VideoITG: Multimodal Video Understanding with Instructed Temporal Grounding

Shihao Wang[1*], Guo Chen[2*], De-An Huang[3], Zhiqi Li[2*], Minghan Li[4], Guilin Liu[3], Jose M. Alvarez[3], **Lei Zhang**[1†], **Zhiding Yu**[3†]

[1]The Hong Kong Polytechnic Univ. [2]Nanjing Univ. [3]NVIDIA [4]Harvard Univ.
https://nvlabs.github.io/VideoITG/

Abstract

*While Video Large Language Models (Video-LLMs) have shown significant potential in multimodal understanding and reasoning tasks, how to efficiently select the most informative frames from videos remains a critical challenge. Existing methods attempt to optimize frame sampling by reducing inter-frame redundancy or employing unsupervised event localization. However, these approaches often fall short in handling complex instruction-following tasks and scenarios that demand precise temporal modeling, resulting in limited performance in both semantic alignment and temporal reasoning. To address the above challenges, we introduce Instructed Temporal Grounding for Videos (**VideoITG**), a framework aiming to adaptively customize frame sampling strategies based on user instructions. Specifically, we design the VidThinker pipeline, which automates annotation by generating instruction-conditioned captions, retrieving relevant video segments, and selecting key frames to enable efficient supervision. Using VidThinker, we build the VideoITG-40K dataset with 40K videos and 500K temporal grounding annotations. Our plug-and-play VideoITG model leverages Video-LLMs' visual-language alignment and reasoning for discriminative frame selection. VideoITG consistently boosts the performance on multiple multimodal video understanding benchmarks, demonstrating its effectiveness and potential.*

1. Introduction

The rapid progress of Video Large Language Models (Video-LLMs) has opened new frontiers in video understanding, enabling complex tasks such as captioning [4, 8, 10, 21, 52, 73], visual question answering [2, 5, 15, 27, 38, 42, 56, 72], and even embodied-agent interaction [3, 7, 9, 16, 23, 31]. However, these models still struggle with long videos, where high memory cost and computation overhead limit their ability to process extended temporal contexts. A common workaround is uniform frame sampling—simple yet suboptimal—often missing key frames critical for semantic and temporal reasoning, thereby constraining overall performance.

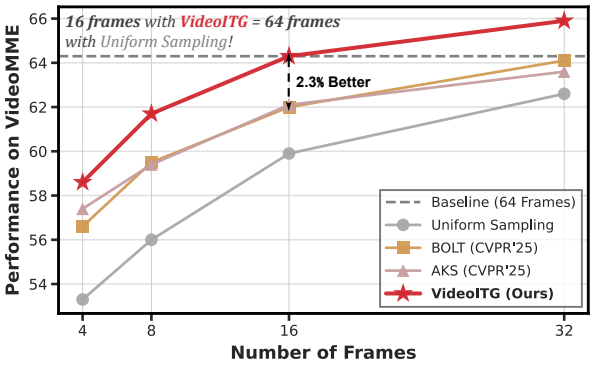

Figure 1. **Comparison of different frame selection methods on VideoMME with LLaVA-Video-7B.** VideoITG consistently enhances baseline methods and achieves state-of-the-art performance.

To alleviate this challenge, prior studies have explored multiple directions. One class of approaches focuses on reducing spatiotemporal redundancy through pooling [45, 57], similarity pruning [66], or clustering-based compression [25, 68], retaining only essential frames. Another line extends model sequence length to capture long-term dependencies [49, 53], yet such strategies incur high computation and risk information dilution. Other methods incorporate question-centric cues for frame selection [28, 61], demonstrating superiority over uniform sampling (1). For example, SeViLA [61] applies BLIP-2 [26] to process each frame independently before selecting keyframes, which are then fed into video reasoning pipelines. Nevertheless, the absence of temporal modeling across frames hinders effective reasoning over multi-event or time-sensitive queries.

Despite progress in compressing or extending temporal

*Work done during an internship at NVIDIA.
†Equal corresponding and advising authors.

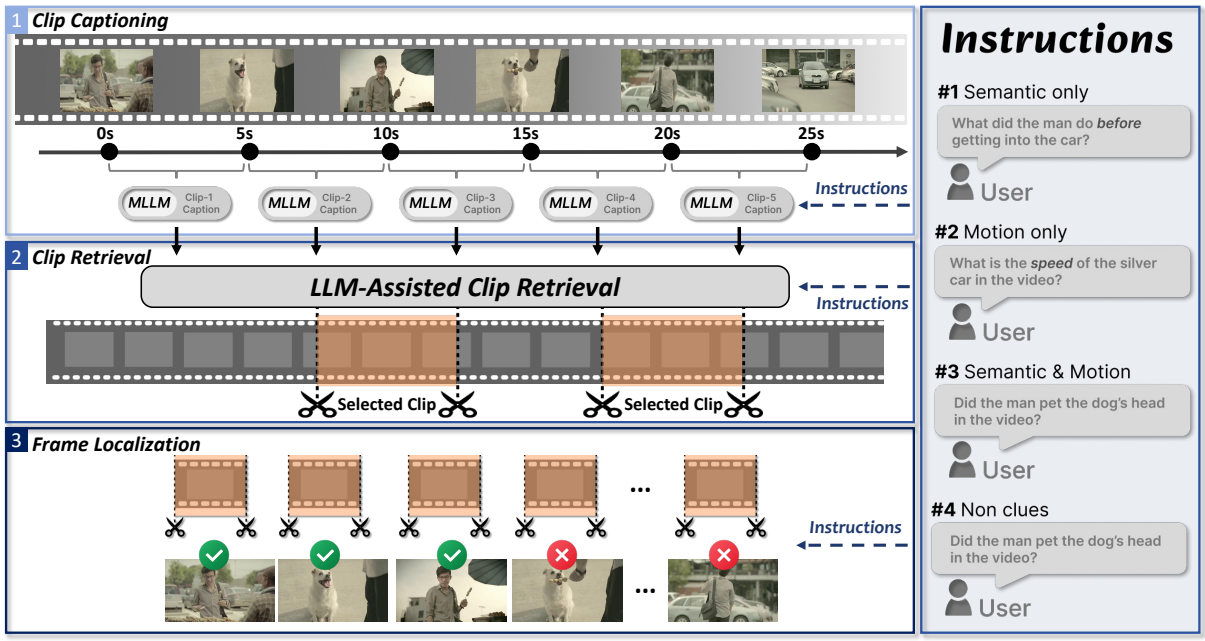

Figure 2. **Overview of the *VidThinker* annotation pipeline for VideoITG.** It consists of three human-inspired stages: (1) clip-level captioning under instructions; (2) instruction-guided relevant clip retrieval; and (3) fine-grained frame-level localization.

contexts, a substantial performance gap remains between short and long videos—primarily due to the lack of large-scale, instruction-guided temporal grounding data. When humans analyze long videos, they rarely process all frames at once; instead, they skim for global context, identify question-relevant cues, and zoom in on discriminative moments. Inspired by this human strategy, we propose **Instructed Temporal Grounding for Videos (VideoITG)**, which integrates user instructions directly into the frame selection process. While traditional temporal grounding [24, 43, 50] focuses on localizing events using single descriptive queries, **VideoITG** introduces instruction-driven temporal reasoning, adaptively customizing the sampling strategy for each task. Unlike prior frame selection frameworks [18, 39, 51, 61, 63], our method handles multi-temporal and multi-cue scenarios by (i) localizing temporal cues across clips for relational reasoning, (ii) employing hybrid sampling for dynamic event variations, and (iii) maintaining holistic coverage for content verification and captioning.

To support VideoITG, we construct a large-scale dataset via an automated annotation pipeline named *VidThinker*. As shown in Fig. 2, VidThinker automates data generation through instruction-conditioned clip captioning, instruction-guided retrieval, and fine-grained frame localization. Driven by GPT-4o [41] reasoning, VidThinker emulates a "Needle-in-a-Haystack" process to retrieve relevant moments and provides balanced supervision across four instruction types: **(1) semantic-only**, focusing on appearance; **(2) motion-only**, emphasizing dynamic cues; **(3) semantic & motion**, for joint reasoning; and **(4) non-clues**, open-ended video-level prompts that require maximizing visual diversity across the entire video.

