Ask About the Right Moment: Training-Free Object Identity Reasoning for Video-Language Models

Di Yang[1,†] Xiaochang Li[1] Muyang Liu[1] Yufei Wang[2,*] Yanhai Xiong[1,*]

[1]College of William & Mary [2]SparcAI Inc.

dyang06@wm.edu xli59@wm.edu mliu22@wm.edu yxiong05@wm.edu yufei.wang@sparclab.ai



Figure 1. **Bind identity, then follow the track** (Perception Test `video_6159`, test set, Qwen3-VL-8B). A lime is visible under the middle mug before three identical mugs are shuffled. The direct answer of a video-language model (Video-LM) is that it stayed in the middle. Our agent asks which mug held the lime just after it was hidden, binds the answer to track #2, and reads its final position from that track. Fig. 2 shows the full method.

Abstract

Many questions about a video concern one physical object. Where does the ball end up after the cups are shuffled? Which card is the queen? How often was the mug covered? Open video-language models (Video-LMs) recognise such objects in single frames but do not keep them apart over time. On shuffled shell games, open models from eight families stay near chance, and so do published tool agents. We separate semantic identity binding from temporal identity maintenance. Binding should happen at a moment when the evidence for the object is visible, and maintenance should be left to a tracker. Track, Then Bind *implements this principle without training. A promptable video tracker follows every candidate object and draws it with a track number. A frozen vision-language model (VLM) answers one question about the identifying evidence, which binds the target to a track. The answer is then derived from a text description of the tracks. With Qwen3-VL-32B, the agent solves both VET-Bench tasks, where a fine-tuned trajectory model reaches 92% and 90%. On real cup games from the Perception Test, it raises accuracy from 35.5% to 52.3%. Without retuning, the same configuration also solves VET-Bench with Gemma 4 and GPT-5.5. Its track-derived visibility timelines add 8.8–17.9 points on TOC-Bench for open models. GPT-6.1 already answers 92.5% of the real cup games directly but 71.0% through our pipeline. The benefit of external tracking thus depends on the underlying model.*

1. Introduction

Many questions about a video are questions about one physical object. Which cup hides the ball after the shuffle? Which card is the queen once all three are face down? Did the mug leave the frame or was it covered before it reappeared?

*Corresponding authors. †Project lead.

Work in progress.

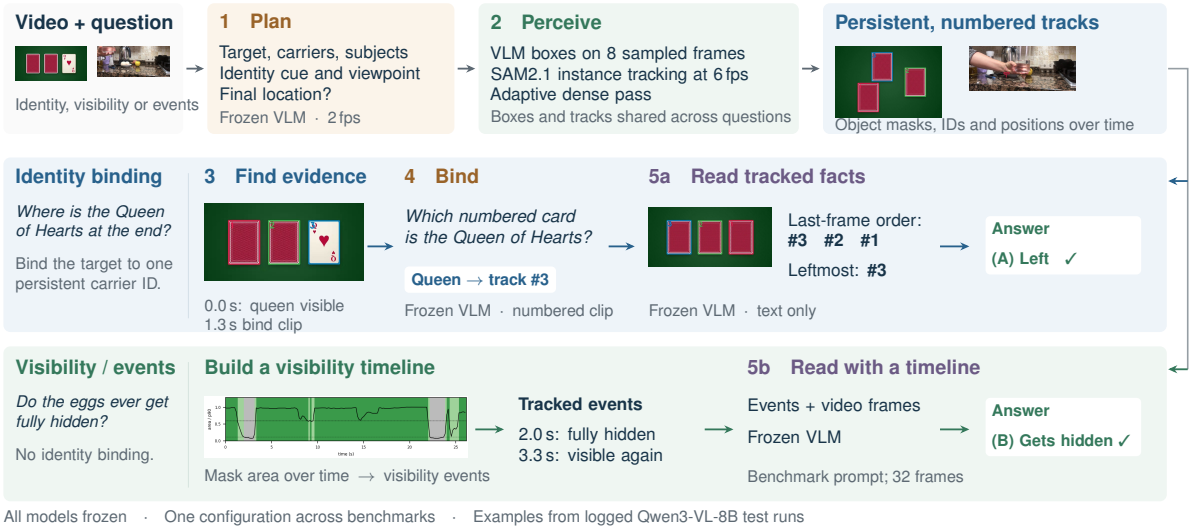


Figure 2. **Method.** A frozen vision-language model (VLM) plans the question (1) and boxes its entities, which SAM2.1 tracks as numbered instances (2). If a target must be told apart from look-alikes or containers, the VLM answers one question on a clip with numbered tracks (3–4). The answer is then read out from a text description of the tracks (5a; VET-Bench card monte `card_003`). Otherwise the tracks become a visibility timeline that is appended to the benchmark prompt (5b; TOC-Bench `8b0c33d38075`). Both examples are logged test runs with Qwen3-VL-8B.

Such questions arise wherever objects are moved, stacked or put away. Examples are assistants that remember where an item was left, robots whose own arm hides the object they manipulate, and laboratory procedures with rows of identical tubes. Answering them requires following one object through occlusion, containment and look-alike copies.

Open Video-LMs fail at this. On VET-Bench [1], open models from eight families answer 22–40% of the shell games, around the chance level of 33% (Appendix D). They include Qwen3-VL up to 235B parameters, Qwen3.5-397B, Gemma 4 and InternVL3.5. Only a model fine-tuned on synthetic trajectories (SGCoT) escapes, and it returns to chance on real cup-game videos from the Perception Test [2]. Thinking variants, a reinforcement-learned tool agent and three training-free tool agents also stay near chance (Table 1). Closed models differ. Gemini-3-Pro stays near chance, while GPT-5.5 answers 94–96% of VET-Bench and 68.2% of the real cup games. GPT-6.1 answers 100% and 92.5%. It also answers all card-monte videos that we render ourselves. On the cup games mirrored left to right it keeps 91.3%, which argues against memorised answers. The capability can therefore be learned end to end, but none of the open models we evaluated has it.

A central difficulty is that the model must re-derive which object is which from a sparse sample of frames. Identical objects that cross or disappear leave almost no visual trace from which to do so. A promptable video tracker such as SAM2 [4] has the opposite strengths. It follows each object mask through motion and brief occlusion, but does not know which object a question refers to. We study two design choices in combining them. *When* should the system ask the VLM about an object's identity? A hidden ball can only be recognised before it is hidden, a card only while it is face up. Once the identity is fixed, *what* should decide the answer: the VLM's reading of the video or the tracked state?

We answer them with two design rules and test each in isolation. First, the VLM is asked about an object's identity only through a static question about the moment at which the identity can be seen. We call this moment the evidence moment. A tracker then carries the identity, and the VLM is not asked again. Second, once the target is bound, the answer is read from the tracked state as text, without showing the video again. The tracked state is the position of each track at the end of the video. *Track, Then Bind* (Fig. 1) implements both rules without training. Every candidate carries a track number on the clip that the VLM sees, so its answer binds the target to one track. A plan made by the VLM decides which objects to follow and how the target can be recognised. As an extension to visibility and event questions that need no identity binding, the same framework supplies a visibility timeline built from the tracks. One program and one set of thresholds serve all benchmarks.

Our contributions are:

- **A diagnosis across model families.** Open Video-LMs from eight families, their thinking variants, a reinforcement-learned tool agent and three training-free ones stay near chance on shuffled cup games. Among closed models, GPT-5.5 and GPT-6.1 are far above chance; GPT-6.1 keeps its accuracy on mirrored videos.
- **Binding through a static question about the evidence**

Table 1. **Accuracy (%) on the test sets.** VET: VET-Bench, 50 videos per task. PT: Perception Test cup games filtered by the VET-Bench authors (107 videos); PT-65: its 65 videos with three cups and at least one swap. TOC: TOC-Bench test split (1,845 questions). Gray numbers are cited from other papers. The VET, PT and PT-65 accuracies of Gemini-3-Pro, Doubao-Seed-2.0-mini and GLM-4.6V-Flash are from VET-Bench [1], rescored from the predictions its authors released. The TOC accuracy of Gemini is from TOC-Bench [3]. All black numbers are our runs with each benchmark's prompt, frames and scorer. Best result per column in bold, best without a closed model underlined.

Method	Type	VET cup	VET card	PT	PT-65	TOC
Random	–	33.3	33.3	33.3	33.3	30.3
Closed Video-LMs						
Gemini-3-Pro [62]	thinking (API default)[§]	34.0	40.0	36.4	30.8	43.9*
Doubao-Seed-2.0-mini [65]	API default[§]	30.0	32.0	46.7	40.0	–
GPT-5.5 [66]	reasoning, medium effort	94.0	96.0	68.2	72.3	47.6
GPT-6.1 [67]	reasoning, medium effort	**100**	**100**	**92.5**	**90.8**	**52.5**
Open Video-LMs						
Qwen3-VL-8B [17]	direct answer	28.0	28.0	33.6	27.7	25.1
Qwen3-VL-32B [17]	direct answer	32.0	38.0	35.5	33.8	37.1
Qwen3-VL-235B-A22B [17]	direct answer	22.0	38.0	38.3	27.7	32.8
InternVL3.5-8B [69]	direct answer	32.0	32.0	43.0[‡]	38.5[‡]	31.3
Gemma 4 31B [68]	direct answer	36.0	28.0	36.4	32.3	40.0
GLM-4.6V-Flash [64]	thinking (API default)[§]	34.0	28.0	43.9	41.5	–
Molmo2-8B [18]	direct answer	30.0	40.0	33.6	30.8	32.3
Reasoning-enhanced and tool-using						
Qwen3-VL-8B-Thinking [17]	thinking	28.0	24.0	37.4	36.9	30.0
Qwen3-VL-32B-Thinking [17]	thinking	24.0	22.0	33.6	32.3	36.9
Molmo2-SGCoT [1]	trajectory chain-of-thought, fine-tuned	92.0	90.0	32.7	38.5	39.2
Open-o3-Video-v2-8B [5]	reinforcement-learned crop tool	26.0	30.0	38.3	33.8	33.9[†]
SpatialClaw [15] (Qwen3-VL-32B)	code agent, SAM3 + depth	28.0	36.0	26.2	30.8	35.7
VideoAgent [7] (Qwen3-VL-32B)	object-memory agent	40.0	28.0	37.4	36.9	33.0
VideoINSTA [8] (Qwen3-VL-32B)	clip-reasoning agent	42.0	28.0	42.1	38.5	34.5
Track, Then Bind (Qwen3-VL-8B)	training-free agent	**100**	**100**	51.4	52.3	43.0
Track, Then Bind (Qwen3-VL-32B)	training-free agent	**100**	**100**	<u>52.3</u>	<u>55.4</u>	48.6
Track, Then Bind (Gemma 4 31B)	training-free agent	**100**	**100**	48.6	49.2	<u>48.8</u>

–: not evaluated. *Reported on all 2,323 TOC-Bench questions (Gemini: version 3.1-Pro [63]). †The official evaluator cannot express the 89 ordering questions, which count as wrong; on the other 1,756 questions the accuracy is 35.6%. Random: expected accuracy of uniform guessing over each question's answer space. SpatialClaw, VideoAgent and VideoINSTA run their official code with Qwen3-VL-32B-Instruct as the model (Appendix C). GPT-5.5 and GPT-6.1 answer through the OpenAI API with frames as images and reason at the default effort, medium (Appendix D). §Run by the VET-Bench authors through each API with its default thinking setting: always on for Gemini-3-Pro (level high), on for GLM-4.6V (the model decides per question), undocumented for Doubao. ‡InternVL3.5 omits the required `\boxed{}` answer in 83 of 107 direct answers, so we read its first option letter; the VET-Bench scorer gives 9.3 (PT) and 7.7 (PT-65). Qwen3-VL rows answer in vLLM, except the TOC-Bench thinking rows, which use the TOC-Bench authors' Transformers runner; Molmo2 rows use the authors' Transformers code (Appendix C). Qwen3-VL answering directly in Transformers, the engine of our agent, reaches 22.0 / 34.0 / 31.8 / 29.2 / 25.2 at 8B and 24.0 / 28.0 / 27.1 / 27.7 / 36.9 at 32B (same columns).

moment. With the tracks and the clip held fixed, asking which cup the target is under right after it is hidden, instead of where it is at the end, raises accuracy on every test set (Table 4). At 8B it lifts VET-Bench from near chance to 100% and the real cup games by 15.0 points. Drawing the same tracks and asking the question directly stays at 32.7–44.0% at 8B, against 51.4–100% for the agent, and 12.1 points below the agent on real cup games at 32B.

- **Answering from the tracked state.** Given the same correct tracked state together with the shuffled video, the VLM follows its own reading of the video. The synthetic cup game drops from 100% to 38.0% at 8B and to 76.0% at 32B, and real cup games drop by 3.7–4.7 points.
- **One configuration across benchmarks and VLM families.** The agent is tuned once on development videos and runs unchanged with Qwen3-VL, InternVL3.5, Gemma 4 and GPT-5.5. It solves VET-Bench with three of these families. Its track-derived timelines raise TOC-Bench accuracy for every open model, by 8.8–17.9 points.

2. Related Work

Video-LMs and their temporal limits. Video-LMs pair a visual encoder with a language model and are tuned on video instructions [22–24]. Current open families such as Qwen2.5-VL [20], InternVL3 [21], Qwen3-VL [17]

and Molmo2 [18] add dynamic frame rates, timestamps and grounding outputs. Benchmarks that isolate time find that much of their accuracy comes from single frames. MVBench [25] and NExT-QA [26] probe temporal and causal actions, EgoSchema [27] and Video-MME [19] long videos, TempCompass [28] finds that many models barely exceed chance once single-frame and language shortcuts are removed, and TOMATO [29] measures a 57-point gap to humans on questions that need the frame order. VELOCITI [30] finds weak binding of agents to actions across events, and VSI-Bench [31] limited visual-spatial reasoning. VET-Bench [1] and TOC-Bench [3] target object identity directly, the setting of this paper.

