VLMs Hate Ads Too: Evaluating Robustness to Ad Interruptions in Videos

Anonymous ACL submission

Abstract

Current video benchmarks mostly rely on clean footage, overlooking the ubiquity of advertisements. To bridge this gap, we introduce Ads-VideoMMMU, a benchmark evaluating VLM robustness against realistic ad interruptions. Our experiments reveal that ads serve as semantic distractors that systematically impair perception and comprehension, causing state-of-the-art models like GPT-4o to suffer accuracy drops of up to 9.3%. Notably, we identify a "Prefix Penalty": Ads at the beginning of a video cause more damage than ads in other positions. We find two entangled causes for this: Models find it much harder to distinguish initial ads from the main content,and a "Visual Primacy Effect" where models over-prioritize early visual inputs. Furthermore, we characterize "last-mile errors" as a common failure mode under ad interference and propose a lightweight two-agent framework that effectively mitigates these failures.

1 Introduction

Vision language models (VLMs) (Zhang et al., 2024) have demonstrated remarkable capabilities in understanding long videos across diverse domains (Tang et al., 2025). However, a critical disconnect exists: While advertisements are ubiquitous on platforms like YouTube, current video benchmarks (Li et al., 2024; Fu et al., 2025) predominantly rely on curated, clean footage. This oversight leaves a blind spot: the robustness of VLMs in "in-the-wild" environments, where ad interference is inevitable, remains largely unexplored. Crucially, we argue that advertisements represent a unique challenge compared to traditional robustness tests. Unlike random pixel noise or simple occlusions, advertisements serve as "natural adversarial examples." They are not merely noise, but coherent, high-production-value segments designed specifically to capture attention. As a result, they compete directly with the main narrative for the model's focus and limited context window. This raises a pivotal concern regarding VLM robustness: Are models capable of effectively distinguishing these "semantic distractors" from core content, or does the intrusion of advertisements introduce a latent vulnerability that threatens the reliability of automated video analysis?

In this work, we take a first step toward understanding VLM robustness under realistic ad interruption. We focus on three questions:

- **Q1:** Do realistic ads systematically degrade long-video QA performance?
- **Q2:** Which ad setting is the most harmful? And why is it harmful?
- **Q3:** How do ads manifest in model failures?

To answer these questions, we introduce the **Ads-VideoMMMU** benchmark. We inject real ad clips into educational videos from VideoMMMU (Hu et al., 2025), creating a controllable dataset by systematically varying ad placement (prefix, middle, suffix) and duration. We then evaluate a suite of representative VLMs on this benchmark to assess models' robustness in the wild.

Through extensive experimentation, we report four primary findings: (1) **Systematic Degradation:** Ad insertion systematically degrades VLM video understanding. The state-of-the-art GPT-4o suffers a significant accuracy drop of 3.3% to 9.3%. (2) **Non-uniform Impact:** The impact of ads is not uniform across tasks or positions. Ad insertion primarily impairs low-level perception and comprehension tasks, and prefix ads (inserted at the start) causing the most severe degradation. (3) **Source of Vulnerability:** Through ad detection and "Needle-in-a-Haystack" experiments, we find that this prefix sensitivity stems from the model's difficulty in distinguishing ads from main content, compounded

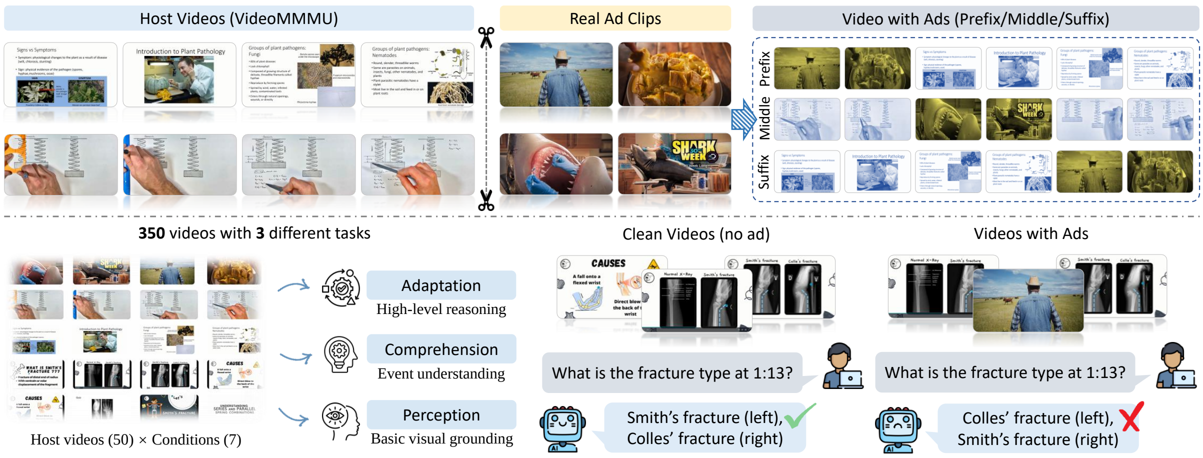


Figure 1: Overview of the Ads-VideoMMMU. **Top row**: Benchmark construction via injecting ad clips into host videos at varying positions. **Bottom row**: Three tasks and a case study showing how ads distract predictions.

by an inherent bias to overemphasize initial frames. (4) **Reasoning Fragility:** The introduction of ads induces fragility in the model's reasoning process, manifesting as an increase in the "inconsistency" of generated responses. In summary, our contributions are:

- We introduce **Ads-VideoMMMU**, to the best of our knowledge, it is the first benchmark to evaluate VLM robustness against realistic ad interference.
- We identify the "Prefix Penalty," where ads at the start cause the most severe damage due to the model's position bias and inability to distinguish ads from content.
- We identify last-mile errors as a common failure mode under ads and propose a lightweight two-agent framework to mitigate them.

We organize the rest of the paper as follows: §3 introduces the benchmark construction and evaluation setup. §4 presents the main experimental results, highlighting the non-uniform impact of ad interference. §5 investigates the mechanisms behind the "Prefix Penalty," uncovering the dual causes of this sensitivity. §6 provides a case study on error patterns, leading to §7, where we propose our method to mitigate these failures.

2 Related Works

2.1 VLMs for Video Understanding

VLMs have advanced significantly in visual understanding, with video emerging as a key and challenging frontier (Zhang et al., 2024). Early VideoQA benchmarks such as MSVD-QA (Xu et al., 2017), MSRVTT-QA (Xu et al., 2017), TGIF-QA (Jang et al., 2017), and ActivityNet-QA (Yu et al., 2019) primarily feature short video clips (under one minute) and factual questions about actions or relations. Recent benchmarks have pushed video understanding into longer durations and richer tasks. EgoSchema (Mangalam et al., 2023) uses 3-minute egocentric clips with over 5,000 questions requiring extended reasoning, while Video-MME (Fu et al., 2025) spans 11-second to 1-hour videos across six domains. MLVU (Zhou et al., 2024) further extends to 3-minute–2-hour videos with nine diverse tasks, and LVBench (Wang et al., 2025a) targets long-term memory with videos averaging 68 minutes. These benchmarks reveal that even state-of-the-art VLMs struggle with long-term context. The field of video understanding has evolved from short-form, single-task analyses to long-form, multi-task scenarios.

Despite great contribution made, most existing benchmarks focus on clean, curated footage and thus overlook a ubiquitous feature of real-world video platforms: advertisements. As a result, the robustness of VLMs to such in-the-wild noise remains largely unexplored.

2.2 Robustness of Vision-Language Models

Recent work shows that semantic distractors can severely disrupt reasoning in large models. For instance, injecting irrelevant but coherent content into prompts often derails multi-step reasoning (Zhang et al., 2025). Beyond text, visual prompt injection also poses concrete risks: adversarial patches can steer model outputs despite standard defenses (Sun et al., 2024), and subtle visual cues in medical images can mislead diagnostic VLMs even with safety filters (Clusmann et al., 2025). A related line of work shows that text embedded in images can reliably mislead VLMs by exploiting their bias toward textual visual signals (Cheng et al., 2024; Qraitem et al., 2024; Wang et al., 2025b). In the video domain, advertisements constitute a natural and pervasive source of noise or distractors. Inherently designed to compete for attention, they serve as an ideal testbed for evaluating robustness in the wild. Yet, despite their ubiquity, this form of realistic interference remains unexplored.

