BENCHMARKING VISUAL STATE TRACKING IN MULTIMODAL VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding a video requires more than recognizing isolated moments, as humans continuously track entities, states, and events over time. This capacity for *visual state tracking* is fundamental to video understanding, yet remains underexplored in current evaluations of Multimodal Large Language Models (MLLMs). We introduce **V**isual **STA**te **T**racking benchmark (VSTAT), a video-based benchmark designed to diagnose visual state tracking in MLLMs. VSTAT consists of 834 clips drawn from both synthetic and real-world videos, paired with 1,500 questions that cannot be answered from any single frame or short segment, requiring continuous perception and integration of events across the entire video stream. Despite their strong performance on existing video benchmarks, we find that state-of-the-art MLLMs perform far below humans and only modestly above answer-prior baselines. To analyze this gap, we compare MLLMs' thinking traces with the underlying video stream to understand *why* and *when* MLLMs fail on VSTAT. Through extensive analyzes, we find that MLLMs reason and track correctly in text, but still fail at visually perceiving the events they need to track.

1 INTRODUCTION

Videos are not just a discrete sequence of RGB pixels; they are records of continuous dynamics and processes in the visual world. When we watch a video, we do not simply perceive each frame independently, but also understand and analyze the underlying dynamics by keeping track of essential information. For instance, when watching a basketball game, we naturally keep track of the score and who attempted each shot by making sense of complex visual procedures. This capacity for *visual state tracking* is fundamental to how humans learn from and reason about visual demonstrations.

Recent Multimodal Large Language Models (MLLMs) have progressed remarkably in video understanding, demonstrating strong capabilities in semantic understanding and action recognition (Bai et al., 2025a; Yang et al., 2026; Wang et al., 2025; Google DeepMind, 2026). However, it remains unclear whether current MLLMs can understand continuous dynamics and track evolving states throughout a procedure presented in the video (Wang et al., 2024c), which is essential for real-world applications such as robotics (Black et al., 2024). This gap stems from the fact that existing video understanding benchmarks are mostly not explicitly designed for evaluating this capability. In many cases, the answer can be inferred by relying on a small subset of keyframes, salient moments, or visible end states, without continuously tracking how the underlying state evolves over time. As a result, strong performance on these benchmarks does not necessarily indicate an ability to track necessary information in the video. While a few recent works have attempted to address this gap (Liu & Lee, 2026; Wang et al., 2024c), their evaluations remain limited to a single synthetic task (*e.g.*, shell game) and do not cover diverse, real-world scenarios.

We introduce **V**isual **STA**te **T**racking benchmark (VSTAT), a video-based benchmark designed to diagnose visual state tracking in MLLMs.[1] VSTAT adopts a standard question-answering format, where the model receives a video stream and a question as input and outputs an answer. VSTAT consists of 834 video clips paired with 1,500 questions drawn from synthetic, self-recorded, and real-world videos in the wild that contain procedural processes. Each task is constructed so that the

[1]We note that our formulation of visual state tracking extends beyond object tracking in pixel space: our benchmark also covers tracking of underlying latent state representations in the video stream.

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.

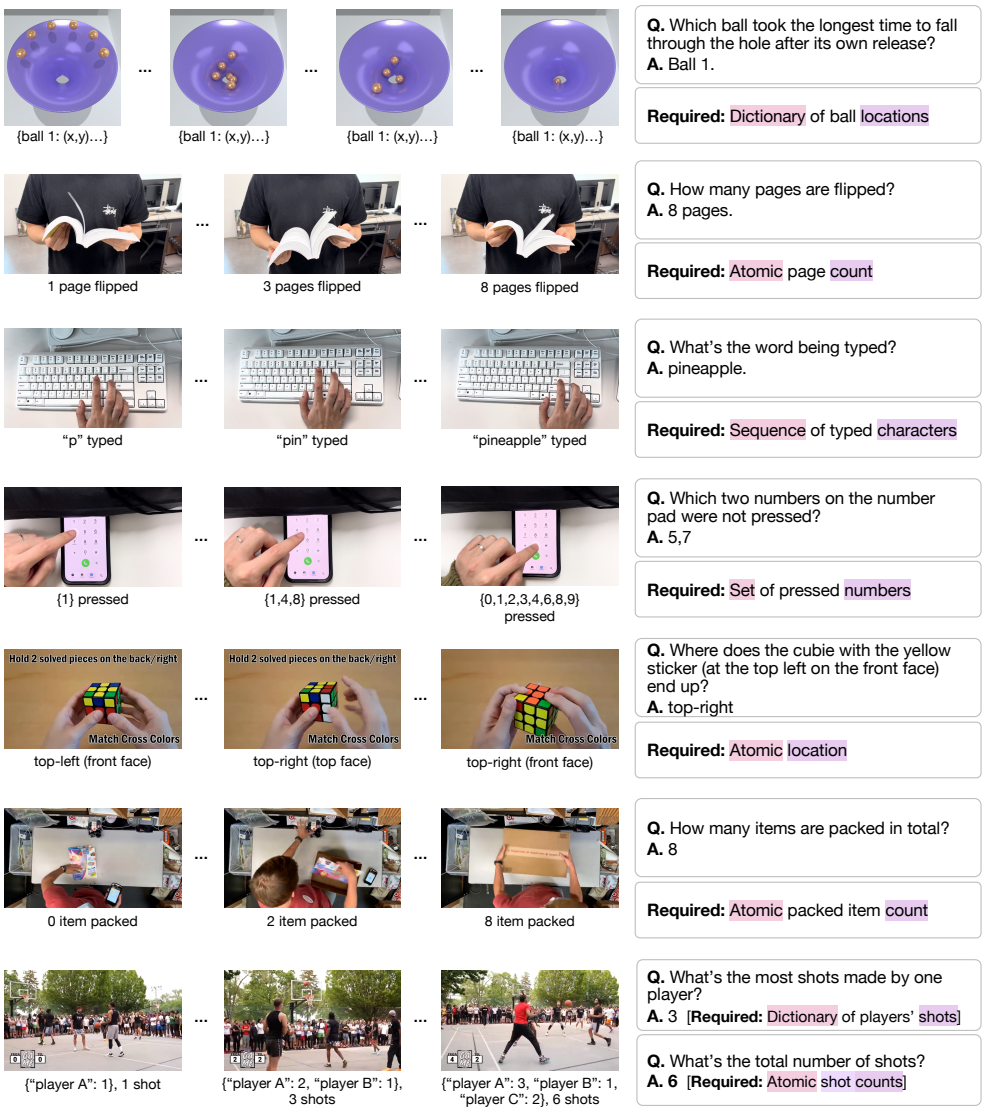


Figure 1: **Task examples in VSTAT.** All questions require visual state tracking to answer. For illustration, we simplified the questions and subsampled video frames. Each example requires tracking different states, which are combinations of **structure** and **element type**.

answer cannot be read off any single keyframe or a few salient moments: critical events may be hidden, visually similar to each other, or distributed across multiple entities and moments. Thus, models must continuously perceive and integrate events throughout the entire video stream. The tasks in VSTAT vary in the complexity of state to be tracked, and exhibit various perceptual challenges in extracting state from video; for instance, in the Rubik’s cube task, the model must track a specific cubie even when it is occluded in some frames (see Figure 1 for more examples). Surprisingly, while these questions can be easily answered by humans, state-of-the-art MLLMs perform far below humans and only modestly above answer-prior baselines.

To understand this gap, we investigate behaviors of MLLMs through several analyses. Firstly, we study *why* MLLMs struggle to solve VSTAT by conducting controlled experiments on synthetic tasks generated in Blender environments. We first test whether frame subsampling, which MLLMs usually apply to videos and may cause them to miss brief events, is the bottleneck. To do so, we compare performance on the original videos against temporally stretched versions, where each event spans more frames to ensure that subsampling does not introduce ambiguity. However, we observe only marginal improvement, suggesting that this is not the case.

Table 1: **Comparison with existing video understanding benchmarks.** We compare VSTAT against existing benchmarks in terms of their coverage of state-tracking tasks. ◐ denotes that the benchmark contains some instances satisfying the categories but only as a small fraction. CP-Bench and VET-Bench focus on 1–2 simple synthetic tasks (counting identical cubes and shell game, respectively). Dataset sources: Scripted (C), Real (R) and Synthetic (Y).

Benchmark	Source	#Clips	#QAs	State Tracking	Real-world	Diverse
VideoMME-v2 (Fu et al., 2026)	C,R	800	3,200	◐	✓	✓
VideoReasonBench (Liu et al., 2026)	C,Y	240	1,440	✓	◐	✗
CP-Bench (Wang et al., 2024c)	Y	101	101	✓	✗	✗
VET-Bench (Liu & Lee, 2026)	Y	100	100	✓	✗	✗
VSTAT (ours)	C,R,Y	834	1,500	✓	✓	✓

Then, to further investigate whether the failures of MLLMs stem from insufficient reasoning or limited perception capabilities, we use several simple tasks in VSTAT whose underlying events can be manually transcribed. We compare model performance under two conditions: the original video input and a text transcription that explicitly describes each frame and event (see Figure 2a). While MLLMs struggle in the video condition, they solve the same tasks almost perfectly when given text transcriptions. This contrast suggests that the fundamental bottleneck lies in visual perception of the video stream, rather than in flawed reasoning. We stress, however, that such transcription is infeasible for most tasks in VSTAT due to their more complex settings and the substantially larger amount of information required to answer the questions (see Figure 1). Ultimately, tackling the state-tracking challenge requires much stronger perceptual capabilities from MLLMs.

Secondly, we diagnose *when* MLLMs fail during visual state tracking by analyzing mismatches between their textual thinking traces and the input video stream in failure cases. From their traces, we identify three major failure modes: event recognition, entity association (*i.e.*, linking the same entity across frames), and state update (*i.e.*, updating the tracked state after each perceived event). For instance, when watching a shell game video, the model may misidentify which cups were swapped, lose track of the cup hiding a target item, or fail to update its location even after correct identification.

We highlight the main contributions of this paper below:

- We introduce VSTAT, a video-based benchmark for evaluating the *visual state tracking* capability of MLLMs, covering both synthetic and real-world videos paired with questions.
- We show that state-of-the-art MLLMs perform far below human performance and only modestly above answer-prior baselines on VSTAT.
- Through controlled experiments and analyses, we find that perceiving task-relevant events from the continuous visual stream is a major bottleneck.

2 VSTAT: Visual State Tracking Benchmark

Our benchmark, VSTAT, is designed to evaluate the *visual state tracking* capability of MLLMs throughout a continuous video stream. VSTAT follows the standard format of video benchmarks for MLLMs: given a video stream $\mathbf{v}$ and a query $\mathbf{q}$, the model f must predict the answer $\mathbf{y}$. Unlike prior video MLLM benchmarks, we construct VSTAT such that the answer cannot be inferred from a single keyframe or a small subset of frames. Instead, every task in VSTAT requires the model to process the entire video stream, track and update the information needed to derive the answer. One of the most popular examples is the shell game (Liu & Lee, 2026), which tests a player's observational skills by having them follow a hidden object as three cups are shuffled; the player must maintain the target cup's location throughout the video.

VSTAT comprises a diverse set of tasks requiring visual state tracking capabilities, drawn from both synthetic and real-world videos. Concretely, VSTAT consists of 834 video clips paired with 1,500 questions, derived from simulated videos rendered with Blender and real-world videos collected from YouTube and our own recordings. VSTAT covers diverse tasks with varied tracking targets, such as counting packed items, recognizing typed words, or attributing shots to players. This enables extensive evaluation and analysis of the visual state tracking capability of models across diverse video streams that contain continuous procedural processes. We provide illustrative examples in Figure 1, along with dataset statistics and a comparison with existing video benchmarks in Table 1.

2.1 DATA CURATION

In the rest of this section, we refer to the task examples illustrated in Figure 1.

