$\mathbf{T^2HTR}$: TEST-TIME HIERARCHICAL TEMPORAL RETRIEVAL FOR LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Foundational Multi-modal Large Language Models (MLLMs) have achieved rapid progress in handling complex tasks across diverse modalities. However, they still struggle to deliver satisfactory performance on Long-video Understanding (LVU) tasks involving thousands of frames. Existing optimization strategies can be broadly categorized into LVU-specific fine-tuning, built-in token compression and training-free keyframe extraction, with the latter being most suitable for flexible deployment across various MLLMs. Unfortunately, current training-free approaches predominantly focus on query-frame relevance retrieval, overlooking other levels of visual information and the inherent heterogeneity of LVU tasks. In this work, we propose the **T**est-**t**ime **H**ierarchical **T**emporal **R**etrieval ($\mathbf{T^2HTR}$) framework, which employs a multi-stage pipeline, including dual scene segmentation, joint score calculation, sub-scene window modeling and dynamic mask-based inference, to extract distinct keyframes sets from the perspectives of relevance, summarization and causality. These keyframes are then blended at varying ratios to construct multiple video sampling pools. Guided by adaptive feedback from the model, $\mathbf{T^2HTR}$ dynamically routes each sample to its optimal video pool, enabling more precise and sample-grained LVU. Extensive experiments demonstrate the advanced performance of our scheme across multiple challenging LVU benchmarks. For instance, integrating $\mathbf{T^2HTR}$ with Qwen-2.5-VL yields performance gains of 3.5% to 13.1% on LVB, VideoMME and MLVU.

1 INTRODUCTION

Long-video Understanding (LVU), as one of the most challenging multi-modal reasoning tasks, has emerged as a critical benchmark for evaluating the advanced capabilities of Multi-modal Large Language Model (MLLM). While current foundational MLLMs (Zhang et al., 2024b; Bai et al., 2025; Chen et al., 2024b) demonstrate strong generalization across diverse multi-modal tasks, they still struggle when directly applied to complex or fine-grained reasoning tasks grounded in long-form videos. To address this limitation, as illustrated in Fig.1(a)-(c), prior works have explored three primary routes: first, researchers have attempted to train specialized MLLMs or fine-tune components of existing models specifically for long-video processing (Chen et al., 2024a; Shen et al., 2024; Zohar et al., 2025; Islam et al., 2025). However, due to the rich content diversity and enormous visual token budget inherent in long videos, such approaches demand prohibitively high training overhead. Moreover, the scarcity of supervised fine-tuning data often fails to adequately cover the full spectrum of video semantics, limiting the generalizability of these improved models.

Furthermore, some efforts have also focused on integrating token compression techniques within MLLMs to expand their effective context capacity (Cheng et al., 2025; Wang et al., 2025a; Gao et al., 2025), thereby enabling the model to ingest more visual frames. Yet, since different MLLMs employ heterogeneous architectures, these compression methods may lack cross-model compatibility, constraining their broad applicability. In addition, and most recently, a promising line of research has proposed externalizing keyframe extraction as a training-free pre-processing pipeline – decoupling it from the MLLM's core architecture (Tang et al., 2025b; Ye et al., 2025). This approach leverages the strengths of pre-trained models while remaining architecture-agnostic (Wang et al., 2025b; Xu et al., 2025), thereby gaining increasing attention in the community.

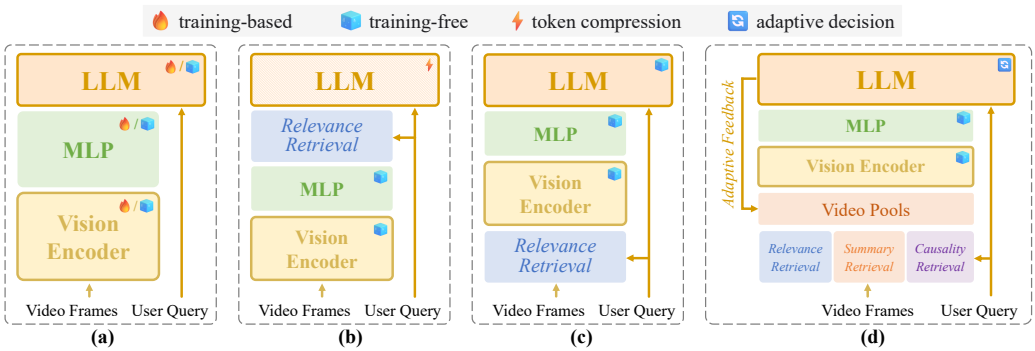


Figure 1: Comparison of different optimization schemes for LVU. (a): feeding uniformly-sampled video frames into (fine-tuned) MLLMs; (b): built-in visual token compression for MLLMs; (c): performing relevance retrieval for the video and feeding the obtained keyframe set into MLLMs; (d): **T^{2}HTR** framework mixes keyframe sets obtained from multiple perspectives and feeds them into MLLMs, with the specific blending ratio (video pool) regulated by the model's adaptive feedback.

However, current keyframe extraction methods predominantly rely on relevance evaluation via CLIP-like models (Radford et al., 2021), which tend to concentrate high scores over narrow temporal segments, resulting in keyframe sets that lack comprehensive event-level and global contextual coverage. Furthermore, beyond mere relevance, we argue that video frames forming causal relationships with the most relevant frames (*i.e.* context supporting evidence) are crucial for accurate reasoning but are largely ignored. Additionally, the optimal scope of keyframe extraction is intrinsically linked to the nature of the user query. For example, queries requiring fine-grained visual details benefit from localized, highly relevant video clips, whereas those involving complex reasoning or event-level understanding necessitate broader and distributed context aggregation, which is a distinction frequently overlooked in prior works.

In this work, we attempt to addresses these limitations mentioned above through the following steps, as shown in Fig.1(d): (i) beyond direct relevance-based retrieval, we further construct distinct keyframe sets from the perspectives of summarization and causality; (ii) these keyframe sets, derived from heterogeneous sources, are then blended at multiple ratios to generate diverse video sampling pools in batch; (iii) leveraging the adaptive feedback capability of MLLMs, we enable fine-grained exploration over these video pools. Our contributions are summarized as follows:

- We propose a unified pipeline for multi-perspective keyframe extraction, integrating hierarchical scene segmentation with relevance scoring, sub-scene window modeling and mask-based causal reasoning to capture complementary visual semantics.
- We point out that video sampling pools constructed under varying mixture ratios are shown to specialize in distinct query intents, thereby refining LVU optimization to the sample granularity.
- To further harness MLLM capabilities, we introduce a family of adaptive and closed-loop pool selection schemes that dynamically refine sampling beyond fixed-ratio baselines.
- Experiments have demonstrated the superiority of our framework: for instance, we outperform a wide range of both proprietary and open-source MLLMs on LVB and MLVU benchmarks.

