FALCONEye: Finding Answers and Localizing Content in ONE-hour-long videos with multi-modal LLMs

Supplementary Material

9. FALCON-Bench: more details

9.1. Examples and video source details

The videos of our benchmark were sourced from three different public datasets:

- **S**occerNet [20] – 56 videos were selected from this dataset, each averaging 92.4 minutes in duration. These structured soccer match recordings include 389 questions. The mean GT time interval is 61.9 seconds.
- **M**ovieChat-1K Films [45] – a total of 140 film clips, each 8 minutes long, were selected from the dataset. Clips from the same film were combined to create 24 film segments, with an average duration of 46.4 minutes each. This subset contains 122 questions, with a mean GT time interval of 22.4 seconds.
- Walking **T**ours Dataset [48] – 12 high-resolution videos (3840 × 2160) were selected from this dataset, averaging 81.3 minutes. These videos capture city tours from an egocentric perspective and include 65 questions. The mean GT time interval is 31.4 seconds.

Overall, the benchmark comprises 575 questions, covering 4 categories, over 90 videos, with an **average video duration of 78.9 minutes** and **answers localized within a GT temporal window of 38.4 seconds**. The dataset is split into a test set (506 questions) and a validation set (70 questions). Figure 3 shows an example question of our benchmark across each dataset.

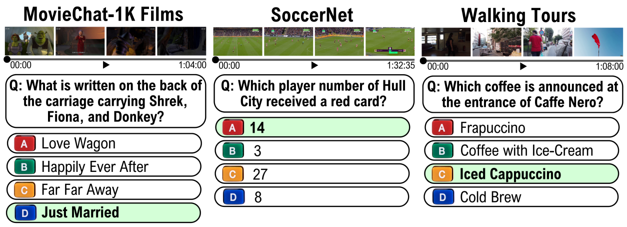


Figure 3. Falcon-Bench question examples for each dataset.

9.2. Question Categories

To design a benchmark for long VAS tasks, each question should have its answer contained within a single, short clip of the video. Based on this consideration, we defined four question categories:

- **Text Reading (TR)**: Questions ask about a piece of text that appears at a certain moment in the video.
- **Visual Observation (VO)**: Questions focus on visual attributes of the items appearing in the video, such as colors, textures, components, or materials.
- **Time Identification (TI)**: Questions about timestamps on clocks or alarms shown in the video.
- **Object Identification (OI)**: Questions focus on identifying specific objects within the video.

Figure 4 provides an overview of the number of questions per video type and per category, illustrating the distribution of tasks across the benchmark.

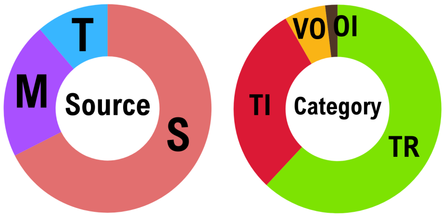


Figure 4. Distribution of questions in Falcon-Bench. The left plot, according to dataset sources: **M**ovieChat-1k, **S**occerNet, and Walking**T**ours. The right plot according to question category: TR, VO, TI, and OI.

9.3. Localization evaluation

FALCON-Bench requires models to provide an evidence of the answer (short clip) rather than precisely matching the entire clip where the answer is located. To achieve this goal, we leverage The Ground Truth over Union (GToU) metric to compare the predicted and the GT temporal interval. Unlike the commonly used Intersection over Union (IoU) in temporal grounding tasks [16], GToU assigns a score of 1.0 if the predicted interval is entirely within the GT interval, regardless of the degree of overlap. In all other cases, GToU behaves identically to IoU (Figure 5). In mathematical terms,

$$\mathbf{GToU} = \frac{|GT|}{|GT \cup \text{Pred}|} \cdot \mathbb{1}_{\{|GT \cap \text{Pred}| > 0\}}. \tag{2}$$

9.4. Human experiments

To evaluate human performance on our benchmark, we performed experiments with 10 participants. Each participant answered 10 questions based on 10 different videos (one question per video) and equally spread across the different dataset sources. Each participant answered 5 MCQs and 5 OQs.

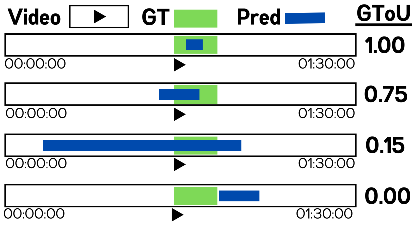


Figure 5. Visualization of GToU metric designed to measure the clip localization/retrieval task in which the answer is contained.