Video curation. We curate our videos from both simulated environments and the real world. For simulated videos, we design 9 environments using the 3D software Blender and synthesize 450 video clips in total. For real-world videos, we collect 304 video clips from YouTube and record 80 additional videos ourselves in scripted settings. As a result, VSTAT contains 834 video clips in total. Across both sources, we focus on videos that contain diverse procedural processes such as solving puzzles, athletic plays, cooking, and order packing. In addition, each clip contains factors that make **perception** difficult; for example, the basketball clip involves continuous camera movement, while the order-packing clip exhibits frequent occlusion between items. We provide details of video categories, preprocessing strategies, and additional example visualizations in Appendix A.

Question-answer generation. From each video clip, we design questions that require visual state tracking over the video stream to predict an answer. We follow two design principles. First, all questions are designed to avoid visual shortcuts: the answer cannot be inferred from a few keyframes or the visible end state, forcing models to track state throughout the video. For instance, the book example's "how many pages are flipped?" cannot be answered without tracking the entire video. Second, we use diverse query types that require tracking of various **element types** and their **structures**. Some queries require tracking locations (*e.g.*, "*where* does the cube with the yellow sticker end up?"), while others require tracking total counts (*e.g.*, "what is the *total number* of shots?") or attributes such as characters (*e.g.*, "what is the *word* being typed?"). Each query also demands a different state structure: it can be atomic when tracking a single position or counter, or more complex structures such as sequences or sets when the query asks about detailed history of the video stream.

> VSTAT questions cannot be solved by visual shortcuts such as sparse keyframes or the end state, requiring models to maintain and update state across the entire video.

We design multiple questions for each video, with each question requiring the model to track different types of information. As illustrated by the basketball example, both the amount of information needed to answer a question and the associated difficulty can vary substantially depending on the query. Consequently, our benchmark contains 1,500 questions in total. We believe this "one video, multiple questions" format enables a comprehensive analysis of different aspects of models' visual state tracking capabilities. Our questions come in two formats: numerical questions (NQs), whose answers are single numbers, and multiple-choice questions (MCQs), which are used for all other question types and include carefully designed distractors. All videos, questions, answers, and category labels are annotated and reviewed through a human-in-the-loop verification; see Appendix A for details.

2.2 TAXONOMY

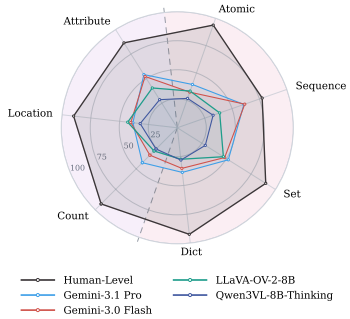
As explained in Section 2.1, our benchmark involves two crucial complementary axes: *perceptual complexity*, which captures factors in the video stream that make visual perception difficult, and *state complexity*, which captures the amount and type of minimum information that must be extracted from the video to answer the question. In what follows, we explain in detail the categories we define along each axis to classify each instance in our benchmark.

State complexity. As shown in the examples of Figure 1, each instance requires a different state complexity, which we decompose into two orthogonal dimensions: **element type** and **structure**. We consider three categories for **element type**: count (book example), location (cube example), and attribute (keyboard example). For **structure**, we consider four categories: atomic (book example), sequence (keyboard example), set (numberpad example), and dictionary (basketball example).

Perceptual complexity. From the video and question-answer pairs we collected, we consider the following six categories related to factors that make video perception difficult: occlusion (*e.g.*, the cube is hidden in some frames), camera motion (*e.g.*, the camera moves and the scene changes in the basketball clip), homogeneity (*e.g.*, multiple cubes share similar appearances), symbolic decoding (*e.g.*, typing events must be transcribed into characters), multi-entity attribution (multiple players move simultaneously in the basketball clip), and event ambiguity (*e.g.*, page flips can occur in either direction in the book example). Note that these categories reflect the major axes we observed across diverse procedural tasks, and may be extended to capture further variations.

Table 2: **Evaluation on VSTAT.** Scores report the reparsed MRA-with-MCQ metric. Dark gray indicates the best result among all models and light gray indicates the best result among open-sourced models. Ranks are computed separately within proprietary API models (1–4) and within open-sourced models (1–20, pooling Thinking and Instruct); baselines are not ranked.

Methods	Rank	Avg.	Count	Location	Attribute	Atomic	Sequence	Set	Dict
			State Element			State Structure			
Baselines									
Chance Level (Random)	-	26.1	25.0	26.7	25.0	28.2	25.0	25.0	25.0
Chance Level (Frequency)	-	37.8	41.3	33.5	35.1	39.2	26.6	41.4	39.9
Human Performance	-	90.5	92.8	89.9	86.4	93.7	77.5	90.0	92.4
Proprietary Models (API)									
Gemini-3.1 Pro (low)	1	44.4	42.6	38.5	54.1	39.5	60.8	51.9	38.7
Gemini-3.1 Pro (high)	2	43.9	42.1	41.6	49.9	40.1	56.8	50.0	39.3
Gemini-3.0 Flash (low)	3	39.8	33.4	40.3	52.2	32.5	61.6	48.2	35.2
Gemini-3.0 Flash (high)	4	38.8	33.2	36.6	52.5	31.4	62.4	48.4	32.4
Open-sourced Models Thinking									
MiMo-VL-7B	11	31.2	28.3	32.6	35.7	26.9	40.0	33.8	32.8
InternVL3.5-8B-Thinking	13	30.2	24.5	32.6	39.5	26.0	35.5	41.4	27.6
GLM-4.1V-9B-Thinking	14	30.2	24.8	33.9	37.2	26.9	33.2	40.8	27.3
Qwen3VL-8B-Thinking	18	28.2	25.9	32.2	28.6	26.8	32.7	28.5	28.0
Qwen3VL-4B-Thinking	19	26.0	21.0	31.8	30.2	25.5	30.8	29.5	21.4
Open-sourced Models Instruct									
LLaVA-OV-2-8B	1	35.1	28.3	43.0	40.5	33.5	38.7	46.9	27.3
LLaVA-OV-2-8B (codec)	2	35.0	28.6	42.0	40.6	33.9	37.0	46.3	27.6
Molmo2-4B	3	34.4	31.6	39.7	34.5	37.1	33.6	36.7	27.1
Cambrian-S-7B	4	34.2	33.2	33.6	36.9	34.0	30.6	40.2	32.5
Molmo2-8B	5	34.0	30.9	37.0	37.0	34.7	36.3	39.1	27.0
Qwen3VL-8B	6	33.2	30.9	37.0	33.9	32.4	33.3	37.9	31.5
InternVL3.5-2B	7	31.8	29.6	33.9	34.1	31.7	29.9	36.3	29.9
Cambrian-S-3B	8	31.8	29.7	32.7	35.0	32.7	31.9	35.1	27.2
VITA-1.5-7B	9	31.5	25.5	36.3	38.6	29.4	33.0	43.1	26.3
Qwen3VL-4B	10	31.3	27.0	33.3	37.9	30.4	32.8	39.8	25.8
InternVL3.5-8B	12	30.6	25.1	33.2	39.2	26.9	33.8	41.8	28.3
Qwen3VL-2B	15	29.4	29.4	28.2	30.5	32.5	24.9	32.1	23.5
Cambrian-S-1.5B	16	29.3	26.0	34.1	31.0	28.0	31.0	31.8	29.3
LLaVA-OV-7B	17	28.6	20.1	34.8	39.4	24.5	30.0	43.8	25.0
LLaVA-OV-0.5B	20	21.3	14.6	33.9	21.7	19.7	25.8	22.2	20.9



Note. Each question is labeled by state element (Count, Location, Attribute) and state structure (Atomic, Sequence, Set, Dict). Avg. is computed over all questions, not as the mean of bucket scores.

With this taxonomy, we label each video-question pair along all three axes and use these labels to ensure a balanced data distribution for the benchmark that is not skewed toward any particular aspect; in Appendix A, we provide the detailed statistics breakdown of our benchmark, including the aforementioned axes, duration of each video clip, and keywords in the questions. For future research, we also open-source these labels along with the questions and video clips in our benchmark.

3 Evaluation on VSTAT

Using VSTAT, we evaluate the performance of recent MLLMs and their shortcomings, including (a) proprietary models (API; Gemini-3.1 Pro (Google DeepMind, 2026) and Gemini 3.0 Flash (Google DeepMind, 2025)) with different thinking levels, and open-sourced models (Qwen3VL (Bai et al., 2025a), Cambrian-S (Yang et al., 2026), MiMo-VL (Xiaomi LLM-Core Team et al., 2025), InternVL3.5 (Wang et al., 2025), LLaVA-OV (Li et al., 2025), LLaVA-OV-2 (Lab et al., 2026), and Molmo2 (Clark et al., 2026)), by varying their model sizes and thinking mode, if applicable.

In particular, we investigate the following questions:

- How well do state-of-the-art MLLMs perform on VSTAT overall? (Table 2, Table 3)
- Why do MLLMs fail to solve tasks in VSTAT? (Table 4, Figure 2)
- When do MLLMs fail to solve tasks in VSTAT? (Figure 3)

3.1 Setup

Metrics and evaluation protocol. Our evaluation pipeline builds on LMMs-Eval (Zhang et al., 2024) and follows the standard evaluation protocol of MLLMs on video benchmarks. Following VSI-Bench

(Yang et al., 2024), we report the average of accuracy on MCQs and mean relative accuracy (MRA) on NQs. For open-sourced models, we sweep the maximum frame budget over $\{16, 32, 64, 128\}$ uniformly sampled frames and report the best score for each model; the selected budgets are 32 frames for Qwen3VL-8B (Bai et al., 2025a) and Cambrian-S-7B (Yang et al., 2026), 64 frames for Qwen3VL-4B, Qwen3VL-2B, and LLaVA-OV-2-8B (Lab et al., 2026), 128 frames for Molmo2-8B (Clark et al., 2026), and 16 frames for all other models like MiMo-VL (Xiaomi LLM-Core Team et al., 2025) and GLM-4.1V (GLM-V Team, 2025). We additionally report LLaVA-OV-2-8B with its codec video backend, which packs codec-sampled frames into canvases (32 canvases from up to 256 sampled frames) instead of uniform frame sampling. For proprietary models (Gemini; Google DeepMind 2026), we set the resolution parameters as MEDIUM for evaluation, as we observe no significant performance difference across resolution parameters, and set `max_tokens=65536` during evaluation for sufficient reasoning budget.

Chance level baselines. Following VSI-Bench, we provide two baselines: Chance Level (Random) is the random selection accuracy for MCQ tasks (and is inapplicable for NQ tasks). Chance Level (Frequency) represents the highest performance MLLMs would achieve by always selecting the most frequent answer for each task. This identifies performance gains that may result from inherently long-tailed answers or imbalanced multiple-choice distributions. We also report human performance as a sanity check, measured by authors who were not involved in constructing the corresponding questions, which shows the difficulty of VSTAT for humans. See Appendix B for more details.

3.2 Main Results

In Table 2, we report the evaluation results across three state **elements** (count, location, and attribute) and four state **structures** (atomic, sequence, set, dictionary), along with the overall average accuracy and rank. As shown in the table, only Gemini-3.1 Pro and Gemini-3.0 Flash are modestly above the Chance-Level (Frequency) answer-prior baseline, while other models perform even worse. In contrast, humans solve our benchmark with an average accuracy of 90.5%, far exceeding the chance-level baseline and existing MLLMs. This reveals that a large gap still exists between the visual state tracking capabilities of humans and MLLMs. One exception is tasks that require tracking of sequence states, which show opposite trends between humans and MLLMs: for humans this is the most challenging category compared with other state structures, but for MLLMs it is the best-performing category, showing the smallest gap. We also observe that all open-sourced models perform worse than Chance-Level (Frequency) across all numbers of frames fed into the models, and usually show only marginal improvement with increased model size, with Molmo2 and InternVL3.5 demonstrating slight degradation.

