LongVideoBench.

4.2 Results on LongVideo Benchmark

As shown in Table 2, TS consistently outperforms US and UTV across all models on both LongVideoBench and VideoMME. Furthermore, incorporating best-first search over a temporal interval tree further improves performance by enabling more efficient exploration. For Qwen2.5-VL, TS-BFS improves accuracy from 51.5% to 57.9% on LongVideoBench and from 48.5% to 55.1% on VideoMME. Similar gains are observed for LLaVA-Video and LLaVA-OV, confirming the general effectiveness of TS. TS achieves competitive results with only 8 frames per inference, outperforming larger models like TimeMarker-8B with 128 frames and Video-XL with 256 frames. Qwen2-VL with TS, using multiple lightweight inferences with 16 frames, also exceeds its baseline of 256 frames. In contrast, UTV yields limited improvements, highlighting the need for adaptive search where the model actively perceives and proposes relevant temporal intervals rather than relying on static aggregation.

4.3 Inference Scaling

We investigate how model performance scales with the number of inference passes. As shown in Figure 6, Uniform Sampling (US) performs single-pass inference and shows modest improvement as the number of sampled frames increases. Although Uniform Temporal Voting (UTV) involves multiple inferences across temporal segments, it yields only marginal gains. In contrast, both TS and TS-BFS consistently improve accuracy with more inference steps, highlighting the effectiveness of confidence-guided temporal search. These results highlight the advantage of adaptive, iterative inference over static multi-pass aggregation, particularly when operating within fixed per-pass frame budgets. Moreover, the tree-based search in TS-BFS can be efficiently parallelised by evaluating multiple candidate intervals. For example, expanding six nodes per step enables six parallel inference passes to be executed in a single batch.

Frames	US	UTV	TS	TS-Best	Memory
8	51.5	50.8	56.4	**57.9**	17
16	53.1	49.2	57.8	**59.8**	22
32	55.3	54.7	59.0	**60.7**	39
64	55.3	55.5	60.1	**61.7**	120

Table 3: Accuracy and GPU memory usage (GB) of different sample methods under varying numbers of sampled frames per interval. Results are based on the Qwen2.5-VL on LongVideoBench.

4.4 Ablation studies

We conduct ablation studies to better understand the design choices and robustness of the proposed TS framework. Specifically, we examine the robustness of TS across different video durations, analyze the effect of the number of sampled frames per interval, and evaluate the influence of stopping conditions and search configurations.

Impact of Video Duration. We evaluate the robustness of TS across videos of varying lengths, including short, medium, and long durations. As shown in Figure 5, UTV provides slight improvements on long videos but degrades performance on short ones. In contrast, TS consistently enhances accuracy across all duration groups, demonstrating strong adaptability to different temporal scales. Specifically, on long videos, TS yields a 7.6% accuracy improvement on LongVideoBench and a 4.3% gain on VideoMME, highlighting its effectiveness in handling extended video.

Effect of Frames per Interval. We study how the number of sampled frames per interval affects

Model	c_2	c_1					
		0.5	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Qwen2.5-VL	**0.5**	53.6	55.0	55.2	54.7	53.3	55.6
	0.6	54.5	55.5	54.8	55.2	55.4	56.0
	0.7	52.9	54.5	56.2	56.2	55.7	56.6
	0.8	53.5	**55.0**	56.6	56.5	56.7	57.2
	0.9	**53.6**	54.8	**57.0**	**57.9**	**57.5**	57.4
	1.0	54.4	55.0	56.6	57.1	57.4	**58.2**
LLaVA-Video	**0.5**	55.3	**57.1**	56.6	55.7	56.9	55.9
	0.6	55.0	55.9	56.6	57.3	56.9	55.9
	0.7	55.9	55.6	56.6	57.5	58.3	57.6
	0.8	55.9	56.4	**56.7**	57.0	57.9	55.8
	0.9	**56.3**	56.5	56.5	57.9	58.3	57.6
	1.0	55.8	56.4	56.5	**58.3**	**58.4**	**58.8**

Table 4: Ablation of stopping thresholds c_1 and c_2 for Temporal Search on LongVideoBench.

