AdaAlloc: Adaptive Visual Token Allocation for Long-Video Question Answering

Joungbin An **Kristen Grauman**
The University of Texas at Austin

Abstract

Long-video question answering forces multimodal large language models to reason under strict visual-token budgets, creating a tradeoff between broad temporal coverage and fine-grained spatial detail. Uniform sampling preserves coverage but often obscures decisive local cues, while caption-based or frame-similarity retrieval operates over lossy proxies that can miss motion, state changes, and before–after relations. We recast long-video reasoning as a *visual-token allocation* problem: given a fixed budget, decide how much to spend on global context versus high-resolution local evidence, and where in the video to draw that local evidence from. We introduce AdaAlloc, a training-free inference framework that adaptively allocates visual tokens between a global video overview and localized visual detail. AdaAlloc first plans a global/local allocation policy, then uses temporal grounding to locate candidate evidence segments and refines them into a compact temporal memory. The selected policy controls both frame sampling density and resolution, combining sparse, lower-resolution global frames with denser, higher-resolution local frames drawn from the refined memory. The final answerer reasons over the allocated global context and refined local evidence. Across LVBench, MLVU, VideoMME, and LongVideoBench, AdaAlloc improves over budget-matched uniform sampling by up to 10.5 percentage points and outperforms the compared frame-selection and agentic methods on their reported benchmarks.[1]

1 Introduction

Long videos pose a challenge for multimodal large language models (MLLMs) operating under a fixed visual-token budget, as highlighted by recent long-video question answering benchmarks and MLLM studies [37, 54, 78, 12, 60]. Answering a question often requires tracking a narrative over many minutes, yet the decisive clue may be a number on a screen, a brief hand motion, or a subtle state change. A sparse sweep of the whole video preserves temporal coverage but blurs such details. A dense look at a few moments preserves detail, but helps only if those moments are the ones needed to correctly answer the question. Effective long-video understanding therefore requires deciding not just which frames to show, but *how many tokens to spend where*.

Existing work treats this primarily as a selection or compression problem. Uniform sampling preserves coverage at the cost of resolution, while query-aware keyframe methods improve relevance but remain constrained by a frame budget [51, 66, 19, 31, 79, 8]. Retrieval and agentic methods locate relevant clips, captions, or memories [71, 56, 58, 3, 33, 47, 80], yet many rely on caption-based indexing or frame-wise similarity, which can underrepresent motion and fine-grained visual cues. Recent advances in adaptive compression, query-type routing, multimodal keyframe selection, and temporal grounding improve which evidence reaches the model [55, 77, 24, 53, 72, 45], but they leave open how to distribute representation capacity across both time and resolution.

[1] Project webpage: https://vision.cs.utexas.edu/projects/adaalloc.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

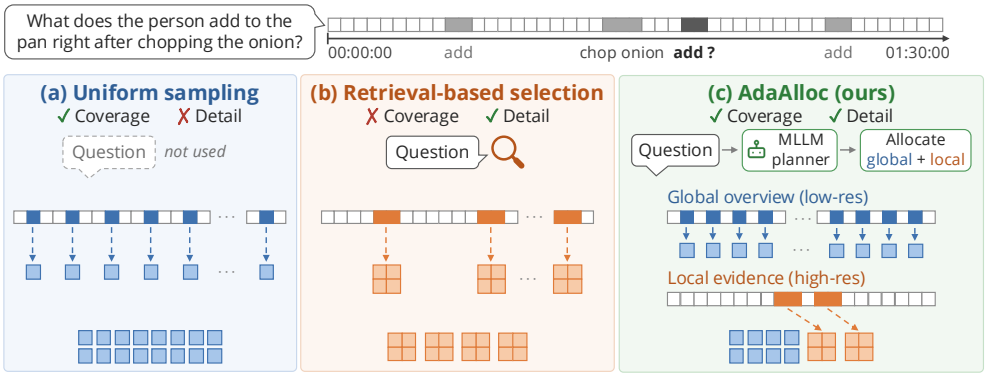

Figure 1: **Visual token allocation for long-video reasoning.** Uniform sampling preserves global coverage but loses fine details, while retrieval-based selection focuses on a few clips but can lose context. AdaAlloc allocates a fixed token budget between low-resolution global coverage and high-resolution local evidence for both context and fine-grained verification.

We recast long-video reasoning as a *visual-token allocation* problem: given a fixed budget, how should tokens be divided between low-resolution global coverage and high-resolution local inspection? This is distinct from frame selection: the objective is to allocate fidelity, not just to choose frames. Intuitively, some questions are global-dominant, such as summarization, event ordering, or identifying the overall activity; these often benefit more from broad temporal coverage than from high-resolution detail at every frame, and can therefore be represented by lower-resolution, lower-token-per-frame views. Other questions are local-dominant, such as reading text, counting objects, or distinguishing a brief hand action, where the answer depends on fine-grained visual evidence and would benefit from denser tokens at those time points. Many questions require both: global context to situate the episode and local evidence to resolve the answer.

This problem formulation yields two coupled decisions. First, how should the budget be split between global and local? Second, conditioned on a local budget, where should those tokens be placed? The second decision is critical. Spending local tokens on top-ranked frames can fail when retrieval is noisy, when similar events repeat, or when the query requires a before–after comparison rather than a single salient moment. Local allocation must therefore identify the right temporal evidence—not just salient frames, but candidate intervals that contain the events, context, and transitions needed to answer the question.

We introduce AdaAlloc, a training-free framework for budget-aware long-video reasoning. Starting from a sparse visual overview of the full video, AdaAlloc runs a structured evidence-planning step to identify the question's evidence needs, select one of five global/local allocation policies—global-only, global-heavy, balanced, local-heavy, or local-only—and determine whether targeted temporal search is needed. When local evidence is required, it generates evidence-seeking queries and invokes a lightweight video temporal grounding model (we use UniversalVTG [2]) to propose candidate evidence regions. These proposals, together with timestamps inferred from the question or overview, form a temporal memory. Because grounding outputs can be noisy, redundant, or fragmented, AdaAlloc further refines this memory with the MLLM, filtering and consolidating candidates conditioned on the question and evidence plan. The final visual input follows the selected policy: a global stream of sparse frames over the full video, and a local stream of denser, higher-resolution frames drawn from the refined memory. In this way, AdaAlloc decides both how broadly to look and where to look closely.

This view subsumes two common extremes: uniform sampling [36, 74] is a global-only policy that preserves coverage but underspecifies detail, while retrieval-only [19, 49] is a local-only policy that concentrates detail but risks losing context. See Figure 1. AdaAlloc makes this trade-off question-conditioned. Experiments on LVBench [54], MLVU [78], VideoMME [12], and LongVideoBench [60] show that, under the same token budget and MLLM backbone, AdaAlloc improves over uniform sampling by up to 10.5 percentage points, with the largest gains on the longest-video benchmarks. Improvements on both Qwen3-VL-8B and VideoLLaMA3-7B support its use as a plug-and-play, training-free agentic framework for improving existing MLLMs' long-video understanding.

Our contributions are:

- We formulate long-video understanding as visual-token allocation under a fixed budget, making explicit the tradeoff between temporal coverage and spatial detail.
- We introduce AdaAlloc, a training-free agentic framework that jointly determines how to split a fixed visual-token budget between global coverage and local detail, and where to spend the local share, using overview-conditioned planning, temporal grounding, and MLLM-based memory refinement.

2 Related Work

Long-video MLLMs under token constraints. Recent long-video benchmarks [37, 54, 78, 12, 60] expose a shared constraint: evidence is dispersed over minutes or hours, but MLLMs can process only a limited number of visual tokens at adequate resolution. Existing methods typically address this constraint from one of two directions. One line preserves broad temporal coverage by extending context windows or compressing visual tokens [74, 26, 46, 48, 55, 77]. This helps maintain a global view of the video, but can lose fine-grained evidence when many frames must share a fixed token budget. Another line selects a subset of frames or clips to improve relevance under a strict budget [66, 51, 19, 31, 79, 67, 50, 17, 65, 8, 42]. These methods can preserve local detail, but their success depends on the selected evidence being both correct and sufficient, and they may discard the global context needed for ordering, repeated events, or disambiguation. Query-adaptive variants further show that the sampling strategy should depend on the question [24, 53], and related ideas have been explored for allocating text reasoning tokens in LLMs [28]. AdaAlloc builds on this insight but changes the decision: rather than choosing between global coverage and local detail, it allocates a fixed visual-token budget between them, deciding when to see broadly, when to inspect locally, and when both are needed.