2 Related Works

Mainstream Foundational MLLMs. Recent advances in foundational MLLMs, including model families such as GPT-4o (Hurst et al., 2024), LLaVA (Li et al., 2024; Zhang et al., 2024b), Qwen-VL (Bai et al., 2025), InternVL (Chen et al., 2024b) and NVILA (Liu et al., 2025b), have demonstrated remarkable capabilities across diverse multi-modal tasks. These models are typically trained through multi-stage Pre-training and Instruction Fine-tuning on heterogeneous data modalities, including image, text, video, chart, document, mathematical expressions and Graphical User Inter-

faces (GUIs), enabling them to perform complex operations such as visual question answering, cross-modal grounding, temporal localization and structured reasoning. Despite their broad generalization power, when applied to LVU, these models still exhibit a substantial performance gap relative to human-level evaluation when simply applying uniform frame sampling and directly feeding the resulting frame sequence.

Training-based Models for LVU. Supervised Fine-tuning (SFT) of MLLMs remains a dominant strategy for enhancing the reasoning capability and performance on LVU tasks (Chen et al., 2024a; Shen et al., 2024). Frame-Voyager (Yu et al., 2024) leverages the prediction loss of a pre-trained MLLM to collect high-quality training data from diverse frame combinations, enabling the learning of an automated scoring component for keyframe selection. Hu et al. (2025b) annotate video samples with dual pseudo labels (spatial and temporal) to train a lightweight frame selector tailored for LVU. Zohar et al. (2025) systematically distill empirical guidelines for model training and inference in the domain of LVU, based on which they propose the family of Apollo models that can effectively address long-range temporal understanding. In addition, BIMBA (Islam et al., 2025) employs a state-space model with selective scan mechanisms to dynamically transmit only those most informative tokens to the language decoder. Similarly, ViLaMP (Cheng et al., 2025) integrates differential keyframe selection with weighted feature fusion, significantly suppressing temporal redundancy while retaining critical visual semantics. It is worth noting that recent works have further explored the integration of Reinforcement Learning (RL) with keyframe extraction to enable preference-aware optimization. Li et al. (2025) generate contrastive response pairs based on query-frame relevance, then jointly apply SFT and Direct Preference Optimization (DPO) to align the MLLM's outputs with human-like reasoning patterns. Inspired by the idea of Group Relative Policy Optimization (GRPO) algorithm, TSPO (Tang et al., 2025a) constructs a minimal-parameter Temporal Agent trained selectively on two challenging benchmarks (comprehensive temporal understanding and Needle-in-a-Haystack), demonstrating improved long-video reasoning through lightweight policy adaptation.

Training-free Keyframe Extraction. Training-free approaches can be broadly categorized into three paradigms based on their optimization logic: Pre-processing, Built-in Compression and Iterative Refinement. Pre-processing methods aim to deliver a curated set of keyframes to the MLLM prior to inference: VideoTree (Wang et al., 2025b) organizes long videos into hierarchical tree structures via clustering, enabling top-down keyframe search with spatial-temporal coherence. CoS (Hu et al., 2025a) first encodes video stream into binary coding to perform pseudo temporal grounding, then feeds them into the MLLM for co-reasoning. BOLT (Liu et al., 2025a) and AKS (Tang et al., 2025b) respectively identify the cumulative distribution and local peaks along the CLIP-based relevance score curve to efficiently extract salient frames without exhaustive scanning, while Ye et al. (2025) perform multi-round temporal search using keyword-driven object detection, dynamically updating the relevance distribution of the frame sequence. Nar-KFC (Fang et al., 2025) constructs interleaved image-text streams that preserve both semantic relevance and temporal continuity, enhancing the MLLM's ability to track evolving events.

In contrast, Built-in Compression methods focus on increasing the token capacity and efficiency of MLLMs during inference. SF-LLaVA (Xu et al., 2024) introduces parallel token transmission channels with varying sampling rates and pooling intensities, achieving a free lunch on performance improvement. AdaRETAKE (Wang et al., 2025a) performs adaptive token compression over both the temporal dimension and transformer layers, dynamically allocating compression ratios to enable efficient processing of thousands of frames in a single pass. APVR (Gao et al., 2025) unifies query-aware semantic expansion with adaptive visual token selection, performing hierarchical key information extraction at both frame and token levels.

Iterative Refinement methods exploit the MLLM's intrinsic reasoning and self-reflection capabilities to dynamically adjust the keyframe set during inference. VideoAgent (Wang et al., 2024), as an early pioneer, iteratively selects frames based on the LLM's confidence level over current answers and associated captions. Ma et al. (2025) reformulates long-video understanding as a long-document retrieval task, employing multi-stage agent interaction to progressively refine the quality of retrieved contents. E-VRAG (Xu et al., 2025) explores multi-round self-reflection mechanisms within MLLMs, combined with hierarchical filtering of video content, to achieve effective long-video comprehension with relatively low computational overhead.

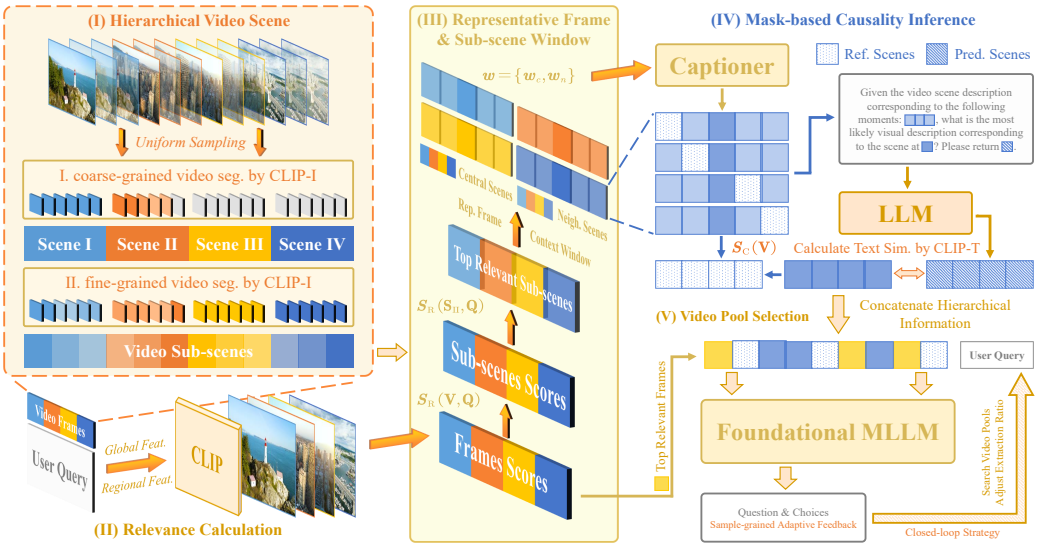


Figure 2: An overview of $\mathbf{T^2HTR}$ framework: for the given video and user query, we sequentially perform dual scene construction (left-top), joint relevance calculation (left-bottom), Representative frame and sub-scene window modeling (middle), causality evaluation (right-top) and closed-loop video pool selection (right-bottom).

3 METHODS

In this section, we will provide a detailed explanation for our $\mathbf{T^2HTR}$ framework, which consists of video scene construction (§3.1), frame-based relevance calculation & scene-based causality evaluation (§3.2) and the adaptive strategy for selecting video pools (§3.3-3.4).

