VIDEOMEMBENCH: HOW WELL DO MLLMS REMEMBER IN VIDEO CONVERSATIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) can now process hour-long videos, making extended multi-turn dialogue over a single video a natural mode of use. Sustaining such a dialogue requires more than answering each question in isolation: the model must retrieve and reuse information established earlier in the conversation, both what it has seen and what it has said, an ability we term *in-context video memory*. Existing video QA benchmarks leave this ability unmeasured: their questions are mutually independent, never reference one another, and each is answerable from the video alone, so nothing ever requires carrying information across turns. We introduce **VideoMemBench**, in which each sample pairs a single video with a long multi-turn dialogue, whose turns are connected by typed cross-turn dependencies: an *Episodic Dependency* refers back to a visual episode described in an earlier turn, probing memory of *what the model has seen*, while an *Answer Dependency* builds on the model's own earlier answer, probing memory of *what the model has said*. Every dependency is verified considering two properties: *1) necessity*, meaning the question cannot be answered without its earlier turns, and *2) non-leakage*, meaning the question does not restate the information from its earlier turns. To enforce these properties, we design a construction pipeline with three LLM roles: a planner, a generator, and a judger, with each dialogue reviewed by human experts. VideoMemBench comprises 326 videos with 2,476 turns, of which 1,387 depend on earlier turns, including multi-parent dependencies and long-range call-back. Among over 20 evaluated open- and closed-source MLLMs, Claude Opus 5 performs the best with an overall score of only 45.7%, highlighting the challenge of in-context video memory. Our dataset and evaluation will be released.

1 INTRODUCTION

Frontier multimodal large language models (MLLMs) (Comanici et al., 2025; Singh et al., 2025; Team et al., 2026; Yang et al., 2025a; Anthropic, 2026a) can now process hour-long videos, with million-token contexts that accommodate an entire film, a full match, or a day-long recording. Building on this capability, extended multi-turn dialogue over a single long video is emerging as a natural mode of use: the model accompanies a user through the video with a stream of follow-up questions about its characters, events, and details. Sustaining such a conversation raises a key question: when consecutive questions concern the same video, can the model correctly retrieve and reuse what the dialogue has already established? We refer to this ability as *in-context video memory*.

Recently, in-context memory has been systematically evaluated for large language models (LLMs) in text-only settings (Maharana et al., 2024; Hsieh et al., 2024; Kuratov et al., 2024; Tan et al., 2025; Wu et al., 2025; Huet et al., 2025). However, extending this evaluation to video poses a key challenge: distinguishing questions that genuinely depend on prior dialogue context from those answerable using only the video and the current question. Existing video understanding benchmarks (Fu et al., 2025; Li et al., 2024; Mangalam et al., 2023; Wu et al., 2024; Wang et al., 2025; Fu et al., 2026) are structurally unable to evaluate it. Typically, they decompose the QA evaluation into mutually independent questions: no question references the content or answer of another, so nothing ever needs to be carried across turns, and every question is answerable from the video alone. In other words, their dialogue histories are never load-bearing.

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.

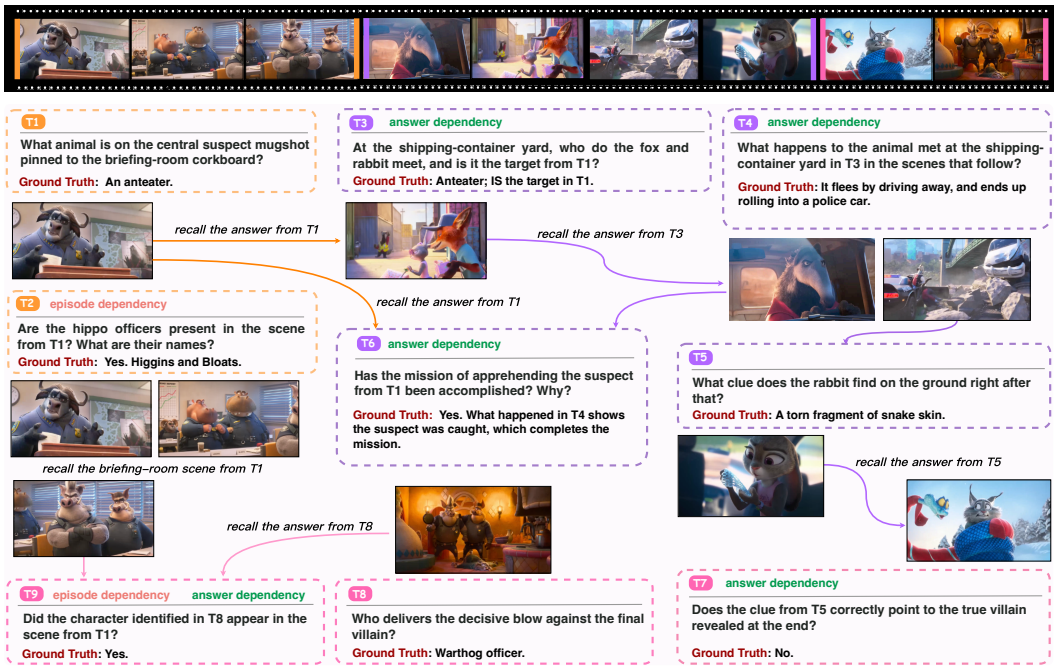


Figure 1: **Example of VideoMemBench with cross-turn dependencies:** Independent turns (*e.g.*, T_1, T_8) establish visually grounded facts; dependent turns refer back to earlier turns through expressions such as "the scene from T_1" or "the target from T_1", without restating earlier content. An *Episodic Dependency* (*e.g.*, T_2) requires recalling *what the model has seen*: the model must recover which scene T_1 described before answering the question. An *Answer Dependency* (*e.g.*, T_3, T_4) requires recalling *what the model has said*: each builds on the answer produced at an earlier turn.

To this end, we introduce **VideoMemBench**. Each sample pairs a single video with a long multi-turn dialogue whose turns are connected by explicit, *typed* cross-turn dependencies, so that a sample is not a concatenation of separate questions but a memory chain running through the entire conversation. The key design is that answering a dependent turn should require the dialogue and the video jointly: the dialogue supplies what an earlier turn established, and the video supplies the visual facts to ground the current one, so that neither source alone suffices. Specifically, we design two dependency types, enforcing two forms of memory: *Episodic Dependency*, for recalling previously discussed visual episodes, and *Answer Dependency*, for recalling the model's own earlier responses.

(1) Episodic Dependency arises when a question depends on the visual episode that an earlier turn described: only by recalling that referent from the dialogue can the model determine what is being asked. As shown in Figure 1, T_1 asks about the suspect mugshot pinned to the briefing-room corkboard, and T_2 then asks whether the hippo officers are present in "the scene from T_1" and what their names are, without restating where that scene is. Unless the model retains that T_1 was about the briefing room, it cannot tell which part of the video "the scene from T_1" refers to, let alone check who is in it. Episodic dependencies thus hinge on memory of *what the model has seen*.

(2) Answer Dependency arises when a question builds on the answer the model itself gave at an earlier turn: the model must continue from its own earlier response. As shown in Figure 1, T_1 establishes that the suspect is an anteater, T_3 then asks whether the animal that fox and rabbit meet at the shipping-container yard is "the target from T_1", which cannot be decided without the answer given at T_1. T_4 in turn asks what happens next to "the animal met in T_3", building on the answer to T_3. Answering thus demands two abilities at once: *retrieval*, locating the earlier answer within the accumulated dialogue, and *consistency*, keeping the current answer faithful to it rather than silently re-judging the video. Answer dependencies thus hinge on memory of *what the model has said*.

Furthermore, as shown in the closing T_9 of Figure 1, VideoMemBench stresses memory along two additional dimensions: a turn may depend on several earlier turns at once, requiring parallel recall of multiple episodes (T_1) or answers (T_8); and a dependency may span eight turns ($T_1 \rightarrow T_9$), keeping content from the beginning of the dialogue live throughout.

Table 1: **Comparison with existing benchmarks.** *Max Dur.* is the maximum video duration in the benchmark. *Multi-turn* indicates multi-turn dialogue evaluation. *Max Turns* is the largest number of QA turns in one dialogue. *Dep. Graph* denotes structured dependency graphs for individual turns; *Typed* distinguishes answer and episodic dependencies; *M-Parent* indicates multiple parents.

Benchmark	Input	#Videos	#Dialogs	#QAs	Max Dur.	Max Turns	Multi-turn	Dep. Graph	Typed	M-Parent
MVBench (Li et al., 2024)	Video	3,641	–	4,000	3 min	1	✗	✗	✗	✗
Video-MME (Fu et al., 2025)	Video	900	–	2,700	1 h	1	✗	✗	✗	✗
LVBench (Wang et al., 2025)	Video	103	–	1,549	2 h	1	✗	✗	✗	✗
MMDU (Liu et al., 2024b)	Images	–	110	1,645	–	27	✓	✗	✗	✗
MT-Video-Bench (Pan et al., 2026)	Video	135	1,000	5,887	20 min	8	✓	✗	✗	✗
VideoMemBench (ours)	**Video**	**326**	**326**	**2,476**	**23.47 h**	**21**	✓	✓	✓	✓

These structurally diverse dependencies ensure that answering questions from our **VideoMemBench** requires the model to exercise *in-context video memory* throughout the dialogue. We further verify every dependent turn for *necessity* and *non-leakage*: the question must be unanswerable without its parent turns, and must not restate the information they established before. In total, VideoMemBench comprises 326 videos, each paired with a human-reviewed dialogue, totaling 2,476 turns, of which 1,387 depend on earlier turns. Videos are drawn from five existing benchmarks and the web, covering 31 semantic domains across four categories (lifestyle & entertainment, knowledge & education, art & literature, sports & competition), and last up to 23.47 hours each. The comparison in Table 1 shows that single-turn benchmarks evaluate questions independently, whereas multi-turn benchmarks extend evaluation to dialogues. VideoMemBench further provides explicit dependency graphs with typed edges and multiple parents, enabling structured evaluation of in-context video memory.

