Density-Aware Graph Construction for Efficient Long Video RAG

Anonymous ACL submission

Abstract

Graph-based retrieval-augmented generation (RAG) has emerged as a promising paradigm for long video understanding, capturing both temporal continuity and cross-segment semantic associations through structured video graphs. However, existing methods construct one graph node per fixed-length chunk regardless of visual content similarity, leading to redundant nodes in semantically stable regions and unnecessary vision-language model (VLM) extraction costs. We observe that long videos exhibit highly non-uniform semantic density: some temporal regions contain rapidly changing events requiring fine-grained representation, while others consist of visually stable and redundant content. To address this inefficiency, we propose Density-Aware Graph Construction (DAGC), a plug-and-play module that estimates local semantic density from visual similarity signals and adaptively merges redundant neighboring chunks into compact super-nodes before graph construction. DAGC requires no additional training and retains original chunk indices for lossless evidence recovery at inference time. Experiments on MLVU, VideoMME, and LongVideoBench demonstrate that DAGC reduces graph nodes to 45% of the original while achieving 1.3-1.7× graph construction speedup, with less than 1% average performance drop.

1 Introduction

Long video understanding has become increasingly important as video content grows rapidly across diverse domains such as surveillance, education, and entertainment. Comprehending such videos requires models to reason over extended temporal contexts, track evolving entities, and integrate information scattered across thousands of frames. Existing VLMs typically address this by either processing uniformly sampled frames within a fixed context window, or compressing visual tokens to fit within model capacity (Song et al., 2024; Ren et al., 2024; Wang et al., 2025a). However, uniform sampling risks missing critical temporal details, while aggressive token compression may discard semantically important content. More fundamentally, both approaches treat the video as a flat sequence of frames, failing to capture the structured semantic relationships that span across distant temporal segments.

Retrieval-Augmented Generation (RAG) offers a more scalable alternative by selecting only the most relevant video segments for downstream generation, avoiding the need to process the entire video at once (Luo et al., 2025; Jeong et al., 2025). Among RAG-based approaches, graph-based methods (Shen et al., 2025; Xu et al., 2025b) further organize video content into structured graphs that capture both temporal continuity and cross-segment semantic associations, achieving strong performance on long video QA benchmarks. However, a critical inefficiency in graph construction remains overlooked: existing methods assign one graph node to every fixed-length chunk regardless of visual content similarity. As illustrated in Figure 1, long videos exhibit highly non-uniform semantic density: some temporal regions contain rapidly changing events requiring fine-grained representation, while others consist of visually stable and semantically redundant content. Treating these regions with the same granularity leads to unnecessary graph nodes, inflating VLM extraction costs and retrieval search space without proportional gain in accuracy.

To address this inefficiency, we propose **Density-Aware Graph Construction (DAGC)**, a plug-and-play module that estimates local semantic density from visual signals and adaptively merges redundant neighboring chunks into compact super-nodes before graph construction. Specifically, we compute the visual similarity between adjacent chunks as a proxy for semantic density: high similarity

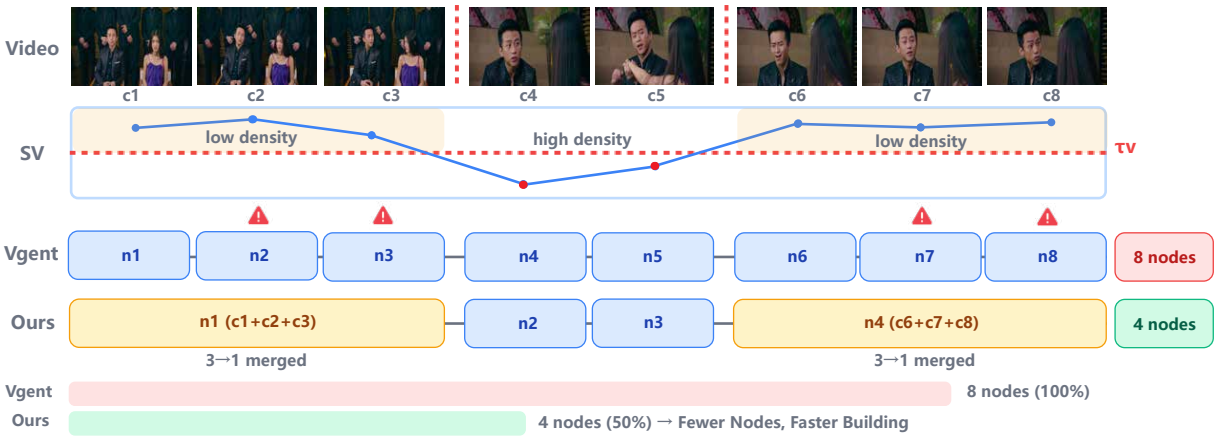

Figure 1: Fixed-granularity methods (Vgent) construct one node per chunk regardless of visual similarity, producing redundant nodes in low-density regions (marked by red exclamation symbols). DAGC merges visually stable chunks into compact super-nodes while preserving event boundaries.

indicates a low-density region where chunks can be safely merged, while low similarity signals a potential event boundary that should be preserved. A maximum merging window W prevents over-compression in extended static scenes. The resulting compressed graph contains significantly fewer nodes, reducing VLM extraction calls and graph construction cost, while each super-node retains its original chunk indices for lossless fine-grained evidence recovery at inference time. DAGC is instantiated on the Vgent framework and requires no additional training or architectural modification.

We evaluate DAGC on three long-video benchmarks: MLVU, VideoMME, and LongVideoBench. Experimental results demonstrate that DAGC reduces graph nodes to approximately 45% of the original while achieving 1.3-1.7× speedup in graph construction, with less than 1% average performance drop. These results suggest that semantic density non-uniformity is a key but previously overlooked source of inefficiency in video graph RAG. Density-aware node merging proves to be an effective and lightweight solution to this problem.

