FrameRouter: Frame Budget Routing for Long-Video Understanding on Video-MME-v2 and Beyond

Bingjun Luo
Tsinghua University
Beijing, China
bingjunluo@outlook.com

Jialin Guo
Harbin Engineering University
Harbin, China
guojialin@hrbeu.edu.cn

Siqi Li*
BNRist, THUIBCS, School of Software
Tsinghua University
Beijing, China
lisiqi19971013@gmail.com

Abstract

Long-video question answering benchmarks like Video-MME-v2 typically evaluate systems under a uniform frame budget for every question, despite stark variation in the evidence each query actually demands. We analyze this mismatch on Video-MME-v2 and find that different question subsets prefer different frozen frame-budget policies, while uniformly increasing the frame count does not reliably improve all metrics. These findings motivate fixed-mean frame-budget routing, a controlled inference setting in which each question may receive a different frame budget while the average budget matches a uniform reference. We introduce FrameRouter, a training-free test-time framework that combines an evidence-demand router, a budget-constrained allocation rule, and a frozen bank of frame-sampling policies. By controlling *how much* visual evidence each question receives while leaving the underlying samplers unchanged, FrameRouter is orthogonal to query-aware frame selection. Across Video-MME-v2 and several long-video understanding benchmarks, FrameRouter improves long-video question answering by using the same average visual budget more selectively across questions.

1 Introduction

Long-video question answering has become a stress test for multimodal models: the system must search, aggregate, and reason over visual evidence spread across minutes or hours of video. Recent benchmarks such as Video-MME-v2, MLVU, and LongVideoBench make this pressure explicit by evaluating models across long durations, diverse domains, subtitles, and multi-step question groups [Fu et al., 2026, Zhou et al., 2025, Wu et al., 2024]. Because compute and context length restrict the number of frames that current video–language models can ingest, frame budget has become a central evaluation knob. The de facto protocol is to fix a uniform per-question budget and apply it to every instance. This is simple and reproducible, but it tacitly assumes that every question deserves the same amount of visual evidence.

That assumption sits uneasily with the structure of modern long-video benchmarks. Their questions span heterogeneous evidence demands: some are resolvable from sparse global context, scene-level commonsense, or subtitle cues, while others hinge on local visual details, long-range temporal

*Corresponding author: Siqi Li (lisiqi19971013@gmail.com).

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

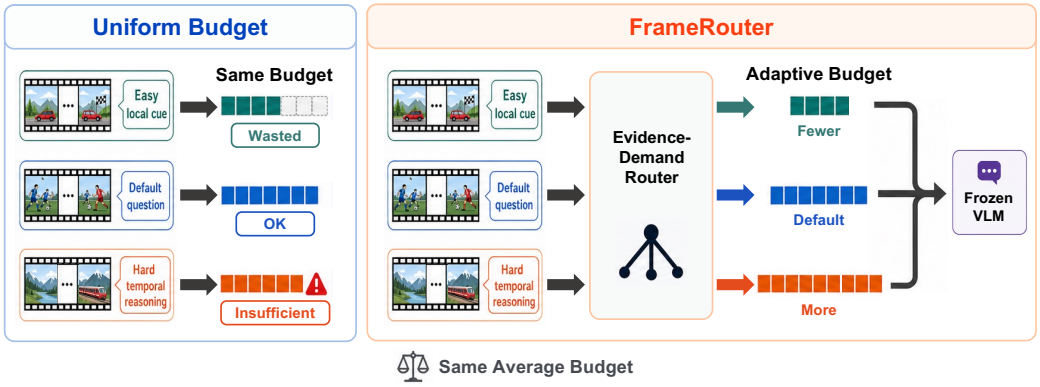


Figure 1: Overview of fixed-mean frame-budget routing. Uniform inference gives every question the same visual budget despite heterogeneous evidence demands. FrameRouter routes each question to an appropriate frozen budget policy using question-only signals, while preserving the same dataset-level average budget.

association, action ordering, or repeated-action counting. Spending the same number of frames on every question therefore creates two coupled inefficiencies: low-demand questions consume frames they may not need, and high-demand questions remain under-observed. Recent efficient video-QA work makes a related observation, showing that uniformly sampled long videos contain substantial redundancy and that question-aware frame selection can recover crucial spatiotemporal cues [Park et al., 2026, Zhang et al., 2025, Yao et al., 2025, Ben-Ami et al., 2026]. Our focus is orthogonal to which frames a question keeps: we ask how much evidence each question should receive under the same average cost.

Our diagnostic analysis on Video-MME-v2 reinforces this view. Within a frozen bank of low-, mid-, and high-budget full-video uniform policies, neighboring budgets repair a non-trivial slice of mid-budget errors while also breaking questions that the mid-budget policy answers correctly. The three policies do not form nested error sets; their disagreement varies systematically with question group and difficulty. This motivates *fixed-mean frame-budget routing*: given a uniform reference budget B_{ref} over N questions, an admissible router may assign per-question budgets B_i subject to $\frac{1}{N}\sum_{i=1}^{N} B_i = B_{\text{ref}}$. Under this strict equality, any gain must come from reallocating visual evidence across questions, not from increasing the average number of frames. This framing is complementary to query-aware frame selectors and to architecture-level token compressors, which adapt what or how to sample within a fixed per-question quota; here, we ask how the dataset-level budget itself should be distributed.

We instantiate this formulation as **FrameRouter**, a training-free test-time framework that routes *frame budgets*, not frame locations. FrameRouter scores the benchmark question prompt, whose multiple-choice options are visible to any evaluated system, and uses those question-only priorities to choose among frozen low-, mid-, and high-budget full-video policies under an exact dataset-level mean-budget constraint. The router never sees the video, subtitles, sibling questions, other videos, labels, or candidate policy outputs, and it neither fine-tunes the underlying VLM nor introduces a new within-question frame selector. This keeps the comparison tightly controlled: low- and high-budget assignments change the distribution of visual evidence across questions, while the average number of frames remains identical to the uniform reference. Across Video-MME-v2, LongVideoBench, MVBench, and VideoMMMU, FrameRouter yields stable matched-budget improvements. Quota sweeps and post-hoc route audits further show that the benefit is not simply a high-budget effect, but comes from reallocating the fixed average budget across heterogeneous questions.

In summary, our contributions are threefold as follows:

- We diagnose non-monotonic frame-budget effects on Video-MME-v2 and formalize fixed-mean frame-budget routing as a controlled inference problem, separating dataset-level budget reallocation from variable-frame efficiency and from within-question frame selection.

- We instantiate this problem as FrameRouter, combining a frozen full-video uniform policy bank, a question-only evidence-demand priority parser, and a quota-constrained allocator that holds the mean frame budget exactly equal to the uniform reference.
- We achieve the state-of-the-art open-source result on Video-MME-v2 under the matched-budget protocol, and show consistent improvements across other benchmarks, including LongVideoBench, MVBench, and VideoMMMU.

