VideoNav: Cost-Aware Index Navigation for Long Video Understanding

Anonymous ACL submission

Abstract

The rising demand for long-video understanding presents significant challenges due to massive information and complex temporal dependencies. Direct processing with Large Visual Language Models (VLMs) incurs high computational costs and risks information loss due to frame sampling. Meanwhile, existing agent-based methods often struggle with fragmented contexts and the high overhead of iterative processing. Therefore, we propose VideoNav, which introduces a new paradigm of index-centric navigation, reformulating video understanding as cost-aware structured query processing. VideoNav decouples heavy perception from online reasoning, including offline index construction and online adaptive navigation. In the offline phase, we transform unstructured video data into a structured, multi-granular index tree. During the online phase, the framework executes a top-down, coarse-to-fine navigating strategy. Leveraging the LLM agents, it efficiently locates relevant information and answers the query with navigating guidance, multi-level navigation, dual-path routing and visual inspection. Crucially, this adaptive mechanism enables early termination when high-level information suffices, minimizing redundant computation. Extensive experiments across different benchmarks demonstrate the effectiveness of our framework. VideoNav achieves decent accuracy while delivering 2.5×-5.7× cost savings and a 2.9×-5.8× speedup. [1]

1 Introduction

Video content has become a prevalent information medium (Tang et al., 2025), driving demand for automated video understanding in surveillance, content management, and interactive platforms. Long videos, with their complex temporal dependencies and massive information volume, pose significant challenges (Wang et al., 2024a; Zhou et al., 2025; Wang et al., 2025a).

Despite the impressive capabilities of recent Large Visual Language Models (LVLMs or VLMs)(Comanici et al., 2025; Bai et al., 2025; Guo et al., 2025), deploying them directly for long video processing poses several limitations (Figure 1). Directly inputting plenty of video frames into a VLM incurs significant computational costs. Furthermore, the context constraints of VLMs often force frame down-sampling, risking the loss of critical visual details and decreased accuracy (Yang et al., 2025b; Wang et al., 2025b).

To address these, agent-based frameworks have emerged as a promising approach (Wang et al., 2024a; Zuo et al., 2025; Zhang et al., 2025b). They leverage Large Language Models (LLMs) and VLMs to inspect frames, analyze information, reflect processing and select next frame sequence (Figure 1). However, existing agent-based approaches face two critical hurdles. First, many frameworks execute analysis based on **fragmented frames**, failing to capture the temporal relationships, which are essential to understand causality and global progression (Ren et al., 2025; Luo et al., 2024). Second, these frameworks typically rely on heavy online processing (e.g., iterative reflection or complex tool-use loops) for each incoming query, and the complexity of these operations often necessitates the use of expensive leading LLMs (Zhang et al., 2025b; Pang and Wang, 2025; Wang et al., 2025c). This combination leads to **excessive latency and token costs**, ill-suited for frequent online user queries. This motivates an efficient framework, that can comprehensively capture video content and maintain an acceptable computational cost for online deployment.

To this end, we propose a new paradigm that reformulates video understanding as **index-centric query processing**. The key idea is to **decouple heavy perception from adaptive reasoning**

[1] We provide the anonymous code as supplementary materials via https://anonymous.4open.science/r/VideoNav-51C1.

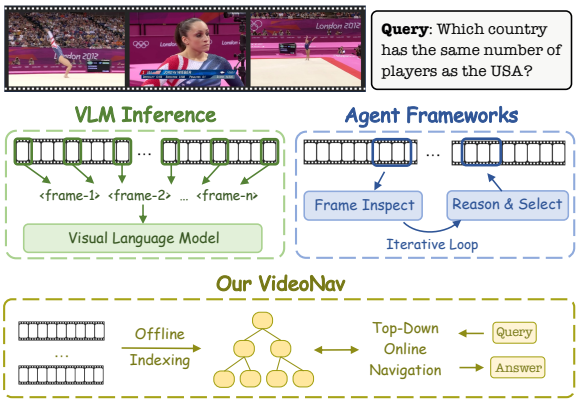

Figure 1: **Comparison of video understanding paradigms.** Standard VLM inference processes a down-sampled massive frame sequence. Agent frameworks involve fragmented, costly loops. VideoNav introduces index-centric navigation with multi-granular indexing and adaptive navigation.

and **structure** the voluminous information. We structure the video into a comprehensive index offline, via a one-time multi-modal perception pass, encapsulating both global progression and specific details. Then the online queries only require lightweight language-based reasoning and navigation. This fundamentally differs from prior agent methods that repeatedly invoke expensive VLM perception online, where costs surge with a large number of incoming queries. Guided by this paradigm, we introduce VideoNav, a cost-aware framework with two stages: offline *index construction* and online *adaptive navigation*.

In the offline stage, we construct a multi-granular hierarchical index, capturing information at different granularity (Figure 2). This four-level structure retains raw frames at the base (L4), generates clip-level textual descriptions (L3), aggregates clips into semantic blocks (L2), and produces a global summary (L1). In the online stage, the framework performs coarse-to-fine adaptive navigation. Upon receiving a query, it first examines high-level information. If specific details are needed, it locates relevant clips via dual-path routing (time-range zooming and semantic matching), and invokes visual inspection only when necessary. Critically, this navigation process will adopt early termination, if the information at higher levels suffices to answer the user query.

Besides, to further enhance the reliability of the index and improve the efficacy of navigation, we introduce several **optimization components**, such as context-aware construction, adjacent clustering, and navigation guidance (more details in Section 3). Our experiments across multiple benchmarks demonstrate that VideoNav achieves great accuracy while delivering 2.5×-5.7× cost savings and a 2.9×-5.8× speedup.

In summary, we make the following contributions:

- We propose an index-centric paradigm that decouples offline multi-modal perception from online language reasoning.
- We design and implement the VideoNav framework, featuring hierarchical indexing and adaptive navigation, capable of handling diverse user queries efficiently.
- We provide comprehensive experiments, demonstrating that VideoNav advances the feasibility of long-video understanding in practical settings.

2 Preliminary

Given a video $\mathcal{V}$ and a user query Q, a video understanding framework should derive an accurate answer A, covering tasks such as temporal grounding, event understanding and summarization. A video is always represented as a frame sequence $\mathcal{V} = \{f_1, \ldots, f_T\}$. For long videos (30 minutes to several hours), the massive volume of visual data poses substantial challenges to VLMs.

Let K_v denote the visual tokens per frame. Taking Qwen3-VL (Bai et al., 2025) as an example: a 2-hour video at 2 FPS would yield T=14400 frames, each frame requiring K_v=380 tokens and totaling 5.4M tokens, which far exceeds most VLMs' context limits. Even aggressive down-sampling to 384 frames still costs ∼146k tokens per query, with inevitable loss of critical visual details. This motivates our index-centric approach that balances accuracy and cost.