2 Related Work

2.1 Long Video Understanding with VLMs

Long video understanding has attracted growing attention as vision-language models (VLMs) struggle to process thousands of frames within limited context windows (Song et al., 2024; Ren et al., 2024). Existing methods broadly fall into two categories.

Token compression methods reduce visual redundancy before or during LLM inference. Representative works (Wang et al., 2025a; Jiang et al., 2025; Liu et al., 2025) address temporal redundancy by selecting keyframes or compressing token sequences, while VideoTree (Wang et al., 2025b) constructs a hierarchical tree structure to dynamically allocate representation capacity based on query relevance. Although effective, these methods assume fixed-granularity video segmentation and do not adapt the representation structure to the semantic density of video content.

Graph-based memory methods offer a complementary direction, representing videos as structured graphs of entities and events (Chu et al., 2025). This enables more coherent long-range reasoning but comes at the cost of dense graph construction over the entire video. Our work targets precisely this bottleneck: we reduce the number of graph nodes before construction by merging semantically redundant segments, achieving efficient graph-based understanding without sacrificing retrieval coverage.

2.2 Retrieval-Augmented Generation for Video

RAG has emerged as a scalable strategy for long video understanding, avoiding the need to feed all frames into the LLM by retrieving only the most relevant segments (Lewis et al., 2021). Early video RAG methods extract multimodal features from uniformly sampled clips and retrieve by dense similarity (Luo et al., 2025; Jeong et al., 2025). More recent work introduces graph structures to capture cross-segment dependencies: Ren et al. (2025) propose a dual-channel architecture combining graph-based textual knowledge with multimodal visual features for extreme-length videos,

while E-VRAG (Xu et al., 2025b) reduces retrieval-stage computation by 70% via lightweight VLM scoring and inter-frame similarity filtering.

Most relevant to our work is Vgent (Shen et al., 2025), which constructs video graphs with clips as nodes connected by semantic and temporal edges, and introduces an intermediate reasoning step to reduce retrieval noise. We build upon Vgent but address a key limitation: its graph construction uses fixed-granularity segmentation, treating semantically homogeneous segments as separate nodes and introducing unnecessary redundancy. Our adaptive node merging mechanism, applied before graph construction, reduces node count to 46% of the original while maintaining comparable accuracy.

2.3 Video Segmentation / Scene Detection

Segmenting videos into semantically coherent units is a prerequisite to structured video understanding. Early approaches detect shot boundaries using pixel-level frame differences (e.g., PySceneDetect), while supervised methods such as LGSS (Rao et al., 2020) train on annotated scene boundaries using multi-modal clip, segment, and movie-level features. More recently, MDLSeg (Mahon and Lapata, 2025) proposes a parameter-free algorithm based on the minimum description length (MDL) principle, jointly determining segment boundaries via dynamic programming without any user-specified threshold, and directly demonstrating downstream gains on long video QA and summarization.

Our work differs from scene segmentation in both goal and mechanism. Scene segmentation aims to recover narrative boundaries as an end goal; our adaptive merging is a graph construction optimization driven by visual similarity between adjacent segments, with a bounded merge window to preserve retrieval granularity. We do not segment the video globally but selectively consolidate adjacent nodes that are sufficiently similar in visual appearance.

2.4 Graph Construction for RAG

Graph-based RAG has become a mainstream paradigm for knowledge-intensive NLP tasks, with methods such as GraphRAG (Edge et al., 2025), LightRAG (Guo et al., 2025), and NodeRAG (Xu et al., 2025a) progressively improving graph construction efficiency and retrieval quality. GraphRAG organizes entity-relation graphs into semantic communities for global querying; LightRAG introduces dual-level indexing for scalable retrieval; NodeRAG further improves efficiency through heterogeneous node design. Reducing graph construction cost has received attention: Min et al. (2025) replace LLM-based entity extraction with dependency parsing, achieving 94% of LLM-level performance at a fraction of the cost.

These results motivate our core hypothesis: in video graph RAG, the primary efficiency bottleneck lies not in retrieval or generation, but in the construction of an unnecessarily large graph due to fixed-granularity segmentation. Our method addresses this bottleneck directly by merging semantically redundant nodes, offering an efficiency gain that is orthogonal to existing retrieval-side optimizations.

3 Methodology

Long videos exhibit highly non-uniform semantic density: some temporal regions contain rapidly changing events that require fine-grained representation, while others consist of visually and semantically stable content that can be safely compressed. Existing graph-based video RAG methods (Shen et al., 2025) ignore this property and construct one graph node per fixed-length chunk, resulting in redundant nodes for low-density regions and unnecessary semantic extraction cost.

We propose **Density-Aware Graph Construction (DAGC)**, a plug-and-play module that estimates local semantic density from visual signals and adaptively determines graph node granularity before semantic extraction. Specifically, we group visually stable neighboring chunks into compact super-nodes in low-density regions, while preserving fine-grained nodes in high-density regions. This produces a compressed graph whose structure reflects the semantic content of the video rather than its uniform temporal partitioning.

The overall pipeline contains three stages, illustrated in Figure 2: **(1)** Adaptive Node Merging; **(2)** Graph Construction; **(3)** Retrieval & Recovery.

In the first stage, temporally redundant neighboring chunks are compressed into super-nodes according to local semantic density. In the second stage, a semantic graph is constructed over the compressed nodes. In the final stage, retrieved super-nodes are mapped back to their original chunk indices for fine-grained evidence recovery.