2 Related work

2.1 Long-video understanding benchmarks

Long-video understanding has moved from short, isolated clips toward benchmarks that require models to integrate evidence over minutes or hours. EgoSchema introduced a diagnostic multiple-choice setting over three-minute egocentric clips, emphasizing that temporal difficulty is not captured by clip length alone [Mangalam et al., 2023]. Video-MME broadened evaluation across video duration, domain, and modality, including subtitles and audio, and exposed a large gap between frontier proprietary systems and open models on long-video analysis [Fu et al., 2025]. MLVU and LongVideoBench further expanded the evaluation surface with multi-task long-video understanding and interleaved video-language contexts [Zhou et al., 2025, Wu et al., 2024]. These benchmarks made frame budget a practical evaluation variable: model quality depends not only on the backbone, but also on how many frames or visual tokens are admitted into the context.

Video-MME-v2 sharpens this pressure by changing both the data distribution and the scoring rule [Fu et al., 2026]. It contains 800 recent videos and 3,200 questions, organized into groups of four interrelated questions. Beyond average per-question accuracy, the benchmark reports group-level non-linear scores, including Non-Lin, Consistency, and Coherence, that reward models for answering related questions consistently and for preserving multi-step reasoning chains within a group. These metrics matter for budgeted inference because the same video can contain questions with very different evidence demands: global story context, long-range temporal aggregation, local perceptual details, OCR, or screen evidence. A uniform frame budget is therefore poorly matched to the heterogeneity that Video-MME-v2 is well suited to surfacing.

2.2 Efficient video question answering

The high cost of video QA has motivated methods that reduce redundant visual processing. Early long-video QA systems often uniformly sampled many frames and converted them into captions or visual tokens, but long videos contain large regions that are irrelevant to a particular question. As models and benchmarks move to longer inputs, evaluation quality depends not only on the VLM backbone but also on frame budget, sampling density, and whether auxiliary channels such as subtitles or interleaved text are admitted into the context. The official 64-frame-with-subtitle setting in Video-MME-v2 makes this cost–accuracy tradeoff explicit: more evidence can help difficult questions, but increasing the per-question budget uniformly also spends frames on questions that may not need them.

Existing efficient video-QA methods mostly reduce the cost of processing each question. LVNet explicitly argues that many sampled frames are unnecessary and uses a hierarchical keyframe selector to reduce the captioning burden for long-form QA [Park et al., 2026]. Agentic approaches such as VideoAgent ask an LLM controller to iteratively search for useful visual information under a limited frame budget [Wang et al., 2024]. Architecture-level approaches also attack the cost problem by compressing the video representation: MA-LMM maintains a memory bank for long-term video information [He et al., 2024], while LongVU uses spatiotemporal adaptive compression to reduce token redundancy while preserving salient visual detail [Shen et al., 2024]. These works establish that long-video QA is governed by a cost–accuracy tradeoff, but their primary unit of adaptation is usually the video, the segment, or the selected frame set for a single query.

2.3 Adaptive visual evidence and test-time compute

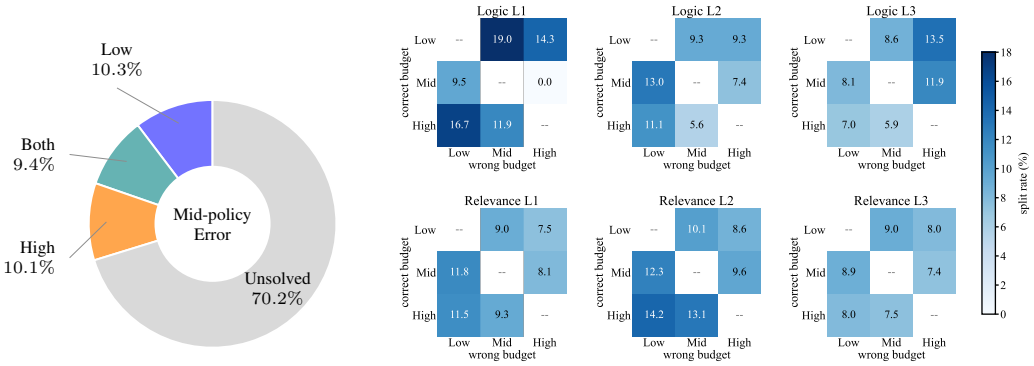
Most recent efficient video-LLM methods adapt *which* visual evidence is shown to the model, while a broader adaptive-compute literature studies inference-time allocation more generally. Query-aware

frame selection methods score frames by their relevance to the current question, often combining temporal coverage with visual-textual matching. Q-Frame selects frames and adapts resolution according to the video content and query [Zhang et al., 2025]; GenS trains a generative frame sampler under capped frame budgets [Yao et al., 2025]; and HiMu composes expert grounding signals for hierarchical frame selection [Ben-Ami et al., 2026]. In language-model serving and conditional computation, FrugalGPT uses query-dependent cascades to reduce cost [Chen et al., 2023], RouteLLM routes between stronger and weaker LLMs as a cost-quality tradeoff [Ong et al., 2024], and Mixture-of-Depths allocates transformer computation dynamically across tokens [Raposo et al., 2024]. Input-dependent mixture-of-experts routing has likewise been adopted beyond language modeling [Xi et al., 2025]. These works share the premise that uniform computation is wasteful when inputs have heterogeneous difficulty.

Our work connects these two perspectives in long-video QA. Rather than learning a new retriever, selecting timestamps, or routing between different models, FrameRouter reallocates a fixed dataset-level frame budget across questions using frozen VLM inference configurations. Thus it is complementary to within-question frame selection: those methods ask which evidence to keep, whereas we ask how much visual evidence each question should receive under the same average budget.

3 Analysis: why route frame budgets?

We argue that uniform frame budgets are a poor unit of comparison on Video-MME-v2, and that the right unit is a per-question budget assignment chosen under a dataset-level mean-frame constraint. To make this argument without conflating budget allocation with other design choices, we analyze a *frozen policy bank* of three full-video uniform sampling policies: low, mid, and high budget, instantiated as 48, 64, and 96 frames. The policies use the same video–language model, prompt, subtitles setting, and full-video uniform sampler, differing only in the number of sampled frames. We use this bank only as a diagnostic object: it lets us ask whether errors made by the mid-budget reference are sometimes recoverable by neighboring budgets, and whether such recovery has structure across question types. Any routed policy is compared to the uniform reference under the same dataset-level mean frame budget; the formal budget constraint is introduced in Section 4. Figure 2 evaluates the question-level headroom and structure of this bank in the main Think-with-subtitles setting on the Video-MME-v2 full set.



(a) Mid-policy error recovery.