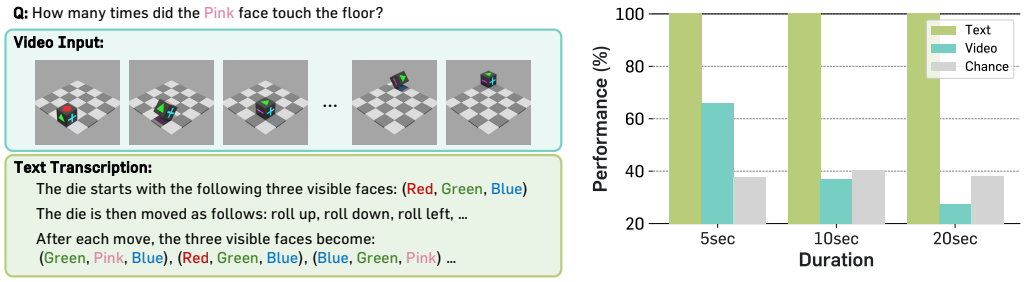
Notably, although LLaVA-OV-2 (Lab et al., 2026) and Molmo2 (Clark et al., 2026) are specifically trained with motion-grounded codec streams and pixel-space object tracking data, respectively, they do not demonstrate substantial improvements over other open-source MLLMs, despite being the two best-performing open-source models. This further suggests that VSTAT evaluates a more complex form of state tracking that goes beyond pixel-level tracking or motion-grounding objectives, requiring models to track the underlying latent state representations evolving throughout the video stream.

We also observe that enabling thinking mode or increasing thinking levels hurts performance, as shown in Table 3. Gemini-3.1-Pro is only mildly affected by higher thinking levels, with performance changing from 44.4 to 43.9, while Gemini-3.0-Flash drops from 39.8 to 38.8. Among open-source models, Qwen3VL-8B exhibits a substantial decline from 33.2 to 28.2, whereas InternVL3.5-8B shows only slight degradation, moving from 30.6 to 30.2. Notably, this observation aligns with the findings of Fu et al. (2026). After inspecting examples in Appendix C.4, we find that, for tasks with higher perceptual complexity, a larger thinking budget can increase the likelihood of hallucination for these models.

Model	Thinking	Performance	Δ
Gemini-3.1-Pro	low → high	44.4 → 43.9	−1.1%
Gemini-3.0-Flash	low → high	39.8 → 38.8	−2.5%
Qwen3VL-8B	w/o → w/	33.2 → 28.2	−15.1%
InternVL3.5-8B	w/o → w/	30.6 → 30.2	−1.3%

Table 3: **Thinking does not reliably improve performance.** Δ reports the relative performance change.

VSTAT is generally solvable by humans, but existing MLLMs struggle to solve it.



(a) Example task and its text transcription.

(b) Performance across video durations.

Figure 2: **Analyzing bottlenecks of MLLMs in visual state tracking.** (a) An example Blender task (rolling die) with its video frames and text transcription. (b) Performance across video durations on the selected task subset. Recent MLLMs, such as Gemini-3.1 Pro (Google DeepMind, 2026), solve the task perfectly with text conditions, but their video performance drops to near chance and degrades further as videos grow longer.

3.3 *Why* DO CURRENT MLLMS FAIL TO SOLVE VSTAT?

To analyze why this large gap occurs, we conduct additional experiments from two different perspectives. First, we examine whether this performance gap stems from event ambiguity caused by the information loss from frame subsampling when feeding the video to the model. Second, we investigate whether the gap arises from the model's visual perception or its reasoning capability. To this end, we take several simple tasks from the Blender environment, which allow us to control the number of events and the video duration, making them suitable for controlled analysis.

This gap does not mainly stem from event ambiguity. To rule out potential event ambiguity caused by the model's relatively low video sampling rate, we first compare the performance of Gemini-3.1 Pro on the original 5-second Blender videos with that on their temporally stretched versions, where each original frame is duplicated five times. This ensures that every event in the video is fully visible to Gemini even under the 1 FPS sampling rate.[2] However, as shown in Table 4, performance only marginally improves, suggesting that event ambiguity from frame subsampling is not the primary cause of the gap, which instead reflects fundamental limitations in the model's visual perception capability.

Data	Avg.
Chance level (Freq.)	39.2
5sec	51.4
5sec + stretch	53.6

Table 4: **Impact of video stretching**, evaluated on Gemini-3.1 Pro.

This gap stems from visual perception. We conduct an additional experiment to disentangle whether MLLM failures on visual state tracking tasks stem from visual perception or from reasoning. Specifically, we compare model performance under two conditions: the original video, and a text-only counterpart in which the video is replaced by a textual transcription of the visible states and events. If the gap between the two conditions is large, it suggests that visual perception, rather than reasoning, could be the primary bottleneck. Specifically, we consider three simple Blender tasks whose visible observations and events can be easily transcribed into text. For example, we consider the rolling die task, which requires counting how often a specific face lands on the bottom. Here, the transcription describes the three visible faces and the rolling direction at each step (see Figure 2a).

As shown in Figure 2b, state-of-the-art MLLMs solve these tasks *perfectly* when given textual transcriptions, yet their performance on video inputs drops to random-guess level once the video exceeds 10 seconds. Crucially, even on 5-second videos, where context length is negligible, performance already falls considerably short of the perfect text-only accuracy. While longer videos further degrade performance, the gap is already substantial at 5 seconds, suggesting that visual perception is the primary bottleneck, with errors compounding over longer videos. We observe the same pattern on other simple Blender-synthesized tasks (see Appendix C.2).

Visual perception appears to be a main bottleneck for models on VSTAT.

[2]Every event in the Blender videos lasts at least 0.2s; after temporal stretching, each event spans at least one second, making it fully visible to the model.

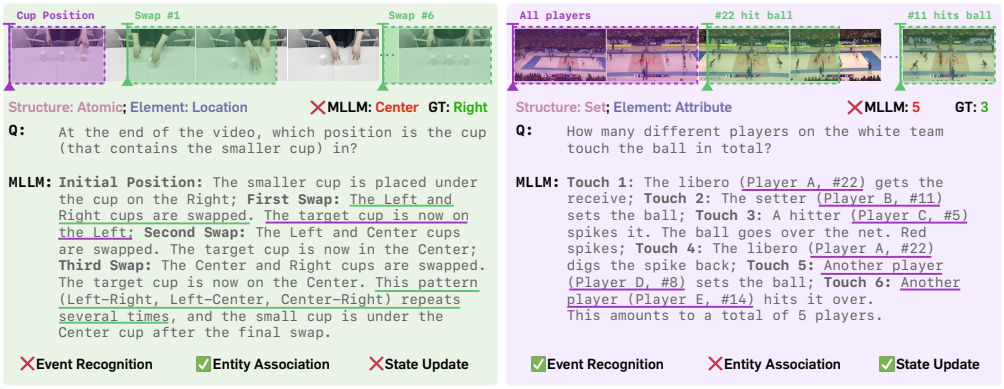


Figure 3: **Failures in event identification.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. For better illustration, we subsampled video frames related to the failures and simplified the thinking traces.

We emphasize that text transcription is not intended as a solution, but rather as a diagnostic tool for probing the fundamental bottleneck behind MLLMs' failures on VSTAT. On these Blender tasks, the perceptual gap is already so severe that we had to provide the text transcriptions by hand: even state-of-the-art MLLMs fail to reliably transcribe these simple synthetic videos into text. For real-world videos containing more complex dynamics and richer visual details (see Figure 1 and examples in Appendix A.2), text transcription would be even more challenging; in many cases, the descriptions can exceed the length of the videos themselves, making this approach infeasible.

3.4 *When* DO CURRENT MLLMS FAIL TO SOLVE VSTAT?

Next, we examine when MLLMs fail by analyzing the thinking traces of Gemini-3.1 Pro, the best-performing model in our benchmark. Comparing each video with its trace, we identify three recurring failure modes, summarized in Figure 3. We also conduct a quantitative error analysis, as shown in Figure 4. More examples and details are in Appendix C.3.

Event recognition. Even for relatively straightforward events in the video, the model can fail to correctly recognize the event and extract the corresponding state information. In the left example of Figure 3, the person swaps the center and right cups, but the model identifies the event as "the Left and Right cups are swapped," leading to an incorrect cup location and ultimately an incorrect final answer. In more challenging cases, we observe that the model may even hallucinate the entire event trace without correctly identifying any of the actual events (see Appendix C.3).

Entity association. Besides misidentifying events in the video, the model often fails when the state requires consistent association with a specific entity among visually similar objects. In the volleyball example, all players wear identical uniforms, so distinguishing them requires motion-based tracking. While the model correctly identifies each ball-touch event, it assigns a new random jersey number each time the ball is touched, even when the same player handles the ball repeatedly and their actual number becomes visible later.

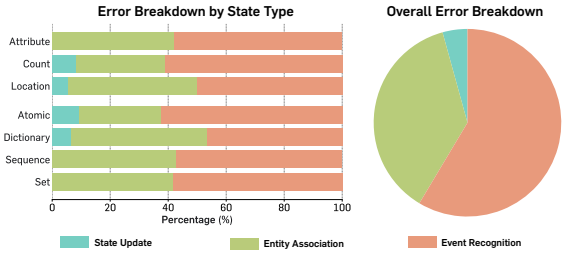


Figure 4: **Human-conducted error analysis.** More than 50% are from event recognition.

State update. Lastly, we observe an interesting failure pattern in which the model correctly recognizes the events and/or maintains the correct entity associations throughout the video, but fails to use this information to update the state needed for question answering. For example, in the left example of Figure 3, during the third swap, the model correctly identifies that "the Center and Right cups are swapped" and that the target cup was previously in the Center. However, it incorrectly updates the target cup's location as remaining in the Center, whereas it should move to the Right. We observe that this occurs more often when the model needs to track continuous trajectories, as it tends to over-simplify the observations and loses significant information from the video stream.

Quantitative analysis. As summarized in Figure 4, most failures are due to errors in either event recognition or entity association. In particular, more than 50% of failures stem from event recognition, suggesting that the dominant bottleneck of current MLLMs may lie in low-level perception, rather than visual reasoning. We also observe that state update errors highly correlate with the model's textual reasoning capability. Therefore, this error type remains relatively limited, likely due to the advanced reasoning abilities of current state-of-the-art MLLMs.

4 Related Work

Video Multimodal Large Language Models (MLLMs). Recent progress in multimodal understanding largely stems from MLLMs (Tong et al., 2024; Yang et al., 2026; Bai et al., 2023b; Wang et al., 2024a; Bai et al., 2025a), which incorporate powerful foundational visual encoders (Tschannen et al., 2025; Radford et al., 2021; Sim'eoni et al., 2025) into the strong linguistic understanding capabilities of LLMs (Brown et al., 2020; Bai et al., 2023a; Touvron et al., 2023). This success in the image domain has naturally led to the exploration of video-based MLLMs Li et al. (2024b; 2025); Zhang et al. (2025); Song et al. (2024); Bai et al. (2025b); Zhu et al. (2025); Yang et al. (2026); Comanici et al. (2025); Google DeepMind (2026); Bai et al. (2025a), which is essential for real-world applications that require multimodal intelligence, such as robotics (Black et al., 2024; Bjorck et al., 2025) and web agents (Gou et al., 2025). Our benchmark evaluates and diagnoses the visual state tracking capabilities of MLLMs, which are essential for many areas like robotics.