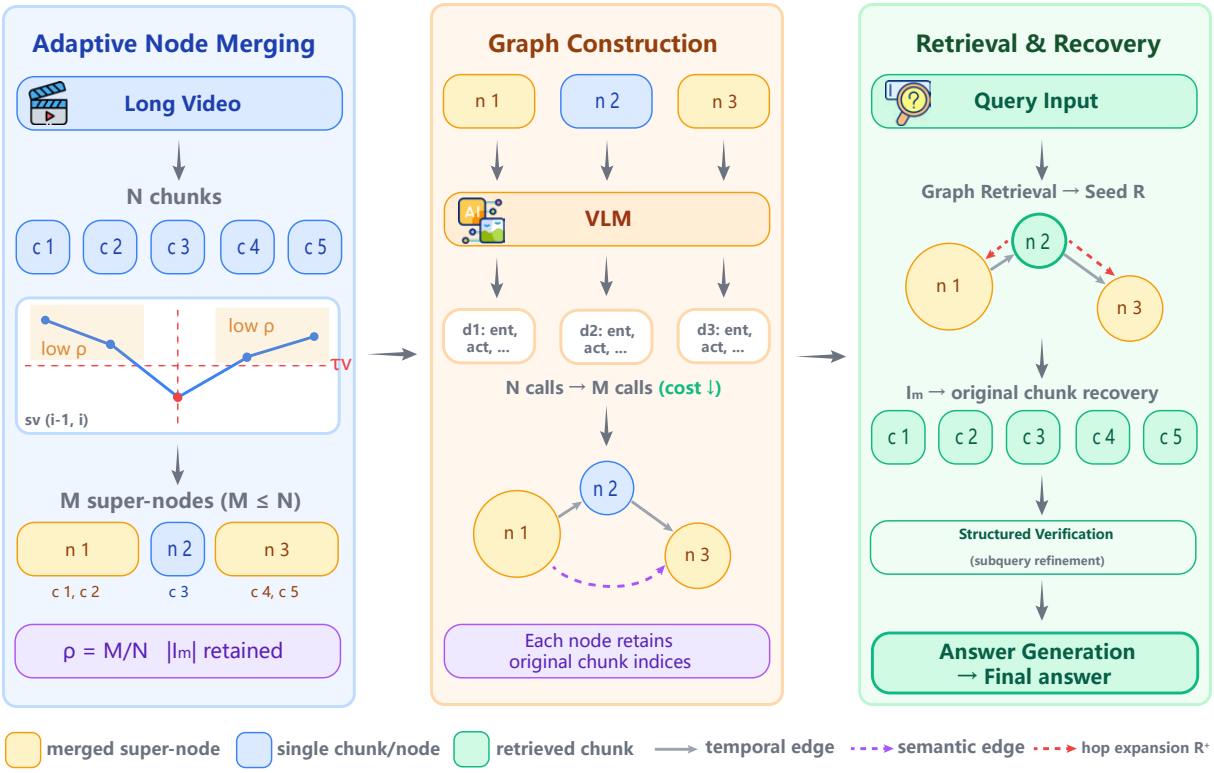

Figure 2: Long videos exhibit non-uniform semantic density: low-density regions (low ρ) contain redundant chunks that can be safely merged, while high-density regions preserve fine-grained nodes. Our method compresses N chunks into M super-nodes ($\rho = M/N$) before graph construction, reducing VLM calls and graph size while retaining original chunk indices $\mathcal{I}_m$ for lossless evidence recovery at inference time.

3.1 Semantic Density Estimation and Adaptive Node Merging

We divide the input video X into N fixed-length chunks $\{c_1, c_2, \ldots, c_N\}$, where each chunk c_i contains K sampled frames. We estimate the local semantic density at the boundary between adjacent chunks using their visual similarity. Specifically, for each chunk we compute a compact visual representation by temporal pooling and L_2 normalization:

$$\mathbf{v}_i = \mathrm{Norm}(\mathrm{Pool}(c_i)). \tag{1}$$

The visual similarity between adjacent chunks c_{i-1} and c_i serves as a proxy for local semantic continuity:

$$s_v(i-1, i) = \mathbf{v}_{i-1}^{\top} \mathbf{v}_i. \tag{2}$$

A high similarity value indicates that the two chunks share similar visual content and are likely to belong to the same semantic event, suggesting low semantic density at their boundary. Conversely, a low similarity value indicates a significant visual transition and high semantic density, implying a potential event boundary that should not be merged.

Based on this density estimate, we merge adjacent chunks greedily into super-nodes. Chunk c_i is merged into the current super-node n_m only when two conditions are jointly satisfied:

$$s_v(i-1, i) \geq \tau_v \quad \text{and} \quad |I_m| < W, \tag{3}$$

where τ_v is the visual similarity threshold controlling the sensitivity of density-based boundary detection, W is the maximum merging window, and I_m is the set of original chunk indices already assigned to n_m. The window constraint W is critical: without it, extended low-density regions such as static scenes would be collapsed into a single node, harming performance on counting and temporal-order questions.

Only temporally adjacent chunks are eligible for merging, preserving the chronological structure of the video. After merging, N original chunks are compressed into $M \leq N$ super-nodes. Each super-node $n_m = (\tilde{c}_m, I_m)$ stores its merged video segment $\tilde{c}_m$ and original chunk index set I_m. The compression ratio $\rho = M/N$ directly reflects the average semantic density of the video: a lower ρ indicates more low-density redundant regions.

3.2 Compressed Semantic Graph Construction

Rather than extracting semantic information from all N original chunks, we construct the semantic graph over the M compressed super-nodes. For each super-node n_m, a video-language model extracts structured semantic information:

$$\mathbf{d}_m = \mathcal{F}_\theta(\tilde{c}_m), \tag{4}$$

where $\mathbf{d}_m$ includes entities, actions, scenes, object states, temporal cues, and a concise summary. Since $M \leq N$, the number of costly VLM inference calls is substantially reduced compared to fixed-granularity construction.

