arXiv:2611.24616v1 [cs.CV] 18 Nov 2026

EgoJev: Finding Evidence in Egocentric Video by Reading VLM Decisions

Chunggi Lee Hanspeter Pfister
Harvard University

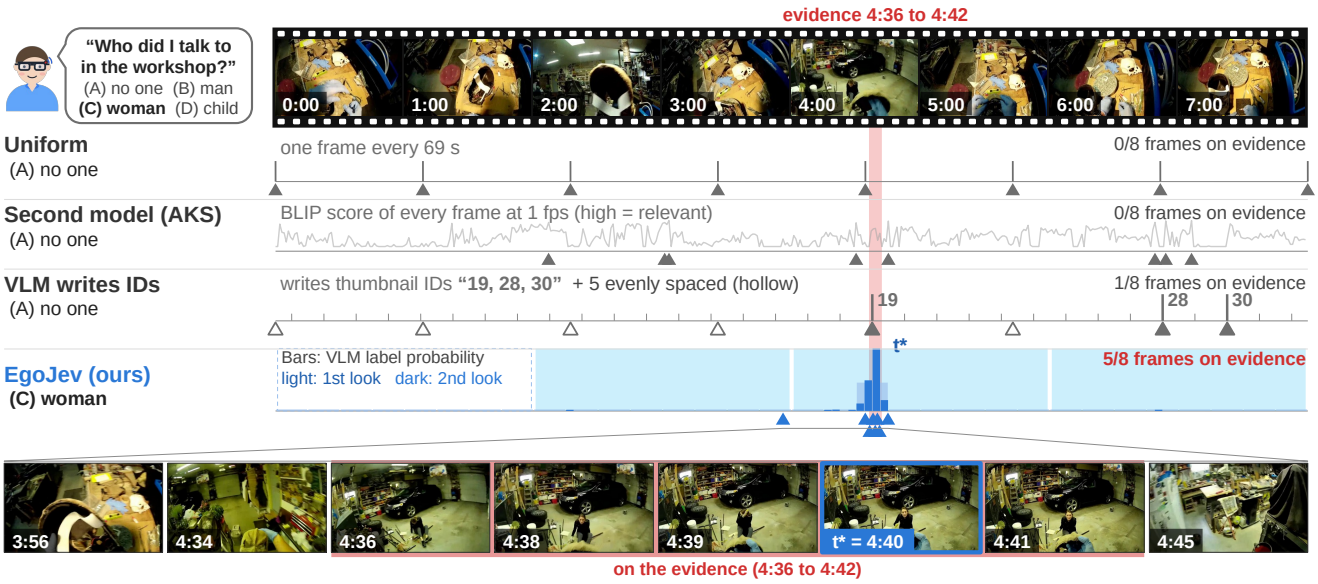

Figure 1. **Four ways to choose 8 frames** for one question on an 8-minute video (Qwen3.5-4B, QAEgo4D). Uniform sampling, a second model (AKS) and written IDs put at most one of their frames (triangles) inside the brief evidence window (red band), and all three answer wrongly. EgoJev reads the VLM's choice over labeled thumbnails as a distribution (bars, square-root scale), looks again in the windows it points to (shaded; dashed: skipped), and places its frames at the quantiles. Five of its 8 frames lie inside the evidence window. It answers correctly and returns the evidence time t^* without generating text.

Abstract

Questions about long egocentric video often depend on one brief moment, such as the second a hand puts an object down. A vision-language model (VLM) that sees a few evenly spaced frames usually misses it. Training-free methods rank frames with a separate image-text model or ask the VLM to write a time. In a controlled test, the VLM itself locates the moment better when it chooses among labeled frames, and we read this choice from its output distribution. With the same candidates, this choice lands on the evidence more often than image-text scores or a written time span, and it compares all candidates in one decision. Based on this finding, we propose EgoJev, a training-free frame selector for wearable assistants. EgoJev reads where the evidence is from labeled thumbnails and looks again at the windows it points to. It places its few answer frames at the quantiles of the resulting distribution. One frozen VLM selects, answers and returns the evidence time, with no second model. A wearable device can encode its coarse thumbnails while it records. With Qwen3.5-4B and 8 answer frames, EgoJev raises QAEgo4D accuracy from 51.2% with uniform sampling to 62.0%, and it outperforms recent training-free frame selectors on QAEgo4D and MyEgo. A single-level version runs on a Mac and an iPhone.

1. Introduction

Smart glasses can record what their wearer sees and later answer questions about it, such as *"Where did I put the scissors?"* or *"What did I take out of the fridge?"* [5, 11]. Such an assistant works under three limits. First, it can afford

only a few frames per question. The device has little compute, and sending video to a server costs bandwidth and privacy. Second, a single small model must do the work, since a second network that watches every frame adds memory and power. Third, the answer should come with the moment it was taken from, because a user who asks about the past wants to see that moment.

Training-free methods choose better frames in three ways, and none fits these limits well. Similarity-based selectors [17, 25] score every frame with a separate image-text model such as CLIP [21] or BLIP [14]. They need that second model, and they only compare frames with the question text. A second way asks the VLM to rate each frame on its own, for example by the probability of "Yes" [34]. This needs one forward pass per frame, and frames scored in separate calls are never compared. The third way asks the VLM to write when the evidence appears, as a time span or as frame IDs [2]. The spans it writes land on the evidence less often.

Our idea is to let the VLM choose and to read its choice from the output probabilities (Fig. 1). The VLM sees the candidate frames as separate images, each after a number label such as "Segment 3:", and is asked where the answer is. One forward pass then gives the probability of each label. This treats the VLM as a decision model that returns a probability for each option, as in Jev [1]. Based on this idea, we propose EgoJev, a training-free frame selector for wearable assistants. EgoJev finds the evidence in two steps. It first reads one small labeled thumbnail per video segment. Brief evidence often falls between two thumbnails, so it then reads again inside the windows with the most probability and combines the two levels. It then places the answer frames at the quantiles of this distribution, so they gather when the model is sure and spread when it is not. The same VLM answers from these frames, and one more readout returns the evidence time. Unlike per-frame rating [34] or similarity selectors [17, 25], EgoJev compares all candidates in one decision. Unlike ID writing [2], it reads this decision as a distribution and generates no text. One frozen VLM thus selects, answers and points to its evidence, with no second model. EgoJev is also plug-and-play, since it needs only the output probabilities of the VLM that answers. Because the thumbnails come before the question, a wearable device can encode them while it records.

We evaluate EgoJev on three egocentric benchmarks with four VLMs from three families. On the same candidates, reading the choice finds the evidence more often than rating frames, writing spans or scoring with BLIP and CLIP. It also works much better with labeled images than with one video. Its lead is largest on evidence shorter than 3 s, the kind of moment that egocentric questions ask about.