The resulting **VideoITG-40K** dataset contains 40K videos (30s–3min) and 500K instruction-grounded annotations—surpassing existing temporal grounding datasets by more than **4**× in both scale and instruction quality. Building on this foundation, we design a family of VideoITG models—featuring text generation, anchor-based causal attention, and full-attention pooling—to effectively align temporal cues with user instructions.

In summary, our key contributions are as follows:

- **VideoITG-40K dataset.** Built via the automated *VidThinker* pipeline, we curate **VideoITG-40K**, it contains 40K videos and 500K fine-grained, instruction-aligned annotations, substantially expanding the scale and diversity of existing temporal grounding resources.
- **VideoITG models.** We propose three complementary model variants that explore distinct attention and decoding mechanisms, offering a unified, plug-and-play framework adaptable to diverse Video-LLMs.
- **Consistent improvement.** Across benchmarks and models, VideoITG boosts accuracy with fewer frames: on VideoMME (Fig. 1), 16 frames with VideoITG match 64-frame uniform sampling and are comparable to SOTA methods using 32 frames.

2. Related work

Video large language models. Recent advances in Video-LLMs address the temporal and spatial complexity of

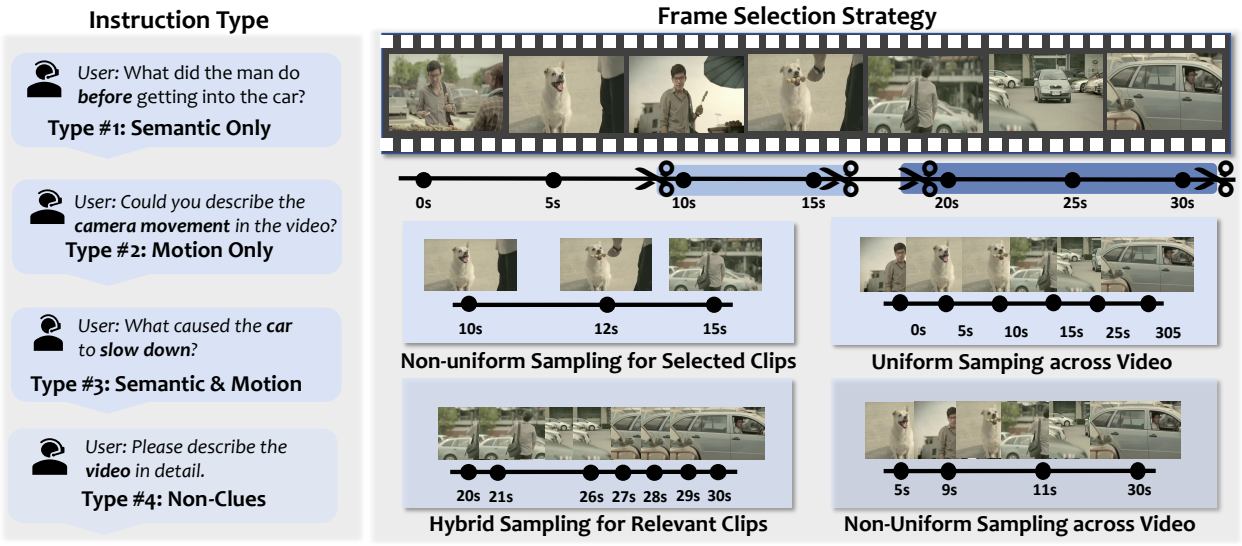

Figure 3. **Illustration of four instruction types and their corresponding frame selection strategies in VidThinker.** For semantic-focused instructions, the system selects diverse frames capturing key visual clues. For motion-focused instructions, frames are uniformly sampled to capture dynamic changes. When both semantic and motion cues are required, a hybrid sampling strategy is applied. For vague or open-ended instructions, the system samples a minimal yet diverse set of frames across the video for holistic coverage.

long videos through several strategies. Visual feature compression [33, 35, 54, 60, 74] is achieved by modules like Q-Former [47] and Perceiver Resampler [74], which merge frame features into fixed queries. Spatial pooling [37, 57, 58] helps preserve long-range temporal information efficiently. Some models extend sequence length for longer inputs [6, 49, 53, 69], but this often increases computational cost [46, 55]. To reduce redundancy, similarity-based frame filtering is used [22, 45], though fixed thresholds may miss real-world diversity.

Keyframe selection for Video-LLMs. The goal of keyframe selection is to choose a compact subset of frames that preserves task-critical semantics and temporal evidence while maintaining sufficient video-level coverage for long-form understanding. Recent methods increasingly incorporate user instructions to rank or retrieve frames most relevant to the query and then feed only these keyframes to the downstream Video-LLM [32, 36, 48, 62, 70]. AKS [48] performs adaptive sampling guided by temporal characteristics to reduce redundancy without sacrificing content coverage. Q-Frame [70] introduces a query-adaptive importance estimator with multi-resolution selection, allocating higher resolution to query-critical segments. A subset of methods selects frames via text–video and video-context similarity, scoring alignment under budget constraints [59, 62].

Video temporal grounding. Video Temporal Grounding [14, 43, 44, 50] is a common task in video understanding that associates specific video moments with their corresponding timestamps, while Temporal Localization focuses on accurately identifying these moments within untrimmed videos [1, 29, 34]. Current Video-LLMs [19, 45, 50] have begun to leverage temporal grounding for frame selection by linking video with temporal cues; however, existing methods [20, 61, 63] mostly focus on single-time retrieval, which take descriptive annotations as input, limiting their generality and robustness in handling diverse real-world scenarios.

In this paper, our VideoITG leverages instructed temporal grounding, automated annotation, and a plug-and-play design to align sampling with user instructions, achieving superior performance and scalability on multimodal video understanding benchmarks.

3. VideoITG-40K: dataset construction

3.1. VidThinker: automated annotation pipeline

When humans search for information in long videos, they typically proceed in three steps: (i) extracting key cues from the instruction, (ii) retrieving a coarse temporal window, and (iii) fine-grained localization of the target event. We therefore propose *VidThinker*, a fully automated and interpretable pipeline that mimics this three-step reasoning for instruction-guided temporal localization. It comprises three interdependent stages—Instructed Clip Captioning, Instructed Clip Retrieval, and Instructed Frame Localization—that progressively narrow the search space while strengthening alignment with the instruction.

i) Instructed Clip Captioning: The video v is uniformly divided into short clips (5 seconds each), denoted as $\{v_i\}_{i=0}^{n}$. For each segment, we employ LLM to extract salient phrases that capture the core information needed to fulfill the instruction. For example, given the question ($q =$"*What does the man playing the drums do with his feet as he plays the*

drum?") and the answer (a ="*moves his feet*"), the system distills the essential action phrase: k ="*The man playing the drums moves his feet and hits the drums with his hands.*" We then input the extracted phrases alongside raw video clips into the VLM to generate clip-level descriptions $\{c_i\}_{i=0}^{n}$ in a recurrent manner. The extracted phrases serve as reference cues to guide the model's attention towards salient elements within each clip. However, the VLM strictly adheres to visual evidence and it only incorporates information from the extracted phrases when it is explicitly observable in the current clip. This ensures that the system will not hallucinate or infer content solely based on the extracted phrases, maintaining descriptions grounded in visual content. The process can be formulated as follows:

$$k = \mathrm{LLM}(q, a), \quad c_i = \mathrm{VLM}(k, v_i). \tag{1}$$

Conditioning on these instruction- and answer-derived cues, we ensure each segment's annotation is relevant and informative, facilitating precise instructed temporal grounding.

ii) Instructed Clip Retrieval: The generated clip descriptions $\{c_i\}_{i=0}^{n}$ are organized sequentially and evaluated by an LLM for the relevance to the QA pairs. Instead of simply assigning binary relevance scores, the LLM is instructed to perform chain-of-thought reasoning, explicitly considering both keyword matches and temporal relationships to directly output the indexes of relevant clips:

$$\mathcal{I}_{\mathrm{rel-clip}} = \mathrm{LLM}(\{c_i\}_{i=0}^{n}, q, a). \tag{2}$$

The chain-of-thought prompting requires the model to justify its selections based on both semantic and temporal cues, rather than relying solely on trivial keyword matching. This automation significantly improves the efficiency and the interpretability of relevant segment selection.

iii) Instructed Frame Localization: After coarse localization of video segment, *VidThinker* further refines the annotation by selecting key frames according to the instruction type. For each frame within the candidate segment, we prompt a LLM to perform a binary classification task: given the QA pair and a single frame, the LLM determines whether the frame is relevant (`yes`) or not (`no`) to the instruction. Formally, for each frame f_i in the candidate segment, the LLM is prompted as follows:

$$y_i = \mathrm{LLM}(f_i, q, a), \quad \text{where} \quad y_i \in \{\texttt{yes}, \texttt{no}\}, \tag{3}$$

where y_i indicates whether frame f_i is relevant to the QA. Only frames with positive responses ($y_i = \texttt{yes}$) are retained as the final temporal grounding results. This instruction-guided filtering allows *VidThinker* to achieve high precision in identifying the most informative frames for instructions.