The compressed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}_t \cup \mathcal{E}_s)$ is constructed following Vgent (Shen et al., 2025), where temporal edges $\mathcal{E}_t$ connect adjacent super-nodes to preserve chronological structure, and semantic edges $\mathcal{E}_s$ link nodes sharing semantically equivalent entities via global prototype matching. Each super-node retains its original chunk index set I_m, enabling lossless recovery of fine-grained video evidence at inference time.

3.3 Graph-guided Retrieval

Graph-guided retrieval and answer generation are performed on the compressed graph following the same backbone. For questions requiring temporal context, retrieved seed nodes are expanded along temporal edges:

$$\mathcal{R}^+ = \mathcal{R} \cup \mathcal{N}^t_{h_s}(\mathcal{R}), \tag{5}$$

where $\mathcal{N}^t_{h_s}(\mathcal{R})$ denotes the h_s-hop temporal neighbors on $\mathcal{G}$. After retrieval, each selected super-node is mapped back to its original chunk indices for fine-grained evidence recovery:

$$\mathcal{C}_q = \bigcup_{n_m \in \mathcal{R}^+} \{c_i \mid i \in I_m\}. \tag{6}$$

This two-level design keeps the graph compact during indexing and retrieval, while ensuring the answer model accesses original video granularity during inference.

4 Experiments

4.1 Experimental Settings

Baselines. We evaluate DAGC on three open-source LVLM backbones, including Qwen2.5-VL-7B (Bai et al., 2025), Qwen2.5-VL-3B (Bai et al., 2025), and Qwen2-VL-7B (Yang et al., 2024). We compare with two baselines: (1) **Vanilla LVLM**, which directly performs inference on uniformly sampled video frames without graph-based retrieval; and (2) **Vgent** (Shen et al., 2025), which constructs fixed-granularity video graphs for retrieval-augmented long-video reasoning. DAGC builds upon Vgent and introduces adaptive node compression to reduce redundant graph nodes.

Benchmarks. We conduct experiments on MLVU (Zhou et al., 2025), VideoMME (Fu et al., 2025), and LongVideoBench (LVB) (Wu et al., 2024). These benchmarks cover diverse long-video understanding tasks, including event counting, temporal ordering, visual grounding, topic reasoning, and referred video reasoning.

Implementation Details. All experiments are conducted on 8 NVIDIA A100 40GB GPUs. Following Vgent, videos are sampled at 1 FPS and divided into fixed-length chunks of 64 sampled frames. For DAGC, adjacent chunks are adaptively merged before graph construction. The adjacent visual similarity threshold is set to $\tau_v = 0.95$, and the maximum merging window is set to $W = 3$ by default. Each super-node n_m retains its original chunk index set I_m, which is used for fine-grained evidence recovery after graph retrieval. During retrieval, we keep the top-$k_s = 12$ seed super-nodes before temporal expansion. Unless otherwise specified, the same hyperparameters are used across all benchmarks and backbones.

Metrics. We report **Acc**, **Nodes**, **Speed**, **Avg.**, and **Ret.** as the main evaluation metrics. **Acc** denotes the multiple-choice accuracy on each benchmark. **Nodes** denotes the retained graph-node ratio compared with Vgent, computed as the number of DAGC graph nodes divided by the number of fixed-granularity graph nodes in Vgent. **Speed** denotes the end-to-end runtime speedup relative to Vgent under the same backbone and benchmark. **Avg.** denotes the average accuracy across MLVU, VideoMME, and LVB. **Ret.** denotes the average performance retention relative to Vgent, computed by comparing the average accuracy of DAGC with that of Vgent under the same backbone. For efficiency decomposition, we additionally report wall-clock seconds per minute of video, including offline graph construction, online inference with the pre-built graph reused, and first-query total time.

Models	MLVU			VideoMME			LVB			Avg.	Ret.
	Acc	Nodes	Speed	Acc	Nodes	Speed	Acc	Nodes	Speed		
Qwen2.5-VL-7B	69.0	–	–	70.1	–	–	59.4	–	–	66.2	–
Qwen2.5-VL-7B + Vgent	73.3	100%	1.0×	73.3	100%	1.0×	63.3	100%	1.0×	70.0	100%
Qwen2.5-VL-7B + DAGC	73.4	45%	1.3×	71.1	45%	1.4×	63.1	47%	1.6×	69.2	99%
Qwen2.5-VL-3B	65.0	–	–	67.0	–	–	56.3	–	–	62.8	–
Qwen2.5-VL-3B + Vgent	70.0	100%	1.0×	69.0	100%	1.0×	60.0	100%	1.0×	66.3	100%
Qwen2.5-VL-3B + DAGC	69.9	45%	1.3×	66.2	45%	1.3×	60.6	47%	1.3×	65.6	99%
Qwen2-VL-7B	65.7	–	–	68.6	–	–	56.1	–	–	63.5	–
Qwen2-VL-7B + Vgent	71.7	100%	1.0×	69.7	100%	1.0×	58.9	100%	1.0×	66.8	100%
Qwen2-VL-7B + DAGC	72.1	45%	1.5×	67.3	45%	1.4×	58.6	47%	1.7×	66.0	99%

Table 1: Main results on three long-video benchmarks. Acc. denotes accuracy. Nodes denotes the retained graph-node ratio compared with Vgent. Speed denotes the runtime speedup relative to Vgent. Ret. denotes the average performance retention relative to Vgent.

4.2 Main Results

Table 1 reports the main results on three long-video benchmarks. Compared with Vgent, DAGC consistently reduces the number of graph nodes from 100% to 45%–47% across all backbones and datasets, while achieving 1.3×–1.7× runtime speedup. Despite this aggressive graph compression, DAGC preserves most of the accuracy of the original graph-based retrieval pipeline. On Qwen2.5-VL-7B, DAGC achieves 73.4 on MLVU, which is slightly higher than Vgent, and obtains 63.1 on LVB with only a marginal accuracy drop. The same trend is observed on Qwen2.5-VL-3B and Qwen2-VL-7B: DAGC reduces more than half of the graph nodes while maintaining an average performance retention of about 99%. These results indicate that a large portion of fixed-granularity video graph nodes are redundant for long-video question answering, and that adaptive graph compression can substantially improve efficiency without significantly damaging retrieval quality.