3 Method

We present VideoNav, a flexible framework for efficient long-video understanding. We first provide an overview, then detail the offline index construction (§3.2) and online adaptive navigation (§3.3).

3.1 Overview

Figure 2 illustrates the processing pipeline of VideoNav. We reformulate the video understanding as an index-centric query processing problem, which consists of two decoupled phases: offline hierarchical indexing and online adaptive navigation.

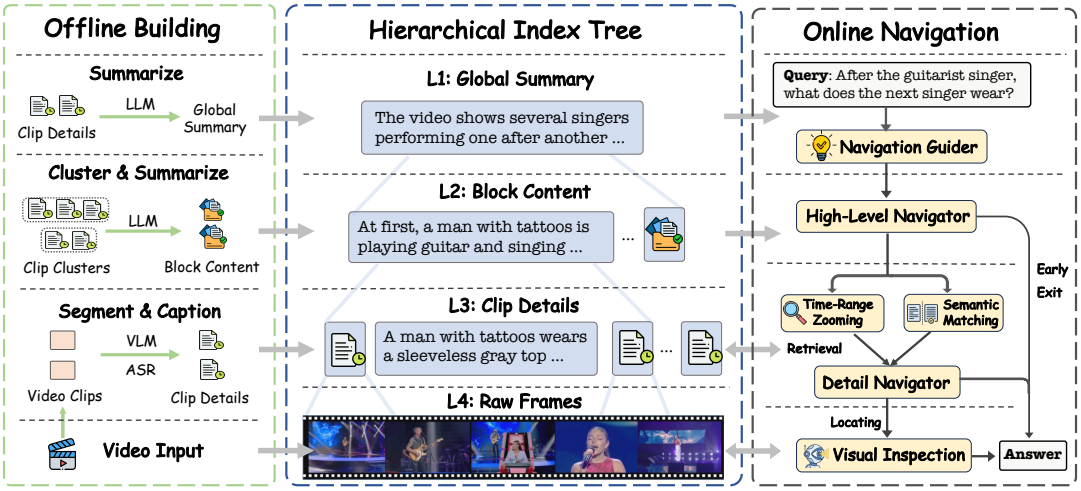

Figure 2: Overview of the VideoNav framework. Offline building forms a hierarchical index tree to represent video content. In the online phase, VideoNav executes adaptive index navigation to efficiently resolve queries. More detailed examples are shown in Fig 5 and Fig 6.

The hierarchical index tree serves as the bridge between these phases, allowing comprehensive perception to be pre-computed while keeping the online process lightweight and effective.

Offline Phase: Index Construction. Given a long video, VideoNav transforms the unstructured data into a structured, multi-granular index tree. Starting at the foundation, VideoNav retains raw video frames (L_4) and segments the video into clips. It then leverages the VLM and ASR (automatic speech recognition) model to processes visual and audio information, generating textual descriptions and incorporating timestamp metadata, forming the clip level details (L_3). To ensure holistic understanding, VideoNav first synthesizes a Global Summary (L_1) directly from these clip details. After that, clips are clustered and summarized into semantic blocks (L_2). Crucially, this block-level summarization is context-aware, utilizing the global summary as the context to ensure alignment with the overall narrative. Finally, this multi-granular structure, spanning from raw frames to the global summary, serves as a tree-based index that facilitates organized online information access.

Online Phase: Adaptive Navigation. The online phase mirrors a human-like cognitive process, adaptively executing a top-down, coarse-to-fine navigation strategy governed by LLM agents. Upon receiving a user query, a navigation guider first analyzes the query with global summary to formulate a high-level guidance. The high-level navigator then analyzes block-level contents to determine whether the current information is sufficient to answer the query; if so, it terminates early to save costs. If finer details are required, the LLM navigator would output a zoomed time-range, pinpointing the potential location of relevant information. VideoNav then invokes a dual-path routing mechanism, combining this target time-range with semantic matching, to retrieve specific clip details. After the detail navigator, if precise visual evidence is required, the framework will trigger Visual Inspection, invoking the VLM to examine raw frames. By dynamically proceeding the navigation, VideoNav ensures comprehensive understanding while reducing unnecessary computation.

3.2 Hierarchical Index Construction

VideoNav constructs a structured, four-layer hierarchical index offline, transforming unstructured data into semantically organized textual representations at varying granularities ($\{L_4, L_3, L_2, L_1\}$).

3.2.1 Clip-Level Multi-modal Description

At the foundational level, we segment the video $\mathcal{V}$ into a sequence of fixed-duration clips $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$. For each clip c_i, we perform multi-modal alignment to generate a comprehensive textual description. Specifically, we first employ an Automatic Speech Recognition (ASR) model to extract the audio transcript $\mathcal{T}_i^{aud}$. To enhance the semantic consistency, the visual description is generated by a VLM, conditioned on both the visual frames $f_{i_1}, f_{i_2}, \ldots, f_{i_t}$ and the corresponding audio context:

$$\mathcal{T}_i^{vis} = \text{VLM}(f_{i_1}, f_{i_2}, \ldots, f_{i_t} \mid \mathcal{T}_i^{aud})$$

This allows the VLM to ground visual entities with cross-modal audio context. The final clip representation D_i concatenates the audio transcript, visual caption, and timestamp metadata.

We store clip details in a Key-Value store indexed by timestamp ranges for temporal access, and encode visual descriptions into vectors $e_i = \text{Enc}(\mathcal{T}_i^{vis})$ in a vector database for semantic matching.

3.2.2 Global Summarization

Prior to constructing the intermediate block level, we first synthesize a global overview directly from the foundational clip-level details. This top-level context is essential for capturing the overall narrative and serves as contextual guidance for the subsequent block construction. Specifically, we aggregate the textual descriptions of all clips $D = \{D_1, D_2, \ldots, D_N\}$ and employ the LLM to generate the Global Summary S^{glb}:

$$S^{glb} = \text{LLM}(\text{Concat}(D_1, D_2, \ldots, D_N)) \quad (1)$$

This summary functions as the root node (L_1), encapsulating the video's core themes and main events, providing a global prior for both block construction and online navigation.

3.2.3 Coherent Block Construction

Clip-level details are too voluminous to feed into the LLM entirely, while purely retrieving top-k similar clips disrupts temporal continuity. We therefore construct a higher-level layer of Blocks by clustering semantically coherent and temporally adjacent clips.