Model	$k \setminus n$	**2**	**4**	**6**	**8**
Qwen2.5-VL	1	53.6	53.3	55.0	55.0
	5	55.4	56.3	56.2	56.2
	10	55.7	**56.4**	**57.9**	**56.7**
LLaVA-Video	1	55.9	56.2	56.7	56.7
	5	55.5	56.2	**57.9**	57.4
	10	**56.5**	56.4	57.8	**57.5**

Table 5: Ablation on the number of greedy search steps k and the number of node expansions n in Temporal Search (TS). Accuracy (%) is reported on the LongVideoBench benchmark.

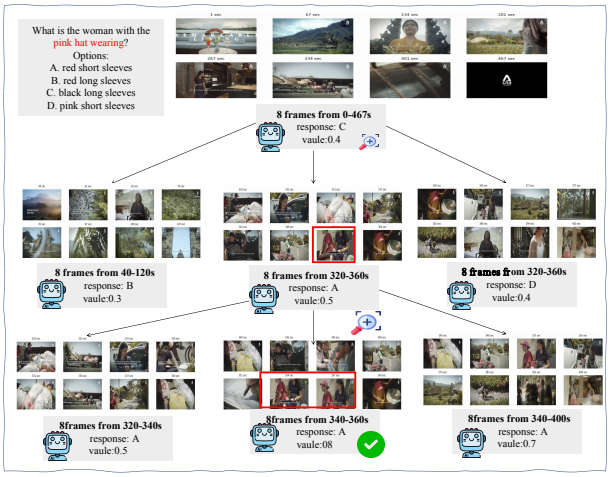

Figure 7: Illustration of the Temporal Search framework with iterative zoom-in on valuable temporal segments.

accuracy and memory usage. As shown in Table 3, using 8 frames may result in missing key visual cues within long intervals. Increasing the frame count to 16 provides noticeable accuracy gains, while further increases (32 or 64 frames) yield diminishing returns but significantly raise memory usage. Notably, TS-BFS with 16-frame input attains 59.8% accuracy using just 22GB, surpassing the 55.3% accuracy of US with 64 frames, which requires over 120GB memory. This demonstrates that TS achieves strong performance with high memory efficiency, making it practical for deployment on consumer GPUs.

Effect of Stopping Conditions and Search Depth. We examine how the stopping thresholds and search parameters affect performance. As shown in Table 4, moderately high values of the key-frame acceptance threshold c_1 help avoid premature convergence, while stricter early termination thresholds c_2 (1.0) promote more confident final decisions. In parallel, Table 5 analyses the impact of the number of search iterations k and node expansions per step n. Increasing both parameters generally improves accuracy, as a deeper and broader search improves the model's ability to identify informative intervals. Performance saturates around $k = 10$, $n = 6$, beyond which further gains diminish relative to the increased inference cost.

4.5 Case Analysis

We present a case from the LongVideoBench dataset to illustrate the behavior of our Temporal Search (TS) framework. The question is: "What is the woman with the pink hat wearing?" TS begins by exploring coarse-grained segments across the video. Based on model confidence, it iteratively refines its focus to more informative regions. In this example, TS converges on the time window around 350–365 seconds, where the target subject appears. These frames clearly reveal the woman in a pink hat wearing red long sleeves, allowing the model to confidently choose the correct answer: A.

5 Conclusion

In this work, we present Temporal Search (TS), a self-refine framework that enables MLLMs to iteratively locate and refine task-relevant intervals in long videos. TS leverages model confidence to guide temporal exploration and focuses on finer-grained regions through fixed-frame inputs. To enhance efficiency, we introduce TS-BFS, a best-first search strategy that prioritises high-confidence intervals. TS requires no model changes and improves performance across multiple benchmarks. Experiments show that TS-BFS significantly boosts accuracy on LongVideoBench and VideoMME, demonstrating its effectiveness and generality for long-video understanding.

Limitations

The Temporal Search (TS) framework offers significant improvements in long-form video understanding, but it has some limitations and potential, the performance of TS is affected by the quality of the sampled frames. In cases where the frames are of low quality or unclear, the model may fail to capture relevant information. Furthermore, the search depth and stopping thresholds influence the trade-off between accuracy and resource consumption. Regarding potential risks, reliance on model confidence might lead to misdirected focus on irrelevant segments. In sensitive applications, there is also a risk of focusing on private or personal content, raising ethical concerns.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal models. *arXiv preprint arXiv:2503.02175*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. 2024a. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*.

