SAVE: LET VLMS ALLOCATE THEIR OWN VIEWING BUDGETS FOR LONG-VIDEO FRAME SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video understanding under a limited visual budget requires selecting a small set of frames that captures question-relevant evidence. Existing query-aware selectors often rely on external models to score frame relevance. Balancing relevance with temporal coverage remains challenging: prioritizing high-scoring frames can yield redundant views of a narrow interval, whereas broad coverage can allocate scarce frames to irrelevant content. We propose SAVE (Self-guided Allocation of Visual budgets via Evidential demand), a VLM-native framework for visual budget allocation. We use the same frozen vision-language model (VLM) for evidence scoring and final question answering, converting frame-level relevance scores into a semantic demand distribution over time. Given a fixed frame budget, we formulate selection as an optimal transport problem that minimizes the demand-weighted squared temporal distance from each candidate frame to its nearest selected frame. This objective couples relevance and temporal coverage, adaptively allocating the visual budget without a separate balancing coefficient. Exploiting the one-dimensional temporal structure, we solve this objective exactly and efficiently via dynamic programming, guaranteeing a globally optimal allocation under the proposed objective. Experiments across multiple long-video benchmarks and diverse VLM architectures show consistent improvements over the evaluated baselines in benchmark-level average performance.

1 INTRODUCTION

Allocating a limited visual budget is a central challenge in long-video understanding. A video may contain tens of thousands of frames, yet a vision–language model (VLM) can accommodate only a small visual context in its final answering call. This extreme capacity gap is especially consequential on modern benchmarks, where success may require finding a sparse detail, connecting events far apart in time, or following an extended process or narrative (Fu et al., 2025; Zhou et al., 2025; Wu et al., 2024). Uniformly sampling more frames eventually runs into context and compute limits. The practical question is therefore not merely how many frames a VLM can process, but *which M frames can most effectively utilize this strictly limited visual budget*.

Recent training-free selectors combine question relevance with diversity, segmentation, or exploration (Tang et al., 2025; Zhu et al., 2026; Chen et al., 2026). Two limitations motivate SAVE. First, many methods score relevance with an auxiliary image–text encoder, whose semantic similarity need not reflect the downstream answerer's usable evidence. Second, fixed segments, diversity penalties, exploration schedules, or manually weighted relevance–coverage objectives do not directly specify how much visual budget each interval deserves. This trade-off is difficult because question-relevant evidence can be unevenly distributed across time: several high-scoring frames may depict the same event, while complementary evidence may occur in distant intervals. Prioritizing relevance alone can therefore exhaust the budget on redundant observations and miss evidence needed to connect events, whereas enforcing broad temporal coverage can allocate frames to irrelevant intervals. The challenge is to determine how much of the limited budget each interval deserves while accounting for both its evidential value and its temporal relation to the rest of the video. We address both limitations with **SAVE**: **S**elf-guided **A**llocation of **V**isual budgets via **E**vidential demand. Instead of relying on external locators, we use the frozen answering VLM's own confidence to assess the question relevance of candidate frames. Specifically, we derive question-relevance log-odds from the VLM's output logits, yielding a question-conditioned evidence signal across the video, which we

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

term *Q-SelfYes*. Normalizing these odds across candidate frames yields a *semantic demand* distribution over time, assigning greater weight to frames judged more useful for answering the question. Building on VLM-based frame assessment (Yu et al., 2023; Zou et al., 2026), SAVE couples this frozen-model signal with globally optimized temporal budget allocation.

Given B candidate frames with normalized evidence demand weights p_i at times t_i, SAVE selects $M_{\text{eff}} = \min(M, B)$ frames to minimize

$$\mathcal{S}^\star = \arg \min_{\mathcal{S} \subseteq [B],\, |\mathcal{S}| = M_{\text{eff}}} \sum_{i=1}^{B} p_i \min_{j \in \mathcal{S}} (t_i - t_j)^2. \tag{1}$$

All candidate frames contribute to the objective through their demand weights. High-demand frames incur larger costs when poorly represented, while the squared temporal distance penalizes leaving distant demand far from selected frames. Together, these properties couple relevance and temporal coverage without a separate weighting coefficient. The one-dimensional temporal structure enables exact optimization by dynamic programming, yielding a globally optimal allocation under this objective.

Our contributions are threefold:

- **Principled visual budget allocation.** We formulate visual budget allocation as demand-weighted temporal quantization with an optimal transport interpretation, and solve it exactly using classical one-dimensional dynamic programming.
- **Answerer-aligned evidence scoring.** We introduce Q-SelfYes, which reuses the frozen answering VLM to estimate question-relevant evidence without an auxiliary scoring model. Its grouped Yes/No log-odds are converted into semantic demand, linking frame assessment to budget allocation without additional training.
- **Extensive empirical validation.** Experiments across multiple long-video understanding benchmarks and diverse VLM architectures show consistent improvements over the evaluated baselines in benchmark-level average performance.

2 Related Work

Training-free frame selection. AKS balances query relevance and coverage (Tang et al., 2025); MDP^3 combines query similarity, determinantal diversity, segmentation, and dynamic programming (Sun et al., 2025). Nar-KFC selects relevant, diverse keyframes (our KFC baseline) before interleaving textual narratives (Fang et al., 2026). FOCUS uses bandit-based exploration and exploitation (Zhu et al., 2026), while WFS-SB combines wavelet denoising, boundary detection, clip-budget allocation, and maximal marginal relevance (Chen et al., 2026). SAVE instead derives temporal budget allocation from question-conditioned demand through a single objective.

VLM-based frame assessment. SeViLA ranks frames using a BLIP-2 Localizer's relevance judgments and separately fine-tunes Localizer and Answerer adapters through self-chaining (Yu et al., 2023). A.I.R. reuses the same VLM for frame analysis and answering without training, combining CLIP-based initialization, iterative reasoning, local resampling, and early stopping (Zou et al., 2026). SAVE converts the frozen answerer's grouped Yes/No log-odds into temporal demand and globally allocates a fixed frame budget by minimizing demand-weighted temporal distortion, without separately tuned localization parameters or iterative reasoning-based frame validation.

Optimal transport and ordered allocation. Optimal transport matches probability mass under a ground cost (Peyré & Cuturi, 2019). With squared temporal distance and optimized target weights, our objective is discrete weighted squared-error quantization: selected frames represent the source demand assigned to them (Graf & Luschgy, 2000). Ordered one-dimensional clustering admits exact dynamic programming because optimal clusters are contiguous (Wang & Song, 2011). SAVE applies this classical structure to question-conditioned demand, optimizing both selected frames and their weights under mass conservation rather than fixing each target weight to $1/M$. Thus, "free-mass" denotes optimized target weights, not relaxed mass conservation.

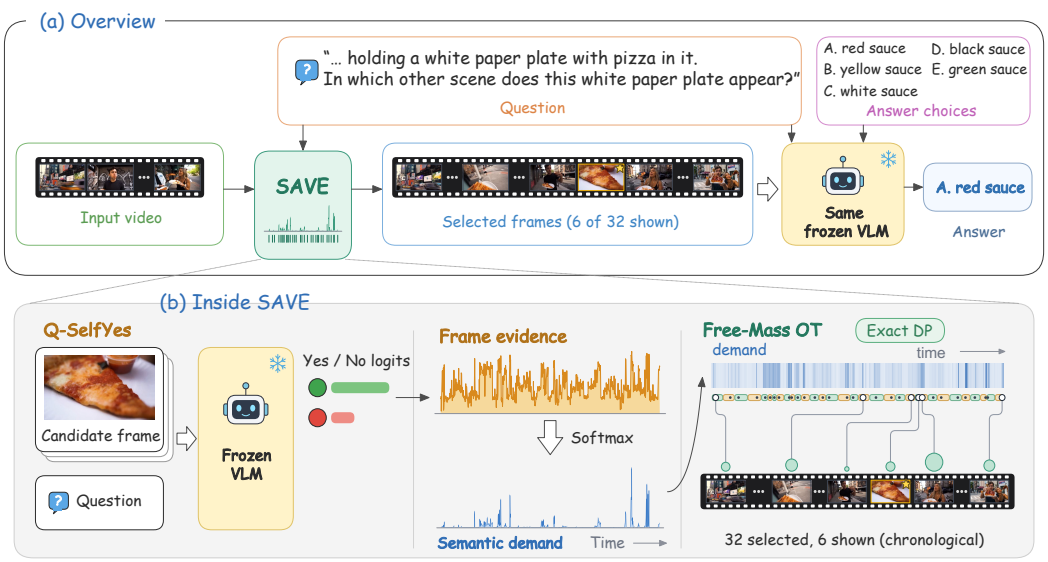


Figure 1: Overview of SAVE. (a) SAVE selects question-relevant frames and passes them, together with the original question and answer choices, to the same frozen VLM. (b) Q-SelfYes converts frame-level Yes/No log-odds into semantic demand via temporal Softmax. Free-mass temporal allocation selects frames by minimizing demand-weighted squared temporal distance, solved exactly by dynamic programming. Answer choices are used only for final answering.