The clustering must satisfy: (1) *Temporal Continuity*: clips remain adjacent to preserve narrative flow; (2) *Information Balance*: cluster count and sizes should be balanced. However, LLM-based clustering is expensive, and traditional methods (e.g., K-Means) ignore these constraints.

Therefore, we propose an *iterative adjacent merging* algorithm. We first determine the cluster granularity: given N clips, we define the maximum cluster capacity k and target cluster count M:

$$k = \lfloor \sqrt{N} \rfloor; \quad M = \left\lceil \frac{\beta N}{k} \right\rceil, \quad (\beta \geq 1) \quad (2)$$

where β is a relaxation factor (empirically set to 1.2). Starting with each clip as its own cluster, we iteratively merge adjacent pairs (B_j, B_{j+1}) with maximal cosine similarity, subject to capacity $\leq k$:

$$e_{new} = \frac{|B_j| \cdot e_{B_j} + |B_{j+1}| \cdot e_{B_{j+1}}}{|B_j| + |B_{j+1}|} \quad (3)$$

Merging stops when M clusters remain.

After that, we generate a summary for each cluster, forming the block content. Instead of independent processing, we leverage the pre-computed Global Summary S^{glb} as a context prior. This ensures the **Context-Aware Construction**, where the block summary S_j^{blk} could be aligned with the global narrative flow, reducing information bias:

$$S_j^{blk} = \text{LLM}\left(\{D_i\}_{i \in B_j} \mid S^{glb}\right) \quad (4)$$

This produces the mid-level context (L_2) that captures key entities and event progression while maintaining global consistency.

3.3 Adaptive Cognitive Navigation

During the online phase, VideoNav resolves queries through adaptive index navigation, employing an agent-based, coarse-to-fine strategy that traverses the hierarchical index from root to leaves.

3.3.1 Query Guidance and High-Level Navigation

Upon receiving a query Q, VideoNav first invokes a **Navigation Guider** that analyzes the query intent using the Global Summary S^{glb}: $H = \text{LLM}_{\text{guide}}(Q, S^{glb})$.

This hint serves as a directive for subsequent navigation agents. For example, given "where did they go after swimming?", the hint might be: "Locate the swimming scene and inspect subsequent actions." This guidance helps prioritize task-relevant features and reduces decision errors when navigating massive data.

The framework then proceeds to **High-Level Navigation**, where the LLM agent analyzes the block-level summaries $\mathcal{S}^{blk} = \{S_1^{blk}, \ldots, S_M^{blk}\}$ with query Q and hint H, deciding the action $a_{high} \in \{\textsc{Answer}, \textsc{Seek}\}$.

If sufficient evidence exists (e.g., for event recognition), the agent terminates early with answer A. Otherwise, it predicts a focused temporal range $\mathcal{R}_{time}$ (or $\emptyset$ if ambiguous):

$$\text{Output} = \begin{cases} \langle \textsc{Answer}, A \rangle, & \text{if info is sufficient} \\ \langle \textsc{Seek}, \mathcal{R}_{time} \rangle, & \text{if time inferred} \\ \langle \textsc{Seek}, \emptyset \rangle, & \text{otherwise} \end{cases} \quad (5)$$

3.3.2 Dual-Path Retrieval and Detail Navigation

For queries requiring fine-grained details, VideoNav retrieves clip-level information via a

Dual-Path Retrieval mechanism:

- **Path A: Time Range Zooming.** Using the predicted $\mathcal{R}_{time}$ from high-level navigation, this path selects corresponding clips $\mathcal{C}_{\text{time}}$, preserving temporal order and causality that pure semantic matching would lose.
- **Path B: Semantic Matching.** An LLM rewrites the query into a retrieval-optimized form q', then retrieves top-k clips from the vector database. This excels at identifying specific visual elements (e.g., "an old lady in a white shirt").
- **Candidate Merging.** We select top-$N/2$ clips from each path, de-duplicate, and sort by timestamp. If $\mathcal{R}_{\text{time}} = \emptyset$, the full budget goes to semantic matching.

With the retrieved clips, **Detail Navigation** analyzes clip descriptions D_i and determines to answer or seek. If textual evidence suffices, the answer is generated; otherwise, specific timestamps $(\tau_{start}, \tau_{end})$ are identified for visual inspection.

3.3.3 Frame-Level Inference

For identified timestamps, VideoNav retrieves raw frames $f_{\tau_{start}:\tau_{end}}$ and invokes the VLM for **Visual Inspection**:

$$A = \text{VLM}(Q, f_{\tau_{start}:\tau_{end}}, I) \tag{6}$$

where I provides contextual information (timestamp and global summary). Thanks to the preceding adaptive navigation, many queries are resolved before this stage, and the VLM only processes a small subset of task-relevant frames when needed.

As a **fallback**, if hierarchical navigation fails to produce a confident answer, the system reverts to standard VLM inference on a broader frame set, ensuring robust performance.

4 Evaluation

4.1 Experimental Setups

Datasets and Metrics. We conduct experiments on three video understanding benchmarks. **LV-Bench** (Wang et al., 2025a) targets hour-long videos with temporal grounding, key information retrieval, and summarization tasks. **Video-MME** (Fu et al., 2025) covers real-world scenarios with short, medium, and long videos (1 min to 1 hour). **MLVU** (Zhou et al., 2025) provides diverse tasks on 3-minute to 2-hour videos. For consistent evaluation, these benchmarks have formulated their understanding tasks as **multi-choice questions**. Following prior work (Luo et al., 2024; Wang et al., 2025c; Pang and Wang, 2025; Zuo et al., 2025; Wang et al., 2025b), we report *accuracy*.

Baselines. We compare our method against two categories: (1) **VLM inference** methods, including general-purpose models (GPT-4o (Hurst et al., 2024), Gemini-2.5 (Comanici et al., 2025), Seed-1.5VL (Guo et al., 2025), Qwen3-VL (Bai et al., 2025)) and specialized Video-VLMs (ViLAMP (Cheng et al.), Oryx (Liu et al., 2024)); (2) **Agent-based frameworks**, including RAG-style methods (Video-RAG (Luo et al., 2024), MemVid-RAG (Yuan et al., 2025)) and iterative multi-agent systems (Video-Agent (Wang et al., 2024a), VCA (Yang et al., 2025b)).