In summary, our contributions are:

- We find that a frozen VLM (Qwen3.5-4B) localizes brief evidence in egocentric video better when it chooses among labeled frames and we read its decision from the output distribution. With the same model and candidates, this beats rating each frame and writing a span in less than a third of their localization time.
- We propose EgoJev, a training-free and plug-and-play frame selector for wearable assistants. It reads the VLM's decision from coarse to fine and places the answer frames at its quantiles. One frozen VLM selects, answers and returns the evidence time, with no second model. Its thumbnails can be encoded during recording.
- With Qwen3.5-4B, EgoJev raises QAEgo4D accuracy from 51.2% to 62.0% and outperforms recent training-free frame selectors on QAEgo4D and MyEgo. Its mean accuracy exceeds that of AKS, BOLT, FOCUS and Write IDs with each of the four VLMs in Tabs. 1 and 3, and a single-level version runs on a Mac and an iPhone.

2. Related Work

Frame selection for video VLMs. A VLM can read only a few frames of a long video, so many training-free methods decide which frames to show it. Most of them score frames by image-text similarity. AKS [25] balances relevance and coverage with BLIP [14] scores. BOLT [17] samples frames in proportion to sharpened CLIP scores, with a sharpness tuned per benchmark. Others add multi-resolution inputs [37], bandit search [40], answer options [30], or scene changes [8]. These methods run a second embedding model over every candidate frame. Other methods read attention maps inside a second, smaller MLLM whose layer and budget are tuned on held-out videos [27] or let an LLM pick frames from generated captions [26, 28, 36]. EcoFrame [15] reads the attention and output entropy of the answering VLM to decide when to add frames, and still scores relevance with CLIP. A few methods let a VLM guide the search. T* [33] searches grids of frames with an object detector, and A.I.R. [41] scores frames at 1 fps with CLIP and then has a VLM write a score for each candidate, for up to six rounds. EgoJev targets egocentric questions whose evidence is brief. In contrast to these methods, it needs no second model and generates no text: the answering VLM sees all candidates in one prompt, and we read one distribution over them.

Temporal grounding. Egocentric benchmarks ask about brief past moments [5, 11] or about whole clips [18]. The first kind rewards finding the moment, and its evidence annotations let us measure localization directly. MyEgo [31] asks about the camera wearer and marks the answer moment. Video LLMs can be trained to write timestamps [12, 23], and GroundVQA [7] trains a model to answer and ground egocentric questions together. NumPro [29] draws frame numbers on the frames so that a model can write the

number of the relevant frame, and TimeMarker [6] interleaves time tokens with frames during training. TCoT [2] shows the VLM segments of 64 frames with IDs and has it write the IDs of the relevant ones, one call per segment. These methods output time or frame IDs as generated text, and most need training. EgoJev is training-free and reads the evidence time from label probabilities.

Decision readout. Reading a choice from label probabilities is an old way to use a language model as a classifier [38]. Jev [1] recently framed it as a decision model that returns a probability for each option in one forward pass, and open frozen LLMs match it without training [9, 16]. Label wording strongly shapes these decisions [3]. Listwise rerankers write an order of passage IDs [24], and FIRST [22] reads the ranking from the logits of the first output token. For video, SeViLA [34] scores each frame by the probability of "Yes" to whether it helps answer the question, and UG-Sample [13] by the entropy of the answering MLLM's output given that frame. Each frame is then scored in its own forward pass and is never compared with the others. We read one decision over all candidate images and use its whole distribution to place the frame budget. One VLM thus makes three decisions: where the evidence is, what the answer is, and when it was seen.

3. Method

Our goal is to answer a multiple-choice question about a long egocentric video with a frozen vision-language model (VLM), a fixed frame budget, and no training. The answer usually depends on a short moment, while uniform sampling spends most of the budget elsewhere. EgoJev asks one VLM for three decisions and reads each one from its output probabilities: *where* the evidence is (Sec. 3.2), *what* the answer is (Sec. 3.3), and *when* the evidence was seen (Sec. 3.4), as shown in Fig. 2. The key step is the first one: its probabilities, used as a whole distribution, decide where the frame budget goes. One distribution drives every later step: it chooses which windows to look at again, decides how many answer frames go where, and these frames are read once more for the evidence time.

3.1. Problem setup

A video V comes with a question q and options $O = \{o_1, \dots, o_K\}$. The VLM can see only B frames, far fewer than the video has, and by default they are sampled uniformly. A vision encoder and a projector turn each frame into visual tokens, and the language model with parameters θ reads them with the text as one input X. With a fixed budget B, the answer depends on which frames are selected. Similarity-based methods [17, 25] score each frame by its cosine similarity to the question in a separate image-text model, and generation-based methods ask the VLM to write a time or a frame ID. EgoJev scores the frames with the output probabilities of the VLM that answers.

3.2. Where: finding the evidence

The central step of EgoJev asks the VLM where the evidence is, as one multiple-choice question over candidate frames. Given candidate frames $f_1, \dots, f_N$, each frame is shown as its own image after a number label, followed by the question, the options and an instruction:

$$X_{\text{loc}} = \big[\, \ell_1, v_1, \dots, \ell_N, v_N,\ q,\ O,\ u,\ \text{“Segment: [”} \,\big], \tag{1}$$

where ℓ_i is the label "Segment i:", v_i are the visual tokens of f_i, and u is *"Which segment shows what is needed to answer this question?"* The options help the model look for the object or action that tells them apart.

Reading the choice. The VLM would answer this prompt by writing a number, but before it writes, it already gives a probability to every possible next token. Let $p_\theta(\cdot \mid X)$ be these probabilities after the input X. The answer completes "Segment: [", so label i is scored as the tokens of "i]". A label such as "17]" is several tokens (*e.g.*, "1", "7", "]"), so its probability is the product of the probabilities of its tokens $c_{i,1}, \dots, c_{i,L_i}$, each given the tokens before it. The model also gives some probability to tokens that are not labels, so these products are normalized over the N labels:

$$P(i) \;\propto\; \prod_{l=1}^{L_i} p_\theta\big(c_{i,l} \mid X_{\text{loc}}, c_{i,<l}\big), \qquad \sum_{i=1}^{N} P(i) = 1. \tag{2}$$

This readout $P(i)$ is the model's choice among the N frames, as a distribution. Each label ends with "]" so that the probability of "1]" does not include "17]". This takes one forward pass plus a few cached steps per label, writes no text, and uses only these output probabilities, not hidden states or attention.

Because the model sees all candidates before it answers, it can compare them. Methods that rate each frame on its own, such as SeViLA [34], score the frames in separate calls, so the frames are never compared and their scores need not share a scale. Here all candidates share one distribution. The frames must enter as separate labeled images: a video encoder may merge neighboring frames, and then a single frame has no label to choose (Sec. 4.4).