3 Method

3.1 Problem Formulation and Overview

Let V be a video, q its original question, $\mathcal{O}$ the answer choices, F_θ a frozen VLM answerer, and M the maximum number of frames in the final answering call. Candidate scoring is a separate stage and may inspect more than M frames. We first construct a temporally ordered candidate stream $\mathcal{C} = \{(I_i, t_i)\}_{i=1}^{B}$, where I_i is a candidate frame and

$$t_i = \frac{i-1}{\max(B-1, 1)} \tag{2}$$

is its normalized position in that stream. Our goal is to select an index set $\mathcal{S} \subseteq [B]$ with $|\mathcal{S}| = M_{\text{eff}}$, where $M_{\text{eff}} = \min(M, B)$, before invoking the final answerer. Selection has no access to labels and uses question-only text; the answer choices are introduced only during final question answering.

Figure 1 summarizes SAVE. Q-SelfYes uses the frozen answering VLM to score candidate frames conditioned on the question. Temporal Softmax converts the resulting Yes/No log-odds into semantic demand. An exact dynamic program selects M_{eff} frames that minimize demand-weighted temporal distortion. The same VLM receives these frames in chronological order and answers the original question with its answer choices. Selection uses no answer choices, and all model parameters remain frozen.

3.2 Ordered Candidate Stream

We use a temporally ordered candidate stream, sampled at approximately 1 fps by default. The candidate count B varies with video duration; streams shorter than M are retained without duplicate padding. Model-scaling and deployment experiments instead use a fixed grid of 128 source frames; the candidate-density study varies the grid budget over $B \in \{64, 128, 256, 512\}$. Sampling details are given in Appendix D.

3.3 Q-SelfYes: Answerer-Aligned Frame Evidence

Q-SelfYes asks the frozen answering VLM whether a candidate frame provides evidence for the original question, without answer choices or timestamps. Let $\ell_i(v)$ be the next-token logit for token v, and let $\mathcal{Y}$ and $\mathcal{N}$ group Yes and No token variants. We define

$$s_i = \mathrm{LSE}_{v\in\mathcal{Y}}\, \ell_i(v) - \mathrm{LSE}_{v\in\mathcal{N}}\, \ell_i(v), \tag{3}$$

where LSE denotes log-sum-exp. Hence s_i is a grouped Yes–No logit margin, not a thresholded binary decision or sigmoid probability. A large value means that the VLM considers I_i more likely to contain visual evidence useful for q. Equation 3 requires a scoring forward pass rather than autoregressive answer generation, and the independent frame–question pairs can be evaluated in batches.

Throughout our Q-SelfYes experiments, the same frozen VLM supplies the relevance scores and produces the final answer. Each answer backbone therefore scores its own candidate frames, directly aligning frame relevance with the downstream representation.

3.4 From Evidence Scores to Semantic Demand

Q-SelfYes scores may exhibit a question-dependent additive offset. Since frame allocation is a within-question decision, we normalize all B scores jointly across time:

$$p_i = \frac{\exp((s_i - \max_j s_j)/\tau)}{\sum_{u=1}^{B} \exp((s_u - \max_j s_j)/\tau)}, \qquad p_i > 0, \quad \sum_{i=1}^{B} p_i = 1. \tag{4}$$

We use the fixed temperature $\tau = 1$ throughout. Numerical details are given in Appendix D. Equation 4 makes candidate frames compete for one unit of semantic-demand mass and is invariant to a common additive shift of all frame scores s_i for a question. We emphasize that p_i is not the probability that frame i must be selected, nor do we sample frames independently from this distribution. Instead, it describes how much relative visual demand is located near each time point: high-mass intervals should receive a larger share of the viewing budget, while the temporal allocator decides which observations can jointly serve nearby demand. Because s_i is a log-odds, Equation 4 with $\tau = 1$ normalizes the grouped Yes/No odds e^{s_i} across frames: frames compete on their relative odds, and the resulting demand is sharper than normalizing the per-frame Yes-probabilities $\sigma(s_i)$, which saturate as positive margins grow and can retain more relative mass on lower-scoring frames. Table 3 compares the two choices.

3.5 Free-Mass Temporal Allocation

Given the demand p over the ordered candidate times $t_1 < \cdots < t_B$, the allocator must choose M_{eff} observations that serve this demand. We treat the selected frames as a sparse support and charge every unit of demand the squared temporal distance to its closest selected frame:

$$\mathcal{S}^\star = \arg\min_{\mathcal{S}\subseteq[B],\, |\mathcal{S}|=M_{\text{eff}}} J(\mathcal{S}), \qquad J(\mathcal{S}) = \sum_{i=1}^{B} p_i \min_{j\in\mathcal{S}} (t_i - t_j)^2. \tag{5}$$

Equation 5 restates Equation 1. We call the resulting selector the *free-mass OT allocator* (abbreviated OT in tables and ablations), for the reason made precise next.

Optimal-transport interpretation. Let $\mu = \sum_{i=1}^{B} p_i\, \delta_{t_i}$ be the semantic-demand measure. For a support $\mathcal{S}$ and target weights w in the probability simplex over $\mathcal{S}$, write $\nu_{\mathcal{S},w} = \sum_{j\in\mathcal{S}} w_j\, \delta_{t_j}$, and let $W_2^2(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} \sum_{i,j} \pi_{ij}(t_i - t_j)^2$ denote the squared 2-Wasserstein distance over couplings $\Pi(\mu,\nu)$ with marginals μ and ν (Peyré & Cuturi, 2019). Let $\mathrm{nn}_{\mathcal{S}}(i) \in \arg\min_{j\in\mathcal{S}} |t_i - t_j|$ denote the selected frame nearest to candidate i, with ties broken toward the earlier frame.

Proposition 1 (Free-mass transport equals nearest-frame quantization)**.** *For every non-empty $\mathcal{S} \subseteq [B]$,*

$$\min_{w} W_2^2\big(\mu, \nu_{\mathcal{S},w}\big) = J(\mathcal{S}), \tag{6}$$

Algorithm 1 Exact temporal allocation in SAVE

Require: Ordered candidate times $t_{1:B}$, demand $p_{1:B}$, frame budget M
Ensure: Selected indices $\mathcal{S}^\star$ in chronological order
1: **if** $B \le M$ **then**
2: **return** $\{1, \dots, B\}$
3: **end if**
4: Compute prefix sums of p_i, $p_i t_i$, and $p_i t_i^2$.
5: For any segment $[a, b]$, let $j^\star(a, b)$ be its candidate nearest the weighted centroid and evaluate $c(a, b)$ using Equation 10.
6: Initialize $D_1(b) \leftarrow c(1, b)$ for $b = 1, \dots, B$.
7: **for** $k = 2, \dots, M$ **do**
8: **for** $b = k, \dots, B$ **do**
9: Store the minimizing boundary $A_k(b)$ and cost:

$$D_k(b) = \min_{k \le a \le b} \left[D_{k-1}(a-1) + c(a, b) \right]. \tag{7}$$

10: **end for**
11: **end for**
12: Backtrack from (M, B) through $A_k(b)$ to recover segments $[a_k, b_k]$.
13: **return** $\mathcal{S}^\star = \{j^\star(a_k, b_k)\}_{k=1}^{M}$, with $J(\mathcal{S}^\star) = D_M(B)$.

and the minimum is attained by the nearest-frame plan $\pi_{ij} = p_i \, \mathbf{1}[j = \mathrm{nn}_{\mathcal{S}}(i)]$, whose induced target weights are $w_j = \sum_{i:\, \mathrm{nn}_{\mathcal{S}}(i) = j} p_i$.

The proof is given in Appendix A.

Hence Equation 5 is the squared-Wasserstein projection of the demand onto M_{eff}-atom measures supported on candidate times, where the atom masses are *free* and induced by the optimal assignment. An equal-target-mass formulation would instead fix $w_j \equiv 1/M_{\text{eff}}$. Our formulation optimizes these weights while retaining standard mass-conserving OT for each given target distribution; w_j is the total demand assigned to frame j. In the language of quantization, $\mathcal{S}^\star$ is an optimal M_{eff}-point squared-error quantizer of μ with codewords restricted to the candidate grid (Graf & Luschgy, 2000).