Table 8. Mean performance metrics of all participants across the three video types (MovieChat-1K, SoccerNet, WalkingTours).

Dataset	Accuracy		mGToU		Time (s)		Score (0-5)
	MCQs	OQs	MCQs	OQs	MCQs	OQs	OQs
MovieChat-1k	100.0	84.6	63.6	60.1	118.5	120.2	4.31
SoccerNet	100.0	95.0	79.5	89.2	55.2	67.4	4.80
WalkingTours	76.9	76.5	76.9	58.2	173.1	200.2	4.06

Methodology Participants were seated in front a test computer equipped with two monitors, one for watching the video and another for editing a .json file. They were provided with a structured .json file containing the question details.

Participants were required to fill in the `answer` and `temporal_window` fields. About the answer, if the `options` field is `null`, the answer was open-ended; otherwise, the answer was a letter in [A, B, C, D]. In the timestamp the participants had to indicate a short video clip where the answer is observed. A supervisor recorded the total time spent for each question, starting when the participant opened the video and stopping when they finished entering both required fields.

Results The results of the human experiments were analyzed in terms of both individual performance of each participant and the aggregated performance across all 10 participants, in terms of accuracy, mean GToU (mGToU) and spent time to answer. Figure 6 shows individual performance per participant for both MCQs and OQs. Participants 2, 3, 4, and 9 answered correctly all questions. Regarding the spent time, participant 3 was the fastest. Additionally, Table 8 shows the mean accuracy, mGToU, time and score (only for open-ended questions) of the answers across the three video types (MovieChat-1K, SoccerNet, WalkingTours). SoccerNet questions are generally easier to answer correctly, simpler to locate within the video, and quicker to respond to. In contrast, WalkingTours questions are the most challenging overall due to the extended duration and continuous nature of the videos.

10. FALCONEye: more details

This section gives further implementation details of our FALCONEye video agent explained in Sec. 3.

10.1. Exploration Algorithm pseudo-code

Algorithm 1 shows the pseudo-code of FALCONEye exploration algorithm (explained in Sec. 3.2 and Fig. 2).

Algorithm 1: FALCONEye Exploration Algorithm

```
Input: Video $\mathcal{V}$, Question $Q$
Output: Answer $\mathcal{A}$, Confidence $p(\mathcal{A})$, $Clip\ v \in \mathcal{V}$
Hyperparams: $it_{max}, dur_t$
Stage ① Pre-processing
$Clips \leftarrow$ Segment($\mathcal{V}$)
$\mathcal{C} \leftarrow$ VLM($Clips$)
$\mathcal{S} \leftarrow$ LLM($\mathcal{C}$)
$AllAns \leftarrow [\ ]\ ;\ it \leftarrow 0$
while $it \leq it_{max}$ do
    Stage ② Reasoning
    $Cand\_Clips \leftarrow$ LLM($Q, Captions$)
    $Cand\_Captions \leftarrow Cand\_Clips.Captions$
    while $dur(Cand\_Clips) \geq dur_t$ do
        $Ans \leftarrow [\ ]$
        Stage ③ Evaluation
        foreach $v_i^* \in Cand\_Clips$ do
            $\mathcal{A}_i^*, p(\mathcal{A}_i^*) \leftarrow$ VLM($Q, v_i^*$)
            $Ans.append(\{\mathcal{A}_i^*, p(\mathcal{A}_i^*), v_i^*, c_i\})$
            $it \leftarrow it + 1$
        end
        Stage ④ Decision
        $\mathcal{A}, Promis\_Clips \leftarrow$ LLM($Q, Ans$)
        if $\mathcal{A} \neq None$ then
            return $\mathcal{A}, p(\mathcal{A}), v$
        end
        $AllAns.append(Ans)$
        $Cand\_Clips \leftarrow$ Segment($Promis\_Clips$)
    end
    $\mathcal{C}.remove(Cand\_Captions)$
end
$\mathcal{A} \leftarrow$ LLM($Q, AllAns$)
return $\mathcal{A}, p(\mathcal{A}), v$
```

10.2. Prompts

Figure 8 shows the prompts sent to the LLM during the different stages of our FALCONEye exploration algorithm. Regarding the stages explained in Sec. 3.2: *Summary generation* from stage ① Pre-processing, *Select candidate clips from captions* from stage ② Reasoning, *Final answer or keep exploring* and *Select candidate clips as promising clips to keep exploring* from stage ④ Decision, and *Return final answer* when the maximum number of evaluated candidate clips is reached.