3.1 VIDEO SCENE CONSTRUCTION

Coarse-grained Scene Segmentation. We first utilize the pre-trained CLIP-Series model to perform frame-by-frame feature extraction for the given video sample $\mathbf{V}_{1:T} \in \mathbb{R}^{T\times C\times H\times W}$ (T, C, H, W are the number of frames and channels, height and width, respectively), here $\mathbf{F}_{1:T} = \text{CLIP}_\text{I}(\mathbf{V}_{1:T}) \in \mathbb{R}^{T\times D}$ denotes the visual features processed by the vision tower and D is the number of feature dimensions.

For a given scene starting index $\tilde{t}$, we define a judgment function $\boldsymbol{J}_\text{I}(\cdot)$, which is used to confirm whether the index interval $[\tilde{t}, t]$ will be split into an independent video scene. Considering the possibility of brief shot switching in the same scene, we extend the current frame index t backward by a small amount to obtain t^* during scene segmentation. That is to say, as long as there is visual description related to the existing scene content in $(t, t^*]$, even if the correlation between $\mathbf{F}_{\tilde{t}:t}$ and $\mathbf{F}_t$ is not enough, we still consider $\mathbf{V}_t$ as a shot switching in the same scene, rather than immediately cutting $[\tilde{t}, t]$ into a new scene, as shown in Eq.(1). Θ_I represents cosine similarity threshold for scene segmentation.

$$\boldsymbol{J}_\text{I}(\tilde{t}, t) := \left(\max_{i\in[\tilde{t},t),\ j\in[t,t^*]} \frac{\mathbf{F}_i \cdot \mathbf{F}_j}{\|\mathbf{F}_i\|\|\mathbf{F}_j\|} < \Theta_\text{I} \right). \tag{1}$$

Fine-grained Sub-scene Segmentation. The above scene segmentation has preliminarily identified the relevant locations where visual semantics have changed. However, on the one hand, there may still be some visual detail changes in the scene at this time; on the other hand, a series of consecutive video frames contain a large amount of redundant information. Therefore, we further perform sub-scene segmentation within each scene.

$$\boldsymbol{J}_\text{II}(i, t) := \left(t \in \arg_k \min_{j\in\mathbf{S}_\text{I}^i} \frac{\mathbf{F}_j \cdot \mathbf{F}_{j+1}}{\|\mathbf{F}_j\|\|\mathbf{F}_{j+1}\|} \ \wedge \ \frac{\mathbf{F}_t \cdot \mathbf{F}_{t+1}}{\|\mathbf{F}_t\|\|\mathbf{F}_{t+1}\|} < \Theta_\text{II} \right). \tag{2}$$

Specifically, as shown in Eq.(2), we segment the current scene $\mathbf{S}_{\mathrm{I}}^i$ for k times based on the expected average sub-scene length, with the splitting positions being the indices corresponding to the bottom-k similarities between adjacent frames, here $\boldsymbol{J}_{\mathrm{II}}(i,t)$ represents whether sub-scene segmentation will be performed between the t-th and $t+1$-th frames in $\mathbf{S}_{\mathrm{I}}^i$. In addition, for positions with minimal visual changes (*i.e.* $\frac{\mathbf{F}_t \cdot \mathbf{F}_{t+1}}{\|\mathbf{F}_t\|\|\mathbf{F}_{t+1}\|} < \Theta_{\mathrm{II}}$), we will skip segmentation.

3.2 Frame Assessment for Relevance and Causality

Relevance Calculation. For a given user query and video sample, we can use the CLIP-Series model to achieve frame-level relevance retrieval. In addition to global-level visual semantics, regional-level visual semantics $\hat{\mathbf{F}}_{1:T} = \mathrm{CLIP}_{\mathrm{I}}(\mathbf{V}_{1:T}) \in \mathbb{R}^{T \times p^2 \times D}$ (p^2 is the number of visual patches) also plays a significant role as it may form corresponding relationships with keywords in the user query. Therefore, we propose a hybrid retrieval scheme based on global and regional semantics to achieve more discriminative scoring.

$$\boldsymbol{S}_{\mathrm{R}}(\mathbf{V}_t, \mathbf{Q}) = \alpha \cdot \frac{\mathbf{F}_t \cdot \mathbf{Q}}{\|\mathbf{F}_t\|\|\mathbf{Q}\|} + (1-\alpha) \sum_i^{p^2} \frac{\hat{\mathbf{F}}_{t,i} \cdot \mathbf{Q}}{\|\hat{\mathbf{F}}_{t,i}\|\|\mathbf{Q}\|}. \tag{3}$$

Here $\mathbf{Q} \in \mathbb{R}^{1 \times D}$ is the textual feature processed by the text tower and $\alpha \in [0,1]$ controls the importance degree of the above two calculation schemes. According to the frame-level relevance scores, we can further obtain the relevance score $\boldsymbol{S}_{\mathrm{R}}(\mathbf{S}_{\mathrm{II}}^i, \mathbf{Q}) = \sum_{t \in \mathbf{S}_{\mathrm{II}}^i} \boldsymbol{S}_{\mathrm{R}}(\mathbf{V}_t, \mathbf{Q})$ for any sub-scene. The frames within the same sub-scene usually have extremely high similarity, so we can consider extracting one Representative (abbreviated as Rep.) frame for each sub-scene, which not only effectively reduces temporal redundancy, but also facilitates efficient processing in subsequent steps.

$$\boldsymbol{J}_{\mathrm{III}}(i,t) := \left(\boldsymbol{S}_{\mathrm{R}}(\mathbf{V}_t, \mathbf{Q}) = \max_{j \in \mathbf{S}_{\mathrm{II}}^i} \boldsymbol{S}_{\mathrm{R}}(\mathbf{V}_j, \mathbf{Q}) \ \wedge \ \boldsymbol{S}_{\mathrm{R}}(\mathbf{S}_{\mathrm{II}}^i, \mathbf{Q}) \in \arg_k \max \boldsymbol{S}_{\mathrm{R}}(\mathbf{S}_{\mathrm{II}}, \mathbf{Q}) \right). \tag{4}$$

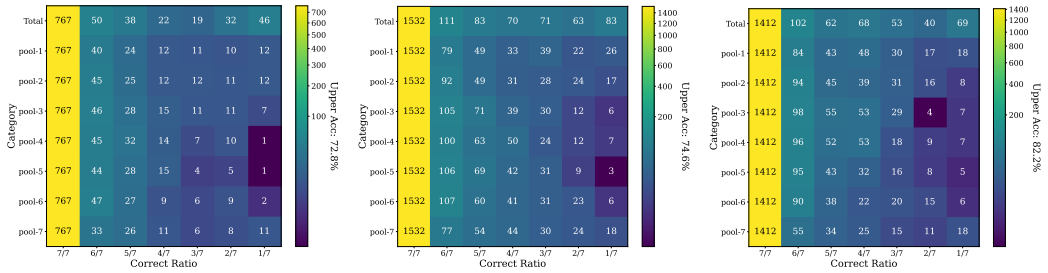
In Eq.(4), $\boldsymbol{J}_{\mathrm{III}}(i,t)$ denotes whether the t-th frame is the Rep. frame of $\mathbf{S}_{\mathrm{II}}^i$. Here we choose the frame with the highest relevance score within the sub-scene as the corresponding Rep. frame.