All source demand mass contributes to the allocation objective. Temporal proximity serves as a proxy for evidence substitutability, which may fail across shot boundaries or for brief visual events. Thus, the exact solution minimizes demand-weighted temporal distortion; it does not guarantee preservation of all visual evidence or optimal QA accuracy.

The allocation depends on the relative demand profile and is invariant to a common positive rescaling of times or masses. Approximately flat demand yields approximately uniform spacing; concentrated demand attracts more representatives. Appendix A details these properties and their limits.

3.6 Exact Solution by Dynamic Programming

Ordered time makes the optimal allocation a partition into contiguous segments (Wang & Song, 2011). Within each segment, the representative is the candidate nearest its demand-weighted centroid. Algorithm 1 computes this partition and recovers the selected frames. Prefix sums make each segment cost $c(a, b)$ available in $O(1)$ on our evenly spaced grid; the proof, tie rules, and solver complexity are given in Appendix A.

3.7 Answering with the Selected View

Let $\mathcal{S}^\star = \{j_1 < \cdots < j_{M_{\text{eff}}}\}$ denote the indices returned by the temporal allocator. The frozen answerer receives the corresponding frames in chronological order together with the original question and complete option set:

$$\widehat{a} = F_\theta\Big(I_{j_1}, \dots, I_{j_{M_{\text{eff}}}}; q, \mathcal{O}\Big). \tag{8}$$

Table 1: Comparison of frame-selection methods across three long-video benchmarks and four answer backbones.

Backbone	Method	Video-MME				LongVideoBench				MLVU	Overall
		Short	Medium	Long	Avg.	Short	Medium	Long	Avg.	M-Avg	3-Avg
LLaVA-OV	Uniform	70.89	55.22	50.44	58.85	71.75	52.14	51.81	57.37	65.27	60.50
	KFC	71.00	57.33	49.22	59.19	71.75	56.57	51.45	59.61	69.99	62.93
	AKS	71.67	57.00	50.78	59.81	71.19	56.71	55.43	60.36	66.60	62.26
	FOCUS	67.11	56.67	49.67	57.81	67.87	56.86	60.14	60.51	65.35	61.22
	WFS-SB	71.56	56.11	49.44	59.04	70.64	56.43	54.35	59.84	67.40	62.09
	SAVE (ours)	71.33	59.89	50.89	**60.70**	70.91	56.86	59.06	**61.11**	**70.13**	**63.98**
Qwen2.5-VL	Uniform	72.33	58.44	52.00	60.93	72.02	52.00	52.54	57.52	62.70	60.38
	KFC	73.00	61.33	50.78	61.70	66.76	55.86	55.07	58.64	68.26	62.87
	AKS	70.56	57.22	50.44	59.41	65.65	52.71	53.99	56.47	63.87	59.92
	FOCUS	69.00	59.67	51.56	60.07	68.42	58.71	56.88	60.96	65.30	62.11
	WFS-SB	69.89	56.78	49.56	58.74	66.20	55.86	54.35	58.34	65.36	60.81
	SAVE (ours)	74.56	61.67	52.00	**62.74**	73.96	56.86	55.43	**61.18**	**68.75**	**64.22**
InternVL3	Uniform	75.89	64.78	56.00	65.56	75.07	56.14	52.54	60.51	69.55	65.21
	KFC	74.89	65.11	55.56	65.19	73.13	59.86	56.88	62.83	73.70	67.24
	AKS	75.56	66.56	56.22	66.11	72.85	60.00	56.16	62.68	71.98	66.92
	FOCUS	72.33	61.56	56.11	63.33	69.25	59.57	59.42	62.15	68.96	64.81
	WFS-SB	77.67	64.22	55.22	65.70	70.64	60.29	59.42	62.90	71.85	66.82
	SAVE (ours)	77.11	68.11	56.44	**67.22**	75.35	60.71	57.97	**64.10**	**74.02**	**68.45**
Qwen3.5	Uniform	77.22	63.22	55.89	65.44	72.30	57.29	53.62	60.58	68.36	64.80
	KFC	76.56	63.78	56.67	65.67	73.68	62.14	60.51	64.92	74.90	68.50
	AKS	77.89	65.33	56.78	66.67	72.85	60.71	61.23	64.10	72.53	67.76
	FOCUS	72.11	61.11	56.00	63.07	69.53	61.86	63.77	64.32	71.08	66.16
	WFS-SB	77.22	64.00	55.78	65.67	73.96	61.29	63.41	65.15	74.28	68.36
	SAVE (ours)	80.00	67.00	56.89	**67.96**	72.85	63.86	59.78	**65.45**	**75.80**	**69.74**

We use deterministic decoding and return a single option letter. The Q-SelfYes prompt and scores are not exposed to the final call. Consequently, SAVE changes only which visual observations enter the model; it leaves the original question, choices, answering prompt, and VLM parameters unchanged.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and models. We evaluate zero-shot video question answering on Video-MME (2,700 questions) (Fu et al., 2025), the seven multiple-choice tasks of MLVU (2,174 questions) (Zhou et al., 2025), and LongVideoBench validation (1,337 questions) (Wu et al., 2024). We report question accuracy for Video-MME and LongVideoBench, task-macro accuracy for MLVU, and their equal-weight mean, 3-Avg. The four frozen answer backbones are LLaVA-OneVision-Qwen2-7B (Li et al., 2024), Qwen2.5-VL-7B-Instruct (Bai et al., 2025), InternVL3-8B (Zhu et al., 2025), and Qwen3.5-4B (Qwen Team, 2026).

Comparison settings. We compare SAVE with Uniform, AKS (Tang et al., 2025), KFC (Fang et al., 2026), FOCUS (Zhu et al., 2026), and WFS-SB (Chen et al., 2026) on the same benchmark splits and answering pipeline. The main setting uses 1-fps candidates and at most 32 unique answering frames; FOCUS retains its native adaptive candidate access. We fix the answering-frame budget at 32 for consistent comparison across methods; applications can choose the budget according to their requirements. Each SAVE backbone scores its own frames using question-only text and Softmax temperature $\tau = 1$. Answer choices are supplied only for final answering, with deterministic decoding. Video-MME and LongVideoBench use no audio or subtitles. Model scaling and deployment use the fixed $B = 128$, $M = 32$ setting; budget ablations specify their changes alongside the results. Appendix D provides sampling, prompts, baseline implementations, and metric definitions.

Table 2: Downstream QA accuracy with different relevance scorers under the same Softmax normalization and free-mass OT allocator. Uniform is a score-independent reference.

Relevance scorer	Video-MME	LVB	MLVU	3-Avg
No relevance (Uniform)	65.44	60.58	68.36	64.80
BLIP-ITM	67.56	65.30	74.04	68.96
CLIP	66.22	65.37	74.76	68.78
SigLIP	67.30	**65.82**	74.75	69.29
SigLIP2	66.44	63.43	73.17	67.68
BLIP2-ITM	67.22	65.30	73.64	68.72
Q-SelfYes (ours)	**67.96**	65.45	**75.80**	**69.74**

Table 3: TopK and allocation ablations with a shared frozen Q-SelfYes score cache, $M = 32$, and the same answering configuration. Uniform does not use relevance scores.

Method	Evidence	Softmax	OT	Video-MME	LVB	MLVU	3-Avg
Uniform	No	No	No	65.44	60.58	68.36	64.80
TopK	Yes	—	No	64.81	63.65	73.85	67.44
w/o Softmax + greedy	Yes	No	No	66.70	63.58	73.16	67.81
Softmax + greedy	Yes	Yes	No	67.00	62.83	73.49	67.77
w/o Softmax + OT	Yes	No	Yes	67.78	65.07	74.99	69.28
SAVE (ours)	Yes	Yes	Yes	**67.96**	**65.45**	**75.80**	**69.74**

4.2 Main Results and Cross-Model Generalization

Overall comparison. Under a common answering budget of at most 32 frames, SAVE achieves the highest benchmark-level averages and 3-Avg across all four backbones (Table 1). Its 3-Avg exceeds the strongest reproduced baseline for each backbone by 1.05–1.35 percentage points. With Qwen3.5, SAVE reaches 69.74, outperforming Uniform by 4.94 percentage points and KFC, the strongest reproduced baseline for this backbone, by 1.24. Each SAVE configuration uses its own frozen answer backbone for both scoring and answering. These gains show that SAVE makes more effective use of the fixed answering-frame budget across the evaluated model architectures.