Evaluation of video MLLMs. To effectively measure progress, pinpoint current limitations, and guide future research, a series of benchmarks have been proposed to evaluate video MLLMs from different perspectives, including general video understanding (Fu et al., 2025a; Li et al., 2024a; Fu et al., 2026), event recognition (Xiao et al., 2021; Caba Heilbron et al., 2015), knowledge reasoning (Hu et al., 2025; Zhao et al., 2025), and temporal grounding and reasoning (Gao et al., 2017; Cai et al., 2024; Shangguan et al., 2025). More recent efforts impose stricter requirements, challenging models to comprehend hours- or even day-long videos (Song et al., 2024; Mangalam et al., 2023; Chandrasegaran et al., 2024; Wu et al., 2024; Wang et al., 2024b; Zhou et al., 2024; Wang et al., 2023) and to reason about the spatial information underlying video frames (Yang et al., 2024; 2026). Despite the breadth of these efforts, little to none attention has been paid to *visual state tracking*—the ability to continuously monitor visual states and events as they evolve over time. This capability is effortless for humans yet indispensable for real-world applications such as robotic manipulation, assistive agents, and surveillance, and it remains conspicuously absent from existing evaluation suites. This paper aims to fill this gap by proposing a benchmark to evaluate and diagnose existing MLLMs.

Video world models. Our benchmark shares some conceptual similarity with video world models (Gao et al., 2026; LeCun, 2022; Hafner et al., 2025; Guo et al., 2025; Sun et al., 2025; Kanervisto et al., 2025; Hong et al., 2025; Team Robbyant et al., 2026; Kong et al., 2025; Alonso et al., 2024; Ball et al., 2025), which aim to predict future states from previous states, actions, and observations. The main difference is that these methods typically assume actions are given explicitly and define the state representation as an approximation of the entire visual world, usually represented as latent video representations (LeCun, 2022) or the entire sequence of video frames (Gao et al., 2026). In contrast, our setting assumes actions are implicitly given through events, and the state is defined in latent space and related to the query, capturing only the partial information from the video necessary to answer it. We hope this connection also facilitates better evaluation of world modeling.

5 Conclusion

We present VSTAT, a video-based benchmark for diagnosing the visual state tracking capability of MLLMs. Our evaluation reveals a substantial gap between humans and current MLLMs, which only modestly exceed answer-prior baselines. Through controlled analyses, we further identify visual perception, rather than textual tracking, as the primary bottleneck, and diagnose recurring failure modes. We hope VSTAT serves as a useful diagnostic tool for the community to understand and improve the visual perception of MLLMs on continuous, real-world video streams. We discuss limitations and future directions in Appendix D.

AI USE STATEMENT

We use a generative AI tool to polish the writing of our manuscript. We did not use AI tools in other aspects of the research. Additionally, we carefully reviewed our manuscript to ensure that it does not contain AI-generated misinformation, such as hallucinated references. We take responsibility for the final content of this work. We also use coding agents (*e.g.*, Opus (Anthropic, 2026)) to implement simulation data using Blender.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. In *Advances in Neural Information Processing Systems*, 2024.

Anthropic. Introducing Claude Opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, April 2026. Accessed: 2026-05-02.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023b.

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.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Fei-Fei Li. Hourvideo: 1-hour video-language understanding. In *Advances in Neural Information Processing Systems*, 2024.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Chris Dongjoo Kim, Yue Yang, Ali Farhadi, and Ranjay Krishna. Molmo2: Open Weights and Data for Vision-Language Models with Video Understanding and Grounding. *arXiv preprint arXiv:2601.10611*, 2026.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2025a.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, Long Ma, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. VITA-1.5: Towards GPT-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025b.

Chaoyou Fu, Hao Yuan, Yuhao Dong, Yifan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yong Xie, Xiawu Zheng, Xuejiao Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding. *arXiv preprint arXiv:2604.05015*, 2026.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *IEEE International Conference on Computer Vision*, 2017.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

GLM-V Team. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Google DeepMind. Gemini 3 flash. https://deepmind.google/models/gemini/flash/, 2025.

Google DeepMind. Gemini 3.1 pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *International Conference on Learning Representations*, 2025.

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. MineWorld: A Real-Time and Open-Source Interactive World Model on Minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models. *Nature*, 2025.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. RELIC: Interactive Video World Model with Long-Horizon Memory. *arXiv preprint arXiv:2512.04040*, 2025.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-MMMU: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025.

Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, et al. World and Human Action Models Towards Gameplay Ideation. *Nature*, 2025.

Lingdong Kong, Wesley Yang, Jianbiao Mei, Youquan Liu, Ao Liang, Dekai Zhu, Dongyue Lu, Wei Yin, Xiaotao Hu, Mingkai Jia, et al. 3D and 4D World Modeling: A Survey. *arXiv preprint arXiv:2509.07996*, 2025.

Glint Lab, AIM for Health Lab, and MVP Lab. LLaVA-OneVision-2: Towards Next-Generation Perceptual Intelligence. *arXiv preprint arXiv:2605.25979*, 2026.

Yann LeCun. A Path Towards Autonomous Machine Intelligence. *Open Review*, 2022.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2024a.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2024b.

Tiedong Liu and Wee Sun Lee. Can vision-language models solve the shell game? *arXiv preprint arXiv:2603.08436*, 2026.

Yuanxin Liu, Kun Ouyang, Haoning Wu, Yi Liu, Lin Sui, Xinhao Li, Yan Zhong, Y.Charles, Xinyu Zhou, and Xu Sun. Videoreasonbench: Can MLLMs perform vision-centric complex video reasoning? In *International Conference on Learning Representations*, 2026.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models. In *International Conference on Learning Representations*, 2025.

Oriane Sim'eoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herv'e J'egou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards Long-Term Geometric Consistency for Real-Time Interactive World Modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Team Robbyant, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing Open-Source World Models. *arXiv preprint arXiv:2601.20540*, 2026.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. LVBench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024b.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. *arXiv preprint arXiv:2312.05269*, 2023.

Zeyu Wang, Zhenzhen Weng, and Serena Yeung-Levy. Continuous perception matters: Diagnosing temporal integration failures in multimodal models. *arXiv preprint arXiv:2408.07867*, 2024c.

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

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Xiaomi LLM-Core Team, Zihao Yue, Zhenrui Lin, Yi-Hao Song, Weikun Wang, Shu-Qin Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zi-Ang Jiang, Zhixian Zheng, Zhichao Song, Zhen Luo, Yue Yu, Yudong Wang, Yu Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xin dan Xu, Xin Ran Song, Xing Zhang, Xing Yong, Xin Zhang, Xia Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shi liang Yu, Shao yang Liu, Shan yong Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kang Zhou, Kang Zhou, Kai Fang, Jun-Miao Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongsheng Xu, Hengxu Qu, Hao-Song Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong-Yi Ma, Chang Liu, Can Cai, and Bing Xia. Mimo-vl technical report. *arXiv preprint arXiv:2506.03569*, 2025.

Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L Brown II, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-s: Towards spatial supersensing in video. In *International Conference on Learning Representations*, 2026.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024. URL https://arxiv.org/abs/2407.12772.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025.

Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8475–8489, 2025.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A BENCHMARK BREAKDOWN

A.1 DETAILED INFORMATION

Formal definition In Table 5 and 6, we provide a more formal definition of each category in our taxonomy, which is used for our labeling process (detailed in Appendix A.3).

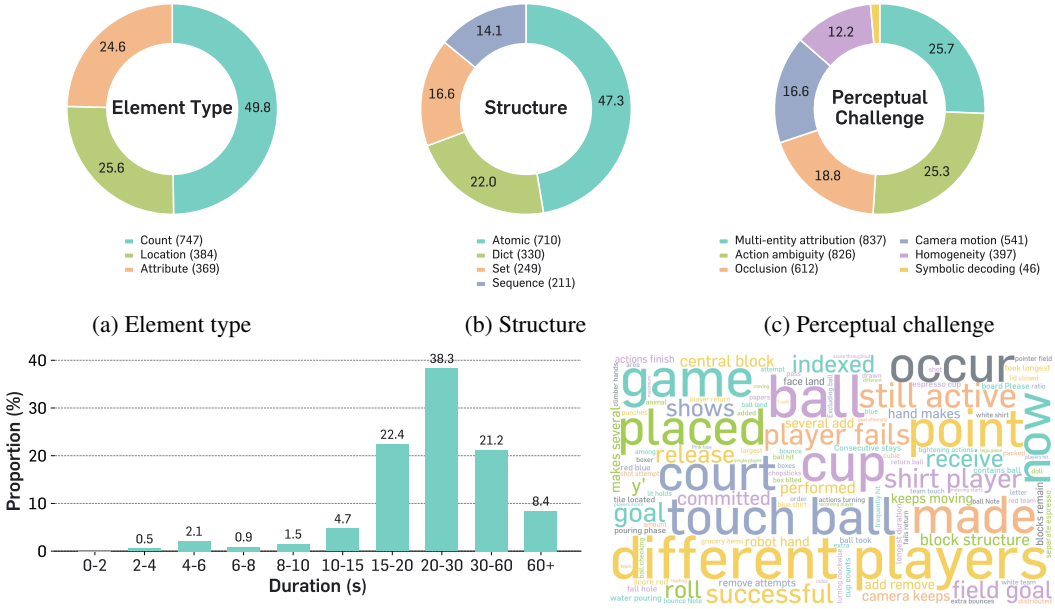
Table 5: **Taxonomy of state structures** across diverse queries and tasks (Table 7 and 8).

Type	Definition	Examples
Element type		
Count	An integer accumulated over time	# of passes in Basketball; # of steps in Cube
Location	A position in a discrete / continuous space	Position of ball in Shell game / Tilt box
Attribute	A categorical or vector-valued property	Characters in Morse code; longest Latte art
Structure		
Atomic	A single value at all time points	# of pages in Book; # of ingredients in Cooking
Sequence	An ordered series of values over time	Typing in Keyboard; scoring order in Tennis
Set	A subset of values, unordered	Distinct players in Volleyball; unpressed button in Numberpad
Dict	A map binding each entity to a value	hits per player in Tennis; max shots made in Basketball

Table 6: **Taxonomy of perceptual challenges** across diverse tasks (Table 7 and 8).

Challenge	Definition	Examples
Occlusion	The target is physically hidden behind other objects	Shell game; Cup stacking
Camera motion	Viewpoint shifts disrupt the spatial reference frame	Basketball; Carousel
Homogeneity	Multiple targets share identical appearance, making them hard to track individually	Cube; Lego
Symbolic decoding	A continuous visual pattern must be segmented and mapped to discrete symbols	Keyboard; Graffiti
Multi-entity attribution	Multiple objects act simultaneously, requiring state changes to be attributed to the correct one	Volleyball; NeuroTracker
Event ambiguity	Visually similar events produce different state outcomes	Tightening bolts; Numberpad

Statistics. We provide the statistics of VSTAT in Figure 5. As shown in the Figure 5a and 5b, our benchmark contains a balanced distribution across both element type and state structure, without skewing toward the simplest question types such as atomic count. Also, as shown in Figure 5c, VSTAT includes various perceptual challenges (*e.g.*, action ambiguity and camera motion). Moreover, as shown in Figure 5d, a majority of our videos have a duration shorter than 30 seconds, which is much shorter than the context length of frontier models like Gemini-3.1 Pro or Gemini-3.0-Flash. Lastly, as shown in Figure 5e, our questions consist of diverse keywords, covering a wide range of situations.



(a) Element type (b) Structure (c) Perceptual challenge

(d) Duration (e) Question keywords

Figure 5: **Benchmark statistics of VSTAT.** We show the distribution of (a) element types, (b) state structures, (c) perceptual challenges, and (d) video durations, along with (e) a word cloud of question keywords. The benchmark exhibits a balanced distribution across all dimensions.

A.2 CATEGORIES AND EXAMPLE VISUALIZATION

Video categories and examples. Table 7 and 8 list the tasks and their descriptions: the former covers tasks implemented in Blender, while the latter covers tasks we recorded ourselves or curated from YouTube. We also visualize video examples for each category in Figure 6 and 7.