Implementation. VideoNav uses lightweight open-source models: `Qwen3-VL-8B` (Bai et al., 2025) as the VLM, `Qwen3-30B-A3B` (Yang et al., 2025a) as the LLM, `bge-large-en-v1.5` (Xiao et al., 2023) for embeddings, and `faster-whisper-large-v3` (Radford et al., 2023) for ASR. We use 1-minute clips at 1 FPS and a retrieval budget of 16 clips. For baselines, given the expensive cost associated with long-video evaluation, we follow common practice in plenty of prior works (Yuan et al., 2025; Chen et al., 2025; Wang et al., 2025c; Pang and Wang, 2025; Yang et al., 2025b), primarily citing the results from official leaderboards or corresponding papers. To establish a critical reference, we explicitly re-evaluate the performance of the standalone `Qwen3-VL-8B` (128k context, 768 frames). More details are shown in the Appendix B.1.

4.2 Accuracy Results

Tables 1 and 2 present accuracy on three benchmarks. VideoNav consistently outperforms standard inference with comparable or larger VLMs (e.g., GLM-4V-Plus, LLaVA-OneVis-72B). Despite using lightweight open-source models, our framework achieves competitive results with other expensive leading models (e.g., Gemini-2.5-Flash), demonstrating the effectiveness of coarse-to-fine index design.

Compared to agentic frameworks, VideoNav also shows great performance across different benchmarks. While some frameworks extensively invoke expensive LLMs (e.g., MR.Video with Gemini model) for higher accuracy (Wang et al., 2025c; Pang and Wang, 2025), they diverge from our cost-

Table 1: Accuracy performance on LV-Bench. Results are reported for overall performance and six tasks: Entity Recognition (ER), Event Understanding (EU), Key Information Retrieval (KIR), Temporal Grounding (TG), Reasoning, and Summarization.

	LV-Bench						
	ER	EU	KIR	TG	Rea	Sum	Overall
Processed by Leading VLMs							
GPT-4o (Hurst et al., 2024)	48.9	49.5	48.1	40.9	50.3	**50.0**	48.9
Gemini-2.5-Flash (Comanici et al., 2025)	-	-	-	-	-	-	62.2
Seed-1.5-VL-200B-A20 (Guo et al., 2025)	**65.4**	**63.4**	**68.0**	**53.6**	**63.7**	46.6	**64.6**
GLM-4V-Plus (GLM et al., 2024)	46.2	47.8	54.1	42.7	46.5	37.9	48.7
Qwen3-VL-8B (Bai et al., 2025)	57.6	54.4	65.9	50.9	51.7	37.9	57.0
InternVL2.5-78B (Chen et al., 2024)	43.8	42.0	42.1	36.8	51.0	37.9	43.6
VILAMP-7B (Cheng et al.)	-	-	-	-	-	-	46.3
VideoLLaMA3-7B (Zhang et al., 2023)	45.8	42.4	47.8	35.9	45.8	36.2	45.3
Agent-Based Video Framework							
Video-Agent (Wang et al., 2024a)	28.0	30.3	28.0	29.3	28.0	36.4	29.3
Video-Tree (Wang et al., 2025b)	30.3	25.1	26.5	27.7	31.9	25.5	28.8
VCA (Yang et al., 2025b)	43.7	40.7	37.8	38.0	46.2	27.3	41.3
MemVid-RAG (Yuan et al., 2025)	53.4	40.6	46.3	34.9	43.2	28.1	44.4
MR. Video (Gemini) (Pang and Wang, 2025)	59.8	57.4	**71.4**	**58.8**	**57.7**	**50.0**	**60.8**
Video-Nav (Ours)	**60.6**	**58.9**	66.7	53.6	55.7	43.1	60.5

Table 2: Accuracy performance on Video-MME and MLVU. Video-MME results are reported across different video durations. By default, Video-MME results incorporate subtitle assistance; approaches marked with asterisks * do not use subtitles.

	Video-MME				MLVU
	Short	Medium	Long	Overall	
Processed by Leading VLMs					
GPT-4o (Hurst et al., 2024)	**82.8**	**76.6**	**72.1**	**77.2**	64.6
Gemini-2.5-Flash (Comanici et al., 2025)	-	-	-	-	**77.8**
MiniCPM-o2.6-8B (Yao et al., 2024)	78.3	69.1	56.3	67.9	-
Qwen3-VL-8B (Bai et al., 2025)	79.4	72.1	62.7	71.4	73.7
LLaVA-OneVis-72B (Li et al., 2024a)	79.3	66.9	62.4	69.6	68.0
VideoLLaMA3-7B (Zhang et al., 2023)	80.2	69.6	61.0	70.3	73.0
Oryx-1.5-32B (Liu et al., 2024)	-	-	-	-	72.3
Long-VITA-14B (Shen et al., 2025)	77.0	73.0	66.0	72.0	-
Agent-Based Video Framework					
Video-Agent* (Wang et al., 2024a)	-	-	46.4	-	-
Video-Tree* (Wang et al., 2025b)	67.8	59.9	54.2	60.6	60.4
VCA* (Yang et al., 2025b)	-	-	56.3	-	-
Video-RAG (Luo et al., 2024)	67.1	60.4	60.1	62.6	72.4
MemVid-RAG (Yuan et al., 2025)	75.4	64.6	57.1	65.7	58.1
MR. Video*(Gemini) (Pang and Wang, 2025)	-	-	61.8	-	-
Video-Nav (Ours)	**78.9**	**73.9**	**67.8**	**73.5**	**73.9**

Table 3: Efficiency performance across various benchmarks. Input token counts are reported in thousands (k). Latency values represent the average per query. Costs are calculated using official model prices and include output tokens; costs are normalized per benchmark.

Method	Model Used	In Tok.	$ Cost	Latency	Acc
LV-BENCH					
MR. Video	Gem-2.0-Flash + GPT-4o	109.5k	120×	-	60.8
AVP	Gemini-2.5-Flash	∼130k	-	-	**63.8**
VLM Inference	Qwen3-8B-VL	116.8k	2.8×	37.7s	57.0
VideoNav	Qw3-8B-VL + 30B-A3B	**38.4k**	**1**×	**13.2s**	60.5
VIDEO-MME					
VLM Inference	Qwen3-8B-VL	98.7k	5.7×	30.4s	71.4
VideoNav	Qw3-8B-VL + 30B-A3B	**13.8k**	**1**×	**5.2s**	**73.5**
MLVU					
VLM Inference	Qwen3-8B-VL	100.1k	2.5×	29.8s	73.7
VideoNav	Qw3-8B-VL + 30B-A3B	**36.6k**	**1**×	**9.5s**	**73.9**

aware practical setting. We will further discuss them in Section 4.3 (Table 3).