4.3 Understanding the Source of the Gains

We examine relevance scoring and temporal allocation separately with Qwen3.5-4B, full-1fps candidates, and $M = 32$.

Relevance scoring. Table 2 examines whether the frozen answerer can provide useful relevance scores without an auxiliary scoring model. We fix Softmax normalization and OT allocation and compare Q-SelfYes with five external scorers using downstream QA accuracy; Uniform is a score-independent reference. Reusing the answerer achieves performance competitive with the strongest evaluated external scorers, with the highest three-benchmark average of 69.74, compared with 69.29 for SigLIP. These results show that the answerer's own relevance signal can support competitive frame selection without requiring an additional scoring model.

Demand normalization and allocation. Does relevance ranking alone suffice, and how do demand normalization and temporal allocation affect performance? Table 3 holds the frozen Q-SelfYes scores, candidate pool, frame budget, and answering configuration fixed. TopK selects the highest-scoring frames without accounting for temporal coverage. To examine demand construction and allocation, we cross two demand transformations—normalized Yes-probabilities and Softmax over log-odds—with two allocation methods: greedy Gaussian coverage and exact OT. Uniform provides a score-independent reference. Appendix E specifies the transformations and selection rules.

OT consistently improves temporal allocation. Holding demand normalization fixed, replacing greedy Gaussian coverage with OT improves 3-Avg by 1.47 percentage points with normalized Yes-probabilities (67.81 to 69.28) and by 1.97 with Softmax demand (67.77 to 69.74). Both comparisons

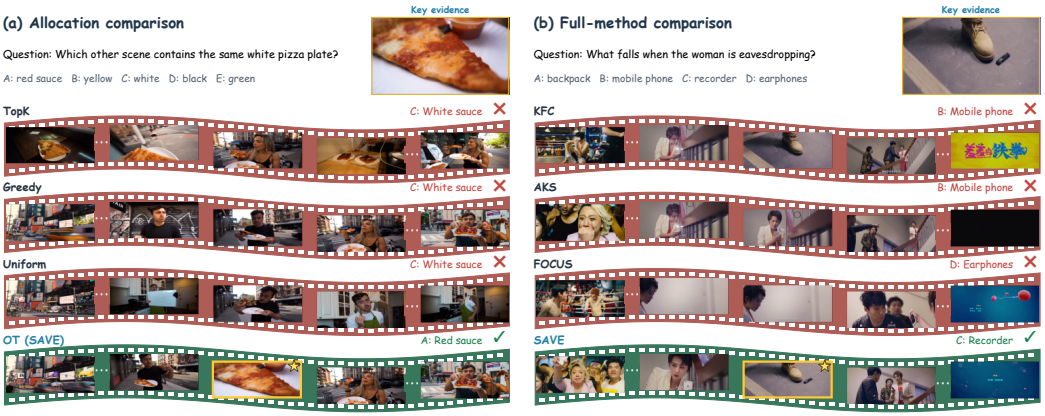


Figure 2: Qualitative examples of (a) frame allocation and (b) complete selection methods. Stars mark key evidence; ellipses indicate omitted selected frames.

show gains on all three benchmarks, supporting the effectiveness of the proposed OT allocation across the two demand transformations.

Softmax further improves OT allocation. With OT fixed, Softmax raises 3-Avg from 69.28 to 69.74, with improvements on all three benchmarks. In contrast, greedy coverage changes little in aggregate (67.81 to 67.77). Thus, Softmax provides an additional benefit when paired with OT.

The complete method makes better use of the fixed frame budget. SAVE achieves the highest 3-Avg of 69.74, exceeding TopK by 2.30 percentage points and Uniform by 4.94. These results support combining question-conditioned demand with temporal allocation to improve on relevance ranking alone and uniform sampling at the same final frame budget.

Qualitative examples. Figure 2 shows two selected success cases. In (a), TopK, greedy, and OT share the same scores: TopK concentrates on repeated scenes, while greedy misses the sauce close-up retained by OT. In (b), SAVE retains the fallen recorder, whereas KFC and AKS predict a phone and FOCUS predicts earphones. In these examples, OT/SAVE retains key visual evidence useful for answering the question. Appendix E describes the frame display and candidate-access settings.

4.4 Frame-Budget and Model Scaling

Frame-budget scaling. Figure 3(c) varies $M \in \{8, 16, 32, 64\}$ on a linear frame-count axis using Qwen3.5-4B and native 1-fps candidates. The displayed SAVE 3-Avg rises from 63.34 to 66.60, 69.74, and 70.61. The $M = 32$ result is also reported in Table 1. Appendix B reports the per-benchmark results and full analysis. Shading marks the gap to Uniform, not uncertainty.

Model-scale sensitivity. Figure 3(a,b) uses the separate $B = 128 \rightarrow M = 32$ protocol. Each model computes its own Q-SelfYes scores, and SAVE is compared with Uniform using the same model and answering prompt. The dark column segment is Uniform accuracy; the light segment adds the SAVE gain. Model sizes are categorical checkpoints. SAVE improves 3-Avg at all five Qwen3.5 scales by 1.37, 3.10, 4.33, 3.55, and 3.18 points, and at the three Qwen2.5-VL scales by 2.50, 3.61, and 3.98 points. Full values and analysis remain in Appendix C.

4.5 Larger Contexts, Candidate Density, and Cost

We examine how the final answering budget and the candidate-scoring budget affect accuracy, then measure the computational cost of a fixed-grid deployment configuration.

Comparison with larger answering contexts. With fixed per-frame spatial resolution and Qwen3.5-4B answering, Uniform improves from 64.80 to 69.95 and 71.91 in 3-Avg as the frame cap increases from 32 to 128 and 256 (Appendix F). SAVE reaches 69.74 with a 32-frame cap, only 0.21 percentage points below Uniform at 128 frames, using one quarter of its maximum answering-frame

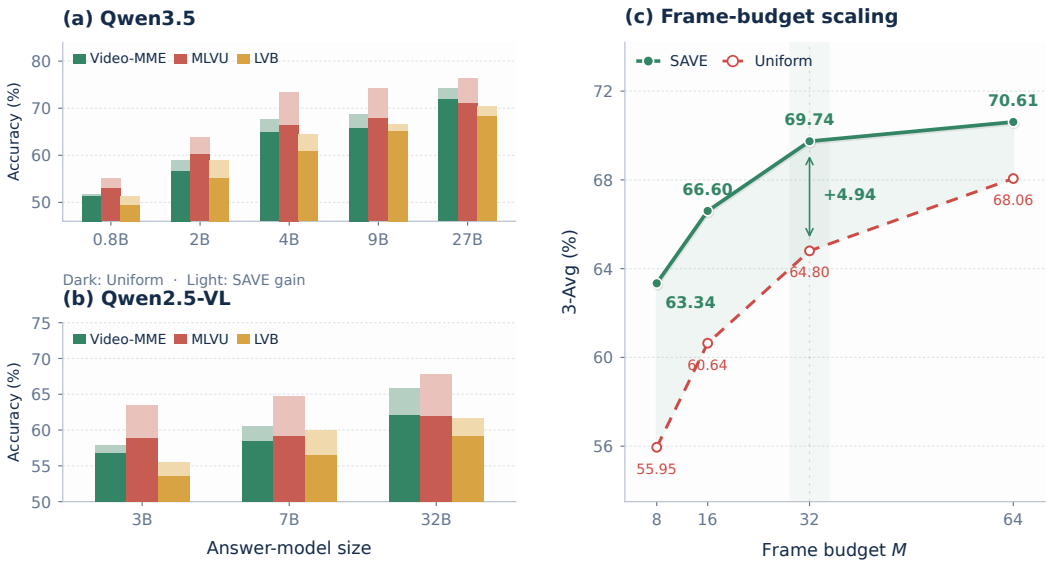


Figure 3: Model-scale sensitivity (left) and frame-budget scaling (right).

budget. Thus, question-conditioned selection can approach the accuracy of a substantially larger uniformly sampled context while keeping the final visual input compact. Uniform at 256 frames achieves a higher score, showing that additional visual input can still improve accuracy when a larger answering budget is available.

Candidate-budget sensitivity. Holding the final budget at $M = 32$, increasing the fixed candidate grid from $B = 64$ to 128, 256, and 512 raises 3-Avg from 67.71 to 68.52, 68.93, and 69.70, respectively. The $B = 512$ result is within 0.04 percentage points of the full-1fps reference of 69.74, showing that a fixed candidate grid can closely approach the reference accuracy. A smaller grid offers a different operating point: $B = 128$ scores 84.8% fewer frame–question pairs across the evaluation set than full-1fps, with a 1.22-point reduction in 3-Avg. These results expose the trade-off between candidate-scoring workload and answer accuracy while keeping the final answering budget unchanged. Appendix G reports the full results, and Appendix H gives the workload counts.