3 Benchmark Construction

3.1 Base Datasets

Host Videos. We construct our benchmark upon VideoMMMU (Hu et al., 2025), a widely adopted standard for long-video understanding. Our choice is motivated by two key factors: (1) **Ecological Validity**: The videos originate from YouTube educational channels, where ad interruptions are the norm, making this benchmark an ideal testbed for evaluating robustness under realistic noise. (2) **Diagnostic Granularity**: The benchmark features a hierarchical task design spanning Perception, Comprehension, and Adaptation. This structure enables us to disentangle how ads differentially impact basic visual grounding versus high-level reasoning.

To ensure rigorous evaluation while maintaining computational tractability, we employ a stratified sampling strategy to select 50 host videos. This subset is carefully aligned to preserve the original benchmark's distribution across diverse disciplines and durations (ranging from 40s to over 20 minutes). Crucially, this core set serves as the foundation for a combinatorial expansion: By systematically injecting ads at varying positions and durations, we generate 350 unique evaluation variants. This design allows for high-density probing of model robustness across distinct conditions without incurring prohibitive computational costs.

Advertisement Pool. To construct a diverse and challenging noise source, we sample 100 professionally produced commercial and public-service ads from AdsQA (Long et al., 2025). These ads are selected for their high production value and wide thematic coverage, ensuring they are visually distinct yet semantically disruptive to the educational host content. With runtimes typically spanning 30–120 seconds, this pool closely mirrors the temporal distribution of real-world advertisements.

3.2 Ad Insertion Protocol

Insertion Positions. We consider three positions: Prefix (pre-roll), Middle (mid-roll), and Suffix (post-roll). This design simulates the vast majority of advertisement placement scenarios encountered on real-world video platforms.

Duration Control. Instead of fixing absolute ad length, we define the ad budget ρ as the ads' share of the final video (original content + ads). Evaluation is restricted to a single ad insertion per video. Given host video duration T_{video} and ad duration T_{ad}, we set:

$$T_{\text{ad}} = \rho \cdot (T_{\text{video}} + T_{\text{ad}}), \quad \rho \in \{0.10, 0.25\}$$

This design ensures that (1) longer videos receive proportionally longer ads, matching platform norms, and (2) under uniform frame sampling, ρ directly controls the fraction of ad frames, enabling comparable interference across videos. We fill T_{ad} by randomly sampling and concatenating ads from the ad pool, truncating the final clip if needed.

3.3 Evaluation Task

Our benchmark evaluates whether VLMs can maintain video QA performance when videos are interrupted by realistic advertisements. Each test case presents an ad-interrupted video alongside a multiple-choice question from VideoMMMU, and the model must select the correct option.

Task Design. Adopting the VideoMMMU framework, we pair each video with questions targeting three distinct cognitive levels: **Perception** evaluates the recognition of explicit visual entities and extraction of on-screen text; **Comprehension** assesses the understanding of events, temporal relations, and the logical flow; and **Adaptation** challenges the model to generalize learned knowledge to solve novel problems in unseen domains. These tasks form a progressive difficulty gradient (Perception < Comprehension < Adaptation), enabling a granular analysis of how ad interference differentially affects capabilities ranging from basic visual grounding to high-level reasoning. Applying seven experimental settings (1 clean + 6 ad-inserted) to the 50 host videos yields **350** unique video variants. Paired with the three targeted questions per video, this results in a total of **1,050** evaluation instances.

Model	Baseline (No ad)	Prefix		Middle		Suffix	
		ρ=0.10	ρ=0.25	ρ=0.10	ρ=0.25	ρ=0.10	ρ=0.25
Closed-source Vision–Language Models							
GPT–4o	0.640	0.553 (-8.7%)	0.547 (-9.3%)	0.597 (-4.3%)	0.600 (-4.0%)	0.607 (-3.3%)	0.600 (-4.0%)
GPT–4o-mini	0.433	0.379 (-5.4%)	0.400 (-3.3%)	0.408 (-2.5%)	0.407 (-2.6%)	0.420 (-1.3%)	0.420 (-1.3%)
Gemini–2.5-Flash	0.620	0.600 (-2.0%)	0.597 (-2.3%)	0.604 (-1.6%)	0.600 (-2.0%)	0.607 (-1.3%)	0.613 (-0.7%)
Claude–3.5-Sonnet	0.564	0.540 (-2.4%)	0.551 (-1.3%)	0.520 (-4.4%)	0.527 (-3.7%)	0.560 (-0.4%)	0.560 (-0.4%)
Open-source Vision–Language Models							
Llama–4	0.560	0.540 (-2.0%)	0.533 (-2.7%)	0.547 (-1.3%)	0.527 (-3.3%)	0.553 (-0.7%)	0.550 (-1.0%)
InternVL3–78B	0.632	0.597 (-3.5%)	0.604 (-2.8%)	0.611 (-2.1%)	0.625 (-0.7%)	0.630 (-0.2%)	0.628 (-0.4%)
Qwen2.5–VL-72B	0.597	0.547 (-5.0%)	0.541 (-5.6%)	0.544 (-5.3%)	0.547 (-5.0%)	0.573 (-2.4%)	0.567 (-3.0%)
Qwen2.5–VL-32B	0.533	0.503 (-3.0%)	0.453 (-8.0%)	0.473 (-6.0%)	0.507 (-2.6%)	0.520 (-1.3%)	0.520 (-1.3%)
Average	0.572	0.532 (-4.0%)	0.528 (-4.4%)	0.538 (-3.4%)	0.543 (-3.0%)	0.559 (-1.4%)	0.557 (-1.5%)

Table 1: Overall accuracy under different ad-insertion settings. Δ shows the absolute percentage-point drop from the model's baseline (No Ad).

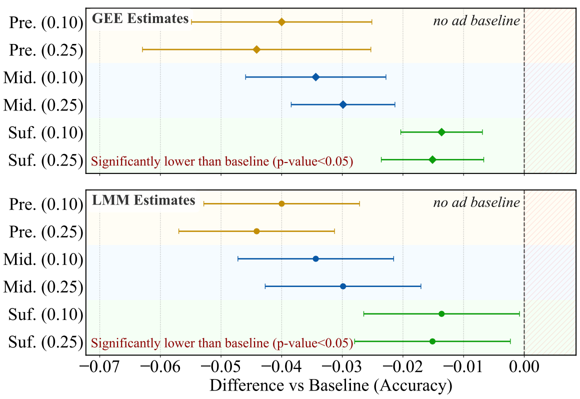


Figure 2: GEE and LMM estimates for accuracy differences vs. the no-ad baseline. Error bars represent 95% confidence intervals. All ad variants (Pre., Mid., Suf.) show a statistically significant negative impact on performance, with no intervals crossing the zero baseline.

4 Experiments

4.1 Settings

We evaluate a broad set of VLMs, including four closed-source systems (GPT-4o (OpenAI, 2024a), GPT-4o-mini (OpenAI, 2024b), Gemini-2.5-flash (Google, 2025), Claude-3.5-Sonnet (Anthropic, 2024)) and four open-source models (LLaMA-4-Maverick (AI, 2025), Qwen2.5-VL-72B/32B (Bai et al., 2025), InternVL3-78B (Zhu et al., 2025)). Following the original VideoMMMU protocol, we uniformly sample frames from each video and adapt the number of frames to each model family: 48 frames for GPT-4o and Gemini-2.5, 24 for GPT-4o-mini, 20 for Claude-3.5, and 32 for the Qwen2.5-VL, InternVL3, and LLaMA-4. All frames are fed to the models together with the question and choice options. Prompt design and evaluation strictly follow the official VideoMMMU setup without further task-specific tuning. For the QA task, we report accuracy as the primary metric.