Coarse to fine. EgoJev first splits the video into N equal segments and shows each by its middle frame as a small thumbnail of a few dozen visual tokens. The readout over these thumbnails is the coarse distribution P^c_j, $j = 1, \dots, N$. Brief evidence is often shorter than a segment, so it can fall between two thumbnails. EgoJev therefore takes a second, closer look where P^c points. For this, the video

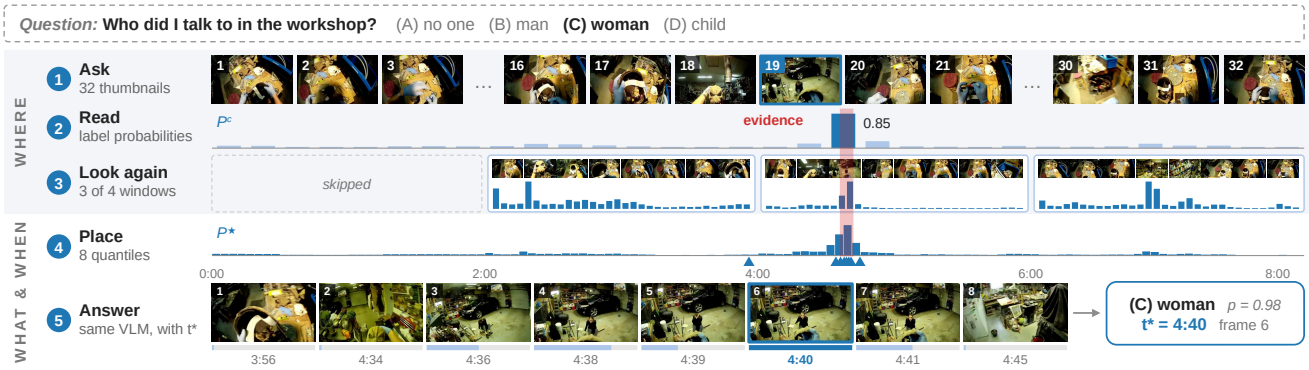

Figure 2. **Overview of EgoJev** on one video (Qwen3.5-4B, QAEgo4D). (1) The VLM sees 32 labeled thumbnails with the question and options. (2) EgoJev reads the probability of each label (bars on a square-root scale, largest 0.85, red band: annotated evidence). (3) The VLM reads 32 denser thumbnails in each of the 3 windows with the most probability. (4) The two readouts give $P^\star$, and the 8 answer frames are its quantiles (triangles). (5) The same VLM answers from these frames (p: probability of the chosen answer) and reads which frame shows the evidence (bars under the frames), which gives t^*. No step generates text.

is also split into W windows, and each window is divided again into N fine segments s with their own thumbnails. For the k windows with the most coarse probability, EgoJev reads these new, denser thumbnails with the same prompt. This gives the fine distribution P^f_s over all fine segments of the video. In each of these k windows, P^f is the new readout, scaled so that the window keeps its coarse probability in total. In the other $W-k$ windows, P^f spreads the coarse probability evenly over its fine segments. The two levels are combined by a geometric mean,

$$P^\star_s \propto \sqrt{P^c_{j(s)}\, P^f_s}, \tag{3}$$

where $j(s)$ is the coarse segment that contains s, and $P^\star$ is normalized over all fine segments. Both levels are readouts of the answering VLM, unlike FOCUS [40], which refines with BLIP scores, and T* [33], which zooms in with a detector. The square root averages the two levels in log space. A plain product would count the evidence twice and make $P^\star$ sharper (Sec. 4.5). With $k = 0$ this is the single-level readout.

3.3. What: from the readout to the answer

$P^\star$ says how strongly the VLM points to each segment, so EgoJev places the B answer frames where it points. Spreading each $P^\star_s$ evenly over its segment gives a distribution μ over time, with cumulative distribution F. EgoJev places the b-th answer frame at

$$t_b = F^{-1}\Big(\frac{b-1/2}{B}\Big), \qquad b = 1, \dots, B. \tag{4}$$

Each frame covers an equal share $1/B$ of the probability and sits in the middle of its share. Among frames that each stand for $1/B$ of the probability, these minimize the average time between an evidence moment drawn from μ and the frame that stands for it, which is the 1-Wasserstein distance [32] (proof in Sec. A):

$$\{t_b\} = \arg\min_{t_1,\dots,t_B} \mathcal{W}_1\Big(\mu,\ \frac{1}{B}\sum_{b=1}^{B}\delta_{t_b}\Big), \tag{5}$$

where δ_{t_b} is the point mass of a frame at time t_b.

The placement rule adds no threshold and no per-dataset sharpness. When the model is sure, most frames land in one segment and cover the brief moment densely. When it is unsure, they spread over the likely segments. When $P^\star$ is flat, the frames are evenly spaced, so even spacing is the special case with no information. BOLT [17] also samples by inverse transform, but on CLIP similarities that are made into a distribution with a sharpness α set per dataset, while $P^\star$ is already normalized over the candidates. A temperature τ in $(P^\star)^{1/\tau}$ could make the frames more or less concentrated, and $\tau = 1$ is used for every model and dataset. The answer frames use the answer-frame cap of Sec. 4.1, each after the label "Frame b:". The answer is read the same way, with the option letters as labels. Each option y is shown with a letter c_y (*e.g.*, "A"), and the answer is the most likely letter,

$$\hat{y} = \arg\max_{y}\ p_\theta(c_y \mid X_{\text{ans}}), \tag{6}$$

where X_{ans} holds the labeled answer frames, the question and the options.

3.4. When: the evidence time

An assistant should also show the moment it used. After the answer, EgoJev adds *"Which frame shows what is needed to answer the question?"* to the same context and reads the probabilities of the B frame labels as in Eq. (2). The evidence time t^* is the time of the most likely frame. This adds 41 text tokens and no visual tokens.

Table 1. **Accuracy and time** (Qwen3.5-4B, 8 answer frames). Avg.: mean of the three datasets. # frames: frames the selector reads per question, with T the video length in seconds. Time: seconds per question on QAEgo4D. On EgoSchema EgoJev uses the label check (Sec. 4.1). In all tables, best is bold and second best is underlined.

Method	# frames	Extra model	Time (s)	QAEgo4D	MyEgo	EgoSchema	Avg.
Uniform	–	none	1.7	51.2	36.1	**65.2**	50.8
AKS [25]	1 fps$\times T$	BLIP	90.7	56.4	39.4	60.4	52.1
BOLT [17]	1 fps$\times T$	CLIP	52.9	54.2	41.3	63.6	53.0
A.I.R. [41]	1 fps$\times T$	CLIP	82.9	<u>60.8</u>	<u>41.9</u>	63.4	<u>55.4</u>
Write IDs [2]	32	none	7.5	55.8	39.8	<u>64.8</u>	53.5
FOCUS [40]	230	BLIP	49.0	56.2	40.3	62.2	52.9
EgoJev	128	none	26.7	**62.0**	**42.6**	**65.2**	**56.6**

Table 2. **Same candidates, and EgoJev without options** (Qwen3.5-4B, 8 answer frames). AKS, BOLT and Write IDs choose from the 32 thumbnails that EgoJev reads first. The last two rows read all 128 thumbnails. EgoJev without options is read without the label check.