Evidence seeking for long-video reasoning. A second line of work treats long-video reasoning as evidence seeking: first identify relevant video regions, then answer from the selected evidence. Caption-based systems convert frames or clips into textual memories for retrieval [71, 22, 35], while agentic and retrieval-augmented methods use planning, tools, captions, entities, or graph-structured memories to decide what evidence to inspect [38, 56, 10, 58, 3, 33, 44, 21, 47, 80, 59]. SpotEM [42] learns to select promising clips using low-cost semantic features, concentrating expensive feature extraction on query-relevant regions. Recent agents learn this search behavior with supervised fine-tuning and/or reinforcement learning: VSeek [16] learns to issue natural-language search queries and reason over the retrieved clips using verifiable rewards, while LOVE-R1 [13] and Video-o3 [69] start from a low-resolution view of the whole video and, over multiple turns, inspect selected clips at higher resolution. AdaAlloc is an entirely training-free framework that jointly allocates a fixed visual-token budget between global temporal coverage and local high-resolution evidence. Unlike LOVE-R1 and Video-o3, which add high-resolution clips to a fixed global view, AdaAlloc spends less on global context whenever it spends more on local detail. Overview-conditioned planning determines this budget split, while temporal grounding and MLLM-based memory refinement determine where the local share is spent, producing a single budget-matched visual input for final answering.

Temporal grounding as a mechanism, not a task. Video temporal grounding (VTG) localizes language queries in untrimmed video [14, 23, 30, 39, 1, 2]. Recent time-aware MLLMs aim to improve the model's own timestamp awareness, through temporal instruction tuning, visual timelines, reinforcement learning, or zoom-in verification [43, 18, 41, 40, 52, 72, 73, 57, 6, 25, 11, 45, 27]. Instead, AdaAlloc assigns temporal localization a different role: rather than making timestamp prediction the reasoning objective, it uses VTG as an inference-time tool for visual-token allocation. It decides when localized evidence is needed, grounds evidence-seeking queries with UniversalVTG [2], and refines the resulting intervals into a temporal memory before spending high-resolution tokens. Unlike methods that primarily decide *which* frames or segments to process [24, 45], AdaAlloc also decides how much global context to preserve and where local detail is worth the budget.

3 Problem Formulation

We study long-video multiple-choice question answering under a fixed visual-token budget. Let V be an untrimmed video of duration T, q a natural-language question with answer choices $\mathcal{A} = \{a_1, \ldots, a_m\}$, and B the maximum number of visual tokens available to the MLLM. The goal is to predict $\hat{a} \in \mathcal{A}$ from a visual representation $E(V, q; B)$ whose total visual-token count does not exceed B. This count depends on both the number of sampled frames and their encoded spatial resolution. Encoding the same frames at higher resolution requires more tokens but can preserve finer visual details. A fixed token budget therefore couples temporal coverage and spatial detail, requiring decisions about both how many frames to sample and at what resolution to encode them.

Throughout, B denotes the visual-token budget of a single MLLM call. Every MLLM call in AdaAlloc, including planning and memory refinement (Section 4), is capped at the same B, and our controlled comparisons match B for the final answering call that receives $E(V, q; B)$. In our experiments, we instantiate B with 16K and 32K visual-token regimes.[2]

We decompose the evidence into a global overview stream E_g and a local detail stream E_l. The global stream samples the full video to preserve temporal context, such as scene changes, event order, repeated episodes, and coarse narrative structure. The local stream allocates denser or higher-resolution frames to selected temporal regions, preserving fine-grained cues such as text, counts, attributes, spatial relations, brief actions, and state changes. The framework chooses both a budget split $\pi = (\rho_g, \rho_l)$

$$\rho_g, \rho_l \in [0, 1], \qquad \rho_g + \rho_l = 1, \qquad B_g = \rho_g B, \quad B_l = \rho_l B,$$

and, when $B_l > 0$, a set of local intervals $\mathcal{R} = \{(s_j, e_j)\}_{j=1}^{K}$. The final MLLM input is

$$E(V, q; B) = \mathrm{concat}(E_g(V; B_g), E_l(V, \mathcal{R}; B_l)) .$$

This formulation separates two decisions that are often conflated: *how much* budget should remain global versus local, and *where* local budget should be placed. Uniform sampling is the global-only endpoint, preserving coverage but limiting detail. Retrieval-only pipelines approximate the local-only endpoint, concentrating budget on selected intervals but risking loss of context needed for temporal order, repeated events, or disambiguation.

4 AdaAlloc Approach

Building on the formulation in Section 3, AdaAlloc instantiates the evidence representation $E(V, q; B)$ with an adaptive three-stage pipeline, shown in Figure 2. **First**, overview-conditioned planning samples a sparse low-resolution overview of V and uses it to select a global/local allocation policy $\pi = (\rho_g, \rho_l)$ and generate evidence-seeking temporal grounding queries. These queries are used to retrieve candidate intervals from the original video. **Second**, temporal-memory refinement consolidates the retrieved intervals, explicit timestamps, and overview-derived temporal seeds into a compact refined memory $\mathcal{M}^*$. **Third**, budgeted hybrid answering constructs the final representation

$$E(V, q; B) = \mathrm{concat}(E_g(V; B_g), E_l(V, \mathcal{R}^*; B_l)) ,$$

where $B_g = \rho_g B$, $B_l = \rho_l B$, and $\mathcal{R}^*$ denotes the local intervals selected from $\mathcal{M}^*$. The model then answers from this hybrid input, using E_g for global temporal context and E_l for high-resolution inspection of the selected evidence regions.

4.1 Overview-conditioned evidence planning

AdaAlloc begins by sampling a sparse, low-resolution overview O of the full video V. It captures coarse information about the setting, actors, scene progression, repeated events, and approximate temporal anchors to guide evidence acquisition and global/local allocation. The final answerer then receives a question-specific combination of global context and local detail within B (Section 4.3).

[2] Including planning and refinement, AdaAlloc's end-to-end runtime is 2.3–2.5× that of single-pass uniform sampling, with an average accuracy gain of 4.8–5.3 percentage points across the four benchmarks. Appendix F reports total compute and runtime for all inference stages.

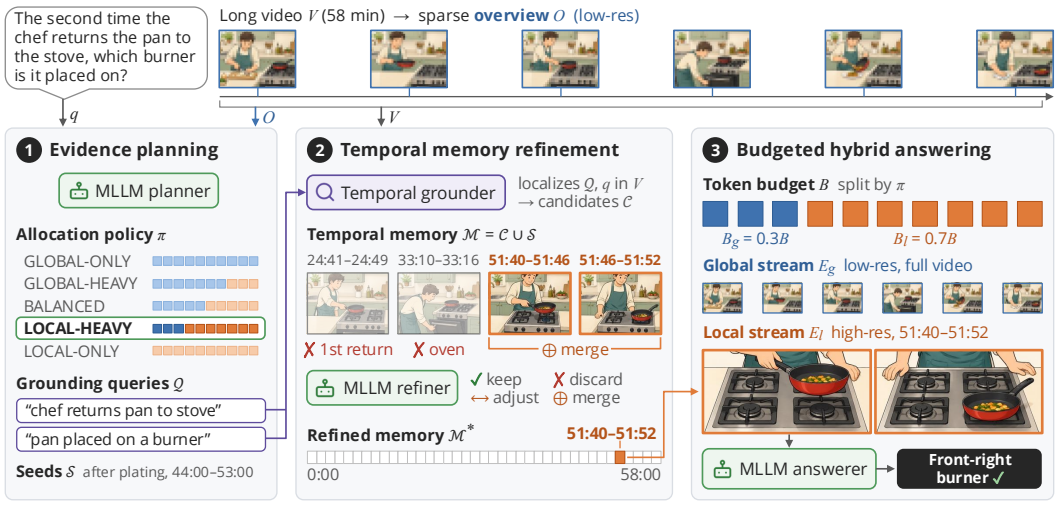

Figure 2: **AdaAlloc pipeline.** AdaAlloc splits a fixed visual-token budget B between low-res global context (blue) and high-res local evidence (orange). **(1)** From a sparse overview O, an MLLM planner picks a policy π and generates queries $\mathcal{Q}$. **(2)** A temporal grounder (UniversalVTG) localizes $\mathcal{Q}$; the MLLM refines the candidates into $\mathcal{M}^*$, discarding an off-topic clip and the *first* return to the stove. **(3)** The MLLM answers from a global stream E_g and a local stream E_l drawn from $\mathcal{M}^*$.