Evaluating over 20 representative open- and closed-source MLLMs on VideoMemBench, we find that current models clearly struggle with cross-turn dependencies: the best model we tested (*Claude Opus 5*) achieves an overall score of only 45.7%, with 35.7% on answer-dependent turns and 49.1% on episodic-dependent turns, both below 53.9% on independent turns. These results indicate that in-context video memory remains an open problem for current MLLMs.

Our main contributions are summarized as follows:

- We study *in-context video memory*, the ability of a model to retrieve and reuse what earlier turns have established in a consecutive video QA dialogue, which existing benchmarks leave unmeasured.
- We build **VideoMemBench**, in which each sample is a long multi-turn dialogue grounded in a single video, with turns connected by typed cross-turn *Episodic* and *Answer* dependencies. VideoMemBench comprises 326 video dialogues, totaling 2,476 turns with 1,387 dependent, verified for necessity and non-leakage, and reviewed by human experts.
- We conduct extensive evaluations of over 20 open- and closed-source MLLMs on VideoMemBench. The best model achieves only 45.7% overall, with performance declining as dependency complexity increases, indicating that in-context video memory remains challenging for current MLLMs.

2 Related Works

Video Understanding Benchmarks. Existing video understanding benchmarks usually focus on temporal perception and reasoning. MVBench (Li et al., 2024) and TempCompass (Liu et al., 2024a) evaluate temporal understanding, while EgoSchema (Mangalam et al., 2023) targets long-form video QA. Broadening the scope, Video-MME (Fu et al., 2025) covers diverse video durations and tasks, and Video-MMMU (Hu et al., 2026) evaluates knowledge acquisition from multidisciplinary videos. For longer contexts, LongVideoBench (Wu et al., 2024) and LVBench (Wang et al., 2025) focus on long-video understanding, while EgoLife (Yang et al., 2025d) examines egocentric experiences spanning hours and days. Beyond temporal understanding, VSI-Bench (Yang et al., 2025c) evaluates video-based spatial reasoning. Despite this growing coverage and temporal range, these benchmarks evaluate questions independently, without requiring information established in the earlier dialogue.

Memory and Multi-Turn Evaluation. Existing benchmarks evaluate conversational memory and multi-turn interaction from several perspectives. LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025) assess long-term conversational memory, while MemBench (Tan et al.,

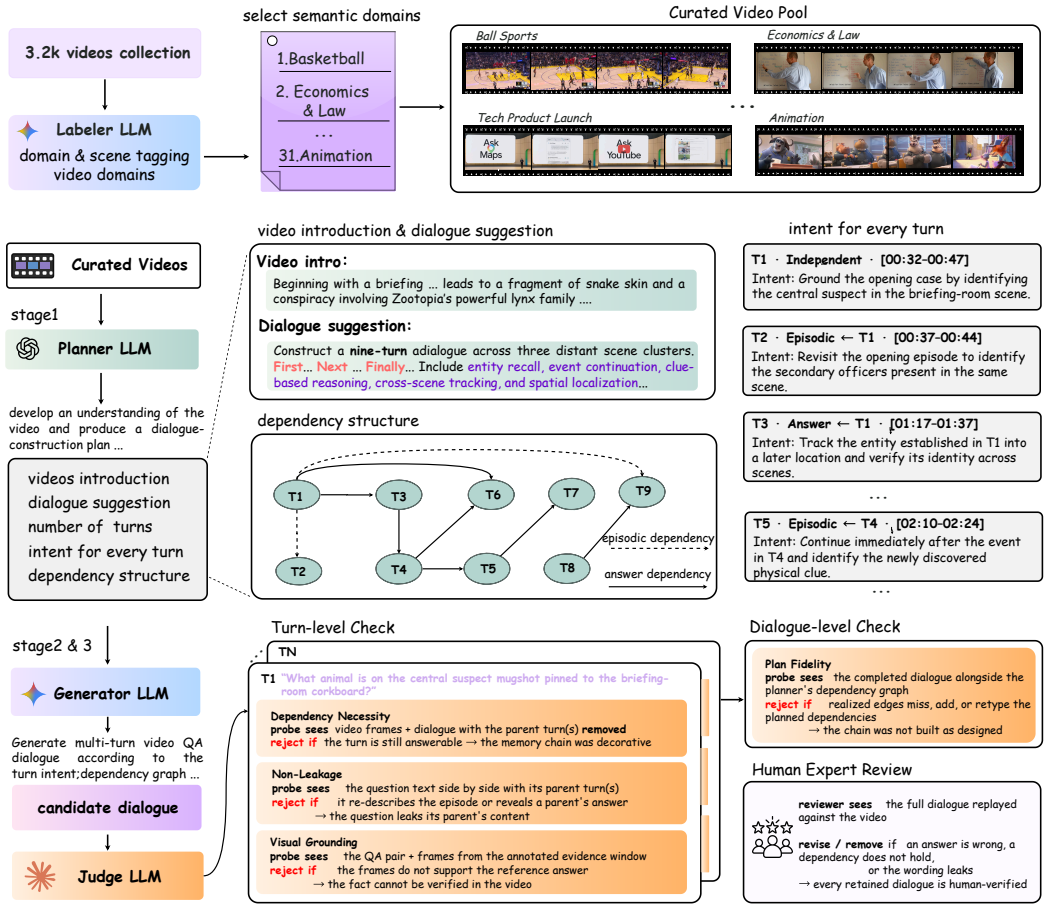


Figure 2: **The VideoMemBench construction pipeline.** Starting from a curated video pool, a planner specifies turn intents and a typed dependency graph. A generator constructs the dialogue turn by turn, while a judger checks necessity, non-leakage, and visual grounding, returning failed turns for regeneration. Completed dialogues are checked against the plan and reviewed by human experts.

2025) distinguishes factual and reflective memory. MT-Bench (Zheng et al., 2023) evaluates two-turn response quality. In multimodal settings, MMDU (Liu et al., 2024b) evaluates multi-turn, multi-image conversations, and MMRC (Xue et al., 2025) includes multi-turn image reasoning and memory recall, but neither targets dialogue memory grounded in a single video. VideoMemBench addresses this gap through explicit episodic and answer dependencies that link later questions to previously discussed video scenes and generated answers, with support for multiple parents and long-range connections.

3 VideoMemBench

Each VideoMemBench sample pairs a video with a multi-turn dialogue, annotated with reference answers and a typed dependency graph. Each edge links a dependent turn to a parent turn and is labeled as Episodic or Answer. Below, we describe the requirements for valid dependencies and visually grounded answers, followed by the construction pipeline and dataset statistics.

3.1 Design Requirements

Constructing valid dependencies requires going beyond simple references to earlier turns. The key challenge lies in ensuring that cross-turn dependencies are genuinely necessary, do not leak prior information, and remain grounded in the video. This leads to three requirements:

Req-1: Dependency Necessity. A dependent question must become unanswerable when its parent turns are removed, even with access to the video and the remaining dialogue. Considering a cooking

video, an implicit reference alone does not ensure this: asking how much of "`that ingredient`" is used requires no dialogue history if the video contains only one ingredient. Such a question avoids revealing the ingredient's identity but fails the necessity requirement.

Req-2: Dependency Non-leakage. A dependent question must not restate the episode descriptions or answers from its parents, even partially. For example, if an earlier answer identifies "`the flour in the red bowl`", asking "`How much of the flour identified earlier is used?`" partially reveals that answer. If both a red and a blue bowl contain flour, the parent remains necessary to identify the target bowl. Thus, a turn may satisfy necessity while violating non-leakage.

Req-3: Visual grounding. Every turn, whether dependent or not, must be anchored in a verifiable visual fact in the video. For dependent turns, this evidence must support the answer under the interpretation established by the parent turns.

3.2 Construction Pipeline

Guided by these requirements, we construct VideoMemBench through a pipeline combining three LLM roles with human expert review, as shown in Figure 2. A *planner* lays out the dependency structure of a dialogue before any question is written, providing the structural precondition for necessity (Req-1). A *generator* realizes the plan turn by turn, referring to earlier turns only through referring expressions (Req-2) and anchoring every answer in frame evidence (Req-3). A *judger* then verifies each generated turn against the three requirements and checks completed dialogues for consistency with the initial structural plan. Human experts finally review each retained dialogue.

• **Video Collection.** We collect videos from multiple sources: Video-MME (Fu et al., 2025), VSI-Bench (Yang et al., 2025c), Video-MMMU (Hu et al., 2026), LongVideoBench (Wu et al., 2024) and the web. Web videos are sourced from YouTube across sports, talks and lectures, tutorials, film recaps, and cooking. We also include EgoLife (Yang et al., 2025d) recordings, allowing recurring people and activities to support dependencies across distant moments. This yields an initial pool of about 3.2k videos. We then use a pre-trained LLM to annotate video domains and scene composition, guiding the selection of videos across different semantic domains with recurring entities and events suitable for cross-turn dependencies. Further collection details are provided in Appendix A.

• **Dependency Planning.** To make dependencies structural rather than incidental (Req-1), the dialogue is designed before any question is generated. The planner samples keyframes across the video and produces a global dialogue plan with four components: (i) a video synopsis summarizing characters, events, and settings; (ii) dialogue guidelines specifying target content, capabilities to assess, and the number of turns; (iii) a dependency graph assigning parent turns and edge types to each dependent turn; and (iv) per-turn intents specifying the information to establish, the relevant video segment, and the information required from parent turns. The plan guides subsequent generation under two constraints: dependencies must arise from recurring video content, and the planned turns must collectively cover the full video. More planning details are provided in Appendix B.

• **Turn-wise Generation.** Following the plan, the generator constructs the dialogue turn by turn, densely sampling frames from each turn's target segment. For dependent turns, the segments are associated with their parents. It first identifies a verifiable visual fact as the reference answer, then formulates a question according to the planned intent and dependency types. Dependent questions use referring expressions without restating the information supplied by their parents. The supporting temporal interval is recorded as the answer's evidence window.