Method	QAEgo4D	MyEgo	EgoSchema
AKS, same 32 thumbnails	54.8	40.0	63.2
BOLT, same 32 thumbnails	53.0	39.3	62.6
Write IDs, same 32 thumbnails	55.8	39.8	64.8
EgoJev single level, same 32 thumbnails	57.2	42.2	<u>65.2</u>
EgoJev without options	<u>59.0</u>	<u>42.5</u>	**65.8**
EgoJev	**62.0**	**42.6**	<u>65.2</u>

Cost. EgoJev runs $1 + k$ localization passes and one answer pass per question, and generates no text. Because the thumbnails come before the question, the coarse thumbnails can be encoded and cached while the video is recorded, so coarse localization at question time only reads the question.

4. Experiments

Secs. 4.2 and 4.3 compare EgoJev with uniform sampling and recent training-free frame selectors on four VLMs. Secs. 4.4 and 4.5 then test the idea behind it and each of its parts, in the order of Sec. 3. Last, Secs. 4.6 and 4.7 measure the evidence time and the cost on a device.

4.1. Setup

Benchmarks. All three benchmarks are egocentric and multiple choice, which matches the wearable setting. QAEgo4D [5] asks about objects and actions in Ego4D [11] videos of about 8 minutes and marks the time window of the evidence. Its multiple-choice test set, as released by GroundVQA [7], has 500 questions over 148 videos. MyEgo [31] asks about moments long before the question and marks the answer moment. The evaluation uses all 996 of its questions asked at least 15 s after that moment, and the model sees the video only up to the question. EgoSchema [18] asks about whole 3-minute clips and has no evidence annotation. The evaluation uses its public subset of 500 questions.

Baselines. *Uniform* samples B frames evenly, including the first and last frame. *AKS* [25] scores every frame at 1 fps with BLIP and balances relevance and coverage. *BOLT* [17] scores every frame at 1 fps with CLIP and samples frames in proportion to the sharpened scores. *A.I.R.* [41] filters frames with CLIP and lets the VLM rate them over up to six rounds. *FOCUS* [40] searches clips as bandit arms with BLIP scores. *Write IDs* lets the same VLM write the labels of the relevant thumbnails among the 32 that EgoJev reads first, as TCoT [2] does, with up to 256 generated tokens. Each written label places a frame in its segment, and evenly spaced frames fill the rest of the 8. AKS, BOLT, A.I.R. and FOCUS run with their official code, settings and queries.

Metrics. Accuracy is the share of correct answers. For localization, a hit means that the top segment overlaps the evidence, and a time counts as correct within 8 s of the evidence.

Implementation details. The main VLM is Qwen3.5-4B [20], kept frozen, and the same VLM answers for every method. Every method answers from $B = 8$ labeled frames with the same cap on visual tokens per frame: 384 on QAEgo4D, 88 on EgoSchema and 72 on MyEgo. Apart from these shared caps, every method keeps its own settings fixed across datasets and models. Time is the mean end-to-end time per question over 50 QAEgo4D questions on one A100 MIG slice, including frame scoring. EgoJev uses $N = 32$ thumbnails, $W = 4$ windows, $k = 3$ refined windows and $\tau = 1$. VLMs can favor some option labels whatever the content [39]. When a short test on 40 questions shows that the coarse readout follows the label numbers, it is also read in reverse order and the two are combined by

Table 3. **Other VLMs** (8 answer frames, same settings as Tab. 1). QA: QAEgo4D, My: MyEgo, ES: EgoSchema, Avg.: mean of the three.

	Qwen3-VL-8B [19]				MiniCPM-V-4.5 [35]				GLM-4.1V-9B [10]			
Method	QA	My	ES	Avg.	QA	My	ES	Avg.	QA	My	ES	Avg.
Uniform	55.4	36.1	68.8	53.4	53.6	34.9	53.8	47.4	53.0	33.1	61.8	49.3
AKS (1 fps)	60.0	40.7	66.0	55.6	59.6	39.8	49.2	49.5	57.4	**39.1**	59.0	51.8
BOLT ($\alpha = 2.5$)	54.0	40.4	68.8	54.4	57.4	39.3	**54.0**	50.2	53.2	37.7	**63.0**	51.3
Write IDs	57.8	42.8	68.8	56.5	56.8	38.6	51.6	49.0	55.8	36.9	61.6	51.4
FOCUS	58.2	40.8	66.0	55.0	**61.2**	38.7	51.4	50.4	56.4	39.0	61.6	52.3
EgoJev	**60.4**	**44.0**	**69.0**	**57.8**	57.2	**41.9**	53.8	**51.0**	**58.0**	38.9	62.2	**53.0**

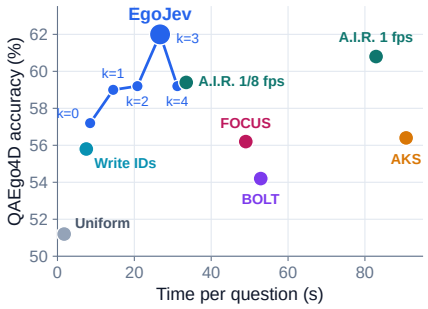

Figure 3. **Accuracy vs. time per question** (QAEgo4D, Qwen3.5-4B, 8 answer frames, one A100 MIG slice). Baselines at the settings of Tab. 1, and A.I.R. also at 1/8 fps (59.4% in 33.5 s). The EgoJev line refines $k = 0$ to 4 windows. Time: mean end-to-end time over 50 questions.

their geometric mean, as in permutation debiasing [39]. We call this the label check. With all four VLMs this fires only on EgoSchema. AKS uses its official candidates and default thresholds, and BOLT, which has no setting for QAEgo4D or MyEgo, uses $\alpha = 2.5$, its setting for EgoSchema.

4.2. Main results

Tab. 1 compares choosing with the answering VLM against choosing with a separate scorer. EgoJev is the most accurate on QAEgo4D and MyEgo (62.0% and 42.6%). It is also 3.1 times faster per question than A.I.R., which uses a second model. On EgoSchema, which asks about whole clips, every selector that uses a second model falls below uniform sampling, while EgoJev with the label check matches it. Given the same 32 thumbnails, AKS, BOLT and Write IDs all stay below single-level EgoJev on all three datasets (Tab. 2). Reading the choice thus beats both similarity scores and written IDs. Without options in its prompt, EgoJev still leads AKS and BOLT (Tab. 2), and options do not help AKS or BOLT on QAEgo4D (56.0% and 52.6%, Sec. D). EgoJev also puts an answer frame within 8 s of the evidence for 67.2% of QAEgo4D and 66.7% of MyEgo questions, against at most 59.2% and 58.0% for the baselines. Its lead over the best baseline is largest when the evidence lasts less than 3 s (62.6% against 58.0%, Sec. G).

Table 4. **How to ask.** Segment hit rate with the same 16 candidate frames (one per segment). The first three rows use the same VLM (Qwen3.5-4B). Time: localization seconds per question on QAEgo4D.