4.2 Overall Performance Drop

Impact of Ad Insertion. Table 1 shows that the average accuracy on clean videos is 0.572. With prefix ads (ρ=0.10/0.25), accuracy drops by 4.0-4.4% to 0.532 and 0.528. Middle ads result in a 2.9-3.4% decrease (scores of 0.538 and 0.543), while suffix ads show a 1.3-1.5% decline, with values of 0.559 and 0.557.

Model Characteristics. Despite this shared trend, models differ in how strongly they are affected. GPT-4o achieves the highest accuracy on clean videos but also suffers the largest performance drop (3.3-9.3%) once ads are inserted, showing a "strong-but-fragile" profile. In contrast, Gemini-2.5-flash demonstrates greater robustness, maintaining solid baseline performance (0.62) with minimal degradation (around 1–2%) under ad exposure.

Statistical Validation. We conducted a statistical analysis to validate these observations. Specifically, we fitted a Linear Mixed Model (LMM) and a Generalized Estimating Equation (GEE). The results under all ad conditions were consistent: both models yielded negative and statistically significant coefficients (all $p < 0.05$, Figure 2, and detailed results are provided in Appendix A.2), providing statistical confirmation that ad insertion reliably degrades VLM performance on video QA.

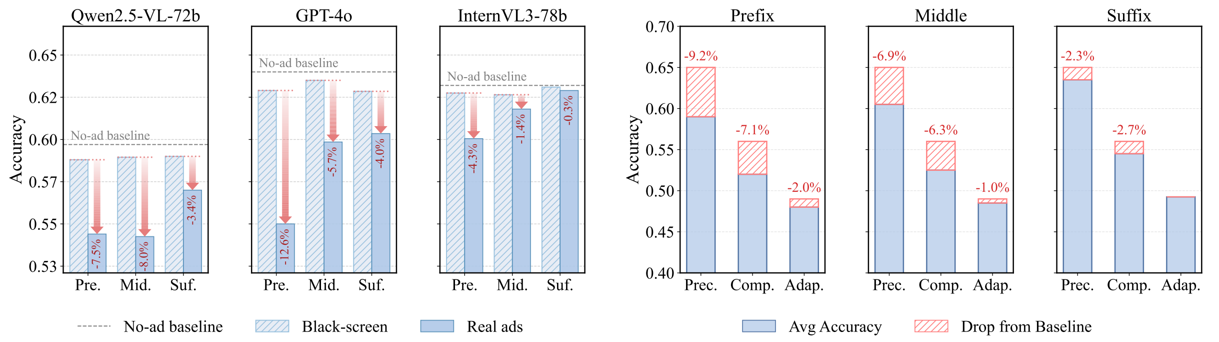


Figure 3: Impact of ad insertion on video QA. **Left:** QA accuracy with black-screen vs. real-ad insertion across three models. Performance with black screens stays close to the no-ad baseline, while realistic ads cause a larger drop, especially in prefix positions. **Right:** Task-wise breakdown shows that ads affect tasks unevenly: Perception and Comprehension drop more than Adaptation across all insertion positions.

4.3 Ruling Out Duration

While the performance drop is strong and clear, ad insertion inevitably increases video length. This raises a key question: does the decline stem from the dilution of original video content (due to subsampling) or from the semantic interference of the ads themselves? To disentangle these two factors, we conduct a controlled black-screen experiment.

Experimental Design. We replace real ads with black frames while preserving identical positions and durations. Balancing budget cost and representativeness, we conduct this control on three models (Qwen2.5-VL-72B, GPT-4o, and InternVL3-78B), each evaluated under all settings (prefix/middle/suffix $\times$ $\rho \in 0.10, 0.25$).

Results. Figure 3 (left) summarizes QA accuracy for three representative models under different conditions (we average the results across the two ad budgets). All three models exhibit a consistent pattern: inserting black screens results in only a minor performance drop relative to the no-ad baseline, whereas realistic ads lead to a significantly more pronounced decline, particularly in prefix settings. Our results indicate that the performance drop is primarily driven by the irrelevant semantic and visual content of the advertisements—not by the increased video duration.

4.4 The Uneven Impact

Having established that semantic content drives the performance degradation, we now examine how this impact varies across different insertion positions and task types. The results reveal two uneven patterns.

The Prefix Penalty. Initially, we hypothesized that Middle ads would be the most disruptive, as they physically interrupt the continuous flow of video information. However, our experiments reveal a contrary trend: Prefix ads consistently cause the most severe performance degradation. On average, Prefix insertion leads to a drop of 4.0-4.4%, compared to 3.0-3.4% for Middle and only 1.4-1.5% for Suffix. This trend is particularly pronounced in stronger models like GPT-4o, where Prefix ads cause accuracy drops of up to 9.3%, nearly double the impact of Middle ads.

Collapse in Perception. Across task types, degradation is not uniform. From Figure 3 (right) Perception and Comprehension tasks show sharp declines (6-9%), while high-level Adaptation tasks remain resilient (1-2%). This selective vulnerability reveals that ads primarily degrade low-level visual perception, while higher-level reasoning reliant on global knowledge remains largely intact.

5 Why Do Prefix Ads Hurt Most?

§4 shows that realistic ads broadly degrade long-video QA and reveals an interesting *prefix penalty*: pre-roll ads are uniquely destructive. This is particularly concerning given that pre-roll ads are the most prevalent format in real-world streaming platforms. To explain this phenomenon, we conduct a targeted mechanistic analysis. Across our experiments, we identify two intertwined factors that jointly produce the prefix penalty: (1) **semantic interference**, where the model fails to separate ads from the main storyline; and (2) **visual primacy effect**, where the model pays more attention to visual content presented early on. We probe these

Model	ρ	Prefix		Middle		Suffix		No Ad		Overall	
		Recall	F1	Recall	F1	Recall	F1	Recall	F1	Acc.	F1
Closed-source Vision-Language Models											
GPT-4o	ρ=0.10	0.82	0.42	0.73	0.63	0.87	0.69	0.93	0.88	0.84	0.65
	ρ=0.25	0.80	0.37	0.68	0.60	0.86	0.65	0.99	0.98	0.83	0.65
GPT-4o-mini	ρ=0.10	0.72	0.20	0.68	0.25	0.74	0.54	0.51	0.50	0.66	0.37
	ρ=0.25	0.72	0.28	0.67	0.35	0.74	0.56	0.66	0.59	0.70	0.44
Gemini-2.5-flash	ρ=0.10	0.77	0.13	0.82	0.73	0.84	0.71	0.86	0.72	0.82	0.57
	ρ=0.25	0.77	0.13	0.70	0.60	0.88	0.76	0.92	0.83	0.82	0.58
Claude-3.5-sonnet	ρ=0.10	0.79	0.28	0.88	0.37	0.79	0.63	0.55	0.46	0.75	0.43
	ρ=0.25	0.76	0.07	0.76	0.30	0.76	0.58	0.55	0.46	0.71	0.35
Open-source Vision-Language Models											
Llama-4	ρ=0.10	0.76	0.33	0.75	0.00	0.73	0.54	0.67	0.58	0.73	0.36
	ρ=0.25	0.75	0.25	0.74	0.00	0.64	0.52	0.79	0.68	0.73	0.36
Qwen2.5-VL-72b	ρ=0.10	0.78	0.27	0.83	0.53	0.81	0.51	0.56	0.52	0.74	0.46
	ρ=0.25	0.72	0.23	0.84	0.56	0.87	0.71	0.68	0.60	0.78	0.52
Qwen2.5-VL-32b	ρ=0.10	0.76	0.12	0.78	0.31	0.78	0.57	0.58	0.54	0.72	0.38
	ρ=0.25	0.77	0.22	0.77	0.33	0.84	0.69	0.66	0.59	0.76	0.46
Intern3VL-78b	ρ=0.10	0.89	0.62	0.78	0.49	0.89	0.77	0.69	0.61	0.81	0.62
	ρ=0.25	0.82	0.55	0.91	0.78	0.93	0.87	0.82	0.71	0.87	0.73
Average	–	0.77	0.28	0.77	0.43	0.81	0.64	0.71	0.64	0.77	0.50

Table 2: Ad-detection performance of eight Vision-Language Models across four insertion positions (Prefix, Middle, Suffix, No Ad) and two ad-budget levels ($\rho = 0.10$ / $\rho = 0.25$).

factors with an ad-position detection task and a needle-in-a-haystack test.