Computational cost. The $B = 128$, $M = 32$ deployment configuration achieves **68.52** 3-Avg and averages **4.118 s/question** of computation over all 6,211 questions on RTX 4090 workers. Candidate scoring takes 2.882 s, OT allocation 0.093 s, and final answering 1.143 s per question. Scoring accounts for approximately 70% of this computation, whereas OT allocation accounts for approximately 2.3%. The main computational expense is therefore evaluating candidate evidence, making the candidate budget an important deployment choice. These compute timings exclude video I/O, initial candidate extraction, model loading, and warmup. Appendix H provides implementation details, memory measurements, and the timing protocol.

5 Conclusion

We introduced SAVE, a VLM-native framework for visual budget allocation without an auxiliary scoring model. Q-SelfYes converts frame-level relevance scores into semantic demand, which guides frame selection through an optimal transport objective that couples relevance and temporal coverage without a separate balancing coefficient. The one-dimensional temporal structure enables exact optimization via dynamic programming. Experiments across multiple long-video benchmarks and diverse VLM architectures show consistent improvements over the evaluated baselines in benchmark-level average performance. With Qwen3.5-4B and a 32-frame budget, SAVE raises 3-Avg from 64.80 with uniform sampling to 69.74. Candidate scoring adds computational cost, and exact optimization of temporal distortion does not guarantee optimal downstream QA accuracy. Future work can explore more efficient evidence scoring and richer measures of evidence substitutability beyond temporal proximity.

AI USE STATEMENT

Generative AI tools were used to assist manuscript editing, code development, and experimental-result consistency checks. The authors are responsible for the final manuscript and all reported results.

ETHICS STATEMENT

This work studies frame selection using existing video-question-answering benchmarks and frozen models. It introduces no new human-subject data collection. Frame selection can omit relevant context, and the underlying models may produce inaccurate or biased answers; the reported benchmark gains do not establish reliability in safety-critical applications. Use and redistribution of benchmark videos and model weights remain subject to their respective terms.

REPRODUCIBILITY STATEMENT

Section 3.6 specifies the allocation objective and recurrence. Appendix D describes score transforms, numerical conventions, and evaluation metrics; Appendices F–H give the supplementary protocols and measured costs. We will release both reference and accelerated implementations, with explicit candidate-budget, precision, and backend settings, environment specifications, and frozen selection and prediction manifests with source hashes.

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Wang Chen, Yuhui Zeng, Yongdong Luo, Tianyu Xie, Luojun Lin, Jiayi Ji, Yan Zhang, and Xiawu Zheng. Wavelet-based frame selection by detecting semantic boundary for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24052–24061, June 2026.

Bo Fang, Yuxin Song, Haoyuan Sun, Qiangqiang Wu, Wenhao Wu, and Antoni B. Chan. Threading keyframe with narratives: MLLMs as strong long video comprehenders. In *The Fourteenth International Conference on Learning Representations*, 2026.

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, Ran He, and Xing Sun. 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 (CVPR)*, pp. 24108–24118, June 2025.

Siegfried Graf and Harald Luschgy. *Foundations of Quantization for Probability Distributions*, volume 1730 of *Lecture Notes in Mathematics*. Springer Berlin, Heidelberg, 2000. doi: 10.1007/BFb0103945.

Allan Grønlund, Kasper Green Larsen, Alexander Mathiasen, Jesper Sindahl Nielsen, Stefan Schneider, and Mingzhou Song. Fast exact k-means, k-medians and bregman divergence clustering in 1D, 2018.

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of*

the 39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 17–23 Jul 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 23–29 Jul 2023.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

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

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

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Haizhou Wang and Mingzhou Song. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal*, 3(2):29–33, 2011. doi: 10.32614/RJ-2011-015.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857. Curran Associates, Inc., 2024. doi: 10.52202/079017-0907.

Songlin Yang and Yu Zhang. FLA: A triton-based library for hardware-efficient implementations of linear attention mechanism. Software, January 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76749–76771. Curran Associates, Inc., 2023. doi: 10.52202/075280-3354.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023.

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

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuanhao Zou, Shengji Jin, Andong Deng, Youpeng Zhao, Jun Wang, and Chen Chen. A.I.R.: Enabling adaptive, iterative, and reasoning-based frame selection for video question answering. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Proofs and Segment-Cost Derivation

Proof of Proposition 1.

Proof. Any coupling π with source marginal μ satisfies $\sum_{j\in\mathcal{S}} \pi_{ij}(t_i - t_j)^2 \geq p_i \min_{j\in\mathcal{S}}(t_i - t_j)^2$ for each i, so $W_2^2(\mu, \nu_{\mathcal{S},w}) \geq J(\mathcal{S})$ for every w. The nearest-frame plan is a valid coupling between μ and $\nu_{\mathcal{S},w}$ for the stated w, and its cost equals $J(\mathcal{S})$. □

Lemma 1 (Contiguity)**.** *For any $\mathcal{S} = \{j_1 < \cdots < j_K\} \subseteq [B]$, the nearest-frame cells $\mathcal{C}_k = \{i \in [B] : \mathrm{nn}_{\mathcal{S}}(i) = j_k\}$ are non-empty contiguous index intervals with $j_k \in \mathcal{C}_k$ and $\max \mathcal{C}_k < \min \mathcal{C}_{k+1}$. Consequently,*

$$\min_{|\mathcal{S}|=K} J(\mathcal{S}) = \min_{\substack{1=a_1\leq b_1<a_2\leq b_2<\cdots<a_K\leq b_K=B \\ a_{k+1}=b_k+1}} \sum_{k=1}^{K} c(a_k, b_k), \qquad c(a,b) = \min_{j\in[a,b]} \sum_{i=a}^{b} p_i\,(t_i - t_j)^2, \tag{9}$$

where the minimum on the right ranges over partitions of $[B]$ into K non-empty contiguous segments.

Proof. Because t_i is strictly increasing in i, the candidates whose nearest selected time is t_{j_k} are exactly those with t_i between the midpoints $(t_{j_{k-1}} + t_{j_k})/2$ and $(t_{j_k} + t_{j_{k+1}})/2$ (with the outer boundaries at $\pm\infty$), an interval that contains j_k itself; this gives the structural claim. For "$\geq$" in Equation 9, take any $\mathcal{S}$: its cells form a feasible partition, and $J(\mathcal{S}) = \sum_k \sum_{i\in\mathcal{C}_k} p_i(t_i - t_{j_k})^2 \geq \sum_k c(\min \mathcal{C}_k, \max \mathcal{C}_k)$ because $j_k \in \mathcal{C}_k$. For "$\leq$", take any feasible partition and let j_k attain $c(a_k, b_k)$; the j_k lie in disjoint segments and are therefore distinct, so $\mathcal{S} = \{j_k\}_{k=1}^K$ has cardinality K and $J(\mathcal{S}) \leq \sum_k \sum_{i=a_k}^{b_k} p_i(t_i - t_{j_k})^2 = \sum_k c(a_k, b_k)$, since serving each candidate by its own segment's representative can only overestimate its nearest-frame cost. □

Segment cost in closed form. Let $P_0(a,b) = \sum_{i=a}^{b} p_i$, $P_1(a,b) = \sum_{i=a}^{b} p_i t_i$, and $P_2(a,b) = \sum_{i=a}^{b} p_i t_i^2$, each available in $O(1)$ from prefix sums, and let $\bar{t}_{ab} = P_1(a,b)/P_0(a,b)$ be the demand-weighted centroid of the segment. Then

$$\sum_{i=a}^{b} p_i\,(t_i - t_j)^2 = \Big[P_2(a,b) - \frac{P_1(a,b)^2}{P_0(a,b)}\Big] + P_0(a,b)\,(t_j - \bar{t}_{ab})^2. \tag{10}$$

The bracketed term is the unnormalized within-segment weighted squared error and does not depend on j, so the optimal representative $j^\star(a,b)$ is simply the candidate frame closest to the centroid $\bar{t}_{ab}$. On the evenly spaced grid used here, the two candidates bracketing $\bar{t}_{ab}$ are obtained by floor and ceiling in $O(1)$ time; binary search would be needed for a general nonuniform grid. Each selected frame is therefore the candidate nearest to the demand centroid of the interval it represents, and the mass it receives, $w_{j^\star} = P_0(a,b)$, is the demand share of that interval; both quantities can be read directly off the solution.