Scorer over 16 candidates	QAEgo4D	MyEgo	Passes	Time (s)
VLM, choose one (EgoJev)	**28.0**	**18.5**	1	0.44
VLM, rate each ("Yes")	18.2	13.5	16	1.41
VLM, write a span	15.8	12.6	1 + decode	1.68
BLIP image-text matching (AKS)	19.0	12.9	16	1.32
CLIP similarity (BOLT)	15.6	12.0	16	**0.13**
Random segment	8.9	6.2	–	–

Accuracy and time. Fig. 3 plots accuracy against the end-to-end time per question. EgoJev reaches 62.0% in 26.7 s, while AKS and BOLT take 53 to 91 s at their published settings.

4.3. Other VLMs

To test whether the gains hold beyond one model, Tab. 3 runs uniform sampling, AKS, BOLT, Write IDs, FOCUS and EgoJev on VLMs from three families with the settings of Tab. 1. Over the three datasets, the mean accuracy of EgoJev is higher than the better of AKS and BOLT by 2.2 points with Qwen3-VL-8B, 1.2 with GLM-4.1V-9B and 0.8 with MiniCPM-V-4.5 (3.6 with Qwen3.5-4B). It is also higher than that of FOCUS and Write IDs with all three. With Qwen3-VL-8B it is above AKS and BOLT on every dataset.

4.4. How to ask the VLM where to look

Tab. 4 asks whether the way the VLM is asked changes how well it finds the evidence. Only the question changes: the VLM chooses one of the candidate frames, rates each frame

Table 5. **Coarse-to-fine depth** k (Qwen3.5-4B, number of refined windows out of 4, 8 answer frames, without the label check). Time: end-to-end per question, as in Tab. 1. +32 at quantiles: a second readout over 32 thumbnails placed at the quantiles of P^c instead of in windows.

k	QAEgo4D	MyEgo	EgoSchema	Mean	Time (s)
0 (1 level)	57.2	42.2	**65.2**	54.9	8.5
1	59.0	41.9	64.2	55.0	14.5
2	59.2	43.0	65.0	55.7	20.8
3	**62.0**	42.6	62.8	**55.8**	26.7
4	59.2	43.4	63.4	55.3	31.2
+32 at quantiles of P^c	59.2	41.8	64.4	55.1	13.6
One readout, 128	58.6	**43.5**	65.0	55.7	26.4
3, plain product	58.0	**43.5**	62.6	54.7	26.7

Table 6. **Frame layout** from the single-level readout (Qwen3.5-4B, 8 answer frames). Focus: 6 frames in the top segment and 2 uniform. Spread: the relevance and coverage rule of AKS on P^c. Strata: one frame per eighth of the video, placed by a second readout within that eighth.

Layout	QAEgo4D	MyEgo	EgoSchema	Mean
Focus	**58.6**	38.7	61.8	53.0
Spread	56.2	**42.4**	63.2	53.9
Strata	56.6	40.4	**65.2**	54.1
Quantiles (EgoJev)	57.2	42.2	**65.2**	**54.9**

alone by the probability of "Yes", or writes a time span. A method finds the evidence when the segment it ranks first overlaps the evidence window. The test uses 16 candidates, the middle frames of 16 equal segments. Choosing finds the evidence segment in 28.0% of QAEgo4D and 18.5% of MyEgo questions. Rating each frame alone finds it in 18.2% and 13.5%, and writing a span in 15.8% and 12.6%. BLIP and CLIP, the scorers of AKS and BOLT, reach only 19.0% and 15.6% on QAEgo4D. With the 32 candidates of EgoJev, the order is the same on QAEgo4D: 28.6% for choosing, against 21.0% for rating, 17.8% for a span, and 19.4% and 18.8% for BLIP and CLIP.

Input format. The readout needs the model to tell candidate frames apart. In a simple test, one of 16 gray frames has a red square. Given as labeled images, Qwen3.5-4B and Qwen2.5-VL-7B [4] name the marked frame in 16 of 16 cases. Given as one video, they name it in only 3 and 2 cases. On QAEgo4D, giving the 32 thumbnails as one video lowers single-level accuracy from 57.2% to 53.2% (Sec. G).

4.5. Ablations

Tab. 5 varies how many windows are read at the fine level. On QAEgo4D, accuracy rises by 4.8 points from a single level to three refined windows, the setting used everywhere. A fourth window adds time and lowers the mean.

Table 7. **Evidence time.** Share of questions whose single predicted time is within 8 s of the evidence or inside it (QAEgo4D), and within 8 s / 30 s of the answer moment (MyEgo). Localization only: the time with the most probability in $P^\star$.

	QAEgo4D		MyEgo	
Method	$\pm$8 s	inside	8 s	30 s
AKS (1 fps, BLIP)	27.8	18.8	21.3	46.7
BOLT (1 fps, CLIP)	22.8	13.8	20.5	43.8
VLM, rate each	14.2	8.4	17.6	39.3
VLM, write a span	22.8	11.0	17.3	41.8
Random time	5.9	2.7	9.9	30.9
EgoJev, localization only	32.2	21.6	25.6	46.1
EgoJev, evidence readout	**33.6**	**22.6**	**26.6**	**50.4**

One readout over 128 thumbnails is 3.4 points lower on QAEgo4D at the same mean and time. Combining the two levels by a plain product instead of the geometric mean lowers QAEgo4D by 4.0 points and the mean by 1.1 (Tab. 5). For the 54% of QAEgo4D questions whose evidence no thumbnail shows, refinement raises the share with an answer frame within 8 s of the evidence from 45.7% to 54.6% (Sec. C).

Frame layout. Quantile placement is a known rule, so we test whether it suits the readout. Tab. 6 starts from the single-level readout and changes where the 8 answer frames go. Putting most frames in the top segment (Focus) helps QAEgo4D, where the evidence is brief, but costs 3.5 points on MyEgo and 3.4 on EgoSchema. Spreading them out helps elsewhere: Spread is best on MyEgo and Strata ties on EgoSchema, but both lose on QAEgo4D. The quantiles follow the readout: they gather when it is sure and spread when it is not, so they are first or second on every dataset and give the best mean.

4.6. Evidence time

Besides the answer, an assistant should show the moment it used (Sec. 3.4). Tab. 7 checks one time per method against the evidence, such as the frame it scores highest. The evidence readout of EgoJev costs 41 text tokens and lies within 8 s of the evidence for 33.6% of QAEgo4D questions, against 27.8% for the frame that AKS scores highest. In Fig. 4, only EgoJev answers the first question correctly, and for two questions about one video its readout peaks at each question's evidence.

4.7. On device

A wearable assistant records continuously and answers on the device.