• **Turn-level Check.** For dependent turns, the judger checks necessity by removing parent turns and rejecting questions that remain answerable from the video frames and remaining dialogue. It checks non-leakage by comparing each question with its parents for restated episode descriptions or disclosed answers. For every turn, it verifies that frames from the annotated evidence window support the reference answer and rejects questions that semantically duplicate earlier ones. Rejected turns are regenerated using the judger's feedback; two consecutive failures trigger replanning of the affected portion of the dialogue. Our prompt templates for all stages are given in Appendix C.

• **Expert Review.** Eight human experts review all dialogues that pass the automated checks against the corresponding videos. Turns with incorrect answers, invalid dependencies, or information leakage are revised or removed, ensuring that every retained dialogue is human-reviewed. Details of the dataset review and the separate human baseline evaluation are provided in Appendix D.

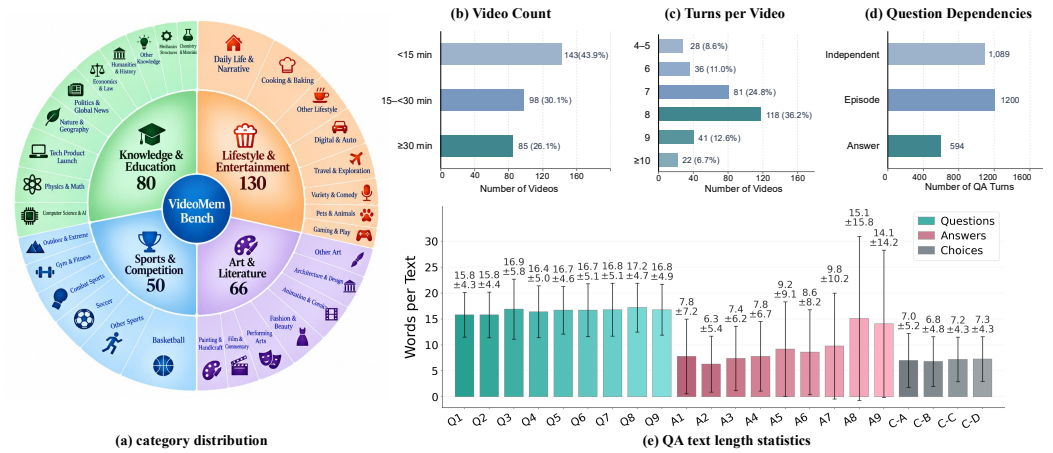


Figure 3: **Dataset statistics of VideoMemBench.** (a) Category distribution across four major domains and their fine-grained subcategories. (b) Distribution of videos by duration. (c) Distribution of the number of dialogue turns per video. (d) Distribution of question dependencies, including independent questions, episodic dependencies, and answer dependencies. (e) Text-length statistics for questions, answers, and answer choices, reported as mean ± standard deviation.

3.3 Dataset Statistics

Figure 3 summarizes the overall statistics of VideoMemBench. VideoMemBench contains 326 dialogues grounded in 326 videos, totaling 2,476 QA turns. Of these, 1,089 (43.98%) are independent, while 1,387 (56.02%) are dependent, with 1,794 typed dependencies in total: 1,200 Episodic and 594 Answer dependencies. Among dependent turns, 348 have multiple parents, including 289 double-parent and 59 triple-parent turns. Dialogues span 4-21 turns, averaging 7.6 turns. Among dependent turns, dependency distance ranges from one to nine turns (*e.g.*, $T_1 \rightarrow T_5$ has distance 4), with up to three parents (*e.g.*, $T_1 + T_4 \rightarrow T_6$ has 2 parents) and a maximum chain depth of seven (*e.g.*, $T_1 \rightarrow T_3 \rightarrow T_4 \rightarrow T_7$ has depth 3). VideoMemBench covers 4 major categories, comprising 31 specific domains, and totaling approximately 154 hours, with an average duration of 28.4 minutes and a maximum of 23.47 hours. Each answer is paired with its corresponding evidence interval. Overall, 63% of questions are multiple-choice and 37% are open-form.

4 Experiments

4.1 Experimental Setup

Models. We evaluate 25 MLLMs, including 8 closed-source models: Claude Opus 5 (Anthropic, 2026a), Claude Sonnet 5 (Anthropic, 2026b), GPT-5.6-sol (OpenAI, 2026d), GPT-5.5 (OpenAI, 2026c), GPT-5.4 (OpenAI, 2026a), Gemini 3.8 Flash (Google DeepMind, 2026b), Gemini 3.5 Flash (Google DeepMind, 2026a), and Gemini 3.1 Pro Preview (Google, 2026), as well as 17 open-weight models: Kimi K3 (Team et al., 2026), Qwen3.5 (0.8B, 4B, 9B, 27B, 35B-A3B, 122B-A10B, 397B-A17B) (Team, 2026), Qwen3-VL (4B, 8B, 30B-A3B, 32B, 235B-A22B) (Bai et al., 2025a), Qwen2.5-VL (3B, 7B, 32B) (Bai et al., 2025b), and Keye-VL-1.5-8B (Yang et al., 2025b).

Evaluation protocol. All models receive the same 128 frames sampled across the full video, with timestamps indicating their temporal positions. The sampled frame sequence is fixed across models. The video frames are provided at the beginning of the conversation and remain accessible throughout the dialogue. Questions are presented strictly in dialogue order, with the model's own previous responses retained in the conversation history. Specifically, the input at turn t is

$$V \,\|\, (q_1, \hat{a}_1) \,\|\, \cdots \,\|\, (q_{t-1}, \hat{a}_{t-1}) \,\|\, q_t, \tag{1}$$

where V denotes the timestamped frame sequence, q_i the question and its answer options where applicable, $\hat{a}_i$ is the model response at turn i, and $\|$ denotes concatenation.

Table 2: **Main results on VideoMemBench.** Models are evaluated under the self-rollout protocol. TurnAcc: accuracy over all turns, scored individually. EpiDepAcc: accuracy on turns with at least one Episodic dependency. AnsDepAcc: accuracy on turns with at least one Answer dependency under chain crediting. Overall: the arithmetic mean of the three metrics. The best and second-best results in each metric within each model group are **bold** and underlined, respectively.

Model	Frames	TurnAcc (%)	EpiDepAcc (%)	AnsDepAcc (%)	Overall (%)
Human	–	88.1	86.0	84.2	86.1
Closed-source models					
Claude Opus 5	128	**52.3**	**49.1**	**35.7**	**45.7**
Gemini 3.8 Flash	128	50.4	46.6	30.3	42.4
GPT-5.6-sol	128	50.0	47.8	27.1	41.7
Gemini 3.5 Flash	128	49.6	45.5	29.7	41.6
GPT-5.5	128	48.0	46.5	27.0	40.5
Claude Sonnet 5	128	47.8	45.3	27.3	40.2
Gemini 3.1 Pro Preview	128	45.7	44.8	27.0	39.1
GPT-5.4	128	44.4	42.6	21.9	36.3
Open-weight models					
Kimi K3	128	**50.5**	**49.0**	**27.1**	**42.2**
Qwen3-VL-235B-A22B	128	43.1	41.6	18.6	34.4
Qwen3.5-27B	128	42.2	39.8	20.9	34.3
Qwen3.5-397B-A17B	128	42.3	40.1	20.3	34.2
Qwen3.5-122B-A10B	128	42.4	39.5	19.9	33.9
Qwen3.5-35B-A3B	128	38.2	36.3	18.0	30.8
Qwen3-VL-32B	128	39.3	35.9	17.2	30.8
Qwen3-VL-30B-A3B	128	36.4	35.1	15.8	29.1
Qwen3.5-9B	128	34.6	33.4	12.9	27.0
Qwen3-VL-8B	128	34.8	32.1	12.3	26.4
Qwen3.5-4B	128	33.8	32.6	12.5	26.3
Qwen3-VL-4B	128	32.4	29.4	12.9	24.9
Qwen2.5-VL-32B	128	31.7	32.6	9.6	24.6
Qwen2.5-VL-7B	128	26.2	24.9	8.4	19.8
Keye-VL-1.5-8B	128	25.4	24.5	8.2	19.4
Qwen2.5-VL-3B	128	23.5	24.4	5.7	17.9
Qwen3.5-0.8B	128	21.6	21.3	5.7	16.2

4.2 Evaluation Metrics

We use GPT-5.4 mini (OpenAI, 2026b) as the judge for open-ended responses and extract the predicted option for multiple-choice questions. Each response is graded as correct or incorrect. We report three metrics with distinct evaluation subsets and scoring rules, plus an aggregate score.

Turn Accuracy (TurnAcc). TurnAcc measures accuracy over all turns. Each response is scored independently of whether the model answered its parent turns correctly.

Episodic Dependency Accuracy (EpiDepAcc). EpiDepAcc measures accuracy over turns with at least one Episodic dependency. Each response is scored independently of its parents' answer correctness, since an Episodic dependency refers to the visual context described by a parent question rather than the parent's answer. This metric measures recall of *what the model has seen*.

Answer Dependency Accuracy (AnsDepAcc). AnsDepAcc evaluates turns with at least one Answer dependency using *chain crediting*: a turn receives credit only if its response and all responses along its Answer-dependency ancestry are correct. Formally,

$$\mathrm{AnsDepAcc} = \frac{1}{|\mathcal{T}_{\mathrm{ans}}|} \sum_{t \in \mathcal{T}_{\mathrm{ans}}} \mathrm{corr}(t) \prod_{a \in \mathrm{Anc}_{\mathrm{ans}}(t)} \mathrm{corr}(a), \tag{2}$$

where $\mathcal{T}_{\mathrm{ans}}$ contains turns with at least one Answer dependency, $\mathrm{corr}(t) \in \{0, 1\}$ indicates whether the response at turn t is correct, and $\mathrm{Anc}_{\mathrm{ans}}(t)$ contains all ancestors reachable by recursively following only Answer edges toward earlier turns. Episodic edges are excluded from this traversal. By accounting for both current-turn errors and upstream failures across answer-dependent chains, this metric evaluates recall of *what the model has said*.