Table 7: **Simulated video tasks** rendered in Blender. #Clips denotes the number of clips per task.

Task	Description	#Clips
Block count	Blocks in a 3D stack are shown in the video and blocks are randomly removed or added. The task is to predict the total number of blocks.	50
Rolling die	A die rolls across a surface, changing which face is up at each step. The task is to predict the total number of times a specific face is down.	50
Americano making	There is first a water-pouring phase and then a separate espresso-pouring phase. A cup counts as successful only if it receives both water in the first phase and espresso in the second phase. The task is to infer the number of successful cups.	50
Tightening bolts	Bolts are randomly tightened or loosened throughout the video. The task is to predict the total number of tightening actions.	50
Rotating shell game	A ball is hidden under one of several cups, the cups are shuffled, and the task is to track and predict which cup the ball ends up under. The camera also rotates throughout the video.	50
Sliding puzzle	Like the 15-puzzle, tiles on a grid are slid into the empty space and each tile randomly moves throughout the video. The task is to predict the final position of a specific block.	50
Tilt box	A box containing an object (e.g., a ball) is tilted in various directions, and the task is to predict where the object will end up.	50
Air hockey	A sequence of multiple air hockey plays. The task is to predict the total score, the longest game, or the number of own goals.	50
Funnel drop	Multiple balls are released into funnels at different times, where they roll around before falling through the hole. Balls are indexed left-to-right in the last frame before any release from `ball_1` through `ball_6`. The task is to predict which ball took the longest time to fall through the hole after its own release.	50

Table 8: **Real-world video tasks.** #Clips denotes the number of unique video clips per task.

Task	Description	#Clips
Book	A reader turns pages of a book either forward or backward. The task is to predict the net number of pages turned (signed).	10
Tilt box	A real ball inside a box is tilted in various directions starting from a known corner. The task is to predict the corner where the ball ends up.	10
Shell game	Several cups are shuffled with a smaller cup hidden under one of them. The task is to predict the final position of the cup containing the smaller cup.	10
Keyboard	A word is typed on a physical keyboard. The task is to identify the typed word.	10
Morse code	A light flashes a sequence in Morse code. The task is to decode the transmitted text.	10
Numberpad	A sequence of numbers is pressed on a number pad. The task is to identify which two digits were not pressed.	10
Cup stacking	Cups with animal drawings are stacked in a tower. The task is to identify the animal on the cup at a given position from the bottom.	10
Distributing items	Colored papers and chopsticks are distributed into cups. The task is to predict how many more items of a given type are needed for equal distribution.	10

Continued on next page

Table 8 – continued from previous page

Task	Description	#Clips
Basketball	Real basketball plays including shots, passes, and 3-pointers. The task is to predict shot counts, field goal percentages, and per-player or per-team statistics.	30
Bouldering	A climber moves on a wall with specific lit holds. The task is to count the total or maximum number of times the climber's hands or feet touch the lit holds.	10
Boxing	Two boxers exchange punches in a match. The task is to count punches by player, hand (left/right), or to determine the punch sequence.	19
Carousel	A carousel ride filmed from a rider's or external viewpoint. The task is to count people, complete rounds, or exit passes.	9
Cooking & barista	A chef or barista prepares foods and beverages such as sandwiches, burgers, noodles, espresso, latte, latte art, and sliced street food (yokan). The task is to count ingredients, cuts, cups, pours, or slices, and to identify preparation sequences or compare preparation times.	23
Cube	A Rubik's cube is manipulated through several moves. The task is to track where a specific colored cubie ends up.	16
Eating contest	Contestants eat burgers in a competition. The task is to count consumed burgers or determine the finishing order.	4
Graffiti	A person draws letters, words, or shapes on a wall. The task is to identify the drawn character or count the drawn shapes.	16
Horse racing	A horse race with multiple riders. The task is to predict final ranks of specific riders or count overtakes.	4
Lego	A person assembles a Lego model. The task is to count specific colored pieces, connections, or evaluate symmetry of the final build.	13
Marching band	A marching band performs on a field. The task is to count players of a specific instrument crossing the centerline.	4
Matryoshka	A set of nested Russian dolls is opened sequentially. The task is to count dolls or analyze headscarf and decoration patterns.	8
Order packing	Grocery items are packed into boxes or bags. The task is to count items, identify packing order, or reason about the minimum items to remove for visibility.	21
Soccer	A real soccer game with multiple players. The task is to count goals, passes, possessions, or compute success rates.	20
Tennis	Real tennis matches with players exchanging shots. The task is to count returns, identify ball landing zones, or determine scoring order.	20
Table tennis	Real table tennis matches between players. The task is to count hits, identify the server or winner, or track ball-table contacts.	30
Volleyball	A volleyball game with two teams. The task is to count total hits, distinct players touching the ball, or identify the team-contact sequence.	20
Sokoban	A Sokoban puzzle is played with boxes pushed onto target destinations. The task is to count pushes, identify which box is pushed, or determine the remaining optimal moves.	4
NeuroTracker	A subset of moving balls is highlighted at the start. The task is to track and identify the originally highlighted balls at the end among numbered candidates.	3
Memory card	A memory matching card game with face-down cards. The task is to identify matching pairs based on previously revealed cards.	9
Guess Who	Multiple players kick or throw balls; the task is to identify who successfully lands the ball in the basket.	21

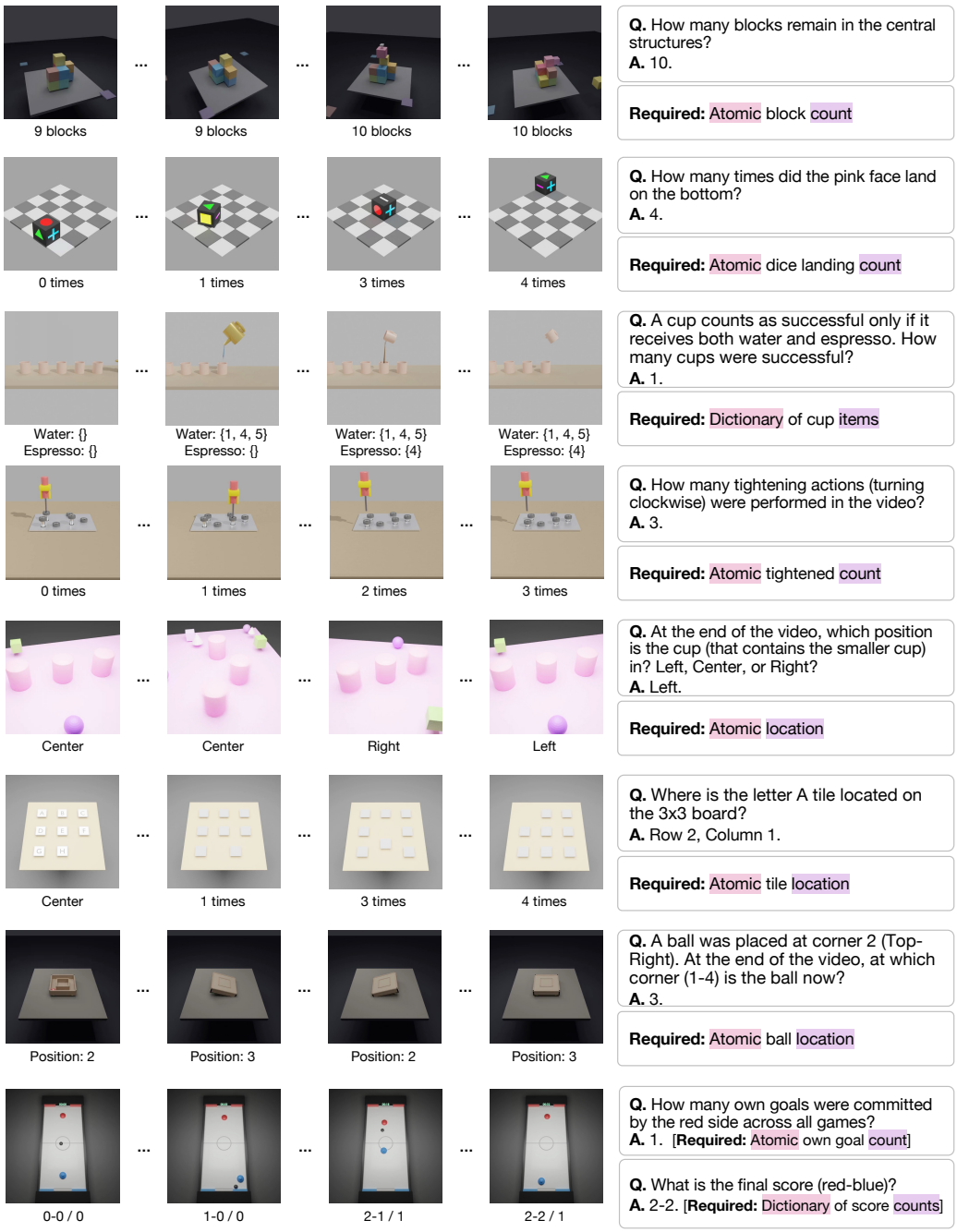


Figure 6: **Additional task examples in VSTAT synthesized with Blender.** Each task requires different state complexity and involves diverse perceptual challenges.

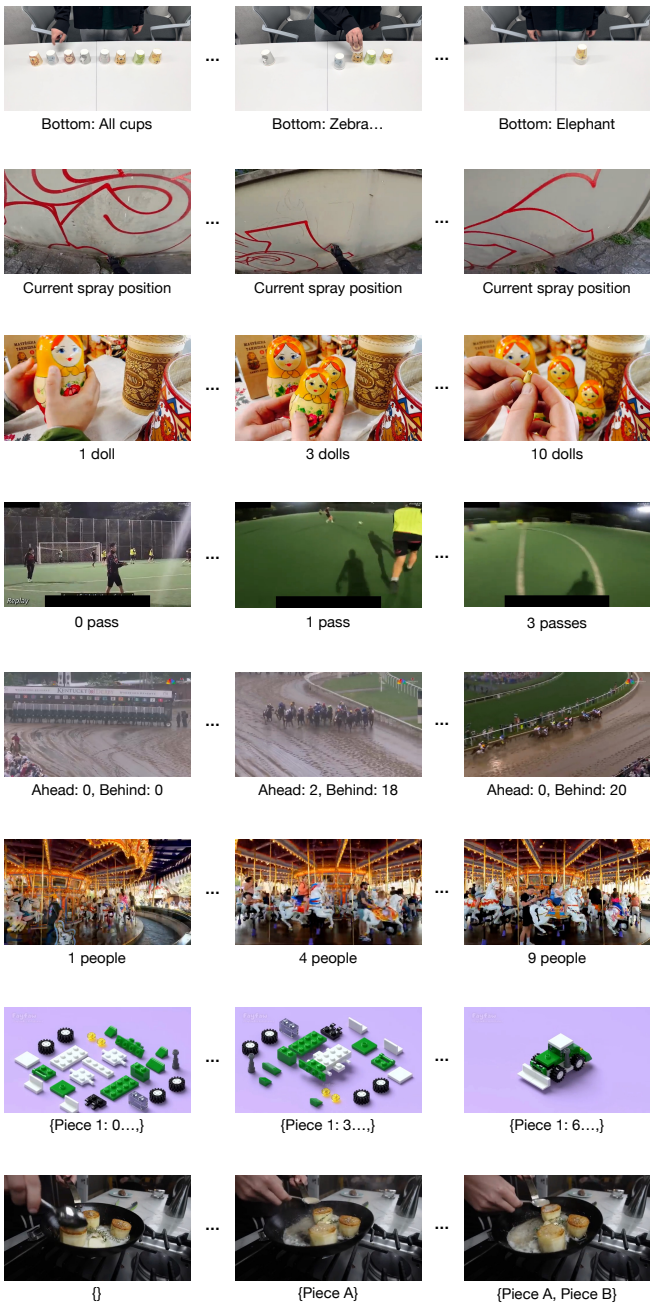


Figure 7: **Additional real-world task examples in VSTAT.** Each task requires different state complexity and has diverse perceptual challenges.