Planning and refinement respect the same per-call budget B. Conditioned on the overview, question, and answer choices, the MLLM planner produces

$$(\pi, \mathcal{Q}, \mathcal{S}) = f_{\text{plan}}(O, q, \mathcal{A}),$$

where π is the allocation policy, $\mathcal{Q}$ is a set of grounding queries, and $\mathcal{S}$ is a set of temporal memory seeds.

The allocation policy $\pi = (\rho_g, \rho_l)$ is selected from five discrete options:

$$\pi \in \left\{ \begin{array}{ll} \textsc{Global-only} & (1.0, 0.0) \\ \textsc{Global-heavy} & (0.7, 0.3) \\ \textsc{Balanced} & (0.5, 0.5) \\ \textsc{Local-heavy} & (0.3, 0.7) \\ \textsc{Local-only} & (0.0, 1.0) \end{array} \right\},$$

where each pair denotes the fraction of the visual-token budget assigned to global overview frames and local evidence clips, respectively. These five discrete policies simplify the planner's decision while allowing the allocation to adapt to each video–question pair. In practice, selecting among them matches or exceeds the average accuracy of a variant of our approach that directly predicts the global/local ratio (Appendix D).

The query set $\mathcal{Q}$ is generated when the planner determines that temporal search is needed. Each query describes visually grounded evidence relevant to q, such as an action, object interaction, state change, repeated event, or temporal anchor. In parallel, the planner may produce temporal seeds $\mathcal{S}$ directly from the question or overview, including explicit timestamps, intervals around overview-visible events, or coarse windows likely to contain relevant before/after context. The resulting $(\pi, \mathcal{Q}, \mathcal{S})$ defines the allocation intent and temporal search space passed to the memory refinement stage. Our framework builds in some tolerance to imperfect planning and seeds: grounding and memory refinement can correct inaccurate temporal seeds, while our overview-density analysis shows limited aggregate sensitivity to sparse planning input (Appendix D).

4.2 Temporal memory construction and refinement

Given the planner outputs $(\pi, \mathcal{Q}, \mathcal{S})$, AdaAlloc constructs a temporal memory when the policy assigns nonzero budget to local evidence. For each grounding query in $\mathcal{Q}$, together with the original question q as a complementary query, AdaAlloc invokes UniversalVTG [2] over the original video. UniversalVTG is a lightweight ∼60M-parameter VTG model that localizes open-vocabulary queries directly

Table 1: Summary of the long-form video QA benchmarks used in our evaluation.

Benchmark	#Video	#Question	Avg. (min.)	Range (min.)	Total Hours	Main Stressor
LVBench [54]	103	1,549	67.3	30.1–139.9	115.5	Hour-scale context
MLVU-Test [78]	349	502	17.8	3.1–136.8	103.4	Hard six-way QA
VideoMME [12]	900	2,700	17.0	0.18–59.7	255.3	Duration robustness
LongVideoBench-Val [60]	753	1,337	7.9	0.13–59.6	99.8	Long-context referring

in video time, producing ranked candidate intervals before costly MLLM inspection. We choose UniversalVTG for its strong grounding performance and efficiency, though alternative VTG models could be dropped in, and AdaAlloc with simple query–frame similarity retrieval also outperforms uniform sampling (Appendix D). We denote the resulting VTG candidates as $\mathcal{C}$.

AdaAlloc initializes the temporal memory by combining VTG candidates with planner-derived seeds:

$$\mathcal{M} = \mathcal{C} \cup \mathcal{S}.$$

Each entry $m_i \in \mathcal{M}$ stores a temporal span $[s_i, e_i]$, source provenance, the associated query or rationale, and the VTG retrieval score for grounded candidates or the planner's confidence for temporal seeds. The memory is constructed for recall rather than precision: VTG predictions, overview-derived seeds, and explicit timestamps may be shifted, redundant, fragmented, or irrelevant. AdaAlloc therefore treats $\mathcal{M}$ as a set of timestamped hypotheses, not as final evidence.

Before spending the local budget B_l, AdaAlloc refines the initialized memory with:

$$\mathcal{M}^* = f_{\text{refine}}(\mathcal{M}, O, q, \mathcal{A}).$$

For each candidate interval, the model observes sampled frames together with the question, answer choices, planner rationale, source provenance, associated query, and VTG score when available. It then edits the memory using four operations: *keep* useful evidence regions, *discard* off-topic or duplicate candidates, *adjust* shifted intervals, and *merge* adjacent or complementary regions that capture one event, repeated action, or before/after context.

The output $\mathcal{M}^*$ is a compact set of timestamped regions with provenance and sampling priority. This stage uses temporal grounding for candidate generation, not final evidence selection: UniversalVTG provides high-recall temporal hypotheses, which the model refines according to the planned evidence needs. The refined memory determines the local intervals $\mathcal{R}^*$ used in the final budgeted representation.

4.3 Budgeted hybrid answering

AdaAlloc assembles the final visual input from the selected allocation policy and refined memory $\mathcal{M}^*$. The global budget is spent on sparse, low-resolution frames spanning the full video, while the local budget is spent on denser, higher-resolution clips sampled from $\mathcal{M}^*$. When multiple memory regions survive refinement, the local budget is divided across the selected regions; if no reliable local memory remains, it falls back to additional overview coverage.

This produces a hybrid representation with a global stream for temporal context and detail streams for localized inspection. Global questions rely primarily on the overview, localized questions allocate more budget to grounding-guided evidence clips, and mixed-evidence questions preserve context while reserving local capacity for decisive cues. The final answerer reasons over this hybrid input, turning the refined memory into an explicit allocation of visual tokens across time and resolution. The full pipeline improves accuracy over uniform sampling across all four benchmarks under matched final-answering budgets, with its end-to-end accuracy–cost tradeoff quantified in Section 3 and Appendix F.

5 Experiments

Benchmarks. We evaluate AdaAlloc on four complementary long-form video question-answering benchmarks, summarized in Table 1. **LVBench** [54] targets extreme long-video comprehension, with hour-scale videos that stress global narrative understanding and sparse evidence retrieval. **MLVU-Test** [78] is the newer and harder official test split of MLVU: compared with MLVU-Dev, it expands multiple-choice questions from four to six options and adds new categories such as Sports QA and Tutorial QA, making chance performance lower and multi-task reasoning more challenging.

Table 2: **Controlled main results under matched visual-token budgets.** Following the protocol in Section 5, all methods use the same backbone, decoding setup, and visual-token budget within each block. We report accuracy (%) on four long-form video QA benchmarks in the video-only setting. Numbers in parentheses denote absolute gains over uniform sampling under the same budget.

Budget	Method	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val
16K tokens	Uniform sampling	42.7	55.4	67.1	62.8
	AdaAlloc	**51.7** (+9.0)	**61.2** (+5.8)	**68.4** (+1.3)	**66.0** (+3.2)
32K tokens	Uniform sampling	46.1	61.4	69.5	63.9
	AdaAlloc	**56.6** (+10.5)	**67.1** (+5.7)	**70.7** (+1.2)	**67.5** (+3.6)

VideoMME [12] provides a balanced evaluation across short, medium, and long videos, testing whether a method remains robust across duration regimes rather than only on extreme long videos. **LongVideoBench** [60] focuses on long-context video-language reasoning with referring questions that require precise localization of relevant visual evidence; we report results on the validation split. Together, these benchmarks test dense local evidence retrieval, global temporal comprehension, temporal ordering, counting, OCR/detail perception, and long-context referring reasoning across diverse video lengths and domains. All benchmarks are evaluated in a video-only setting without subtitles, so models must answer from visual evidence rather than transcript cues.

Implementation details. We implement AdaAlloc with Qwen3-VL-8B-Instruct [4] and use the same model as the final answerer for the main controlled baselines. Following the fixed-budget formulation in Section 3, our controlled comparisons use matched final-answering visual-token budgets. We report two budget regimes, 16K and 32K visual tokens, corresponding to uniform-sampling reference settings of 64 and 128 frames, respectively. These settings cover a practical range for current long-video MLLM inference with 7–8B-class models: the 64-frame setting represents a constrained regime where allocation is especially important, while the 128-frame setting gives uniform sampling substantially stronger temporal coverage while remaining feasible on typical 40GB-class GPUs in our setup.