(b) Correction structure by question group.

Figure 2: Question-level budget heterogeneity in the frozen Think with-subtitles policy bank. Left: among errors made by the mid policy, 29.8% are answered correctly by at least one neighboring budget. Right: pairwise correction matrices split by `group_type` and `level`; rows denote the policy that is correct and columns denote the policy that is wrong.

3.1 Mid-budget error recovery

Figure 2(a) first asks whether the standard mid-budget policy is merely making irrecoverable errors, or whether some of its errors are artifacts of choosing one uniform budget for every question. We use the mid policy as the reference and inspect only its errors. This is an oracle diagnostic over the frozen

policy bank: it is not available to the deployed router, but it measures the headroom available to a question-conditioned budget selector restricted to the same bank.

The recoverable errors are split almost evenly between the two neighboring budgets: low-only and high-only recovery occupy similar shares, with a smaller overlap recovered by both neighboring policies. Thus the headroom is not only a high-budget effect. Increasing the budget can also hurt: the high policy introduces its own errors on cases that the mid policy answers correctly. The frozen bank therefore does not define a nested error set in which more frames are uniformly better. Instead, it defines an allocation problem: under the same mid-budget mean, can we promote the questions that need denser coverage while compressing questions whose evidence is already sufficiently localized?

3.2 Question-level disagreement

We next ask where this non-nested behavior concentrates. If disagreement were uniformly scattered across the benchmark, a question-conditioned router would have little structure to exploit. Instead, the correction patterns vary with observable question groups and difficulty levels.

Figure 2(b) shows that this disagreement changes with both question group and difficulty level. We split the questions with official group and level labels by $(\texttt{group_type}, \texttt{level})$ and plot a small pairwise correction matrix for each split: each entry is the percentage of questions in that split for which the row budget is correct and the column budget is wrong.

This pattern indicates that the three uniform policies do not form nested error sets: they make systematically different mistakes at the question level. We use this disagreement only as evidence that budget heterogeneity exists. A deployable route must infer budget demand before answering, from the current question and options alone.

3.3 From diagnostics to design

These diagnostics suggest a narrow design space for a deployable router. To isolate budget allocation from frame-selection effects, the route should choose among the frozen full-video policies in Π rather than introduce a new sampler or temporal localizer. To avoid test-time leakage, the routing signal should be computed from the current question prompt only, excluding labels, videos, subtitles, sibling questions, other videos, model answers, and policy outcomes. Because the route cannot observe which policy will be correct, the signal should be an interpretable prior over evidence demand rather than a direct correctness predictor. Finally, comparisons must enforce the same dataset-level mean frame budget as the uniform reference; higher means are separate operating points.

4 FrameRouter

FrameRouter turns the question-level budget disagreement in Section 3 into a deployable routing method: it estimates question-side evidence demand, routes each question to one frozen full-video uniform policy, and enforces an exact mean-frame budget. It routes *frame budgets*, not frame locations, and never modifies the underlying video–language model.

4.1 Problem formulation

Let $\mathcal{D} = \{(v_i, q_i, s_i, y_i)\}_{i=1}^{N}$ denote a video question-answering dataset, where v_i is a video, q_i is the complete question prompt including the answer options, s_i is any auxiliary textual context associated with the question (e.g., subtitles when provided by the benchmark), and y_i is the ground-truth answer. The auxiliary context s_i may be used by the answer model, but it is not used as a routing signal. The downstream task is multiple-choice video question answering: produce a prediction $\hat{y}_i$ for each instance i. We treat the number of video frames spent on instance i as a problem-level variable $B_i \in \mathbb{N}$, abstracting away the mechanism that produces those frames. A frame-budgeted predictor is any procedure that, given B_i frames of v_i together with (q_i, s_i), returns an answer

$$\hat{y}_i \;=\; F(v_i,\, q_i,\, s_i;\, B_i)\,. \tag{1}$$

To make accuracy comparisons faithful to a fixed compute envelope, any solution must spend, on average, exactly B_{ref} frames per question. We denote the corresponding uniform reference policy by π_{ref} and require $\frac{1}{N}\sum_{i=1}^{N} B_i = B_{\mathrm{ref}}$. Operating points that under- or over-spend this reference

are different points on the cost–accuracy plane and must be reported separately. FrameRouter is a practical approximation to this fixed-budget assignment problem: each parser call is question-only, and the solver uses only parser scores and route quotas rather than labels or policy outcomes.

4.2 Method overview

FrameRouter instantiates the fixed-mean budget assignment problem with a method-specific frozen policy bank

$$\Pi \;=\; \{\pi_{\mathrm{low}},\, \pi_{\mathrm{mid}},\, \pi_{\mathrm{high}}\}, \tag{2}$$

where each policy uniformly samples from the full video and uses the same frozen video–language model and prompting protocol. The policies differ only in their frame budgets, with $B_{\mathrm{low}} < B_{\mathrm{mid}} < B_{\mathrm{high}}$. FrameRouter does not modify the policy bank, the sampler, or the underlying model; it only decides which bank member to use for each question.

The routing pipeline has two stages. First, an evidence-demand priority parser assigns each instance two priorities: whether it is safe to compress to the low-budget policy, and whether it is worth promoting to the high-budget policy. Second, a quota-constrained allocator converts these priorities into routes over Π while preserving the same dataset-level mean frame budget as the uniform reference. At inference time, FrameRouter executes only the selected policy for each question; the other bank policies are not run, so the method performs budget routing rather than late selection over candidate answers.

4.3 Evidence-demand priority parser

The analysis in Section 3 shows that budget effects vary across questions rather than following a simple high-budget-fixes-low-budget pattern. Because group labels, videos, subtitles, and policy outcomes are not available to a deployed router, FrameRouter estimates evidence demand from the question prompt itself. To keep this signal interpretable and auditable, it does not directly predict which policy will answer correctly. Instead, a frozen text-only parser assigns two priorities to the current question:

$$g_\phi(q_i) \;=\; \left(p_i^{\mathrm{low}},\, p_i^{\mathrm{high}}\right), \tag{3}$$

where $p_i^{\mathrm{low}}, p_i^{\mathrm{high}} \in [0,1]$ are direction-specific scores rather than a normalized distribution. The low-priority score estimates how safe it is to compress instance i to the low-budget policy, for example when the answer appears anchored to local, static, single-moment, or explicit textual evidence. The high-priority score estimates how useful additional coverage may be, for example for counting, ordering, change tracking, broad temporal evidence, or fine visual discrimination across moments. For auditing, the same parser also returns auxiliary fields such as confidence, evidence-anchor category, optional route suggestion, and a short rationale; these fields are not consumed by the allocator.