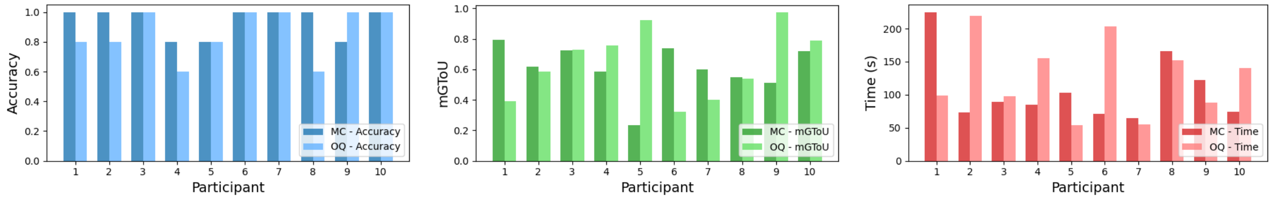


Figure 6. Performance metrics across all participants. Figures show accuracy, mean GToU (mGToU), and time spent per question.

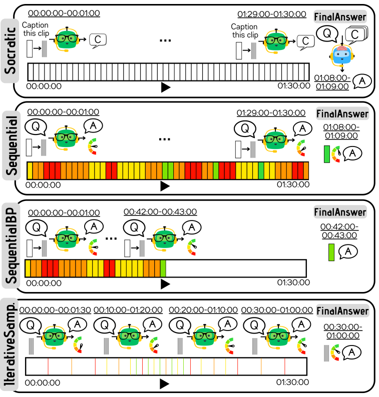
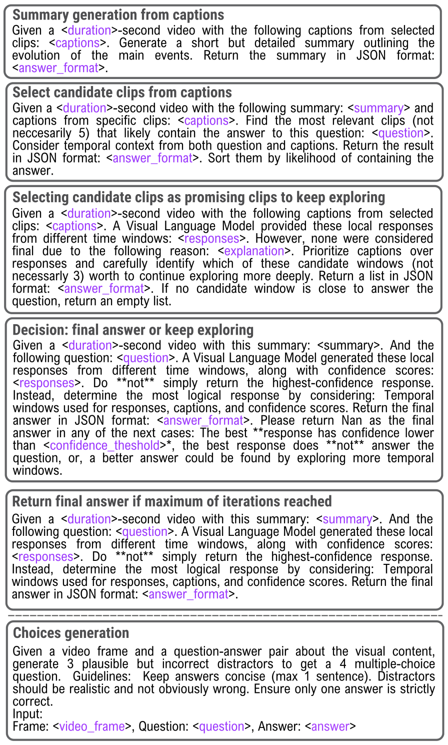


Figure 7. Visualization of the Socratic baseline approach together with our three exploration baselines considered to address VAS.



Summary generation from captions

Given a <duration>-second video with the following captions from selected clips: <captions>. Generate a short but detailed summary outlining the evolution of the main events. Return the summary in JSON format: <answer_format>.

Select candidate clips from captions

Given a <duration>-second video with the following summary: <summary> and captions from specific clips: <captions>. Find the most relevant clips (not neccesarily 5) that likely contain the answer to this question: <question>. Consider temporal context from both question and captions. Return the result in JSON format: <answer_format>. Sort them by likelihood of containing the answer.

Selecting candidate clips as promising clips to keep exploring

Given a <duration>-second video with the following captions from selected clips: <captions>. A Visual Language Model provided these local responses from different time windows: <responses>. However, none were considered final due to the following reason: <explanation>. Prioritize captions over responses and carefully identify which of these candidate windows (not necessarly 3) worth to continue exploring more deeply. Return a list in JSON format: <answer_format>. If no candidate window is close to answer the question, return an empty list.

Decision: final answer or keep exploring

Given a <duration>-second video with this summary: <summary>. And the following question: <question>. A Visual Language Model generated these local responses from different time windows, along with confidence scores: <responses>. Do **not** simply return the highest-confidence response. Instead, determine the most logical response by considering: Temporal windows used for responses, captions, and confidence scores. Return the final answer in JSON format: <answer_format>. Please return Nan as the final answer in any of the next cases: The best **response has confidence lower than <confidence_theshold>*, the best response does **not** answer the question, or, a better answer could be found by exploring more temporal windows.