4.3 Efficiency Performance

Token Efficiency. Table 3 presents the online cost results. We emphasize two separate comparisons. (1) Same-backbone comparison: against standard VLM inference with the same Qwen3-VL-8B backbone, VideoNav consumes only **13.9%-36.6%** of input tokens with **2.5×-5.7×** cost savings, while achieving superior accuracy. This directly demonstrates the efficiency of our navigation paradigm itself.[2] (2) Cross-model comparison: complex agent frameworks like Mr. Video (Pang and Wang, 2025) and AVP (Wang et al., 2025c) rely on expensive proprietary models (e.g., Gemini) for iterative reasoning. VideoNav achieves comparable performance using only small open-source models with **29.5%-34.9%** of their input tokens, demonstrating a practical deployment path. This efficiency stems from our adaptive navigation with early termination, reducing expensive processing. Meanwhile, the hierarchical index preserves both global context and fine-grained details, enabling accuracy.

Latency Performance. For a fair comparison, we adopt a standardized model API deployment. We deploy Qwen3-8B-VL and Qwen3-30B-A3B models respectively, both with the vLLM engine on an NVIDIA H200 GPU (141GB memory). As shown in Table 3, VideoNav achieves a **2.9×-5.8× speedup** compared to standard VLM inference, with better accuracy.

Offline Overhead. Index construction inevitably introduces a one-time offline cost. For a 1-hour video, this consumes ∼500k input tokens. However, standard VLM inference consumes ∼120k tokens for *each* online query. So the index construction cost is equivalently amortized after just ∼4 queries, with greater accuracy and latency performance. Besides, once built, the index serves all subsequent queries at low marginal cost and can also be reused for other downstream tasks (e.g., video search).

4.4 Ablation Study

We conduct ablation studies on a randomly sampled subset of LV-Bench, given its high complexity and comprehensive coverage (comprising 30

[2]Table 5 provides detailed VLM & LLM token breakdowns.

Table 4: Ablation study results on a subset of LV-Bench.

Variant	ACC (%)	Input Token
VideoNav	**60.9**	33.6K
Ablation on Navigation Granularity		
w/o High-Level	57.4	48.3K
w/o Clip-Level	55.6	46.7K
w/o Frame-Level	47.6	12.1K
Basic RAG	34.6	**8.9K**
Ablation on Design Components		
w/o Context-Aware	57.8	36.4K
w/o Clustering	58.9	36.1K
w/o Navigation Guider	58.4	37.5K
w/o Time Zooming	57.9	39.2K
w/o Semantic Matching	57.0	43.4K

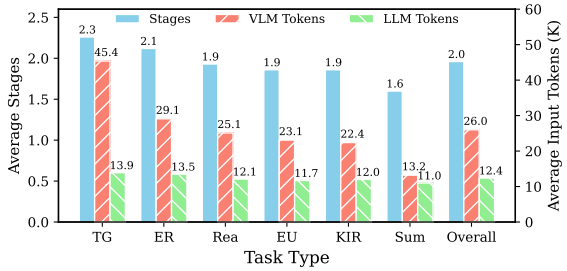

Figure 3: Navigation stages and token consumption across different tasks, evaluated on LVBench.

videos and 455 questions).

Effectiveness of Hierarchical Navigation. As shown in Table 4, the full framework achieves the best accuracy with good cost-efficiency. Removing high-level or clip-level navigation degrades both accuracy and token usage, as the system misses contextual information from early stages. Removing frame-level inference causes a large accuracy drop, confirming that textual descriptions alone are insufficient for fine-grained visual tasks. A basic RAG baseline that directly retrieves clips for LLM answering performs poorly, highlighting the necessity of structured hierarchical navigation.

Design Components. We further evaluate the contribution of specific design components. For offline construction, removing context-aware summarization ("w/o context-aware") or semantic clustering ("w/o clustering") hurts accuracy by breaking narrative coherence. For online navigation, the navigation guider, time-range zooming, and semantic matching each contribute. These results confirm that each component boosts overall robustness and efficiency.

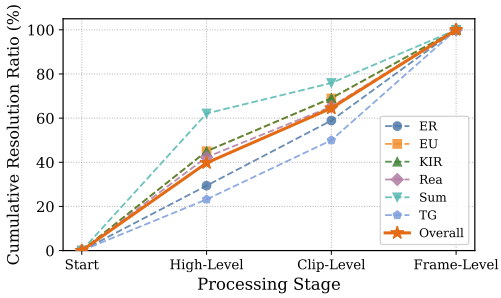

Figure 4: Cumulative query resolution rates across navigation stages, evaluated on LVBench.

4.5 Further Analysis of Navigation

We further analyze the navigation efficiency across different tasks on LVBench. The processing spans three stages: high-level navigation, detail navigation (clip-level) and frame-level inference.

Cost Efficiency across Tasks. Figure 3 shows the average navigation stages and token usage per task type. Summarization (Sum) exhibits the shallowest navigation depth (1.6 stages) and lowest VLM token usage, as high-level content captures the core narrative and many queries terminate early. Conversely, Temporal Grounding (TG) and Entity Recognition (ER) require deeper navigation and higher costs, since these tasks demand specific visual details and more frequent visual inspection. Furthermore, due to the complexity of temporal dynamics in TG tasks, the agent may resort to the fallback inference, which also increases the token consumption.

Effectiveness of Early Termination. Figure 4 shows the adaptive behavior via the cumulative query resolution across stages. For summarization, over 60% of queries are answered solely by high-level information, while some local-event summarization tasks demand subsequent processing. In contrast, the resolution curves for Temporal Grounding (TG) and Entity Recognition (ER) rise slowly, indicating reliance on clip-level details and frame inspection. Overall, about 40% of queries are resolved at the high-level stage, and clip details account for roughly 20% of resolutions, validating the efficiency our adaptive navigation design.

4.6 Impact of Different Model Backbones

We analyze the impact of different backbone models using the Qwen3 series on the LVBench subset (same as §4.4), with results in Table 5.

First, a stronger LLM (Qwen3-Next-80B-A3B)

Table 5: Accuracy and efficiency across different model backbones, evaluated on a sampled subset of LVBench. The token usage aggregates input and output tokens (4× weight for output). Normalized financial costs are calculated using official prices.

Model Used	Acc	$ Cost	VLM Tok	LLM Tok
VLM Inference Baselines				
Qwen3-VL-8B	57.8	3.3x	116.5k	None
Qwen3-VL-4B	54.4	2x	116.5k	None
VideoNav				
Qw3-VL-8B + Qw3-Next-80B-A3B	**64.3**	1.6x	29.3k	**12.9k**
Qw3-VL-8B + Qw3-30B-A3B	60.9	1.1x	**21.9k**	**12.9k**
Qw3-VL-8B + Qw3-8B	52.7	1.6x	41.8k	13.9k
Qw3-VL-4B + Qw3-30B-A3B	59.3	**1x**	29.1k	13.1k

yields higher accuracy than Qwen3-8B. This performance gap underscores the importance and complexity of online navigation, where reliable analysis is essential for precise decision-making. Notably, although the Qwen3-Next imposes a 2× unit inference cost, the total expenditure remains identical to the 8B version. Because its superior capability achieves more accurate navigation and effective early termination. Compared to the 8B version, Qwen3-Next yields an 11.6% accuracy improvement with no additional cost, and Qwen3-30B-A3B (our default setting) achieves an 8.2% accuracy gain, with only 69% of the cost.