Reasoning-enhanced Video-LMs. Chain-of-thought prompting [32] and its video variants [33] write intermediate steps before the answer, and self-consistency [34] votes over sampled chains. On VSI-Bench neither improves spatial answers [31]. Reinforcement-learned models such as Video-R1 [35] and Open-o3-Video-v2 [5] add textual or grounded reasoning traces. SGCoT [1] fine-tunes Molmo2 to write object trajectories before answering, and RGA3 [6] trains a Video-LM to refer to and ground objects with spatio-temporal prompts. In all of these, identity is held in learned representations or generated tokens, which our diagnosis finds unreliable in open models once objects look alike or are hidden. Our call-matched baseline (Table 3) is self-consistency with one sample per agent VLM call.

Object permanence. Infants expect a hidden object to persist [36] and parse scenes into cohesive bodies that move continuously [37]. In vision, CATER [38] and CLEVRER [39] test these abilities on synthetic videos with containment and collisions. Learned models localise contained and carried objects [40], keep tracks through full occlusion [41], learn intuitive physics from object-centric codes [42], and segment objects inside containers [16]. Xie and Alanwar [72] find that video world models keep visible objects but drop hidden ones. We use a frozen tracker and VLM to maintain explicit identity associations across occlusion and containment.

Segmentation and tracking. Promptable segmentation [43] extends to video with a streaming memory in SAM2 [4] and to concept prompts in SAM3 [44]. Memory-based video object segmentation [45, 46], point tracking [47] and tracking-by-detection [48] keep identities through motion and short occlusion, and open-vocabulary detectors [49] supply the initial boxes. None of them knows which object a question refers to. Our agent obtains the boxes from the VLM and the identities from SAM2.1.

Visual prompting with tracks. Marks drawn on an image steer a model to a region: a red circle for CLIP [50], numbered regions in Set-of-Mark [10], and free-form prompts learned by ViP-LLaVA [51]. Coarse Correspondences [11] draws tracker IDs on video frames and lets the model answer the whole question. Elysium [52], VideoRefer [53] and Sa2VA [54] instead train Video-LMs to track, refer to or segment objects. We draw the same kind of numbered tracks, but the model only answers a static question about one moment, and the answer comes from the tracks. Section 5.3 shows that the overlay alone recovers little of the gain at 8B and part of it at 32B.

Tool-using agents. VisProg [55] and ViperGPT [56] compose vision modules through generated programs, and ReAct [57] interleaves reasoning with tool calls. For video, agents retrieve frames iteratively [58], summarise clip captions [8, 59], plan over a symbolic memory [60], or store object-centric tracking states that a large language model (LLM) queries [7]. STRAND [9] links per-chunk object states by attribute and position similarity, making identical objects difficult to separate. SpatialClaw [15] writes code that calls SAM3 and Depth Anything 3 [61]. AgentR-VOS [12] reasons over SAM3 tracks to segment a referred, visible object, and Dual-Agent [13] and ACTrack [14] use a VLM to verify or re-prompt a tracker. These agents, like ours, keep object state outside the model, and they leave the final judgement of which object is which to the model. Our design differs in when the model is asked about identity and in what decides the answer once identity is fixed.

3. Problem Formulation

Setting. A video $V = (I_1, \ldots, I_T)$ shows a set of physical objects $\mathcal{O}$. A question q with answer space $\mathcal{A}$ concerns a *target* $o^\star \in \mathcal{O}$. It may ask where the target is at the end, which of several objects it is, or when and how often an object is visible. The objects that hide, cover or resemble the target are its *carriers* $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathcal{O}$. A Video-LM f answers directly from a sparse frame sample $V_S = (I_t)_{t \in S}$:

$$\hat{a}_{\text{direct}} = f(V_S, q), \qquad \hat{a}_{\text{direct}} \in \mathcal{A}. \tag{1}$$

Our agent uses the same frozen model f, which we call the *VLM* when it is asked about single frames, short clips or text.

Evidence moments. An *evidence moment* t has a local observation window $W_t = V_{[t-\delta,\, t+\delta]}$ in which the target or its containing carrier can be identified:

$$\mathcal{E} = \{\, t : o^\star \text{ is identifiable among } \mathcal{C} \text{ from } W_t \,\}. \tag{2}$$

The target is identified either *by its look*, as the queen while it is face up, or as *hidden in a carrier*, as the ball at the moment it is covered. In shell and card games, this evidence often disappears before the final frame. A direct answer (Eq. 1) must then carry identity from an evidence moment $t_e \in \mathcal{E}$ to T through a sparse sample of frames in which identical carriers cross. Under occlusion over time, q asks when an object disappears or returns, and these events may fall between the sampled frames.

Principle. A tracker follows each carrier c_k with a mask track $\tau_k = (m_{k,1}, \ldots, m_{k,T})$. We fix identity once, at an evidence moment, and keep it outside the model afterwards. The answer factors into a *bind* and a *read-out*:

$$k^\star = f\big(W_\mathrm{b} \oplus \{\tau_k\}_{k=1}^{K},\ q_\mathrm{b}\big), \tag{3}$$

$$\hat{a} = f\big(q,\ \mathcal{F}(k^\star, \{\tau_k\}_{k=1}^{K})\big). \tag{4}$$

In Eq. 3, the bind clip W_b is W_{t_e} when a moment is located, or the sampled whole video otherwise. The operator $\oplus$ draws each track with its number k on this clip, and q_b asks about the evidence moment in it. The VLM returns the track index $k^\star$ of the target. In Eq. 4, $\mathcal{F}$ states the tracked state in text, for example the left-to-right order of the tracks in the last frame. The VLM answers q from these facts, without the video. Identity thus crosses the interval $(t_e, T]$ only through the track $\tau_{k^\star}$, and the VLM never has to recognise the target after the evidence is gone. Questions without a target to bind, such as how often an object is hidden, receive a timeline:

$$\hat{a} = f\big(V_S,\ q \oplus \mathcal{L}(\{\tau_k\})\big), \tag{5}$$

where $\mathcal{L}$ lists per-object visibility segments and events computed from mask areas; $\oplus$ appends them to the prompt.

4. Method

Track, Then Bind organises inference into five steps (Fig. 2; Algorithm 1 in Appendix A). A plan decides which steps run, and all thresholds are the same for every benchmark.

Step 1: Plan. The VLM watches V at 2 fps and fills a fixed schema. It lists the entities in q with their roles: the target, its carriers, and other *subjects*. It states how the target is recognised, by its look or as hidden in a carrier, and whose left and right the question uses. A separate text-only question asks whether q is about the target's final location. If the target is recognised by its look and no carrier is named, all objects of the target's category become its carriers. The plan selects these operations from the question using the same schema across benchmarks.

Step 2: Perceive. The VLM boxes all entities in one call per frame, on 8 frames spaced evenly over the video. Entities found on none of them receive a second round of 8 frames in between. SAM2.1 [4] is prompted on the earliest frame that shows the most common number of instances, or on the largest box for a single object. It propagates each instance forward and backward at 6 fps. Carrier tracks that overlap, move more than a quarter of their width per frame, or end early trigger a second pass. This pass tracks the carriers and the target on every k-th native frame. Here k is the native frame count over 300, rounded, and at least 1. The 12 s VET-Bench videos are thus tracked at their native 30 fps. Tracking all candidates jointly separates identical objects by their trajectories, and the dense pass is spent only where they cross. All questions on one video share its boxes and tracks.

Step 3: Find evidence moments. The bind question of step 4 always names the moment it is about: the target in view, or the target right after it is hidden. Step 3 tries to locate that moment t_e in the tracks, so that the VLM sees a short clip (Eq. 3). For a target recognised by its look, the moments are the frames on which the VLM found it in step 2. A mask track alone cannot tell a face-up card from its back. For a hidden target, the moment is its last view before its own track vanishes inside the frame. Small hidden objects are often lost by the tracker or never vanish cleanly. In that case no moment is located, and the clip covers the whole video. The VLM then finds the moment named by the question itself. Section 5.3 reports how often each case occurs (Table 5) and when a located moment matters (Table 4).

Step 4: Bind. Each carrier track is drawn as a numbered box on a 4 fps clip, with at most 96 frames. The clip spans 3 s around a located moment ($\delta = 1.5$ s), or the whole video. The VLM answers one static question q_b and returns $k^\star$. For a target recognised by its look it names the numbered carrier that is the target. For a hidden target it names the carrier the target is under right after it is hidden. When several moments exist, the last one with a valid number decides. The question concerns a single moment, not the history after it, so the VLM never has to follow the target through the shuffle.

Step 5: Read out. A bound target is answered from facts. The agent states them in plain words, for example "the hidden object was hidden under mug #2 … so the hidden object is under the first mug from the camera's left". If q uses a person's point of view, one more VLM call on four frames asks whether that person is opposite the camera. Its answer is added as a viewpoint fact. The facts then also give the person's left-to-right position. These facts are $\mathcal{F}$ in Eq. 4. The VLM answers q from them alone, without the video, making the tracked facts the evidence for read-out. Questions without a bound target use the timeline $\mathcal{L}$ of Eq. 5. Per-subject visibility segments, an ordered list of events and the coarse direction of motion are appended to the benchmark's prompt. The VLM then answers with the benchmark's own frames, template and parser. An answer that cannot be parsed falls back to the direct answer.

5. Experiments

5.1. Experimental Setup

Benchmarks. We evaluate on three benchmarks about object identity and temporal object consistency. VET-Bench [1] has 50 synthetic shell-game videos (*cup*) and 50 card-monte videos (*card*). The Perception Test [2] cup-game subset (*PT*) has 107 real test videos. The VET-Bench authors filtered it to remove single-frame shortcuts. Following them, we also report its 65 videos with three cups and at least one swap (*PT-65*). TOC-Bench [3] asks about object consistency in real videos along ten dimensions, such as event counts,

durations and reappearance. Almost none of its questions needs binding (Table 5), so it tests visibility timelines. We score each benchmark with its own answer parser.

Development and test data. Design choices and thresholds are selected on development data. These are 195 cup-game videos from the Perception Test validation split, 56 card-monte videos rendered with the VET-Bench generator, and 478 TOC-Bench questions. We split the 2,323 TOC-Bench questions by video into these 478 development questions and 1,845 test questions. The configuration is frozen and then run once on the test sets. Each ablation on the test sets changes one step of the frozen configuration.

Baselines. Open Video-LMs answer directly with each benchmark's prompt, frames and parser. We run them in vLLM [71]. The footnote of Table 1 also gives Qwen3-VL answering in Hugging Face Transformers, the engine of our agent. The reasoning-enhanced baselines are the Qwen3-VL thinking variants and two trained models. Molmo2-SGCoT [1] is fine-tuned to write object trajectories before it answers. Open-o3-Video-v2 [5] is trained with reinforcement learning to crop and inspect frames. We run the official code of three training-free agents with Qwen3-VL-32B-Instruct, the backbone of our 32B agent. SpatialClaw [15] writes Python code calling SAM3 [44] and Depth Anything 3 [61]. VideoAgent [7] builds an object memory from a detector, a tracker and re-identification, and queries it with an LLM. VideoINSTA [8] reasons over captions and object lists of event clips. Section 5.2 adds call-matched and increased-input baselines. Section 5.2 runs the agent with InternVL3.5-8B [69], Gemma 4 31B [68] and GPT-5.5, and compares it with GPT-6.1 [67] answering directly. Appendices C and D list all settings.

Implementation details. The VLM is Qwen3-VL-8B- or 32B-Instruct [17] with greedy decoding, and the tracker is SAM2.1-hiera-large [4]. A timeline marks an object as fully visible when its mask covers at least 0.6 of its 90th-percentile area, and as partly hidden down to 0.12 of it. Segments shorter than 0.5 s are merged. A disappearance counts as leaving the frame if the last box lies within 2% of the border. All prompts are listed in Appendix I. Paired comparisons report item-level bootstrap 95% confidence intervals (CI) and exact McNemar tests.

5.2. Main Results

Comparison with existing methods. Table 1 compares Track, Then Bind with direct answers, reasoning-enhanced models and tool agents. On VET-Bench, the only open method substantially above chance on both tasks is SGCoT, which is fine-tuned on synthetic trajectories and reaches 92.0% and 90.0%. Our agent solves both tasks with both backbones and without training. Among closed models, GPT-5.5 answers 94.0% and 96.0%, and GPT-6.1 solves both. On the 107 real cup games, SGCoT falls back to 32.7%. The agent reaches 51.4% at 8B and 52.3% at 32B. That is 17.8 (CI [3.7, 30.8]) and 16.8 points (CI [3.7, 29.9]) above the same models answering directly. The shuffled three-cup subset separates the methods most clearly. Every open baseline stays within ten points of chance on it, and the 32B agent reaches 55.4%. GPT-5.5 and GPT-6.1 reach 72.3% and 90.8%. On TOC-Bench the agent reaches 43.0% at 8B and 48.6% at 32B, 17.9 (CI [15.4, 20.4]) and 11.5 points (CI [9.0, 14.0]) above the direct answers. The 32B agent is also 11.7 points above the 32B thinking variant and 14.7 points above Open-o3-Video-v2. GPT-5.5 answers 47.6% and GPT-6.1 52.5% of these questions.

Table 2. **One configuration across VLM families** (test sets, %). D: the VLM answering directly (GPT-5.5 and GPT-6.1 reason at medium effort); T: Track, Then Bind with the same VLM and the frozen configuration. ‡First option letter, as in Table 1; the VET-Bench scorer gives 9.3 (D) and 34.6 (T). –: not evaluated.