The performance difference across benchmarks is mainly caused by their different information density. On MLVU, DAGC almost preserves the original accuracy and even slightly improves the results on Qwen2.5-VL-7B and Qwen2-VL-7B. Many MLVU questions focus on salient events, temporal order, counting, topic reasoning, or anomaly recognition, where adjacent chunks often contain repeated entities and slowly changing scenes. In this case, merging adjacent similar chunks removes redundant context but rarely removes the decisive evidence. In contrast, VideoMME suffers a larger accuracy drop. A likely reason is that VideoMME contains more extremely long and information-dense videos, including videos close to or above 100 minutes. These videos often include dense narration, frequent scene transitions, and fine-grained factual details. After compression, the retrieval stage can still locate the coarse temporal region, but some answer-distinguishing details may be weakened inside the merged super-nodes.

The speedup is also not strictly proportional to the retained node ratio. Although DAGC keeps only 45%–47% of graph nodes, the end-to-end speedup ranges from 1.3× to 1.7×. This is because the total runtime includes not only graph construction, but also video loading, subtitle processing, retrieval, structured refinement, and final LVLM generation. When LVLM-based graph construction dominates the runtime, node compression brings larger acceleration, as observed on LVB and Qwen2-VL-7B. When online inference or final answer generation occupies a larger proportion, the speedup becomes smaller, even if the graph node ratio is similarly reduced.

There are also several useful outlier observations. On MLVU, DAGC slightly outperforms Vgent with fewer nodes, suggesting that compression can act as a denoising step by reducing redundant or weakly relevant retrieval candidates. However, the larger drop on VideoMME shows the opposite case: when adjacent chunks are visually similar but contain different fine-grained facts, aggressive merging may remove answer-critical details. Therefore, DAGC is most effective when temporal redundancy is high, but it is less reliable for extremely long,

Method	Graph Const.	Online Inf.	First-query
Qwen2.5-VL-7B	–	–	3.12
+ Vgent	24.14	4.12	28.26
+ DAGC	10.73	4.24	14.97

Table 2: Inference time comparison. Graph Const. denotes offline graph construction, Online Inf. denotes query-dependent inference with reused graphs, and First-query denotes graph construction plus one online inference pass.

Variant	Nodes	Acc	Speed
Qwen2.5-VL-7B	–	59.4	–
+ Vgent	100%	63.3	1.0×
+ DAGC w/o SN	100%	62.6	1.0×
+ DAGC w/o RR	47%	60.6	1.6×
+ DAGC w/o TE	47%	62.2	1.6×
+ DAGC	47%	63.1	1.6×

Table 3: Component ablation on LongVideoBench using Qwen2.5-VL-7B. SN denotes super-node compression, RR denotes retrieval and recovery from retrieved super-nodes to original chunks, and TE denotes temporal expansion after recovery.

dense, and detail-sensitive videos. This suggests that future versions should use question-aware or dataset-aware compression instead of a fixed compression setting.

4.3 Efficiency Analysis

Table 2 reports the normalized wall-clock time, where each value denotes the average number of seconds required to process one minute of video. Compared with Vgent, DAGC reduces the graph construction time from 24.14 to 10.73, corresponding to a 55.6% reduction. This confirms that the main efficiency gain comes from reducing the number of graph nodes before LVLM-based graph construction.

The online inference time remains close to Vgent, with 4.24 for DAGC and 4.12 for Vgent. This is expected because DAGC keeps the graph-guided retrieval and structured reasoning pipeline after graph construction. The slight overhead mainly comes from supernode-to-original-chunk backtracking and re-ranking, which are used to recover fine-grained visual evidence.

For the first-query setting, DAGC reduces the total processing time from 28.26 to 14.97, achieving a 47.0% reduction over Vgent. This setting reflects the cost when a video is processed for the first time and no reusable graph is available. Once the graph has been constructed, subsequent queries only require the online inference stage. Therefore, DAGC is especially beneficial for long videos with multiple associated questions, where the reduced graph construction cost can be reused across queries.

5 Analysis and Discussion

This section analyzes the internal behavior of DAGC, including where the efficiency gain comes from, how each component affects the final performance, and how the compression behavior changes under different hyperparameter settings. Different from the main benchmark comparison, this section focuses on efficiency decomposition, component contribution, parameter sensitivity.

5.1 Component Ablation

Table 3 compares six variants. Qwen2.5-VL-7B denotes the vanilla backbone without graph-based retrieval. Vgent denotes the fixed-granularity graph baseline, where all original graph nodes are retained. DAGC w/o SN disables super-node compression and performs retrieval on the original fixed-granularity graph. DAGC w/o RR keeps super-node compression but removes retrieval and recovery from retrieved super-nodes to original chunks. DAGC w/o TE performs super-node compression and original-chunk recovery, but removes temporal expansion around the recovered chunks. The full DAGC combines super-node compression (SN), retrieval and recovery (RR), and temporal expansion (TE).

Vgent improves the backbone from 59.4 to 63.3, confirming that graph-guided retrieval is effective for long-video reasoning. However, it retains all graph nodes and therefore does not reduce graph construction or retrieval cost. DAGC w/o SN also keeps 100% of the nodes and remains at 1.0× speed, indicating that the additional recovery pipeline alone does not bring efficiency gain without adaptive graph compression.

DAGC w/o RR reduces the node ratio to 47% and achieves 1.6× speedup, but its accuracy drops to 60.6. This result shows that directly using compressed super-nodes as final evidence is too coarse for fine-grained video question answering. The gap between DAGC w/o RR and full DAGC indicates that recovering original chunks from retrieved super-nodes is necessary for preserving local visual evidence.