The parser prompt and underlying LLM checkpoint, denoted by ϕ, are fixed before evaluation and are never updated using benchmark labels, policy outcomes, or category-level performance. The parser uses deterministic decoding and sees only q_i, including the multiple-choice options that are part of the task input. It does not see the video, subtitles, sibling questions, other videos, the correct answer label, or any candidate answers generated by the policy bank. Thus the priority pair is a question-only prior over evidence demand, not a temporal localization claim: FrameRouter never predicts timestamps, and all policies in Π remain full-video uniform samplers.

4.4 Quota-constrained allocation

Given the priority scores produced by g_ϕ in Eq. (3), FrameRouter must turn them into routes $\{r_i\}_{i=1}^N$ that satisfy the dataset-level mean-budget constraint $\frac{1}{N}\sum_{i=1}^N B_{r_i} = B_{\mathrm{ref}}$ exactly. We do this through a quota-constrained allocator that decouples *how many* questions are sent to each policy from *which* questions are sent there. The first choice is fixed by an operating-point quota under the budget constraint; the second is determined by two priority rankings.

Quota selection. Let $n_{\mathrm{low}}, n_{\mathrm{mid}}, n_{\mathrm{high}} \in \mathbb{Z}_{\geq 0}$ denote the number of instances assigned to each policy in Π. For a chosen operating point, FrameRouter fixes a feasible quota triple with $n_{\mathrm{low}} + n_{\mathrm{mid}} + n_{\mathrm{high}} = N$ and $(B_{\mathrm{low}} n_{\mathrm{low}} + B_{\mathrm{mid}} n_{\mathrm{mid}} + B_{\mathrm{high}} n_{\mathrm{high}})/N = B_{\mathrm{ref}}$. This encodes the mean-budget

constraint exactly at the dataset level rather than only in expectation. Specializing to the common setting $B_{\mathrm{ref}} = B_{\mathrm{mid}}$, this condition is equivalently

$$(B_{\mathrm{mid}} - B_{\mathrm{low}})n_{\mathrm{low}} = (B_{\mathrm{high}} - B_{\mathrm{mid}})n_{\mathrm{high}}, \tag{4}$$

with $n_{\mathrm{mid}} = N - n_{\mathrm{low}} - n_{\mathrm{high}} \geq 0$. Thus feasible quotas form an integer family that trades low-budget assignments against high-budget assignments while preserving the reference mean. The quota triple is fixed before evaluation and does not use labels, policy correctness, or category-level performance. It is selected as a pre-specified operating point under the budget equation rather than by test-set accuracy. Different feasible triples correspond to different routing operating points.

Deterministic assignment. Given $(n_{\mathrm{low}}, n_{\mathrm{mid}}, n_{\mathrm{high}})$, we assign routes with a high-first, low-second rule. Let $\mathrm{argTopK}^{m}_{i \in S}\, \tau_i$ denote the index set of the m largest scores in S, breaking ties deterministically with a fixed hash of the dataset index. FrameRouter first selects

$$H = \mathrm{argTopK}^{n_{\mathrm{high}}}_{i \in \{1,\ldots,N\}}\, p_i^{\mathrm{high}}, \tag{5}$$

and assigns $r_i = \mathrm{high}$ for $i \in H$. It then selects the low-budget set from the remaining instances,

$$L = \mathrm{argTopK}^{n_{\mathrm{low}}}_{i \in \{1,\ldots,N\} \setminus H}\, p_i^{\mathrm{low}}, \tag{6}$$

and assigns $r_i = \mathrm{low}$ for $i \in L$. All remaining instances are routed to the mid-budget policy. The high-first order is intentional: if an ambiguous question receives high scores for both the low- and high-policy priorities, the allocator treats the high-policy signal as the stronger constraint: wrongly compressing a high-demand question can remove needed evidence, whereas wrongly expanding a low-demand question mainly spends redundant frames. The construction guarantees, by the quota constraint, that $\frac{1}{N}\sum_{i=1}^{N} B_{r_i} = B_{\mathrm{ref}}$ holds exactly, so an accuracy gap against the uniform reference π_{ref} is not explained by a change in average compute, but by how that compute is distributed across questions. Inference then executes only the selected policy π_{r_i} for instance i; the other policies in Π are not run, so there is no late selection over multiple candidate answers. The parser adds one short text-only LLM call per question, which is separate from the video-frame budget.

5 Experiments

We evaluate FrameRouter on Video-MME-v2 and three additional video understanding benchmarks, always matching the mean frame budget of the corresponding uniform mid-budget baseline.

5.1 Setup

Benchmarks and metrics. We evaluate on four video understanding benchmarks: Video-MME-v2 [Fu et al., 2026], LongVideoBench [Wu et al., 2024], MVBench [Li et al., 2024], and VideoMMMU [Hu et al., 2025]. Video-MME-v2 provides the most detailed protocol for our main analysis: in addition to average accuracy, it reports official group-level scores for non-linear reasoning, three difficulty levels, consistency, and coherence, and we evaluate both subtitle and no-subtitle settings. LongVideoBench is also evaluated in both subtitle settings, while MVBench and VideoMMMU are evaluated with their official accuracy metrics. Unless otherwise noted, all reported numbers follow the corresponding benchmark evaluation protocol.

Baselines. Our controlled baseline is uniform mid-budget inference with the same Qwen3.5-397B-A17B backbone [Qwen Team, 2026]. We evaluate both *Instruct* and *Think* regimes, and constrain FrameRouter to the same average visual frame budget using the frozen 48/64/96 full-video policy bank. The parser is a deterministic text-only call to the same model used by the answer-stage VLM, which receives only question/options, processes no video frames, and is capped at 160 output tokens, so its overhead is negligible compared with the heavy multimodal answer call over 48/64/96 frames and up to 1024 generated tokens. For context, we also include frontier commercial systems such as Gemini-3-Pro [Google DeepMind, 2025], GPT-5 [OpenAI, 2026], and Kimi-K2.5 [Kimi Team, 2026], together with open-source video LMMs such as Qwen3-VL [Bai et al., 2025a], LLaVA-Video [Zhang et al., 2024], InternVL3.5 [Wang et al., 2025], and Qwen2.5-VL [Bai et al., 2025b].

Table 1: Main results on Video-MME-v2 under the same 64-frame mean-budget protocol in %. We report the group-level metrics and average accuracy with and without subtitles.