Modeling of Sub-scene Windows. Generally, high-relevance frames are considered to have a higher probability of directly pointing to the answer of the user query. However, these frames are generally distributed in a few local locations in the form of video clip. The frames near these frames may contain causal content of related events, serving as context to assist video understanding. Therefore, how to construct and filter causal frames becomes a critical problem.

Our specific approach is as follows: Firstly, we extract top-k relevant sub-scenes and utilize these sub-scenes as the central anchor points to expand forward and backward, thereby obtaining corresponding k sub-scene windows. Subsequently, according to Eq.(4), we make the window consist of Rep. frames corresponding to each sub scene. For example, for a $(2n+1)$-window $\boldsymbol{w}$ with $\mathbf{S}_{\mathrm{II}}^i$ as the central anchor point $\boldsymbol{w}_c$, we can write it as $\boldsymbol{w} = \{\mathbf{V}_t | \boldsymbol{J}_{\mathrm{III}}(j,t) = 1, j \in [i-n, i+n]\}$. We use $\boldsymbol{w}_n = \boldsymbol{w} \setminus \{\boldsymbol{w}_c\}$ to represent the contextual set composed of neighboring sub-scenes within the window. Our goal is to identify which sub-scenes in $\boldsymbol{w}_n$ have stronger causal relationships with $\boldsymbol{w}_c$, which can better facilitate MLLM's understanding of visual content in keyframes.

Causality Evaluation. We use a Captioner model to generate captions for Rep. frames in each sub-scene window and request that the captions should highlight content related to the user query. For each $(2n+1)$-window $\boldsymbol{w}$, we can construct a set of examples with batch-size equal to $2n$: $\{\mathbf{V}_i : \boldsymbol{w}_n \setminus \{\mathbf{V}_i\}, \forall \mathbf{V}_i \in \boldsymbol{w}_n\}$. Here, we select a frame $\mathbf{V}_i$ to be evaluated each time and pass the caption set corresponding to $\boldsymbol{w}_n \setminus \{\mathbf{V}_i\}$ into LLM to infer the visual description of $\boldsymbol{w}_c$. Then, we compare the inferred caption with the actual caption generated by Captioner before. If there is a significant difference, it indicates a strong causal dependence between $\mathbf{V}_i$ and $\boldsymbol{w}_c$ and we need to refer to $\mathbf{V}_i$ to better understand $\boldsymbol{w}_c$.

$$\boldsymbol{S}_{\mathrm{C}}(\mathbf{V}_t) = \sum_{\boldsymbol{w} | \mathbf{V}_t \in \boldsymbol{w}_n} \sqrt{1 - \left(\frac{\tilde{\mathbf{F}}_{\mathrm{LLM}(\boldsymbol{w}_c | \mathrm{Captioner}(\boldsymbol{w}_n \setminus \{\mathbf{V}_t\}))} \cdot \tilde{\mathbf{F}}_{\mathrm{Captioner}(\boldsymbol{w}_c)}}{\|\tilde{\mathbf{F}}_{\mathrm{LLM}(\boldsymbol{w}_c | \mathrm{Captioner}(\boldsymbol{w}_n \setminus \{\mathbf{V}_t\}))}\| \|\tilde{\mathbf{F}}_{\mathrm{Captioner}(\boldsymbol{w}_c)}\|} \right)^2}. \tag{5}$$



(a) LVB, Qwen2.5-VL-7B (b) VideoMME, Qwen2.5-VL-7B (c) MLVU, Qwen2.5-VL-7B

Figure 3: Distribution of correctly answered samples in multiple video sampling pools. Here the j-th column of pool-i is the number of samples answered correctly by j video pools (including pool-i) simultaneously; **Upper Acc.** denotes the accuracy upper-bound of video pools, which means that the corresponding sample is considered correct as long as any pool answers correctly.

Eq.(5) points out the specific calculation scheme for causal scores. For a given frame $\mathbf{V}_t$, we integrate the importance levels of $\mathbf{V}_t$ in all its relevant windows $\{\boldsymbol{w}|\mathbf{V}_t \in \boldsymbol{w}_n\}$. $\tilde{\mathbf{F}}_{\text{LLM}(\boldsymbol{w}_c|\text{Captioner}(\boldsymbol{w}_n \backslash \{\mathbf{V}_t\}))}, \tilde{\mathbf{F}}_{\text{Captioner}(\boldsymbol{w}_c)}$ denote the textual features processed by CLIP_T for LLM's and Captioner's captions, respectively.

3.3 The Construction of Video Sampling Pools

From the above procedures, we can obtain video frames from three sources: among them, high-ranking frames sorted by $\boldsymbol{S}_\text{R}(\cdot, \mathbf{Q})$ tend to focus on strong-correlated and locally continuous visual information, Rep. frames of video sub-scenes reflect high-quality visual summary content, while causal frames reveal contextual content with auxiliary reference value. By mixing these three different types of frames in a certain proportion, we integrate a total set of keyframes for the given video.

$$\mathbf{P}_{i:j} = \arg_i \max_{t\in[1,T]} \boldsymbol{S}_\text{R}(\mathbf{V}_t, \mathbf{Q}) \cup \left\{ \boldsymbol{w}_c, \arg \max_{\mathbf{V}_t \in \boldsymbol{w}_n} \boldsymbol{S}_\text{C}(\mathbf{V}_t) | \boldsymbol{w} \in \arg_j \max_{\boldsymbol{w}} \boldsymbol{S}_\text{R}(\boldsymbol{w}_c, \mathbf{Q}) \right\}. \quad (6)$$

Eq.(6) outlines the complete construction scheme. We begin by selecting the top-i high-ranking frames from $\boldsymbol{S}_\text{R}$ and the top-j Rep. frames from $\boldsymbol{S}_\text{C}$. For each Rep. frame, we then include the contextual frame with the highest causal score within its sub-scene window. Afterward, we remove duplicates to avoid selecting the same frame more than once. The final sampling outcome is governed by the mixing ratio of i and j: by varying these two values, we can flexibly control the sampling process and create a video pool that reflects a wide range of keyframe selection strategies.

Evaluating Performance Diversity and Upper Bound in the Video Sampling Pool. It is interesting to analyze how different strategies in the pool affect the model response. Figure 3 visualizes the response behavior of seven different sampling strategies in the video pool across LVU benchmarks. The horizontal axis indicates the proportion of sampling strategies that correctly answer each sample, while the grid entries denote the number of samples correctly answered by a specific strategy. As we move rightward along the axis, query difficulty increases, and the intersection of correct responses correspondingly shrinks. For instance, Column $7/7$ corresponds to samples that all seven sampling strategies lead to the correct answer. In contrast, Column $1/7$ contains samples that only a single particular sampling strategy yields the correct answer. By comparing the numbers in column $1/7$, we can conclude that the existence of those extreme sampling strategies in pool, such as pool-1 and pool-7 hold the value of increasing the diversity of responses and enhancing the potential of the pool to handle difficult queries. In the ideal condition, if one could perfectly match each sample to the strategy that yields correct answer, the resulting performance would represent the upper-bound achievable by combining all strategies in the pool, which has denoted as Upper Acc in Fig.3.