Return final answer if maximum of iterations reached

Given a <duration>-second video with this summary: <summary>. And the following question: <question>. A Visual Language Model generated these local responses from different time windows, along with confidence scores: <responses>. Do **not** simply return the highest-confidence response. Instead, determine the most logical response by considering: Temporal windows used for responses, captions, and confidence scores. Return the final answer in JSON format: <answer_format>.

Choices generation

Given a video frame and a question-answer pair about the visual content, generate 3 plausible but incorrect distractors to get a 4 multiple-choice question. Guidelines: Keep answers concise (max 1 sentence). Distractors should be realistic and not obviously wrong. Ensure only one answer is strictly correct.
Input:
Frame: <video_frame>, Question: <question>, Answer: <answer>

Figure 8. LLM prompts used in FALCONEye algorithm and and FALCON-Bench.

11. Additional Experiments

11.1. FALCON-Bench

Given the poor performance of state-of-the-art VLMs on FALCON-Bench, we further evaluated VLMs under a simplified VAS setup. Specifically, we extracted 1-minute clips centered within the ground truth intervals to ensure that the answers were contained within these shorter segments (Table 9). Qwen2.5-VL achieved the best accuracy and score in OQs, and LLaVA-Video in MCQs. However, even after reducing the search space from an hour-long video to just one minute, the performance remains relatively low. To address this, we took an additional step and tested the VLMs by providing a single frame that contains the answer, which is always within the ground truth interval defined by FALCON-Bench (Table 10). These results represent the maximum performance our FALCONEye agent could achieve with each VLM, assuming the ground truth frame is the optimal frame for answering the question. In this evaluation, GPT-4o achieved the highest performance, followed by Qwen2.5-VL.

Table 9. Model performance on FALCON-Bench test split with GT 1min-length clips containing the answer. We report average accuracy for MCQs and OQs across MovieChat (M), SoccerNet (S), WalkingTours (T), and overall (Avg.).

	MCQs	**OQs**							
	Acc.	Accuracy				Score (0-5)			
Model Name	Avg.	M	S	T	**Avg.**	M	S	T	**Avg.**
Baselines									
Full Mark	100	100	100	100	100	5.00	5.00	5.00	5.00
Random	25.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Open-Source VLMs with Multi-Image Support									
LLaVa-v1.5	33.3	6.86	4.23	4.00	5.03	0.49	0.37	0.40	0.42
LLaVa-v1.6	41.9	9.80	23.4	8.00	13.7	0.58	1.33	0.58	0.83
mPLUG-Owl3 [58]	57.6	39.2	21.1	26.0	28.7	2.24	1.20	1.58	1.67
LLaVA-OV [22]	72.2	45.0	28.5	46.0	39.8	2.46	1.57	2.42	2.15
Qwen2.5-VL	75.9	64.7	52.2	64.0	60.3	3.49	3.08	3.26	3.27
Open-Source VLMs designed for Videos									
VideoChat2-HD [25]	27.2	18.6	12.1	22.0	17.5	1.24	0.67	1.34	1.08
Video-LLAVA [26]	34.0	19.6	6.49	18.0	14.6	1.20	0.46	1.14	0.93
LLaVA-Video [23]	79.2	61.7	44.0	52.0	52.6	3.31	2.38	2.74	2.81
Open-Source VLMs specific for long videos									
MovieChat-OV [45]	44.5	5.88	18.3	14.0	12.7	0.49	1.02	0.82	0.77
Apollo [70]	71.3	50.9	31.3	48.0	43.4	2.77	1.79	2.62	2.39
Meta-architectures built from a LLM (GPT4o-mini) and a VLM (Qwen2.5-VL)									
FALCONEye-Pro	81.7	63.7	70.0	75.0	66.7	3.39	3.51	3.56	3.49
FALCONEye-Flash	**81.9**	66.6	69.7	70.0	**68.8**	3.54	3.62	3.62	**3.59**

Table 10. Model performance on the test set providing the model with GT frames containing the answer. We report average accuracy for MCQs and OQs across MovieChat (M), SoccerNet (S), WalkingTours (T), and overall (Avg.).