Model	Frames	Non-Lin		Level 1		Level 2		Level 3		Consistency		Coherence		Avg. Acc.	
		w. sub	wo sub	w. sub	wo sub	w. sub	wo sub	w. sub	wo sub	w. sub	wo sub	w. sub	wo sub	w. sub	wo sub
Commercialization Models															
Gemini-3-Pro	1fps	49.4	38.2	64.0	43.2	50.0	45.4	40.6	30.2	50.8	39.7	47.0	35.4	66.1	56.8
GPT-5	50	37.0	26.4	44.5	32.2	39.1	28.6	31.1	21.3	37.4	28.1	36.3	23.2	55.6	44.7
Kimi-K2.5	64	36.1	27.3	44.3	30.5	40.0	32.8	28.5	21.6	36.0	28.7	36.2	24.9	54.4	46.0
Open-Source Models (Thinking)															
MiMo-VL-7B-RL	64	19.4	11.9	24.9	16.1	18.8	12.5	16.6	9.0	18.9	12.5	20.3	10.8	36.0	27.0
GLM4.5V	64	21.8	13.3	26.2	14.9	21.7	14.7	19.4	11.4	20.5	14.7	24.3	10.6	39.2	29.6
Qwen3-VL-235B-A22B	64	26.3	15.9	32.0	19.1	27.5	16.8	22.3	13.4	26.0	17.2	27.0	13.4	44.9	33.3
Qwen3.5-397B-A17B	64	30.6	20.7	34.8	20.7	30.4	23.1	28.4	19.0	29.5	20.9	32.7	20.4	48.9	39.3
FrameRouter	**64**	**34.9**	**24.6**	**38.8**	**26.1**	**38.1**	**29.6**	**30.3**	**20.0**	**33.2**	**25.6**	**37.9**	**22.6**	**53.3**	**42.9**
Open-Source Models (Instruct)															
MiMo-VL-7B-SFT	64	14.7	10.7	19.8	14.3	13.2	9.6	12.8	9.3	14.1	11.2	15.8	9.7	31.2	24.6
LLaVA-Video-72B	64	17.2	11.3	21.8	14.8	14.8	11.1	16.3	9.5	17.1	12.3	17.5	9.5	34.4	27.3
Qwen2.5-VL-72B	64	22.1	13.7	25.8	15.3	21.2	13.7	20.6	12.6	20.9	14.1	24.3	12.9	38.9	30.3
InternVL3.5-241B-A28B	64	23.1	15.8	28.2	18.1	23.7	16.3	19.6	14.1	23.4	17.2	22.5	13.2	41.4	32.9
Qwen3-VL-235B-A22B	64	25.0	16.5	30.7	20.6	25.2	16.8	21.6	13.9	25.0	17.7	25.0	14.2	43.3	33.8
Qwen3.5-397B-A17B	64	24.5	16.9	27.3	16.8	25.9	18.1	21.9	16.1	23.9	18.2	25.7	14.6	42.1	35.0
FrameRouter	**64**	**25.7**	**18.5**	**30.6**	**21.3**	**27.2**	**19.2**	**21.8**	**16.3**	**26.2**	**19.6**	**24.9**	**16.4**	**44.8**	**36.8**

5.2 Main results

Video-MME-v2. Table 1 reports Video-MME-v2 results at the same 64-frame mean budget. The strongest controlled gain appears in the Think regime with subtitles: relative to the uniform Qwen3.5-397B-A17B row, FrameRouter improves average accuracy by 4.4% and raises every group-level metric. The largest gains concentrate on the harder diagnostic axes: Level 2 improves by 7.7%, Coherence by 5.2%, and Non-Lin by 4.3%. This routed Think setting is also the strongest open-source Think row in the table while using the same 64-frame mean budget. The no-subtitle and Instruct settings show the same overall direction, indicating that the benefit is not confined to a single input or decoding configuration.

Additional benchmarks. Table 2 gives the corresponding results on MVBench, VideoMMMU, and LongVideoBench. In the Think block, FrameRouter improves over the same Qwen3.5-397B-A17B uniform row on every reported benchmark, with gains ranging from roughly 1% on shorter or more saturated settings to nearly 3% on LongVideoBench. The routed LongVideoBench Think result with subtitles also exceeds the listed GPT-5 reference in Table 2. The Instruct block shows a similar positive pattern, with the clearest gains appearing on VideoMMMU and LongVideoBench. Across Table 1 and Table 2, the most consistent message is that fixed-budget routing transfers across benchmarks and is especially effective on long-video and reasoning-heavy settings.

5.3 Ablation study

We ablate the two ingredients most specific to FrameRouter: direction-specific priority scores and strict-budget quota allocation. All variants use the same Think-with-subtitles setting on Video-MME-v2 and keep the mean frame budget matched to the uniform baseline.

Score channels. Table 3 shows that both priority channels contribute useful routing signal. Using the full direction-specific parser improves average accuracy by 4.4% over uniform inference, with larger gains on the diagnostic metrics most tied to long-video reasoning: Coherence improves by 5.2% and Non-Lin by 4.3%. The single-channel variants also improve substantially, indicating that each score captures a meaningful part of evidence demand; combining them yields the most consistent profile across the reported metrics.

Table 2: Comparison across long-video and video understanding benchmarks in %. Missing or unreported entries are denoted by "-".

Model	MVBench		VideoMMMU		L.V.Bench (w. sub)		L.V.Bench (wo sub)	
	Frames	Acc.	Frames	Acc.	Frames	Acc.	Frames	Acc.
Commercialization Models								
Gemini-3-Pro	-	74.1	-	87.6	-	77.7	-	-
GPT-5	50	74.1	50	84.6	50	72.6	-	-
Kimi-K2.5	-	73.5	-	86.6	-	79.8	-	-
Open-Source Models (Thinking)								
GLM4.5V	-	73.0	-	72.4	64	68.8	-	-
MiMo-VL-7B-RL	-	-	256	53.1	-	-	-	-
Qwen3-VL-235B-A22B	-	75.2	-	80.0	-	65.6	-	-
Qwen3.5-397B-A17B	64	74.2	64	84.7	64	70.4	64	68.1
FrameRouter	**64**	**75.4**	**64**	**86.0**	**64**	**73.1**	**64**	**71.0**
Open-Source Models (Instruct)								
LLaVA-Video-72B	-	64.1	-	49.7	-	-	64	61.9
Qwen2.5-VL-72B	16	70.4	-	60.2	64	60.7	-	-
InternVL3.5-241B-A28B	16	76.5	-	-	64	67.1	-	-
Qwen3.5-397B-A17B	64	74.0	64	73.3	64	69.7	64	66.8
FrameRouter	**64**	**74.2**	**64**	**75.1**	**64**	**72.1**	**64**	**69.7**

Table 3: Score-channel ablation on Video-MME-v2 with the Think backbone and subtitles. All routed variants use the same route quota and mean frame budget; in single-channel variants, the enabled score supplies the ranking signal for both quota-constrained selections.