Multi-dependent turns with both dependency types contribute to both EpiDepAcc and AnsDepAcc under their respective scoring rules.

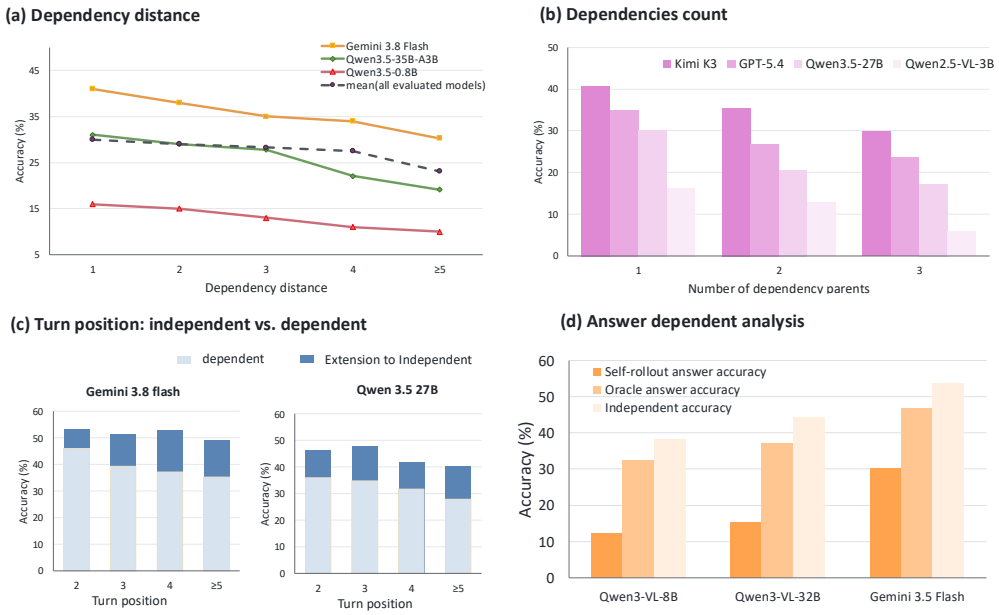


Figure 4: **Analysis of factors affecting in-context video memory.** (a) Accuracy on dependent turns by dependency distance, measured from the current turn to its earliest parent; the dashed line shows the mean across all evaluated models. (b) Accuracy on dependent turns by the number of parents. (c) Independent- and dependent-turn accuracy by dialogue position for two models: the light segment shows dependent-turn accuracy, the full bar height shows independent-turn accuracy, and the dark segment represents their difference. (d) Accuracy on answer-dependent turns under self-rollout and oracle histories. Oracle histories replace previous model responses with ground-truth answers.

Overall. Overall is the arithmetic mean of TurnAcc, EpiDepAcc, and AnsDepAcc, which is used to simply rank models. It aggregates complementary, overlapping evaluation subsets rather than representing accuracy on a separate set of questions.

4.3 Benchmark Results

Table 2 reports results for 25 models, including 8 closed-source and 17 open-weight MLLMs. Claude Opus 5 achieves the highest overall score of 45.7%, compared with 86.1% for human annotators. This gap suggests substantial room for MLLMs to improve in-context video memory.

Closed-source Models. Claude Opus 5 leads the closed-source models across all three metrics, achieving 52.3% on TurnAcc, 49.1% on EpiDepAcc, and 35.7% on AnsDepAcc. Gemini 3.8 Flash and GPT-5.6-sol follow with overall scores of 42.4% and 41.7%, respectively, while GPT-5.4 scores lowest at 36.3%. Across these closed-source models, AnsDepAcc is 16.6–22.9 percentage points below TurnAcc, compared with only 3.9 percentage points for human annotators. Since AnsDepAcc requires correctness throughout the Answer-dependency ancestry, this gap reflects both current-turn errors and accumulated upstream failures. These results highlight the difficulty of maintaining correct answers across dependent turns in extended video conversations, even for frontier models.

Open-weight Models. Kimi K3 leads the open-weight models with an overall score of 42.2%, ranking third overall and second on both TurnAcc and EpiDepAcc across all evaluated models. It achieves 50.5% on TurnAcc, 49.0% on EpiDepAcc, and 27.1% on AnsDepAcc. All other open-weight models score below GPT-5.4, the lowest-performing closed-source model (36.3% overall). The runner-up, Qwen3-VL-235B-A22B, scores 34.4% overall, 7.8 percentage points below Kimi K3 and narrowly ahead of Qwen3.5-27B (34.3%). Within the evaluated dense Qwen3.5 models, the overall score increases with model size, from 16.2% at 0.8B to 34.3% at 27B. At comparable scales, Qwen3-VL-32B outperforms Qwen2.5-VL-32B by 6.2 percentage points. These results show substantial variation among open-weight models: Kimi K3 is competitive with frontier closed-source models in our in-context video memory setting, while the other evaluated models remain behind.

Table 3: **Independent versus dependent turns.** Indep.: accuracy over the independent turns. Dep.: mean of EpiDepAcc and AnsDepAcc. $\Delta = \mathrm{Indep.} - \mathrm{Dep.}$, measured in percentage points (pp).

Model	Indep. (%)	EpiDepAcc (%)	AnsDepAcc (%)	Dep. (%)	Δ (pp)
Kimi K3	51.4	49.0	27.1	38.1	13.4
GPT-5.6-sol	50.6	47.8	27.1	37.5	13.1
Claude Sonnet 5	50.0	45.3	27.3	36.3	13.7
Qwen3.5-27B	46.0	39.8	20.9	30.3	15.7
Qwen3-VL-235B-A22B	47.4	41.6	18.6	30.1	17.3

4.4 Analysis

Dependency Distance. Figure 4(a) groups dependent turns by the distance to their earliest parent. Mean accuracy across all evaluated models decreases from 30.2% at distance 1 to 23.1% at distances of at least five turns, with a larger decline in the longest-distance group. The model-specific trends differ: Gemini 3.8 Flash declines relatively steadily from 41.1% to 30.2%, whereas Qwen3.5-35B-A3B drops more sharply beyond distance 3, falling from 27.8% to 19.2%. These trends suggest that longer dependencies pose greater challenges for in-context video memory, although models differ in their sensitivity to dependency distance.

Number of Parents. Figure 4(b) shows a consistent decline in accuracy as the number of parent turns increases across all four models. Kimi K3 remains the strongest at every parent count, but its accuracy falls from 40.8% with one parent to 29.8% with three. GPT-5.4 and Qwen3.5-27B show the same trend, falling from 34.9% to 23.7% and from 30.1% to 17.3%, respectively. These results suggest that integrating information from multiple earlier turns poses an additional challenge for in-context video memory, even for the strongest model in this comparison.

Turn Position and Dependency. Table 3 shows that the mean of EpiDepAcc and AnsDepAcc is 13.1–17.3 percentage points below independent-turn accuracy across the five reported models, although this gap also reflects the stricter chain-crediting rule used by AnsDepAcc. Figure 4(c) further compares independent and dependent turns at different dialogue positions. For Gemini 3.8 Flash, dependent-turn accuracy decreases from 46.1% at turn 2 to 35.2% at turns 5 and beyond, while independent-turn accuracy declines more modestly, from 53.8% to 49.2%. Qwen3.5-27B exhibits a similar pattern: dependent-turn accuracy falls from 36.2% to 28.1%, compared with a decline from 46.1% to 40.3% on independent turns. Although independent-turn accuracy fluctuates at intermediate positions, dependent-turn accuracy declines consistently for both models. These patterns suggest that later dialogue positions are associated with greater difficulty, particularly when answering requires information from earlier turns.

Effect of Answer History. Figure 4(d) compares accuracy on answer-dependent turns under self-rollout and oracle histories, with independent-turn accuracy as a reference. Oracle histories replace previous model responses with reference answers. All three models benefit substantially: Qwen3-VL-8B improves from 12.3% to 32.4%, Qwen3-VL-32B from 15.4% to 37.3%, and Gemini 3.5 Flash from 30.1% to 47.2%. These gains indicate that errors in earlier responses are an important source of difficulty for subsequent answer-dependent questions. Nevertheless, oracle accuracy remains below independent-turn accuracy, suggesting that correct history alone is insufficient for reliable in-context video memory.

We further provide frame-count ablations with 16, 32, and 64 sampled frames in Appendix E.1 and additional dialogue examples in Appendix E.2.

5 Conclusion

We introduce VideoMemBench to evaluate video memory through explicit Episodic and Answer dependencies, distinguishing what a model has seen from what it has said, with automated verifications and human reviews. Evaluation of over 20 MLLMs shows that the best model achieves only 45.7% overall, with lower accuracy on longer-range and multi-parent dependencies. These findings highlight the difficulty of maintaining and using dialogue context in video conversations. We hope VideoMemBench supports the development of video MLLMs that reliably recall and build on earlier observations and responses.

AI USE STATEMENT

In this work, we used generative AI tools to assist with dataset construction, including video domain and scene classification, dialogue and dependency planning, question–answer generation, and automated quality checks, as described in subsection 3.2 and Appendix C. In VideoMemBench, all retained dialogues underwent human expert review for answer correctness, visual evidence, dependency validity, necessity, and non-leakage. AI suggestions were treated as assistance rather than independent evidence for scientific claims. We take responsibility for the final content of this work, including all text, claims, data, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the dataset construction pipeline in section 3 and the evaluation protocol and metrics in section 4. Additional construction details, prompt templates, and human review procedures are provided in Appendix A, Appendix C and Appendix D. To support reproducibility, we provide an anonymous repository https://anonymous.4open.science/r/VideoMemBench-4C21 containing additional benchmark examples, scripts for both dataset construction and benchmark evaluation.

REFERENCES

Anthropic. Introducing Claude Opus 5, 2026a. URL https://www.anthropic.com/news/claude-opus-5.