A.3 Curation and Filtering Process

Collecting and preprocessing strategies. For Blender videos, we set the video duration to 20 seconds for all tasks. For our analyses, we also synthesize shorter (5 seconds and 10 seconds) videos, but they are not included in our main benchmark and only used for the studies. For YouTube videos, we curate long-form footage and preprocess each video into clips with durations between 10 seconds and 1 minute, ensuring that no clip contains ambiguous events caused by clip boundaries. For example, in soccer clips, each video clearly shows whether a shot resulted in a goal. For recorded videos, all videos featuring identifiable persons were recorded by the authors with explicit consent for research and public release.

Question-answer generation with a human-in-the-loop process. For each video clip, we design various questions to ensure that each question requires a different minimum amount of information (*i.e.*, state complexity) to answer. For example, our questions include keywords such as "second-to-last" or "total" count, requiring the model to track information over the entire video. For videos that contain interactions among multiple entities with identical appearances, we construct questions that include keywords such as "how many people" or "who performed the action most", as these require distinguishing each entity, which is possible only if the model keeps track of the trajectory of each entity over time. Due to the lack of ground truth in video metadata for our hand-designed questions, as well as the limited visual state tracking capability of current MLLMs, automatic annotation for QA pairs is largely infeasible. We therefore manually labeled the answers to all questions. To ensure accuracy and eliminate ambiguity, every QA pair underwent at least two rounds of human validation. Any QA pair that human reviewers still deemed ambiguous after multiple rounds of review was removed from the final benchmark.

Multiple-choice question (MCQ) distractors. For MCQs, distractors are generated from plausible alternative states that could result from common tracking errors, rather than from semantically unrelated answers. Specifically, we provide the questions and answer choices without the video stream and check whether the model can predict the answer. In such cases, we reconstruct the other answer choices to avoid such shortcuts.

Labeling and filtering. To analyze performance with a breakdown, we label each question using our taxonomy. Each label is double-checked by a reviewer who has not labeled the question. We use the more formal definitions of each taxonomy in Table 5 and 6 to remove any ambiguity in labeling.

B Evaluation Setup Details

Human evaluation. To measure human performance, we internally built a website for evaluation. The evaluation was conducted by participants including the authors, but excluding those who had contributed specific videos or questions, to avoid any prior knowledge or information leakage. Participants were allowed to watch each video multiple times and think freely, but were strictly limited to a single answer per question. The ground-truth answer was never shown during the task, and each response was locked once submitted. We visualize our evaluation UI in Figure 8.

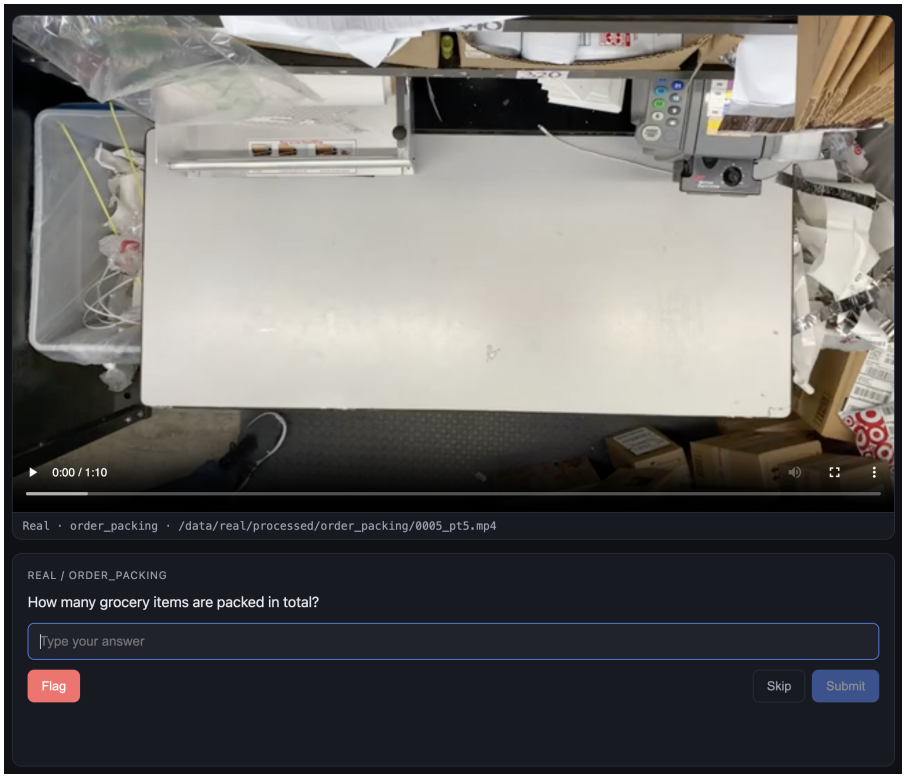


Figure 8: **Human evaluation UI.**

Chance-level performance. Following VSI-Bench (Yang et al., 2024), we consider two types of chance-level performance: (a) random and (b) frequency-based. For (a), we assume no access to the answer distribution and guess uniformly at random. We compute this accuracy only for multiple-choice questions (MCQs), not for numerical answers (NA). For (b), we estimate the empirical answer distribution p over both MCQs and NAs, and report the expected score of the best deterministic predictor: always predicting the most frequent answer (mode) for accuracy, and the optimal constant for MRA.

$$\begin{aligned} \mathrm{Acc}^{\mathrm{mcq}}_{\mathrm{rand}} &= \tfrac{1}{k}, \\ \mathrm{Acc}^{\mathrm{mcq/num}}_{\mathrm{freq}} &= \max_i p_i, \\ \mathrm{MRA}^{\mathrm{num}}_{\mathrm{freq}} &= \max_{c \in [\ell, h]} \mathbb{E}_a[\mathrm{MRA}_{\mathrm{thr}}(c, a)]\,. \end{aligned}$$

Here MRA denotes the threshold-based MRA following OpenEQA (Majumdar et al., 2024), with thresholds $\theta \in \{0.5, 0.55, \ldots, 0.95\}$. Note that we compute these accuracies separately for each question type, since the magnitude of the answers can vary substantially across them. For example, counting questions typically have maximum values below 10, whereas success/failure rates range from 0 to 100.

C Additional Results

C.1 Results across video sources

We provide the performance decomposition of different models on VSTAT across the three video sources: Blender, Recorded, and YouTube.

Table 9: **Evaluation on VSTAT by video category.** Scores report the reparsed MRA-with-MCQ metric, broken down by video source: *YouTube* (in-the-wild clips), *Synthetic* (rendered tasks), and *Recorded* (lab-recorded tasks). Dark gray indicates the best result among all models and light gray indicates the best result among open-sourced models. Ranks are computed separately within proprietary API models (1–4) and within open-sourced models (1–20, pooling Thinking and Instruct); baselines are not ranked.

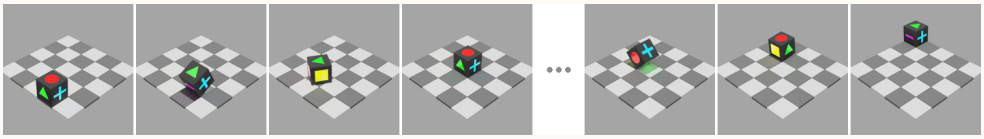
Methods	Rank	Avg.	YouTube	Synthetic	Recorded
			Video Category		
Baselines					
Chance Level (Random)	-	26.1	25.7	26.4	26.2
Chance Level (Frequency)	-	37.8	38.2	37.7	34.3
Human Performance	-	**90.5**	86.5	98.0	82.8
Proprietary Models (API)					
Gemini-3.1 Pro (low) (Google DeepMind, 2026)	1	44.4	42.6	38.5	54.1
Gemini-3.1 Pro (high) (Google DeepMind, 2026)	2	43.9	42.1	41.6	49.9
Gemini-3.0 Flash (low) (Google DeepMind, 2025)	3	39.8	33.4	40.3	52.2
Gemini-3.0 Flash (high) (Google DeepMind, 2025)	4	38.8	33.2	36.6	52.5
Open-sourced Models Thinking					
MiMo-VL-7B (Xiaomi LLM-Core Team et al., 2025)	11	31.2	35.3	24.3	34.3
InternVL3.5-8B-Thinking (Wang et al., 2025)	13	30.2	29.5	30.4	35.5
GLM-4.1V-9B-Thinking (GLM-V Team, 2025)	14	30.2	31.8	26.4	37.4
Qwen3VL-8B-Thinking (Bai et al., 2025a)	18	28.2	29.3	26.1	29.8
Qwen3VL-4B-Thinking (Bai et al., 2025a)	19	26.0	26.7	23.7	33.3
Open-sourced Models Instruct					
LLaVA-OV-2-8B (frames) (Lab et al., 2026)	1	35.1	40.6	27.7	29.0
LLaVA-OV-2-8B (codec) (Lab et al., 2026)	2	35.0	40.5	27.1	32.0
Molmo2-4B (Clark et al., 2026)	3	34.4	32.4	37.1	34.7
Cambrian-S-7B (Yang et al., 2026)	4	34.2	32.5	39.6	18.7
Molmo2-8B (Clark et al., 2026)	5	34.0	35.5	31.5	34.0
Qwen3VL-8B (Bai et al., 2025a)	6	33.2	36.9	29.2	23.9
InternVL3.5-2B (Wang et al., 2025)	7	31.8	31.7	33.1	26.0
Cambrian-S-3B (Yang et al., 2026)	8	31.8	33.2	30.0	29.4
VITA-1.5-7B (Fu et al., 2025b)	9	31.5	34.1	28.6	25.0
Qwen3VL-4B (Bai et al., 2025a)	10	31.3	34.1	27.1	30.5
InternVL3.5-8B (Wang et al., 2025)	12	30.6	32.7	27.6	30.1
Qwen3VL-2B (Bai et al., 2025a)	15	29.4	28.8	31.7	21.5
Cambrian-S-1.5B (Yang et al., 2026)	16	29.3	31.6	26.5	25.7
LLaVA-OV-7B (Li et al., 2025)	17	28.6	27.5	30.7	26.8
LLaVA-OV-0.5B (Li et al., 2025)	20	21.3	16.6	27.8	25.0

Note. Open-sourced rows use the same best frame setting as Table 2; the corrected `guess_who_make_it` subset is assigned to the YouTube source bucket.

C.2 TEXT TRANSCRIPTION EXAMPLES

We provide the full text transcription results of three Blender tasks (rolling die, shell game, and tilt box), along with their reasoning traces from Gemini-3.1 Pro (Google DeepMind, 2026) in Figure 9, 10, and 11. As shown in the figure, the model can do tracking and reasoning near perfectly.

Q. During the roll, how many times did the Pink face land on the bottom? (Consecutive stays count as 1, including the initial position)



[Text description.]
The die starts with the following three visible faces: Red on top, Green in front, and Blue on the right. The die is then moved as follows: roll up, roll down, roll left, ... After each move, the three visible faces become: (Green, Pink, Blue), (Red, Green, Blue), (Blue, Green, Pink) ...

[Gemini-3.1 Pro Thinking Trace]
To determine how many times the Pink face touched the floor, we first need to deduce the opposite pairs of faces on the die.