5.1 The Blind Spot: Ad Detection Failure

Task Design. We construct a four-way classification task where models must detect and localize whether a video contains advertisements or other non-essential content at the beginning (Prefix), middle (Middle), end (Suffix), or not at all (No Ad). From the same 50 host videos, we generate 350 samples by varying insertion positions and budgets ($\rho \in \{0.10, 0.25\}$), yielding a balanced dataset with 25% samples per class.

Observation. Models exhibit a strong positional asymmetry regarding detection accuracy. Prefix ads are the hardest to detect (F1=0.28), followed by Middle ads (F1=0.43), while Suffix ads and clean videos (No-Ad) achieve the highest performance (both F1=0.64). Higher-budget insertions ($\rho = 0.25$) are slightly easier to detect than lower-budget ones ($\rho = 0.10$), with average F1 scores of 0.51 and 0.48, respectively. As shown in Figure 4, the dominant failure mode is a false negative prediction: models frequently misclassify ad-containing videos as No Ad.

Discussion. The detection error pattern closely matches the QA degradation in §4 (Prefix>Middle>Suffix): the position where

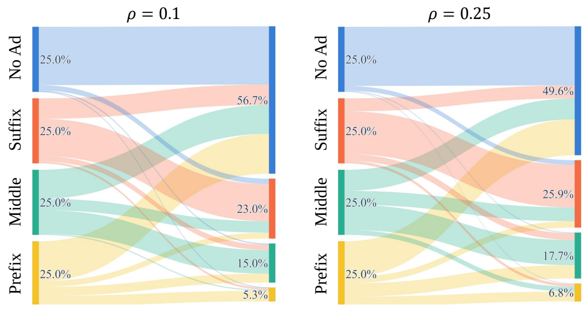


Figure 4: Sankey diagram illustrating the flow from ground truth (left) to model predictions (right). The dominant flow into the "No Ad" category reveals a false negative bias, where models frequently fail to detect inserted ads and mistake them for clean content.

models are most "blind" to ads is exactly where QA accuracy drops the most. Specifically, when a model fails to recognize an ad segment as irrelevant, its answer accuracy is significantly compromised (prefix and middle ads). Conversely, when models successfully *gate* these extraneous segments, performance remains robust (suffix ads).

Why are prefix ads the hardest to detect? While the solid performance on Suffix ads (F1=0.64) proves that VLMs are capable of distinguishing ads, they paradoxically lose this ability at the very beginning of the video. We attribute this to the 'cold start' problem: without preceding video content as a contextual reference, the model lacks a

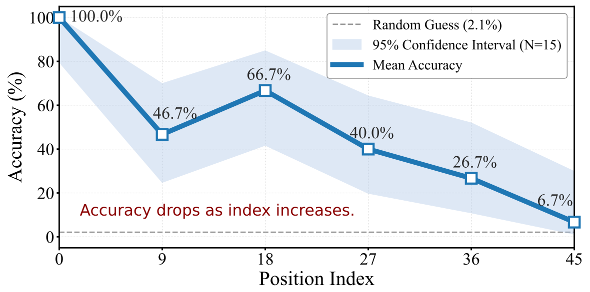


Figure 5: Illustration of the visual primacy effect. Accuracy drops significantly as the target information moves away from the start (Index 0).

reference to identify the ad as an outlier. It erroneously accepts the ad as the video's introduction. Crucially, this error is likely compounded by how VLMs process long sequences. When models overweight the initial frames to build global context, failing to detect a prefix ad can derail the reasoning process. To verify, we next probe the model's positional bias.

5.2 Position Bias of VLM

Task Design. We therefore probe whether the prefix position itself has disproportionate influence on context construction via a controlled *needle-in-a-haystack* retrieval task. Each sample contains 48 images, where each image is a receipt-like table with 30 unique items—yielding 1,440 candidate rows in total. A single target row (the needle) is specified by its `ITEM NAME`; given the full sequence, the model must locate the target and extract all fields from that row (`ASSET ID`, `SECURITY CODE`, `LAST MAINT`, `STATUS`) in JSON. We place the target image at six predefined positions (i.e., 1, 10, 19, 28, 37, 46), randomize all other images, and measure exact-match retrieval accuracy. To reduce randomness, we repeat each setting 15 times with different random shuffles. We evaluate GPT-4o as a representative strong model. For more details, please see the Appendix A.4.

Results and Discussion. Figure 5 reveals a strong position dependence on GPT-4o: accuracy is highest when the target appears at the beginning (Index 0: 100.0%) and generally declines as it moves later in the sequence (e.g., 40.0% at 27, 26.7% at 36, and 6.7% at 45; averaged over 15 shuffles). The overall trend indicates a pronounced position bias, where early inputs disproportionately anchor what the model retains and uses for retrieval. Such position bias is known to become more salient under *long-context* and more demanding retrieval settings—precisely the regime of long-video understanding—and is consistent with recent analyses of order/position sensitivity in multimodal models (Tian et al., 2025; Tan et al., 2024) as well as "lost-in-the-middle" effects observed in long-context reasoning and retrieval (Liu et al., 2024).

Takeaways

We attribute the "Prefix Penalty" to an interaction between two mechanisms:
1. **Detection Failure** (The Trap): Due to a "cold start" lack of context, models fail to identify prefix ads as outliers ($F1 = 0.28$), erroneously accepting them as the narrative introduction.
2. **Visual Primacy** (The Amplifier): This error is compounded by a position bias, where models disproportionately anchor their reasoning on initial frames.

6 Case Study

To further understand how ads degrade video understanding, We examined all failure cases from two representative closed-source models (GPT-4o and Claude-3.5-Sonnet) and observed a consistent instability caused by ad insertion. Specifically, we identified a 'Last-Mile Error', where models correctly deduce the answer in their reasoning but select the wrong option (Figure 6, more details are in Appendix A.5).Using a two-stage review pipeline (GPT-4o screening followed by human auditing), we found that this inconsistency is rare in clean videos (about 5%). However, it surges to 15–16% in videos with Prefix and Middle ads. Suffix ads have a much milder impact, raising the rate to only 6–7%. This observation is consistent with prior evidence that subtle, irrelevant perturbations in prompts or images can make model outputs brittle and increase internal inconsistencies (Turpin et al., 2023; Balasubramanian et al., 2025).

7 A Two-Agent Consistency Framework

To mitigate the ad-induced reasoning fragility identified in our case study, we propose a lightweight two-agent consistency framework (Figure 7). The key idea is to directly target the observed failure mode—explanation-answer inconsistency.

Reasoning Agent: GPT-4o generates reasoning and a candidate answer from video frames, ques-