3.2. Fine-grained grounding instruction

We adopt fine-grained frame selection strategies tailored to each instruction type, ensuring that the visual evidence matches the reasoning needs of each QA task. Since different instructions demand varying visual understanding, we categorize instructions by whether they require static semantics, dynamic motion, both, or no explicit cues at all (video-level). For each type, we adopt a matching frame selection strategy to align visual evidence with QA reasoning needs.

- **Semantic only**: Instructions query static appearance cues (e.g., people, objects, scenes). For example: *"What did the man do before getting into the car?" VidThinker* selects frames revealing the man's clothing and the guitar. After relevant segment localization, we select diverse frames that capture representative semantic clues to ensure comprehensive coverage. Concretely, we extract CLIP features per frame and compute cosine similarity between adjacent frames; a frame is retained when its similarity to the last selected keyframe falls below a scene-change threshold. Further algorithmic details are provided in the appendix.
- **Motion only**: Instructions focus on dynamic patterns (e.g., type, speed, direction). We adopt fixed-rate sampling within the localized segment to capture motion progression. For example: *"How does the person jump off the diving board?" VidThinker* selects frames spanning takeoff, mid-air, and water entry.
- **Semantic & Motion**: Instructions jointly require static semantics and dynamic changes. We apply fixed-rate sampling in motion-relevant regions while preserving semantically informative frames, balancing both needs. For example: *"Could you describe the camera movement in the video?" VidThinker* selects frames showing hand drumming and foot movement simultaneously.
- **Non Clues**: Entire video-level instructions without clear semantic or motion anchors. We samle a compact yet diverse set of frames across the entire video (e.g., beginning–middle–end) to ensure holistic coverage with minimal redundancy. For example: *"Please describe the video in detail."*

3.3. Dataset statistics

Leveraging the proposed *VidThinker* pipeline, we construct **VideoITG-40K** from LLaVA-Video [71]: 40K videos and 500K instruction-grounded annotations for temporal grounding. The entire annotation is automated by *VidThinker*, ensuring efficiency, consistency, and alignment with diverse instruction types. The videos average 120s and cover three duration bands (30–60s, 1–2min, 2–3min). Each video has 10–15 QA pairs (multiple-choice and open-ended). As summarized in Table 1, VideoITG-40K is nearly 4× larger than DiDeMo [1] (10.6K) and QVHighlights [24] (10.2K), and far exceeds QuerYD [40] (2.6K) and HiREST [64] (3.4K). Unlike prior descriptive-query datasets, VideoITG-40K is explicitly instruction-guided, enabling precise, query-

conditioned temporal localization.

Table 1. **Comparison of dataset statistics for temporal grounding and highlight detection datasets.**

Dataset	# Videos	# Queries	Avg. Duration	Instructed?
DiDeMo [1]	10.6K	41.2K	29s	No
QuerYD [40]	2.6K	32K	**278s**	No
HiREST [64]	3.4K	8.6K	263s	No
Charades-STA [17]	6.7K	16.1K	30s	No
QVHighlights [24]	10.2K	10.3K	150s	No
VideoITG-40K	**40K**	**500K**	120s	**Yes**

4. VideoITG: model design

In this section, we explore how to utilize our VideoITG-40K dataset to train the model for the **Instructed Temporal Grounding** task, aiming to optimize video frame selection and enhance the performance of Video-LLMs. As illustrated in Fig. 4, our framework comprises three modules: (1) a vision encoder (*e.g.*, ViT) that maps video frames into text-aligned visual features F, (2) a VideoITG module that performs instruction-guided frame selection $\mathcal{I}_{\text{rel}}$, and (3) a VideoLLM that generates answers a conditioned on the selected frames $F_{\mathcal{I}_{\text{rel}}}$ and the question q. The process can be described as folllows:

$$F = \text{VIT}(v) \tag{4}$$

$$\mathcal{I}_{\text{rel}} = \text{VideoITG}(F, q) \tag{5}$$

$$a = \text{VideoLLM}(F_{\mathcal{I}_{\text{rel}}}, q) \tag{6}$$

The VideoITG module follows a plug-and-play design philosophy, driven by two core objectives: (1) enhancing the alignment between visual and language tokens to improve instruction following, and (2) strengthening contextual encoding to capture multi-granular temporal cues. With the above considerations, we develop three model variants: text generation-based classification, anchor-based classification, and pooling-based classification, as illustrated in Fig. 4 (b).

Variant A: Text-generation-based classification. As shown in Fig. 4(b, left), this variant reformulates the Instructed Temporal Grounding task as a next-token prediction problem, where the model sequentially outputs text tokens conditioned on video and instruction features. This formulation naturally aligns with the core training paradigm of existing Video-LLMs, thereby preserving their strong vision-language alignment and instruction-following abilities. Similar generative frameworks have also been adopted in prior time-sensitive models such as TimeChat [44] and Grounded-VideoLLM [50].

Variant B: Anchor-based classification. To move beyond token-by-token generation, this variant adopts a discriminative paradigm that directly classifies visual tokens at the frame level (Fig. 4(b, middle)). We initialize the model from a pretrained Video-LLM while maintaining its causal attention mask to retain temporal consistency. However, the causal mask prevents visual tokens from accessing the instruction beforehand and restricts early frames from leveraging subsequent temporal cues. To mitigate this limitation, we insert an *anchor token* after the instruction, serving as a temporal mediator for each frame. Formally, for a video frame at timestamp t, the anchor token A^t is derived by global averaging over all spatial locations:

$$A^t = \frac{1}{M} \sum\nolimits_{i,j} F_{ij}^t, \quad t \in [1, T], \tag{7}$$

where F_{ij}^t denotes the visual feature at grid (i, j) of the t-th frame and M is the total number of patches per frame. The set $\{A^t\}_{t=1}^T$ bridges temporal dependencies across frames under causal attention.

Variant C: Pooling-based classification. As the causal attention mask restricts inter-frame communication, we further remove this constraint to enable full bidirectional attention between visual and textual tokens (Fig. 4(b, right)). For each frame, we aggregate its visual tokens through average pooling, followed by a classification head that determines instruction relevance, without introducing explicit anchor tokens. This full-attention design enriches temporal context modeling across frames and facilitates stronger interaction between instructions and visual evidence.

5. Experiments

5.1. Implementation details

We follow the training approach of LLaVA-Video [71], using the pretrained model as the initialization for our VideoITG model's pre-training. We employ SigLIP [65] as the vision encoder and Qwen2 [53] as the language model. Initially, we train the MLP projector on image caption datasets with a batch size of 256 and a learning rate of 1×10^{-3}. Then, we fine-tune all model parameters on the LLaVA-OV-SI [25] and LLaVA-Video datasets. During this stage, the video frame sampling rate is set to 64, and the LLM's maximum sequence length is set to 16K. We then train the VideoITG model on the proposed VideoITG-40K dataset, adjusting the video sampling rate to 1 fps.

Throughout training and inference, we employ a dynamic token spatial size strategy [35]. Across all stages, the LLM's learning rate is 2×10^{-5}, and in the final stage, the learning rate for the classification head is 2×10^{-4}. To fairly compare with other leading video LMMs, we primarily use results from their original papers. When results are unavailable, we integrate the models into LMMs-Eval [67] and assess them under consistent settings. Due to context length constraints, we support up to 512 video frames as input (with 16 visual tokens per frame) for the VideoITG model, from which we select the top 32 frames based on their scores by default.