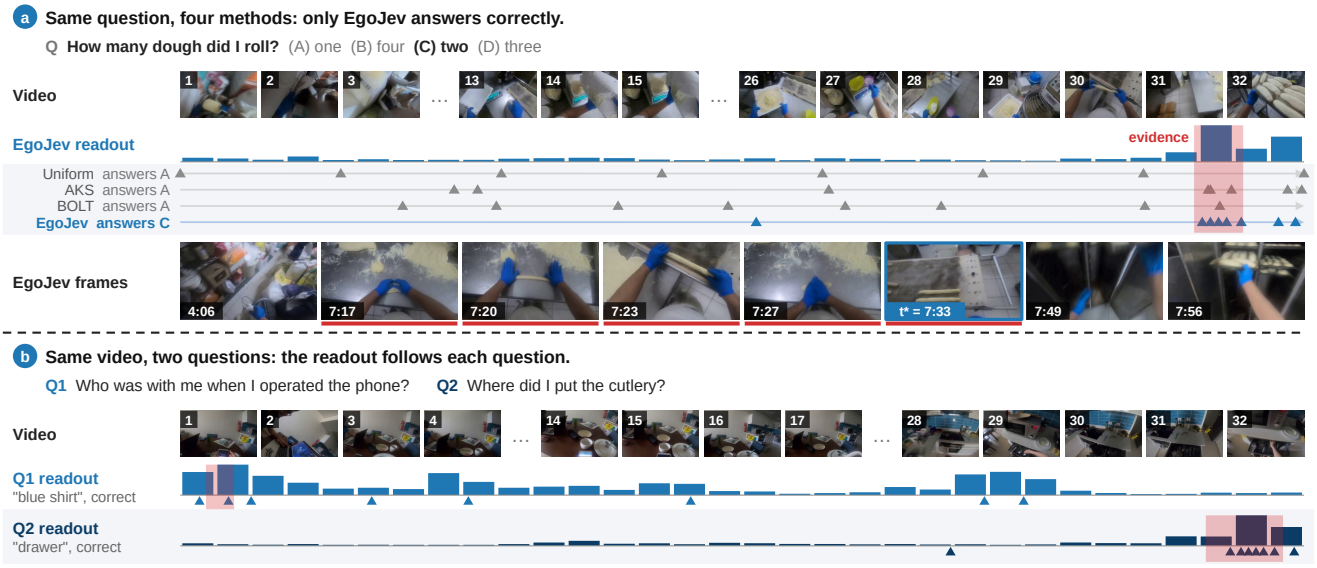

Figure 4. **Examples** (Qwen3.5-4B, QAEgo4D). (a) Four methods choose 8 answer frames (triangles) for one question. Only EgoJev answers correctly, and 5 of its 8 frames lie inside the evidence window (red band; underlined frames). Its coarse readout P^c over 32 segments (bars, square-root scale) peaks at this window. (b) For two questions about the same video, the readout (Q1 in blue, Q2 in dark blue) peaks at each question's evidence.

Table 8. **On device** (Mac, 4-bit weights, 6 GB memory cap, medians over 20 questions). EgoJev-S: single level. Rec.: compute while 8 minutes are recorded. Ans.: from question to answer. QA: QAEgo4D accuracy from GPU runs with the same frames. ‡iPhone 15 Pro, answer frames capped at 192 tokens, 10 questions.

Method	Extra model	Rec. (s)	Ans. (s)	Total (s)	QA
AKS, $B = 8$	BLIP 446M	53.8	13.6	67.4	56.4
BOLT, $B = 8$	CLIP 428M	28.3	10.4	38.7	54.2
EgoJev-S, $B = 8$	none	**8.2**	10.7	18.8	57.2
EgoJev-S, $B = 4$	none	**8.2**	**5.9**	**14.0**	**59.6**
EgoJev-S, $B = 4$, iPhone‡	none	18.3	8.0	26.2	58.0

To measure the cost there, every method runs on a Mac with 4-bit weights and a 6 GB memory cap, in one process from recording to answer (Tab. 8). While the video is recorded, EgoJev adds one thumbnail every 15 s to the VLM's own cache, which costs 8.2 s per 8 minutes. At their published settings, BOLT and AKS run a second model on every frame at 1 fps, which costs 28.3 s and 53.8 s. A.I.R. and FOCUS also run a CLIP or BLIP scorer over many frames, and on the GPU they take 82.9 s and 49.0 s per question, against 26.7 s for EgoJev (Tab. 1). With 8 answer frames, recording 8 minutes plus one question costs 18.8 s of compute, against 38.7 s for BOLT and 67.4 s for AKS. With 4 answer frames, the answer time halves (5.9 s against 10.7 s). Accuracy also rises to 59.6%, as fewer frames gather near the peak (like Focus in Tab. 6). On an iPhone 15 Pro the same pipeline costs 18.3 s for recording and 8.0 s per question, with 5.6 GB of memory at peak. On 36 to 72 minute recordings, EgoJev keeps an answer frame near the evidence for 44% of questions, against 12% for uniform sampling (Sec. G).

5. Conclusion

We asked whether a frozen VLM can find the evidence for an egocentric question by itself. It finds the evidence more often when it chooses among labeled frames and we read its choice from the output distribution. EgoJev refines this choice from coarse to fine, places a few answer frames at its quantiles, and returns the evidence time with the answer. With Qwen3.5-4B it is more accurate than recent training-free frame selectors on QAEgo4D and MyEgo, and a single-level version runs on a Mac and an iPhone.

Limitations. Like other training-free frame selectors, EgoJev can only find what the VLM recognizes: if the VLM does not recognize the evidence in a thumbnail, the readout cannot point to it. It also needs a VLM that takes interleaved images and returns the full probabilities of its output tokens, which most closed APIs do not.

References

[1] Diogo Almeida. Introducing System One models and Jev. TypeSafe AI blog, https://typesafe.ai/

blog/introducing-system-one-models-and-jev, 2026. 2, 3

[2] Anurag Arnab, Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. Temporal chain of thought: Long-video understanding by thinking in frames. In *NeurIPS*, 2025. 2, 3, 5

[3] Seyedarmin Azizi, Erfan Baghaei Potraghloo, and Massoud Pedram. Labels override definitions in Jev-style typed decision models. *arXiv preprint arXiv:2610.02586*, 2026. 3

[4] Shuai Bai et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 7

[5] Leonard Bärmann and Alex Waibel. Where did I leave my keys? Episodic-Memory-Based question answering on egocentric videos. In *CVPR Workshops*, 2022. 1, 2, 5

[6] Shimin Chen, Xiaohan Lan, et al. TimeMarker: A versatile video-LLM for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*, 2024. 3

[7] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *CVPR*, 2024. 2, 5

[8] Sunqi Fan, Qingle Liu, Runqi Yin, Meng-Hao Guo, and Shuojin Yang. Bridging VideoQA and video-guided agentic tasks via generalized keyframe extraction. *arXiv preprint arXiv:2606.29445*, 2026. 2

[9] Jonathan Geuter, Andrew Cheng, and David Alvarez-Melis. Releasing the inner Jev: A simple recipe for open decision models. *arXiv preprint arXiv:2610.17205*, 2026. 3

[10] GLM-V Team. GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025. 6

[11] Kristen Grauman et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1, 2, 5

[12] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower LLM to grasp video moments. In *CVPR*, 2024. 2