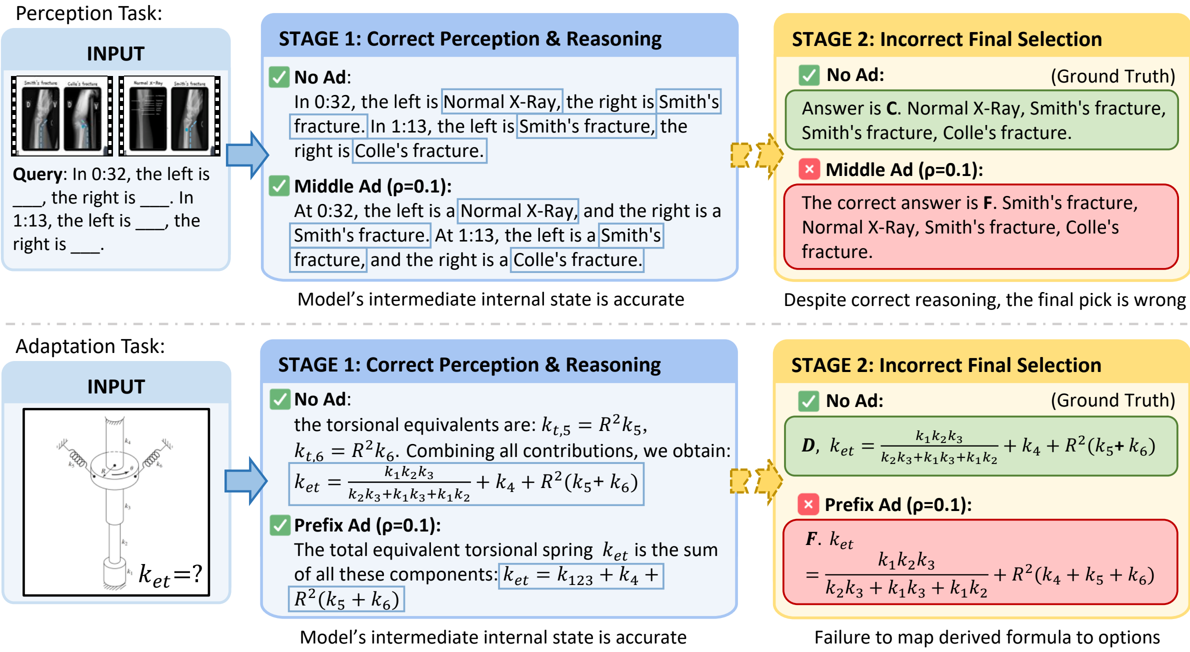


Figure 6: Case study of GPT-4o. Top: a sample from Perception task; Bottom: a sample from Adaptation task. In both cases, the model's perception and reasoning are correct, but it still selects the wrong final option.

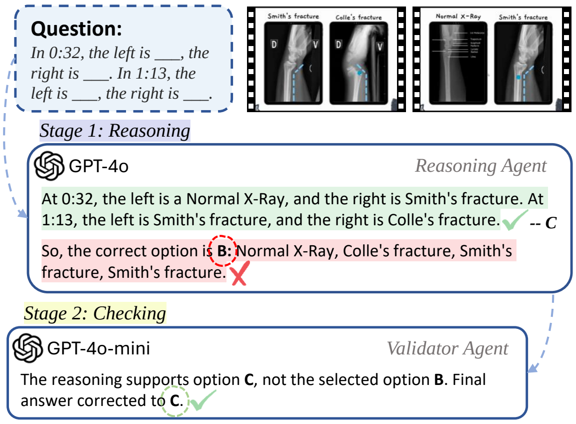


Figure 7: Two-agent framework. The Reasoning Agent (GPT-4o) generates an answer and explanation. The Validator Agent (GPT-4o-mini) checks their consistency, outputting a validated or revised answer.

Setting	GPT-4o	2-agent	Δ
Baseline (no ad)	0.640	0.647	+0.7% ↑
Prefix (ρ=0.10)	0.553	0.620	+6.7% ↑
Prefix (ρ=0.25)	0.547	0.593	+4.6% ↑
Middle (ρ=0.10)	0.597	0.620	+2.3% ↑
Middle (ρ=0.25)	0.600	0.627	+2.7% ↑
Suffix (ρ=0.10)	0.607	0.613	+0.7% ↑
Suffix (ρ=0.25)	0.600	0.613	+1.3% ↑
Average extra tokens (validator)			*+5.8%*

Table 3: Accuracy comparison between single-agent GPT-4o and our 2-agent framework. The last row reports the average token overhead.

tion, and candidates. For consistency with the baseline, we used the standard VideoMMMU prompt without any specialized prompt engineering. **Validator Agent:** GPT-4o-mini checks whether the candidate answer follows from the reasoning. If consistent, it accepts; if not, it requests revision.

Table 3 shows the framework improves accuracy across all ad settings, proving its effectiveness despite its lightweight design. Most notably, the performance recovery is most pronounced in the Prefix conditions, which originally suffered the most most severe degradation. However, this method is limited to correcting inconsistencies and cannot fix fundamentally flawed reasoning.

8 Conclusion

This work introduces Ads-VideoMMMU, the first benchmark dedicated to assessing VLM robustness against realistic ad interference. We uncover a distinct 'Prefix Penalty,' where initial ads cause the most severe degradation. While we offer a preliminary mitigation strategy, our findings primarily highlight a vulnerability in state-of-the-art models, emphasizing the urgent need to move beyond curated data and address the robustness challenges of in-the-wild video understanding.

Limitations

Due to computational resource constraints, our proposed mitigation strategy is restricted to a lightweight, inference-time framework. We did not conduct a more comprehensive exploration of training-based interventions, such as fine-tuning VLMs on ad-interrupted videos or designing specialized attention mechanisms to filter out distractors. While our two-agent approach effectively reduces "last-mile errors," more fundamental architectural defenses against semantic noise remain a subject for future research.

References

Meta AI. 2025. Llama 4 (multimodal) — llama 4 scout & maverick. https://ai.meta.com/blog/llama-4-multimodal-intelligence/. Model announcement, accessed 2025-11.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com. Model page / overview, accessed 2025-11.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Sriram Balasubramanian, Samyadeep Basu, and Soheil Feizi. 2025. A closer look at bias and chain-of-thought faithfulness of large (vision) language models. *arXiv preprint arXiv:2505.23945*.

Hao Cheng, Erjia Xiao, Jindong Gu, Le Yang, Jinhao Duan, Jize Zhang, Jiahang Cao, Kaidi Xu, and Renjing Xu. 2024. Unveiling typographic deceptions: Insights of the typographic vulnerability in large vision-language models. In *European Conference on Computer Vision*, pages 179–196. Springer.

Jan Clusmann, Dyke Ferber, Isabella C Wiest, Carolin V Schneider, Titus J Brinker, Sebastian Foersch, Daniel Truhn, and Jakob Nikolas Kather. 2025. Prompt injection attacks on vision language models in oncology. *Nature Communications*, 16(1):1239.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.

Google. 2025. Gemini 2.5 flash. https://ai.google.dev/gemini-api/docs/models. Model card, accessed 2025-11.

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

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xinwei Long, Kai Tian, Peng Xu, Guoli Jia, Jingxuan Li, Sa Yang, Yihua Shao, Kaiyan Zhang, Che Jiang, Hao Xu, and 1 others. 2025. Adsqa: Towards advertisement video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23396–23407.

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

OpenAI. 2024a. Gpt-4o. https://platform.openai.com/docs/models#gpt-4o. Model card, accessed 2025-11.

OpenAI. 2024b. Gpt-4o-mini. https://platform.openai.com/docs/models#gpt-4o-mini. Model card, accessed 2025-11.

Maan Qraitem, Nazia Tasnim, Piotr Teterwak, Kate Saenko, and Bryan A Plummer. 2024. Vision-llms can fool themselves with self-generated typographic attacks. *arXiv preprint arXiv:2402.00626*.

Jiachen Sun, Changsheng Wang, Jiongxiao Wang, Yiwei Zhang, and Chaowei Xiao. 2024. Safeguarding vision-language models against patched visual prompt injectors. *arXiv preprint arXiv:2405.10529*.

Zhijie Tan, Xu Chu, Weiping Li, and Tong Mo. 2024. Order matters: Exploring order sensitivity in multimodal large language models. *arXiv preprint arXiv:2410.16983*.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, and 1 others. 2025. Video understanding with large language models: A survey.

IEEE Transactions on Circuits and Systems for Video Technology.

Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. 2025. Identifying and mitigating position bias of multi-image vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10599–10609.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, and 1 others. 2025a. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967.

Zhaochen Wang, Bryan Hooi, Yiwei Wang, Ming-Hsuan Yang, Zi Huang, and Yujun Cai. 2025b. Text speaks louder than vision: Ascii art reveals textual biases in vision-language models. *arXiv preprint arXiv:2504.01589*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644.