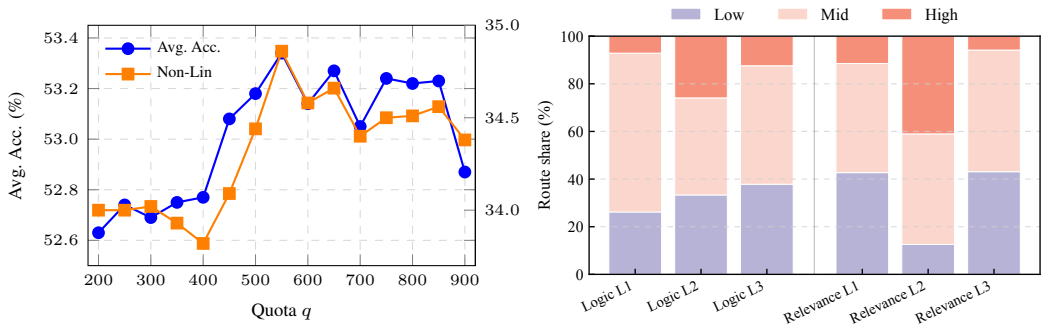
Method	p^{low}	p^{high}	Non-Lin	Consistency	Coherence	Avg. Acc.
Baseline	×	×	30.60	29.50	32.70	48.90
High-only	×	✓	33.94	33.10	35.47	53.03
Low-only	✓	×	34.64	32.86	37.91	52.66
Ours	✓	✓	**34.86**	**33.20**	**37.93**	**53.34**

Quota sensitivity. Figure 3(a) sweeps the high-route quota while preserving the same mean frame budget. The selected operating point lies inside a broad high-performing region rather than at an isolated spike, and nearby quotas change the reported metrics by only a few tenths of a percent. The two curves also show that the gain is not explained by simply assigning more questions to the high-budget policy; under a fixed mean budget, useful allocation requires a balanced redistribution between compressed and promoted questions.

5.4 Router behavior analysis

Route assignments in FrameRouter are determined before answer generation from the current question alone, without access to the video, subtitles, labels, or candidate-policy outputs. We therefore use route behavior only as a post-hoc diagnostic of whether the question-only parser induces structured budget allocation under the fixed mean budget.

Figure 3(b) shows a structured allocation pattern across the official group×level splits. The mid route remains the dominant fallback, while the low- and high-budget shares vary across question categories rather than following a uniform mixture. This supports the central premise of FrameRouter: even without observing video content or outcomes, question prompts contain enough evidence-demand signal to redistribute a fixed visual budget in a category-sensitive way.



(a) Quota sensitivity.

(b) Route distribution by group×level.

Figure 3: Allocator diagnostics on Video-MME-v2 Think with subtitles. Left: quota sensitivity under the matched mean budget. Right: realized route shares within each official group×level split.

6 Conclusion

This work studies whether a uniform per-question frame budget is the best use of a fixed average visual budget for long-video question answering. Our diagnostics on Video-MME-v2 show that nearby frame budgets help different question subsets, motivating fixed-mean frame-budget routing as a controlled test-time setting. FrameRouter instantiates this setting with a training-free question-only evidence-demand parser and a quota-constrained allocator over a frozen bank of full-video uniform policies. Across Video-MME-v2, LongVideoBench, MVBench, and VideoMMMU, the results show that matched-budget gains can be obtained by reallocating visual evidence across heterogeneous questions rather than increasing the average frame count. Budget routing is therefore a simple inference axis complementary to frame selection, token compression, and model scaling.

Acknowledgments and Disclosure of Funding

Funding. This work received no external funding.

Competing interests. The authors declare no competing interests.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL `https://arxiv.org/abs/2511.21631`.

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL `https://arxiv.org/abs/2502.13923`.

Dan Ben-Ami, Gabriele Serussi, Kobi Cohen, and Chaim Baskin. Himu: Hierarchical multimodal frame selection for long video question answering. *arXiv preprint arXiv:2603.18558*, 2026.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, et al. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding. *arXiv preprint arXiv:2604.05015*, 2026.

Google DeepMind. Gemini 3 pro model card. `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf`, 2025. Updated: November 18, 2025; Accessed: 2026-05-06.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13504–13514, 2024.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025.

Kimi Team. Kimi k2.5: Visual agentic intelligence, 2026. URL `https://arxiv.org/abs/2602.02276`.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

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

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

OpenAI. Openai gpt-5 system card, 2026. URL `https://arxiv.org/abs/2601.03267`.

Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryu, Donghyun Kim, and Michael S Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3569–3588, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

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

Weiyun Wang, Zhangwei Gao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL `https://arxiv.org/abs/2508.18265`.

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*, pages 58–76. Springer, 2024.

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*, 37:28828–28857, 2024.

Haoning Xi, Zhiqi Shao, David A Hensher, John D Nelson, Huaming Chen, and Kasun Wijayaratna. A multi-task transformer with mixture-of-experts for personalized periodic predictions of individual travel behavior in multimodal public transport. *Transportation Research Part C: Emerging Technologies*, 179:105287, 2025.

Linli Yao, Haoning Wu, Kun Ouyang, Yuanxing Zhang, Caiming Xiong, Bei Chen, Xu Sun, and Junnan Li. Generative frame sampler for long video understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17900–17917, 2025.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22056–22065, 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data, 2024. URL `https://arxiv.org/abs/2410.02713`.

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

A Technical appendix

A.1 Additional policy-bank corrections

Section 3.2 analyzes the Think setting and shows that the low-, mid-, and high-budget policies do not form nested error sets. Here we repeat the same diagnostic for the Instruct regime to verify that budget complementarity is not specific to chain-of-thought style inference. For each official group×level split, we compute pairwise correction matrices among the frozen full-video uniform policies. Rows denote the policy that answers a question correctly, columns denote the policy that misses it, and each mini-matrix is normalized within its split. Off-diagonal mass therefore indicates recoverable disagreement: questions that one budget handles while another budget fails.

This diagnostic is intentionally separated from the routed results. It does not use the parser, the allocator, or any learned route prediction; it only examines the frozen policy bank under different frame budgets. The Instruct setting shows that budget disagreement remains structured even without explicit reasoning traces, suggesting that frame-budget complementarity is a property of the question–evidence interaction rather than an artifact of a particular decoding style.