Regarding the VLM backbone, VideoNav maintains relative robustness when replacing Qwen3-VL-8B with the lightweight 4B variant. This stems from our hierarchical index construction, decomposing complex long-video modeling into manageable clip descriptions, where smaller VLMs can also achieve decent performance. These highlight the effectiveness and generalization of our design.

5 Related Work

5.1 VLM for Long Video Understanding

Recent VLMs have advanced video understanding (Li et al., 2024a, 2025b; Tang et al., 2025; Bai et al., 2025; Guo et al., 2025), but applying them to long-form videos remains challenging due to the volume of visual information and context window constraints (Wang et al., 2025a; Zhou et al., 2025; Chen et al., 2025). Two main directions address this: context extension (Team et al., 2024; Wang et al., 2024b; Zhang et al., 2024; Shen et al., 2025), which expands model capacity but still incurs high computational cost from massive frame inputs, and visual compression (Li et al., 2025a, 2024b; Song et al., 2024; Cheng et al.), which reduces input size but risks losing fine-grained details.

5.2 Agent-based Video Frameworks

Recent studies have proposed agent-based frameworks as a promising solution for long video understanding (Zuo et al., 2025; Fan et al., 2024; Wang et al., 2024a; Ma et al., 2025; Pang and Wang, 2025). Typically, they decompose the task and iteratively perform planning, reasoning, frame selection, and information localization (Yang et al., 2025b; Zuo et al., 2025; Zhang et al., 2025b; Wang et al., 2025c; Zhang et al., 2025a). These approaches alleviates the length constraint of long videos and improves understanding performance (Pang and Wang, 2025; Chen et al., 2025; Wang et al., 2025c). For instance, VCA (Yang et al., 2025b) continuously retrieves video clips, computes their relevance via a reward model, and selects useful content into a memory buffer until the task is solvable. AVP (Wang et al., 2025c) adopts an iterative plan-observe-reflect pipeline. At each step, the planner determines what, where, and how to interact with the video; and the observer executes the plan to extract query-related evidence; the reflector then evaluates the evidence to guide the next decision. However, most of these methods share two limitations: (1) they reason over fragmented frames without global temporal context, and (2) their iterative online loops incur heavy computation that scales with each query. VideoNav fundamentally differs by decoupling offline perception from online reasoning: the one-time index construction captures comprehensive context, while online navigation is lightweight and effective.

6 Conclusion

We introduce VideoNav, a framework that reformulates long-video understanding as index-centric query processing. By decoupling offline multimodal perception from online language reasoning, VideoNav constructs a multi-granular hierarchical index once and serves queries through adaptive coarse-to-fine navigation. Experiments across three benchmarks show that VideoNav achieves decent accuracy while significantly reducing token usage and latency, highlighting the potential of structured indexing and adaptive navigation for scalable, cost-effective video understanding.

Limitation

While VideoNav demonstrates significant effectiveness in long video understanding, we acknowledge its limitations. First, VideoNav relies on a multi-stage pipeline, so errors at early stages (e.g., inaccurate clip descriptions) may inevitably propagate and affect downstream results. Second, while we demonstrate that the offline index construction cost is amortized over $\sim$4 queries, applications with very few queries per video may not benefit from our design trade-offs.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. 2025. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. *arXiv preprint arXiv:2503.10200*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2024. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, and 1 others. 2025. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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*.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. 2024b. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*.

Yixuan Li, Changli Tang, Jimin Zhuang, Yudong Yang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. 2025a. Improving llm video understanding with 16 frames per second. *arXiv preprint arXiv:2503.13956*.

Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. 2025b. A survey of state of the art large vision language models: Benchmark evaluations and challenges. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1587–1606.

Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. 2024. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv preprint arXiv:2409.12961*.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and 1 others. 2024. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *arXiv preprint arXiv:2411.13093*.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. 2025. Drvideo: Document retrieval based long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18936–18946.

Ziqi Pang and Yu-Xiong Wang. 2025. Mr. video:" mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. 2025. Videorag: Retrieval-augmented generation with extreme long-context videos. *arXiv preprint arXiv:2502.01549*.

Yunhang Shen, Chaoyou Fu, Shaoqi Dong, Xiong Wang, Yi-Fan Zhang, Peixian Chen, Mengdan Zhang, Haoyu Cao, Ke Li, Shaohui Lin, and 1 others. 2025. Long-vita: Scaling large multi-modal models to 1 million tokens with leading short-context accuracy. *arXiv preprint arXiv:2502.05177*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

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*.

Nano vectordb Team. 2025. nano-vectordb: A simple, easy-to-hack vector database. https://github.com/gusye1234/nano-vectordb.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, and 1 others. 2025a. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967.

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

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. 2024b. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2025b. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3272–3283.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S Ryoo, and Juan Carlos Niebles. 2025c. Active video perception: Iterative evidence seeking for agentic long video understanding. *arXiv preprint arXiv:2512.05774*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. 2025b. Vca: Video curious agent for long video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20168–20179.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Huaying Yuan, Zheng Liu, Minghao Qin, Hongjin Qian, Yan Shu, Zhicheng Dou, Ji-Rong Wen, and Nicu Sebe. 2025. Memory-enhanced retrieval augmentation for long video understanding. *arXiv preprint arXiv:2503.09149*.

Ce Zhang, Yan-Bo Lin, Ziyang Wang, Mohit Bansal, and Gedas Bertasius. 2025a. Silvr: A simple language-based video reasoning framework. *arXiv preprint arXiv:2505.24869*.

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*.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. 2025b. Deep video discovery: Agentic search with tool use for long-form video understanding. *arXiv preprint arXiv:2505.18079*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, and 1 others. 2025. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13691–13701.

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. 2025. Videolucy:

Deep memory backtracking for long video understanding. *arXiv preprint arXiv:2510.12422*.

A Qualitative Case Study of VideoNav

A.1 Examples of the Navigating Process

To provide a concrete demonstration of VideoNav's mechanism, we present two qualitative examples illustrating how the framework handles queries with varying granularity.

Query: Which country has the same number of players as the USA?