	VET cup		VET card		PT		TOC	
VLM	D	T	D	T	D	T	D	T
Qwen3-VL-8B	28.0	100	28.0	100	33.6	51.4	25.1	43.0
Qwen3-VL-32B	32.0	100	38.0	100	35.5	52.3	37.1	48.6
InternVL3.5-8B	32.0	48.0	32.0	32.0	43.0‡	35.5‡	31.3	41.7
Gemma 4 31B	36.0	100	28.0	100	36.4	48.6	40.0	48.8
GPT-5.5	94.0	100	96.0	100	68.2	61.7	47.6	49.2
GPT-6.1	100	100	100	100	92.5	71.0	52.5	57.7

Comparison with tool-using agents. SpatialClaw, VideoAgent and VideoINSTA run with our 32B backbone. All three stay within ten points of chance on the cup games and below the direct answer on TOC-Bench. Our 32B agent is 10.3–72.0 points higher on every set (Appendix E). Only the 10.3-point gap to VideoINSTA on real cup games is not significant. The tools alone do not carry identity. SpatialClaw propagates masks through only a quarter of the VET-Bench videos, VideoAgent's detector misses playing cards, and VideoINSTA keeps no object track.

Across VLM families. Table 2 runs the frozen configuration with four VLM families. Each backend adapts only its model's input format and box convention (Appendix D). With Qwen3-VL, Gemma 4 and GPT-5.5, the agent solves both VET-Bench tasks. The plan of InternVL3.5 names no carrier on card monte, so nothing is bound. Given the plans of Qwen3-VL-8B, InternVL3.5 answers 98% of card monte. On TOC-Bench the agent answers from track-derived timelines and binds almost no question (Table 5). These timelines narrow the gap between VLMs. Direct answers span 25.1–47.6%, and the agent spans 41.7–49.2%. The gain shrinks as the direct answer improves, from 17.9 points with Qwen3-VL-8B to 1.6 points with GPT-5.5 (CI $[-0.7, 4.0]$). On real cup games a stronger VLM still helps the agent: 48.6–52.3% with Qwen3-VL and Gemma 4, 61.7% with GPT-5.5 and 71.0% with GPT-6.1. InternVL3.5 gains nothing there (43.0% direct, 35.5% agent). For the strongest model, the

Table 3. **Call-matched and increased-input baselines** (test sets, %). *Self-consistency* [34]: k answers sampled at the model's recommended temperature and a majority vote, with k the agent's mean VLM calls on that benchmark (at most 16). *More input tokens*: one answer from 16 fps and larger frames. *SpatialClaw*: as in Table 1. Calls: VLM calls per question on cup/card/PT/TOC. Tokens: input tokens per PT question. Transformers runs: Appendix E.

VLM	Baseline	cup	card	PT	TOC	Calls	Tokens
8B	direct answer	28.0	28.0	33.6	25.1	1	12k
	self-consistency, as many calls	28.0	30.0	46.7	25.9	16/12/13/11	157k
	direct answer, more input tokens	24.0	30.0	27.1	24.7	1	34k
	Track, Then Bind	**100**	**100**	**51.4**	**43.0**	19/12/13/11	25k
32B	direct answer	32.0	38.0	35.5	37.1	1	12k
	self-consistency, as many calls	32.0	30.0	37.4	37.3	16/12/15/11	181k
	direct answer, more input tokens	30.0	24.0	31.8	36.7	1	34k
	SpatialClaw code agent	28.0	36.0	26.2	35.7	14/9/18/10	508k
	Track, Then Bind	**100**	**100**	**52.3**	**48.6**	19/12/15/11	27k

external pipeline becomes the bottleneck. GPT-6.1 answers 92.5% directly, 21.5 points above its agent (CI [13.1, 29.9]). Most of the agent's 31 errors with GPT-6.1 come from the bound track (Appendix D).

Comparison with larger inference budgets. The agent issues 11–19 VLM calls and 7–27k input tokens per question (Appendix A), while a direct answer issues one call. Table 3 increases the direct answer's call or input budget. Self-consistency samples as many answers as the agent makes calls and takes the majority vote. The second baseline increases input tokens in a single answer, with 16 fps and larger frames. On VET-Bench the voted answers stay at 28–32%. After the shuffle no frame shows where the target is, so more samples of the same frames add no evidence. On the real cup games, the 8B vote over 13 samples reaches 46.7%, against 33.6% for a single answer. The agent is 4.7 points higher (CI $[-9.3, 18.7]$) and reads one sixth of the input tokens (25k against 157k). At 32B the vote reaches 37.4%, and the agent is 15.0 points higher (CI [2.8, 27.1]). On TOC-Bench the vote changes the direct answer by less than one point at both sizes. A single answer with more input tokens reads 34k tokens per cup game, and its accuracy falls below that of the plain direct answer. SpatialClaw spends the largest budget, about 19 times the agent's input tokens per real cup game.

5.3. Ablation Study

The principle has three parts. A static question binds the target to a track, it is asked about the evidence moment, and the answer is read from the tracks. Table 4 removes or replaces each part on the test sets, with the plan, the boxes and the tracks fixed, and also removes the located clip.

Binding. Without the bind question, the numbered tracks are drawn on the whole video and the benchmark question is asked directly, as in Coarse Correspondences [11]. The 8B model then answers 44.0%, 40.0% and 32.7% of the cup, card and real cup games, close to its direct answer. The 32B model reads the numbers better. It solves card monte and reaches 78.0% on the cup game. On the real cup games it stays at 40.2%, 12.1 points below the agent (CI [0.0, 24.3]).

Table 4. **Ablation of the three steps** (test sets, accuracy %). Each variant keeps the plan, boxes and tracks and changes one step. *Tracks drawn, question asked directly*: the numbered whole-video clip with the benchmark question, as in visual prompting with track IDs. *Bind question on the whole video*: no located moment. *Bind question about the end*: the same clip, asked where the target is at the end. *Facts read out with the video*: the same facts are appended to the benchmark prompt, which is answered with the video.

VLM	Variant	cup	card	PT
8B	tracks drawn, question asked directly	44.0	40.0	32.7
	bind question on the whole video	100	100	53.3
	bind question about the end	28.0	36.0	38.3
	facts read out with the video	38.0	72.0	46.7
	Track, Then Bind	100	100	51.4
32B	tracks drawn, question asked directly	78.0	100	40.2
	bind question on the whole video	72.0	100	54.2
	bind question about the end	46.0	88.0	49.5
	facts read out with the video	76.0	96.0	48.6
	Track, Then Bind	100	100	52.3

Evidence moment. With the same tracks and the same whole-video clip, asking where the target is at the end, instead of right after it is hidden, lowers accuracy on every test set at both scales. At 8B, VET-Bench falls to 28.0% and 36.0%, near chance, and the real cup games fall by 15.0 points (CI $[-25.2, -4.7]$).

Located clip. Asking the bind question on the numbered clip of the whole video keeps both VET-Bench tasks at 100% at 8B. On the real cup games it gives 53.3%, 1.9 points above the agent (CI $[-1.9, 5.6]$). At 32B, card monte stays at 100% and the real cup games reach 54.2%, but the cup game drops to 72.0%. A located moment shortens the bind clip from about 47 to 12 frames on the cup game, and the 32B model needs the short clip there.

Text-only read-out. Appending the same facts to the benchmark prompt and answering with the video lowers accuracy on all three sets. At 8B the cup game drops from 100% to 38.0% and card monte to 72.0%. At 32B they drop to 76.0% and 96.0%. Reintroducing the shuffled video can reduce accuracy even when the tracked facts are unchanged.

Bind clips. Table 5 shows which bind clip step 3 selects. On VET-Bench every question is bound on a located clip: 12 frames around the hiding of the ball, or 6 frames while the queen is face up. In real cup games the hidden object is small. Its own track yields a moment in only 21% (8B) and 11% (32B) of the videos. The other videos are bound on the whole numbered clip, about 60 frames over 15 s. They are answered more often than the located ones (55.0% against 40.9% at 8B). On real videos the gain thus comes from the static question and the tracks; a short window adds little.

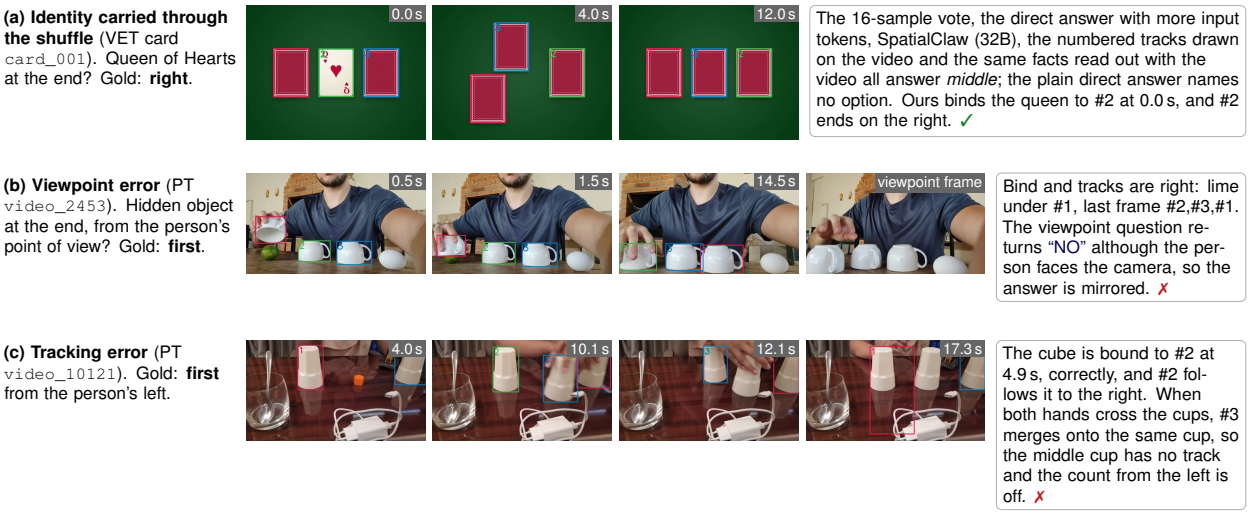


Figure 3. **Qualitative results** (test videos, Qwen3-VL-8B unless noted). (a) A success that no method without binding reproduces. (b, c) Two failures on real cup games: a wrong viewpoint fact with correct tracks, and a cup whose track is lost when both hands cross the cups. Numbered boxes are the agent's tracks as the VLM sees them.

Table 5. **Bind clips selected by step 3** (test sets). *Located*: the VLM sees a 3 s clip around the evidence moment. *Whole video*: no moment is found, and the bind question is asked on the numbered clip of the whole video. *No binding*: the plan finds no target to tell apart, or fewer than two carrier tracks, and the question is answered with a timeline. Each cell gives the share of questions and, in parentheses, the accuracy on them. Frames: mean frames per bind call; –: no bind call is made.

Test set	Bind clip	8B	32B	Frames
VET cup	located (hidden)	100% (100)	100% (100)	12
VET card	located (in view)	100% (100)	100% (100)	6
PT	located (hidden)	20.6% (40.9)	11.2% (33.3)	12
PT	whole video	74.8% (55.0)	81.3% (57.5)	60
PT	no binding	4.7% (40.0)	7.5% (25.0)	–
TOC	no binding	100% (43.0)	99.9% (48.7)	–

5.4. Qualitative Analysis

Success case. In the card-monte video of Fig. 3(a), the queen is face up for one second. Every method that leaves the final judgement to the VLM answers that the queen stayed in the middle. SpatialClaw (32B) loses the queen's mask once the card turns over and then guesses the middle (Fig. 4). Our agent asks which card is the queen only while it is face up. The tracks then carry the queen to the right.

Failure cases. We annotated whether the person in each real cup game faces the camera (Appendix D) and attribute each wrong answer under this viewpoint. At 8B, 7 of the 52 errors come from a wrong viewpoint fact (Fig. 3b) and 38 from binding or tracking. We inspected these 38 by hand: the bound track never moves to another cup. The errors arise in the bind answer, or when the track of another cup is lost, which shifts the count from the left (Fig. 3c). At 32B, 14 errors come from the viewpoint fact and 22 from binding or tracking. Most of the remaining errors at both scales are questions that are never bound.

6. Limitations

The agent inherits the errors of its tracker and of its VLM's plan. Identical cups crossed by hand can still confuse the tracker, very small hidden objects are rarely tracked, and a plan that names no carrier binds nothing. Real-video evidence for binding comes from the Perception Test cup games, the only real benchmark we know of in which a hidden object moves among look-alike carriers. For a model that already solves these games directly, such as GPT-6.1, the external pipeline adds errors. The agent also makes several VLM calls per question (Table 7).

7. Conclusion

Open Video-LMs lose object identity when the identifying evidence disappears or the objects look alike, while GPT-6.1 solves the same games directly. Track, Then Bind asks the VLM one question about the moment the evidence is visible, then carries the identity with tracks and reads the answer from the tracked state. Without training, it solves both VET-Bench tasks with three VLM families and raises Qwen3-VL-32B from 35.5% to 52.3% on real cup games. Its track-derived timelines add 8.8–17.9 points on TOC-Bench to open models. The benefit of external tracking depends on the model: GPT-6.1, which already solves these games directly, loses accuracy through the pipeline.

LLM Usage Statement

Vision-language models are components of the method and the baselines studied in this paper. Beyond these, LLMs were used as coding assistants and to polish the wording of the manuscript. The authors verified all code and results and take full responsibility for the content.