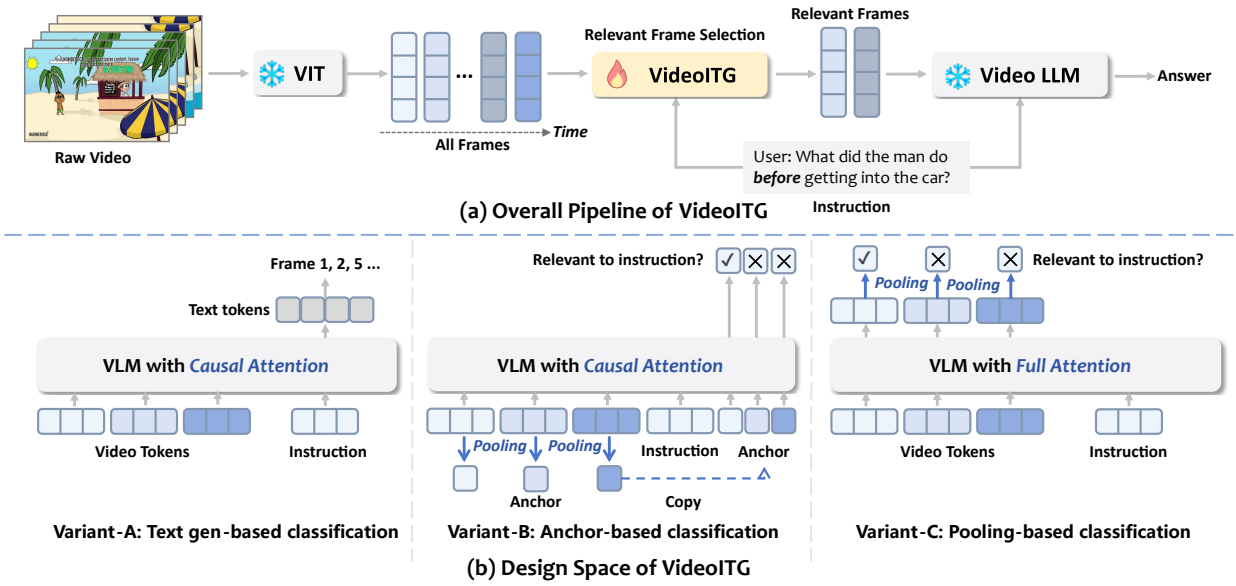

Figure 4. **VideoITG model design**: (A) Text generation aligns video and language tokens for sequential predictions. (B) Classification with causal attention utilizes anchor tokens for temporal cue management. (C) Classification with full attention facilitates interaction across visual and text tokens without anchors.

Table 2. **Results with different selection methods.** We bold the best results on each benchmark under the same Answering LMM. When comparing with Q-Frame, we adopt a slow–fast strategy of 4 high-resolution, 8 medium-resolution, and 32 low-resolution frames, where the total number of tokens is equivalent to that of 8 high-resolution frames.

Selection Methods	Answering LMM	Frames	LongVideoBench	MLVU	VideoMME			Average
			8 min	*12 min*	*S (2 min)*	*M (10 min)*	*L (40 min)*	
Uniform	LLaVA-OneVision-7B	8	54.2	58.9	63.6	52.0	45.7	54.9
BOLT [32]	LLaVA-OneVision-7B	8	56.1	63.4	66.8	54.2	47.3	57.6
Frame-VOYAGER [62]	LLaVA-OneVision-7B	8	-	65.6	67.3	56.3	48.9	59.5
VideoITG-8B	LLaVA-OneVision-7B	8	**60.1**	**68.7**	**72.0**	**57.7**	**49.4**	**61.6**
Uniform	Qwen2-VL	8	53.5	56.9	65.0	50.7	45.3	54.3
Q-Frame [70]	Qwen2-VL	8+16+32	58.4	65.4	69.4	57.1	48.3	59.7
VideoITG-8B	Qwen2-VL	8+16+32	**58.6**	**66.6**	**69.8**	**57.3**	**49.2**	**60.3**
Uniform	LLaVA-Video-7B	64	59.9	70.2	75.8	63.0	54.7	64.7
AKS [48]	LLaVA-Video-7B	32	59.6	74.3	75.1	63.9	51.7	64.9
QuoTA [36]	LLaVA-Video-7B	64	59.0	71.9	71.1	58.8	52.2	62.6
Gen-S [59]	LLaVA-Video-7B	54/50	63.3	73.4	-	-	-	-
VideoITG-8B	LLaVA-Video-7B	32	61.6	74.6	**77.3**	65.9	55.2	66.9
VideoITG-8B	LLaVA-Video-7B	64	60.9	**76.3**	76.1	**66.0**	**56.1**	**67.1**

5.2. Main results

Comparisons with other frame selection methods. As shown in Table 2, our proposed VideoITG demonstrates ***three notable advantages*** over existing selection strategies:

(i) Consistant Improvement: VideoITG consistently outperforms Uniform sampling across all benchmarks and model settings. It achieves an average gain of **+6.7** (54.9→61.6) on LLaVA-OneVision-7B and **+6.2** (54.1→60.3) on Qwen2-VL, with similar trends on LLaVA-Video-7B (+2∼3 points). These steady improvements across diverse models and datasets highlight the **robustness** and **general applicability** of our frame selection strategy in enhancing video understanding under challenging scenarios.

(ii) Precision in Selection: When using LLaVA-Video as baseline, VideoITG achieves superior accuracy using only 32 selected frames, outperforming other methods that use **more frames** (*e.g.*, 50-64) on MLVU and VideoMME, highlighting the effectiveness of our strategy in identifying the most informative frames. As shown in Figure 1, our VideoITG method achieves comparable performance to uniform sampling with 64 frames using only 16 frames on VideoMME, and also outperforms other methods with 32 frames, highlighting the precision of our frame selection strategy.

(iii) Robust Performance: Compared to training-based methods such as Q-Frame and Frame-Voyager, VideoITG

Table 3. **Performance comparison of VideoITG integrated with different Video-LLMs, varying in both the size of the answering LLM and the number of sampled frames.**"UNI-k" denotes UNIform sampling of k frames, while "ITG-k" refers to selecting the Top k frames based on relevance scores generated by our proposed VideoITG.

LMM	Selection	LongVideoBench	MLVU	VideoMME			CG-Bench-mini	Average
		8min	*12min*	*S (2 min)*	*M (10 min)*	*L (40 min)*	*27min*	
InternVL2.5-8B	UNI-32	58.3	66.4	75.1	61.7	53.1	37.7	58.7
	ITG-32	**61.9 (+3.6)**	**75.0 (+8.6)**	**78.0 (+2.9)**	**67.1 (+5.4)**	**56.9 (+3.8)**	**46.7 (+9.0)**	**64.3 (+5.6)**
InternVL2.5-26B	UNI-32	55.6	71.3	78.1	67.1	56.9	40.6	61.6
	ITG-32	**63.0 (+7.4)**	**78.9 (+7.6)**	**80.8 (+2.7)**	**69.0 (+1.9)**	**59.9 (+3.0)**	**48.7 (+8.1)**	**66.7 (+5.1)**
InternVL3.5-8B	UNI-32	60.0	70.0	77.0	62.4	53.4	40.9	60.6
	ITG-32	**65.7 (+5.7)**	**74.1 (+4.1)**	**78.4 (+1.4)**	**65.9 (+3.5)**	**59.0 (+5.6)**	**47.6 (+6.7)**	**65.1 (+4.5)**
Qwen3-VL	UNI-32	59.1	64.1	76.0	60.9	55.1	40.1	59.2
	ITG-32	**63.6 (+4.5)**	**77.2 (+13.1)**	**79.9 (+3.9)**	**66.6 (+5.7)**	**60.3 (+5.2)**	**47.3 (+7.2)**	**65.8 (+6.6)**
LLaVA-Video-7B	UNI-32	58.7	66.8	76.3	60.3	52.7	35.8	58.4
	ITG-32	**61.6 (+2.9)**	**74.6 (+7.8)**	**77.3 (+1.0)**	**65.9 (+5.6)**	**55.2 (+2.5)**	**42.8 (+7.0)**	**62.9 (+4.5)**
Eagle2.5-8B	UNI-32	63.0	67.8	78.8	64.1	55.9	41.2	61.8
	ITG-32	**66.8 (+3.8)**	**76.5 (+8.7)**	**80.0 (+1.2)**	**67.8 (+3.7)**	**60.3 (+4.4)**	**49.0 (+7.8)**	**66.7 (+4.9)**

Table 4. **Empirical studies on the VideoITG-40k dataset and VideoITG model design.** We adopt Variant-C for subsequent experiments. "No Images" and "No Videos" indicate that image-text data (LAION-CC-SBU-558K & LLaVA-OV-SI) or video data (LLaVA-Video-178K) are excluded from pre-training, respectively.