Anthropic. Introducing Claude Sonnet 5. https://www.anthropic.com/news/claude-sonnet-5, June 2026b.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

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

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, February 2026.

Google DeepMind. Gemini 3.5: Frontier intelligence with action. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-5/, May 2026a.

Google DeepMind. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber. https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/, September 2026b.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Xiang Yue, Bo Li, Yuanhan Zhang, and Ziwei Liu. Video-MMMU: Evaluating knowledge acquisition from multidisciplinary professional videos. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27798–27828, 2026.

Alexis Huet, Zied Houidi, and Dario Rossi. Episodic memories generation and evaluation benchmark for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 51409–51457, 2025.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MvBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video llms really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024a.

Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, et al. MMDU: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for lvlms. *Advances in Neural Information Processing Systems*, 37: 8698–8733, 2024b.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

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

OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, March 2026a.

OpenAI. Introducing GPT-5.4 mini and nano. `https://openai.com/index/introducing-gpt-5-4-mini-and-nano/`, March 2026b.

OpenAI. Introducing GPT-5.5. `https://openai.com/index/introducing-gpt-5-5/`, April 2026c.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. `https://openai.com/index/gpt-5-6/`, July 2026d.

Yaning Pan, Qianqian Xie, Guohui Zhang, Zekun Moore Wang, Yongqian Wen, Yuanxing Zhang, Haoxuan Hu, Zhiyu Pan, Yibing Huang, Zhidong Gan, et al. MT-Video-Bench: A holistic video understanding benchmark for evaluating multimodal llms in multi-turn dialogues. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8105–8126, 2026.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards more comprehensive evaluation on the memory of llm-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19336–19352, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

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

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. LVBench: An extreme long video understanding benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, volume 2025, pp. 86809–86836, 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Haochen Xue, Feilong Tang, Ming Hu, Yexin Liu, Qidong Huang, Yulong Li, Chengzhi Liu, Zhongxing Xu, Chong Zhang, Chun-Mei Feng, et al. MMRC: A large-scale benchmark for understanding multimodal large language model in real-world conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22477–22503, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai Keye-VL 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025b.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025c.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, et al. EgoLife: Towards egocentric life assistant. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28885–28900. IEEE, 2025d.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A Video Collection Details

Video sources. We assemble an initial pool of approximately 3.2k videos from five existing video-understanding benchmarks and the web. For web collection, we search YouTube using queries spanning sports ("full match" and "match replay"), talks and lectures ("keynote" and "course lecture"), tutorials ("full lesson"), film recaps ("movie recap" and "film summary"), and cooking ("cooking lesson" and "chef pov").

Domain taxonomy. The videos are organized into four broad categories, each comprising finer-grained domains:

- **Lifestyle & Entertainment (130 videos).** This category covers daily activities and entertainment, including Daily Life & Narrative, Cooking & Baking, Digital & Auto, Travel & Exploration, Variety & Comedy, Pets & Animals, Gaming & Play, and Other Lifestyle.
- **Knowledge & Education (80 videos).** This category spans scientific, technical, and social topics: Computer Science & AI, Physics & Math, Tech Product Launch, Nature & Geography, Politics & Global News, Economics & Law, Humanities & History, Mechanics Structures, and Chemistry & Materials, together with Other Knowledge.
- **Art & Literature (66 videos).** This category includes Painting & Handicraft, Film & Commentary, Performing Arts, Fashion & Beauty, Animation & Comics, Architecture & Design, and Other Art.
- **Sports & Competition (50 videos).** This category covers Basketball, Soccer, Combat Sports, Gym & Fitness, Outdoor & Extreme, and Other Sports.

B Dependency Planning Details

Before question generation, the planner examines keyframes sampled across the video and produces a complete dialogue specification. The plan contains four components:

Video synopsis. The synopsis summarizes the main characters, events, and settings, providing the generator with global context for interpreting individual video segments.

Dialogue guidelines. The planner identifies which parts of the video to cover, which capabilities to assess, and how many turns to construct. Target capabilities include entity recall, event continuation, and cross-scene character tracking.

Dependency graph. The planner constructs a directed acyclic graph whose nodes represent dialogue turns. Each dependent turn is assigned one or more earlier parent turns, with each incoming edge labeled as Episodic or Answer. Specifying the graph before generation allows dependencies to be coordinated across the dialogue.

Per-turn intents. For each turn, the planner specifies what information it should establish and which video segment supports it. For dependent turns, the intent additionally identifies the information required from each parent, giving the generator an explicit basis for constructing the question.

The plan must satisfy two constraints. First, dependencies must be motivated by recurring entities, events, or details in the video, rather than imposed between unrelated topics. Second, the selected segments must span the video rather than concentrate on a local clip. The complete plan is passed to the generator as the specification for turn-wise generation.

C Prompt Templates

We report the prompt templates used to construct and evaluate VideoMemBench, following the pipeline in Figure 2. During initial video curation, a pre-trained LLM first assigns each collected video a semantic domain and scene tags. In Stage 1, a Planner LLM watches each curated video and produces a dialogue-construction plan: a video introduction, a dialogue suggestion, the number of

turns, a per-turn intent with its evidence window, and a typed dependency graph. In Stages 2 and 3, a Generator LLM writes the dialogue turn by turn according to the plan, and a Judge LLM applies three turn-level checks (dependency necessity, non-leakage, and visual grounding); turns that fail are returned to the Generator LLM with the judge's feedback for regeneration. Each completed dialogue is then checked against the planner's dependency graph (plan fidelity) and reviewed by human experts. We preserve runtime placeholders such as `{video_frames}`, `{plan}`, and `{parent_turns}`. In all prompts, an *answer* dependency means the child turn needs the parent turn's correct answer, and an *episodic* dependency means the child turn only needs to relocate the same scene or entity as the parent turn.

Video curation.

Labeler LLM (Gemini 3 Flash): Domain and Scene Tagging

```
You are tagging videos for a multi-turn video memory benchmark.
You will see frames sampled uniformly across one video, each preceded by a <MM:SS>
    time tag.

Assign exactly ONE semantic domain from this list:
{domain_taxonomy}

Then describe the scene structure of the video.

Rules:
- Choose the domain from what is shown on screen, not from the video title or file
    name.
- A scene is a continuous stretch with the same location and the same main
    participants.
  List every distinct scene with its time window.
- Mark a video as unsuitable if it has fewer than 3 visually distinct scenes, if
    most frames are
  static text or slides, or if the content cannot be verified from the frames
    alone.
- Do not guess what happens between sampled frames.

Output ONLY JSON:
{
  "domain": string,                    // one entry from the domain list
  "domain_confidence": number,         // 0..1
  "scenes": [
    {"scene_id": "S1", "t_start": number, "t_end": number,
     "location": string, "main_entities": [string, ...], "summary": string}
  ],
  "num_distinct_scenes": int,
  "suitable": true|false,
  "unsuitable_reason": string|null
}
```

Stage 1: planning.

Planner LLM (GPT 5.5): Dialogue-Construction Plan

```
You are planning a multi-turn question-answering dialogue about one video for a
    benchmark
that tests in-context video memory: later questions should only be answerable by
    remembering
what earlier turns established.
You will see frames sampled across the whole video, each preceded by a <MM:SS>
    time tag, and
the video's domain and scene list from the labeler.

Do NOT write questions or answers. Produce a PLAN that another model will follow.

Step 1: Video introduction
```

```
Write 2—4 sentences describing what happens in the video, in order, naming the
    main entities
and scene changes. Only describe what is visible in the frames.

Step 2: Dialogue suggestion
Decide how many turns the dialogue should have (between {min_turns} and
    {max_turns}) and how
it should move through the video. Pick at least three scene clusters that are far
    apart in time
and describe the route: what is established first, what is revisited next, what is
    tracked last.
List the memory abilities the dialogue will exercise, chosen from: entity recall,
    event
continuation, clue—based reasoning, cross—scene tracking, spatial localization.

Step 3: Intent for every turn
For each turn, give:
— turn_id (T1, T2, ...)
— question_type: multi_choice_question, true_or_false_question, or open_structured
— dependency: independent, or a list of parent turns, each typed as "answer" or
    "episodic"
— evidence window [t_start, t_end] in seconds where the answer is visible
— intent: one sentence saying what the turn should establish or test

Step 4: Dependency structure
Design the dependency graph with these rules:
— "answer": the child cannot be answered without the parent's CORRECT answer.
  Test: if the parent were answered wrongly, would the child's correct answer
    change? Only then use "answer".
  The parent of an "answer" edge may use any supported question type, provided
    that its answer supplies information required by the child.
— "episodic": the child only needs to return to the same scene, moment, or entity
    as the parent.
  The parent's answer being wrong does not change the child's answer.
— Parents must come earlier than the child. The graph must have no cycles.
— More than half of the turns must have at least one parent.
— At least one turn must have two or three parents.
— At least one edge must span two or more turns (not only adjacent turns).
— T1 must be independent and may use any supported question type.
— Evidence windows must cover the beginning, middle, and end of the video.
— Every edge must be genuine: the child's target must be hard to describe without
    the parent.
  Do not connect turns that are about unrelated things.

Input:
domain: {domain}
scenes: {scenes}
frames: {video_frames}

Output ONLY JSON:
{
  "video_intro": string,
  "dialogue_suggestion": {
    "num_turns": int,
    "scene_route": string,
    "abilities": [string, ...]
  },
  "turns": [
    {"turn_id": "T1", "question_type": string,
     "dependency": "independent" | [{"parent": "T#", "type": "answer"|"episodic"},
    ...],
     "evidence": {"t_start": number, "t_end": number},
     "intent": string}
  ],
  "dependency_graph": [{"from": "T#", "to": "T#", "type": "answer"|"episodic"},
    ...]
}
```

Stage 2: generation.