References

[1] T. Liu and W. S. Lee. Can Vision-Language Models Solve the Shell Game? arXiv:2603.08436, 2026. 2, 3, 4, 5, 6, 12, 14

[2] V. Pătrăucean et al. Perception Test: A Diagnostic Benchmark for Multimodal Video Models. NeurIPS Datasets and Benchmarks, 2023. 2, 5, 12

[3] J. Chen et al. TOC-Bench: A Temporal Object Consistency Benchmark for Video Large Language Models. arXiv:2605.09904, 2026. 3, 4, 5, 12, 15

[4] N. Ravi et al. SAM 2: Segment Anything in Images and Videos. ICLR, 2025. 2, 4, 5, 6

[5] J. Meng et al. Open-o3-Video-v2: Seek, Ground, and Reason with Verifiable Spatio-Temporal Evidence. arXiv:2610.17656, 2026. 3, 4, 6

[6] H. Wang et al. Object-centric Video Question Answering with Visual Grounding and Referring. ICCV, 2025. 4

[7] Y. Fan et al. VideoAgent: A Memory-augmented Multimodal Agent for Video Understanding. ECCV, 2024. 3, 4, 6

[8] R. Liao et al. VideoINSTA: Zero-shot Long Video Understanding via Informative Spatial-Temporal Reasoning with LLMs. Findings of EMNLP, 2024. 3, 4, 6

[9] T. Nguyen et al. STRAND: Benchmarking and Improving Object-Centric Spatio-Temporal Monitoring in Video Large Language Models. arXiv:2609.29607, 2026. 4

[10] J. Yang et al. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V. arXiv:2310.11441, 2023. 4

[11] B. Liu et al. Coarse Correspondences Boost Spatial-Temporal Reasoning in Multimodal Language Model. CVPR, 2025. 4, 7

[12] W. Jin et al. AgentRVOS: Reasoning over Object Tracks for Zero-Shot Referring Video Object Segmentation. arXiv:2603.23489, 2026. 4

[13] V. Bundele et al. When Trackers Fail: VLM-Guided Verification and Recovery for Robust Video Object Segmentation. NeurIPS, 2026 (arXiv:2610.15929). 4

[14] W. Cai et al. Models as Tools: An Agentic Coordination Framework for Unified Multimodal Visual Tracking. arXiv:2608.00847, 2026. 4

[15] S. Cho et al. SpatialClaw: Rethinking Action Interface for Agentic Spatial Reasoning. arXiv:2606.13673, 2026. 3, 4, 6

[16] B. Van Hoorick et al. Tracking through Containers and Occluders in the Wild. CVPR, 2023. 4

[17] S. Bai et al. Qwen3-VL Technical Report. arXiv:2511.21631, 2025. 3, 6

[18] C. Clark et al. Molmo2: Open Weights and Data for Vision-Language Models with Video Understanding and Grounding. arXiv:2601.10611, 2026. 3, 4

[19] C. Fu et al. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. CVPR, 2025. 4, 12, 16

[20] S. Bai et al. Qwen2.5-VL Technical Report. arXiv:2502.13923, 2025. 3

[21] J. Zhu et al. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. arXiv:2504.10479, 2025. 3

[22] B. Li et al. LLaVA-OneVision: Easy Visual Task Transfer. arXiv:2408.03326, 2024. 3

[23] B. Lin et al. Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. EMNLP, 2024.

[24] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. ACL, 2024. 3

[25] K. Li et al. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. CVPR, 2024. 4

[26] J. Xiao, X. Shang, A. Yao, and T.-S. Chua. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. CVPR, 2021. 4

[27] K. Mangalam, R. Akshulakov, and J. Malik. EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding. NeurIPS Datasets and Benchmarks, 2023. 4

[28] Y. Liu et al. TempCompass: Do Video LLMs Really Understand Videos? Findings of ACL, 2024. 4

[29] Z. Shangguan et al. TOMATO: Assessing Visual Temporal Reasoning Capabilities in Multimodal Foundation Models. ICLR, 2025. 4

[30] D. Saravanan et al. VELOCITI: Benchmarking Video-Language Compositional Reasoning with Strict Entailment. CVPR, 2025. 4

[31] J. Yang et al. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. CVPR, 2025. 4

[32] J. Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS, 2022. 4

[33] H. Fei et al. Video-of-Thought: Step-by-Step Video Reasoning from Perception to Cognition. ICML, 2024. 4

[34] X. Wang et al. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ICLR, 2023. 4, 7

[35] K. Feng et al. Video-R1: Reinforcing Video Reasoning in MLLMs. NeurIPS, 2025. 4

[36] R. Baillargeon, E. S. Spelke, and S. Wasserman. Object Permanence in Five-Month-Old Infants. Cognition, 20:191–208, 1985. 4

[37] E. S. Spelke. Principles of Object Perception. Cognitive Science, 14:29–56, 1990. 4

[38] R. Girdhar and D. Ramanan. CATER: A Diagnostic Dataset for Compositional Actions and TEmporal Reasoning. ICLR, 2020. 4

[39] K. Yi et al. CLEVRER: CoLlision Events for Video REpresentation and Reasoning. ICLR, 2020. 4

[40] A. Shamsian, O. Kleinfeld, A. Globerson, and G. Chechik. Learning Object Permanence from Video. ECCV, 2020. 4

[41] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon. Learning to Track with Object Permanence. ICCV, 2021. 4

[42] L. S. Piloto, A. Weinstein, P. Battaglia, and M. Botvinick. Intuitive Physics Learning in a Deep-Learning Model Inspired by Developmental Psychology. Nature Human Behaviour, 2022. 4

[43] A. Kirillov et al. Segment Anything. ICCV, 2023. 4

[44] N. Carion et al. SAM 3: Segment Anything with Concepts. ICLR, 2026. 4, 6

[45] H. K. Cheng and A. G. Schwing. XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model. ECCV, 2022. 4

[46] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing. Putting the Object Back into Video Object Segmentation. CVPR, 2024. 4

[47] N. Karaev et al. CoTracker: It is Better to Track Together. ECCV, 2024. 4

[48] Y. Zhang et al. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. ECCV, 2022. 4

[49] S. Liu et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. ECCV, 2024. 4

[50] A. Shtedritski, C. Rupprecht, and A. Vedaldi. What Does CLIP Know About a Red Circle? Visual Prompt Engineering for VLMs. ICCV, 2023. 4

[51] M. Cai et al. ViP-LLaVA: Making Large Multimodal Models Understand Arbitrary Visual Prompts. CVPR, 2024. 4

[52] H. Wang et al. Elysium: Exploring Object-level Perception in Videos through Semantic Integration Using MLLMs. ECCV, 2024. 4

[53] Y. Yuan et al. VideoRefer Suite: Advancing Spatial-Temporal Object Understanding with Video LLM. CVPR, 2025. 4

[54] H. Yuan et al. Sa2VA: Marrying SAM2 with LLaVA for Dense Grounded Understanding of Images and Videos. arXiv:2501.04001, 2025. 4

[55] T. Gupta and A. Kembhavi. Visual Programming: Compositional Visual Reasoning Without Training. CVPR, 2023. 4

[56] D. Surís, S. Menon, and C. Vondrick. ViperGPT: Visual Inference via Python Execution for Reasoning. ICCV, 2023. 4

[57] S. Yao et al. ReAct: Synergizing Reasoning and Acting in Language Models. ICLR, 2023. 4

[58] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy. VideoAgent: Long-form Video Understanding with Large Language Model as Agent. ECCV, 2024. 4

[59] C. Zhang et al. A Simple LLM Framework for Long-Range Video Question-Answering. EMNLP, 2024. 4

[60] Z. Yang et al. DoraemonGPT: Toward Understanding Dynamic Scenes with Large Language Models (Exemplified as a Video Agent). ICML, 2024. 4

[61] H. Lin et al. Depth Anything 3: Recovering the Visual Space from Any Views. ICLR, 2026. 4, 6

[62] Google DeepMind. Gemini 3 Pro Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf, 2025. 3

[63] Google DeepMind. Gemini 3.1 Pro Model Card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026. 3

[64] W. Hong et al. GLM-4.5V and GLM-4.1V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. arXiv:2507.01006, 2025. (Covers the GLM-4.6V series.) 3, 14

[65] ByteDance Seed. Seed2.0 Model Card: Towards Intelligence Frontier for Real-World Complexity. 2026. 3

[66] OpenAI. GPT-5.5 System Card. https://deploymentsafety.openai.com/gpt-5-5, 2026. 3

[67] OpenAI. GPT-6.1 Sol. https://developers.openai.com/api/docs/models/gpt-6.1-sol, 2026. 3, 6

[68] Gemma Team. Gemma 4 Technical Report. arXiv:2607.02770, 2026. 3, 6

[69] W. Wang et al. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. arXiv:2508.18265, 2025. 3, 6

[70] J. H. Cho et al. PerceptionLM: Open-Access Data and Models for Detailed Visual Understanding. arXiv:2504.13180, 2025. 14

[71] W. Kwon et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. SOSP, 2023. 6, 13

[72] P. Xie and A. Alanwar. Tracking Is Not Permanence: What Video World Models Keep of a Hidden Object. arXiv:2610.07355, 2026. 4

A. Algorithm and Implementation Details

Algorithm 1 gives the control flow of Track, Then Bind. The same program and the values of Table 6 run on every benchmark. The plan decides which branch a question takes; no branch is selected by the benchmark name.

Algorithm 1 Track, Then Bind. f: the frozen VLM; $\mathcal{T}$: the SAM2.1 video tracker. Read-outs use the benchmark's own prompt template and answer parser.

Input: video V, question q. **Output:** answer $\hat{a}$.
$P \leftarrow f(V \text{ at } 2\,\text{fps}, \textsc{Plan}(q))$; $P.\textit{final} \leftarrow f(\textsc{FinalLocation}(q))$ ▷ step 1: entities, roles, cue, viewpoint
if $P.\textit{cue} =$ visible, $P.\textit{final}$, and no carrier is named **then** add the target's category as the carriers
if P has no entities **then return** $\textsc{Parse}(f(V_S, q))$ ▷ direct answer
for each entity e **do** ▷ step 2
    boxes $\leftarrow f(\textsc{Ground})$ on 8 uniform frames, then on 8 frames in between if e was found on none
    $\tau_e \leftarrow \mathcal{T}(V \text{ at } 6\,\text{fps})$ from the earliest frame with the modal box count (one box: the largest)
if carrier tracks overlap, jump, or are not all tracked after 90% of the video **then**
    re-track carriers and target on every k-th frame, $k = \max(1, \text{round}(N_{\text{native}}/300))$ ▷ dense pass
$\hat{a} \leftarrow \varnothing$
if $P.\textit{final}, P.\textit{cue} \in \{\text{visible}, \text{contained}\}$, a target track and ≥ 2 carrier tracks exist **then**
    $\mathcal{M} \leftarrow$ evidence moments ▷ step 3
        visible: frames on which the target's box was found, merged within 3 s
        contained: the last frame before the target's own track vanishes inside the frame
    for each window $W \in \mathcal{M}$, or the whole video if $\mathcal{M} = \varnothing$ **do** ▷ step 4
        $k \leftarrow f(W \oplus \{\tau_c\} \text{ at } 4\,\text{fps},\ \leq 96 \text{ frames},\ q_{\text{b}}(P.\textit{cue}))$; keep the last valid k as $k^\star$
    if $k^\star$ exists **then** ▷ step 5a
        $\mathcal{F} \leftarrow$ the bound track and the left-to-right track order in the last frame where all carriers are tracked
        if $P.\textit{viewpoint} =$ person **then** $\mathcal{F} \leftarrow \mathcal{F} \cup$ viewpoint fact from $f(4 \text{ frames}, \textsc{Opposite})$
        $\hat{a} \leftarrow \textsc{Parse}(f(q, \mathcal{F}))$ ▷ text only
if $k^\star$ does not exist **then** $\hat{a} \leftarrow \textsc{Parse}(f(V_S,\ q \oplus \mathcal{L}(\{\tau_e\})))$ ▷ step 5b: timeline
if $\hat{a} = \varnothing$ **then** $\hat{a} \leftarrow \textsc{Parse}(f(V_S, q))$ ▷ unparsable read-out
return $\hat{a}$

Table 6. **Settings of the single configuration** (all benchmarks). p_{90}: the 90th percentile of a track's mask area over the video.

Step	Setting	Value
1 Plan	video rate for the plan call	2 fps
	entities per plan	at most 3
2 Perceive	grounding frames	8 uniform, plus 8 in between for entities found on none
	tracking resolution	960 px on the long side
	first tracking pass	6 fps, at most 900 frames
	dense pass triggers	intersection over union (IoU) of two instances > 0.05; centre shift > 0.25 box widths per sampled frame; carriers not all tracked after 90% of the video
	dense pass rate	target rate $r = \min(\text{native},\ 300/\text{duration})$; the pass runs if r exceeds $1.5\times$ the first-pass rate and reads every k-th native frame, $k = \max(1, \text{round}(\text{native}/r))$, so 300 is a target, not a cap (366 frames on VET-Bench)
3 Evidence, timeline	fully visible / partly hidden	mask area $\geq 0.6\,p_{90}$ / $\geq 0.12\,p_{90}$
	shortest segment	0.5 s (shorter segments are merged into a neighbour)
	left the frame	last box within 2% of the image border
	coarse motion	box-centre shift between the first and last third of the visible frames; "stays" below 0.1 of the frame
4 Bind	clip	4 fps, 3 s around a moment ($\delta = 1.5$ s) or the whole video, at most 96 frames
	several moments	the last one with a valid number decides
5 Read out	viewpoint question	4 uniform frames
	benchmark frames (timeline read-out)	PT and VET video at 8 fps; TOC-Bench 32 uniform frames
	unparsable answer	falls back to the direct answer
All	decoding	greedy

Table 7. **Cost per question on the test sets** (means). VLM calls by input: *video*, the plan call and a read-out on the benchmark's own frames; *frame*, grounding and the viewpoint question; *clip*, the bind call; *text*, the final-location question and the text-only read-out. Times are wall-clock seconds on A800 GPUs, one GPU per process at 8B and two at 32B, with other jobs sharing the node. The tracker time includes the dense pass.