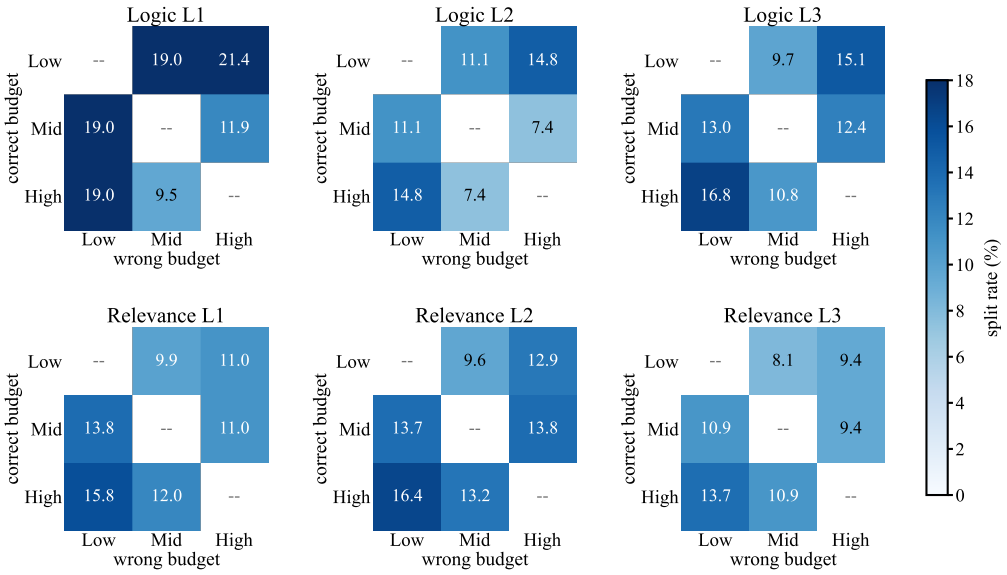


Figure 4: Pairwise correction matrices split by `group_type` and `level` in the Instruct setting. We show the labeled L1–L3 splits for logic and relevance questions; Video-MME-v2 logic questions with empty level fields are excluded from this diagnostic plot. Each mini-matrix is normalized within its split; rows denote the policy that is correct and columns denote the policy that is wrong. The off-diagonal mass varies across question groups and difficulty levels, supporting question-conditioned budget routing rather than a single monotone frame-scaling rule.

Figure 4 repeats the group×level pairwise correction analysis for the Instruct regime. The off-diagonal entries remain visibly non-uniform across groups and difficulty levels, meaning that the low-, mid-, and high-budget policies recover different subsets of questions rather than forming a simple nested ordering. This mirrors the Think-setting diagnostic in the main text and supports the same design choice: FrameRouter should route among frozen full-video budget policies instead of assuming that a single uniform budget is uniformly preferable for all question types.

A.2 Parser evidence-anchor profiles

Section 5.4 reports how the final routes distribute across official group×level splits. The diagnostic below looks at the same routed questions from the parser side. During routing, the question-only parser emits an evidence-anchor category together with its low- and high-budget priorities. These categories are not used as ground-truth annotations and are not meant to localize evidence in the video; they are self-reported parser fields that help interpret why different questions are assigned

to different budget routes. We aggregate these anchor categories within each final route to check whether the route assignments correspond to coherent evidence-demand profiles.

This analysis should be read as an interpretability check rather than as an additional source of supervision. The allocator consumes only the two priority scores described in Section 4.3; the evidence-anchor field is logged for auditing and is not used to choose the route. A meaningful anchor–route association is therefore useful because it shows that the parser priorities are not arbitrary scalar scores: they are accompanied by stable, human-readable evidence profiles that align with the intended roles of the low, mid, and high routes.

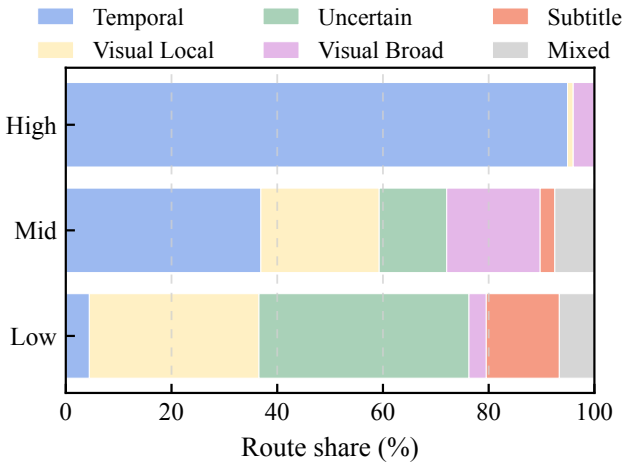


Figure 5: Parser evidence-anchor profiles within each route on Video-MME-v2 Think with subtitles. Routes are decided from the current question only, prior to answer generation. Evidence-anchor labels are parser outputs and are used only to interpret routing behavior.

Figure 5 gives a parser-side view of the realized routes. The high route is strongly enriched for temporal evidence-anchor outputs, while the low route contains more uncertain, local-visual, and subtitle-oriented profiles. The mid route remains more mixed, consistent with its role as the fallback route when neither compression nor promotion is strongly preferred. These anchors are not human annotations and do not imply temporal localization, but they help verify that the question-only parser induces interpretable route structure rather than an arbitrary budget mixture.

A.3 Implementation details

This section summarizes implementation details of our method. The default configuration matches the main Video-MME-v2 Think-with-subtitles setting unless otherwise specified.

Hyperparameters. FrameRouter uses a three-policy full-video uniform bank with low, mid, and high budgets set to 48, 64, and 96 frames, respectively. For the 48/64/96 bank this gives twice as many low routes as high routes. Parser and answer calls use temperature 0.0 by default. The parser response length is 160 tokens, while the answer response length is 1024 tokens.

Deterministic allocation and parsing. The parser returns `priority_48`, `priority_96`, `confidence`, `evidence_anchor`, and a short rationale. Scores are clamped to $[0, 1]$, and malformed or missing scores default to conservative mid-range values. The allocator first selects the high route by descending `priority_96`, then selects the low route from the remaining questions by descending `priority_48`; all remaining questions use the mid route. Ties are broken by a deterministic hash of the dataset index, and the code raises an error if the allocated mean frame count differs from the mid-budget target. The answer stage then runs only the selected route for each question, using uniform sampling and the baseline Video-MME-v2 prompt layout inherited from the evaluation script.

Evidence-demand priority parser prompt. The released parser prompt is presented below. The prompt explicitly states the information boundary of the router, the fixed-mean budget setting, the semantics of the two priority scores, and the required JSON output format.

```
You are a text-only priority router for long-video multiple-choice QA.

You cannot see the video and you do not know the answer. Use only the current
question and answer options. Do not answer the question. Do not use labels,
sibling questions, other videos, or dataset statistics.

A strict solver will keep the average budget fixed at 64 frames:
- a small top group by priority_96 will use 96 frames;
- twice as many questions by priority_48 will use 48 frames;
- all remaining questions will use the default 64 frames.

Your job is to output the two final ranking priorities directly. Do not output a
weighted formula; do not choose the number of questions assigned to each budget.

priority_48 should be high only when reducing 64f to 48f is likely safe:
- the decisive evidence is local, static, repeated, or easy to locate;
- 64f looks more than sufficient from the question/options;
- the question is anchored by a clear subtitle/dialogue/text/object cue;
- no broad temporal coverage, counting, order tracking, or subtle multi-scene
  comparison is needed.

priority_96 should be high only when extra frames are likely useful:
- subtitle/dialogue/text anchoring is weak or absent;
- the decisive visual evidence is hard to localize from text alone;
- broad temporal coverage, counting, event order, changes, or repeated actions
  are central;
- the options require fine visual discrimination across moments.

Important calibration:
- Most questions should remain 64f, so both priorities should often be modest.
- Do not make priority_96 high merely because the question seems hard; extra
  frames only help when the missing evidence is temporal/visual coverage.
- Do not make priority_48 high for OCR, tiny text, rare brief details, or fine
  visual distinctions even if the evidence is local.
- If unsure, set both priorities near 0.40-0.55 and confidence below 0.60.

Return compact JSON only:
{
"policy": "DEFAULT_64F",
"priority_48": 0.00,
"priority_96": 0.00,
"confidence": 0.00,
"evidence_anchor": "subtitle|visual_local|visual_broad|temporal|mixed|uncertain",
"rationale": "<=10 words"
}

Question:
{question}

Options:
{options}
```