[13] Sanghwan Kim, Rui Xiao, Stephan Alaniz, Yongqin Xian, and Zeynep Akata. Training-free uncertainty guidance for complex visual tasks with MLLMs. *arXiv preprint arXiv:2510.00705*, 2025. 3

[14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[15] Ke Li, Jiayu Chen, Maoliang Li, Zihao Zheng, Hailong Zou, Hengyi Zhang, Xuanzhe Liu, and Xiang Chen. When and where to look: Adaptive visual evidence scheduling for efficient long video understanding. *arXiv preprint arXiv:2608.03918*, 2026. 2

[16] Yinheng Li, Justin Wagle, et al. LLM2Jev: LLMs are already Jev-style decision models. When and how to fine-tune them. *arXiv preprint arXiv:2610.02076*, 2026. 3

[17] Shuming Liu et al. BOLT: Boost large vision-language model without training for long-form video understanding. In *CVPR*, 2025. 2, 3, 4, 5

[18] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023. 2, 5

[19] Qwen Team. Qwen3-VL. https://github.com/QwenLM/Qwen3-VL, 2025. 6

[20] Qwen Team. Qwen3.5: Towards native multimodal agents. https://huggingface.co/Qwen, 2026. 5

[21] Alec Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[22] Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. FIRST: Faster improved listwise reranking with single token decoding. In *EMNLP*, 2024. 3

[23] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*, 2024. 2

[24] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? Investigating large language models as re-ranking agents. In *EMNLP*, 2023. 3

[25] Xi Tang et al. Adaptive keyframe sampling for long video understanding. In *CVPR*, 2025. 2, 3, 5

[26] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *ECCV*, 2024. 2

[27] Yilin Wang, Xiangxi Zheng, Dongxing Mao, Linjie Li, Zhengyuan Yang, Ping Yu, Rui Yan, Yuan Yao, and Alex Jinpeng Wang. Efficient frame selection for long videos at test time with attention-based MLLM selectors. In *Findings of EMNLP*, 2026. 2

[28] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. In *CVPR*, 2025. 2

[29] Yongliang Wu et al. Number it: Temporal grounding videos like flipping manga. In *CVPR*, 2025. 2

[30] Zhirong Wu, Xiaodong Wang, Langling Huang, Teng Xu, and Peixi Peng. A training-free framework for long video understanding via video-query-options similarity. In *ICLR*, 2026. 2

[31] Junbin Xiao, Shenglang Zhang, Pengxiang Zhu, and Angela Yao. Ego-grounding for personalized question-answering in egocentric videos. In *CVPR*, 2026. 2, 5

[32] Chuang Xu and Arno Berger. Best finite constrained approximations of one-dimensional probabilities. *Journal of Approximation Theory*, 244:1–36, 2019. 4, 1

[33] Jinhui Ye, Zihan Wang, Haosen Sun, et al. Re-thinking temporal search for long-form video understanding. In *CVPR*, 2025. 2, 4

[34] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *NeurIPS*, 2023. 2, 3

[35] Tianyu Yu, Zefan Wang, Chongyi Wang, et al. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe. In *CVPR*, 2026. 6

[36] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In *EMNLP*, 2024. 2

[37] Shaojie Zhang et al. Q-Frame: Query-aware frame selection and multi-resolution adaptation for video-LLMs. In *ICCV*, 2025. 2

[38] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *ICML*, 2021. 3

[39] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *ICLR*, 2024. 5, 6

[40] Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. In *ICLR*, 2026. 2, 4, 5

[41] 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 *ICLR*, 2026. 2, 5

EgoJev: Finding Evidence in Egocentric Video by Reading VLM Decisions

Supplementary Material

A. Proof of the quantile placement

Lemma. Let μ be a probability measure on $[0, T]$ with distribution function F and quantile function $F^{-1}(u) = \inf\{t : F(t) \geq u\}$. Among all measures $\nu = \frac{1}{B}\sum_{b=1}^{B} \delta_{t_b}$, the 1-Wasserstein distance $W_1(\mu, \nu)$ is minimized by $t_b = F^{-1}\big(\frac{b-1/2}{B}\big)$, $b = 1, \ldots, B$.

Proof. In one dimension, $W_1(\mu, \nu) = \int_0^1 |F^{-1}(u) - G^{-1}(u)|\, du$, where G^{-1} is the quantile function of ν. Without loss of generality $t_1 \leq \cdots \leq t_B$. Then $G^{-1}(u) = t_b$ for $u \in \big(\frac{b-1}{B}, \frac{b}{B}\big]$, so $W_1(\mu, \nu) = \sum_{b=1}^{B} \int_{(b-1)/B}^{b/B} |F^{-1}(u) - t_b|\, du$. The terms are independent, and the b-th term is $\frac{1}{B}\,\mathbb{E}|F^{-1}(U_b) - t_b|$ with U_b uniform on $\big(\frac{b-1}{B}, \frac{b}{B}\big]$. An absolute deviation is minimized at a median. Since F^{-1} is non-decreasing, $F^{-1}(\operatorname{med} U_b) = F^{-1}\big(\frac{b-1/2}{B}\big)$ is a median of $F^{-1}(U_b)$. The minimizers are non-decreasing in b, so the ordering assumption holds. □

Remarks. (i) If μ is uniform on $[0, T]$, then $t_b = \frac{(b-1/2)T}{B}$, which is evenly spaced sampling. (ii) The general result for prescribed weights is Theorem 5.5 of [32].

B. Fewer frames

We want to know how few answer frames EgoJev needs, since each frame costs time on a device. On QAEgo4D, single-level EgoJev with 4 frames is more accurate than AKS and BOLT with 8 frames at every resolution (Tab. 9).

Table 9. **4 frames vs. 8 frames** (QAEgo4D, Qwen3.5-4B, EgoJev single level). Cap: visual tokens per answer image.

Cap	Uniform, $B = 8$	AKS, $B = 8$	BOLT, $B = 8$	EgoJev, $B = 4$
88	48.4	<u>55.8</u>	51.2	**56.2**
192	51.4	<u>55.8</u>	54.6	**58.0**
384	51.2	<u>56.4</u>	54.2	**59.6**

C. Evidence between thumbnails

On QAEgo4D no thumbnail shows the evidence for 54% of the questions, whose evidence lasts 3 s at the median. For these questions, the coarse-to-fine readout raises the share with at least one answer frame within 8 s of the evidence from 45.7% to 54.6% (8 answer frames).

D. Options in the query

The localization prompt of EgoJev includes the answer options, while AKS and BOLT score frames against the question only, as in their code. Tab. 10 removes the options from EgoJev and adds them to the AKS and BOLT queries in the format that A.I.R. uses. Without options EgoJev stays above AKS and BOLT on every dataset, and on QAEgo4D the options do not help AKS or BOLT.

E. Temperature

The main tables use fixed settings for every method. Here we vary τ for EgoJev. It changes accuracy by at most 0.6 points over $\tau = 1$ when $\tau \leq 1$, and lowers it when $\tau > 1$ spreads the frames (Tab. 11).