		VLM calls					VLM tokens		Time (s)		
VLM	Set	video	frame	clip	text	all	in (k)	out	VLM	tracker	total
8B	cup	1.0	15.0	1.0	2.0	19.0	10.8	1058	40	69	110
	card	1.0	8.0	1.0	2.0	12.0	7.3	987	37	64	103
	PT	1.1	9.3	1.0	1.9	13.4	24.7	1289	50	71	126
	TOC	2.1	7.8	0.0	1.0	10.8	15.3	443	18	14	35
32B	cup	1.0	15.0	1.0	2.0	19.0	10.8	984	91	62	154
	card	1.0	8.0	1.0	2.0	12.0	7.3	216	22	64	87
	PT	1.2	10.7	0.9	1.9	14.7	27.0	929	98	56	159
	TOC	2.0	8.1	0.0	1.0	11.1	15.3	373	39	12	54

Table 7 breaks down the cost of Table 3. Grounding dominates the number of calls. On the VET cup game the ball is found on none of the first 8 frames, so a second round of 7 calls follows. Each call sees one frame, and the input stays at 7–27k tokens per question. Tracking takes 60–70 s per cup or card video because the dense pass runs on 88–100% of these videos. On TOC-Bench it runs on 1% of the questions, and tracking takes 12–14 s.

B. Benchmarks, Splits and Evaluation

Table 8. **Evaluation data.** Duration: mean video length in seconds. Dev: used to select the design and the thresholds. Test: run once with the frozen configuration.

Benchmark	Split	Use	Videos	Questions	Duration	Source
VET-Bench cup	test	test	50	50	12.4	VET-Bench [1] (synthetic)
VET-Bench card	dev	dev	56	56	12.1	rendered with the VET-Bench generator
VET-Bench card	test	test	50	50	12.1	VET-Bench [1] (synthetic)
Perception Test cup games	validation	dev	195	195	23.7	Perception Test [2]
Perception Test cup games	PT	test	107	107	15.6	filtered by the VET-Bench authors
three cups, $\geq$ 1 swap	PT-65	test	65	65	–	subset of PT
TOC-Bench	dev	dev	389	478	28.6	TOC-Bench [3], 20% of videos
TOC-Bench	test	test	1,561	1,845	28.9	TOC-Bench [3], the other 80%
Perception Test, all areas	validation	test	1,602	2,000	23.3	stratified sample, Table 15
Video-MME short + medium	–	test	600	1,800	298.7	Video-MME [19], Table 15

Development sets. The cup-game development set is the Perception Test validation question *"Where is the hidden object at the end of the game from the person's point of view?"* on 195 of its 198 videos; three videos were unavailable. The card-monte development set consists of 56 videos rendered with the VET-Bench generator. We changed only the encoder, from H.264 to VP9 and back to H.264 at a constant rate factor (CRF) of 18, because headless Chromium has no H.264 encoder. Each video carries the test-set question about the Queen of Hearts, with the answer taken from the generator's metadata. VET-Bench has no cup-game development set, so the cup game is tuned only through the real cup games and the card videos.

TOC-Bench split. We use the released answer subset of TOC-Bench, 2,323 questions over 1,951 videos. One development video could not be downloaded; its single question counts as wrong, and the development set has 389 usable videos. Its videos are assigned at random (seed 20261017) to development (20%) and test (80%), so that the questions of one video stay in one split. The test split has 1,068 four-way multiple-choice questions, 447 two-statement questions, 241 numerical questions and 89 ordering questions. TOC-Bench scores each format with its own parser, and an unparsable answer counts as wrong.

Disjointness. No video appears in two of the following sets: the cup-game development and test sets, the 2,000 general Perception Test questions, and the TOC-Bench development and test splits. The rendered card videos are new renders and share no file with the VET-Bench test videos.

Scoring and statistics. Perception Test and VET-Bench answers are parsed with the VET-Bench scorer, which reads the boxed letter. Every comparison in the paper is paired over the same questions. The confidence interval is the 2.5th and 97.5th percentile of 10,000 bootstrap resamples of the questions (seed 0). The p-value is the exact two-sided McNemar test on the questions that exactly one of the two methods answers correctly.

C. Baselines

Table 9. **Baseline settings.** All baselines receive the benchmark's prompt and are scored with its parser. Self-consistency and the larger input appear in Table 3.

Baseline	Engine	Input	Decoding
Qwen3-VL, direct	vLLM 0.15.1 [71]	PT and VET: video at 8 fps, default resizing of qwen-vl-utils; TOC: 32 uniform frames	greedy; 2,048 new tokens (TOC: the benchmark's budget per format)
Qwen3-VL, direct	Transformers 4.57.6	same	greedy
Self-consistency	vLLM; Transformers (Table 17)	same	k samples at temperature 0.7, top-p 0.8, top-k 20 (Qwen3-VL's recommended settings), majority vote; k equals the agent's mean VLM calls, at most 16
More input tokens	vLLM; Transformers (Table 17)	PT and VET: 16 fps, at most 288,000 pixels per frame, no resizing; TOC: 96 frames	greedy
Qwen3-VL-Thinking	vLLM (VET, PT); TOC-Bench runner (TOC)	as the direct answer	greedy; 16,384 new tokens
Qwen3-VL-235B-A22B	vLLM, 8 GPUs	video at 8 fps	greedy; 2,048 new tokens
Molmo2-8B, Molmo2-SGCoT	Transformers, the authors' code	at most 8 fps	greedy; 2,048 new tokens; SGCoT is the authors' low-rank adaptation (LoRA) weights with their prompt
Open-o3-Video-v2-8B	the official multi-turn evaluator	1 fps, 96 frames, crop tool, at most 5 turns	the evaluator's settings; it cannot express ordering questions, which count as wrong
SpatialClaw	the official code, Qwen3-VL-32B-Instruct (bf16) in vLLM	all frames at the native rate to the code, 32 key frames to the model	SpatialClaw's configuration for this model: temperature 0.7, top-p 0.8, top-k 20; at most 30 steps
VideoAgent	the official code; Qwen3-VL-32B-Instruct in vLLM as the agent LLM, Video-LLaVA-7B (4-bit) as the visual question answering tool	2 s segments; tracking at 15 fps	temperature 0; the official ReAct prompts
VideoINSTA	the official code; Qwen3-VL-32B-Instruct in vLLM as the LLM, CogAgent for captions and objects	1 fps; 4 event clips (CLIP ViT-L/14); UniVTG grounding	temperature 0; the official NExT-QA configuration

SpatialClaw. We run the official code with its prompts, tools and model configuration. The SAM3 and Depth Anything 3 tools are checked out at the commits the code pins. An adapter passes each benchmark prompt verbatim as the question and scores the returned answer with the benchmark's parser. Some sessions met a tool server that was temporarily unavailable. We reran every such question once, 12 on the cup games and 28 on TOC-Bench, and score the rerun. Table 10 summarises the runs.

Table 10. **SpatialClaw with Qwen3-VL-32B on the test sets.** Calls and input tokens are means per question. Mask propagation: share of questions whose code propagated a SAM3 mask through the video at least once without an error.

Set	n	Acc. (%)	VLM calls	Input tokens	Mask propagation (%)	Rerun after tool failure
cup	50	28.0	14.5	419k	26	0
card	50	36.0	9.0	229k	26	0
PT	107	26.2	17.7	508k	77	12
TOC	1845	35.7	10.2	300k	62	28

SpatialClaw propagates masks through the video in about a quarter of the VET-Bench questions, and the other sessions work on single frames. On the real cup games it propagates masks in 77% of the questions and still stays below chance. Fig. 4 shows one session.

VideoAgent. We run the official code (commit 42e1b70) with its prompts, tools and memory construction. The memory holds LaViLa captions of 2 s segments, ViCLIP segment features, RT-DETR-L tracks from ByteTrack at 15 fps, and CLIP and DINOv2 features for re-identification. Video-LLaVA-7B answers its visual questions. Four parts need a closed API or the network, and we replace them. Qwen3-VL-32B-Instruct at temperature 0 takes the place of GPT-4 as the agent LLM. The OpenAI text embedding is replaced by all-mpnet-base-v2, a sentence model that the official encoder already offers. LangChain's prompt download is replaced by the same template held locally, and an unused caption-metric import by a stub. The official multiple-choice prompt fixes five choices, so it states the actual number of choices. LangChain returns format errors in the ReAct output to the agent instead of stopping (handle_parsing_errors). TOC-Bench ordering questions go to the official free-form runner. A question takes 9.8 LLM calls and 9.0k LLM input tokens on average. Eleven of the 2,052 questions end with a malformed tool input and count as wrong.



step 1, last frame step 3, first frame step 4, last frame

1 `SAM3.segment_image_by_text(last frame, "playing card")`: three masks (left image).
2 `SAM3.segment_image_by_text(first frame, "Queen of Hearts")`: error, no usable mask.
3 `vlm.locate(first frame, "Queen of Hearts")` → (500, 496); a point prompt segments the queen (middle image).
4 `SAM3.segment_video_by_points(..., prompt_frame_idx=0)`: "Segmented mask area in final frame: 0 pixels" (right image).
5 `vlm.ask_with_thinking(last frame, ...)`: "Cannot determine from the images."
6 Code comment: "it is most likely still in the middle"; `ReturnAnswer("B")`. ✗ Gold: C.

Figure 4. **SpatialClaw on VET card monte** `card_001` (Qwen3-VL-32B, logged test run). The images are the session's own `show()` outputs. The queen's mask is empty once the card turns over, and the model answers from the last frame. Track, Then Bind answers the same video correctly (Fig. 3a).

VideoINSTA. We run the official code (publication branch) with the authors' NExT-QA configuration. It samples one frame per second and splits the video into four event clips by clustering CLIP ViT-L/14 features. CogAgent describes the actions and objects of each clip, and UniVTG locates the spans that match the question. The LLM summarises each clip, rates how well it answers the question, and merges the ratings into an answer. Three changes are needed. Qwen3-VL-32B-Instruct at temperature 0 takes the place of GPT-3.5 as the LLM. The authors release CogAgent captions and object lists only for their own benchmarks, so we compute them with their extraction configuration. The prompts state the actual number of choices instead of five, and ordering questions use the authors' free-form prompts. A question takes 14.8 LLM calls and 11.2k LLM input tokens on average, measured from the server logs. One answer names a choice that the question does not have and counts as wrong.

D. Other VLM Families

Open models on VET-Bench. Table 11 adds the open models whose VET-Bench predictions the VET-Bench authors released, rescored with the same parser. With our own runs, the open models cover eight families. All of them stay between 22% and 40% on both tasks.

Table 11. **Open Video-LMs on VET-Bench** (accuracy, %). Top, gray: cited, rescored with the VET-Bench parser from the predictions released by the VET-Bench authors [1]. Bottom: our runs (Table 1).

Model	cup	card
Qwen3.5-397B-A17B	38.0	32.0
Kimi-K2.5	28.0	32.0
ERNIE-4.5-VL-28B-A3B-Thinking	28.0	30.0
Perception-LM-8B [70]	40.0	34.0
GLM-4.6V-Flash [64]	34.0	28.0
Qwen3-VL-8B / 32B / 235B-A22B	28.0 / 32.0 / 22.0	28.0 / 38.0 / 38.0
InternVL3.5-8B	32.0	32.0
Gemma 4 31B	36.0	28.0
Molmo2-8B	30.0	40.0

Backends. The agent, its prompts and its configuration are the same for every VLM. Each backend adapts the input format and the box convention of its model.

- *InternVL3.5-8B* is served by vLLM. The frames of a clip or video are separate images of one tile each, each preceded by "Frame i (t s):", at most 128 per request. TOC-Bench frames follow the benchmark's own InternVL runner.
- *Gemma 4 31B* runs in Transformers with thinking disabled. It returns boxes as $[y_1, x_1, y_2, x_2]$, which the backend reorders. A request with several frames shares 4,480 visual tokens, and each frame receives the largest supported budget that fits (70 to 1,120). At most 64 frames are sent.
- *GPT-5.5* (`gpt-5.5-2026-04-23`) and *GPT-6.1* (`gpt-6.1-sol`) answer through the OpenAI Chat Completions API. Both are reasoning models and reason at the API's default effort, medium, in every call, direct or within the agent. A direct answer on a real cup game uses on average 2,740 reasoning tokens with GPT-5.5 and 125 with GPT-6.1, and no answer is cut off. Frames are images of at most 360×640 pixels, each preceded by its time. Boxes come back in pixels and are converted to the 0–1000 grid. TOC-Bench frames follow the benchmark's OpenAI runner. With GPT-6.1, one of the 107 agent runs on the real cup games ends in an API error after one rerun and counts as wrong.

On VET-Bench and the Perception Test, direct answers use the frame rate of the Qwen3-VL baselines, 8 fps, within each backend's frame limit. GPT-5.5 answers 47.6% of our 1,845 TOC-Bench test questions, close to the 47.1% that the TOC-Bench authors report on all 2,323 [3].

GPT-6.1 on unseen videos. The videos of VET-Bench and the Perception Test were public before GPT-6.1 was trained. Two tests argue against memorisation. On the 56 card-monte development videos, which we render with the VET-Bench generator and have not released, GPT-6.1 answers all 56. We also mirror the real cup games left to right and remap the answers. Four of the 107 videos have no mirrored answer among their options. On the other 103, GPT-6.1 answers 91.3% of the mirrored videos and 92.2% of the originals (-1.0, CI $[-4.9, 1.9]$). On the 56 videos whose answer changes under the mirror, only four answers keep the original option.