Answer-stage prompts. The answer stage uses the baseline Video-MME-v2 prompt layout from the official evaluation script.

A.4 Code release

The supplementary material includes a code implementation of FrameRouter. The code is organized to match the method contract in Section 4.2: the router first reads only the current question and its

answer options, assigns one frame budget, and then runs exactly one selected full-video uniform policy. It does not inspect video frames, subtitles, labels, sibling questions, other videos, or any candidate-policy outputs before route assignment. The released pipeline therefore implements the deployable path used in the paper rather than a post-hoc selector that runs multiple answer policies and chooses among their outputs.

Module structure. The entry point is `run_framerouter.py`, with command-line arguments defined in `framerouter/cli.py`. Runtime and frame-budget settings are kept in `framerouter/config.py`. The question-only routing prompt is in `framerouter/router_prompt.py`, while `framerouter/router.py` performs the parser call, JSON extraction, and parser TSV normalization. The deterministic fixed-mean budget assignment is implemented in `framerouter/allocation.py`. After allocation, `framerouter/policy.py` runs the selected 48/64/96-frame Video-MME-v2 inference policy, and `framerouter/scoring.py` wraps Video-MME-v2 scoring. The end-to-end orchestration lives in `framerouter/pipeline.py`; `run_videommev2.sh` provides an example command for the main Video-MME-v2 setting.

```
current question/options
  -> framerouter.router       # evidence-demand scores
  -> framerouter.allocation   # fixed-mean budget assignment
  -> framerouter.policy       # selected full-video policy only
  -> framerouter.scoring      # Video-MME-v2 metrics
```

Environment and execution. The paper experiments were run on an 8×NVIDIA H200 server using vLLM with the OpenAI-compatible chat-completions API and a Python 3.10+ client. The inputs are Video-MME-v2 parquet metadata, extracted video files, and subtitle JSONL files. Frames may be transported as base64-encoded JPEGs or through local media paths supported by the inference server. The implementation is otherwise model-server agnostic as long as the server accepts OpenAI-compatible multimodal chat-completions requests.

A typical run first starts a vLLM server:

```
python3 -m vllm.entrypoints.openai.api_server \
  --model <MODEL_PATH_OR_NAME> \
  --served-model-name qwen35-think \
  --tensor-parallel-size 8 \
  --port 8000 \
  --trust-remote-code
```

FrameRouter can then be launched with the provided script:

```
bash run_videommev2.sh
```

For quick verification, the release also supports a small smoke test via `-limit` and an allocation-only mode via `-only-allocate` when an existing parser TSV is available.

Outputs. The script writes `parser.tsv` for question-only parser scores, `routes.tsv` for route assignments and allocated frame counts, `predictions.tsv` for selected-policy predictions and raw responses, `predictions_scored.tsv` for scored question-level outputs, and `predictions_rating.json` for aggregate Video-MME-v2 metrics.

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 fixed-mean frame-budget routing problem, the FrameRouter method, and the matched-budget empirical claims. The Analysis, Method, and Experiments sections then support these claims with diagnostic budget curves, a formal routing setup, Video-MME-v2 results, ablations, and cross-benchmark evaluations.

   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 states the main scope constraints of the work: FrameRouter routes among frozen full-video uniform policies rather than selecting timestamps, uses only the current question as the routing signal, enforces a dataset-level mean-budget constraint, and evaluates specific operating points and benchmarks.

   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 formalizes the budgeted predictor, fixed-mean constraint, frozen policy bank, and quota-constrained allocator, but it does not claim theoretical theorems or formal proof results.

   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 specifies the evaluated benchmarks, same-backbone baselines, Think/Instruct settings, frame-budget policies, fixed-mean routing constraint, evaluation metrics, ablation variants, and route-analysis protocol needed to reproduce the main empirical 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: [Yes].

   Justification: The anonymized supplementary package is intended to provide the FrameRouter implementation, routing configuration, evaluation scripts, and instructions needed to reproduce the main routed and uniform-budget comparisons.

   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 Method and Experiments sections specify the test-time setting, parser inputs, policy bank, route quotas, benchmark choices, baselines, inference regimes, frame budgets, and reported metrics needed to interpret the results.

   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: [Yes].

   Justification: The paper evaluates on full benchmark splits where applicable and provides matched-budget controls, score-channel ablations, quota-sensitivity sweeps, and route-behavior diagnostics to support the robustness of the main empirical findings.

Guidelines:

- The answer [N/A] 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: The paper characterizes compute primarily through the number of video frames processed per question, the matched mean-frame budget, and the additional text-only parser call; implementation details in the supplementary package document the resources needed to reproduce the experiments.

   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 work evaluates a test-time inference method on existing video QA benchmarks, does not collect private or human-subject data, and restricts the router from using labels, sibling answers, videos, subtitles, or policy outcomes as routing signals.

   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: The paper frames the positive impact as more efficient long-video understanding under a fixed average frame budget, while its stated scope constraints and diagnostics clarify that the method is an inference-time budget-routing study rather than a deployed decision system.

    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 work does not release a new high-risk pretrained model, scraped dataset, or generative system; released materials are limited to research code, routing configurations, and evaluation utilities.

    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: [Yes].

Justification: The paper cites the existing benchmarks, models, and related methods used for evaluation and comparison; the supplementary package should retain the corresponding attribution and usage information for reproduced assets.

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: [Yes].

Justification: The new asset is the FrameRouter research code and configuration package, whose main components are documented in the paper through the parser, policy-bank, and quota-allocation definitions and in the supplementary reproduction instructions.

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 work uses existing benchmark data and automated evaluation protocols only; it does not involve new crowdsourcing, user studies, or experiments with human subjects.

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 work does not involve human-subject studies requiring IRB approval or equivalent review.

    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: LLM and multimodal model usage is central to the method and evaluation, including the Evidence-Demand Priority Parser and the final VQA inference.

    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.