Generate LLM (Gemini 3.5 Flash): Turn Generation

```
You are writing one turn of a multi—turn video QA dialogue, following a plan.
You will see frames from the video with <MM:SS> time tags, the plan, and the turns
    already written.

Write turn {turn_id} so that it matches its planned intent, question type,
    dependency, and
evidence window exactly.

Build the answer first
1. Look at the frames inside the planned evidence window and pick ONE specific
    fact you can
   clearly see (an identity, position, count, color, action, text, or change).
2. That fact is the answer. If you cannot see a clear fact that serves the intent,
    say so in
   "plan_issue" instead of guessing.
3. Write a short question (one sentence, at most two) that asks for that fact.
4. For multiple choice, write 4 options (A—D) of the same kind, each under 15
    words.
   Every wrong option should be plausible for this video and wrong for exactly one
    reason.
5. Write answer_reasoning: 1—2 sentences naming what in the frames makes the
    answer correct.

Dependencies
— Refer to each parent turn inside the sentence by its turn id, for example
  "the player identified in T1" or "the shelf asked about in T3".
— For an "answer" parent, the question must be impossible to answer without
    knowing which
  option was correct in the parent. Refer to that result abstractly; never state
    it.
— For an "episodic" parent, the question returns to the same scene or entity. Give
    just enough
  visual anchors (appearance, location, nearby landmark) to find it again, but do
    not describe
  the parent's scene so fully that the parent becomes unnecessary.
— Never repeat a parent's correct option, and never use words that make it obvious.
  The answering model will see earlier questions but not earlier correct answers.
— Do not add dependencies that are not in the plan, and do not drop planned ones.

Wording
— No exact timestamps in the question; use scene descriptions ("when the
    scoreboard first appears").
— One question per turn. Natural language, no placeholders, no labels like "object
    A".
— true_or_false_question: options are exactly A = "Yes", B = "No".
— open_structured: no options; give a short reference answer and 1—3 acceptable
    paraphrases.

Input:
plan: {plan}
turns written so far: {previous_turns}
turn to write: {turn_plan}
frames: {video_frames}

Output ONLY JSON:
{
  "turn_id": "T#",
  "type": "multi_choice_question"|"true_or_false_question"|"open_structured",
  "dependency": {"depends_on": ["T#", ...], "dep_type": {"T#":
    "answer"|"episodic"}} | null,
  "question": string,
  "options": {"A": string, "B": string, "C": string, "D": string} | null,
  "answer": string,
  "acceptable_answers": [string, ...] | null,
  "answer_reasoning": string,
  "evidence": [{"t_start": number, "t_end": number}],
  "plan_issue": string|null
}
```

Generator LLM (Gemini 3.5 Flash): Regeneration of a Failed Turn

```
A turn you wrote was rejected by a quality check. Rewrite it so it passes, keeping
    everything
the plan requires.

You will see the plan entry for this turn, the turns written so far, the rejected
    turn, the
name of the failed check, and the judge's reason.

How to fix each failed check
— dependency_necessity: the turn could be answered without its parent(s). Remove
    the details
  that let the target be found on its own, and refer to the parent turn instead,
    so that the
  parent is genuinely required.
— non_leakage: the question re—described the parent's episode or revealed the
    parent's answer.
  Replace those words with a neutral reference to the parent turn by its turn id.
— visual_grounding: the frames in the evidence window do not support the answer.
    Pick a fact
  that IS visible in that window and rebuild the question and answer around it.

Keep fixed: turn_id, question type, planned dependency and its types, evidence
    window.
If the turn cannot be fixed without changing the plan, return the turn unchanged
    and explain
in "plan_issue".

Input:
plan entry: {turn_plan}
turns written so far: {previous_turns}
rejected turn: {turn}
failed check: {check_name}
judge reason: {judge_reason}
frames: {video_frames}

Output ONLY JSON with the same schema as Turn Generation.
```

Stage 3: turn-level checks.

Judge LLM (Claude Opus 4.8): Dependency Necessity Probe

```
You are answering one question about a video. Some earlier turns of the
    conversation have been
removed on purpose. Answer only if the frames and the remaining conversation are
    enough.

You will see frames from the video with <MM:SS> time tags, the conversation so far
    with some
turns removed, and one question.

Rules:
— If the question refers to an earlier turn that is not shown, you do not know
    what that turn
  established. Do not guess what it said.
— If you can still identify the answer with confidence from the frames alone, give
    it.
— If the answer depends on information you do not have, output "ABSTAIN".

Input:
frames: {video_frames}
conversation (parent turns removed): {history_without_parents}
question: {question}
options: {options}

Output ONLY JSON:
```

```
{"answer": string, "confidence": number, "why": string}
// answer: an option letter, a short answer for open questions, or "ABSTAIN"

Decision rule (applied by the pipeline, not shown to the probe): the turn is
    REJECTED if the probe
answers correctly with confidence >= 0.5, because then the memory chain was
    decorative.
```

Judge LLM (Claude Opus 4.8): Non-Leakage Check

```
You are checking whether a later question gives away its parent turns.
You will see the later question next to each of its parent turns (question,
    options, and correct
answer). You do not see the video.

A question LEAKS if either condition holds:
1. Answer leak: it states, paraphrases, or reveals any part of a
parent's correct answer, even if the parent is still needed to
recover the remaining information.
2. Episode re—description: it repeats identifying details of the
scene, moment, or entity established by an episodic parent,
rather than referring to it without restating those details.

Generic references such as "the player identified in T1" or
"the moment discussed in T3" are allowed.

For example, if a parent answer is "the flour in the red bowl",
"How much of the flour identified earlier is used?" leaks the
ingredient, even if the parent is still needed to identify the bowl.
Leakage does not require making the parent unnecessary.

Input:
later question: {question}
dependency types: {dep_type}
parent turns: {parent_turns}

Output ONLY JSON:
{
  "leak": true|false,
  "answer_leak_parents": ["T#", ...],
  "redescribed_parents": ["T#", ...],
  "offending_text": [string, ...],
  "reason": string
}
```

Judge LLM (Claude Opus 4.8): Visual Grounding Check

```
You are checking whether a question and its reference answer are supported by the
    video.
You will see ONLY the frames from the turn's annotated evidence window, each with
    a <MM:SS> tag,
and the question—answer pair.

Steps:
1. Before reading the reference answer, answer the question yourself from these
    frames.
2. Compare your answer with the reference answer.
3. Check that the question itself contains no facts that contradict the frames
    (wrong numbers,
   text, colors, or objects).
4. For multiple choice, check that exactly one option is correct in these frames.

The turn is SUPPORTED only if the frames clearly show the fact the reference
    answer states and
```

```
no other option is also correct. If the key object or action is not visible in
    these frames,
it is NOT supported, even if you believe it happens elsewhere in the video.

Input:
frames (evidence window): {evidence_frames}
question: {question}
options: {options}
reference answer: {answer}

Output ONLY JSON:
{
  "supported": true|false,
  "my_answer": string,
  "support_ts": [string, ...],     // <MM:SS> tags of frames showing the fact
  "question_factual_errors": [string, ...],
  "multiple_correct_options": true|false,
  "reason": string
}
```

Dialogue-level check.

Judge LLM (Claude Opus 4.8): Plan Fidelity Check

```
You are checking whether a finished dialogue was built the way it was planned.
You will see the planner's per—turn intents and dependency graph, and the
    completed dialogue.
You do not see the video.

For every turn, read the question and decide which earlier turns it actually
    needs, and for each
one whether it needs the parent's correct ANSWER or only the same scene or entity
    (EPISODIC).
Base this on the question text, not on the dependency fields the generator wrote.

Then compare this realized graph with the planned graph:
— missing edge: planned but the question does not actually need that parent
— extra edge: the question needs a parent that the plan did not include
— retyped edge: planned as "answer" but only episodic is needed, or the reverse
Also flag any turn whose question does not serve its planned intent.

The dialogue is REJECTED if any edge is missing, extra, or retyped, or if any turn
    does not
serve its intent.

Input:
planned turns: {plan_turns}
planned dependency graph: {dependency_graph}
completed dialogue: {dialogue}

Output ONLY JSON:
{
  "realized_graph": [{"from": "T#", "to": "T#", "type": "answer"|"episodic"}, ...],
  "missing_edges": [...],
  "extra_edges": [...],
  "retyped_edges": [...],
  "intent_mismatches": [{"turn_id": "T#", "reason": string}, ...],
  "pass": true|false
}
```