Zhehao Zhang, Weijie Xu, Shixian Cui, and Chandan K Reddy. 2025. Distractor injection attacks on large reasoning models: Characterization and defense. *arXiv preprint arXiv:2510.16259*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv e-prints*, pages arXiv–2406.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Appendix

A.1 Detailed Ad Insertion Protocol

To simulate realistic viewing experiences while maintaining the integrity of the evaluation, we employ a precise temporal splicing strategy for injecting advertisement clips (V_{ad}) into the host videos (V_{host}). The insertion logic varies by position as follows:

Prefix (Pre-roll): We perform a standard concatenation at the video start. The advertisement is prepended to the host video, such that the last frame of the ad clip is immediately followed by the first frame of the original host video.

Middle (Mid-roll): We adopt a center-aligned insertion strategy. The host video is split at its temporal midpoint ($T_{mid} = T_{video}/2$). The advertisement clip is then injected between the two halves. This ensures the ad interrupts the narrative flow at the most central point.

Suffix (Post-roll): A unique constraint of the VideoMMMU dataset is that all videos contain a "Quiz Segment" (e.g., a static question slide) in the final seconds, which is crucial for the Adaptation tasks. Appending ads to the absolute end would displace or obscure this critical evaluation context. To address this, we define the insertion point at 2 seconds prior to the video's end ($t_{insert} = T_{video} - 2s$). The advertisement is inserted such that its first frame follows the frame at t_{insert}. This ensures the advertisement concludes before the final 2-second window, leaving the critical Quiz/Adaptation segment intact at the very end of the sequence.

A.2 Statistical Verification

A.2.1 Method Selection

We bypass standard ANOVA tests due to the violation of the **Independence assumption**. Our experimental design introduces inherent correlations: (1) *Repeated Measures*: the same models are evaluated across multiple conditions; (2) *Stimulus Dependence*: all models respond to the identical set of videos. To rigorously address these dependencies, we employ Linear Mixed Models (LMM) for conditional effects and Generalized Estimating Equations (GEE) for marginal population effects.

A.2.2 Linear Mixed Model (LMM)

The LMM framework extends linear regression by introducing random effects to capture hierarchical dependencies. We model the accuracy y_{mc} for model m under ad-insertion condition c as:

$$y_{mc} = \underbrace{\beta_0 + \sum_{k=1}^{K} \beta_k \mathbb{I}(c = k) +}_{\text{Fixed Effects}} \underbrace{u_m}_{\text{Random Effect}} + \epsilon_{mc} \quad (1)$$

where:

- β_0 is the global intercept, representing the baseline performance (No-Ad).
- β_k represents the fixed effect (performance drop) of the k-th ad condition (e.g., Prefix, Middle) relative to the baseline. $\mathbb{I}(\cdot)$ is the indicator function.
- $u_m \sim \mathcal{N}(0, \sigma_u^2)$ is the *random intercept* for model m. This term is crucial as it accounts for the intrinsic capability variance across different models (i.e., some models are inherently stronger), absorbing the model-level correlation.
- $\epsilon_{mc} \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is the independent and identically distributed (i.i.d.) residual error.

Hypothesis testing is performed on the fixed effect coefficients β_k to determine if the performance degradation is statistically significant ($H_0 : \beta_k = 0$).

A.2.3 Generalized Estimating Equations (GEE)

While LMM specifies the full distribution, GEE provides a robust estimate of population-averaged (marginal) effects by relying only on the first two moments (mean and variance). This is particularly advantageous when the exact correlation structure is unknown.

Let $\mathbf{y}_m$ be the response vector for model m, and $\boldsymbol{\mu}_m = E[\mathbf{y}_m]$ be the marginal mean linked to covariates $\mathbf{X}_m$ via $\boldsymbol{\mu}_m = \mathbf{X}_m\boldsymbol{\beta}$. We estimate $\boldsymbol{\beta}$ by solving the generalized quasi-likelihood score equation:

$$\sum_{m=1}^{M} \mathbf{D}_m^\top \mathbf{V}_m^{-1}(\mathbf{y}_m - \boldsymbol{\mu}_m) = \mathbf{0} \quad (2)$$

Here, the covariance matrix $\mathbf{V}_m$ is structured as:

$$\mathbf{V}_m = \phi \mathbf{A}_m^{1/2} \mathbf{R}(\alpha) \mathbf{A}_m^{1/2} \quad (3)$$

where $\mathbf{A}_m$ is a diagonal matrix of variances, ϕ is a dispersion parameter, and $\mathbf{R}(\alpha)$ is the *working correlation matrix* characterized by parameter α.

Robust Inference: A key property of GEE is its use of the Huber-White "sandwich" estimator for the covariance of $\hat{\boldsymbol{\beta}}$. This ensures that the standard errors (and thus p-values) remain consistent (asymptotically correct) even if the working correlation structure $\mathbf{R}(\alpha)$ is misspecified, providing a strong safeguard against Type I errors in our robustness analysis.

A.2.4 Results

Table 4 summarizes the regression results from both Linear Mixed Models (LMM) and Generalized Estimating Equations (GEE). Consistent across both statistical methods, we observe that advertisement injection significantly impairs VLM performance ($p < 0.05$ for all conditions).

A.3 Ad Detection

We report performance metrics in a One-vs-All manner for each category (Prefix, Middle, Suffix, No Ad).

1. **Recall.** We report the Recall (also known as Sensitivity) for each specific category. This measures the proportion of actual samples for a given category that were correctly identified by the model.

$$\text{Rec}_c = \frac{TP_c}{TP_c + FN_c}$$

2. **F1-Score.** The F1-Score is the harmonic mean of Precision and Recall for the specific category, treating it as the positive class and all others as negative.

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$

where $\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}$.

We provide the full prompt structure for the Ad Detection Failure experiment (Section 5.1) in Figure 8. To ensure rigorous evaluation, our instruction sensitizes the model to **narrative inconsistencies** alongside explicit advertisements. By directing the model to identify segments disrupting the video's flow, this task effectively probes *semantic interference*, capturing failures in filtering contextually irrelevant information.

A.4 The "Needle-in-a-Haystack" Experiment

To rigorously test the *Visual Primacy Effect* discussed in Section 5.2, we designed a controlled "Needle-in-a-Haystack" retrieval task. This experiment isolates the impact of visual position from semantic context.

Condition	Descriptive Mean Acc.	Linear Mixed Model (LMM) β (Diff)	S.E.	p-value	GEE (Robust) β (Diff)	S.E.	p-value
Baseline (No Ad)	0.572	–	–	–	–	–	–
Prefix Injection							
Ratio 0.10	0.532	-0.040	0.007	$< 0.001^{***}$	-0.040	0.008	$< 0.001^{***}$
Ratio 0.25	0.528	-0.044	0.007	$< 0.001^{***}$	-0.044	0.010	$< 0.001^{***}$
Middle Injection							
Ratio 0.10	0.538	-0.034	0.007	$< 0.001^{***}$	-0.034	0.006	$< 0.001^{***}$
Ratio 0.25	0.543	-0.030	0.007	$< 0.001^{***}$	-0.030	0.004	$< 0.001^{***}$
Suffix Injection							
Ratio 0.10	0.559	-0.014	0.007	0.038^{*}	-0.014	0.003	$< 0.001^{***}$
Ratio 0.25	0.557	-0.015	0.007	0.021^{*}	-0.015	0.004	$< 0.001^{***}$

Note: β represents the difference compared to the Baseline. Significance levels: $^{*}p < 0.05$, $^{***}p < 0.001$.

Table 4: **Statistical Analysis of Model Accuracy under Different Ad Injection Conditions.** The table presents the Mean Accuracy across all models, along with the estimated coefficients (β), Standard Errors (S.E.), and p-values from both Linear Mixed Models (LMM) and Generalized Estimating Equations (GEE). The Baseline represents the no-ad condition.

Prompt for Ad Detection Task

System Prompt: "You are an expert in video content analysis, specialized in detecting advertisements and content inconsistencies in videos."

User Instruction:
Watch this video carefully. Your task is to detect if there are any advertisements or segments that deviate from the main content.

Classification Criteria:

- **No_AD:** The video content is consistent; no advertisement or unrelated segment found.
- **Begin:** An advertisement or unrelated segment appears at the beginning of the video.
- **Middle:** An advertisement or unrelated segment appears in the middle of the video.
- **End:** An advertisement or unrelated segment appears at the end of the video.