The uniform baseline spends the entire budget on sparse global coverage: it samples the video at 1 FPS and uniformly subsamples up to the corresponding frame limit. Retrieval-based baselines spend the same budget on localized clips returned by temporal retrieval. AdaAlloc receives the same total budget, but distributes it between a global overview stream and local detail streams according to one of the five policies defined in Section 4.1: GLOBAL-ONLY, GLOBAL-HEAVY, BALANCED, LOCAL-HEAVY, and LOCAL-ONLY. When uniform full-video sampling yields sufficiently dense temporal coverage under the assigned budget, AdaAlloc answers directly. These routed examples are answered from uniformly sampled global frames with the standard multiple-choice QA prompt, bypassing VTG queries, temporal proposals, memory construction, and refinement. For VTG-guided candidate generation, we use UniversalVTG [2] and retain the top-5 temporal proposals per query before temporal-memory refinement. Frames in the overview and local streams are dynamically sampled and resized to satisfy the selected global/local visual-token split, using lower-resolution frames for broad overview coverage and higher-resolution frames for local inspection. Within each stream, the allocated budget and sampled frame count determine the available visual tokens per frame and hence the spatial resolution. Frames are resized accordingly, preserving aspect ratio with height and width aligned to multiples of 32 pixels. All MLLM calls use greedy decoding.

Main results. Table 2 compares methods under matched backbone, decoding setup, and visual-token budgets. Using Qwen3-VL-8B-Instruct as a fixed backbone isolates the effect of visual-token allocation. Uniform or fixed-rate frame sampling is a common inference strategy for VideoLLMs and a standard baseline for adaptive frame selection and efficient long-video understanding methods [36, 29, 74, 51, 67, 75, 31]. AdaAlloc consistently improves over this standard sampling baseline across all four benchmarks and both budget regimes, with the largest gains on LVBench [54] and MLVU-Test [78]. These benchmarks contain the longest videos in our evaluation, where answer-relevant evidence is often sparse and temporally localized. Under a fixed token budget, uniform sampling may miss temporally localized evidence or capture it at insufficient spatial resolution to resolve fine details. AdaAlloc addresses these limitations by retaining a global overview while using temporal grounding to allocate high-resolution inspection to likely evidence regions. The gains are smaller but consistent

Table 3: **Comparison with recent frame-selection methods under the same backbone.** All methods use Qwen3-VL-8B-Instruct. †Approximate budgets converted from the reported 128- and 256-frame settings at about 150 visual tokens per frame [24]; these exceed AdaAlloc's 16K and 32K budgets, making the budget favorable to the baselines. We report LongVideoBench and the medium/long subsets of VideoMME$_{\text{w/o sub}}$ to match DIG [24].

Method	Visual Tokens	LongVideo Bench-Val	VideoMME$_{\text{w/o sub}}$	
			Medium	Long
AKS [51]	≈19K†	53.0	54.4	51.1
DIG [24]	≈19K†	58.3	58.4	51.1
AdaAlloc	16K	**66.0**	**68.7**	**57.2**
AKS [51]	≈38K†	55.6	59.2	53.2
DIG [24]	≈38K†	61.2	61.6	53.8
AdaAlloc	32K	**67.5**	**70.6**	**62.2**

on VideoMME [12] (duration and sub-task breakdowns in Appendix B) and LongVideoBench-Val [60], suggesting that adaptive allocation remains useful even when uniform sampling already provides stronger coverage. Overall, these results support our central claim: under a fixed visual-token budget, long-video QA benefits from allocating visual fidelity across time rather than spending the entire budget uniformly.

Comparison with same-backbone selection methods. Table 3 compares AdaAlloc with recent frame-selection methods under the same Qwen3-VL-8B-Instruct backbone. Because methods differ in visual preprocessing, equal frame counts do not imply equal visual-token budgets, so we compare at approximate token budgets. DIG [24] reports results at fixed frame counts, with each frame encoded into approximately 150 visual tokens. Exact 16K and 32K results are not reported for prior methods, so we use the nearest reported settings, 128 and 256 frames (≈19K and ≈38K tokens), which give AKS and DIG more visual tokens than AdaAlloc. Despite the smaller budget, AdaAlloc outperforms AKS [51] and DIG [24] at both operating points, with the largest gains on the VideoMME medium subset (+10.3 and +9.0 points over DIG). These results suggest that AdaAlloc's benefit is not merely from selecting a better subset of frames, but from allocating a fixed visual budget between global temporal context and high-resolution local inspection.

Comparison with existing systems. Table 4 situates AdaAlloc among representative long-video understanding systems. We focus on methods that report results on comparable benchmarks and provide a frame budget or sampling setting for similar-scale models. Because prior methods differ in backbone, training data, inference pipeline, visual preprocessing, and evaluation protocol, this table should be read as a cross-paper comparison rather than as a strictly controlled head-to-head comparison. AdaAlloc achieves competitive accuracy without additional model training at 16K and 32K visual tokens, roughly equivalent to 64 and 128 frames at 256 tokens per frame. Compared with training-free and inference-time selection methods, AdaAlloc performs favorably while using a unified allocation mechanism that preserves global overview context and selectively spends high-resolution tokens on temporally grounded local evidence.

Ablations. Table 5 isolates the main design choices in AdaAlloc under the 32K visual-token budget. Uniform sampling follows a common inference setup for base video MLLMs, in which the model answers directly from frames sampled uniformly across the video. Retrieval-only grounds the original question with UniversalVTG [2] and answers from frames sampled from the resulting temporal proposals. Fixed policy keeps the same global/local input structure as AdaAlloc but replaces question-conditioned allocation with a fixed BALANCED split. W/o temporal memory refinement removes the refinement stage from Section 4.2 and samples directly from all the candidate proposals. AdaAlloc performs best across all benchmarks, indicating that its gains come from combining global context, adaptive allocation, and refined local evidence selection. A complementary 16K ablation in Appendix C separates planning, the global/local hybrid input, and adaptive allocation: adding a fixed global stream to a local-only variant changes accuracy only slightly, whereas making the split question-conditioned improves all four benchmarks by 1.5–3.2 points.

Table 4: **Contextual comparison with representative long-video understanding systems.** Prior numbers are taken from the respective papers when available. We include methods that report results on comparable long-video benchmarks and specify a frame budget or sampling setting for similar-scale systems. Visual inputs are reported in their original units, since frame-to-token conversion depends on the backbone, resolution, and compression scheme. Because methods differ in backbone MLLM, training data, inference pipeline, visual preprocessing, and evaluation protocol, this table is intended as a cross-paper comparison rather than a strictly controlled head-to-head evaluation. Bold marks the best result excluding proprietary models.

Method	Size	Visual input	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val
Proprietary models						
GPT-4o [20]	–	384 frames	48.9	54.9	71.9	66.7
Gemini-1.5-Pro [15]	–	256 frames	33.1	53.8	75.0	64.0
Open-source long-video MLLMs / trained or adapted video models						
mPLUG-Owl3 [64]	7B	128 frames	43.5	–	59.3	52.1
NVILA [32]	8B	256 frames	–	–	64.2	57.7
ZoomV [76]	8B	80 frames	51.6	–	65.6	63.6
TimeMarker [7]	8B	128 frames	–	–	57.3	56.3
LongVILA [9]§	7B	256 frames	–	49.0	60.1	57.1
SF-LLaVA-1.5 [62]	7B	128 frames	–	–	63.9	62.5
LongVA [74]	7B	128 frames	–	–	52.6	–
MemVid [68]	7B	128 frames	44.4	58.1	63.7	–
Training-free / inference-time selection and retrieval methods						
BOLT [31]§	8B	32 frames	–	–	65.7	62.2
F2C [49]	7B	32 frames	–	–	65.6	60.8
VideoDetective [63]	8B	32 frames	43.2	56.3	–	60.2
QuoTA [34]	7B	64 frames	–	–	65.9	59.0
Video-RAG [33]	7B	64 frames	–	–	–	58.7
Q-Frame [75]§	7B	64 frames	–	–	64.5	59.9
FOCUS [79]	7B	64 frames	–	–	65.4	63.5
AdaAlloc	8B	16K tokens‡	51.7	61.2	68.4	66.0
AdaAlloc	8B	32K tokens‡	**56.6**	**67.1**	**70.7**	**67.5**

‡At approximately 256 visual tokens per frame, 16K and 32K tokens correspond to **64** and **128** frames, respectively. AdaAlloc adapts frame counts and resolutions within these budgets. Other methods are listed using their reported frame counts. §From other sources: BOLT (Qwen3-VL-8B) from its official GitHub; Q-Frame (LLaVA-Video-7B) from FOCUS [79]; LongVILA MLVU-Test from MemVid [68].