Solver implementation and complexity. Algorithm 1 states the direct recurrence. Equal DP transition costs retain the earlier boundary. Representative selection compares the floor and ceiling candidates and chooses the ceiling on equal computed costs; floating-point rounding can affect mathematical ties. Selection is deterministic for the fixed implementation.

The direct recurrence has $O(M_{\text{eff}}B)$ states and $O(B)$ transitions per state. On the evenly spaced grid of Equation 2, the two candidates around the centroid are obtained by floor and ceiling, so each segment cost takes $O(1)$ time after prefix sums. The direct implementation therefore costs $O(M_{\text{eff}}B^2)$ time and $O(M_{\text{eff}}B)$ memory. Our C++ implementation uses monotone split bounds in a divide-and-conquer DP, reducing the transition search to $O(M_{\text{eff}}B\log B)$ time while retaining the exact objective; related accelerations for ordered clustering are discussed by Grønlund et al. (2018). A separate C++ replay of native-1-fps score curves for all 6,211 frozen questions took 33.16 CPU wall-seconds, or 5.34 ms per question, including score-curve loading and demand normalization. This is a dataset-wide mean, not a measured latency for a one-hour video. Appendix H reports scoring, Python OT, and answering costs for the accelerated $B = 128$ pipeline.

What the allocation does. Three consequences are useful for reading the results. (i) *Flat demand yields approximately uniform spacing*: if $p_i \equiv 1/B$, Equation 5 becomes one-dimensional squared-error quantization on an evenly spaced grid, with representatives restricted to candidate positions. It admits a balanced partition with segment lengths differing by at most one, yielding approximately uniform spacing; discrete ties can also admit other optimal partitions. Its representatives need not coincide with the midpoint Uniform baseline. (ii) *Concentration is sub-proportional*: in the high-resolution limit for a regular continuous one-dimensional density, the density of optimal squared-error quantization points grows as the cube root of the source density (Graf & Luschgy, 2000), so evidence-rich intervals receive more frames, but not in proportion to their demand; the quadratically growing distance cost encourages temporal coverage, and it replaces a hand-set relevance–coverage coefficient. (iii) *Invariance*: rescaling all t_i or all p_i by a positive constant leaves $\mathcal{S}^\star$ unchanged, so the normalized time in Equation 2 and the unit total mass in Equation 4 are without loss of generality. For a fixed candidate grid and frame budget, the allocation depends on the relative demand profile, which is affected by the temperature τ; we keep τ fixed at 1.

B Detailed Frame-Budget Results

Frame-budget scaling. We study the final budget $M \in \{8, 16, 32, 64\}$ to test whether SAVE remains effective when visual context becomes especially scarce. This study uses the same approximately 1-fps candidate stream and Qwen3.5-4B self-scoring setup as the main experiments, with an effective budget $M_{\text{eff}} = \min(M, B_{1\text{fps}})$ for short videos. Table 4 retains the main-table reference at $M = 32$ from Table 1. The displayed scores show a monotonic but saturating trend: 3-Avg rises from 63.34 at $M = 8$ to 66.60, 69.74, and 70.61 at $M = 16, 32, 64$. The displayed 3-Avg differences are 3.14 points between $M = 16$ and $M = 32$, and 0.87 points between $M = 32$ and $M = 64$. The corresponding $M = 64$ versus $M = 32$ benchmark differences are 1.48, 0.44, and 0.69 points for Video-MME, LongVideoBench, and MLVU. Figure 3(c) plots the same values on a linear frame-count axis.

C Detailed Model-Scaling Results

Table 5 retains all tested model sizes under the separate $B = 128 \rightarrow M = 32$ protocol. Each Uniform/SAVE pair uses the same answering model and prompt; SAVE uses that model's own original-question Q-SelfYes scores. MLVU is the seven-task M-Avg, and 3-Avg equally weights the three benchmarks.

Table 4: Effect of the frame budget on SAVE across three long-video benchmarks.

Frames M	Video-MME	LVB	MLVU	3-Avg
8	61.81	60.28	67.91	63.34
16	65.11	62.15	72.53	66.60
32	67.96	65.45	75.80	69.74
64	**69.44**	**65.89**	**76.49**	**70.61**

Table 5: Effect of answer-model scale on Uniform and SAVE.

Answer model	Params.	Video-MME	LVB	MLVU	3-Avg
Qwen3.5-Uniform	0.8B	51.41	49.51	53.15	51.36
Qwen3.5-SAVE	0.8B	**51.78**	**51.31**	**55.10**	**52.73**
Qwen3.5-Uniform	2B	56.70	55.35	60.42	57.49
Qwen3.5-SAVE	2B	**58.89**	**59.01**	**63.86**	**60.59**
Qwen3.5-Uniform	4B	65.04	61.03	66.49	64.19
Qwen3.5-SAVE	4B	**67.63**	**64.55**	**73.39**	**68.52**
Qwen3.5-Uniform	9B	65.81	65.15	68.01	66.32
Qwen3.5-SAVE	9B	**68.78**	**66.57**	**74.27**	**69.87**
Qwen3.5-Uniform	27B	72.04	68.44	71.18	70.55
Qwen3.5-SAVE	27B	**74.33**	**70.38**	**76.46**	**73.73**
Qwen2.5-VL-Uniform	3B	56.78	53.55	58.94	56.42
Qwen2.5-VL-SAVE	3B	**57.85**	**55.50**	**63.42**	**58.92**
Qwen2.5-VL-Uniform	7B	58.56	56.54	59.30	58.13
Qwen2.5-VL-SAVE	7B	**60.56**	**59.91**	**64.75**	**61.74**
Qwen2.5-VL-Uniform	32B	62.15	59.16	61.92	61.08
Qwen2.5-VL-SAVE	32B	**65.85**	**61.56**	**67.77**	**65.06**

Model-scale sensitivity. Finally, Table 5 retains the separate controlled $B = 128 \rightarrow M = 32$ protocol across the Qwen3.5 and Qwen2.5-VL families. For each SAVE configuration, the answer model itself computes the Q-SelfYes curve. We compare SAVE against a matched Uniform baseline using the same model, candidate pool, frame budget, and answering prompt. The fixed candidate count bounds the self-scoring workload at larger model sizes. These results are compared within each model pair; the $B = 128$ grid differs from the native 1-fps pool used in the main table and matches the $B = 128$ configuration in the candidate-density study (Appendix G).

SAVE outperforms its matched Uniform baseline at every tested scale, improving 3-Avg by 1.37, 3.10, 4.33, 3.55, and 3.18 points from 0.8B to 27B, respectively. Absolute accuracy rises substantially with model scale, while the positive gain at all five scales shows that answerer-aligned temporal allocation is not confined to a particular model capacity.

The Qwen2.5-VL family shows the same paired pattern at 3B, 7B, and 32B, where the model computes its own evidence curve: SAVE improves 3-Avg by 2.50, 3.61, and 3.98 points, respectively (Figure 3). This consistent gain makes the scaling trend independent of cross-model evidence transfer.

D Implementation and Evaluation Details

Benchmarks and metrics. We evaluate zero-shot multiple-choice video question answering on three long-video benchmarks. **Video-MME** contains 2,700 questions evenly divided into its official short, medium, and long duration groups (Fu et al., 2025). **MLVU** uses all 2,174 questions in the seven multiple-choice task annotation files of our downloaded release (Zhou et al., 2025), excluding the two free-form generation tasks. The evaluation manifest matches these annotations exactly; no further questions are removed for video availability or decoding. All MLVU comparisons are reproduced on this same set, rather than copied from the 2,593-example, nine-task setting reported elsewhere. **LongVideoBench** uses its official validation split, comprising 1,337 questions

over 753 videos (Wu et al., 2024). Following FOCUS (Zhu et al., 2026), we additionally divide LongVideoBench into short (< 3 min), medium (3–20 min), and long (> 20 min) groups. We report exact-match question-level accuracy for Video-MME and LongVideoBench (LVB). MLVU M-Avg equally weights its seven task-category accuracies:

$$\text{3-Avg} = \frac{\text{Acc}_{\text{Video-MME}} + \text{Acc}_{\text{LVB}} + \text{M-Avg}}{3}, \qquad \text{M-Avg} = \frac{1}{7}\sum_{c=1}^{7} \text{Acc}_{\text{MLVU},c}. \tag{11}$$

Answer models. We test four frozen VLM families: LLaVA-OneVision-Qwen2-7B (Li et al., 2024), Qwen2.5-VL-7B-Instruct (Bai et al., 2025), InternVL3-8B (Zhu et al., 2025), and Qwen3.5-4B (Qwen Team, 2026). The model receives the selected frames in temporal order and answers the original multiple-choice question with deterministic greedy decoding. For Video-MME and LongVideoBench, inputs contain only these visual frames, the original question, and answer choices: no audio, subtitle files, or transcripts are supplied. Text embedded in the video frames remains visible. Q-SelfYes assesses a single frame with the question text. Selection is training-free; no selector receives answer choices. Gold labels are loaded only after selections and predictions are frozen.