High-Level Nav:
The blocks list all eight competitors in the Women's Floor Exercise, each representing the country. Australia has one player, and …
Therefore, the answer is Italy.
Action: Answer; Content: Italy

Figure 5: Example of high-level early termination.

As shown in Figure 5, the user asks: *Which country has the same number of players as the USA*, over a long gymnastics competition video. This query requires aggregating information scattered across the video, as player nationalities typically appear on-screen only before or after their individual performances. During the offline phase, VideoNav's hierarchical index successfully captures these details within the Block-Level content. Therefore, during the online phase, the High-Level Navigation can extracts the answer directly from these textual content. The process triggers *early termination*, avoiding the consequent processing. This case demonstrates VideoNav's capability to capture global information and its cost-efficiency with adaptive navigation.

Query: What did the vlogger do, right before going to sleep?

High-Level Nav:
The blocks do not explicitly describe the actions immediately before sleeping … I should seek …
Action: Seek; Content: [00:24:00 - 00:29:00]

Detail Nav:
The clips mention several activities … but it's not sure about the exactly pre-sleep action …
Action: Seek; Content: [00:27:00 - 00:29:00]

Visual Inspection:
The video shows the vlogger …, directly followed by the scenes of sleeping …
Answer: He attended a party

Figure 6: Example of adaptive top-down navigation.

Figure 6 illustrates a query involving temporal

dependencies and scene recognition: *What did the vlogger do right before going to sleep*. Unlike the first case, the high-level navigation finds the block summaries insufficient to identify the action, and provides the target time-range containing the pre-sleep event. The framework then proceeds to detail navigation. Since the textual description of the clips are still not enough to decide the exact behavior, the agent triggers visual inspection. By invoking the VLM to process the raw frames within the focused time range, the system accurately identifies the action. This highlights the effectiveness of VideoNav's adaptive, coarse-to-fine strategy in resolving multi-granular tasks.

A.2 Failure Discussion

Similar to most other agent systems, VideoNav's performance is affected by the quality of LLM and VLM processing. We primarily observe failures stemming from two sources: perception errors and reasoning errors. First, during the offline description or online visual inspection, the VLM may sometimes hallucinate visual content (e.g., misidentifying a medicine bottle as a water cup) or overlook some details. These inaccuracies inevitably lead to erroneous scene localization and incorrect information understanding. Second, the LLM agent may exhibit reasoning failures during online navigation, such as errors in calculating global statistics or sometimes wrongly predicting the target time ranges. While our framework is designed with some robust components (e.g., context-aware construction, dual-path retrieval), these model-dependent errors remain a bottleneck. We anticipate that continued advancements in backbone models will drive further improvements. And we will explore additional optimization strategies to mitigate these errors in future work.

B More Details

B.1 Implementation

We implement VideoNav leveraging the Qwen3 (Yang et al., 2025a; Bai et al., 2025) series of open-source models, specifically utilizing the instruct-tuned versions. For our primary experiments, we employ Qwen3-30B-A3B-Instruct-2507 as the LLM backbone.

We prioritize a balance between accuracy and computational overhead. Consequently, we access the model of instruct mode rather than the intensive deep thinking (reasoning) mode, which may generate redundant reasoning traces. During the navigation, to maintain accuracy within this efficient setting, we prompt the LLM to output a concise preliminary analysis before generating the final decision (see complete prompts in Appendix D.2). Following extensive prior works, we conduct experiments with a single run, due to the significant computational cost of VLMs for long video processing.

Regarding data management, we implemented a lightweight Key-Value (KV) storage system to manage offline textual descriptions, while vector embeddings are stored and retrieved using NanaVectorDB (vectordb Team, 2025). During the online stage, specifically for dual-path retrieval, we set the maximum clip budget of 16. This setting retains essential evidence while reducing irrelevant noise and redundant information.

We also provide the anonymous code as supplementary materials via `https://anonymous.4open.science/r/VideoNav-51C1`.

B.2 Datasets

In our experiments, we utilize three publicly available long-video understanding benchmarks: LV-Bench (Wang et al., 2025a), Video-MME (Fu et al., 2025) and MLVU (Zhou et al., 2025), adhering to their established configurations. LV-Bench and MLVU are released under the CC-BY-NC-SA-4.0 license, while Video-MME is intended solely for academic research. Our utilization is confined to non-commercial academic research, which aligns with their designated purposes and original access conditions. These widely-recognized public benchmarks are curated for video understanding research and do not contain offensive content. They cover diverse real-world domains (e.g., movies, documentaries, sports, vlogs). We rely on the content-filtering procedures applied by the original dataset providers and perform no additional identity extraction. For the main experiments, we evaluate on the full question set of each benchmark. For the ablation studies, due to the high evaluation cost, we use the most challenging benchmark, LV-Bench, and randomly sample a subset of 30 videos with 455 questions.

C The Use of AI Assistants

This paper utilized LLMs only for language polishing in parts of the text.

D LLM and VLM Prompts

D.1 Prompts for Offline Construction

This section presents the prompts for offline index construction. These include the VLM prompt for clip description, as well as the LLM prompts for global summary and block construction.

Prompt for Clip Description

You are an expert video assistant. You will be provided with a video segment. Your task is to describe its content in details. Please focus on scenes, objects & persons, key text information, and central events. Also, there is an audio transcript of this clip. You can refer to it for more context (e.g. get entity names), but please focus on the actual video content and don't repeat the audio content too much.

More instructions:
1. Preserve the sequence flow.
2. Identify well-known entities or persons, if you are certain of their identity.
3. Be concrete. Don't use vague adjectives; instead, provide direct descriptions (e.g., use "snow-capped mountains" instead of "magnificent scenery").
4. Directly describe the observed visually content and text, don't interpret their implicit meanings.
5. Only rely on the provided content. Do not add guesses or extra conclusions/comments.
6. Output the natural language description, without special structure or formatting.

Prompt for Global Summary

You are an expert video content summarizer. You are given ordered video segments with timestamps, audio texts and visual descriptions. Please summarize the main content of the whole video in 5-10 sentences.

Your instructions:
- Please rely more on the visual captions and use audio text for context.
- Preserve the sequence flow. Please maintain key information such as overall scenes, core objects, main person names and main events.
- Focus on the main context, and avoid being distracted by minor details.
- Since the segment descriptions may have small errors, please do not include confusing or uncertain information.
- Your summary should only rely on the provided content, without any guess or assumption.

Prompt for Block Construction

You are an expert content summarizer. You will be provided with a set of ordered video segments, each accompanied by timestamps, summarized audio text, and visual descriptions. Your task is to summarize the content of these segments with of about 5-10 sentences.
A full-video abstract is also provided for contextual reference, and the segments are only a part of the entire video.