Table 5: **Ablation study under the 32K visual-token budget.** All variants use Qwen3-VL-8B-Instruct with the same final-answering visual-token budget. Uniform sampling answers from uniformly sampled frames. Retrieval-only answers from segments localized by UniversalVTG using the original question. Fixed policy uses a BALANCED global/local split. W/o temporal memory refinement removes the refinement stage (Section 4.2).

Method	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val
Uniform sampling	46.1	61.4	69.5	63.9
Retrieval-only	48.2	59.4	67.6	63.5
Fixed policy	55.0	60.6	69.0	64.7
w/o temporal memory refinement	55.9	66.5	70.2	67.2
AdaAlloc	**56.6**	**67.1**	**70.7**	**67.5**

Generalization, robustness, and cost. AdaAlloc's gains are not tied to a specific backbone or grounding model (Appendix D). Applied to VideoLLaMA3-7B [70] under the same 32K budget, it improves over uniform sampling on all four benchmarks (+2.25 points on average), and replacing UniversalVTG with query–frame similarity retrieval retains most of the gains (0.31 average points higher at 16K and 1.66 lower at 32K). Its selected policies also follow interpretable question-type patterns, with global-oriented policies more common for descriptive questions (what or where) than temporal questions (before or after) (Appendix E). The additional planning and refinement calls cost more than single-pass uniform sampling, but AdaAlloc still has lower latency than DIG [24] at both budgets (Appendix F).



Figure 3: **Qualitative example on LVBench.** The overview supplies global context, while the question's request to read a handwritten number motivates a LOCAL-HEAVY policy for detailed inspection. AdaAlloc retrieves multiple temporal candidates and refines them into a compact memory that preserves the decisive interval. AdaAlloc correctly answers **25**, while Uniform predicts **5**.

Qualitative analysis. Figure 3 illustrates how AdaAlloc allocates evidence on a fine-grained LVBench example. The question asks for the number highlighted by a rectangle in a pencil-written chart, requiring localized OCR/detail perception within a nearly hour-long video. Conditioned on the question and sparse overview, the planner selects a LOCAL-HEAVY policy to prioritize detailed inspection of the chart and generates queries targeting the highlighted number. VTG retrieves multiple candidate intervals, including unrelated distractors such as ship, map, and radar-screen regions, as well as a partial chart-like segment. Temporal-memory refinement filters these noisy proposals and keeps the decisive interval at 00:39:04–00:39:29, where the hand-drawn chart and highlighted number are visible. The final hybrid input retains low-resolution global context while spending high-resolution local tokens on the refined chart region, enabling AdaAlloc to answer correctly with "25," whereas uniform sampling predicts "5."

Limitations and future work. This work studies training-free visual-token allocation using five discrete global/local policies. This choice simplifies planning and performs comparably to direct ratio prediction on average (Appendix D), though the selected split need not be optimal for each video–question pair. Exploring how to learn or refine the allocation space is a natural next step. AdaAlloc also relies on a sparse overview and retrieved temporal candidates, so evidence missed by both stages may remain unavailable to memory refinement. Our overview-density analysis suggests limited aggregate sensitivity to sparse planning input (Table 10), while motivating further study of evidence recovery in individual failure cases. The additional planning and refinement calls introduce overhead relative to single-pass inference, making the accuracy–latency tradeoff another direction for improvement (Appendix F). Finally, our evaluation focuses on frame- and clip-level allocation with 7–8B backbones. Extending allocation to individual visual tokens and studying larger MLLMs would broaden this investigation.

6 Conclusion

We introduced AdaAlloc, a training-free agentic framework for long-video understanding that moves beyond frame selection to question-conditioned visual-token allocation. Through overview-conditioned planning, temporal grounding, and memory refinement, it dynamically divides a fixed budget between low-resolution global coverage and high-resolution local evidence, and determines where to spend the local share. Across four long-video QA benchmarks, AdaAlloc consistently improves over budget-matched uniform sampling at 16K and 32K tokens, demonstrating the value of allocating visual capacity across both time and resolution.

Acknowledgments and Disclosure of Funding

This research is supported in part by the UT Austin IFML NSF AI Institute and Lyda Hill Philanthropies. We thank Elias Stengel-Eskin and the UT Austin Vision Group for constructive feedback and helpful discussions.

References

[1] Joungbin An and Kristen Grauman. HieraMamba: Video temporal grounding via hierarchical anchor-Mamba pooling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16954–16965, 2026.

[2] Joungbin An, Agrim Jain, and Kristen Grauman. UniversalVTG: A universal and lightweight foundation model for video temporal grounding. *arXiv preprint arXiv:2604.08522*, 2026.

[3] Anurag Arnab, Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. Temporal chain of thought: Long-video understanding by thinking in frames. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:143018–143046, 2025.

[4] 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.

[5] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:60884–60937, 2025.

[6] Ruizhe Chen, Tianze Luo, Zhiting Fan, Heqing Zou, Zhaopeng Feng, Guiyang Xie, Hansheng Zhang, Zhuochen Wang, Zuozhu Liu, and Huaijian Zhang. Datasets and recipes for video temporal grounding via reinforcement learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP): Industry Track*, pages 983–992, 2025.

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

[8] Wang Chen, Yongdong Luo, Yuhui Zeng, Luojun Lin, Tianyu Xie, Fei Chao, Rongrong Ji, and Xiawu Zheng. Event-anchored frame selection for effective long-video understanding. *arXiv preprint arXiv:2603.00983*, 2026.

[9] Yukang Chen et al. LongVILA: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations (ICLR)*, 2025.

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

[11] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:99114–99137, 2025.

[12] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24118, 2025.

[13] Shenghao Fu, Qize Yang, Yuan-Ming Li, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. LOVE-R1: Advancing long video understanding with an adaptive zoom-in mechanism via multi-step reasoning. *arXiv preprint arXiv:2509.24786*, 2025.

[14] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5267–5275, 2017.

[15] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[16] Harsh Goel, SP Sharan, Sahil Shah, Minkyu Choi, Joungbin An, Kristen Grauman, and Sandeep P Chinchali. Incentivizing vision language models to search for long video question answering. In *European Conference on Computer Vision (ECCV)*, pages 539–558. Springer, 2026.

[17] Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:124389–124422, 2025.

[18] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower LLM to grasp video moments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

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

[21] Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. VideoRAG: Retrieval-augmented generation over video corpus. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21278–21298, 2025.

[22] Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5627–5646, 2025.

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

[24] Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11369–11380, 2026.

[25] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[26] Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision (ECCV)*, pages 323–340. Springer, 2024.

[27] Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:64426–64455, 2025.

[28] Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. SelfBudgeter: Adaptive token allocation for efficient LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 21135–21156, 2026.

[29] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5971–5984, 2024.

[30] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. UniVTG: Towards unified video-language temporal grounding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2794–2804, 2023.

[31] Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. BOLT: Boost large vision-language model without training for long-form video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3318–3327, 2025.

[32] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. NVILA: Efficient frontier visual language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4122–4134, 2025.

[33] Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-RAG: Visually-aligned retrieval-augmented long video comprehension. *Advances in Neural Information Processing Systems (NeurIPS)*, 38: 168008–168033, 2025.

[34] Yongdong Luo, Wang Chen, Weizhong Huang, Shukang Yin, Haojia Lin, Jinfa Huang, Chaoyou Fu, Jiayi Ji, Xiawu Zheng, and Jiebo Luo. QuoTA: Query-oriented token assignment via CoT query decouple for long video comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pages 24160–24168, 2026.

[35] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. DrVideo: Document retrieval based long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18936–18946, 2025.

[36] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, 2024.

[37] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:46212–46244, 2023.

[38] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. MoReVQA: Exploring modular reasoning models for video question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13235–13245, 2024.

[39] Fangzhou Mu, Sicheng Mo, and Yin Li. SnAG: Scalable and accurate video grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18930–18940, 2024.

[40] Ming Nie, Dan Ding, Chunwei Wang, Yuanfan Guo, Jianhua Han, Hang Xu, and Li Zhang. SlowFocus: Enhancing fine-grained temporal understanding in video LLM. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:81808–81835, 2024.

[41] 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 *International Conference on Machine Learning (ICML)*, pages 41340–41356. PMLR, 2024.

[42] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. SpotEM: Efficient video search for episodic memory. In *International Conference on Machine Learning (ICML)*, pages 28618–28636. PMLR, 2023.

[43] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14313–14323, 2024.

[44] Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. VideoRAG: Retrieval-augmented generation with extreme long-context videos. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2390–2401, 2026.