The gap between the theoretical upper-bound and the actual best performance for any single sampling strategy demonstrates that even with strategic frame selection method, the long video sampling bias issue persists: MLLMs only process a sparse slice of frames eventually, each different sampling highlights a different "story", causing the model to generate distinct answers to the same query. The

existence of such sampling bias reveals the shortcoming of the conventional open-loop frame selection pipeline for long video understanding: no room for the adjustment of the selected frames. In light of this, we further propose a closed-loop solution by integrating MLLM's feedback to the inference pipeline.

3.4 Adaptive Sampling Strategy based on Model Feedback Loop

We introduce two types of feedback. The first allows the model to analyze the user query and report the desired sampling before receiving video frames, while the second allows the model to say "I don't know" after looking at the bias-sampled video frames and requires more information. In other words, MLLMs guide the frame selection process in our closed-loop pipeline as an agent. We explain the details in the following.

Closed-loop pipeline 1: User Query Analysis. This pipeline considers the inherent heterogeneity of user queries in the domain of LVU, ask the MLLM to propose the desired video pool prior to the sampling process, and provides the corresponding frames adaptively. The diversity of our video pool allows the query-based routing to end up with frames suitable to answer the given query. Before providing the MLLM with sampled video frames, we let the model first parse and classify the user queries. The model is allowed to determine if looking at only a single scene is sufficient to answer the query, or multiple scenes are necessary to understand the context. Different parsing results of the user query lead to different candidates in our video pool.

Closed-loop pipeline 2: Option of Reporting Missing Information. This pipeline allows the MLLM to judge whether it received sufficient information from the sampled video frames to answer a particular query. The model is provided with an additional option to refuse answering the query whenever it finds the sampled video frames are not enough for determine the answer. Using multiple-choice questions as an example, we expand the option list to include the choice of "Insufficient visual information". Whenever the model chooses this option as an answer, we will increase the capacity of the video pool and repeat the question. This pipeline guarantees that the model's final answer to the question is a fully-informed response.

4 Experiments

To validate the effectiveness of our framework, we choose LLaVA-Video (Zhang et al., 2024b) and Qwen2.5-VL (Bai et al., 2025) as our baseline models, then evaluate on three widely adopted benchmarks in the domain of LVU: LVB (Wu et al., 2024), VideoMME (Fu et al., 2025) and MLVU (Zhou et al., 2024).

LongVideoBench (LVB) contains videos ranging from 8 seconds to 60 minutes across diverse topics, accompanied by highly detailed user queries (average length: 43.53 words). We choose its publicly available validation set, comprising 1,337 QA pairs grounded in 752 videos.

VideoMME includes 2,700 QA pairs with an average video duration of 1,017.9 seconds. Its queries are notably concise (average length: 35.7 tokens), posing a significant LVU challenge without leveraging external subtitles.

MLVU features videos up to 2 hours in length and covers multiple complex task types, including Holistic, Single-Detail and Multi-Detail LVU. We select its multiple-choice subset, consisting of 2,174 QA pairs.

Following prior works, we employ LMMs-Eval (Zhang et al., 2024a) as the evaluation toolkit. For LVB and VideoMME, we report overall accuracy; for MLVU, we report average accuracy across its multi-task setup. Unless otherwise specified, our framework extracts a fixed total of 64 keyframes per video. As defined in Eq.(6), we generate video pools with varying blending ratios by setting $i \in \{8n | n \in [0, 8] \wedge n \in \mathbb{N}\}$. Specifically, for each sample, we first select the top-i frames from the perspective of relevance. Then we sequentially augment the selection with frames chosen according to summarization and causality-oriented criteria, following the designated algorithm mentioned above. If the total frame count remains below 64 after this process, which indicates limited sub-scene diversity in the video, we supplement the selection by greedily choosing additional frames from the unselected frame set, ranked by descending relevance score, until the 64-frame capacity

Table 1: Performance comparison among foundational (proprietary & open-source) MLLMs, training-based & free methods and our framework on three challenging LVU benchmarks. Here the numbers in parentheses indicate the performance improvement compared to the corresponding baseline model after introducing our framework, FPS denotes frames per second.

Method	Type	Model	Frames	LVB	VideoMME	MLVU
GPT-4o mini	Foundational	-	-	56.5	64.8	-
GPT-4V	Foundational	-	384	60.7	59.9	49.2
GPT-4o (Hurst et al., 2024)	Foundational	-	384	66.7	71.9	64.6
LLaVA-OV (Li et al., 2024)	Foundational	LLaVA-OV-7B	32	56.5	58.2	64.7
NVILA (Liu et al., 2025b)	Foundational	NVILA-8B	256	57.7	64.2	70.1
LLaVA-Video (Zhang et al., 2024b)	Foundational	LLaVA-Video-7B	64	58.2	63.3	70.8
Qwen2.5-VL (Bai et al., 2025)	Foundational	Qwen2.5-VL-7B†	64	60.0	63.5	63.2
LongVILA (Chen et al., 2024a)	Training-based	LongVILA-7B	256	57.1	60.1	-
LongVU (Shen et al., 2024)	Training-based	LongVU-7B	1FPS	-	60.6	65.4
Apollo (Zohar et al., 2025)	Training-based	Apollo-7B	2FPS	58.5	61.3	70.9
BIMBA (Islam et al., 2025)	Training-based	BIMBA-7B	128	59.5	64.7	71.4
TPO (Li et al., 2025)	Training-based	LLaVA-Video-7B	64	60.1	65.6	71.1
BOLT (Liu et al., 2025a)	Training-free	LLaVA-OV-7B	32	59.6	59.9	66.8
CoS (Hu et al., 2025a)	Training-free	LLaVA-Video-7B	64	58.9	64.4	71.4
AKS (Tang et al., 2025b)	Training-free	LLaVA-Video-7B	64	62.7	65.3	-
		Qwen2.5-VL-7B†		61.3	64.8	-
T²HTR	Training-free	LLaVA-Video-7B	64	**64.5** (+6.3)	**66.2** (+2.9)	**73.8** (+3.0)
		Qwen2.5-VL-7B		**66.9** (+6.9)	**67.0** (+3.5)	**76.3** (+13.1)

† denotes our reproduced experimental results.

is reached. Specific experimental details and algorithm procedure of **T²HTR** are provided in the Appendix.

4.1 Comparison with other LVU Works

We systematically categorize existing LVU works into three groups: (i) Foundational models, further divided into proprietary and open-source variants; (ii) Training-based models; (iii) Training-free methods. For MLLMs lacking keyframe selection component, we apply uniform sampling for the given video by default. Table 1 presents a comprehensive performance comparison among these categories and our framework.