	MCQs	**OQs**							
	Acc.	Accuracy				Score			
Model Name	Avg.	M	S	T	**Avg.**	M	S	T	**Avg.**
Baselines									
Full Mark	100	100	100	100	100	5.00	5.00	5.00	5.00
Random	25.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Proprietary Long-Context LLMs									
GPT-4o-mini (LR)	75.3	61.7	28.5	42.0	44.0	3.18	1.52	2.28	2.32
GPT-4o-mini (HR)	88.4	70.5	57.9	70.0	66.1	3.58	3.21	3.78	3.52
GPT-4o (LR)	83.7	78.0	47.7	62.0	52.5	3.94	2.63	3.32	3.29
GPT-4o (HR)	**94.1**	80.3	66.9	72.0	**73.0**	4.06	3.67	3.82	**3.85**
Open-Source VLMs for images									
LLaVa-v1.5	56.7	40.1	24.5	34.0	32.8	2.34	1.38	1.96	1.89
LlaVa-v1.6	78.5	53.9	48.5	62.0	54.8	3.08	2.51	3.42	3.24
mPLUG-Owl3 [58]	82.9	59.8	53.1	58.0	65.2	3.25	2.89	3.18	3.10
LLaVA-OV [22]	84.4	40.1	10.4	54.0	34.8	2.02	0.53	2.80	1.78
Qwen2.5-VL []	84.1	71.5	62.9	78.0	70.8	3.80	3.33	4.02	3.72
LLaMa3.2M [1]	85.9	69.6	64.9	70.0	68.1	3.68	3.27	3.70	3.55

11.2. Ablation study of exploration algorithm stages

We validate FALCONEye's exploration algorithm with an ablation study of its different stages, explained in Sec. 3.2, shown in Table 11. It is noted that each stage contributes significantly to achieving the FALCONEye's superior performance compared to the baselines.

We also perform a more detailed study of the influence of the zoom-in effect incorporated in our approach. Qwen2.5-VL can process images of any resolution, dynamically converting them into a variable number of visual tokens. This creates a trade-off between number of frames and frame resolution when processing videos within the context window size limit. This trade-off is analyzed in the validation split of FALCON-Bench by varying the clip length (Table 12). The best configuration per clip-level category is selected for FALCONEye.

Table 11. FALCONEye ablation study of the exploration algorithm. We compare the time and performance gain that each of the four stages of our exploration algorithm brings (**① Preprocessing**, **② Reasoning**, **③ Evaluation**, and **④ Decision**, as detailed in Sec. 3). To validate them, we first measure performance when giving the whole video to the VLM and the captions extracted in Stage ① to the LLM. Lately, we validate the stages adding them sequentially and comparing the reasoning stages ② and ④ vs random guess.

FALCONEye	**MCQs**			**OQs**			
Exploration Algorithm	s	Acc.	Loc.	s	Acc.	Sc.	Loc.
Video → VLM	68.9	23.4	0.00	70.1	11.3	0.71	0.00
Captions from ① → LLM	123.4	39.9	17.8	125.0	13.8	0.89	16.2
①+Random+③	174.9	26.1	3.16	165.2	9.92	0.60	4.06
①+②+③	181.2	51.9	22.1	171.6	34.8	1.90	21.2
①+②+③+Random	383.2	54.8	26.8	290.8	38.9	2.01	21.9
FALCONEYE:①+②+③+④	348.7	59.6	27.3	229.2	46.6	2.47	26.9

Table 12. Qwen2.5-VL performance comparison when varying the GT clip length of FALCON-Bench validation split.

Visual Information			**MCQs**				**OQs**							
Δt	Frames		Accuracy				Accuracy				Score			
s	#	Res.	M	S	T	Avg.	M	S	T	Avg.	M	S	T	Avg.
60	2fps	336 × 616	85.0	54.3	73.3	70.9	60.0	57.1	66.6	61.3	3.15	3.23	3.46	3.28
60	64	476 × 840	80.0	62.8	73.3	72.1	70.0	57.1	53.3	60.2	3.60	3.46	2.93	3.38
60	32	672 × 1204	85.0	71.4	86.7	**81.0**	60.0	60.0	66.7	**62.2**	3.15	3.54	3.53	**3.43**
5	2fps	824 × 1462	85.0	74.3	80.0	79.8	70.0	62.9	73.3	**68.7**	3.70	3.71	3.80	**3.73**
5	5	824 × 1462	85.0	80.0	93.3	**86.1**	55.0	65.7	66.7	62.5	2.95	3.69	3.53	3.39