[45] Xiaoqian Shen, Min-Hung Chen, Yu-Chiang Frank Wang, Mohamed Elhoseiny, and Ryo Hachiuma. Zoom-Zero: Reinforced coarse-to-fine video understanding via temporal zoom-in. *arXiv preprint arXiv:2512.14273*, 2025.

[46] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *International Conference on Machine Learning (ICML)*, pages 54582–54599. PMLR, 2025.

[47] Xiaoqian Shen, Wenxuan Zhang, Jun Chen, and Mohamed Elhoseiny. Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:100805–100830, 2025.

[48] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18221–18232, 2024.

[49] Guangyu Sun, Archit Singhal, Burak Uzkent, Mubarak Shah, Chen Chen, and Garin Kessler. From frames to clips: Training-free adaptive key clip selection for long-form video understanding. *arXiv preprint arXiv:2510.02262*, 2025.

[50] Hui Sun, Shiyin Lu, Huanyu Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Ming Li. MDP3: A training-free approach for list-wise frame selection in video-LLMs. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24090–24101, 2025.

[51] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29118–29128, 2025.

[52] 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. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 959–975, 2025.

[53] Shaoguang Wang, Weiyu Guo, Ziyang Chen, Xuming Hu, and Hui Xiong. Where to focus: Query-modulated multimodal keyframe selection for long video understanding. In *ACM International Conference on Multimedia (ACM MM)*, 2026.

[54] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22958–22967, 2025.

[55] Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. AdaReTaKe: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5417–5432, 2025.

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

[57] Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-R1: Post-training large vision language model for temporal video grounding. *Advances in Neural Information Processing Systems (NeurIPS)*, 38: 83330–83364, 2025.

[58] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3272–3283, 2025.

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

[60] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:28828–28857, 2024.

[61] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786. IEEE, 2021.

[62] Mingze Xu, Mingfei Gao, Shiyu Li, Jiasen Lu, Zhe Gan, Zhengfeng Lai, Meng Cao, Kai Kang, Yinfei Yang, and Afshin Dehghan. SlowFast-LLaVA-1.5: A family of token-efficient video large language models for long-form video understanding. *arXiv preprint arXiv:2503.18943*, 2025.

[63] Ruoliu Yang, Chu Wu, Caifeng Shan, Ran He, and Chaoyou Fu. VideoDetective: Clue hunting via both extrinsic query and intrinsic relevance for long video understanding. *arXiv preprint arXiv:2603.22285*, 2026.

[64] 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 *International Conference on Learning Representations (ICLR)*, pages 98891–98913, 2025.

[65] Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8579–8591, 2025.

[66] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:76749–76771, 2023.

[67] 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 *International Conference on Learning Representations (ICLR)*, pages 84154–84179, 2025.

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

[69] Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. In *International Conference on Machine Learning (ICML)*, pages 153787–153813. PMLR, 2026.

[70] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[71] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 21715–21737, 2024.

[72] Jinglei Zhang, Yuanfan Guo, Rolandos Alexandros Potamias, Jiankang Deng, Hang Xu, and Chao Ma. VTimeCoT: Thinking by drawing for video temporal grounding and reasoning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24203–24213, 2025.

[73] Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. TimeLens: Rethinking video temporal grounding with multimodal LLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10419–10429, 2026.

[74] 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. *Transactions on Machine Learning Research (TMLR)*, 2025.

[75] Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-Frame: Query-aware frame selection and multi-resolution adaptation for video-LLMs. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22056–22065, 2025.

[76] Yuan Zhang, Junwen Pan, Rui Zhang, Xin Wan, Qizhe Zhang, Ming Lu, Qi She, and Shanghang Zhang. ZoomV: Temporal zoom-in for efficient long video understanding. In *ACM International Conference on Multimedia (ACM MM)*, 2026.

[77] Yunzhu Zhang, Yu Lu, Tianyi Wang, Fengyun Rao, Yi Yang, and Linchao Zhu. FlexSelect: Flexible token selection for efficient long video understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:102751–102777, 2025.

[78] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13691–13701, 2025.

[79] Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. In *International Conference on Learning Representations (ICLR)*, pages 98463–98486, 2026.

[80] Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. VideoLucy: Deep memory backtracking for long video understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 38: 25660–25691, 2025.

A MLVU-Dev Results

MLVU-Dev results. Table 6 reports results on MLVU-Dev [78], the development split of MLVU that several prior works report in place of MLVU-Test. It contains 2,174 four-way multiple-choice questions over 1,242 videos in seven tasks. Following the official evaluation, we report M-Avg, the mean of the per-task accuracies; the MLVU-Test results in Table 2 are accuracy over all 502 questions. We use the same Qwen3-VL-8B-Instruct backbone, budgets, and AdaAlloc settings as in Table 2. AdaAlloc improves M-Avg over uniform sampling by 5.2 points at 16K and 4.0 points at 32K, and improves all seven tasks at both budgets. The largest gains are on Action Count (+12.6 at 16K, +9.3 at 32K) and, at 32K, Action Order (+8.1), the multi-detail tasks that require collecting several pieces of evidence spread across the video. Gains on Topic Reasoning are modest (+1.1 and +1.2), as performance is near ceiling.

Table 6: **MLVU-Dev results.** Accuracy (%) on the seven multiple-choice tasks of MLVU-Dev, grouped as in MLVU [78]: Topic Reasoning (TR), Anomaly Recognition (AR), Needle QA (NQA), Ego Reasoning (ER), Plot QA (PQA), Action Order (AO), and Action Count (AC). M-Avg is the mean over the seven tasks. All methods use Qwen3-VL-8B-Instruct. Numbers in parentheses denote absolute gains over uniform sampling under the same budget.

Budget	Method	Holistic		Single-detail			Multi-detail		M-Avg
		TR	AR	NQA	ER	PQA	AO	AC	
16K tokens	Uniform sampling	91.3	76.0	76.9	63.4	75.0	62.9	40.3	69.4
	AdaAlloc	**92.4**	**78.0**	**83.1**	**67.0**	**80.3**	**68.3**	**52.9**	**74.6** (+5.2)
32K tokens	Uniform sampling	92.0	76.5	81.4	66.5	78.3	68.3	49.0	73.2
	AdaAlloc	**93.2**	**79.5**	**83.9**	**67.3**	**81.4**	**76.4**	**58.3**	**77.2** (+4.0)

B VideoMME Breakdown

VideoMME duration-split results. Table 7 reports AdaAlloc results on VideoMME without subtitles across the official short, medium, and long duration splits. The short, medium, and long buckets correspond roughly to videos of 0–2 minutes, 4–18 minutes, and 30–60 minutes, respectively. Performance decreases as video length increases, reflecting the growing difficulty of preserving sufficient temporal coverage under a fixed visual-token budget. Increasing the budget from 16K to 32K leaves short videos unchanged at 79.4 and improves medium videos from 68.7 to 70.6 and long videos from 57.2 to 62.2. The gain grows with video length, suggesting that additional visual capacity is especially useful for longer videos.

Table 7: **VideoMME duration-split breakdown.** We report AdaAlloc performance on VideoMME without subtitles under 16K and 32K visual-token budgets. VideoMME groups videos into short, medium, and long duration buckets.

Budget	$\text{VideoMME}_{\text{w/o sub}}$			
	Short	Medium	Long	Overall
16K	79.4	68.7	57.2	68.4
32K	79.4	70.6	62.2	70.7

VideoMME sub-task breakdown. Table 8 compares AdaAlloc with uniform sampling (32K) across the official VideoMME sub-tasks. AdaAlloc improves 7 of 12 sub-tasks, with the largest gains on Temporal Reasoning (+5.7) and Spatial Perception (+5.6). Four of the five regressions are within one point. The fifth, Temporal Perception, is the sub-task where uniform sampling already scores highest (85.5), leaving little headroom, and contains only 55 questions, so its -5.5 change corresponds to three questions. Counting remains the most challenging category for both methods, likely because it often requires exhaustive visual coverage and reliable aggregation across many frames or repeated instances.

Table 8: **VideoMME sub-task breakdown.** Accuracy (%) of uniform sampling and AdaAlloc on VideoMME without subtitles under the same 32K visual-token budget, broken down by the official sub-task categories. #Q is the number of questions in each sub-task.