Viewpoint annotation. The real cup-game answers count positions from the left of the person who moves the objects, so the read-out must know whether that person faces the camera. One author labelled this for each of the 107 test videos from eight uniformly sampled frames, without the question or any answer. The person faces the camera in 73 videos and sits on the camera's side in 19; the 15 side-on videos are left unlabelled. With the annotated viewpoint in place of the agent's viewpoint fact, the read-out answers 56.1% (8B) and 57.0% (32B) of the real cup games, against 51.4% and 52.3%.

Errors on real cup games by VLM. Table 12 applies the error analysis of Sec. 5.4 to every VLM. A wrong answer is a *bind/track* error if the gold option disagrees with the bound track under the annotated viewpoint, a *viewpoint* error if the agent's viewpoint fact disagrees with the annotation, and a *read-out* error otherwise. For the 15 unlabelled videos, the gold option is checked under both viewpoints. With every VLM except InternVL3.5, most errors come from the bound track; with InternVL3.5, most errors are questions that the plan never binds.

Table 12. **Errors of the agent on the 107 real cup games** (counts). Errors are attributed with the annotated viewpoint (Appendix D). Unbound: the plan binds no target. Other: options that are not positions, or an API error. ‡First option letter, as in Table 1.

VLM	errors	bind/track	viewpoint	unbound	read-out	other
Qwen3-VL-8B	52	38	7	5	0	2
Qwen3-VL-32B	51	22	14	12	1	2
InternVL3.5-8B‡	69	11	4	53	1	0
Gemma 4 31B	55	38	2	14	0	1
GPT-5.5	41	34	4	1	0	2
GPT-6.1	31	26	3	0	0	2

InternVL3.5 with the plans of Qwen3-VL-8B. On card monte, the plan of InternVL3.5 lists the queen but no face-down cards, so the cards are never tracked. With the plans of Qwen3-VL-8B and all other steps by InternVL3.5, card monte rises from 32.0% to 98.0%. The cup game rises from 48.0% to 54.0%. On 40 of its 50 videos the stated facts imply the correct answer, and InternVL3.5 contradicts them in 17 of these 40. On TOC-Bench, the parser rejects 176 direct and 74 agent answers of InternVL3.5. On the 1,668 questions where both parse, the agent is 9.1 points above the direct answer (CI [6.5, 11.6]).

E. Additional Results

Table 13. **TOC-Bench test accuracy by dimension** (%). D: Qwen3-VL answering directly (vLLM); T: Track, Then Bind with the same model.

		8B		32B	
Dimension	n	D	T	D	T
event existence	170	34.7	**67.1**	42.9	**66.5**
reappear vs. disappear	106	41.5	**71.7**	41.5	**75.5**
relative spatial change	131	24.4	**51.9**	31.3	**52.7**
event count	241	13.3	**29.0**	20.3	**32.0**
event ordering	89	12.4	**27.0**	13.5	**32.6**
duration category	245	18.0	**34.7**	31.0	**36.7**
temporal location	473	20.1	**38.7**	41.4	**51.4**
cross-object order	155	40.6	**49.7**	49.7	**58.1**
conditional state	219	33.3	**40.2**	**49.8**	44.7
reappear identity	16	**62.5**	56.2	50.0	50.0

TOC-Bench by dimension. The agent improves the direct answer in nine of ten dimensions at 8B and in eight at 32B (Table 13). The largest gains, 21–34 points, are in event existence, reappearance versus disappearance, and relative spatial

change. A visibility timeline answers these questions almost directly. Conditional-state questions ask about one object at an event of another. On them the 32B agent drops from 49.8% to 44.7%.

Table 14. **One configuration versus per-benchmark configurations** (test sets): accuracy (%) and VLM calls per question.

		Accuracy				Calls	
VLM	Configuration	cup	card	PT	TOC	PT	TOC
8B	per benchmark	100	100	44.9	**43.4**	20	20.2
8B	single	100	100	**51.4**	43.0	**13.4**	**10.8**
32B	per benchmark	100	100	50.5	48.0	20	22.7
32B	single	100	100	**52.3**	**48.6**	**14.7**	**11.1**

One configuration for all benchmarks. Table 14 compares the agent with three variants of itself, each tuned on one benchmark. They fix the bind question, the grounding frames, the tracking rate and the read-out per benchmark. For example, the card variant grounds 4 frames from the first 30% of each video and always tracks at 30 fps. On all four test sets the single configuration lies within the CI of the tuned variants. The single configuration issues 10.8 instead of 20.2 VLM calls per TOC-Bench question at 8B, and 13.4 instead of 20 per cup game.

Table 15. **General video questions** (accuracy, %, Qwen3-VL-8B). Top: 2,000 Perception Test validation questions, stratified by area and type of reasoning. Bottom: Video-MME, short and medium videos. The direct answer is run in the same engine and the same pass as the agent.

Area	n	Direct	Track, Then Bind	Δ
physics	689	60.1	64.0	$+3.9$
semantics	634	86.4	86.9	$+0.5$
abstraction	593	63.4	64.2	$+0.8$
memory	84	58.3	66.7	$+8.3$
all	2,000	69.3	**71.5**	$+2.1$
Video-MME short	900	**78.7**	78.1	-0.6
Video-MME medium	900	**65.1**	63.7	-1.4
Video-MME all	1,800	**71.9**	70.9	-1.0

General video questions. Table 15 tests the agent on questions that are not about object identity. We sample 2,000 questions from the Perception Test validation split, stratified by area and type of reasoning, and exclude the development videos. The agent answers 71.5% of them and the direct answer 69.3% ($+2.1$, CI [0.5, 3.7], p=0.011). The gain is largest on physics and memory questions, and semantic questions are unchanged. On 1,800 short and medium Video-MME [19] questions, the agent answers 70.9% and the direct answer 71.9% (-1.0, CI [-2.1, 0.1], p=0.09). These questions rarely concern a single object. The plan binds none of them, and the appended timeline adds little information.

All paired comparisons. Table 16 lists the difference between our agent and every baseline that uses the same VLM, on the same questions. The agent is ahead in every row. Three intervals include zero, all on the real cup games: the 8B self-consistency vote in either engine and VideoINSTA.

Budget baselines in Transformers. Table 3 runs the call-matched and increased-input baselines in vLLM. Table 17 runs them in Hugging Face Transformers, the engine of our agent, with the same prompts, frames and sampling settings. The two engines lead to the same conclusions. On VET-Bench every baseline stays at 16–34%. On the real cup games the 8B vote reaches 43.0%, and the agent is 8.4 points higher (CI [-4.7, 21.5]). At 32B the vote reaches 37.4%, 15.0 points below the agent (CI [1.9, 29.0]). On TOC-Bench the vote changes the direct answer by at most one point, and the agent is 10.7–18.6 points above every baseline. Table 16 gives all paired intervals.

TOC-Bench by dimension at 32B. Table 18 adds the thinking variant, SpatialClaw, VideoAgent and VideoINSTA to the 32B columns of Table 13. The thinking variant moves no dimension by more than 10.1 points, and its overall accuracy is within 0.2 points of the direct answer. SpatialClaw gains 14.2 points on reappearance questions and loses 12.4 points on event counts. VideoINSTA and VideoAgent are the two best methods on event counts (39.4% and 35.3% against 32.0% for ours). VideoAgent is below the direct answer in seven of the ten dimensions, and VideoINSTA in six.

Perception Test cup games by subset. Table 19 splits the 107 real cup games by the number of cups and swaps given in the VET-Bench metadata. By the number of cups, the gain over the direct answer comes from the three-cup games. By the number of swaps, it is largest with one to three swaps. With four or more swaps the 8B agent gains 2.8 points and the 32B agent 11.4 points. The four four-cup games are answered once at 8B and never at 32B.

Table 16. **Ours minus each baseline with the same VLM** (test sets, percentage points). CI: paired bootstrap 95% interval; p: exact McNemar test.

VLM	Baseline	Set	Baseline (%)	Ours − baseline [95% CI]	p
8B	Direct answer (vLLM)	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	28.0	+72.0 [60.0, 84.0]	<0.001
		PT	33.6	+17.8 [3.7, 30.8]	0.02
		TOC	25.1	+17.9 [15.4, 20.4]	<0.001
	Direct answer (Transformers)	cup	22.0	+78.0 [66.0, 88.0]	<0.001
		card	34.0	+66.0 [52.0, 78.0]	<0.001
		PT	31.8	+19.6 [5.6, 32.7]	0.009
		TOC	25.2	+17.8 [15.3, 20.3]	<0.001
	Self-consistency, as many calls	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	30.0	+70.0 [56.0, 82.0]	<0.001
		PT	46.7	+4.7 [−9.3, 18.7]	0.59
		TOC	25.9	+17.2 [14.7, 19.6]	<0.001
	Direct answer, more input tokens	cup	24.0	+76.0 [64.0, 88.0]	<0.001
		card	30.0	+70.0 [56.0, 82.0]	<0.001
		PT	27.1	+24.3 [11.2, 36.4]	<0.001
		TOC	24.7	+18.4 [15.9, 20.8]	<0.001
	Self-consistency (Transformers)	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	30.0	+70.0 [56.0, 82.0]	<0.001
		PT	43.0	+8.4 [−4.7, 21.5]	0.25
		TOC	26.0	+17.0 [14.5, 19.5]	<0.001
	More input tokens (Transformers)	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	28.0	+72.0 [60.0, 84.0]	<0.001
		PT	29.0	+22.4 [9.3, 35.5]	0.002
		TOC	24.4	+18.6 [16.2, 21.0]	<0.001
	Thinking variant	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	24.0	+76.0 [64.0, 88.0]	<0.001
		PT	37.4	+14.0 [1.9, 26.2]	0.04
		TOC	30.0	+13.0 [10.3, 15.7]	<0.001
	Open-o3-Video-v2-8B	TOC	33.9	+9.2 [6.4, 11.8]	<0.001
32B	Direct answer (vLLM)	cup	32.0	+68.0 [56.0, 80.0]	<0.001
		card	38.0	+62.0 [48.0, 76.0]	<0.001
		PT	35.5	+16.8 [3.7, 29.9]	0.02
		TOC	37.1	+11.5 [9.0, 14.0]	<0.001
	Direct answer (Transformers)	cup	24.0	+76.0 [64.0, 86.0]	<0.001
		card	28.0	+72.0 [58.0, 84.0]	<0.001
		PT	27.1	+25.2 [13.1, 36.4]	<0.001
		TOC	36.9	+11.7 [9.2, 14.2]	<0.001
	Self-consistency, as many calls	cup	32.0	+68.0 [54.0, 80.0]	<0.001
		card	30.0	+70.0 [56.0, 82.0]	<0.001
		PT	37.4	+15.0 [2.8, 27.1]	0.03
		TOC	37.3	+11.3 [8.8, 13.8]	<0.001
	Direct answer, more input tokens	cup	30.0	+70.0 [56.0, 82.0]	<0.001
		card	24.0	+76.0 [64.0, 88.0]	<0.001
		PT	31.8	+20.6 [7.5, 33.6]	0.005
		TOC	36.7	+11.9 [9.4, 14.5]	<0.001
	Self-consistency (Transformers)	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	30.0	+70.0 [56.0, 82.0]	<0.001
		PT	37.4	+15.0 [1.9, 29.0]	0.05
		TOC	37.9	+10.7 [8.2, 13.2]	<0.001
	More input tokens (Transformers)	cup	34.0	+66.0 [52.0, 78.0]	<0.001
		card	16.0	+84.0 [74.0, 94.0]	<0.001
		PT	31.8	+20.6 [7.5, 33.6]	0.005
		TOC	36.7	+11.9 [9.4, 14.5]	<0.001
	Thinking variant	cup	24.0	+76.0 [64.0, 88.0]	<0.001
		card	22.0	+78.0 [66.0, 88.0]	<0.001
		PT	33.6	+18.7 [6.5, 30.8]	0.006
		TOC	36.9	+11.7 [9.2, 14.3]	<0.001
	SpatialClaw	cup	28.0	+72.0 [60.0, 84.0]	<0.001
		card	36.0	+64.0 [50.0, 76.0]	<0.001
		PT	26.2	+26.2 [14.0, 38.3]	<0.001
		TOC	35.7	+12.9 [10.1, 15.7]	<0.001
	VideoAgent	cup	40.0	+60.0 [46.0, 74.0]	<0.001
		card	28.0	+72.0 [60.0, 84.0]	<0.001
		PT	37.4	+15.0 [1.9, 28.0]	0.03
		TOC	33.0	+15.7 [12.8, 18.6]	<0.001
	VideoINSTA	cup	42.0	+58.0 [44.0, 72.0]	<0.001
		card	28.0	+72.0 [60.0, 84.0]	<0.001
		PT	42.1	+10.3 [−1.9, 23.4]	0.15
		TOC	34.5	+14.1 [11.3, 17.0]	<0.001

Table 17. **Call-matched and increased-input baselines in Transformers** (test sets, %). Settings as in Table 3. Self-consistency votes over 16, 12 and 13 (8B) or 15 (32B) samples on cup, card and PT, and over 11 samples on TOC.

VLM	Baseline (Transformers)	cup	card	PT	TOC
8B	direct answer	22.0	34.0	31.8	25.2
	self-consistency, as many calls	28.0	30.0	43.0	26.0
	direct answer, more input tokens	28.0	28.0	29.0	24.4
	Track, Then Bind	100	100	51.4	43.0
32B	direct answer	24.0	28.0	27.1	36.9
	self-consistency, as many calls	28.0	30.0	37.4	37.9
	direct answer, more input tokens	34.0	16.0	31.8	36.7
	Track, Then Bind	100	100	52.3	48.6

Table 18. **TOC-Bench test accuracy by dimension, Qwen3-VL-32B** (%). Direct: answering directly in vLLM. Ours: Track, Then Bind. Best per row in bold.