11.3. VLMs Calibration

To measure VLM calibration, we adopt the Reliability Diagrams [13]. These diagrams group all the predictions in bins according to their confidence and measure the gap between confidence and accuracy per each bin. Specifically, we split the N predictions in $M = 10$ bins according to their confidence and, for each bin B_m, we compute its count N_m average confidence C_m and its average accuracy A_m. From these diagrams, we may compute the Average Calibration Error (ACE) as,

$$\text{ACE} = \frac{1}{M^+}\sum_{m=1}^{M} |C_m - A_m|, \quad \text{MCE} = \max_{m} |C_m - A_m| \tag{3}$$

where M^+ is the number of non-empty bins [34]. We compute Calibration Count (CC), which quantifies the percentage of predictions above a defined confidence threshold, weighted by their accuracy calibration error (1-CE). For example, the Calibration Count at threshold 0.9 is computed as,

$$\text{CC@0.9} = \frac{N_{10}}{N}\,(1 - |C_{10} - A_{10}|). \tag{4}$$

Confidence metrics-. As detailed in Sec.3.2, estimating answer confidence in OQs requires a metric to average token probabilities along the response (see Figure 9).

We evaluated various confidence aggregation methods, specifically likelihood, average, and geometric average, as done in similar calibration studies with LLMs [29]. Looking first at the reliability diagrams (Figure 10 (a-f)), we discard likelihood as a suitable confidence metric. The distribution of confidence values is highly spread out, with many answers assigned extremely low confidence. This occurs because likelihood multiplies the probability of all tokens, thus longer answers tend to receive lower confidence scores, making it unreliable for calibration.

Comparing the average and geometric average, the reliability diagrams do not show major differences between them. However, the calibration metrics in Table 13 indicate that **geometric average** results in lower Brier Score (BS), MCE, and ACE for both LLaVA-Video and Qwen2.5-VL, suggesting slightly better calibration performance.

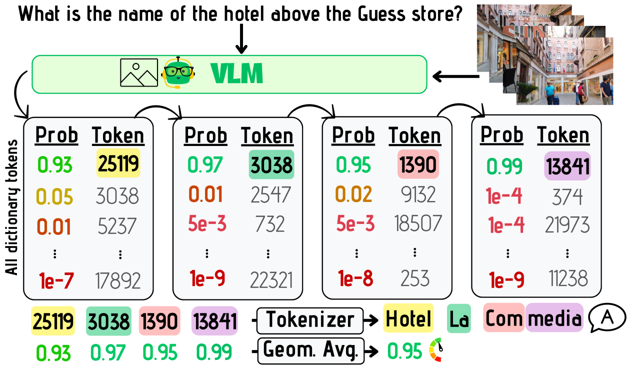


Figure 9. Given a question and a set of frames, **FALCONEye leverages** the **answer** outputted by the VLM but, and its **confidence** (geometric average through all output tokens probabilities).

Table 13. Calibration metrics comparison with GT 1min-length clips in the open-ended (OQs) questions test split of FALCON-Bench. Lower values for BS, MCE and ACE, are better.

Model Name	Δt	F	Likelihood			Average			Geom. Average		
	s	#	BS	MCE	ACE	BS	MCE	ACE	BS	MCE	ACE
LLaVA-Video	60	32	0.23	0.34	0.17	0.24	0.37	0.22	0.22	0.31	0.16
LLaVA-OV	60	32	0.26	0.92	0.41	0.28	0.39	0.29	0.26	0.59	0.29
Qwen2.5-VL	60	2fps	0.19	0.31	0.15	0.26	0.45	0.25	0.24	0.38	0.27