Sub-task	#Q	Uniform	AdaAlloc	Δ
Information Synopsis	323	79.6	82.4	+2.8
Spatial Reasoning	56	78.6	80.4	+1.8
Attribute Perception	222	82.0	81.1	−0.9
Object Recognition	354	73.4	76.8	+3.4
Temporal Perception	55	85.5	80.0	−5.5
OCR Problems	139	77.7	79.9	+2.2
Spatial Perception	54	68.5	74.1	+5.6
Action Recognition	313	70.9	70.6	−0.3
Object Reasoning	454	68.1	67.8	−0.3
Temporal Reasoning	177	55.9	61.6	+5.7
Action Reasoning	285	63.5	62.8	−0.7
Counting Problem	268	48.9	50.4	+1.5
Overall	2,700	69.5	70.7	+1.2

C Component Ablation at 16K

Table 9 complements the 32K ablation in Table 5 with a 16K ablation that separates three components: overview-conditioned planning with memory refinement, the global/local hybrid input, and adaptive allocation. Retrieval-only grounds the original question with UniversalVTG and answers from frames sampled from the retrieved segments, without planning or refinement. Local-only runs the full AdaAlloc pipeline but assigns the entire budget to local evidence, and Fixed policy always uses the BALANCED split.

Table 9: **Component ablation under the 16K visual-token budget.** All variants use Qwen3-VL-8B-Instruct with the same final-answering visual-token budget and answerer. Planning & refinement denotes overview-conditioned planning with temporal-memory refinement; variants that use it share the same overview planner, grounding model, and refinement procedure.

Method	Planning & Refinement	Visual Input	Adaptive Allocation	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val
Uniform sampling	×	Global only	×	42.7	55.4	67.1	62.8
Retrieval-only	×	Local only	×	46.5	58.8	65.9	62.5
Local-only	✓	Local only	×	48.9	58.0	66.1	64.8
Fixed policy	✓	Global + local	×	49.6	58.0	66.9	63.4
AdaAlloc	✓	Global + local	✓	**51.7**	**61.2**	**68.4**	**66.0**

Adding overview-conditioned planning and memory refinement on top of question-only retrieval (Retrieval-only vs. Local-only) improves LVBench and LongVideoBench by 2.4 and 2.3 points and VideoMME by 0.2 points, but lowers MLVU-Test by 0.8 points. Adding a fixed global stream (Local-only vs. Fixed policy) then changes accuracy by only +0.7, +0.0, +0.8, and −1.4 points, so the global/local hybrid representation alone does not explain AdaAlloc's gains. Finally, Fixed policy and AdaAlloc share every component and differ only in whether the global/local split is fixed or selected for each video–question pair. Adaptive allocation improves all four benchmarks, by +2.1, +3.2, +1.5, and +2.6 points, and the same comparison at 32K (Table 5) shows gains of 1.6–6.5 points, so question-conditioned allocation provides a consistent improvement beyond combining global and local evidence.

D Robustness and Design Analysis

Sensitivity to overview density. The planner observes only a sparse overview, so subtle local evidence may be absent from its input, and planning errors could propagate to later stages. To measure the end-to-end effect of this information loss, we keep the full pipeline and the 16K final

answering budget unchanged and vary only the planner's visual input: the default overview versus a 2× denser overview with twice as many uniformly sampled frames at the same resolution. As shown in Table 10, the two conditions differ only marginally, with two benchmarks improving and two declining slightly. Doubling the planner's visual input thus yields no consistent benefit, suggesting that the default overview is sufficient for the planner's decisions in aggregate. This analysis does not eliminate the information loss inherent to any finite overview: individual planning errors can still occur, and when the overview and grounding model both miss the decisive event, later stages may fail to recover it. The 16K component ablation (Table 9) provides complementary evidence: adding overview-conditioned planning and memory refinement on top of question-only retrieval (Retrieval-only vs. Local-only), with the retriever, local budget, and final answerer fixed, improves three of the four benchmarks and the average (+1.0 points), so the overview-conditioned stages improve rather than degrade end-to-end accuracy despite the sparse overview.

Table 10: **Sensitivity to overview density.** The final answering budget is fixed at 16K and only the planner's visual input changes. The 2× denser overview samples twice as many frames at the same resolution and spends additional planning compute, so it is a diagnostic rather than a budget-matched alternative.

Planner Overview	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val
Default	51.7	61.2	68.4	66.0
2× denser	51.6	61.4	68.9	65.8
Difference	−0.1	+0.2	+0.5	−0.2

Discrete policies versus direct ratio prediction. AdaAlloc's planner selects among five predefined policies. To test whether this discrete design limits performance, we compare it with a variant in which the planner directly predicts the global/local ratio. As shown in Table 11, the two variants are nearly tied at 16K, while at 32K policy selection is better on LVBench and LongVideoBench, ties on MLVU-Test and VideoMME, and leads by 0.57 points on average. Allowing the planner to output an arbitrary ratio therefore does not consistently improve performance. The discrete policies provide well-separated operating points across the global-to-local spectrum and reduce the planner's decision from predicting a precise continuous value to choosing among a few clear alternatives, while remaining adaptive to each video–question pair. We do not claim that these ratios are optimal; rather, the results indicate that the discrete policy set is a reasonable and stable design.

Table 11: **Discrete policy selection versus direct ratio prediction.** Both variants use the same planner, grounding model, token budget, and answering pipeline, and differ only in whether the planner selects one of the five predefined policies or directly predicts the global/local ratio.

Budget	Allocation	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val	Avg.
16K	Direct ratio prediction	51.1	**61.4**	**68.8**	65.5	61.68
	Policy selection (ours)	**51.7**	61.2	68.4	**66.0**	**61.82**
32K	Direct ratio prediction	54.9	**67.1**	**70.7**	67.0	64.94
	Policy selection (ours)	**56.6**	**67.1**	**70.7**	**67.5**	**65.51**

Sensitivity to the temporal retrieval module. AdaAlloc only requires a module that maps planner-generated queries to relevant video moments. We use UniversalVTG [2] by default because it is an efficient open-vocabulary grounder that directly returns ranked temporal intervals. As an alternative, we use query–frame similarity retrieval, a common mechanism in long-video QA and frame-selection methods [8, 31, 33, 50, 51, 53, 75, 79], which ranks frames by the cosine similarity between each planner-generated query and the frames, both embedded with the Perception Encoder (PE-Core) image–text model [5]. As shown in Table 12, AdaAlloc with query–frame similarity still outperforms uniform sampling on every benchmark; its average is 0.31 points above UniversalVTG at 16K and 1.66 points below it at 32K. We hypothesize that this reflects how the two retrievers use a small local budget: query–frame similarity selects individual high-scoring frames, which is efficient when few local frames are available, whereas UniversalVTG returns contiguous intervals whose temporal context may be better captured by the denser sampling available at 32K. UniversalVTG is therefore

our default, but AdaAlloc retains most of its gains with a simpler and commonly used retrieval mechanism, indicating that its improvements do not depend on a specific grounding model.

Table 12: **Sensitivity to the temporal retrieval module.** AdaAlloc with UniversalVTG [2] (default) or query–frame similarity retrieval, with the planner, allocation policy, token budget, memory refinement, and final answerer unchanged.

Budget	Retriever	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val	Avg.
16K	Uniform sampling	42.7	55.4	67.1	62.8	56.99
	Query–frame similarity	**53.3**	**61.8**	**69.9**	63.5	**62.13**
	UniversalVTG	51.7	61.2	68.4	**66.0**	61.82
32K	Uniform sampling	46.1	61.4	69.5	63.9	60.21
	Query–frame similarity	51.9	**67.3**	70.5	65.7	63.85
	UniversalVTG	**56.6**	67.1	**70.7**	**67.5**	**65.51**

Generalization to another MLLM backbone. To test whether the gains are specific to Qwen3-VL, we apply AdaAlloc to VideoLLaMA3-7B [70], which differs from Qwen3-VL in both vision encoder and training recipe. We use the same evaluation protocol and a matched 32K visual-token budget for AdaAlloc and uniform sampling. As shown in Table 13, AdaAlloc improves VideoLLaMA3-7B on all four benchmarks, with an average gain of 2.25 points, indicating that its benefits are not tied to a single backbone or model family.

Table 13: **Generalization to another MLLM backbone.** Accuracy (%) with VideoLLaMA3-7B [70] under a matched 32K visual-token budget. Numbers in parentheses denote absolute gains over uniform sampling.