First, compared to the respective baseline models, our method achieves consistent and significant gains: $+2.9\%$ to $+6.3\%$ on LLaVA-Video-7B and $+3.5\%$ to $+13.1\%$ on Qwen-2.5-VL-7B across all benchmarks. Second, as a plug-and-play scheme, **T²HTR** remains competitive even against training-based models. For instance, under the condition of utilizing LLaVA-Video-7B as the baseline model, we outperform TPO (Li et al., 2025) by 4.4% on LVB and 2.7% on MLVU. Third, when compared with similar training-free methods, **T²HTR** also demonstrates clear superiority: on the competitive Qwen2.5-VL-7B model, it surpasses AKS (Tang et al., 2025b) by 5.6%, 2.2% on LVB and VideoMME, respectively. In addition, it is worth noting that our approach even exceeds the performance of all proprietary models on LVB and MLVU, including GPT-4o (Hurst et al., 2024), which highlights the potential of the keyframe retrieval paradigm proposed in this work.

4.2 Comparison of Open-loop and Closed-loop Pipelines

As shown in Tab.2, we compare the performance of different sampling strategies. Here, fixed sampling denotes using an optimal and fixed blending ratio when selecting keyframes for all user queries, which is a conventional open-loop pipeline for inference. In comparison, User Query Analysis and Missing Information refer to the two closed-loop pipelines that leverage feedback from the model to adjust the sampling strategy (§3.3). In Query Analysis, the sampling strategy alternates between the two best candidates in the video pool depending on whether the model determines to focus more on the highest-relevant scenes or the causal content of related events. In Missing Information, models are first given 64 sampled video frames using the same blending ratio as Fixed Sampling. If the model chooses the additional option of insufficient information, we increase the frame number to

Table 2: Comparison of open-loop and closed-loop pipelines on three benchmarks (64 frames).

Sampling Strategy	Qwen2.5-VL-7B			LLaVA-Video-7B		
	LVB	VideoMME	MLVU	LVB	VideoMME	MLVU
Uniform Sampling	60.0	63.5	63.2	58.2	63.3	70.8
Fixed Sampling	66.2	66.7	75.5	63.9	65.9	73.5
User Query Analysis	**66.9**	66.9	75.9	63.9	**66.2**	**73.8**
User Query Analysis†	66.7	**67.0**	**76.3**	**64.5**	66.0	**73.8**
Missing Information	66.3	66.8	75.4	63.7	65.4	73.7

† denotes utilizing a LLM to assist in analyzing the given query.

Table 3: Ablation studies for mixing keyframes from multiple perspectives. Here *Rel-i, Sum-j, Causal-k* denotes a specific video pool composed of i, j, k frames extracted from the perspectives of relevance, summary and causality, respectively.

Video Pool Configuration	Qwen2.5-VL-7B			LLaVA-Video-7B		
	LVB	VideoMME	MLVU	LVB	VideoMME	MLVU
Optimal Video Sampling Pool	66.2	66.7	75.5	63.9	65.9	73.5
Rel-64, Sum-0, Causal-0	65.3	64.8	74.4	**63.9**	64.4	73.2
Rel-40, Sum-24, Causal-0	**65.7**	65.4	74.4	62.8	**65.5**	**73.6**
Rel-40, Sum-0, Causal-24	65.4	**66.3**	**74.9**	62.7	64.9	72.9

128 on those test samples (around 15%) for more visual cues, and we run the inference on them for the second time. No additional option will be given in the second round.

Experimental results suggest that Query Analysis strategies can further improve upon this baseline by up to 0.7%-0.8%, demonstrating the superiority of introducing a dynamic adjustable feedback loop rather than relying on a fixed, one-size-fits-all pool. However, no consistent improvement is observed using the Missing Information strategy. Through an in-depth analysis, we discover two reasons. First, this feedback is unreliable, as current MLLMs may not be able to effectively determine whether the visual information is sufficient. Second, adding an alterative option decreases the performance in general. Nevertheless, the performance improvement in partial cases indicates that powerful MLLMs have the potential of understanding long videos as an agent.

4.3 Ablation Studies for of Different Types of Keyframes

In Tab.3, we investigate how keyframe composition, which is derived from different retrieval criteria, affects overall LVU performance within a single video pool. The configuration *Rel-64, Sum-0, Causal-0* denotes a greedy selection based solely on relevance scores, without sub-scene window modeling or causality-aware evaluation. In contrast, *Rel-40, Sum-24, Causal-0* and *Rel-40, Sum-0, Causal-24* introduce a moderate number of frames selected according to summarization or causality criteria, respectively. Experimental results demonstrate that selecting keyframes based on relevance alone does not yield optimal performance in most cases. Instead, the effective combination of all three keyframe types is essential to fully unleash the inherent reasoning capability of MLLMs.

5 Conclusions

In this work, we present a test-time temporal understanding framework that jointly extracts keyframes from three complementary perspectives: relevance, summarization, and causality. We demonstrate that blending these multi-level cues at varying ratios allows the model to better adapt to user queries with diverse intents. To this end, we propose multiple closed-loop strategies that dynamically assign an optimal blending ratio or keyframe capacity to each sample, aiming to harness the full reasoning potential of MLLMs. We believe that future research in LVU should focus on developing more precise control mechanisms for modulating multi-level information and enabling sample-specific allocation.

REFERENCES

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

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

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

Chuanqi Cheng, Jian Guan, Wei Wu, and Rui Yan. Scaling video-language models to 10k frames via hierarchical differential distillation. In *International Conference on Machine Learning*, 2025.

Bo Fang, Wenhao Wu, Qiangqiang Wu, Yuxin Song, and Antoni B Chan. Threading keyframe with narratives: Mllms as strong long video comprehenders. *arXiv preprint arXiv:2505.24158*, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Hong Gao, Yiming Bao, Xuezhen Tu, Bin Zhong, and Minling Zhang. Apvr: Hour-level long video understanding with adaptive pivot visual information retrieval. *arXiv preprint arXiv:2506.04953*, 2025.

Jian Hu, Zixu Cheng, Chenyang Si, Wei Li, and Shaogang Gong. Cos: Chain-of-shot prompting for long video understanding. *arXiv preprint arXiv:2502.06428*, 2025a.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, et al. M-llm based video frame selection for efficient video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025b.

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

Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Gedas Bertasius, and Lorenzo Torresani. Bimba: Selective-scan compression for long-range video question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

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

Rui Li, Xiaohan Wang, Yuhui Zhang, Zeyu Wang, and Serena Yeung-Levy. Temporal preference optimization for long-form video understanding. *arXiv preprint arXiv:2501.13919*, 2025.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025a.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025b.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Canhui Tang, Zifan Han, Hongbo Sun, Sanping Zhou, Xuchong Zhang, Xin Wei, Ye Yuan, Jinglin Xu, and Hao Sun. Tspo: Temporal sampling policy optimization for long-form video language understanding. *arXiv preprint arXiv:2508.04369*, 2025a.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025b.