Abaltion	Experiment	Videomme			MLVU	LongVideoBench	Average
		Short (%) ↑	**Medium** (%) ↑	**Long** (%) ↑	(%) ↑	(%) ↑	
Architecture Design	Variant-A-7B	51.0	44.8	44.4	45.7	56.8	48.5
	Variant-B-7B	77.9	66.0	56.2	74.6	61.3	67.2
	Variant-C-7B	**78.0**	**67.1**	56.9	**75.0**	**61.9**	**67.8**
	Variant-C-3B	77.1	64.8	56.0	74.5	61.5	66.8
Dataset Construction	No Clip Captioning	77.5	63.1	53.4	73.2	61.7	65.8
	No Frame Localization	77.6	65.8	56.8	74.1	61.5	67.2
Pre-training Data	No Videos	77.2	64.9	**57.4**	74.5	61.6	67.1
	No Images & Videos	76.6	63.0	54.4	69.1	58.6	64.3

achieves more substantial improvements across three long video understanding benchmarks. For instance, it boosts the average score from 59.5 (Frame-Voyager) and 57.6 (Q-Frame) to **61.6** and **60.3** under the same settings. These consistent gains highlight the strong adaptability of VideoITG across diverse scenarios.

We find that on LongVideoBench, increasing frames has limited impact, likely because many tasks are text referring. Our VideoITG also samples at lower resolution, indicating room for finer-grained content recognition.

Results on newer backbone (Qwen3-VL). To verify that our gains are not tied to a specific backbone generation, we evaluate VideoITG on the newer Qwen3-VL model. The results are included in Table 3, showing consistent improvements over the UNI-32 baseline across LongVideoBench, MLVU, VideoMME (S/M/L), and CG-Bench-mini.

Extension to more Video-LLMs. Extending our VideoITG on diverse Video-LLMs with difeerent model scales further presents its ***two compiling advantages***:

(i) Model Size Scalability: Table 3 demonstrates that integrating VideoITG with Video-LLMs of different sizes consistently yields substantial performance improvements over uniform sampling. For example, on InternVL2.5-8B, VideoITG improves the average score from 58.7% to 64.3% ($+5.6$), while on the larger InternVL2.5-26B, the improvement is from 61.6% to 66.7% ($+5.1$). Notably, InternVL2.5-8B with VideoITG even surpasses the InternVL2.5-26B baseline on both average score (64.3% vs. 61.6%) and long-video benchmarks such as CG-Bench (46.7% vs. 40.6%), indicating that effective frame selection can provide greater gains than simply increasing model size.

(ii) **Model Diversity Adaptability**: We further evaluate VideoITG across diverse Video-LLMs, each trained on different data distributions and objectives. The results show that VideoITG consistently outperforms uniform sampling for all models and benchmarks. For instance, on LLaVA-Video-7B, VideoITG raises the average score by 4.5 % (62.9% vs. 58.4%), and on Eagle2.5-8B, the improvement reaches 4.9 points (66.7% vs. 61.8%). These consistent gains across models with varying architectures and training data highlight

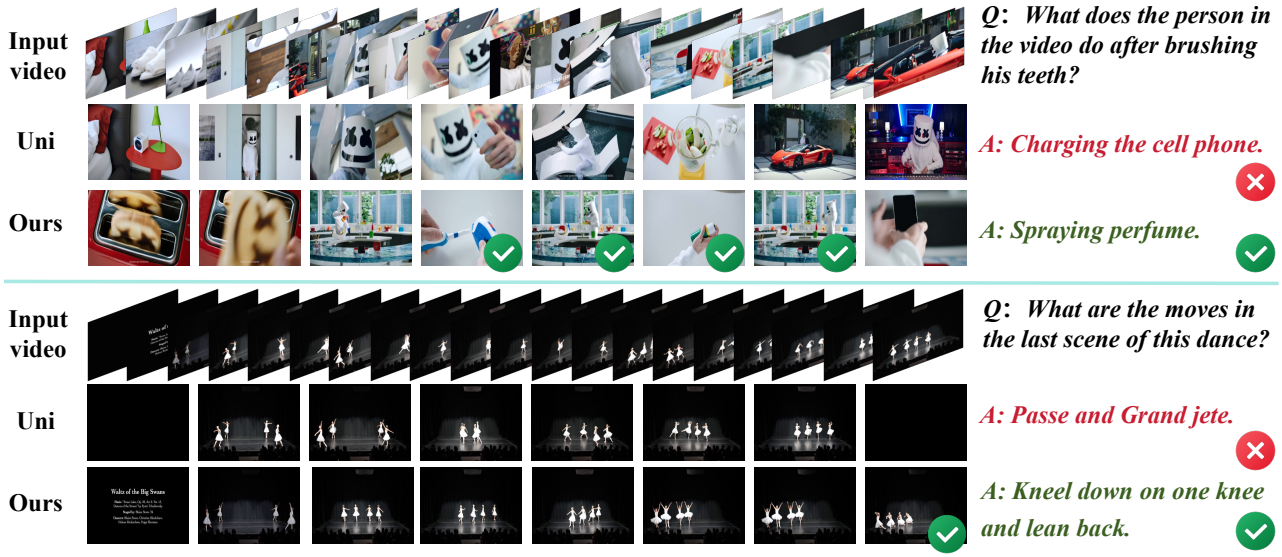

Figure 5. **Two examples of how different sampling strategies impact video understanding.** We mark the identified key frames that directly answer the question with green check-marks.

the strong adaptability and robustness of our method.

5.3. Ablation on VideoITG design choices

Table 4 presents a comprehensive analysis on the design of VideoITG, directly supporting our key contributions.

Architecture Design. First, we compare the three variants of our model architecture in Fig. 4. We observe that Variant A, which is based on the text generation paradigm, performs the worst. One possible reason is that text generation models trained with the next-token prediction paradigm suffer from sparse supervision due to teacher forcing, where previous frame selections influence subsequent ones, making the training process less efficient compared to discriminative classification models. We find that Variant C with full-attention outperforms Variant B with causal attention. This improvement may be attributed to full-attention's larger receptive field, which enables global temporal relationship modeling and allows all tokens to access the textual query. Moreover, model scale yields a consistent but modest gain: the 7B model surpasses the 3B counterpart across all benchmarks, raising the overall average from 66.8% to 67.8%.

Dataset Construction. We analyze our data annotation strategies to demonstrate the effectiveness of our pipeline. Ablation studies show that the performance degrades when Instructed Clip Captioning are removed (using a VLM to directly select temporal boundaries based on visual inputs), with accuracy dropping from 56.9% to 53.4% on Videomme Long videos and from 75.0% to 73.2% on MLVU. This demonstrates that ensuring information diversity is crucial for maintaining comprehensive feature representation of videos. Similarly, removing Instructed Frame Localization (learning all frame index within corase temporal boundaries using VideoITG Model) decreases performance, particularly on Videomme Medium videos (from 67.1% to 65.8%). These results confirm that both stages are essential for optimal model performance and validate our data construction approach of the VideoITG-40K dataset.

Pre-training Data. Finally, we investigate the impact of vision-language alignment pre-training on model performance. Our experiments reveal that removing video pre-training causes modest performance changes across benchmarks. This suggests that the benefits of video data for instructed temporal grounding tasks primarily stem from effective visual context length, yet this impact is relatively minor compared to vision-language alignment. This observation is further validated if we eliminate both image and video pre-training data, starting from a text-only large language model, where performance drops dramatically, with accuracy decreasing from 75.0% to 69.1% on MLVU and from 61.9% to 58.6% on LongVideoBench. This substantial degradation underscores that robust vision-language alignment is crucial to effective VideoITG training.