Method	LVBench	MLVU-Test	VideoMME$_{\text{w/o sub}}$	LongVideoBench-Val	Avg.
Uniform sampling	43.4	56.2	64.8	61.0	56.35
AdaAlloc	**47.2** (+3.8)	**58.4** (+2.2)	**65.9** (+1.1)	**62.9** (+1.9)	**58.60** (+2.25)

E Allocation Policy Analysis

Distribution of selected allocation policies. Figure 4 shows the distribution of allocation policies selected by AdaAlloc across benchmarks. AdaAlloc first checks whether uniform full-video sampling under the assigned visual-token budget already provides sufficiently dense temporal coverage. If so, it bypasses temporal retrieval and memory refinement, and directly answers from the global overview. Otherwise, the planner selects among five policies: global-only, global-heavy, balanced, local-heavy, and local-only. Across most benchmarks, AdaAlloc favors policies that retain global context while allocating substantial budget to localized evidence. In particular, local-heavy allocation is selected most frequently on LVBench, MLVU, and LongVideoBench, suggesting that many long-video questions require targeted high-resolution inspection after a sparse overview. VideoMME shows a more balanced distribution between balanced and local-heavy policies, indicating a larger fraction of examples where broad temporal coverage and localized verification are both important. Local-only allocation is uncommon outside LVBench, while mixed global/local policies dominate across all four benchmarks. This suggests that preserving global context is often useful even when local evidence is needed.

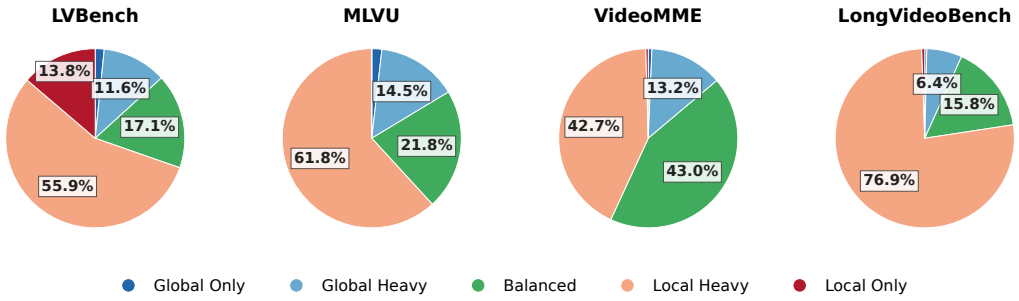

Figure 4: **Distribution of selected allocation policies across benchmarks.** We report the fraction of planner-routed examples assigned to each AdaAlloc policy on LVBench, MLVU, VideoMME, and LongVideoBench. Examples that bypass planning and use direct answering are excluded.

Policy selection by question category. Existing QA benchmarks do not provide a ground-truth global/local ratio, so policy accuracy is not directly defined. We instead examine whether the selected policies follow interpretable patterns, using the question-family metadata of NExT-QA [61], which groups questions into descriptive, causal, and temporal families. Table 14 reports the selected policies on the validation split under the 16K budget. Descriptive questions select global-oriented policies (GLOBAL-ONLY or GLOBAL-HEAVY) more often than the other families, causal questions shift toward BALANCED, and temporal questions have the highest LOCAL-HEAVY rate.

Table 14: **Selected policies by NExT-QA question family (%).** NExT-QA validation split (4,996 questions) under the 16K budget. No question selected LOCAL-ONLY, so the column is omitted. Question families are benchmark metadata and are not available to the planner. Rows may not sum to 100 due to rounding.

Question Family	Global-only	Global-heavy	Balanced	Local-heavy
Descriptive	4.4	13.5	33.1	49.0
Causal	0.2	7.5	43.8	48.6
Temporal	0.1	8.7	38.6	52.5

These labels are unavailable at inference. The planner conditions only on the question, answer choices, and sparse overview, so this is a post-hoc interpretability check. Policy choices vary within each family, reflecting adaptation to individual video–question pairs.

F Inference Cost

Measurement setup. Our accuracy experiments were run on NVIDIA A40 and GH200 GPUs; all cost measurements in this section use NVIDIA A40 GPUs so that methods are compared on the same hardware. We measure total compute and end-to-end wall-clock runtime for uniform sampling, DIG [24], and AdaAlloc using the same Qwen3-VL-8B-Instruct backbone and matched 16K/32K visual-token budgets. Each MLLM call, including AdaAlloc's planning and memory-refinement calls, is capped at the same visual-token budget B; the reported compute and runtime include all calls. Uniform sampling, AdaAlloc, and all DIG components except its initial text-only query classification run on a single NVIDIA A40 GPU. DIG's query classification uses Qwen3-Next-80B-A3B-Instruct, which cannot be served on a single A40 and therefore runs on eight A40 GPUs.

Table 15: **Inference cost under matched visual-token budgets.** Total compute and end-to-end wall-clock runtime of uniform sampling, DIG [24], and AdaAlloc, all with the Qwen3-VL-8B-Instruct backbone.

Budget	Method	Compute (TFLOPs)	Runtime (s)
16K	Uniform sampling	468	14
	DIG [24]	1,871	109
	AdaAlloc	1,146	35
32K	Uniform sampling	1,261	25
	DIG [24]	2,664	119
	AdaAlloc	2,873	58

Results. As shown in Table 15, uniform sampling is the cheapest because it requires a single MLLM call. Compared with DIG, AdaAlloc is $3.1\times$ faster with 39% fewer FLOPs at 16K, and $2.1\times$ faster with 7.8% more FLOPs at 32K. DIG's runtime changes little across budgets because its reward-scoring stage introduces a largely budget-independent cost. Overall, AdaAlloc costs more than single-pass uniform sampling but has substantially lower latency than a comparable training-free selection method.

G Differences from the Submitted Version

While preparing the code release, we found that our UniversalVTG integration did not correctly apply non-maximum suppression and top-K selection to the retrieved temporal proposals. We fixed this issue and now retain the top-5 proposals per query. We also changed all MLLM calls (planning, memory refinement, and answering) from Qwen3-VL's default sampling-based decoding to greedy decoding, which makes predictions deterministic and is better suited to multiple-choice evaluation. We reran our experiments with these changes, and the reported results of those experiments reflect the updated settings. The method, its components, and our conclusions are unchanged. The retriever fix affects every setting that uses UniversalVTG for temporal retrieval, including AdaAlloc and its retrieval-based variants. These changes improve LVBench accuracy (e.g., AdaAlloc at 32K increases from 53.5 to 56.6), with mixed changes on other benchmarks. Some numbers in this version therefore differ from those in the submission and rebuttal.

H Broader Impacts

Improving long-video question answering with existing MLLMs may make video archives easier to search and support accessibility applications. However, missed or misinterpreted evidence can produce incorrect answers, and analyzing personal recordings can create privacy risks or enable intrusive surveillance. Applications should use videos with appropriate authorization and include human review when incorrect answers could have consequential effects.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contributions: framing long-video QA as budgeted visual-token allocation, introducing the training-free AdaAlloc pipeline, and evaluating it under matched token budgets. The scope is limited to long-video multiple-choice QA with fixed MLLM backbones, which matches the experiments and analysis.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations paragraph immediately before the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theoretical results, theorems, or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the datasets, evaluation benchmarks, model backbone, visual-token budgets, baselines, and implementation details needed to reproduce the main experimental comparisons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code is not included with the paper. The paper uses public benchmarks and describes the implementation details, and we will release the full code on the project webpage.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the evaluation benchmarks, data splits, model backbone, visual-token budgets, frame settings, baselines, and implementation details used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports benchmark accuracy under controlled settings but does not include error bars or statistical significance tests.

   Guidelines:

   - The answer [No] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix F reports the GPU type, total compute (TFLOPs), and end-to-end runtime of AdaAlloc and the baselines under both visual-token budgets. Since AdaAlloc is training-free, all compute is inference.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses existing public long-video QA benchmarks and pretrained models, and does not introduce new human-subject data collection or sensitive user data. The work is conducted in accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Appendix H discusses potential benefits for video search and accessibility, as well as risks from incorrect answers, privacy violations, and intrusive surveillance.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained model, image generator, scraped dataset, or other artifact with high misuse risk. The work uses existing public benchmarks and pretrained models, and any released code will be for reproducing the proposed inference pipeline.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper uses existing public benchmarks, pretrained models, and prior code/models, and credits them through citations. However, the paper does not explicitly list the license or terms of use for each asset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release new datasets, model checkpoints, or other high-risk assets. We plan to release code for reproducing the proposed inference pipeline upon publication.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or new research with human subjects. Experiments are conducted on existing public long-video QA benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments, new data collection, or research with human subjects. Experiments are conducted on existing public long-video QA benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper explicitly describes the use of Qwen3-VL-8B-Instruct as the MLLM backbone for planning, temporal-memory refinement, and final answering. The method section and implementation details specify how the MLLM is used as part of the core inference pipeline.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.