Important:

1. Focus on detecting content that disrupts the main storyline or appears semantically irrelevant.
2. Answer with ONLY one of these four options: No_AD, Begin, Middle, or End.
3. Do not provide any explanation, just the classification label.

Figure 8: The prompt used for the Ad Detection task (Section 5.1). We incorporate semantic inconsistency ("deviate from main content") into the instruction to capture both commercial ads and irrelevant inserts.

A.4.1 Task Construction

The "haystack" consists of a sequence of $N = 48$ high-density images. Each image is a receipt-like table containing 30 rows of synthetic asset data (Asset ID, Item Name, Security Code, etc.), resulting in approximately 1,440 candidate entries per video context.

The "needle" is a specific target row within one of these tables. We systematically vary the position of the image containing the needle (e.g., placing it at index 0, 9, 18, ..., 45) while keeping the prompt constant. The model is tasked with retrieving the

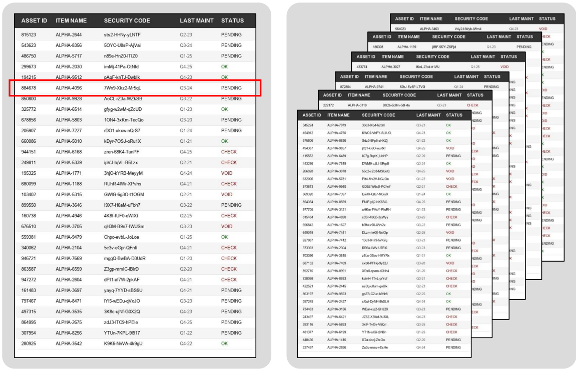

Target image (1/48) Distractors (47/48)

Figure 9: Overview of the Visual Primacy "Needle-in-a-Haystack" Experiment. The task involves retrieving specific attributes for a target item (e.g., "ALPHA-4096", highlighted in red) hidden within a sequence of 48 high-density asset tables. We systematically vary the position of the target image (the "needle") among 47 distractor images to probe the model's retrieval accuracy across different temporal positions.

Prompt for Needle-in-a-Haystack

```
User Instruction:
I have uploaded 48 images representing a dense
asset database.  Please locate the specific
item with ITEM NAME: 'ALPHA-4096'.
Once found, extract and list ALL its details
from that row: ASSET ID, SECURITY CODE, LAST
MAINT, and STATUS.
Output format: JSON.
```

```
Model Response Expectation (JSON Example):
{
    "ITEM NAME": "ALPHA-4096",
    "ASSET ID": "884678",
    "SECURITY CODE": "7Wn9-Xkz2-Mr5qL",
    "LAST MAINT": "Q3-24",
    "STATUS": "PENDING"
}
```

Figure 10: The prompt and expected output format for the Needle-in-a-Haystack experiment. The model must retrieve specific attributes (Security Code, Status, etc.) corresponding to the queried Item Name from a sequence of 48 dense images.

full details of a specific item given its `ITEM NAME`.

A.4.2 Prompt and Ground Truth

We provide the exact prompt used for this evaluation in Figure 10. For the specific case shown in the figure, the ground truth target is defined as:

- **Target Query:** `ALPHA-4096`
- **Expected Output (Ground Truth):**
  - `SECURITY CODE`: "7Wn9-Xkz2-Mr5qL"
  - `LAST MAINT`: "Q3-24"
  - `STATUS`: "PENDING"

Success is measured by an exact match of the JSON values. Through experiments, we discovered a position bias in GPT-4o: the model tends to focus on visually prominent content.

A.5 Detailed Case Analysis

Figures 11 and 12 present examples of the "Last-Mile Error" observed in GPT-4o and Claude-3.5. In these instances, the models generate correct intermediate reasoning but select incorrect options under ad interference.

In the medical perception task (Figure 11, top), the model's textual response accurately identifies the fracture types , yet the final selection contradicts this diagnosis. Similar discrepancies appear in STEM domains (Figure 12). For the RC circuit analysis, the model explicitly states the correct component values (9V, 12Ω, 3μF) but selects an option with a different sequence (9, 3, 12). In the algorithmic tracing task, the model derives a value of 3 but selects an option ending in 1. These cases indicate a dissociation between the reasoning process and the final selection in the presence of advertisements.

A.6 Two-Agent Framework

As proposed in Section 7 (A Two-Agent Consistency Framework), our method employs a lightweight Validator Agent to mitigate reasoning fragility.

Reasoning Agent: We strictly adhere to the standard VideoMMMU evaluation protocol. The Reasoning Agent receives the video frames, question, and options, generating a chain-of-thought response without any prompt engineering modification.

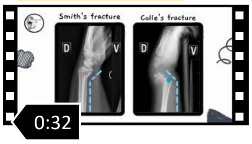
Validator Agent: The Validator Agent operates purely on the textual output of the Reasoning Agent. It does not re-process the video visual tokens, ensuring computational efficiency. Its core function is to verify *self-consistency*: checking if the generated reasoning logically supports the parsed prediction.

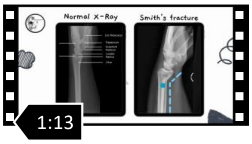
Figure 13 presents the exact prompt template used by the Validator Agent. The system is designed to correct "Last-Mile Errors" by allowing two decision paths:

1. **ACCEPT:** The reasoning aligns with the parsed prediction.

2. **CORRECTED:** The reasoning explicitly supports a different option (indicating a selection error).

Perception Task:





In 0:32, the left is ___, the right is ___. In 1:13, the left is ___, the right is ___.

A. Smith's fracture, Normal X-Ray, Colle's fracture, Smith's fracture
B. Normal X-Ray, Colle's fracture, Smith's fracture, Smith's fracture
C. Normal X-Ray, Smith's fracture, Smith's fracture, Colle's fracture
D. Smith's fracture, Smith's fracture, Colle's fracture, Normal X-Ray
E. Smith's fracture, Colle's fracture, Smith's fracture, Normal X-Ray
F. Smith's fracture, Normal X-Ray, Smith's fracture, Colle's fracture
G. Colle's fracture, Smith's fracture, Normal X-Ray, Smith's fracture
H. Colle's fracture, Normal X-Ray, Smith's fracture, Smith's fracture
I. Colle's fracture, Smith's fracture, Smith's fracture, Normal X-Ray
J. Smith's fracture, Colle's fracture, Normal X-Ray, Smith's fracture

Baseline (no ad)

In 0:32, the left is Normal X-Ray, the right is Smith's fracture. In 1:13, the left is Smith's fracture, the right is Colle's fracture.
Answer: **C**. Normal X-Ray, Smith's fracture, Smith's fracture, Colle's fracture.

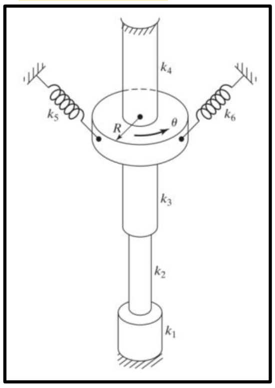
Prefix Ad (ρ=0.1)

At 0:32, the left is a Normal X-Ray, and the right is Smith's fracture. At 1:13, the left is Smith's fracture, and the right is Colle's fracture.
So, the correct option is **B:** Normal X-Ray, Colle's fracture, Smith's fracture, Smith's fracture.

Middle Ad (ρ=0.1)

At 0:32, the left is a Normal X-Ray, and the right is a Smith's fracture. At 1:13, the left is a Smith's fracture, and the right is a Colle's fracture.
The correct answer is **F.** Smith's fracture, Normal X-Ray, Smith's fracture, Colle's fracture.

Adaption Task:



Find the equivalent torsional spring constant of the system shown in <image 1>. Assume that k1, k2, k3, and k4 are torsional and k5 and k6 are linear spring constants.

A. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_4 + k_5 + k_6 + R^2$

B. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_6 + R^2(k_4 + k_5)$

C. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_4 + k_5 + R^2k_6$

D. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_4 + R^2(k_5 + k_6)$

E. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_6 + k_5 + R^2k_4$

F. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + R^2(k_4 + k_5 + k_6)$

G. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_5 + R^2(k_4 + k_6)$

Baseline (no ad)

Torsional springs k_1, k_2, k_3: These are in series, so their equivalent spring constant k_{123} is given by: $k_{123} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2}$ Torsional spring k_4: This spring is in parallel with the equivalent spring constant of k_1, k_2, and k_3, so we simply add it: $k_{123} + k_4$. Linear springs k_5 and k_6**:** These are linear springs attached at a radius R, so they must be converted to equivalent torsional spring constants. Therefore, the torsional equivalents are: $k_{t,5} = R^2k_5$, $k_{t,6} = R^2k_6$. Combining all contributions, we obtain: $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_4 + R^2(k_5 + k_6)$

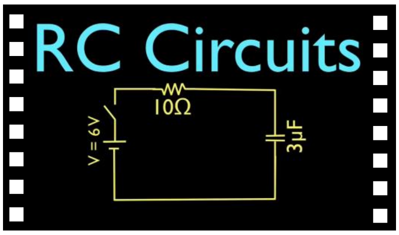
Thus, the correct choice is: ***D***, $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_4 + R^2(k_5 + k_6)$

Baseline (no ad)

we need to consider both the torsional and linear spring constants and their configurations:
k_1, k_2, k_3, and k_4 are torsional springs. k_5 and k_6 are linear springs attached to the edge of a disk, converting linear to torsional with a radius R. The equivalent spring constant for the series of k_1, k_2, k_3 is calculated as: $k_{123} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2}$
The linear springs k_5 and k_6 are converted to torsional by multiplying by R^2 Thus, their contribution is $R^2(k_5 + k_6)$ .The total equivalent torsional spring k_{et} is the sum of all these components:$k_{et} = k_{123} + k_4 + R^2(k_5 + k_6)$.
Thus, the correct answer is: ***F***. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + R^2(k_4 + k_5 + k_6)$

Figure 11: Extended examples of GPT-4o where the model generates correct reasoning for both visual and logical tasks but selects the wrong final option due to ad interference.

Perception Task:



In the circuit diagram shown, the voltage source is ______(1) volts, connected in series with a resistor of ______(2) ohms and a capacitor of ______(3) microfarads.

```
A. 12, 9, 3     B. 9, 3, 12
C. 3, 12, 9     D. 9, 12, 3
E. 12, 3, 9     F. 9, 12, 2
G. 3, 9, 12     H. 12, 9, 2
I. 9, 3, 2      J. 2, 12, 3
```

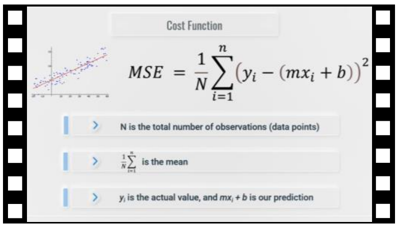
Baseline (no ad)

Looking at the RC circuit diagram shown in the images, the voltage source is 9V, connected in series with a 12Ω resistor and a 3μF capacitor. Looking at the given options, this combination corresponds to option D: 9, 12, 3. Therefore, the answer is D.

Prefix Ad (ρ=0.25)

Looking at the RC circuit diagram shown in the images, I can see that: - The voltage source is 9 volts (V = 9V) - The resistor value is 12 ohms (R = 12Ω) - The capacitor value is 3 microfarads (C = 3μF).
Comparing these values to the given options, the answer is **B:** 9, 3, 12.

Comprehension Task:



Looking at the scatter plot and formula components shown in the video, if a prediction mx_1 + b = 3 while the actual value y_1 = 5, what would be the contribution of this single point to the summation part of the MSE (before dividing by N)?

```
A. 2      B. 4
C. 8      D. 16
E. 6      F. 9
G. -2     H. 1
I. 25     J. 10
```

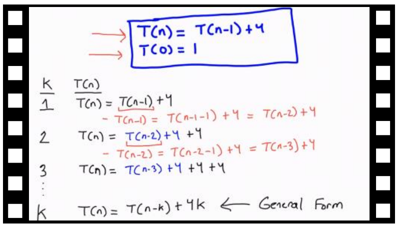
Baseline (no ad)

Let me help solve this step by step: 1) From the MSE formula shown: MSE = (1/N)∑(yi - (mxi + b))² 2) We're asked about one point's contribution to the summation (before dividing by N) 3) Given: - Predicted value (mxi + b) = 3 - Actual value (yi) = 5 4) For this single point, we need to calculate: (yi - (mxi + b))² = (5 - 3)² = (2)² = 4.
Therefore, the contribution of this single point to the summation (before dividing by N) is 4. **B**

Prefix Ad (ρ=0.25)

D Let me explain why: For MSE, we take (y_i - prediction)^2 for each point. In this case: - Actual value (y_1) = 5 - Prediction (mx_1 + b) = 3 - Therefore, (y_1 - prediction) = 5 - 3 = 2 The MSE formula squares this difference: (2)^2 = 4 This is just for one point in the summation, before dividing by N. The answer is **D (16).**

Comprehension Task:



In the given iterative solution, at step k=2, T(n) is expressed as T(n-______(1)) + (2), and by step k=3, the equation becomes T(n) = T(n-(3)) + 4 + 4 + 4.

```
A. 2, 8, 3     B. 1, 4, 2
C. 3, 4, 1     D. 2, 4, 3
E. 1, 8, 2     F. 3, 8, 2
G. 2, 4, 1     H. 1, 4, 3
I. 3, 4, 2     J. 2, 8, 1
```

Baseline (no ad)

Looking at the iterative solution shown in the images, at k=2, T(n) = T(n-2) + 4, and at k=3, T(n) = T(n-3) + 4 + 4 + 4. Therefore, the blanks should be filled with: (1) 2 (2) 4 (3) 3 The answer that matches these values is **D**.

Prefix Ad (ρ=0.25)

Looking at the iterative solution shown in the images, at k=2, T(n) = T(n-2) + 4, and at k=3, T(n) = T(n-3) + 4 + 4 + 4. Comparing this with the blanks in the question: (1) should be 2 (from T(n-2)) (2) should be 4 (the value added at k=2) (3) should be 3 (from T(n-3)) Looking at the options, G matches these values: 2, 4, 1 Therefore, the answer is **G.**

Figure 12: Extended examples of Claude-3.5-sonnet where the model generates correct reasoning for both visual and logical tasks but selects the wrong final option due to ad interference.

Prompt Template for the Validator Agent

System Prompt:
You are a Checker Agent in a two-agent video QA system.
You do NOT redo video reasoning from scratch. You only: 1) Read the QUESTION and OPTIONS. 2) Read the MODEL_RESPONSE (reasoning + conclusion). 3) Read the PARSED_PRED (the option letter extracted by a parser). 4) Decide whether the reasoning actually supports that option.
Your job is to check self-consistency between reasoning and the final answer.
You must choose one of two statuses:

- **ACCEPT:** The reasoning clearly supports PARSED_PRED. In this case, final_answer MUST be exactly the same as PARSED_PRED.
- **CORRECTED:** The reasoning clearly supports a different option. In this case, final_answer MUST be the option letter that is best supported by the reasoning, and MUST be different from PARSED_PRED.

Do NOT use any other status keyword.

User Input Template:
[QUESTION]
{question}

[OPTIONS]
{options_block}

[MODEL_RESPONSE]
{model_response}

[PARSED_PRED]
The parser extracted the model's chosen option as: {parsed_pred}

[OUTPUT FORMAT]
You MUST output exactly:
note=one short English sentence (<= 50 words)
final_answer=ONE_LETTER_FROM_OPTIONS
status=ACCEPT or CORRECTED
Rules:

- If status=ACCEPT, final_answer MUST equal {parsed_pred}.
- If status=CORRECTED, final_answer MUST be different from {parsed_pred}.

Figure 13: The prompt used by the Validator Agent (GPT-4o-mini). The input fields (in curly braces) are dynamically populated with the Reasoning Agent's output. This prompt enforces a strict consistency check between the generated rationale and the final selected option.