Baselines. We compare with uniform sampling and four training-free, question-aware selectors: AKS (Tang et al., 2025), KFC (Fang et al., 2026), FOCUS (Zhu et al., 2026), and WFS-SB (Chen et al., 2026). We run the authors' released implementations and preserve their native selection logic: in particular, KFC uses the official graph-search implementation, whereas FOCUS retains its adaptive observation process and official under-filling behavior. For Table 1, KFC, AKS, and WFS-SB use the controlled 1FPS pool, whereas FOCUS observes the native video adaptively; all are capped at $M = 32$. All displayed predictions are produced by our common answering and evaluation pipeline on the same local benchmark splits; we do not mix in incompatible author-reported MLVU scores.

Full-1fps candidate sampling. The full-1fps configuration decodes candidates from the native video at approximately one frame per second. For source frame rate r, target sampling rate $f_s = 1$, and effective video bounds $[f_{\min}, f_{\max}]$, we use

$$\Delta = \max(\lfloor r/f_s \rfloor, 1), \qquad f_i = f_{\min} + (i-1)\Delta, \quad f_i \le f_{\max}. \tag{12}$$

The resulting candidate count B varies with video duration. If $B < M$, every candidate is retained rather than padding the view with duplicate frames. This gives an at-most-M answering budget under the full-1fps protocol without duplicate padding.

Fixed-grid candidate sampling. The model-scaling, fixed-grid deployment, and candidate-density runs sample directly from the original video's source-frame indices, rather than resampling the approximately 1-fps pool. Let $N \ge 1$ be the number of source frames, $B \in \{64, 128, 256, 512\}$ the requested candidate budget, and $K = \min(B, N)$ the actual candidate count. In this sampling protocol, B denotes the requested budget; the candidate-count symbol B in the allocation equations and algorithm is instantiated with the actual count K. For $K \ge 2$, the zero-based sampled frame indices are

$$g_i = \left\lfloor \frac{i(N-1)}{K-1} \right\rfloor, \qquad i = 0, \ldots, K-1. \tag{13}$$

For $K = 1$, the sole index is $g_0 = 0$. The rule includes both endpoints and rounds fractional indices down. When $N \ge B$, it produces exactly B distinct source-frame indices. When $N < B$, it retains all N source frames, without padding with duplicate frames. Thus B128 supplies 128 candidates whenever the video has at least 128 source frames, regardless of its duration or the size of its approximately 1-fps pool. Here, distinctness refers to source-frame indices. SAVE selects $\min(M, K)$ distinct candidates, with $M = 32$ in these experiments, and restores their chronological order. Its temporal coordinates are normalized candidate positions, not absolute seconds.

Q-SelfYes prompt and token groups. Each candidate frame is paired with the original question using the following prompt; answer choices and timestamps are excluded.

```
Question: {q}
Does this frame contain visual evidence helpful for
answering the question? Answer yes or no.
```

The sets $\mathcal{Y}$ and $\mathcal{N}$ contain the first-token IDs of capitalization- and leading-space variants of "yes" and "no", respectively. Equation 3 groups these logits using log-sum-exp. Scoring uses a forward pass rather than autoregressive answer generation, and independent frame–question pairs can be batched.

Evaluation execution. Full evaluations are sharded over eight GPUs. Selected frames are supplied chronologically. The main-table SAVE result is reused in subsequent comparisons only when the configuration is unchanged. Additional prompts, checkpoint identifiers, split hashes, and per-question predictions will be released with the code.

Protocol scope. We organize the experiments around two candidate protocols: **full-1fps**, which retains the full approximately 1-fps candidate stream, and **B128–M32**, which uses a fixed 128-candidate grid and selects 32 frames. The default full-1fps final budget is 32; frame-budget and larger-context experiments vary only the final budget within this candidate protocol. The candidate-density study varies the fixed-grid budget over $B \in \{64, 128, 256, 512\}$ at $M = 32$. FOCUS retains the adaptive candidate access described in the main text.

The full-1fps candidate stream uses a source-frame stride of $\max(\lfloor r \rfloor, 1)$ for source frame rate r. The allocator operates on normalized candidate indices, not absolute seconds. It retains all candidates when $B \leq M$, and otherwise returns M distinct frames in chronological order. Demand is computed in float64 with shifted scores clipped to $[-80, 0]$ before exponentiation. The C++ OT solver uses prefix sums, divide-and-conquer transition search, and backtracking. Kernel coverage instead uses global marginal-gain greedy search with the bandwidth in Equation 14.

External score transforms. For Table 2, CLIP (Radford et al., 2021) uses normalized image–text cosine similarity multiplied by its checkpoint's learned logit scale. SigLIP (Zhai et al., 2023) and SigLIP2 (Tschannen et al., 2025) use their learned scaled cosine logits plus logit bias; the additive bias cancels in temporal Softmax. BLIP-ITM (Li et al., 2022) and BLIP2-ITM (Li et al., 2023) use match/non-match log-odds, rather than directly normalizing the cached match probabilities. All methods then use temperature 1 and the same demand transformation. Thus the comparison evaluates these fixed score pipelines; it does not establish superiority over individually temperature-tuned external scorers.

Metrics and preserved scaling protocol. Video-MME and LongVideoBench report question accuracy. MLVU reports the equal-weight mean over seven tasks: anomaly recognition (200 questions), counting (206), egocentric reasoning (352), needle QA (355), action order (259), plot QA (539), and topic reasoning (263). These sum to 2,174. The displayed 3-Avg equally weights the three benchmark scores, including MLVU M-Avg. Aggregate metrics are computed from unrounded scores and then rounded to two decimal places. The model-scaling table retains its separate $B = 128$ protocol and all model sizes; its numbers should not be read as measurements on the native 1-fps pool. The frame-budget table also retains every tested budget.

E Allocation under the Full-1fps Protocol

Table 3 compares TopK, Gaussian greedy coverage, and OT on the same frozen full-1fps Q-SelfYes score cache for all 6,211 questions, with $M = 32$ and the same Qwen3.5-4B answering configuration. No candidate scores are recomputed. TopK ranks raw log-odds, breaks ties toward earlier candidates, and returns frames chronologically. Greedy uses Equation 14; OT uses the exact temporal allocation objective.

Interpretation of the allocation comparison. The displayed 3-Avg values are 67.44 for TopK, 67.77 for Softmax + greedy, 69.28 for normalized Yes-probabilities + OT, and 69.74 for SAVE. SAVE exceeds TopK by 2.30 percentage points and Softmax + greedy by 1.97 percentage points. Holding the demand transformation fixed, OT exceeds greedy coverage by 1.47 percentage points with normalized Yes-probabilities and by 1.97 with Softmax, improving all three benchmark scores in both comparisons. Holding OT fixed, Softmax adds 0.46 percentage points; with greedy coverage, it changes 3-Avg from 67.81 to 67.77. Thus, OT improves over the evaluated greedy allocator under both demand transformations, while the additional benefit of Softmax appears with OT. These

comparisons change both the allocation objective and solver, so they do not isolate exact optimization alone. SAVE also provides an exact solution to its stated temporal allocation objective without a Gaussian bandwidth or a separate relevance–coverage weighting coefficient.

Coverage variants. The *Evidence* column in Table 3 marks question-conditioned selection. TopK ranks Q-SelfYes log-odds without normalization. Softmax demand normalizes the odds e^{s_i}; the variants without Softmax instead use $p_i = \sigma(s_i)/\sum_u \sigma(s_u)$, normalizing per-frame Yes-probabilities. Gaussian greedy coverage maximizes

$$G(\mathcal{S}) = \sum_{i=1}^{B} p_i \max_{j \in \mathcal{S}} \exp\left(-\frac{(i-j)^2}{2\sigma^2}\right), \qquad \sigma = \frac{B/M_{\text{eff}}}{\sqrt{2}}. \tag{14}$$

where the bandwidth is in candidate-index units. Starting from zero coverage, greedy repeatedly adds the candidate with the largest global marginal gain, breaking ties toward earlier indices. All methods retain every candidate when $B \leq M$ and otherwise return M distinct frames chronologically. Gaussian coverage saturates with distance, whereas OT minimizes demand-weighted squared temporal distance using Algorithm 1.