Table 10. **Options in the query** (Qwen3.5-4B, 8 answer frames).

Method	QAEgo4D	MyEgo	EgoSchema
AKS, question	56.4	39.4	60.4
AKS, question + options	56.0	–	–
BOLT, question	54.2	41.3	63.6
BOLT, question + options	52.6	–	–
EgoJev, question	**59.0**	**42.5**	**65.8**

Table 11. **Temperature** τ (Qwen3.5-4B, single level, 8 answer frames).

τ	0.25	0.5	1	2	4
QAEgo4D	57.2	**57.8**	57.2	55.8	54.4
MyEgo	**42.3**	41.7	42.2	40.6	39.9

F. Model notes and label check

Model notes. Every localization label ends with "]" so that no label is a prefix of another. For every model we score label j as the tokens of "j]". GLM splits this into the label token and a separate "]" token, so its merged `]</` token is never scored. GLM-4.1V-9B, released as GLM-4.1V-9B-Thinking, writes prose after its default answer prefix, so its answer prefill adds its box token. The localizer prefill is unchanged. Thinking is turned off for Qwen3.5 and MiniCPM-V-4.5.

Label check. The test shifts the labels by $N/2$ and correlates the new coarse readout with the original one, matched once by label number and once by image. The check fires when the match by number is higher. It then reads the coarse readout once more with the labels in reverse order and combines the two by their geometric mean. Because the check reads no answers, all results use the full test sets. Answer letters follow the option format of each dataset, "(A" on QAEgo4D and "A" otherwise, for every method. For models that also offer the other form, we score whichever form has more total probability.

G. Additional results

Evidence coverage. Tab. 12 reports, for the methods of Tab. 1, how often an answer frame lies within 8 s of the evidence, and QAEgo4D accuracy by the length of the evidence window. EgoJev has the highest coverage on both datasets, and its lead in accuracy is largest on the briefest evidence.

Table 12. **Evidence coverage** (Qwen3.5-4B, 8 answer frames). Near: share of questions with an answer frame within 8 s of the evidence (QAEgo4D) or of the answer moment (MyEgo). Right: QAEgo4D accuracy by the length of the evidence window, annotated in whole seconds, with the number of questions.

	Near		QAEgo4D accuracy by evidence length		
Method	QAEgo4D	MyEgo	<3 s (131)	3–10 s (219)	>10 s (150)
Uniform	41.6	49.6	53.4	47.5	54.7
AKS	53.8	55.8	55.7	57.5	55.3
BOLT	56.2	58.0	54.2	54.8	53.3
A.I.R.	59.2	55.4	58.0	**62.6**	60.7
Write IDs	55.2	51.7	56.5	55.3	56.0
FOCUS	55.2	55.8	53.4	58.9	54.7
EgoJev	**67.2**	**66.7**	**62.6**	61.6	**62.0**

Long recordings. A wearable records far longer than one QAEgo4D clip. To test this, each of 100 QAEgo4D questions is asked about its own video joined with five other test videos, which gives recordings of 36 to 72 minutes (49 on average). EgoJev keeps $N = 32$ thumbnails, so its time per question stays the same (5.95 s against 5.82 s of GPU time, without

decoding). It remains the most accurate, and it keeps an answer frame near the evidence almost four times as often as uniform sampling (Tab. 13).

Table 13. **Long recordings** (Qwen3.5-4B, 8 answer frames, 100 QAEgo4D questions). Each video is joined with five other test videos. Near: share of questions with an answer frame within 8 s of the evidence.

	Accuracy		Near	
Method	Original	Long	Original	Long
Uniform	51.0	47.0	43.0	12.0
EgoJev, single level	52.0	44.0	59.0	33.0
EgoJev	**56.0**	**51.0**	**66.0**	**44.0**

Time since the moment. MyEgo asks about moments that passed 15 s to 11 minutes before the question. Tab. 14 splits its questions into quartiles of this gap. EgoJev is the most accurate or tied in every quartile, and it has a frame within 8 s of the moment more often than any other method (Near).

Table 14. **MyEgo by time since the answer moment** (Qwen3.5-4B, 8 answer frames, quartiles over 996 questions). Near: share of questions with an answer frame within 8 s of the moment.

		Accuracy				Near			
Gap (min)	n	Uni.	AKS	BOLT	EgoJev	Uni.	AKS	BOLT	EgoJev
0.25–0.37	254	35.0	36.6	39.4	**43.3**	45.3	64.2	59.4	**73.6**
0.37–0.67	247	36.8	40.5	**41.3**	**41.3**	49.4	68.0	65.2	**73.3**
0.67–1.60	246	34.6	41.1	41.9	**42.7**	60.6	51.2	65.9	**67.5**
1.60–10.7	249	38.2	39.4	42.6	**43.0**	43.4	39.8	41.8	**52.2**

Localization input. Tab. 15 changes how the single-level readout sees the thumbnails. Giving them as one video instead of labeled images costs 4.0 points, and full-resolution thumbnails cost 0.6 points and three times the localization tokens.

Table 15. **Localization input** (QAEgo4D, Qwen3.5-4B, single level, 32 segments, 8 answer frames).

Variant	QAEgo4D
Labeled thumbnails (EgoJev)	**57.2**
Full-resolution thumbnails	56.6
Thumbnails as one video	53.2

H. Timing protocol

We time every method end to end on the same 50 QAEgo4D questions on one A100 MIG slice (3g.20gb), after one warm-up question. Time covers decoding, preprocessing, frame scoring and selection, and answering. For the accuracy runs, A.I.R. selects frames on one A100 80GB GPU, and its answers run on the same GPU as those of the other methods.

I. On-device details

We want to know where the time goes on the device, so Tab. 16 splits the time from question to answer (Mac, 4-bit, 6 GB cap, the 20 questions of Tab. 8, median). Most of it is the answer pass and decoding the answer frames. Localization reads only the question against the cached thumbnails.

Table 16. **Time per question on the device** (s, median, single level). Rest: question prefill on the cache, localization readout and evidence readout. Each row is a median, so the total can differ slightly from Tab. 8.

	Mac, $B = 4$	Mac, $B = 8$	iPhone, $B = 4$
Answer-frame decode and encode	1.3	2.6	1.8
Answer pass	3.7	7.3	4.7
Rest	0.9	0.7	1.5
Question to answer	5.9	10.6	8.0
Peak memory (GB)	5.6	5.6	5.6

Readout variants. The localization readout scores every label sequence. Scoring them one by one takes 627 ms per question at 4.39 GB peak, batching the labels takes 562 ms at 5.38 GB, and one pass with an fp32 cache state takes 304 ms at 5.58 GB. One pass with a bf16 cache state takes 272 ms at 4.64 GB and picks the same top segment as scoring one by one on 50 of 50 questions. The iPhone (iPhone 15 Pro, 8 GB, iOS 26.6) uses this last variant, and the Mac runs in Tab. 8 use the fp32 state. The iPhone caps answer frames at 192 visual tokens.