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. *arXiv preprint arXiv:2503.12559*, 2025a.

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

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. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025b.

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

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Gengshen Zhang, Dawei Leng, and Yuhui Yin. Fg-clip: Fine-grained visual and textual alignment. In *International Conference on Machine Learning*, 2025.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024.

Zeyu Xu, Junkang Zhang, Qiang Wang, and Yi Liu. E-vrag: Enhancing long video understanding with resource-efficient retrieval augmented generation. *arXiv preprint arXiv:2508.01546*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei 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.

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

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

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.

Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

A Experimental Configuration

Hyper-parameter Settings. Prior to hierarchical scene construction, we perform uniform frame sampling for the given video, as illustrated in Fig.2(I). For all experiments reported in this work, we adopt a default sampling rate of 1 FPS. For videos shorter than 128 seconds, we dynamically adjust the sampling interval to ensure that at least 128 frames are extracted, preserving sufficient temporal coverage for downstream processing.

In constructing first-level scenes and second-level sub-scenes, we set $t^* = \min(t+2, T), \Theta_{\mathrm{I}} = \cos(\pi/4), \Theta_{\mathrm{II}} = \cos(\pi/12)$ and segment sub-scenes with an average length of 4 frames. Specifically, for video scene $\mathbf{S}_{\mathrm{II}}^i$ containing n frames, we apply the segmentation criterion defined in Eq.(2), where the partitioning parameter $k = \lfloor n/4 \rfloor$.

Sub-scene windows are initialized with a default length of 5 frames, truncated to a minimum of 4 frames when positioned at temporal boundaries. We retain up to 64 sub-scene windows as Rep. frames for summarization- and causality-based keyframe selection. Here we allow Rep. frames to overlap across windows. To avoid redundancy, we first deduplicate the aggregated Rep. frame set before feeding it into Captioner to generate descriptive captions for each sub-scene. We then apply random masking over neighboring sub-scenes and pass the masked context windows to LLM for causal reasoning.

Employed Models. We utlize FG-CLIP (Xie et al., 2025) for relevance calculation. Specifically, input frames are resized to 224×224 before being fed into the vision tower, while questions and corresponding choices are tokenized and truncated to a maximum of 248 tokens for the text tower. In Eq.(3), we set $\alpha = 0.5, D = 768, p = 24$.

For caption generation and causal inference, we utilize the lightweight Qwen2.5-VL and Qwen3 (Yang et al., 2025) models, respectively. For the Captioner, we set max-pixels $= 128 \times 28 \times 28$, min-pixels $= 28 \times 28$ and constrain caption length to no more than 25 words generally. Both models are configured with max-model-len $= 8192$ and sampling parameters are performed with temperature $= 0$, top-$p = 0.95$, top-$k = 20$. All experiments are conducted on NVIDIA A100-SXM4-40GB and LC NVIDIA A800-SXM4-80GB GPUs.

Table S1: Detailed performance summary of video pools based on different blending ratios. Here *Rel-i, Sum-Causal-j* denotes a specific pool composed of i and j frames, which are respectively extracted from the frame and sub-scene levels, according to the video pool synthesis process mentioned above. Here **Bold** and Underline denote the optimal and suboptimal results.

Video Pool Configuration	Qwen2.5-VL-7B			LLaVA-Video-7B		
	LVB	VideoMME	MLVU	LVB	VideoMME	MLVU
Uniform Sampling	60.0	63.5	63.2	58.2	63.3	70.8
Rel-64, Sum-Causal-0	65.3	64.8	74.4	**63.9**	64.4	73.2
Rel-56, Sum-Causal-8	65.5	65.9	<u>75.1</u>	62.4	64.4	<u>73.4</u>
Rel-48, Sum-Causal-16	<u>66.1</u>	65.7	74.9	63.2	<u>65.7</u>	**73.5**
Rel-40, Sum-Causal-24	**66.2**	<u>66.5</u>	**75.5**	62.2	65.5	73.0
Rel-32, Sum-Causal-32	65.5	66.2	75.0	63.3	<u>65.7</u>	72.9
Rel-24, Sum-Causal-40	64.6	66.4	72.9	63.2	**65.9**	72.9
Rel-16, Sum-Causal-48	64.8	**66.7**	72.6	<u>63.4</u>	<u>65.7</u>	72.2
Rel-8, Sum-Causal-56	64.5	65.9	71.0	62.5	65.6	72.0
Rel-0, Sum-Causal-64	61.8	65.6	68.6	61.8	65.2	71.2

Table S2: Category-wise performance of video pools based on different blending ratios on LongVideoBench. The performance are computed based on four duration group, 15s, 60s, 600s and 3600s.

Video Pool Configuration	Qwen2.5-VL-7B				LLaVA-Video-7B			
	15s	60s	600s	3600s	15s	60s	600s	3600s
Rel-64, Sum-Causal-0	71.4	75.6	65.0	60.3	68.8	73.3	63.1	59.9
Rel-56, Sum-Causal-8	74.6	76.7	64.6	59.8	67.2	72.1	61.7	58.3
Rel-48, Sum-Causal-16	74.6	76.7	65.8	60.3	68.8	72.1	62.4	59.2
Rel-40, Sum-Causal-24	74.6	76.2	66.0	60.5	68.8	72.1	62.1	57.1
Rel-32, Sum-Causal-32	74.1	76.7	64.6	59.9	68.8	72.1	62.6	59.2
Rel-24, Sum-Causal-40	74.1	76.7	63.6	58.5	68.8	72.1	62.1	59.4
Rel-16, Sum-Causal-48	74.1	76.7	63.1	59.4	68.8	72.1	61.9	60.1
Rel-8, Sum-Causal-56	74.1	76.7	62.4	59.0	68.8	72.1	62.1	57.8
Rel-0, Sum-Causal-64	74.1	76.7	60.9	53.7	68.8	72.1	61.4	56.6

Table S3: Category-wise performance of video pools based on different blending ratios on VideoMME. The performance are computed based on three duration group, short, medium and long.

Video Pool Configuration	Qwen2.5-VL-7B			LLaVA-Video-7B		
	Short	Medium	Long	Short	Medium	Long
Rel-64, Sum-Causal-0	75.9	65.9	52.7	76.3	64.4	52.3
Rel-56, Sum-Causal-8	77.3	65.1	55.3	77.6	63.3	52.2
Rel-48, Sum-Causal-16	77.6	65.7	53.8	78.9	64.6	53.8
Rel-40, Sum-Causal-24	77.6	66.9	55.0	78.1	65.1	53.2
Rel-32, Sum-Causal-32	76.4	66.7	55.6	77.6	66.4	53.2
Rel-24, Sum-Causal-40	76.7	67.0	55.4	77.7	66.2	53.9
Rel-16, Sum-Causal-48	76.6	66.8	56.7	77.7	65.6	54.0
Rel-8, Sum-Causal-56	76.6	66.3	54.8	77.6	65.0	54.2
Rel-0, Sum-Causal-64	76.6	65.6	54.6	77.6	63.4	54.6

B Detailed Results on Three Benchmarks

This section expands upon the summary tables presented in the main paper by reporting full benchmark results across all three datasets. For each dataset, we provide comparisons with prior baselines as well as a breakdown of the performance variations across different video pools, as shown in Tab.S1-S3.