DAGC w/o TE achieves 62.2 accuracy, lower than the full DAGC result of 63.1. This suggests

τ_v	W	top-k_s	Nodes	Acc	Speed
Effect of top-k_s					
0.95	3	5	47%	62.7	1.6×
0.95	3	12	47%	63.1	1.6×
0.95	3	20	47%	62.7	1.6×
Effect of W					
0.95	1	12	99%	62.6	1.0×
0.95	3	12	47%	63.1	1.6×
0.95	7	12	34%	62.5	1.7×
Effect of τ_v					
0.80	3	12	35%	62.5	1.7×
0.90	3	12	39%	62.7	1.7×
0.95	3	12	47%	63.1	1.6×

Table 4: Parameter sensitivity on LongVideoBench using Qwen2.5-VL-7B. τ_v denotes the adjacent visual similarity threshold, W denotes the maximum merging window, and top-k_s denotes the number of seed super-nodes retained before temporal expansion.

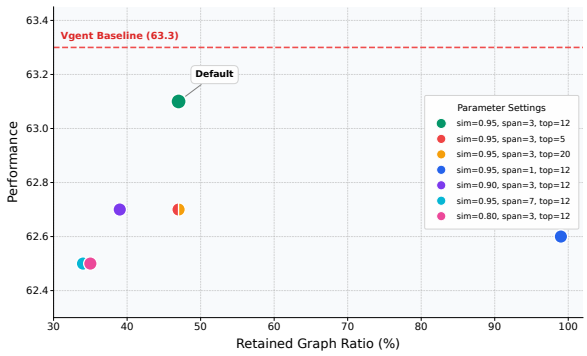

Figure 3: Accuracy–efficiency trade-off under different DAGC compression settings on LongVideoBench.

that recovered chunks may still miss nearby contextual evidence, especially for questions requiring short-range temporal continuity. By combining SN, RR, and TE, the full DAGC retains only 47% of graph nodes while maintaining competitive accuracy and achieving 1.6× speedup, giving the best efficiency–accuracy trade-off among variants.

5.2 Parameter Sensitivity

Table 4 analyzes the sensitivity of DAGC to three key hyperparameters: the adjacent visual similarity threshold τ_v, the maximum merging window W, and the number of seed super-nodes top-k_s retained before temporal expansion. Figure 3 further visualizes the corresponding accuracy–efficiency trade-off. Overall, DAGC remains stable under different compression settings. Although the retained node ratio varies from 34% to 99%, the accuracy remains within a narrow range of 62.5–63.1, indicating that these hyperparameters mainly control compression strength and runtime efficiency rather than causing large performance fluctuations.

We first examine the effect of top-k_s while fixing $\tau_v = 0.95$ and $W = 3$. The default setting with top-$k_s = 12$ achieves the best accuracy of 63.1. Reducing top-k_s to 5 slightly decreases the accuracy to 62.7, suggesting that too few seed super-nodes may miss relevant evidence before temporal expansion. Increasing top-k_s to 20 does not further improve accuracy and also obtains 62.7, indicating that excessive seed nodes may introduce weakly relevant or redundant evidence.

We then analyze the effect of the maximum merging window W. When $W = 1$, DAGC retains 99% of graph nodes and only achieves 1.0× speed, because almost no effective compression is performed while the recovery pipeline is still introduced. Increasing W to 7 reduces the retained nodes to 34% and improves the speed to 1.7×, but the accuracy drops to 62.5. This shows that a larger merging window can improve efficiency, but overly aggressive merging may blur fine-grained temporal boundaries.

Finally, lowering τ_v from 0.95 to 0.90 and 0.80 produces stronger merging, reducing the retained node ratio from 47% to 39% and 35%, respectively. The speed increases to 1.7×, but accuracy decreases from 63.1 to 62.7 and 62.5. This confirms that aggressive compression can improve efficiency but may remove answer-critical details.

Based on these observations, we use $\tau_v = 0.95$, $W = 3$, and top-$k_s = 12$ as the default setting. This configuration retains 47% of graph nodes, achieves 1.6× speedup, and obtains the best accuracy among the tested settings. It therefore provides the most favorable trade-off between graph compression, retrieval quality, and runtime efficiency.

6 Conclusion

In this paper, we present Density-Aware Graph Construction (DAGC), a plug-and-play module for efficient video graph RAG. Motivated by the observation that long videos exhibit non-uniform semantic density, DAGC estimates local semantic density from visual similarity signals and adaptively merges redundant neighboring chunks into compact super-nodes before graph construction. This reduces the number of graph nodes to approximately 45% of the original, achieving 1.3-1.7× graph construction speedup across three benchmarks with less than 1% average performance drop.

Limitations

DAGC currently relies on pixel-level visual similarity for density estimation, which may be unreliable for videos with subtle semantic changes but stable visual appearance, such as talking-head videos where the scene is static but the spoken content evolves rapidly. Future work could explore richer density signals, such as audio or transcript-based features, to improve merging decisions in such scenarios. Additionally, our method is instantiated on a single graph RAG framework; validating its plug-and-play nature across more diverse backbones remains an open direction.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Meng Chu, Yicong Li, and Tat-Seng Chua. 2025. Understanding long videos via llm-powered entity relation graphs. *Preprint*, arXiv:2501.15953.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *Preprint*, arXiv:2405.21075.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. 2025. Videorag: Retrieval-augmented generation over video corpus. *Preprint*, arXiv:2501.05874.

Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, Sungjin Ahn, Jan Kautz, Hongxu Yin, Yao Lu, Song Han, and Wonmin Byeon. 2025. Storm: Token-efficient long video understanding for multimodal llms. *Preprint*, arXiv:2503.04130.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Yudong Liu, Jingwei Sun, Yueqian Lin, Jingyang Zhang, Ming Yin, Qinsi Wang, Jianyi Zhang, Hai Li, and Yiran Chen. 2025. Keyframe-oriented vision token pruning: Enhancing efficiency of large vision language models on long-form video processing. *Preprint*, arXiv:2503.10742.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. 2025. Videorag: Visually-aligned retrieval-augmented long video comprehension. *Preprint*, arXiv:2411.13093.

Louis Mahon and Mirella Lapata. 2025. Parameter-free video segmentation for vision and language understanding. *Preprint*, arXiv:2503.01201.

Congmin Min, Sahil Bansal, Joyce Pan, Abbas Keshavarzi, Rhea Mathew, and Amar Viswanathan Kannan. 2025. Towards practical graphrag: Efficient knowledge graph construction and hybrid retrieval at scale. *Preprint*, arXiv:2507.03226.

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A local-to-global approach to multi-modal movie scene segmentation. *Preprint*, arXiv:2004.02678.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. *Preprint*, arXiv:2312.02051.

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

Xiaoqian Shen, Wenxuan Zhang, Jun Chen, and Mohamed Elhoseiny. 2025. Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. *Preprint*, arXiv:2510.14032.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. 2024. Moviechat: From dense token to sparse memory for long video understanding. *Preprint*, arXiv:2307.16449.

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. 2025a. Retake: Reducing temporal and knowledge redundancy for long video understanding. *Preprint*, arXiv:2412.20504.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2025b. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *Preprint*, arXiv:2405.19209.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Preprint*, arXiv:2407.15754.

Tianyang Xu, Haojie Zheng, Chengze Li, Haoxiang Chen, Yixin Liu, Ruoxi Chen, and Lichao Sun. 2025a. Noderag: Structuring graph-based rag with heterogeneous nodes. *Preprint*, arXiv:2504.11544.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

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. 2025. Mlvu: Benchmarking multi-task long video understanding. *Preprint*, arXiv:2406.04264.

A Experimental Results

Model	Count	Ego	Needle	Order	PlotQA	Topic	Anomaly	Overall
Qwen2.5-VL-7B	42.3	60.0	79.4	66.7	75.1	86.4	73.0	69.0
Qwen2.5-VL-7B + Vgent	59.6	61.4	81.1	73.4	76.1	87.1	74.5	73.3
Qwen2.5-VL-7B + DAGC	59.6	62.0	82.0	72.4	76.6	87.1	74.0	73.4
Qwen2.5-VL-3B	32.3	52.9	78.2	56.2	71.5	88.1	76.0	65.0
Qwen2.5-VL-3B + Vgent	53.3	58.0	80.0	62.5	71.9	89.0	75.5	70.0
Qwen2.5-VL-3B + DAGC	52.4	58.2	78.6	64.0	72.1	88.3	75.5	69.9
Qwen2-VL-7B	33.2	66.1	79.4	53.6	71.1	86.6	70.2	65.7
Qwen2-VL-7B + Vgent	61.2	67.7	82.2	61.0	71.5	87.1	71.0	71.7
Qwen2-VL-7B + DAGC	63.9	67.6	81.9	61.7	71.3	87.1	71.0	72.1

Table 5: Detailed results on MLVU across seven tasks. Count, Ego, Needle, Order, PlotQA, Topic, and Anomaly are the seven evaluated tasks. Overall denotes the aggregate accuracy across all seven tasks.

Model	Short	Medium	Long	Overall	Speed
Qwen2.5-VL-7B + Vgent	78.76	72.63	68.33	73.25	1.0×
Qwen2.5-VL-7B + DAGC	78.67	72.41	62.22	71.11	1.4×
Qwen2.5-VL-3B + Vgent	75.33	68.55	63.66	69.19	1.0×
Qwen2.5-VL-3B + DAGC	74.23	65.22	59.11	66.19	1.3×
Qwen2-VL-7B + Vgent	76.43	69.55	63.22	69.74	1.0×
Qwen2-VL-7B + DAGC	75.55	66.55	59.77	67.30	1.3×

Table 6: Detailed results on VideoMME across different video durations. Short, Medium, and Long denote the three duration subsets. Overall denotes the aggregate accuracy over all duration subsets. Speed denotes the relative speedup over the corresponding Vgent baseline.

Model	E2O	SSS	T2E	S2O	SAA	TAA	S2A	SOS	T2A
Qwen2.5-VL-7B + Vgent	75.4	46.4	72.3	66.7	61.1	59.8	72.1	65.4	70.4
Qwen2.5-VL-7B + DAGC	75.4	44.8	69.2	68.1	58.3	58.5	70.5	65.4	67.9
Qwen2.5-VL-3B + Vgent	64.6	47.4	64.6	62.5	52.8	52.4	62.5	65.4	63.0
Qwen2.5-VL-3B + DAGC	64.6	43.8	66.2	63.9	54.2	53.7	62.5	64.2	64.2
Qwen2-VL-7B + Vgent	70.8	46.4	66.2	56.9	56.9	52.4	71.6	61.7	56.8
Qwen2-VL-7B + DAGC	70.8	47.4	66.2	55.6	52.8	53.7	72.7	63.0	59.3

Table 7: Detailed category-level results on LongVideoBench, Part I. The table reports the first group of question-category accuracies.