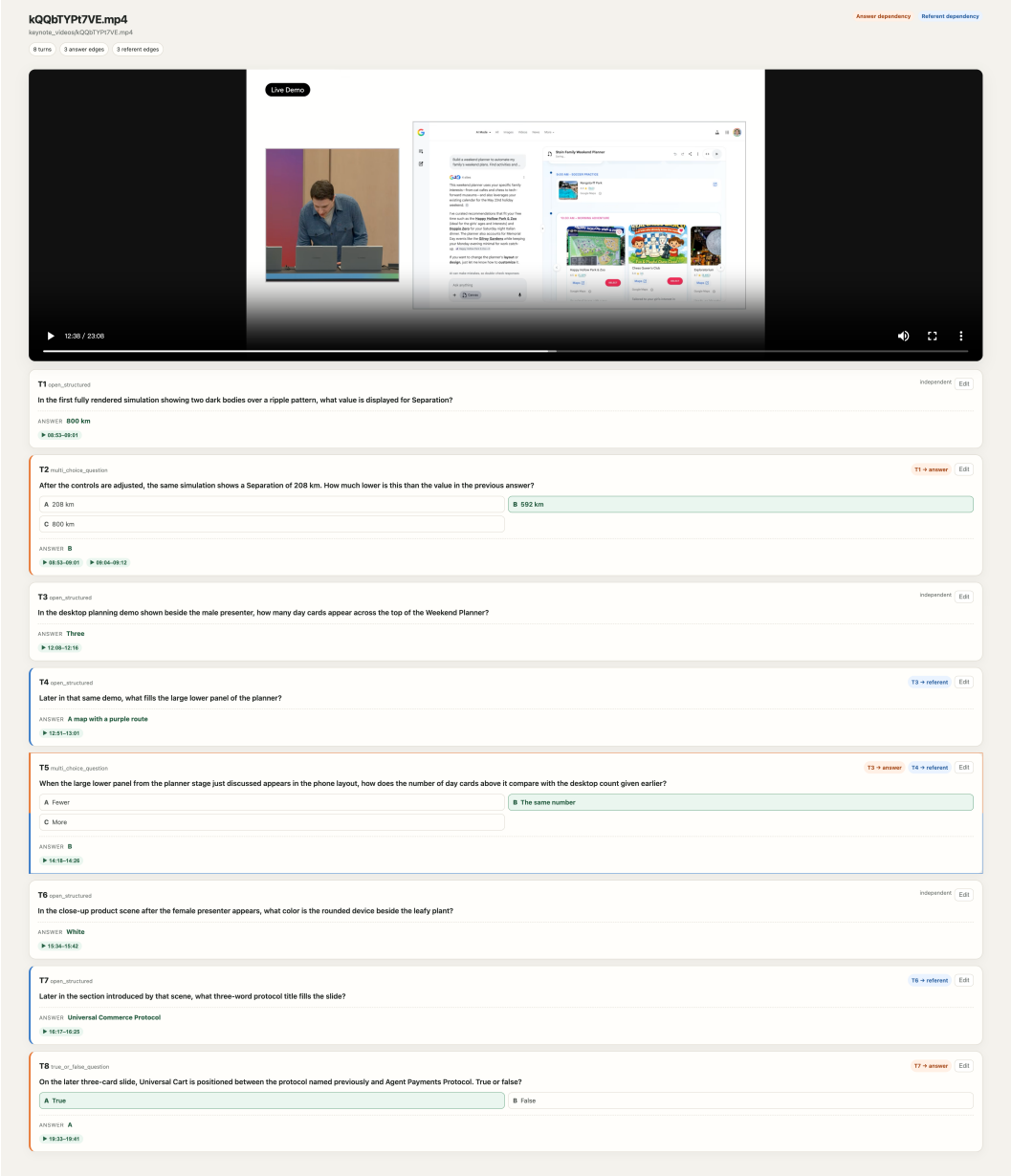


Figure 5: **The expert review interface.** For each dialogue, the expert replays the video in the top panel, where the evidence window of the current turn is marked on the timeline, and reads the dialogue turn by turn, where the parent turns and the type of each dependency are highlighted. The expert answers the current question before the reference answer is revealed, then marks the turn as *correct*, *answer wrong*, *dependency not required*, *wording leaks a parent*, or *duplicate*, and edits the question, the options, or the answer in place. A dialogue is retained only after every turn has been marked correct or revised, and the interface records each decision together with the time spent on the turn. Referent here means episodic dependency. Please zoom in to see the details.

D Human Review Details

D.1 Dataset Review

We conduct a human review of all constructed multi-turn dialogues and dependency annotations before finalizing VideoMemBench. Eight experts participate in this review. As shown in Figure 5, our

Table 4: **Frame-count ablation on VideoMemBench under self-rollout.** Each model is evaluated with 16, 32, and 64 uniformly sampled frames per video, using the same metrics as the 128-frame evaluation in Table 2. Models are ranked by the overall score within each frame budget, with the best result for each metric shown in **bold**.

Model	Frames	TurnAcc (%)	EpiDepAcc (%)	AnsDepAcc (%)	Overall (%)
Qwen3-VL-235B-A22B	16	33.7	32.3	11.7	**25.9**
Qwen3.5-122B-A10B	16	33.2	32.4	**11.9**	25.8
Qwen3.5-27B	16	**33.9**	**33.7**	8.2	25.3
Qwen3-VL-32B	16	30.6	28.9	10.5	23.3
Qwen3-VL-30B-A3B	16	29.6	30.6	7.6	22.6
Qwen3.5-35B-A3B	16	30.7	28.1	9.0	22.6
Qwen3-VL-8B	16	27.5	27.7	6.3	20.5
Qwen3.5-9B	16	28.0	26.5	6.8	20.5
Qwen3-VL-4B	16	26.7	26.5	8.0	20.4
Keye-VL-1.5-8B	16	22.0	22.5	4.7	16.4
Qwen2.5-VL-7B	16	21.4	21.3	3.7	15.5
Qwen2.5-VL-3B	16	20.2	20.1	3.7	14.7
Qwen3.5-0.8B	16	17.8	18.0	3.9	13.2
Qwen3-VL-235B-A22B	32	**36.3**	**37.8**	**14.5**	**29.5**
Qwen3.5-122B-A10B	32	35.0	35.2	10.7	27.0
Qwen3.5-35B-A3B	32	34.5	34.1	12.3	27.0
Qwen3.5-27B	32	34.9	34.2	11.7	26.9
Qwen3-VL-32B	32	34.1	32.6	13.1	26.6
Qwen3-VL-30B-A3B	32	32.1	32.5	8.0	24.2
Qwen3.5-9B	32	29.8	30.4	8.6	22.9
Qwen3-VL-8B	32	29.4	29.4	9.2	22.7
Qwen3-VL-4B	32	29.0	29.0	8.8	22.3
Qwen2.5-VL-7B	32	23.2	23.7	4.9	17.3
Keye-VL-1.5-8B	32	23.3	23.1	5.1	17.2
Qwen2.5-VL-3B	32	22.6	23.4	4.7	16.9
Qwen3.5-0.8B	32	20.1	21.8	4.7	15.5
Qwen3-VL-235B-A22B	64	**40.3**	**39.3**	**15.6**	**31.7**
Qwen3.5-122B-A10B	64	37.8	36.5	15.2	29.8
Qwen3.5-27B	64	37.9	36.7	14.8	29.8
Qwen3-VL-32B	64	37.8	36.1	14.5	29.5
Qwen3.5-35B-A3B	64	37.0	36.7	13.5	29.1
Qwen3-VL-30B-A3B	64	33.6	32.9	11.5	26.0
Qwen3-VL-8B	64	32.7	32.0	11.9	25.5
Qwen3.5-9B	64	33.4	33.2	9.6	25.4
Qwen3-VL-4B	64	31.3	30.3	12.1	24.5
Keye-VL-1.5-8B	64	24.5	24.9	6.3	18.5
Qwen2.5-VL-7B	64	24.1	22.7	6.4	17.7
Qwen2.5-VL-3B	64	22.5	22.5	5.1	16.7
Qwen3.5-0.8B	64	21.0	21.1	4.7	15.6

web interface presents the video alongside the dialogue in turn order. Each turn displays its question, reference answer, supporting timestamps, question type, and dependency links to earlier turns.

Reviewers verify five aspects: (1) **answer correctness**, whether the reference answer is supported by the video; (2) **temporal grounding**, whether the annotated evidence intervals contain the required visual evidence; (3) **dependency validity**, whether the annotated parent turns and dependency types match the question's intended references; (4) **necessity**, whether the question becomes unanswerable without its parent turns, even with access to the video and the remaining dialogue; and (5) **non-leakage**, whether the question restates information that should instead be recalled from its parents. For turns with multiple parents, reviewers additionally assess whether each parent contributes indispensable information.

D.2 Human Baseline Evaluation

The human baseline in Table 2 is evaluated separately from dataset review, using the same web interface configured for evaluation. In this mode, annotators see only the inputs available to models, with reference answers, evidence annotations, and dependency labels hidden. Twenty annotators

evaluate all 326 dialogues, with each dialogue assigned to two annotators who answer independently. This yields 652 complete dialogue evaluations: 12 annotators each evaluate 33 dialogues, and the remaining eight each evaluate 32. Assignments are balanced across video categories, dialogue lengths, and dependency complexity. Where possible, annotators are assigned dialogues they did not previously review.

Evaluation inputs. Annotators receive the same 128 timestamped frames as the models, without access to the full video, reference answers, annotated evidence intervals, or dependency labels. Questions and answer options, where applicable, are presented sequentially. Each annotator can access the preceding questions and their own previous responses, but not the other annotator's answers. Dialogues are evaluated in full rather than divided among annotators, preserving a separate response history for each evaluation.

Scoring and aggregation. Human responses are graded using the same correctness criteria as model responses. TurnAcc, EpiDepAcc, and AnsDepAcc follow the definitions in subsection 4.2. For AnsDepAcc, chain credit is computed separately within each annotator's dialogue: a turn receives credit only if the current response and all of its Answer-dependency ancestors are correct. Scores are then aggregated across both evaluations of every dialogue, with equal weight given to each evaluation. Responses from different annotators are not combined through majority voting. Overall is the arithmetic mean of the three aggregated metrics.

E Additional Results and Examples

E.1 Effect of Frame Count

We evaluate 13 open-weight models with 16, 32, and 64 uniformly sampled frames per video under the same self-rollout protocol and scoring rules as the main experiments. Table 4 reports the results, complementing the 128-frame evaluation in Table 2.

Performance across frame budgets. Increasing the number of frames generally improves performance. All 13 models achieve higher Overall scores with 64 frames than with 16 frames. Qwen3-VL-235B-A22B ranks first at all three frame budgets, improving from 25.9% to 29.5% and 31.7% with 16, 32, and 64 frames, respectively. Its score rises further to 34.4% with 128 frames. Qwen3.5-27B similarly improves from 25.3% at 16 frames to 29.8% at 64 frames and 34.3% at 128 frames. These results suggest that denser temporal sampling provides useful visual evidence for answering questions throughout the dialogue.

Variation across models. The gains from additional frames vary considerably. From 16 to 64 frames, Qwen3-VL-32B improves by 6.2 percentage points overall, compared with 2.1 points for Keye-VL-1.5-8B. Increasing the frame budget from 32 to 64 improves Overall for 12 of the 13 models, with Qwen2.5-VL-3B showing a slight decrease from 16.9% to 16.7%. Qwen3.5-0.8B gains only 0.1 percentage points over the same interval. Thus, additional visual input generally benefits performance, but does not yield uniform gains across models.