Xiuyuan Chen, Yuan Lin, Yuchen Zhang, and Weiran Huang. 2023. Autoeval-video: An automatic benchmark for assessing large vision language models in open-ended video question answering. *arXiv preprint arXiv:2311.14906*.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, and 1 others. 2024b. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*.

Chuanqi Cheng, Jian Guan, Wei Wu, and Rui Yan. 2025. Scaling video-language models to 10k frames via hierarchical differential distillation. *arXiv preprint arXiv:2504.02438*.

Rina Dechter and Judea Pearl. 1985. Generalized best-first search strategies and the optimality of a. *Journal of the ACM (JACM)*, 32(3):505–536.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. 2024. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *arXiv preprint arXiv:2406.14515*.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. 2025. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*.

Shane Frederick. 2005. Cognitive reflection and decision making. *Journal of Economic perspectives*, 19(4):25–42.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Suyuan Huang, Haoxin Zhang, Yan Gao, Yao Hu, and Zengchang Qin. 2024. From image to video, what do we need in multimodal llms? *arXiv preprint arXiv:2404.11865*.

Richard E Korf. 1985. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, 27(1):97–109.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. 2025. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9098–9108.

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

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Yunxin Li, Xinyu Chen, Baotian Hu, Longyue Wang, Haoyuan Shi, and Min Zhang. 2024c. Videovista: A versatile benchmark for video understanding and reasoning. *arXiv preprint arXiv:2406.11303*.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual

representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Ruyang Liu, Haoran Tang, Haibo Liu, Yixiao Ge, Ying Shan, Chen Li, and Jiankun Yang. 2024. Ppllava: Varied video sequence understanding with prompt guidance. *arXiv preprint arXiv:2411.02327*.

Thong Nguyen, Yi Bin, Junbin Xiao, Leigang Qu, Yicong Li, Jay Zhangjie Wu, Cong-Duy Nguyen, See-Kiong Ng, and Luu Anh Tuan. 2024. Video-language understanding: A survey from model architecture, model training, and data perspectives. *arXiv preprint arXiv:2406.05615*.

Yuxiang Nie, Han Wang, Yanjie Wang, Can Huang, Liang Lin, and Guanbin Li. Video q-former: Multimodal large language model with spatio-temporal querying transformer towards video understanding.

Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. 2023. Videobench: A comprehensive benchmark and toolkit for evaluating video-based large language models. *arXiv preprint arXiv:2311.16103*.

Ray Paternoster and Greg Pogarsky. 2009. Rational choice, agency and thoughtfully reflective decision making: The short and long-term consequences of making good choices. *Journal of Quantitative Criminology*, 25:103–127.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. 2024. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. *arXiv preprint arXiv:2411.16044*.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, and 1 others. 2025. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Han Wang, Yuxiang Nie, Yongjie Ye, Deng GuanYu, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. 2024a. Dynamic-vlm: Simple dynamic visual token compression for videollm. *arXiv preprint arXiv:2412.09530*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024a. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857.

Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. 2024b. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. *Advances in Neural Information Processing Systems*, 37:109922–109947.

Junbin Xiao, Nanxin Huang, Hangyu Qin, Dongyang Li, Yicong Li, Fengbin Zhu, Zhulin Tao, Jianxing Yu, Liang Lin, Tat-Seng Chua, and 1 others. 2025. Videoqa in the era of llms: An empirical study. *International Journal of Computer Vision*, pages 1–24.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134.

Jeffrey M Zacks and Khena M Swallow. 2007. Event segmentation. *Current directions in psychological science*, 16(2):80–84.

Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024a. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language

model for video understanding. *arXiv preprint arXiv:2306.02858*.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024b. Lmms-eval: Reality check on the evaluation of large multimodal models. *Preprint*, arXiv:2407.12772.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024c. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024d. Video instruction tuning with synthetic data. *Preprint*, arXiv:2410.02713.