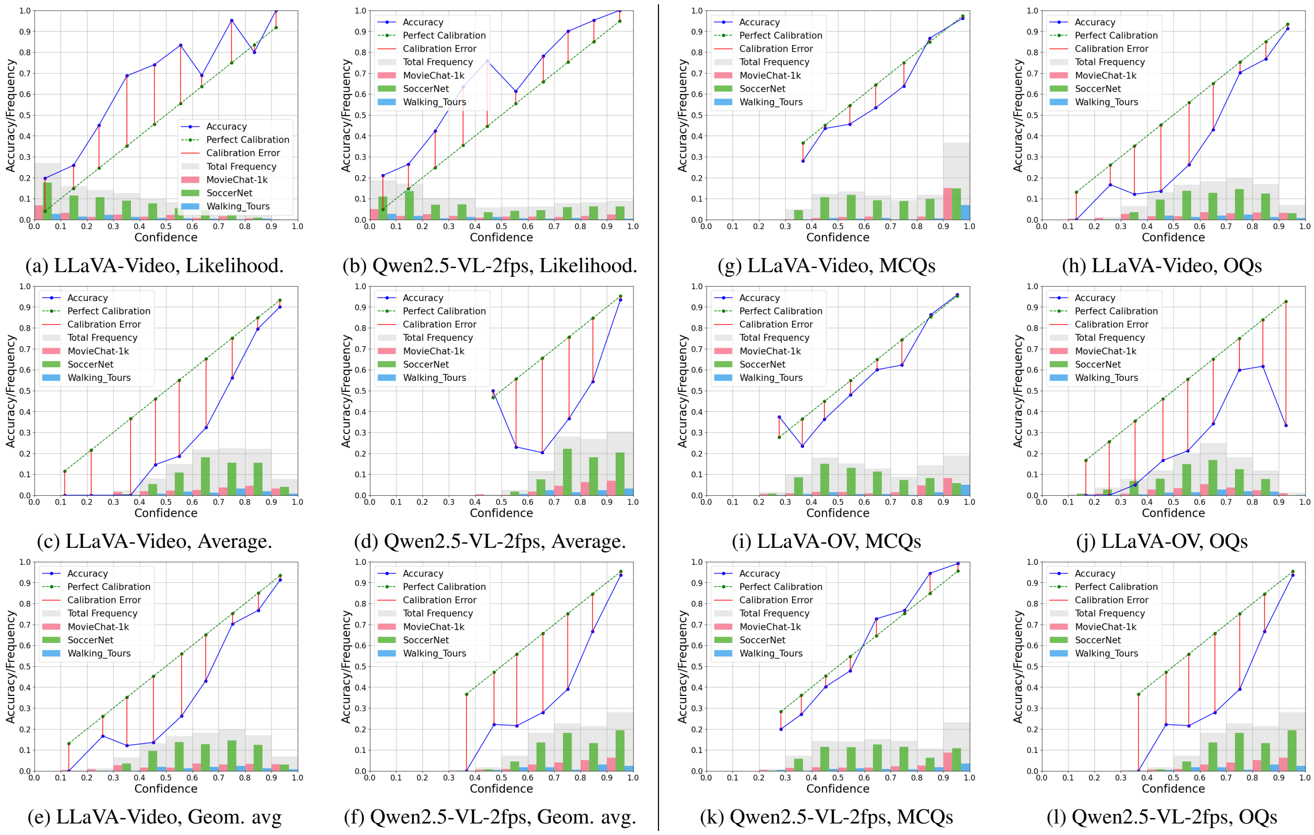
Models-. Regarding model selection, we compare reliability diagrams for LLaVA-Video, LLaVA-OneVision, and Qwen2.5-VL, evaluated on both MCQs and OQs. For MCQs, all models show strong calibration, with low calibration error, specially for high confidence values (Fig. 10 (g-l)). However, for OQs, the distribution of high-confidence answers differs significantly. Both LLaVA-Video and LLaVA-OneVision have a much smaller proportion of answers with 0.9 confidence, whereas Qwen2.5-VL not only produces a higher number of high-confidence answers but also demonstrates extremely low calibration error for those predictions.

11.4. FALCONEye vs GPT4o

Figure 11 shows three FALCON-Bench example questions and compare the responses from GPT4o and FALCONEye.

11.5. FALCON-Bench vs QAEgo4D

The QAEgo4D [5] dataset is the closest to FALCON-Bench in terms of the task definition as it also addresses the problem of VAS, that the authors call *question answering visual language grounding* (VLG). They also have open ended questions. However, besides the time differences in the video durations –FALCON-Bench with an average duration of $\sim$ 80 minutes while QAEgo4D has an average duration of $\sim$ 8 minutes–, the type of questions are completely different. QAEgo4D questions are automatically generated from the sparse video narration, focusing on the main object or action: *Q: What did I Put in the Pan?–A: cheese*, or *Q: What paint can did I open?–A:black paint*. Meanwhile, FALCON-Bench questions are human curated to be challenging: *Q: What is the number of the train that crosses paths with Lightning McQueen at night?–A:A113*; *Q:What message about the flu appears on a city building?–A:Bovril nourishes you to resist 'flu* or *Q:Which player of Chelsea received a red card?–A: Thibaut Courtois*.