Dimension	n	Direct	Thinking	SpatialClaw	VideoAgent	VideoINSTA	Ours
event existence	170	42.9	42.9	45.9	38.2	40.6	**66.5**
reappear vs. disappear	106	41.5	43.4	55.7	34.0	33.0	**75.5**
relative spatial change	131	31.3	33.6	29.8	26.7	29.0	**52.7**
event count	241	20.3	19.9	7.9	35.3	**39.4**	32.0
event ordering	89	13.5	23.6	16.9	13.5	14.6	**32.6**
duration category	245	31.0	24.9	29.0	26.1	31.4	**36.7**
temporal location	473	41.4	40.2	40.2	31.3	31.1	**51.4**
cross-object order	155	49.7	55.5	47.7	47.1	47.7	**58.1**
conditional state	219	**49.8**	47.0	48.4	37.0	36.1	44.7
reappear identity	16	50.0	**56.2**	50.0	**56.2**	**56.2**	50.0
all	1,845	37.1	36.9	35.7	33.0	34.5	**48.6**

Table 19. **Perception Test cup games by subset** (accuracy, %). D: direct answer in vLLM.

Subset	n	D 8B	Ours 8B	D 32B	SpatialClaw	Ours 32B
2 cups	13	53.8	53.8	46.2	23.1	53.8
3 cups	90	30.0	52.2	33.3	27.8	54.4
4 cups	4	50.0	25.0	50.0	0.0	0.0
no swap	27	40.7	55.6	37.0	18.5	51.9
1–3 swaps	45	22.2	53.3	28.9	28.9	51.1
$\geq$4 swaps	35	42.9	45.7	42.9	28.6	54.3
PT-65	65	27.7	52.3	33.8	30.8	55.4

F. Design Choices on Development Data

Table 20. **Design choices on development data** (accuracy, %). Rows 1–3 hold the tracks and the numbered overlays fixed and change only the question to the VLM.

Variant	Set	8B	32B
Direct answer	PT	33.3	40.0
Tracks, asked "where at the end?"	PT	40.0	44.1
Tracks, asked "under which, right after hidden?"	PT	**48.7**	**52.3**
Card tracking at 10 fps / 30 fps	card	37.5 / **100**	–
Viewpoint: "faces the camera" / "opposite the camera"	PT	48.2 / **57.9**	–
Tracks, benchmark question, no binding	PT / card	40.0 / 62.5	42.1 / **100**
Timeline: visibility only / + events and motion	TOC	37.4 / **42.1**	–
Grounding frames: 16 / 8	TOC	40.2 / **42.7**	–

Table 20 lists the comparisons made while the design was chosen. Their p-values are exploratory. The moment of the question matters most. With the same tracks and overlays, asking where the target is at the end of the video gives 40.0% at 8B. Asking which cup it is under right after it is hidden gives 48.7% (+8.7, CI [0.5, 16.9], p=0.046). At 32B the two questions give 44.1% and 52.3% (+8.2, CI [1.5, 14.9], p=0.026). The overlay alone adds 6.7 and 4.1 points over the direct answer, and the moment of the question adds the rest. Drawing the tracks and asking the benchmark question, without any bind question,

gives 42.1% at 32B. This is 17.9 points below the single configuration on the same videos (60.0%, CI [9.2, 27.2]). On card monte the same variant reaches 100% at 32B, as on the test set. The bind answer itself comes from the VLM. The target's own track hands its identity to a cup in only 56 of 195 videos, and that hand-off is right in 48% of them.

Tracking rate and grounding decide whether the tracks are usable. On card monte, tracking at 10 fps instead of 30 fps lowers accuracy from 56 to 21 of 56 videos, close to the direct answer (16). Identity breaks where identical cards cross quickly. The dense pass of step 2 reproduces the 30 fps result without a fixed rate. Grounding on 16 frames instead of 8 lowers TOC-Bench accuracy by 2.5 points and card-monte accuracy by 3 videos, and it nearly doubles the VLM calls.

Two choices concern the read-out. Asking whether the person is opposite the camera, on four frames, gives 57.9% on the development cup games. Asking whether the person faces the camera, on the last frame, gives 48.2%. This wording was selected among six on the same videos, and on the test videos the single configuration is 6.5 (8B) and 1.9 points (32B) above the tuned variants. On TOC-Bench, visibility segments alone reach 37.4% of the development questions. Adding the event list and the direction of motion raises this to 42.1% (+4.6, CI [1.5, 7.7]).

G. Additional Qualitative Examples

Figures 5 and 6 show six more test items from the logged Qwen3-VL-8B run. The images are the agent's own tracks on the frames it read. Rows (c)–(f) were re-run with full logging, and every re-run reproduces the logged plan, evidence and answer. The direct answers are those of Table 1.

Rows (a) and (b) are the two branches of a located moment. In (a) the bind clip is synthetic and short. In (b) the hidden object is a crumpled paper in a real video, and the viewpoint fact mirrors the camera order. Row (c) is one of the five real cup games that are never bound (Sec. 5.4). The VLM grounds the pen holder as the only cup, so the agent never sees two carrier tracks.

Rows (d) and (e) are typical TOC-Bench gains. The timeline states when the object disappears and whether it vanished at the border or inside the frame, which the direct answer misses on its 32 sampled frames. Row (f) is a conditional-state question about two objects. The plan names only one of them, so the timeline cannot place the other object's event. This failure fits the conditional-state results in Table 13.

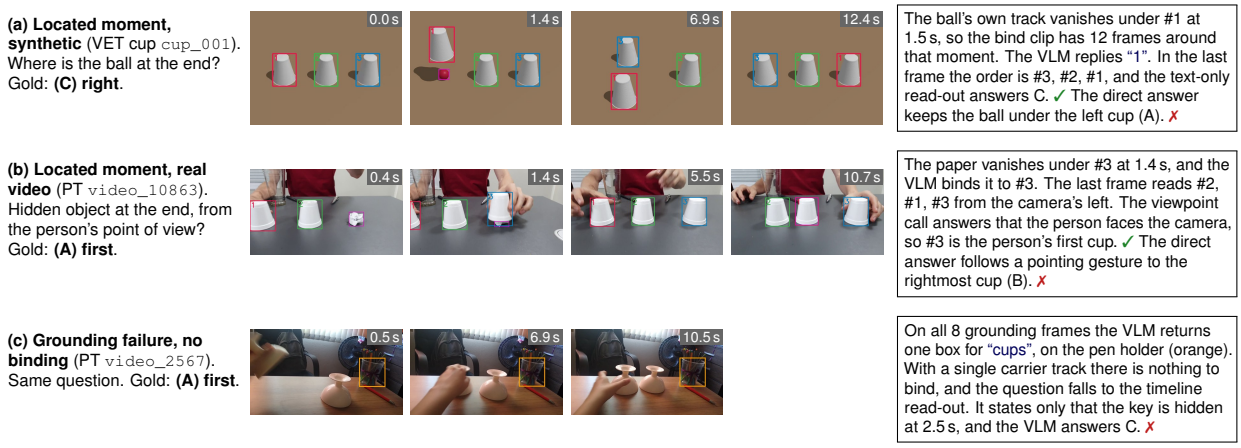


Figure 5. **Identity questions** (test videos, Qwen3-VL-8B). Numbered boxes are the agent's carrier tracks; magenta is the target's own track. (a, b) Located evidence moments. (c) A grounding failure that leaves one carrier track, so the question is not bound.

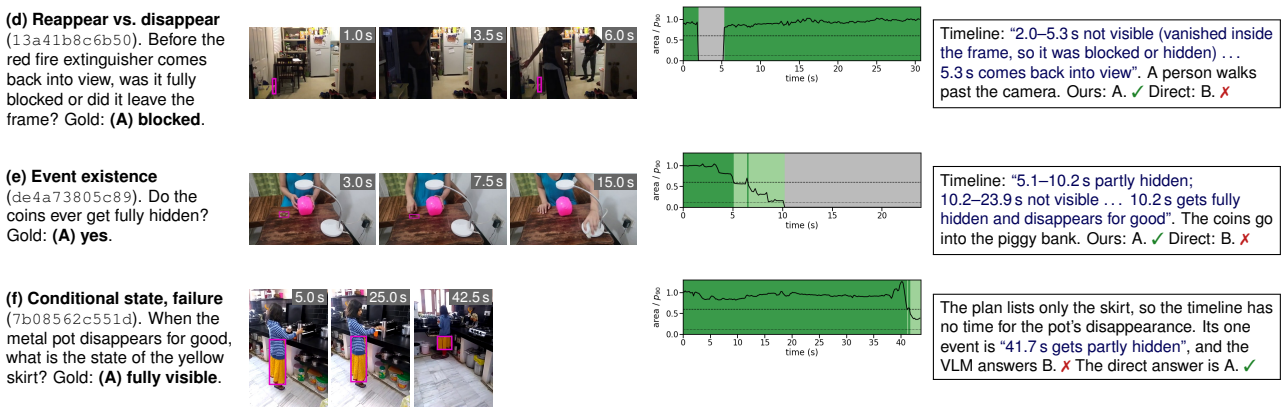


Figure 6. **TOC-Bench timeline questions** (test videos, Qwen3-VL-8B). Magenta: the subject's track box. Bars: mask area over its 90th-percentile area, coloured by the visibility state (dark green: fully visible; light green: partly hidden; grey: not visible); dashed and dotted lines mark the thresholds 0.6 and 0.12.

H. Worked Examples

Figures 7–9 follow three test items through the agent, next to the direct answer of the same VLM (Qwen3-VL-8B). Each panel lists what the VLM sees, the prompt it receives, its reply and the number of calls. Each run reproduces the logged test output of the single configuration. The prompts are listed verbatim in Table 21.

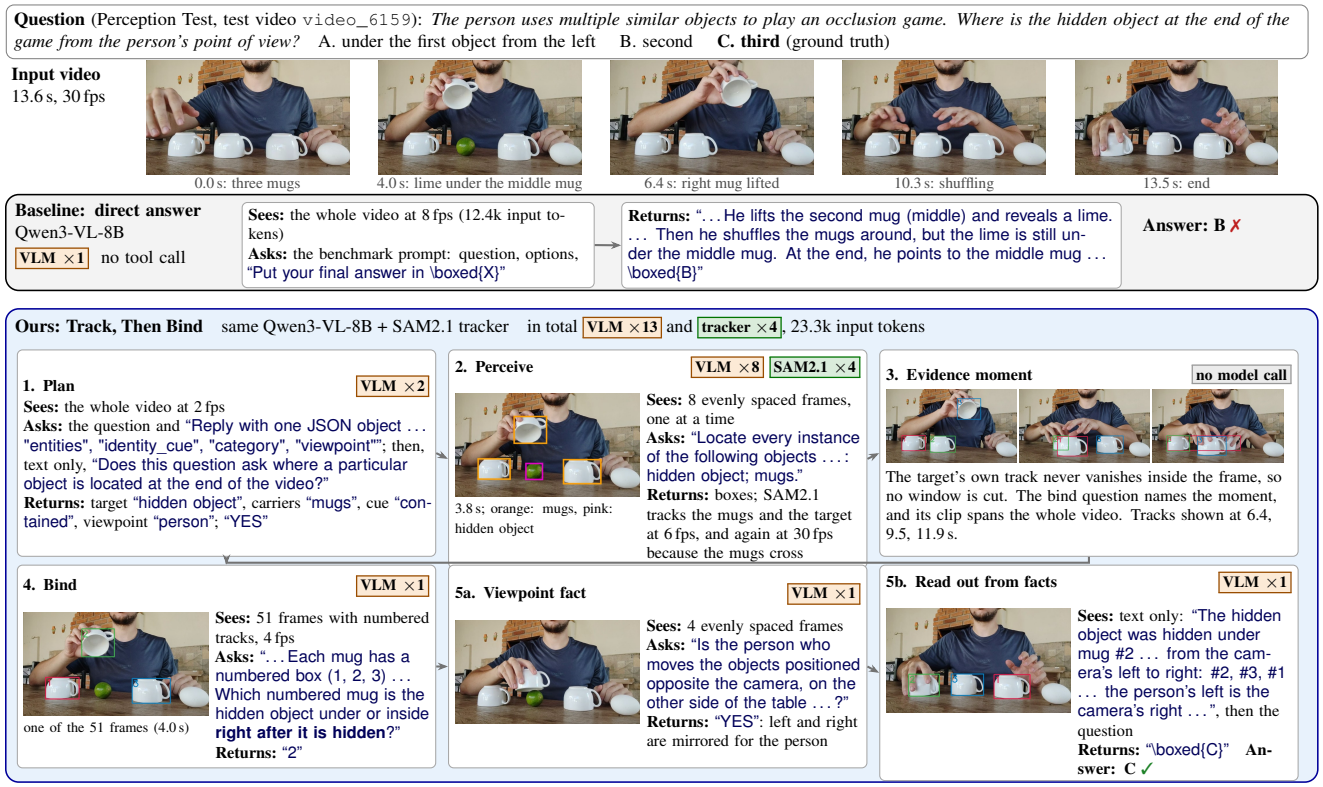


Figure 7. **Hidden target** (Perception Test cup game video_6159, test set). *Grey:* Qwen3-VL-8B answering directly from the video, one VLM call. *Blue:* Track, Then Bind with the same VLM and the SAM2.1 tracker. Badges give the calls of each step: orange for the VLM, green for the tracker sessions, grey for plain code. Prompts are quoted from the run and shortened with "...". This item takes the most common branch on real cup games: the target's own track gives no moment, and the bind question is asked over the whole numbered clip.