1. **Initial State:** The visible faces are Top = Red, Front = Green, Right = Blue.
2. **Move 1 (roll up):** The visible faces become Top = Green, Front = Pink, Right = Blue. When rolling up, the old Bottom face becomes the new Front face. Since the new Front is Pink, the initial Bottom face was Pink. Because the initial Top face was Red, we can conclude that **Red and Pink are opposite faces**.
3. **Move 3 (roll left):** From Move 2 (Top = Red, Front = Green, Right = Blue), rolling left makes the old Right (Blue) the new Top, and the old Bottom (Pink) the new Right. The visible faces are Top = Blue, Front = Green, Right = Pink.
4. **Move 4 (roll up):** Rolling up from Move 3 makes the old Bottom the new Front. The visible faces are Top = Green, Front = Yellow, Right = Pink. Since the new Front is Yellow, the old Bottom was Yellow. Because the Top in Move 3 was Blue, **Blue and Yellow are opposite faces**.
5. By elimination, the remaining two faces, **Green and White, are opposite faces**. Since Red and Pink are opposite faces, the Pink face is on the bottom (touching the floor) exactly when the Red face is on the top. We can track the Top face at the initial position and after each of the 19 moves by looking at the first color in each of the provided visible face triplets:
   - **Initial**: **Red** (Pink is on the bottom)
   - Move 1: Green - Move 2: **Red** (Pink is on the bottom)
   - Move 3: Blue
   - Move 4: Green
   - Move 5: Green
   - Move 6: **Red** (Pink is on the bottom)
   - Move 7: **Red** (Pink is on the bottom)
   - Move 8: **Red** (Pink is on the bottom)
   - ...

Now, we count the distinct times the Pink face was on the bottom, treating consecutive stays as 1:

1. Initial position
2. Move 2
3. Moves 6, 7, and 8 (counted as 1 distinct time since they are consecutive)
4. Move 17

The Pink face touched the floor 4 distinct times, including the initial position. **Answer: 4** ✓ **GT: 4**

Figure 9: **Text transcription and thinking trace summary** for the rolling die task.

Q. At the end of the video, which position is the Cup B (that contains the ball) in?



[Text description.]
Three cups are placed in order — Left, Middle, Right — and the ball starts under the Middle cup. Swaps (each swap exchanges the positions of the two listed cups): (Left, Middle), (Left, Middle), (Left, Middle), (Left, Right)...

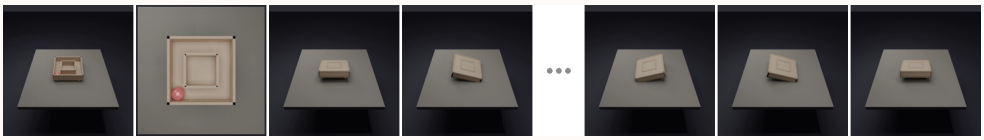
[Gemini-3.1 Pro Thinking Trace]
To determine the final position of the ball, we can track its location after each swap.

1. **(Left, Middle)** - The ball moves from Middle to **Left**.
2. **(Left, Middle)** - The ball moves from **Left** to **Middle**.
3. **(Left, Middle)** - The ball moves from Middle to **Left**.
4. **(Left, Right)** - The ball moves from Left to **Right**.
5. **(Middle, Right)** - The ball moves from Right to **Middle**.
6. **(Middle, Right)** - The ball moves from Middle to **Right**.
7. **(Middle, Right)** - The ball moves from Right to **Middle**.
8. **(Left, Right)** - The ball is not involved in this swap, so it stays at **Middle**.
9. ...
10. **(Left, Right)** - The ball is not involved in this swap, so it stays at **Middle**.

Answer: Center ✓ **GT: Center**

Figure 10: **Text transcription and thinking trace summary** for the shell game task.

Q.Which corner will the ball end up at?



[Text description.]
Label the corners of a box as follows:
1 — 2
| |
4 — 3
Place a ball at corner 2. Then tilt the box in the following directions in order: Down, Left, Down, Up, Right, Down.

[Gemini-3.1 Pro Thinking Trace]
To determine the final position of the ball, let's map out the corners of the box based on the provided diagram: Corner 1: Top-Left, Corner 2: Top-Right, Corner 3: Bottom-Right, Corner 4: Bottom-Left
The ball starts at **corner 2** (Top-Right). Now we follow the sequence of tilts:

1. **Down**: The ball rolls from the top edge to the bottom edge. From Top-Right (2), it goes to Bottom-Right (**3**).
2. **Left**: The ball rolls from the right edge to the left edge. From Bottom-Right (3), it goes to Bottom-Left (**4**).
3. **Down**: The ball is already on the bottom edge, so it stays at Bottom-Left (**4**).
4. **Up**: The ball rolls from the bottom edge to the top edge. From Bottom-Left (4), it goes to Top-Left (**1**).
5. **Right**: The ball rolls from the left edge to the right edge. From Top-Left (1), it goes to Top-Right (**2**).
6. **Down**: The ball rolls from the top edge to the bottom edge. From Top-Right (2), it goes to Bottom-Right (**3**).

Answer: 3 ✓ **GT: 3**

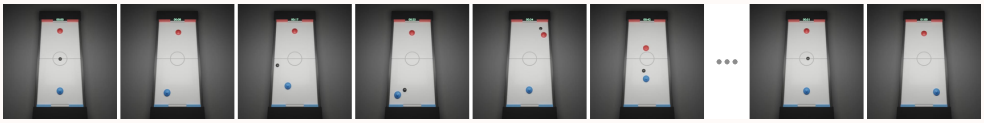
Figure 11: **Text transcription and thinking trace summary** for the tilt box task.

C.3 Additional Failure cases

Quantitative analysis details. From each video category, we select multiple questions, each requiring a different state element and structure, to ensure that our analysis covers the full diversity of question types in our benchmark. This yields a total of 70 questions, providing a comprehensive basis for analyzing failure cases across video content, state elements, and structures.

Illustration of additional failure cases. In what follows, we illustrate the failure cases from Gemini-3.1 Pro with their thinking traces.

Q. What is the final score (red-blue)? (A) 3-1 (B) 2-2 (C) 4-0 (D) 0-4



A. Based on the video, here is the breakdown of the goals:

1. At 00:08, the puck goes into the bottom goal. Red scores. (Red 1 - Blue 0)
2. At 00:18, the puck goes into the top goal. Blue scores. (Red 1 - Blue 1)
3. At 00:27, the puck goes into the top goal. Blue scores. (Red 1 - Blue 2)
4. At 00:39, the puck goes into the bottom goal. Red scores. (Red 2 - Blue 2)
5. At 00:49, the puck goes into the bottom goal. Red scores. (Red 3 - Blue 2)
6. At 00:58, the puck goes into the bottom goal. Red scores. (Red 4 - Blue 2)

The final score is Red 4, Blue 2. However, this option is not available. Let's re-examine the video carefully. It appears there might be a misunderstanding of the video's events or the provided options are incorrect. Given the strict instruction to only output a single letter from the choices, and none of the choices match the observed 4-2 score, I will select the closest option or re-evaluate.
Let's re-watch.
00:08 - Red scores (1-0), 00:18 - Blue scores (1-1), 00:27 - Blue scores (1-2), 00:39 - Red scores (2-2), 00:49 - Red scores (3-2), 00:58 - Red scores (4-2)
Answer: B ✗ GT: C

Figure 12: **Additional failure examples.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Event recognition. The model misunderstands the ball reflected off the red-side wall as a goal.

Q. Which two numbers on the number pad were not pressed? Answer with two numbers separated by a comma. (A) 5,9 (B) 2,9 (C) 1,9 (D) 0,9



A. Based on a careful observation of the video, here is the sequence of actions on the numberpad:

1. The user distinctly taps the numbers 1, 2, 3 (frames 00:01 to 00:03).
2. The user distinctly taps the numbers 4, 5, 6 (frames 00:04 to 00:06).
3. The user distinctly taps the numbers 7, 8 (frames 00:08 to 00:09).
4. After pressing 8, the finger moves downwards to press the decimal point '.' (frame 00:12). During this movement, the finger passes over the 0 key (frame 00:11) but does not perform a distinct tapping motion on it; it appears to be a continuous sliding motion towards the decimal point.
5. The finger never moves to or presses the 9 key.

Therefore, the two numbers that were not pressed are 0 and 9.
Answer: D ✗ GT: A

Figure 13: **Additional failure examples.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Event recognition. Occlusion makes the model infer an incorrect click action, leading to hallucinations.

Q. Does the eighth doll wear a headscarf? (A) No (B) Yes



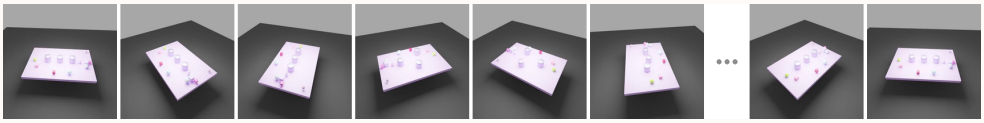
Based on the video, we can count the dolls as they are opened:

1. The first (outermost) doll is opened at 00:00.
2. The second doll is revealed at 00:05.
3. The third doll is revealed at 00:16.
4. The fourth doll is revealed at 00:25.
5. The fifth doll is revealed at 00:34.
6. The sixth doll is revealed at 00:45.
7. The seventh doll is revealed at 00:58.
8. The eighth doll is revealed at 01:07.

When the eighth doll is held up and shown clearly around 01:11, it has a white face, a blue body, and a red painted head covering, which represents a headscarf, just like the larger dolls. Therefore, the eighth doll does wear a headscarf. **Answer**: B ✗ GT: A

Figure 14: **Additional failure examples.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Event recognition. The model misses the reveal of the seventh doll at 00:50, leading to an incorrect prediction.

Q. At the end of the video, which position is the Cup B (that contains the ball) in? Left, Center, or Right? (A) Right (B) Center (C) Left



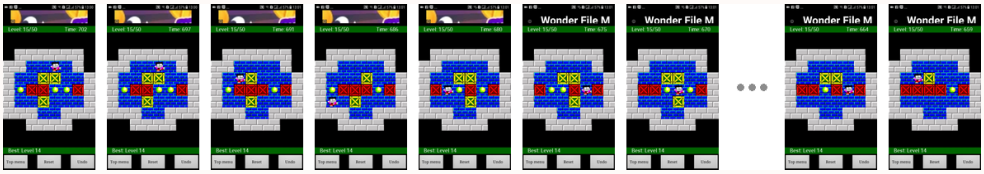
A. Based on a careful tracking of the cup containing the ball throughout the video:

1. 00:00: The ball is initially placed under the Center cup.
2. 00:02 - 00:04: The Center cup moves to the Top position.
3. 00:05 - 00:08: The cup moves from the Top to the Right, and then to the Bottom-Right position.
4. 00:09 - 00:12: The cup moves from Bottom-Right to Bottom-Center, and then back to the Center position.
5. 00:13 - 00:16: The cup moves from the Center to the Right, and then to the Top-Right position.
6. 00:17 - 00:19: The cup moves from Top-Right to the Center position as the cups realign.

The final position of the cup with the ball is in the Center.
Answer: B ✗ GT: C

Figure 15: **Additional failure examples.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Entity association. Continuous camera rotation changes relative positions (e.g., top, bottom, left, and right), leading to incorrect entity association.

Q. At the first frame of the clip, index the six boxes by reading each row from left to right, then moving from the top row to the bottom row (indexing starts from 1). Which indexed box is pushed last before the level is completed? (A) box 5 (B) box 2 (C) box 4 (D) box 1



Based on the indexing instructions, we first identify the six boxes in the initial frame (00:00) by reading from left to right, top to bottom:

- Top row: Box 1 (left yellow box), Box 2 (right yellow box)
- Middle row: Box 3 (left red box), Box 4 (middle red box), Box 5 (right red box)
- Bottom row: Box 6 (bottom yellow box)

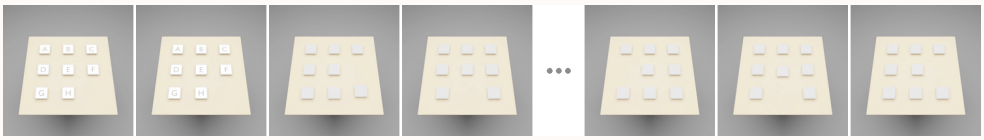
Now, let's trace the movements of the boxes throughout the video:

1. At 00:02, Box 6 is pushed down.
2. At 00:10, Box 3 is pushed right.
3. At 00:16, Box 5 is pushed left.
4. At 00:25, Box 1 is pushed right, placing it next to Box 2.
5. At 00:33, Box 2 is pushed down.
6. At 00:39, Box 1 is pushed right onto a target.
7. At 00:42, Box 2 is pushed right onto the final target, completing the level.