C Visualization of Different Video Sampling Pools

Figure S1 illustrates the performance of video sampling pools with different mixture ratios in responding to user queries of varying intents. Pool *Rel-64, Sum-Causal-0*, which consists entirely of keyframes selected from a relevance-based perspective, is better suited for addressing local, detail-oriented questions, as exemplified by the case in the first row. In contrast, introducing a certain proportion of keyframes derived from the perspective of summarization and causal inference enables the MLLM to more effectively answer queries that require understanding of event context, such as the case shown in the second row.

D Algorithm

We provide the detailed algorithm procedure of $\mathbf{T^2HTR}$ in Alg.1. To summarize, the input video is first segmented into scenes and sub-scenes to establish a structured temporal hierarchy, enabling efficient management of long sequences. Then the frame-level and sub-scene–level relevance scores are computed with respect to the user query, providing an initial ranking of potentially informative

Figure S1: The response status of different video pools on Qwen2.5-VL-7B, VideoMME. Here *Rel-i, Sum-Causal-j* denotes a specific pool composed of i and j frames, which are respectively extracted from the frame and sub-scene levels, according to the video pool synthesis process mentioned above.

segments. Highly relevant representative frames are extracted and used as anchors to form localized temporal windows, capturing context around critical events without processing the full sequence. Next, captions are generated for representative frames, while neighboring segments are masked to assess predictive consistency. This causality check selects the context frames that contain crucial information to understand the event in the video. Finally, multiple video pools are generated, each blending segments at varying ratios. A closed-loop routing strategy iteratively selects the most coherent path, yielding the final MLLM-generated responses aligned with query objectives.

Algorithm 1 **T**est-**t**ime **H**ierarchical **T**emporal **R**etrieval (**T^{2}HTR**)

Input: the given videos $\mathbf{V}_{(1:N),1:T}$ and user queries $\mathbf{Q}_{(1:N)}$

Output: the MLLM response set $\tilde{\mathbf{A}}_{(1:N)}$

 # Hierarchical Video Scene Construction

From Eqs.(1-2), apply $\boldsymbol{J}_{\mathrm{I}}(\cdot), \boldsymbol{J}_{\mathrm{II}}(\cdot)$ for $\mathbf{V}_{(1:N),1:T}$ to sequentially establish video scenes $\bigcup_{i=1}^{N} \mathbf{S}_{(i),\mathrm{I}}$ and sub-scenes $\bigcup_{i=1}^{N} \mathbf{S}_{(i),\mathrm{II}}$

 # Relevance Calculation

Calculate the joint relevance scores $\bigcup_{i=1}^{N} \boldsymbol{S}_{\mathrm{R}}(\mathbf{V}_{(i),1:T}, \mathbf{Q}_{(i)})$ frame by frame according to Eq.(3)

 # Sub-scene Window Modeling

Calculate the relevance scores of video sub-scenes $\bigcup_{i=1}^{N} \boldsymbol{S}_{\mathrm{R}}(\mathbf{S}_{(i),\mathrm{II}}, \mathbf{Q}_{(i)})$

From Eq.(4), extract Rep. frames from $\bigcup_{i=1}^{N} \mathbf{S}_{(i),\mathrm{II}}$ by implementing $\boldsymbol{J}_{\mathrm{III}}(\cdot)$

Construct corresponding sub-scene windows $\bigcup_{i=1}^{N} \{\boldsymbol{w}_c, \boldsymbol{w}_n\}_{(i)}$ by utilizing top-k relevant Rep. frames as central anchor points

 # Causality Evaluation

Generate captions for Rep. frames and randomly mask neighboring sub-scenes $\boldsymbol{w}_n$

Employ LLM to predict the visual description at the center anchor point corresponding to each set of mask data

Calculate the causal scores of relevant Rep. frames $\bigcup_{i=1}^{N} \boldsymbol{S}_{\mathrm{C}}(\mathbf{V}_{(i),1:T} | \mathbf{V}_{(i),1:T} \in \{\boldsymbol{w}_n\}_{(i)}, \mathbf{Q}_{(i)})$ according to Eq.(5)

 # Closed-loop Routing Strategies

Generate M video pools $\mathbf{P}_{(1:N),(1:M)}$ with different blending ratios based on Eq.(6)

Based on the existing video pools, choose a specific closed-loop pipeline to obtain the final model response set $\tilde{\mathbf{A}}_{(1:N)}$ of sample granularity

E Implementation Detail and Analysis of Closed-loop pipelines

In this final section we elaborate more on the implementation details and the analysis of the proposed closed-loop pipelines: User Query Analysis and Missing Information.

User Question Analysis. We adopt the following prompt to ask the MLLM classify the user query into "single" or "multiple" category: *Determine if the following question can be answered by viewing a single scene from the video, or if it requires understanding events and relationships across multiple scenes. Output *exactly* one lowercase word: "single" or "multiple". Do not include any other text, punctuation, or explanation. For example: Question: What color is the car?* $\rightarrow$ *single Question: What is the woman with the pink hat wearing* $\rightarrow$ *single Question: Why did the person run away?* $\rightarrow$ *multiple Question: What is the order of the following event?* $\rightarrow$ *multiple Question:* $< |placeholder| >$. The "$< |placeholder| >$" string will be replaced by the actual user query. No video frames are input into the model at this stage. After the classification, we will determine the blending ratio of i and j for high-ranking frames and representative frames. For user queries that are identified as "single", we select more high-ranking frames, whereas the number of representative frames increase when the question is classified as "multiple".

Missing Information. We add an additional option in the candidate list of the user query. For instance, the original question and candidates are: *What is the color of the car? A. Red B. Gray C. White D. Black + (64 sampled video frames)* and the modified input becomes: *What is the color of the car? A. Red B. Gray C. White D. Black E. Insufficient visual information + (64 sampled video frames)*. If the model still responds with one of the original four options, we record its answer, and the inference ends. We call the union of these test samples as the sufficient set. If the answer is 'E', we sample more frames from the video and modify the input at a second time: *What is the color of the car? A. Red B. Gray C. White D. Black + (128 sampled video frames)*. These samples are included in the insufficient set. We decide to end at the second round for the simplicity of the pipeline. We have discovered interesting phenomenon when we analyzed the imperfect results of this pipeline, as shown in Tab.2. First of all, we observed that the new option influences the MLLM's decision even when the model thinks that the visual information provided is enough. As a result, the model performs worse on the sufficient set when given the first modified input. Therefore, even though increasing the sampling frames on the insufficient set improves the performance, the overall improvement is less significant. Second, by visualizing the samples identified as insufficient, we realized that the model cannot judge whether it receives enough visual clues from the video to answer the question. We hope that reinforcement learning on the model could help improve its capability of determining whether it is given a proper question. We will leave this as the future work.