5.4. Visualization

In Fig. 5, we compare uniform sampling and VideoITG sampling of 8 frames from the VideoMME [15] Benchmark. In the first case, VideoITG captures both brushing teeth and spraying perfume actions, enabling correct temporal ordering, while uniform sampling misses key cues. In the second case, VideoITG accurately captures rapid consecutive movements at the end, whereas uniform sampling fails to do so, leading to incomplete video understanding.

6. Conclusion

In this paper, we presented VideoITG, a novel framework for instruction-aligned frame selection in Video-LLMs. The key to our approach was the *VidThinker* pipeline, which mimics human annotation by generating detailed, instruction-guided clip descriptions, retrieving relevant segments, and performing fine-grained frame selection. Using this pipeline, we constructed the VideoITG-40K dataset with 40K videos and 500K temporal grounding annotations. Based on this resource, we developed plug-and-play VideoITG models that leverage visual-language alignment and reasoning to handle diverse temporal grounding tasks. Experiments showed that VideoITG consistently improves Video-LLMs' performance across multiple video understanding benchmarks, highlighting its effectiveness and potential for advancing instruction-driven video understanding.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv:2307.15818*, 2023.

[4] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jeng-Neng Hwang, Saining Xie, and Christopher D Manning. AuroraCap: Efficient, performant video detailed captioning and a new benchmark. 2025.

[5] Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. CG-Bench: Clue-grounded question answering benchmark for long video understanding. *arXiv:2412.12075*, 2024.

[6] Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Tuomas Rintamaki, et al. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. *arXiv preprint arXiv:2504.15271*, 2025.

[7] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *CVPR*, 2024.

[8] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. ShareGPT4Video: Improving video understanding and generation with better captions. *arXiv:2406.04325*, 2024.

[9] Qirui Chen, Shangzhe Di, and Weidi Xie. Grounded multi-hop videoqa in long-form egocentric videos. In *AAAI*, 2025.

[10] Yang Chen, Sheng Guo, and Limin Wang. A large-scale study on video action dataset condensation. *arXiv:2412.21197*, 2024.

[11] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv:2406.07476*, 2024.

[12] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024.

[13] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

[14] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *CVPR*, 2024.

[15] Chaoyou Fu, Yuhan Dai, Yondong 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. *arXiv:2405.21075*, 2024.

[16] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Shaoqi Dong, Xiong Wang, Di Yin, Long Ma, et al. VITA: Towards open-source interactive omni multimodal llm. *arXiv:2408.05211*, 2024.

[17] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *ICCV*, 2017.

[18] Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In *CVPR*, 2025.

[19] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. LITA: Language instructed temporal-localization assistant. In *ECCV*, 2024.

[20] De-An Huang, Subhashree Radhakrishnan, Zhiding Yu, and Jan Kautz. FRAG: Frame selection augmented generation for long video and long document understanding. *arXiv:2504.17447*, 2025.

[21] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video Recap: Recursive captioning of hour-long videos. In *CVPR*, 2024.

[22] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-LaVIT: Unified video-language pre-training with decoupled visual-motional tokenization. *arXiv:2402.03161*, 2024.

[23] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv:2406.09246*, 2024.

[24] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, 2021.

[25] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv:2408.03326*, 2024.

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[27] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024.

[28] Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *ECCV*, 2024.

[29] Zeqian Li, Qirui Chen, Tengda Han, Ya Zhang, Yanfeng Wang, and Weidi Xie. Multi-sentence grounding for long-term instructional video. In *ECCV*, 2024.

[30] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. In *CVPR*, 2024.

[31] Jihao Liu, Zhiding Yu, Shiyi Lan, Shihao Wang, Rongyao Fang, Jan Kautz, Hongsheng Li, and Jose M Alvare. StreamChat: Chatting with streaming video. *arXiv:2412.08646*, 2024.

[32] Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding, 2025.

[33] Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. *arXiv preprint arXiv:2505.14454*, 2025.

[34] Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. E.T. Bench: Towards open-ended event-level video-language understanding. In *NeurIPS*, 2024.

[35] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: On-demand spatial-temporal understanding at arbitrary resolution. In *ICLR*, 2025.

[36] Yongdong Luo, Wang Chen, Xiawu Zheng, Weizhong Huang, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Jiebo Luo, et al. Quota: Query-oriented token assignment via cot query decouple for long video comprehension. *arXiv preprint arXiv:2503.08689*, 2025.

[37] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *ACL*, 2024.

[38] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2024.

[39] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12309–12318, 2022.

[40] Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. QuerYD: A video dataset with high-quality text and audio narrations. In *ICASSP*, 2021.

[41] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[42] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception Test: A diagnostic benchmark for multimodal video models. In *NeurIPS*, 2023.

[43] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *ICML*, 2024.

[44] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*, 2024.

[45] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. *arXiv:2410.17434*, 2024.

[46] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-XL: Extra-long vision language model for hour-scale video understanding. In *CVPR*, 2025.

[47] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. MovieChat: From dense token to sparse memory for long video understanding. In *CVPR*, 2024.

[48] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29118–29128, 2025.

[49] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

[50] Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-VideoLLM: Sharpening fine-grained temporal grounding in video large language models. *arXiv:2410.03290*, 2024.

[51] Junke Wang, Xitong Yang, Hengduo Li, Li Liu, Zuxuan Wu, and Yu-Gang Jiang. Efficient video transformers with spatial-temporal token selection. In *European Conference on Computer Vision*, pages 69–86. Springer, 2022.

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

[53] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin

Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*, 2024.

[54] Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. VideoLLaMB: Long-context video understanding with recurrent memory bridges. *arXiv:2409.01071*, 2024.

[55] Hongchen Wei and Zhenzhong Chen. Visual context window extension: A new perspective for long video understanding. *arXiv:2409.20018*, 2024.

[56] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

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

[58] 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:2407.15841*, 2024.

[59] Linli Yao, Haoning Wu, Kun Ouyang, Yuanxing Zhang, Caiming Xiong, Bei Chen, Xu Sun, and Junnan Li. Generative frame sampler for long video understanding. *arXiv preprint arXiv:2503.09146*, 2025.

[60] Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mPLUG-Owl3: Towards long image-sequence understanding in multi-modal large language models. In *ICLR*, 2024.

[61] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *NeurIPS*, 2023.

[62] Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. Frame-voyager: Learning to query frames for video large language models, 2025.

[63] 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. In *ICLR*, 2025.

[64] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *CVPR*, 2023.

[65] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[66] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-VStream: Memory-based real-time understanding for long video streams. *arXiv:2406.08085*, 2024.

[67] 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:2407.12772*, 2024.

[68] Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. InternLM-XComposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv:2407.03320*, 2024.

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

[70] Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. *arXiv preprint arXiv:2506.22139*, 2025.

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

[72] 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:2406.04264*, 2024.

[73] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *CVPR*, 2024.

[74] 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. *arXiv:2412.10360*, 2024.

Appendix

A. The Use of Large Language Models (LLMs)

Algorithm 1 Keyframe Extraction via Bidirectional CLIP Similarity

Require: Video frame sequence `frames`, similarity thresholds t_1 (scene change) and t_2 (diversity)
Ensure: Selected keyframe indices `sel`

Initialize `sel` with the first frame index: `sel` $\leftarrow \{0\}$
Extract CLIP feature for the first frame: `prev` $\leftarrow$ clip(`frames`[0])
for each frame c in `frames[1:]` with index i **do**
  `curr` $\leftarrow$ clip(c)
  $s \leftarrow$ sim(`curr`, `prev`)
  if $s < t_1$ **then**
    for each future frame f in `frames`$[i+1:]$ **do**
      `fut` $\leftarrow$ clip(f)
      if sim(`curr`, `fut`) $< t_2$ **then**
        Add index i to `sel`
        `prev` $\leftarrow$ `curr`
        break
      end if
    end for
  end if
end for
if sim(clip(`frames`[-1]), `prev`) $< t_1$ **then**
  Add last frame index to `sel`
end if
return `sel`

In this work, Large Language Models (LLMs) were employed in four main ways: (i) to aid and polish the writing for clarity and style; and (ii) to provide coding assistance, including code generation, debugging, and optimization suggestions.