Qualitative display. Figure 2 uses Qwen3.5-4B and $M = 32$. Each filmstrip shows five actual selected frames in temporal order: the first, last, and three around the selection nearest the OT/SAVE evidence time. Ellipses omit the remaining selected frames; horizontal spacing does not encode elapsed time. Questions and options are abbreviated, and insets enlarge highlighted evidence. The observation in (a) is based on inspection of all 32 selected frames. In (b), KFC, AKS, and SAVE use native 1-fps candidates, whereas FOCUS retains its adaptive candidate access. The complete methods differ in both scoring and allocation.

F Uniform with Larger Final Contexts

All three Uniform runs ($M = 32, 128, 256$) use Qwen3.5-4B FP16/SDPA on the same native 1-fps candidate streams, with budget-specific frozen Uniform selections and multiple-choice prompts, and fresh answers. Per-frame spatial resolution is fixed across budgets: frames retain the initial 106,624-pixel resize limit, and the processor's video-level resize is disabled. Actual frame counts, spatial grids, and visual tokens are verified per question. Short streams use all unique candidates. The $M = 32$ Uniform and SAVE rows reproduce the main-table results. SAVE and these Uniform controls use the same Qwen3.5-4B FP16/SDPA answering backend.

Table 6: Uniform with larger final contexts and the SAVE reference.

Method	M	Video-MME	LVB	MLVU	3-Avg
Uniform	32	65.44	60.58	68.36	64.80
Uniform	128	70.00	64.25	75.59	69.95
Uniform	256	71.44	66.64	77.64	71.91
SAVE	32	67.96	65.45	75.80	69.74

For the fixed-resolution Uniform runs, paired comparisons over the same 6,211 questions give 623 improvements and 306 harms for $M = 128$ versus $M = 32$ ($p = 1.17 \times 10^{-25}$), 704 improvements and 276 harms for $M = 256$ versus $M = 32$ ($p = 9.51 \times 10^{-44}$), and 304 improvements and 193 harms for $M = 256$ versus $M = 128$ ($p = 7.28 \times 10^{-7}$). In each comparison, an improvement means that the larger-budget run answers correctly where the smaller-budget run answers incorrectly; a harm is the reverse. The p-values use two-sided exact McNemar tests. These pooled question-level tests are distinct from the equal-benchmark and equal-task weighted 3-Avg.

Increasing Uniform from 128 to 256 frames improves 3-Avg by 1.96 points. Answering averages 3.725 versus 7.395 s per question, with cumulative cost rising from 6.43 to 12.76 GPU-h and peak allocated memory from 12.21 to 15.80 GiB (Table 10). Thus the additional accuracy requires approximately twice the answering time at fixed per-frame spatial resolution. The Uniform32 reference remains at 64.80 3-Avg, as in the main table.

G CANDIDATE-DENSITY SENSITIVITY

This ablation varies the candidate-grid budget over $B \in \{64, 128, 256, 512\}$ while fixing the final frame budget at $M = 32$. All grids follow the direct source-frame sampling rule in Equation 13, retaining all source frames when fewer than B are available. For each candidate grid, SAVE uses original-question Q-SelfYes scores to construct normalized demand and solves OT to select frames; Qwen3.5-4B then answers using the selected frames. The $B = 128$ configuration is the same as that used for model scaling and deployment, achieving 68.52 3-Avg. The full-1fps result is included as a reference.

Table 7: Effect of the candidate-grid budget B with a fixed final frame budget of $M = 32$.

Candidate budget	Video-MME	LVB	MLVU	3-Avg
B64	66.56	64.47	72.11	67.71
B128	67.63	64.55	73.39	68.52
B256	68.00	64.10	74.69	68.93
B512	68.33	64.92	75.86	69.70
Full 1 fps	67.96	65.45	75.80	69.74

Table 7 shows 3-Avg increasing from 67.71 at $B = 64$ to 69.70 at $B = 512$. Increasing the candidate budget from 128 to 512 gains 1.18 points. At $B = 512$, the score is 0.04 points below the full-1fps reference (69.74). For the fixed-grid B512 versus B256 comparison, there are 258 improvements (B512 correct, B256 incorrect) and 209 harms (B512 incorrect, B256 correct), with $p = 0.026$ from a two-sided exact McNemar test. The full-1fps row reuses the main-table reference; no paired significance claim is made for its 0.04-point difference from B512. No scoring GPU-time comparison across candidate budgets was performed.

Paired-test interpretation. The Uniform frame-budget comparisons in Appendix F and the fixed-grid B512 versus B256 comparison in Appendix G use two-sided exact McNemar tests on discordant correctness pairs across the same 6,211 questions. These tests concern pooled question accuracy, not the equal-benchmark and equal-task weighted 3-Avg. Questions sharing a video may be dependent; the exploratory p-values are not adjusted for multiple comparisons and are not video-cluster-aware significance estimates.

H B128–M32 IMPLEMENTATION AND COMPUTATIONAL COST

Protocol map. Table 8 summarizes the two candidate protocols. Full-1fps supplies the main accuracy comparison and its component and final-budget ablations. B128–M32 bounds candidate scoring for model scaling and deployment. For Qwen3.5-4B, the B128–M32 model-scaling result and deployment measurements use the same scoring and answering configuration described below, achieving 68.52 3-Avg at 4.118 s/question.

Table 8: Candidate protocols for Qwen3.5-4B at $M = 32$. B128 accuracy and timing use the same configuration. A dash means no paired runtime is reported.

Candidate protocol	3-Avg	Compute s/question
Full-1fps	69.74	—
B128–M32	68.52	4.118

Candidate workload. The recorded fixed-grid workload contains 795,008 frame–question pairs across 6,211 questions, compared with 5,239,164 in the full-1fps audit: an 84.8% reduction in scored pairs. This counts scoring inputs, not measured end-to-end speedup.

Deployment implementation. The deployment configuration uses cached $B = 128$ candidate JPEGs, original-question Q-SelfYes scoring, Softmax demand, and exact Python OT with $M = 32$. Scoring uses BF16, FLA 0.4.2 (Yang & Zhang, 2024) fused kernels for 24 Gated-Delta layers (Yang

et al., 2025), and SDPA for full attention in the text and vision modules. Answering uses FP16 with SDPA and the torch Gated-Delta implementation. Candidate images are batched in groups of 16, and the scoring projection uses only the eight Yes/No vocabulary rows. Scoring and answering run in separate processes, with one RTX 4090 per worker.

Timing scope. The retained deployment profile covers all 6,211 questions. Scoring includes preprocessing and forward computation; answering includes encoding and generation. Compute time excludes video I/O, initial candidate extraction, model loading, warmup, and profiling. Table 9 reports component means; totals are computed before rounding.

Table 9: B128–M32 SDPA component timings for the configuration achieving 68.52 3-Avg.

Stage	Mean (s/question)	Peak allocated GPU memory (GiB)
Scoring	2.882	9.35
Python OT	0.093	—
Answering	1.143	9.58
Compute total	4.118	—
With recorded I/O	10.133	—

Recorded I/O includes cached candidate reads and selected-frame decoding/resizing, but excludes initial candidate extraction; this is not raw-video end-to-end latency. Thermal telemetry was clear for 4,945 scoring records and 6,209 answering records; reported costs retain all 6,211 questions, including records with thermal flags.

Cumulative component costs. Table 10 reports B128–M32 SAVE costs alongside full-1fps Uniform final-budget controls on the same 6,211 questions and RTX 4090 hardware. Uniform uses fixed per-frame resolution and SDPA answering as in Appendix F. Its rows are answering-cost references, not matched B128 Uniform runs. GPU-h sums per-question component times over the assigned GPUs. The timing exclusions are the same as above.

Table 10: Cumulative component costs.

Methods	Frames	Scorer GPU-h	OT CPU-s	Answer GPU-h	Answer peak(GiB)
SAVE	32	4.97	576.51	1.97	9.58
Uniform	32	0	—	2.07	9.57
Uniform	128	0	—	6.43	12.21
Uniform	256	0	—	12.76	15.80

Answer peak is the maximum allocated GPU memory across 6,211 questions. Uniform thermal telemetry was clear for 6,198/5,979/5,593 questions at $M = 32/128/256$; all completed records remain in the totals. Python OT costs here have a different implementation and timing scope from the C++ replay in Appendix A.