The last box pushed before the level is completed is Box 2.
Answer: B ✗ GT: C

Figure 16: **Additional failure examples.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Entity association. Changes in the box positions confuse the model's entity association, leading to an incorrect prediction.

Q. Where is the letter D tile located on the 3x3 board? Please answer in 'Row x, Column y' format, where x and y start from 1 (x=1, y=1 is the top-left corner). (A) Row 1, Column 1 (B) Row 2, Column 1 (C) Row 2, Column 3 (D) Row 2, Column 2



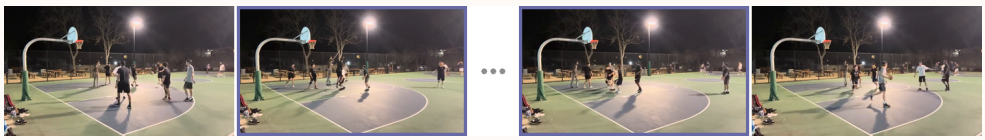
I can see the initial state: 'D' starts at Row 2, Column 1, with the blank space at Row 3, Column 3. I'm visually tracing the movements. First, the tiles shuffle. I note the time stamps and corresponding movements to minimize errors. Okay, I trace each move systematically, making sure I log where 'D' ends up after each step. I'm noting the empty spot's movement as the defining factor in where each tile goes.
Alright, I'm going through the moves step-by-step again. The initial configuration is clear. I see the tile, the empty, and the final state. I am tracing the movements meticulously and keeping track of 'D's position at the time specified. The verification confirms that the final position of D, is indeed Row 2, Column 3.
Answer: C ✗ **GT: B**

Figure 17: **Additional failure examples.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Entity association. Masked tile movement leads to hallucinated entity association from the model.

C.4 Comparison between different Thinking Levels

We illustrate the comparisons between different thinking levels from Gemini-3.0-Flash below.

Q. What's the number of different players that make a shot attempt in the video?



[thinking_level=low]
First pass, I went through and identified all the shot attempts...

1. 00:08-00:10: White shirt, black shorts.
2. 00:33-00:35: Grey shirt, grey pants.
3. 00:41-00:43: Black shirt, black shorts.
4. 00:47-00:49: White shirt, black shorts.

At this point, I think I'm starting to get a good handle on things, and I have a clear sense of the players involved:

1. Player 1 (White shirt, black shorts, shots at 00:08 and 00:47)
2. Player 2 (Grey shirt, grey pants, shots at 00:33)
3. Player 3 (Black shirt, black shorts, shots at 00:41)

Answer: 3 ✓

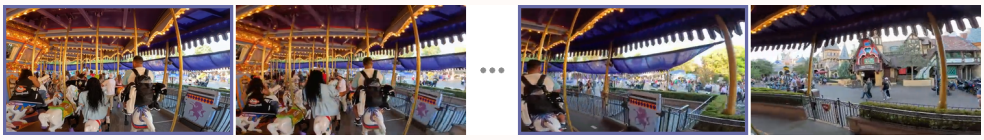
[thinking_level=high]
First, I identified a player in a white shirt (Player 1) making a shot at the start, at the 00:00 mark. Then, a player in a grey shirt (Player 2) takes a shot around 00:08. So far, so good. Then, it gets more interesting. At 00:21, we see a player in a black shirt and shorts, but it is clear that their shoes are white, which clearly distinguishes them (Player 3). After that, the black shirt, shorts, and black shoe combination (I'll call this Player 4 for now) appears repeatedly: at 00:33, 00:41, 00:47, and finally, 00:56. Initially, I was ready to count these as different players, but a closer examination proved this to be an error on my part. The key observation is that the player in the black shirt (with black shoes) appears to be *the same* individual, taking multiple consecutive shots. This is apparent because they grab their own rebound after each shot. We must be very precise with these details! Their position relative to the court markings and other players are identical each time.
Let me verify this one more time. The player at 00:21 clearly has the white shoes, whereas the one that makes multiple shots has black shoes. The person at 00:33, 00:41, 00:47, and 00:56 are the same person. Let us make this perfectly clear.
So, in summary, we have:

1. Player 1: White shirt
2. Player 2: Grey shirt
3. Player 3: Black shirt, white shoes
4. Player 4: Black shirt, black shoes

Answer: 4 ✗

Figure 18: **Thinking level comparisons.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Event recognition. The model with higher thinking level makes multiple perceptual errors in identifying 1. shot made; 2. the appearances of players.

Q. How many complete rounds did the person ride?

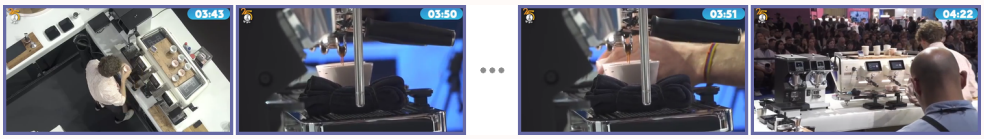


[thinking_level=low]
The video starts mid-ride, so I need a clear landmark to track rotations. I zeroed in on the purple dragon sign on the fence; it seems the most distinct and consistently visible.
Initially, I had a rough estimate based on the dragon sign's appearances – it reappeared roughly around the 32-second mark, and then again around 51 seconds. This suggested two rotations. However, the timing between those reappearances seemed a bit off, which flagged a potential issue with my initial assumption of a consistent speed or the possibility of an unclear landmark.
To confirm, I re-watched and focused closely on a prominent building or area in the background – a broader landmark that would offer a better frame of reference. Tracking that building, I confirmed that a full rotation was achieved around 00:25, and again at around 00:50, which seemed in line with my dragon sign observation.
I then revisited the dragon sign to confirm. Using the center of the frame as my key point, the dragon passes at roughly the 4-second mark, again around 32-seconds (1 rotation), and again around 51 seconds (2 rotations). The video ends at 00:59, and it's clear it doesn't quite make a third full revolution.
Answer: 2 ✓

[thinking_level=high]
The video starts mid-ride, so I need a clear point of reference to track a full rotation.
Alright, the first thing I did was look for a recognizable landmark on the outside of the carousel. At the four-second mark (00:04), bingo! There's a distinctive purple banner with a dragon or unicorn crest. That's my starting point. Now, let's track this crest as the carousel rotates.
Initially, I estimated a full round happened around 00:50. But that seemed off, especially since the video is only 59 seconds long. So, I re-evaluated. I went back and meticulously checked the timing. At 00:15 I spotted a clock tower and at 00:30 trees and a fence. Okay, 00:45 the dragon banner pops back up. Hmm...
Wait a second. The question is "complete" rounds. So, I need to make sure the same scene appears. Let's re-watch. At 00:00 I see the purple dragon sign. It's visible again at 00:48, so that's a completed round. The video ends at 00:59, so we're short of a second round.
To be extra certain, I went back and reviewed the background, specifically at the start and then at 00:48. Fence and purple dragon sign in the exact same view. Bingo! I can confirm the camera is facing the same direction.
So, one round. Just to make doubly sure, I looked at the start and end in relation to the position of the carousel. The carousel is at the same starting position at 00:48, and that's the only one before the video ends. So I can definitely say there is one complete round.
So, 00:00 - Start. 00:48 - One full rotation. 00:59 - End of video. Total complete rounds = 1. The answer is one.
Answer: 1 ✗

Figure 19: **Thinking level comparisons.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Entity association. The model with higher thinking level misses the first re-appearance of the dragon sign.

Q. How many cups of espresso are made or being made by the end of video?



[thinking_level=low]
[...]
Then, at 0:06, we get a close-up. We've got a double-spouted portafilter pouring espresso into two cups simultaneously. That's a good starting point. The camera stays focused on this. At 0:11, another portafilter engages and begins pouring into two more cups. So, during this close-up sequence, four cups are being filled.
Now, at 0:37, we get a wider shot of the whole setup and the barista. By the end, at 0:40, those cups are visible, and I need to account for all of them, so let's carefully re-examine. We see those 4 cups on top from the start.
[...]
Let's re-watch the video carefully, focusing on the timing. From 0:06 to 0:10, two cups, and then from 0:11 to 0:36, two more cups. The total is 4.
Answer: 4 ✓

[thinking_level=high]
[...]
Let's break this down systematically:

1. Initial Observation: At the very start (00:00), I see four cups already positioned on top of the espresso machine. Now, these could be pre-heated, empty cups, or they might contain finished espresso drinks. I need to clarify that.
2. Espresso Extraction: Between 00:06 and 00:36, the video clearly shows espresso being poured into two cups simultaneously from a single portafilter. This confirms that at least two cups are definitely *being made*.
3. Visual Confirmation: The wide shot at 00:37-00:42 reiterates what I saw before - the barista is working, and the four cups are still on top. He's filling the two cups under the group head.
4. The Big Picture: At 00:40, I see the whole setup. Four cups on top, two being filled. I don't see any other cups.

Answer: 6 ✗

Figure 20: **Thinking level comparisons.** We highlight phrases and frames related to state extraction in purple and failures in visual perception in green. *Failure reason:* Entity association. The model with higher thinking level double counts the same cups of espresso after a shot change.

D Limitations and Future Directions

Analysis using thinking traces. Our analysis relies on the thinking traces of frontier models, which are text outputs from MLLMs, as there is no established practice for interpreting their *visual* processing. Exploring *vision-centric* analyses that focus on intermediate visual representations would be an interesting direction toward better understanding MLLMs, and could guide future work on improving them in both pretraining and post-training.

Directions to improve performance on VSTAT. In this paper, we focus on demonstrating that existing MLLMs and agentic frameworks fail to solve VSTAT, and on analyzing why they struggle. A promising future direction is to develop better pre-training and post-training methods that directly target the perceptual bottlenecks revealed by VSTAT.

Video length. Since visual state tracking is already challenging for existing MLLMs at the current video lengths, we do not consider extremely long video streams (*e.g.*, hour-level) in constructing the benchmark. Once MLLMs achieve reasonable performance on VSTAT, a natural extension is to consider more challenging scenarios such as full console or e-sports gameplay, or entire sports matches. For instance, one could ask the model to compute the pass success rate over a full 1.5-hour soccer match.

Broader impact. VSTAT can facilitate better evaluation of MLLMs by exposing perceptual limitations overlooked by existing video benchmarks, which is important for a variety of real-world vision-grounded applications including sports analytics, medical video analysis, and embodied agents. Moreover, since our analysis suggests that perception may be the bottleneck in current MLLMs, VSTAT can guide future directions for MLLM pretraining and post-training. However, there are also potential side effects: as VSTAT gains adoption in the community, models may overfit to its specific patterns rather than develop general visual perception. We therefore encourage treating VSTAT performance as a necessary but not sufficient indicator of progress, complemented by evaluation on diverse out-of-distribution settings and concurrent evaluation across various existing benchmarks.

E Compute Usage

For synthetic data generation using Blender, we use an Apple M2 Max chip, 4× NVIDIA GeForce RTX 3090 GPUs, and 4× NVIDIA A100 Tensor Core GPUs. It takes less than 4 GPU-days to generate all videos in our benchmark. For evaluation, we use APIs from Google and Anthropic, and use 4× NVIDIA A100 Tensor Core GPUs to evaluate open-sourced models. Evaluating all open-sourced models reported in this paper also takes less than 4 GPU-days.