Specifically, LLMs were utilized to improve the clarity, coherence, and readability of the manuscript, with particular attention given to the **Related Work**, **Method**, and **Experiments** sections. In these parts, the initial drafts were carefully reviewed and refined using LLM-powered suggestions for sentence structure, terminology, and logical flow. This process ensured that the technical content was presented in a precise and accessible manner, while maintaining consistency in academic tone and style throughout the paper.

All outputs generated by LLMs were critically reviewed, verified, and further refined by the authors. The core scientific ideas, methodology, and contributions remain entirely the authors' own. The use of LLMs was strictly limited to language enhancement and coding support, without influencing the originality or integrity of the research.

B. Inference time

In Table 5, we evaluated the speed of our model on a single NVIDIA A100 GPU. We employed LLaVA-Video-7B [71] as our answering LLM, implemented a 32-frame sampling strategy from 512 input frames in total, and generated 27 text tokens. Additionally, we leveraged KV Cache and Flash Attention [12, 13] to enhance inference efficiency.

Our detailed analysis of computational costs reveals that processing each video sample requires a total of 6.42 seconds, with the Vision Encoder (2.92 seconds) and LLM (2.89 seconds) dominating the time consumption. These two components collectively consume 90% of the total processing time, indicating the direction for future system optimization. In contrast, our VideoITG module demonstrates remarkable efficiency, requiring only 0.61 seconds to scan 512 frames—a speed that surpasses human visual recognition and thinking capabilities.

Table 5. Computation cost of the model.

Vision Encoder	VideoITG	LLM	Overall
2.92s	0.61s	2.89s	6.42s

C. Dataset details

C.1. Prompt template

Our Question-guided Clip Retrieval process utilizes a carefully designed prompt template (shown in Table 8) that instructs the LLM to analyze chronologically ordered clip-level descriptions and identify the minimal set of clips necessary to answer a given question. The prompt template consists of three main components:

- **Task Description**: Defines the LLM's role as an expert in analyzing video clip descriptions and establishes the goal of selecting clips that cover both question and answer content.
- **Guidelines**: Provides detailed instructions for clip selection, including handling time-related expressions, determining if a single or multiple clips are needed, addressing questions about object existence or movement, and avoiding unnecessary clips.
- **Output Format**: Specifies the required JSON structure for responses, ensuring consistent formatting with explanation and clip number fields.

This template enables the LLM to perform chain-of-thought reasoning when selecting relevant clips. The model analyzes keywords from questions, identifies temporal relationships (*e.g.*, "before," "after"), and provides explicit rationales for its selections. For cases where no relevant clips exist, the model returns "None" to reduce annotation noise.

Table 6. The performance (accuracy) of SOTA methods on video benchmarks. For InternVL2.5-8B results, we report the higher results in the technical report and lmms-eval. We sample 32 frames using VideoITG for our results.

Model	Open-Ended Q&A	Multi-Choice Q&A						
	ActNet-QA	EgoSchema	MLVU	NExT-QA	PerceptionTest	LongVideoBench	VideoMME	MVBench
	test	test	m-avg	mc	val	val	wo/w-subs	val
Open-source models								
VILA-40B [30]	58.0	58.0	-	67.9	54.0	-	60.1/61.1	-
PLLaVA-34B [57]	60.9	-	-	-	-	53.2	-	58.1
VideoLLaMA2-7B [11]	50.2	50.5	-	-	49.6	-	45.1/46.6	53.4
LongVA-7B [69]	50.0	-	56.3	68.3	-	-	52.6/54.3	-
LongVU-7B [69]	-	67.6	65.4	-	-	-	60.6/-	66.9
LLaVA-OV-7B [25]	56.6	60.1	64.7	79.4	57.1	56.5	58.2/61.5	56.7
mPLUG-Owl3-8B [60]	-	-	-	78.6	-	52.1	53.5/-	54.5
LLaVA-Video-7B [71]	56.5	57.3	70.8	83.2	67.9	58.2	63.3/69.7	58.6
Qwen2.5-VL-7B [53]	-	-	-	70.2	70.5	54.7	65.1/71.6	69.6
InternVL2.5-8B [68]	-	51.5	68.9	-	-	60.0	64.2/66.9	72.0
InternVL2.5-8B-ITG-32	57.4	51.6	75.0	79.5	64.9	61.9	67.3/69.6	72.2

Table 7. Dataset quality (IoU). We evaluate the performance in both multiple-choice (MC) and open-ended (OE) questions.

Method	Semantic-MC	Semantic & Motion-OE	Semantic-MC	Semantic & Motion-OE
Qwen2.5-VL-32B	0.31	0.36	0.27	0.37
GPT4o	0.24	0.30	0.26	0.27
Ours	**0.79**	**0.74**	**0.72**	**0.69**

We implement this process using GPT-4o-mini [41], which is sufficient for accurate clip selection while reducing annotation costs by over 10 times compared to larger models. The selected clips are then converted to event boundaries defined by timestamps based on frame indices for the final temporal grounding annotations.

C.2. Human-in-the-loop verification

Ensuring the quality of automatically annotated datasets is critical for the reliability and effectiveness of downstream video understanding models. In this work, we implement a comprehensive quality control protocol for the VideoITG-40K dataset.

Our pipeline begins with diverse sampling: we select a representative subset of the dataset, covering a wide range of instructions and video scenarios. For this subset, we conduct human verification, where expert annotators review the automatically generated annotations to assess their accuracy and relevance. This process allows us to identify and correct potential errors, and to further calibrate our annotation pipeline for improved consistency and quality.

As shown in Table 7, we compare our pipeline with baselines where advanced models such as Qwen2.5VL and GPT-4o are directly prompted to answer the temporal boundaries of relevant events. These direct approaches result in significantly lower performance, highlighting the advantage of our multi-step, instruction-guided annotation strategy.

C.3. Frame Sampling Algorithm

The algorithm in 1 is designed for semantic-only keyframe selection, aiming to extract a diverse set of frames that comprehensively capture the semantic content of a video—such as people, scenes, or objects. By leveraging CLIP features, the algorithm compares each frame to previously selected keyframes using a bidirectional similarity measure. Frames are selected when their semantic features differ significantly

Table 8. Prompt Template: An expert system for temporal localization in video segments. The system analyzes video segment descriptions to determine the minimal and necessary combination of segments required to answer questions.

Task:
You are an expert in analyzing video clip descriptions. Your task is to select which clip or combination of clips is necessary to answer the given question, ensuring the selected clips effectively cover the content of both the question and the answer.

Guidelines:

- Carefully read the descriptions to determine which clip(s) provide relevant content for the question and the answer.
- Clip descriptions are in chronological order. Use clip number to locate clips based on time-related expressions (e.g., "at the beginning of the video" suggests a smaller clip number, while "at the end of the video" suggests a larger one).
- First, determine if one clip can answer the question or if multiple clips are needed. Then, return a list containing the selected clip(s) and an explanation.
- If the question asks about the existence/movement of an object or event. The object/action/movement may not exist, meaning you can't find the answer in the description, but the question might still provide some clues. You need to find the sentence closest to those clues.
- If asked about the whole video description or overall atmosphere, you should return all clip numbers.
- If multiple clips provide similar descriptions of the content and any of them can be used to answer the question, return all corresponding clips.
- If there are no clues in all descriptions and cannot answer the question, return "None.".
- **Important**: Avoid including unnecessary clips.

Output Format:
1. Your output should be formed in a JSON file.
2. Only return the Python dictionary string.
For example:

```
{"explanation": "...", "clip_num": "One clip: [Clip-2]"}
{"explanation": "...", "clip_num": "Multiple clips: [Clip-1, Clip-7, Clip-8]"}
{"explanation": "...", "clip_num": "None."}
```

Table 9. Prompt template for identifying motion-related questions in video QA tasks. The template instructs the system to analyze each question-answer pair and determine whether the question pertains to absolute or relative speed, responding with "Yes" or "No" accordingly. Example cases are provided for clarification.

Task:
Analyze the given QA pair to determine if the question is related to speed. Specifically, check if it involves either absolute speed (the speed of a specific object) or relative speed (comparing the speed of different objects). Provide an output of "Yes" if the question pertains to speed, and "No" otherwise.
Important: Respond with "Yes" or "No" only.