Your instructions:
- Please rely more on the visual captions and use audio text for context.
- Do not use vague adjectives like "beautiful" or "magnificent". Please describe directly (e.g., instead of "magnificent view," say "snow-capped mountains").
- Do not repeat common information that is already covered in the full-video abstract.
- Maintain important information such as scenes, main objects, main person names and key events.
- Since the segment descriptions may have small errors, do not include confusing or uncertain information.
- If the same entities or persons appear in consecutive segments (e.g., wearing the same clothes and having continuous actions), you should identify them as a single person in your summary.
- Please preserve the time flow.

D.2 Prompts for Online Navigation

This section presents the prompts for online adaptive navigation. It mainly covers three processing stages: high-level navigation, detail navigation, and

vision inspection. Notably, the prompts for high-level and detail-level navigation are similar. In addition, we also provide the LLM prompts for the navigation guider and the prompts for basic VLM evaluation (standard VLM inference).

Prompt for High-Level Navigation

You are an expert video analyst. You will be provided with a user question, a full-video summary, and a lists of video blocks (each with timestamps and descriptions).

Your task is to first output your brief analysis, and then execute one and only one of the two actions (answer or seek):
1) Answer: Use this if the current summary and video blocks are sufficient to provide an answer.
2) Seek: Use this if the current content is missing key information, or if the question refers to a specific moment not fully described.

Instructions for Seek:
- Time Window: Output a specific time window (e.g., [00:09:00 - 00:13:00]) to zoom in on relevant events.
- Flexible Range: This time range does not require strict alignment with block boundaries. For example, if the question is "who did Bob meet at 10:05", since this event is related to a narrow time window, you can seek around that specific time (e.g., 00:09:00-00:11:00). This range may be shorter than a full block or span across different block boundaries.
- Other Cases: If the specific time range cannot be decided, output No Range.
- You can also output two time windows if really necessary, separated by commas (e.g., [00:02:00 - 00:05:00], [00:15:00 - 00:19:00]).
- Duration Limit: The total time range should not exceed 15 minutes.
- Time Format: Use mm:ss or hh:mm:ss for timestamps.

Note:
- Individual video blocks may contain minor inaccuracies. If there is conflicting information, you can rely on the overall context.
- Some questions may target at the specific time points or detailed scenes. Since the video blocks just contain high-level information, you may need to seek for these narrow questions.
- There's also a Query Hint to help you on the thought and action.

Output Examples and Format:
Thought:
< your brief analysis here >
Action:
Answer
Answer:
< your answer >

Thought:
< your brief analysis here >
Action:
Seek
Seek:
[00:20:00 - 00:25:00]

Prompt for Detail Navigation

You are an expert video analyst. You will be provided with a user question, a full-video summary, and a lists of video segments (each with timestamps, ASR audio transcripts, and visual captions).

Your task is to first perform brief analysis, and then execute one and only one of the two actions (answer or seek): 1) Answer: Use this if the current summary and video segments can provide an answer.
2) Seek: Use this if the current content is missing key information, or if the question refers to a specific moment not fully described.

Instructions for Seek:
- Time Window: Output a specific time window (e.g., [00:09:30 - 00:10:30]) to zoom in on relevant events.
- Flexible Range: This time range does not require strict alignment with segment boundaries, it may be shorter than a full segment or span across different segment boundaries. For example, if the question is "what happened during 10:05 to 10:32", since this time is specific, you can seek directly (i.e., 00:10:05-00:10:32).

- Other Cases: If the time range cannot be decided (e.g., no related segment or no specific time point), output No Range.
- Duration Limit: The total time range should not exceed 2 minutes.
- Time Format: Use ‘mm:ss‘ or ‘hh:mm:ss‘ for timestamps.

Note: There’s also a Query Hint to help you on the thought and action.

Output Examples and Format:
...

Prompt for Visual Inspection

You are an expert video analyst. You will be provided with a user question, a full-video summary, and a partial video segment.
Your task is to answer the user’s question based on the provided content.

Your Instructions
- Before the answer, please output a brief analysis.
- Note that the video segment is only a small part of the full video, and it starts at a specific timestamp (given in the segment info).
- If the provided content is insufficient to answer the question, you should output "No answer".

Output Format
Thought:
< your brief analysis here>
Answer:
No answer

Prompt of the Navigation Guider

You are a query analyzer for a video understanding system. You will be provided with a video-summary and a user question.
Your task is to analyze the user question and generate a specific Search Hint to guide further video segment retrieval or answering processes.

Note:
- The video summary serves only as a brief context. The hint is used to locate the information among detailed video segments in subsequent steps.
- Your hints should primarily follow the hint examples below. Do not deviate too much from the templates.
- Do not answer the user question directly. Do not fabricate timestamps.

User questions may fall into several types. You should classify the question type first, then generate the corresponding hint.

Your Guidelines and Hint Templates
1. Sequence Logic
- Definition: Questions involving single sequential or ordinal information ("the first performer", "the second award").
- Question Example: "what is the third singer wearing?"
- Hint Example: Identify the [third singer] according to the sequence order, then answer the question or decide the time range.
2. Visual Details
- Definition: Questions involving some visual scenes, events or actions.
- Question Example: "what does the cartoon mouse use to hit the cartoon cat?"
- Hint Example: Locate the scene of [mouse hitting cat], then answer the question or decide its time range.
3. Explicit Time
- Definition: Questions specifying exact time points for analysis.
- Question Example: "why is Nick sad at 10:10?"
- Hint Example: The time is specific, you should locate on the corresponding time range directly.
4. Aggregation
- Definition: Questions requiring aggregation, counting, or event ordering, which may involve multiple clips.
- Question Example: "How many singers are singing with guitar?"
- Hint Example: If the content clearly contains several information clips about [singers singing with guitar], generate the answer. Else, output the time range or No Range.
5. Global
- Definition: Questions targeting the overall content (e.g. summarization, topic, theme).
- Question Example: "what’s the main topic

of this video?"
- Hint Example: This is a global query; please answer based on the overall context.

Format Examples:
Video Summary:
... This video is a wildlife documentary ...
Question:
Which animal does the protagonist encounter in the forest scene?
Type:
Visual Details
Hint:
Locate the scene where the protagonist encounters the animal in the forest, then answer the question or decide the time range.

Prompt for Standard VLM Inference

You are an expert video analyst. You will be provided with a user question and a long video.
Your task is to answer the user's question based on the provided video content (decide the most accurate answer option).
Before the answer, please output a brief analysis.

Output Format:
Thought:
< your brief analysis here >
Answer:
< your answer>