Answer-dependent performance. AnsDepAcc remains below both TurnAcc and EpiDepAcc for every model at every frame budget in this ablation. For Qwen3-VL-235B-A22B, increasing the frame count from 16 to 64 raises AnsDepAcc from 11.7% to 15.6%, while Qwen3.5-27B improves from 8.2% to 14.8%. With 128 frames, their AnsDepAcc scores increase further to 18.6% and 20.9%, respectively. These gains show that additional visual evidence can improve correctness across answer-dependent chains, although the metric remains sensitive to both current-turn errors and upstream failures.

E.2 Additional Dataset Examples

Figure 6 and Figure 7 present more sports video dialogues in which later questions revisit previously discussed players and teams, illustrating the combined challenges of visual recognition, cross-turn reference resolution, and reasoning.



<Question 1>: At kickoff, what brand is on the ponytailed player's jersey?
<Ground truth 1>: Nike
Qwen3-VL-2B: Next (wrong)
Qwen3.5-122B: Nike (correct)
GPT-5.6: Nike (correct)
Gemini-3.1-Pro: Nike (correct)
Claude-Sonnet-5: Nike (correct)

<Question 2>: Is the player from T1 left or right of the center circle in the first attack?
<Ground truth 2>: Left
Qwen3-VL-2B: right (wrong)
Qwen3.5-122B: right (wrong)
GPT-5.6: Right (wrong)
Gemini-3.1-Pro: right (wrong)
Claude-Sonnet-5: Right (wrong)

<Question 3>: How does the player from T1 move relative to the center circle after T2?
<Ground truth 3>: Moves away from the center circle
Qwen3-VL-2B: left (wrong)
Qwen3.5-122B: away from (correct)
GPT-5.6: She moves away from the center circle into the attacking half. (correct)
Gemini-3.1-Pro: right (wrong)
Claude-Sonnet-5: The Birmingham (T1) player dribbles the ball forward, moving away from the center circle and further to the right, advancing toward Nottingham Forest's half. (correct)

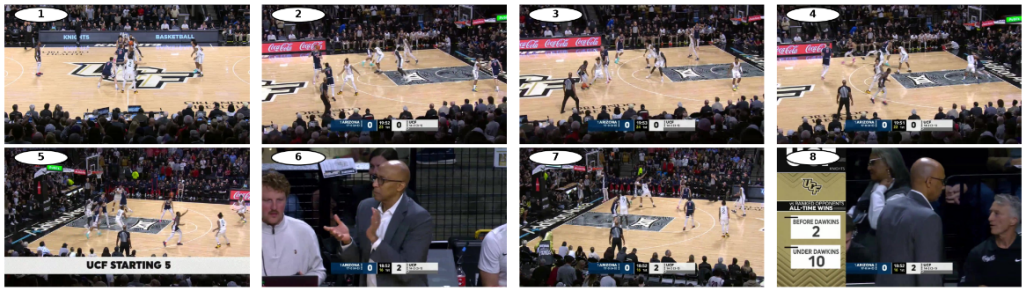
<Question 4>: What does the ponytailed player wearing number 16 do near the penalty arc?
<Ground truth 4>: Wipes the ball and throws it out.
Qwen3-VL-2B: runs (wrong)
Qwen3.5-122B: walks (wrong)
GPT-5.6: She shoots the ball toward goal. (wrong)
Gemini-3.1-Pro: runs and tackles down an opponent player (wrong)
Claude-Sonnet-5: The ponytailed player wearing number 16 dribbles the ball into the penalty arc area and takes a shot on goal. (wrong)

<Question 5>: Who scored Birmingham City's second goal?
<Ground truth 5>: LEIDHAMMAR
Qwen3-VL-2B: Lindsey von Flotow (wrong)
Qwen3.5-122B: Leidhammar (correct)
GPT-5.6: Ledrihamma (wrong)
Gemini-3.1-Pro: Leidhammar (correct)
Claude-Sonnet-5: Leidhammar (correct)

<Question 6>: What color is the opposing goalkeeper's jersey when the player from T5 scores?
<Ground truth 6>: Yellow
Qwen3-VL-2B: yellow (correct)
Qwen3.5-122B: yellow (correct)
GPT-5.6: Yellow (correct)
Gemini-3.1-Pro: yellow (correct)
Claude-Sonnet-5: Yellow (correct)

<Question 7>: Per the match summary graphic, who won?
<Ground truth 7>: Birmingham City
Qwen3-VL-2B: Birmingham City (correct)
Qwen3.5-122B: Birmingham City (correct)
GPT-5.6: Birmingham City (correct)
Gemini-3.1-Pro: Birmingham City (correct)
Claude-Sonnet-5: Birmingham City won 5-1 (correct)

Figure 6: **A soccer dialogue example from VideoMemBench.** Seven dialogue turns cover player identification, spatial tracking, and match events, with later questions referring to previously discussed players. Reference answers are shown in red, and model responses marked correct at the turn level are shown in green.



<Question 1>: At the opening jump ball, which white-jersey UCF player is closest to the referee, and what is their hairstyle?
<Ground truth 1>: UCF #2 with a top-knot
Qwen3-VL-2B: Player with a short, dark hairstyle (wrong)
Qwen3.5-122B: 2, flat top (wrong)
GPT-5.6: UCF No. 2, wearing braided cornrows. (wrong)
Gemini-3.1-Pro: The white-jersey UCF player closest to the referee is the one taking the jump ball (number 2), and he has short hair. (wrong)
Claude-Sonnet-5: The UCF player closest to the referee at the opening jump ball is the tall big man lining up for the tip, and he has short, closely cropped hair. (wrong)

<Question 2>: Right after the jump ball, is the UCF player from T1 guarding the dark-jersey number 0 ball handler?
<Ground truth 2>: Yes.
Qwen3-VL-2B: No (wrong)
Qwen3.5-122B: no (wrong)
GPT-5.6: Yes. (correct)
Gemini-3.1-Pro: Yes. (correct)
Claude-Sonnet-5: Yes, immediately after the tip UCF sets up its defense with a player picking up Arizona's No. 0 as he brings the ball up, guarding him closely near midcourt. (correct)

<Question 3>: What foul count is shown beneath the score for the dark-jersey team from T2?
<Ground truth 3>: 0 fouls (no foul count is displayed beneath Arizona's score during the play right after the jump ball)
Qwen3-VL-2B: 2 (wrong)
Qwen3.5-122B: 1 (wrong)
GPT-5.6: 1 (wrong)
Gemini-3.1-Pro: 1 (wrong)
Claude-Sonnet-5: 0 (correct)

<Question 4>: Do both the player from T1 and the player from T2 wear a wristband on the left arm?
<Options 4>: A. Yes, both players have a visible wristband on their left arm. | B. No, neither player has a visible wristband on their left arm. | C. Only the player with the top-knot has a visible wristband on his left arm. | D. Only the player wearing number 0 has a visible wristband on his left arm.
<Ground truth 4>: B. No, neither player has a visible wristband on their left arm.
Qwen3-VL-2B: B (correct)
Qwen3.5-122B: C (wrong)
GPT-5.6: A (wrong)
Gemini-3.1-Pro: B (correct)
Claude-Sonnet-5: B (correct)

<Question 5>: At 00:38, how is the dark-jersey team from T4 set up defensively relative to the three-point arc and the ball handler?
<Ground truth 5>: The Arizona team is playing a man-to-man defense, with all five defenders positioned inside or near the three-point arc, each closely tracking their respective offensive assignments.
Qwen3-VL-2B: The dark-jersey team from T4 is set up defensively in a zone, with the ball handler positioned near the three-point arc. (wrong)
Qwen3.5-122B: They are positioned inside the three-point arc in a man-to-man defense. (correct)
GPT-5.6: Four defenders are positioned inside the three-point arc, with one defender closely guarding the ball handler on the perimeter. (wrong)
Gemini-3.1-Pro: The entire dark-jersey team is positioned inside the three-point arc. (wrong)
Claude-Sonnet-5: At 00:38, Arizona's dark-jersey defense is spread out beyond the three-point arc in a man-to-man alignment, with defenders positioned above the arc tracking th... (wrong)

<Question 6>: At 01:11, what is the most likely emotional state of the coach in the grey suit?
<Options 6>: A. Frustration after a missed defensive stop | B. Encouraging a player on the bench | C. Celebrating a successful three-point shot | D. Disbelief at a referee's call | E. Calm satisfaction after a well-run play | F. Anxiety about the shot clock | G. Amusement at a teammate's mistake | H. Urgency while calling a timeout
<Ground truth 6>: A. Frustration after a missed defensive stop
Qwen3-VL-2B: A (correct)
Qwen3.5-122B: D (wrong)
GPT-5.6: E (wrong)
Gemini-3.1-Pro: A (correct)
Claude-Sonnet-5: E (wrong)

<Question 7>: For the dark-jersey player established in T2 and T4, what overall win-loss record is shown for that player's team on the score bug at the cited timestamp?
<Ground truth 7>: 17-0
Qwen3-VL-2B: 10-1 (wrong)
Qwen3.5-122B: 10-0 (wrong)
GPT-5.6: 17-0 (correct)
Gemini-3.1-Pro: 17-0 (correct)
Claude-Sonnet-5: 17-0 (correct)

<Question 8>: In the team-leaders scoreboard graphic, is the player from T1, T4, or T7 listed as a points leader?
<Ground truth 8>: False
Qwen3-VL-2B: T1 (wrong)
Qwen3.5-122B: T1 (wrong)
GPT-5.6: No. (correct)
Gemini-3.1-Pro: The player from T1 (wrong)
Claude-Sonnet-5: Yes — the player shown in T4 is UCF guard Themus Fulks (No. 1), and he is listed as a UCF points leader (4 pts) on the team-leaders scoreboard graphic. (wrong)

Figure 7: **A basketball dialogue example from VideoMemBench.** Selected video frames are shown above, followed by eight dialogue turns with reference answers and responses from five models. Reference answers are shown in red, and responses marked correct are shown in green.