Model	S2E	T3O	T2O	O3O	O2E	T3E	TOS	E3E	Overall	Speed	Ret.
Qwen2.5-VL-7B + Vgent	76.3	59.5	64.1	60.6	69.3	49.3	40.0	68.1	63.33	1.0×	–
Qwen2.5-VL-7B + DAGC	74.2	56.8	69.2	68.2	68.2	49.3	42.7	67.0	63.07	1.7×	48%
Qwen2.5-VL-3B + Vgent	77.4	62.2	55.1	54.5	67.8	54.8	44.0	67.0	60.00	1.0×	–
Qwen2.5-VL-3B + DAGC	75.3	59.5	59.0	62.1	70.5	52.1	42.7	68.1	60.46	1.3×	48%
Qwen2-VL-7B + Vgent	68.8	66.2	57.7	62.1	62.5	49.3	32.0	62.8	58.88	1.0×	–
Qwen2-VL-7B + DAGC	71.0	58.1	57.7	62.1	62.5	49.3	33.3	59.6	58.55	1.7×	48%

Table 8: Detailed category-level results on LongVideoBench, Part II. The table reports the remaining question-category accuracies and includes the overall accuracy, relative speedup, and retained-node ratio. Ret. denotes the retained-node ratio after adaptive graph compression.

Variant	E2O	SSS	T2E	S2O	SAA	TAA	S2A	SOS	T2A
Qwen2.5-VL-7B	70.8	46.4	66.2	55.6	62.8	53.7	72.7	63.0	59.3
+ Vgent	75.4	46.4	72.3	66.7	61.1	59.8	72.1	65.4	70.4
+ DAGC w/o SN	72.3	43.8	70.8	65.3	56.3	57.3	70.1	65.0	67.9
+ DAGC w/o RR	72.3	42.7	72.3	63.9	58.3	57.3	70.1	64.2	69.1
+ DAGC w/o TE	75.4	44.8	69.2	66.7	58.3	58.5	70.5	65.4	67.9
+ DAGC	75.4	44.8	69.2	68.1	58.3	58.5	70.5	65.4	67.9

Table 9: Component ablation results on LongVideoBench, Part I. The table reports the first group of question-category accuracies. SN denotes supernode merging, RR denotes original-chunk reranking, and TE denotes temporal expansion.

Variant	S2E	T3O	T2O	O3O	O2E	T3E	TOS	E3E	Overall	Speed	Ret.
Qwen2.5-VL-7B	71.0	58.1	57.7	62.1	62.5	49.3	33.3	59.6	59.4	1.0×	–
+ Vgent	76.3	59.5	64.1	60.6	69.3	49.3	40.0	68.1	63.3	1.0×	100%
+ DAGC w/o SN	75.3	58.1	67.9	66.7	69.3	50.7	40.0	68.1	62.6	1.0×	100%
+ DAGC w/o RR	74.2	58.1	65.4	62.1	65.5	49.3	38.7	67.0	60.6	1.6×	47%
+ DAGC w/o TE	73.1	55.4	67.9	65.2	67.0	48.6	41.3	65.9	62.2	1.6×	47%
+ DAGC	74.2	56.8	69.2	68.2	68.2	49.3	42.7	67.0	63.1	1.6×	47%

Table 10: Component ablation results on LongVideoBench, Part II. The table reports the remaining question-category accuracies together with overall accuracy, relative speedup, and retained-node ratio. The full DAGC variant achieves the best overall accuracy while preserving the efficiency advantage over the Vgent baseline.

Setting	E2O	SSS	T2E	S2O	SAA	TAA	S2A	SOS	T2A
τ_v=0.95, W=3, top-k_s=12 (Default)	75.4	44.8	69.2	68.1	58.3	58.5	70.5	65.4	67.9
τ_v=0.95, W=3, top-k_s=5	71.9	45.4	70.8	65.3	56.9	58.5	69.0	64.2	69.1
τ_v=0.95, W=3, top-k_s=20	73.8	43.6	72.3	65.3	56.9	57.3	69.3	63.0	67.9
τ_v=0.95, W=1, top-k_s=12	72.3	43.8	70.8	65.3	56.3	57.3	70.1	65.0	67.9
τ_v=0.90, W=3, top-k_s=12	73.8	48.5	69.2	66.7	58.3	57.3	70.5	63.0	67.9
τ_v=0.95, W=7, top-k_s=12	72.3	45.4	70.8	65.3	58.3	57.3	70.5	64.2	70.4
τ_v=0.80, W=3, top-k_s=12	72.3	43.3	72.3	65.3	58.3	57.3	69.3	64.2	67.9

Table 11: Parameter sensitivity results on LongVideoBench, Part I. The table reports the first group of question-category accuracies under different adaptive graph compression settings.

Setting	S2E	T3O	T2O	O3O	O2E	T3E	TOS	E3E	Overall	Speed	Ret.
τ_v=0.95, W=3, top-k_s=12 (Default)	74.2	56.8	69.2	68.2	68.2	49.3	42.7	67.0	63.1	1.6×	47%
τ_v=0.95, W=3, top-k_s=5	74.2	58.1	65.4	66.7	68.2	49.3	45.3	68.1	62.7	1.6×	47%
τ_v=0.95, W=3, top-k_s=20	74.2	56.8	67.9	68.2	69.0	49.3	42.7	69.1	62.7	1.6×	47%
τ_v=0.95, W=1, top-k_s=12	75.3	58.1	67.9	66.7	69.3	50.7	40.0	68.1	62.6	1.0×	99%
τ_v=0.90, W=3, top-k_s=12	74.2	58.1	64.9	66.7	69.3	47.9	41.3	69.1	62.8	1.7×	39%
τ_v=0.95, W=7, top-k_s=12	74.2	58.1	65.4	66.7	68.2	49.3	42.7	67.0	62.5	1.7×	34%
τ_v=0.80, W=3, top-k_s=12	73.1	58.1	66.7	65.2	69.3	49.3	44.0	68.1	62.5	1.7×	35%

Table 12: Parameter sensitivity results on LongVideoBench, Part II. The table reports the remaining question-category accuracies together with overall accuracy, relative speedup, and retained-node ratio. The default setting achieves the best overall accuracy while maintaining a clear efficiency gain.