Example:
Question 1: Which is faster, the white car or the bicycle? Options: A. The bicycle. B. The white car. C. Both are at the same speed. D. None of the above.
Answer 1: B. The white car.
Output: Yes.
Question 2: What color is the cat ?Options: A. black B. white C. orange D. gray
Answer 2: C. orange
Output: No.

from both the last keyframe (scene change threshold) and from future frames (diversity threshold), ensuring that each chosen frame represents distinct semantic information. This process produces a set of keyframes with maximal semantic coverage and minimal redundancy, aligning with the goal of representing all major semantic aspects of the video.

D. Visualization

In Fig. 6 and Fig. 7, we present two sets of results comparing sampling results of VideoITG with uniform sampling.

Table 10. Prompt template for identifying semantic-related questions in video QA tasks. The template instructs the system to analyze each question-answer pair and determine whether the question pertains to absolute or relative speed, responding with “Yes” or “No” accordingly. Example cases are provided for clarification.

Task:

Analyze the given QA pair to determine if the question inquires about the existence of an object or action. If it does, and the answer is ”No” (indicating non-existence), output ”Yes.” If the question is not about existence, or the answer is ”Yes” (indicating existence), output ”No.”
Important: Respond with ”Yes” or ”No” only.

Example:

Question 1: After going through the bag, does the person meticulously clean the area around the sink?
Answer 1: No, the person does not clean the area around the sink after going through the bag. The video primarily focuses on the action of the person with the bag and items, not on cleaning activities.
Output: Yes.
Question 2: Is there a cat sitting on the windowsill in the video?
Answer 2: Yes, there is a cat sitting on the windowsill throughout the video.
Output: No.

Fig. 6 demonstrates a temporal reasoning problem, where our model accurately identifies the “workout” mentioned in the question and successfully locates the subsequent actions in the video, leading to the correct answer selection. In contrast, the uniform sampling strategy failed to capture these crucial frames. Fig. 7 illustrates a non-existence question scenario where our model effectively identifies all IMAX movies present in the given options, enabling it to successfully filter out and determine the correct answer.

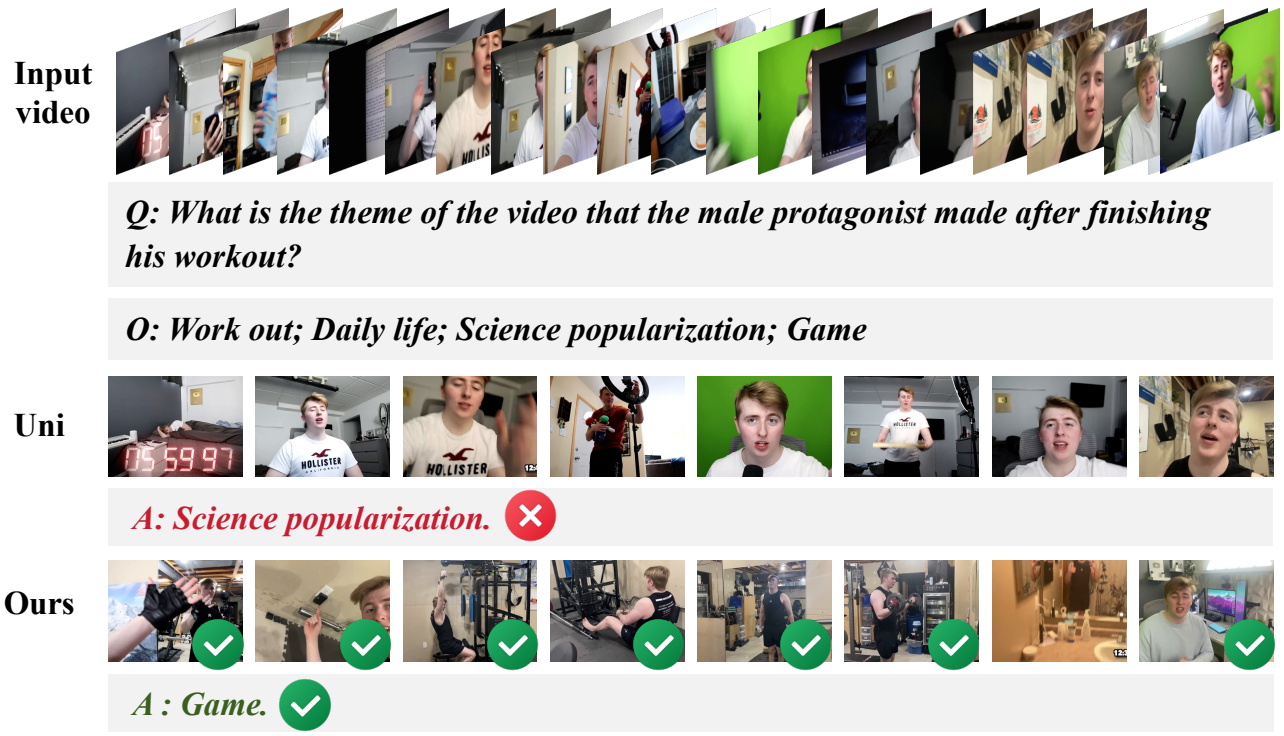

Figure 6. Example-1 shows how different sampling strategies impact video understanding. We mark the identified key frames that directly answer the question with green check-marks.

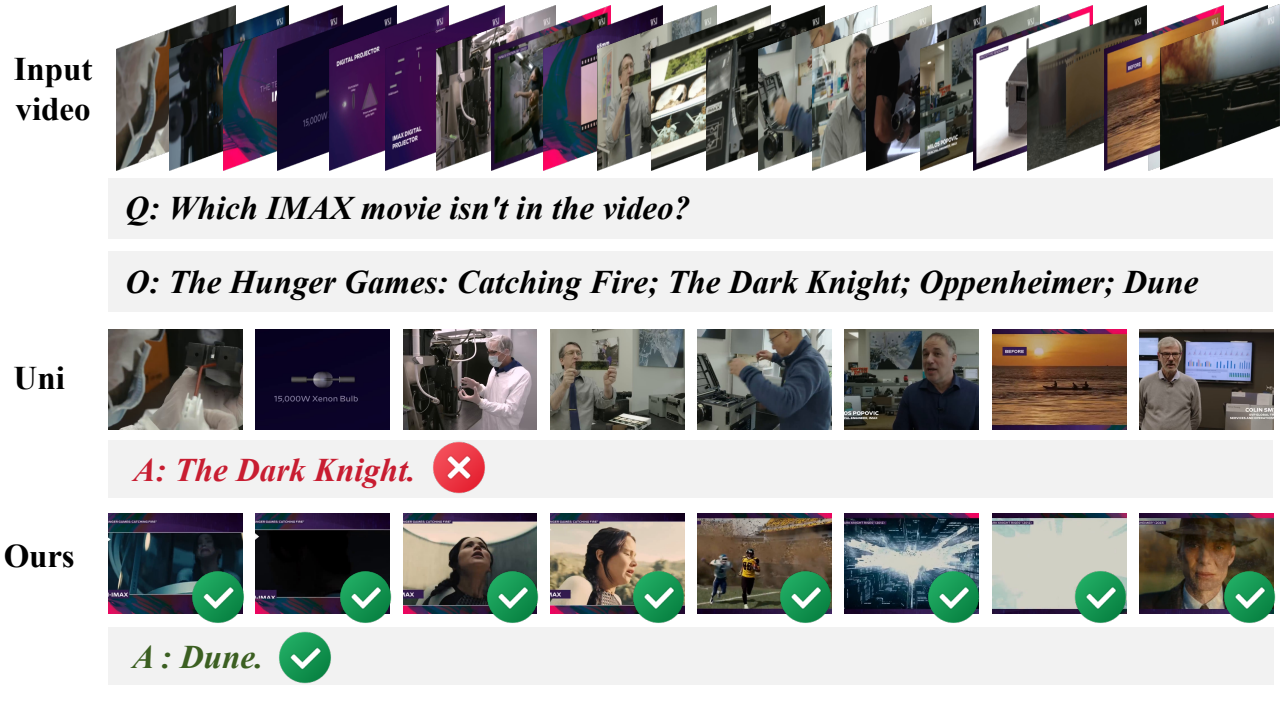

Figure 7. Example-2 shows how different sampling strategies impact video understanding. We mark the identified key frames that directly answer the question with green check-marks.