(a) LLaVA-Video, Likelihood.

(b) Qwen2.5-VL-2fps, Likelihood.

(g) LLaVA-Video, MCQs

(h) LLaVA-Video, OQs

(c) LLaVA-Video, Average.

(d) Qwen2.5-VL-2fps, Average.

(i) LLaVA-OV, MCQs

(j) LLaVA-OV, OQs

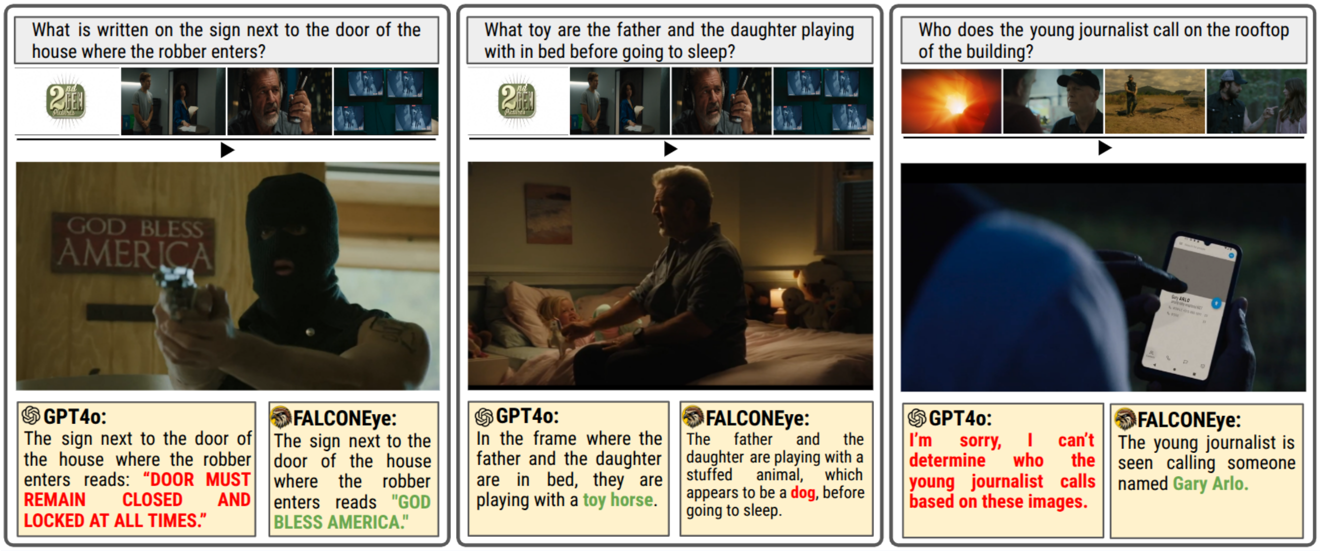
(e) LLaVA-Video, Geom. avg

(f) Qwen2.5-VL-2fps, Geom. avg.

(k) Qwen2.5-VL-2fps, MCQs

(l) Qwen2.5-VL-2fps, OQs

Figure 10. (a-f) Calibration plots for different probabilities aggregation metrics with the GT 1min clips of the FALCON-Bench test split, and (g-l) calibration plots when testing the VLMs with the GT 1min-length clips of the FALCON-Bench test split.



What is written on the sign next to the door of the house where the robber enters?

GPT4o: The sign next to the door of the house where the robber enters reads: "DOOR MUST REMAIN CLOSED AND LOCKED AT ALL TIMES."

FALCONEye: The sign next to the door of the house where the robber enters reads "GOD BLESS AMERICA."

What toy are the father and the daughter playing with in bed before going to sleep?

GPT4o: In the frame where the father and the daughter are in bed, they are playing with a toy horse.

FALCONEye: The father and the daughter are playing with a stuffed animal, which appears to be a dog, before going to sleep.

Who does the young journalist call on the rooftop of the building?

GPT4o: I'm sorry, I can't determine who the young journalist calls based on these images.

FALCONEye: The young journalist is seen calling someone named Gary Arlo.

Figure 11. Answer comparison between GPT4o and FALCONEye for three example questions, showing the frame that contain the answer.