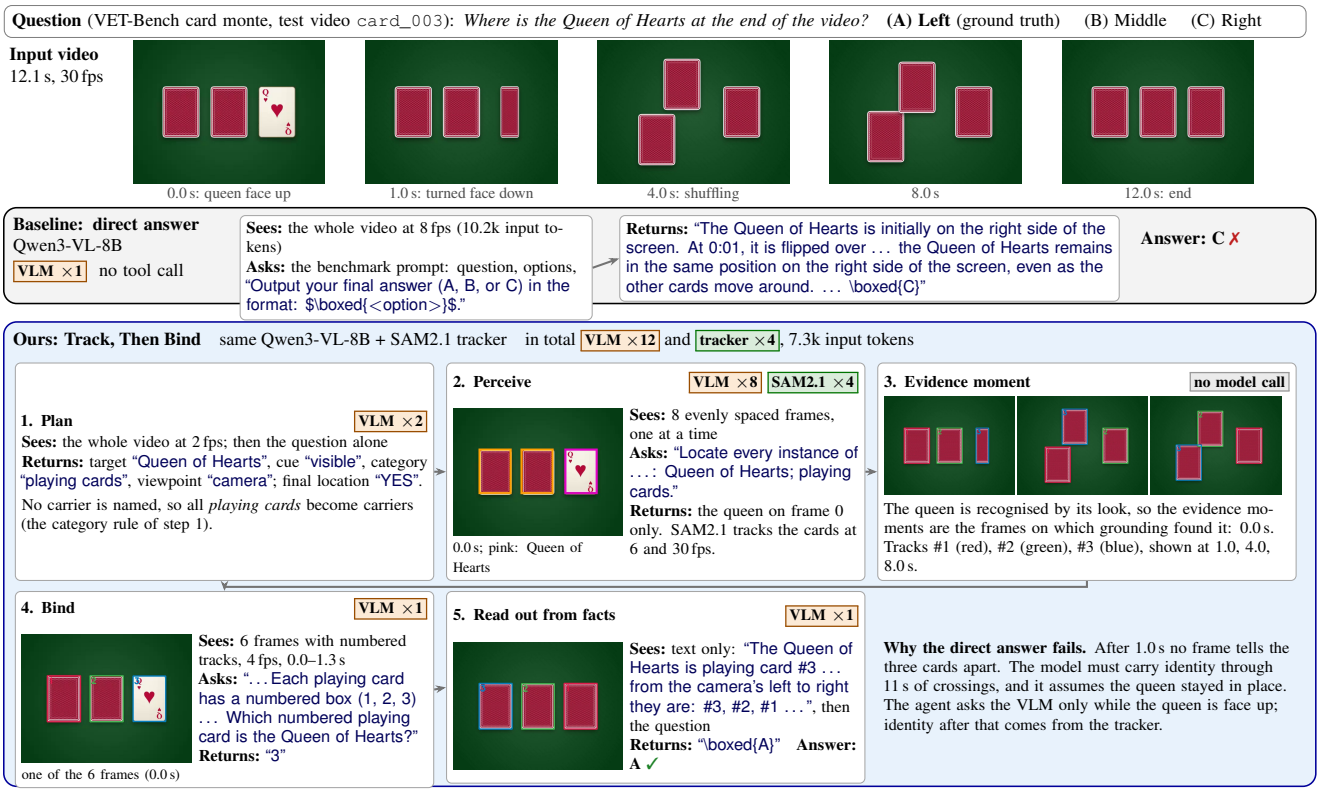


Figure 8. **Target recognised by its look** (VET-Bench card monte `card_003`, test set). The VLM is asked once, on a 1.3 s clip while the queen is face up; the 30 fps tracks carry the identity through the shuffle. Track #1 (red) is hard to see against the red card backs; this is the overlay the VLM receives.

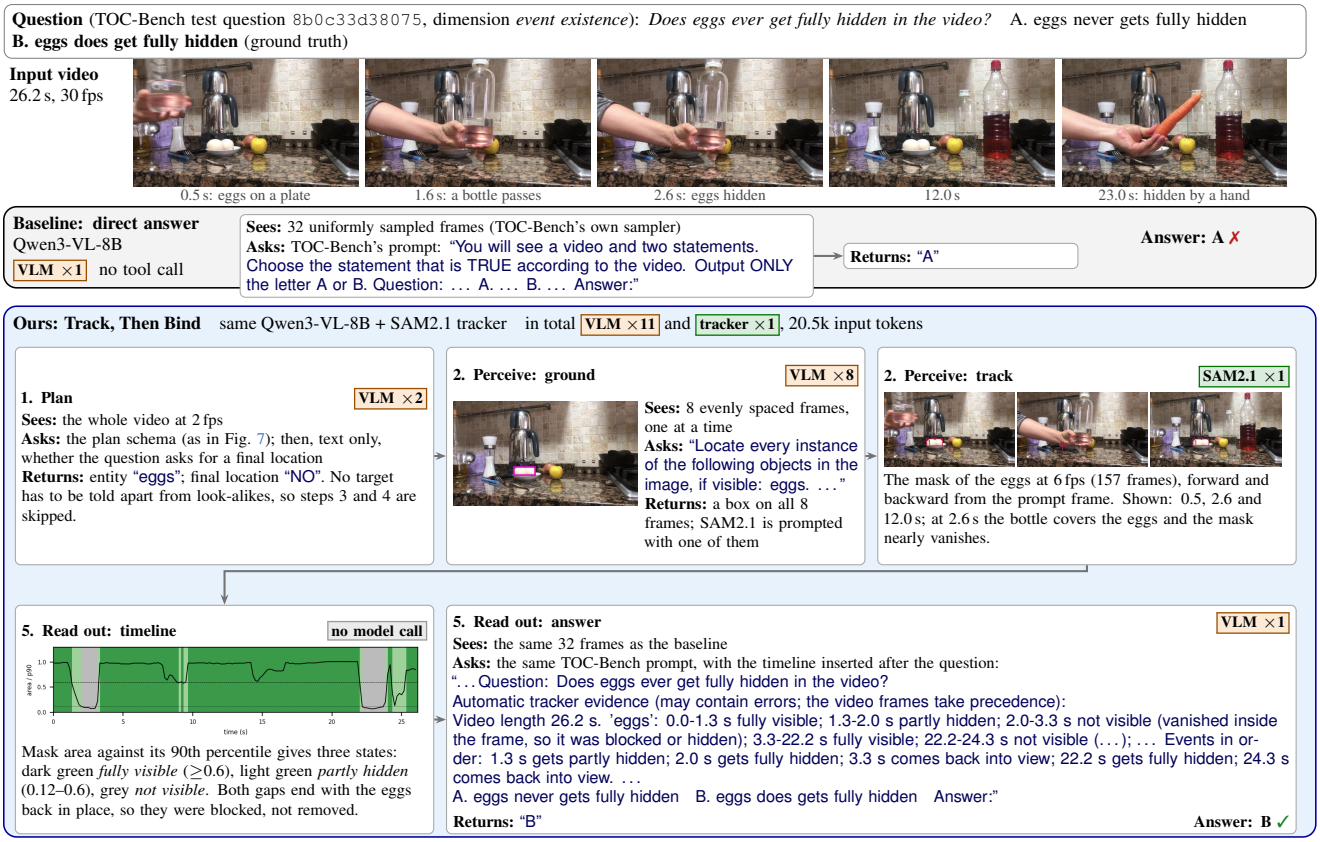


Figure 9. **Timeline question** (TOC-Bench 8b0c33d38075, test set). The plan finds no target to tell apart, so the agent skips binding and answers with a timeline. Both runs see the same 32 frames and the same template; the only difference is the evidence text inserted after the question. It is built from the tracked mask of the eggs, without a model call.

I. Prompts

Table 21. **Unified agent's prompts, verbatim** (one set for all benchmarks). Slots in {braces} are filled from the plan, the tracks or the benchmark item.

Step	Input	Prompt
Plan (1)	whole video, 2 fps	You plan how to answer a question about a video. Question: {question} Reply with one JSON object and nothing else: {"entities": [{"name": . . . , "role": "target" | "carrier" | "subject"}], "identity_cue": "visible" | "contained" | "none", "category": . . . , "asks_final_location": true | false, "viewpoint": "camera" | "person"} followed by one rule per field (target; carriers that move, hide or cover the target or look the same as it; subjects; recognised by its look vs. hidden in a carrier; final location; whose left and right)
Final location (1)	text only	Question about a video: {question} Does this question ask where a particular object is located at the end of the video (for example, which container it is in, or whether it is on the left, in the middle or on the right)? Reply YES or NO.
Ground (8, +8 if missing)	one frame	Locate every instance of the following objects in the image, if visible: {entity 1}; {entity 2}; . . . Output a JSON list of {"bbox_2d": [x1, y1, x2, y2], "label": "<one of the names above>"}; output [] if none is visible.
Bind, by its look (1 per moment)	numbered tracks, 4 fps	These frames are from a video. Each {carrier} has a numbered box (1, 2, 3) drawn on it; a number always stays with the same physical object. Which numbered {carrier} is the {target}? Reply with a single number, or NONE if the {target} cannot be seen.
Bind, hidden (1 per moment)	numbered tracks, 4 fps	. . . Which numbered {carrier} is the {target} under or inside right after it is hidden? Reply with a single number, or NONE if you cannot tell.
Viewpoint (1, if asked)	4 uniform frames	Is the person who moves the objects positioned opposite the camera, on the other side of the table (so the camera looks at them from the front)? Reply YES or NO.
Read out, bound (1)	text only	Facts established by tracking every object through the whole video (identity was fixed when the target could be seen, and kept by the tracker afterwards): - The {target} was hidden under {carrier} #{k}. The tracker followed every {carrier} through the rest of the video. In the last frame where all {n} are tracked ({t} s), from the camera's left to right they are: #. . . . So the {target} is under the {ordinal} {carrier} from the camera's left. [- viewpoint fact - person-view position] Answer the question using these facts. then the benchmark prompt, then The facts above are reliable. Reply with only the final answer in \boxed{X}.
Read out, not bound (1)	the benchmark's frames	the benchmark's prompt with Automatic tracker evidence (may contain errors; the video frames take precedence): {video length; one timeline line per subject} after the question

Table 22. **Prompts of the per-benchmark configurations, verbatim.** Slots in {braces} are filled by the agent from earlier steps or from the benchmark item. These configurations are the comparison of Table 14.

Step	Input	Prompt
Direct baselines (one call; the benchmark's own prompt)		
Perception Test	video, 8 fps	{question}\n\nA. {option}\nB. {option}\nC. {option}\n\nPlease select the correct answer from the options above. Put your final answer in \boxed{X} where X is the letter of your choice.
VET-Bench card monte	video, 8 fps	Where is the {target} at the end of the video?\n(A) Left\n(B) Middle\n(C) Right\nOutput your final answer (A, B, or C) in the format: $\boxed{<option>}$.
TOC-Bench (two-statement format; other formats use their own templates)	32 uniform frames	System: You are a video understanding model. Follow output format exactly. User: You will see a video and two statements. Choose the statement that is TRUE according to the video. Output ONLY the letter A or B.\n\nQuestion: {question}\nA. {statement A}\nB. {statement B}\nAnswer:
SpatialClaw (its question field)	its own frames	the benchmark prompt above, verbatim; for TOC-Bench the system and user messages joined by a blank line
Containment hand-off (Perception Test cup games; 20 calls per item)		
Parse (1)	video, 2 fps	In this video a person plays a shell game: a small object is hidden under one of several similar containers, then the containers are moved around. Reply with two short noun phrases separated by '|': first, the containers (plural, no number); second, the small object that gets hidden and disappears from view under a container. Ignore other objects that stay visible. No other text.
Ground carriers (8)	one frame	Locate every {containers} in the image. Output a JSON list of {"bbox_2d": [x1, y1, x2, y2], "label": "{containers}"}.
Ground target (8)	one frame	Locate the {target} in the image, if it is visible. Output a JSON list of {"bbox_2d": [x1, y1, x2, y2], "label": "{target}"}; output [] if it is not visible.
Viewpoint (1)	last frame	Is the person playing this game sitting or standing across the table, facing the camera? Reply YES if the person faces the camera, or NO if the camera shows the scene from the person's own viewpoint (for example, held by or mounted on the person).
Bind (1)	numbered-track frames, 2 fps	In this video a person hides {target} under one of several {containers} and then moves them around. Each container has a numbered box (1, 2, 3) drawn on it in every frame; a number always stays with the same physical container. Which numbered container is {target} under right after it is hidden? Reply with a single number.
Bind, ablation (1)	same	identical, ending . . . under at the END of the video? Reply with a single number.
Visible identity (VET-Bench card monte; 5 calls per item)		
Ground (4)	one frame	Locate every playing card in the image. Output a JSON list of {"bbox_2d": [x1, y1, x2, y2], "label": "card"}.
Bind (1)	numbered-track frames, 4 fps, first 30%	These frames show the start of a card game. Each card has a numbered box (1, 2, 3) drawn on it; a number always stays with the same physical card. Which numbered card shows the {target} face up? Reply with a single number.
Subject timeline (TOC-Bench; 1 + 16 per subject + 1 calls per item)		
Parse subjects (1)	text only	List the physical objects or people that this video question is about, as short noun phrases exactly as written, one per line, at most 3. No other text.\n\n{question}\n{statements or events}
Ground (16 per subject)	one frame	Locate the {subject} in the image, if it is visible. Output a JSON list of {"bbox_2d": [x1, y1, x2, y2], "label": "{subject}"}; output [] if not visible.
Answer (1)	32 uniform frames	the TOC-Bench template above, with {question} replaced by {question}\n\nAutomatic tracker evidence (may contain errors; the video frames take precedence):\n{evidence lines}