Failures to Surface Harmful Contents in Video Large Language Models

Yuxin Cao[1], **Wei Song**[2,3], **Derui Wang**[3], **Jingling Xue**[2], **Jin Song Dong**[1]

[1]National University of Singapore, Singapore
[2]University of New South Wales, Australia
[3]CSIRO's Data61, Australia

Abstract

Video Large Language Models (VideoLLMs) are increasingly deployed on numerous critical applications, where users rely on auto-generated summaries while casually skimming the video stream. We show that this interaction hides a critical safety gap: if harmful content is embedded in a video, either as full-frame inserts or as small corner patches, state-of-the-art VideoLLMs rarely mention the harmful content in the output, despite its clear visibility to human viewers. A root-cause analysis reveals three compounding design flaws: (1) insufficient temporal coverage resulting from the sparse, uniformly spaced frame sampling used by most leading VideoLLMs, (2) spatial information loss introduced by aggressive token downsampling within sampled frames, and (3) encoder-decoder disconnection, whereby visual cues are only weakly utilized during text generation. Leveraging these insights, we craft three zero-query black-box attacks, aligning with these flaws in the processing pipeline. Our large-scale evaluation across five leading VideoLLMs shows that the harmfulness omission rate exceeds 90% in most cases. Even when harmful content is clearly present in all frames, these models consistently fail to identify it. These results underscore a fundamental vulnerability in current VideoLLMs' designs and highlight the urgent need for sampling strategies, token compression, and decoding mechanisms that guarantee semantic coverage rather than speed alone.

Introduction

Video Large Language Models (VideoLLMs) have recently become state-of-the-art engines for video understanding (Zhao et al. 2023; Tang et al. 2025b; Weng et al. 2024). They distill high-level semantics from diverse footage, such as classroom lectures, tutorials, news segments, sports highlights, entertainment shows, surveillance clips, and more, then generate concise summaries or detailed textual interpretations. By condensing lengthy footage into concise textual summaries, VideoLLMs enable viewers to skim the video casually while relying on the generated text to grasp its main ideas. This new way of video consumption markedly improves accessibility and eases cognitive load, making VideoLLMs indispensable for students, professionals, content moderators, and general users (Qian et al. 2024).

This hybrid "watch-and-read" video viewing style concentrates semantic trust in VideoLLMs' outputs, as users rely on the textual summaries to flag anything harmful or dangerous that a quick visual skim might miss in a video. However, VideoLLMs often omit these cues from their summaries, leaving viewers with no warning and leading them to assume the video is harmless even when harmful frames are present. Such omissions create a semantic blind spot, wherein harmful content remains visible in the video yet absent from VideoLLMs' summary, allowing the video to slip by unchallenged and spread unchecked across platforms. Understanding the mechanisms behind this blind spot is therefore crucial and motivates a systematic study into VideoLLMs' vulnerability to omission, *i.e.,* examining how often clearly visible harmful content remains unacknowledged in their summaries.

To investigate this issue, we dissect VideoLLMs' processing pipeline and identify three structural flaws that give rise to this semantic blind spot. First, VideoLLMs typically adopt sparse uniform frame sampling to keep computation tractable. This leaves large portions of the video unexamined, allowing attackers to insert harmful content in unsampled intervals without detection (Li et al. 2025). Second, the retained frames often undergo aggressive spatial downsampling to trim visual tokens (Li et al. 2024), which leads to the loss of fine-grained information from small regions, such as a small corner patch. Third, the cross-modal decoder downplays visual evidence: linguistic priors dominate the attention budget, so cues that do survive tokenization may still be ignored at generation time (Fu et al. 2025). Combined, these three structural flaws, temporal sparse sampling, spatial downsampling, and modality fusion imbalance, collectively account for VideoLLMs' consistent omission of harmful content. Motivated by these findings, we craft three zero-query, black-box omission attacks, each exploiting one or more of these flaws:

Frame-Replacement Attack (FRA): We replace a segment of the original video with a harmful video clip at a random temporal position. Due to the large interval of sparse uniform sampling, the inserted segment is skipped entirely or nearly entirely during frame selection.

Picture-in-Picture Attack (PPA): We insert a small harmful patch into the corner of each frame. Due to spatial downsampling, information in peripheral regions (*e.g.,* corners) is often lost, and any harmful signals that survive are treated as high-frequency noise and suppressed.

Transparent-Overlay Attack (TOA): We overlay a transparent harmful video clip across each frame. While the visual encoder may capture the harmful signal, it is often overridden by strong linguistic priors during fusion and thus omitted in the final response due to unbalanced modality fusion.

To quantify the severity of the omission vulnerability, we comprehensively test the three proposed attacks against five representative VideoLLMs, LLaVA-Video-7B-Qwen2 (Zhang et al. 2024b), LLaVA-NeXT-Video-7B-DPO (Zhang et al. 2024a), LLaVA-NeXT-Video-32B-Qwen (Zhang et al. 2024a), VideoLLaMA2 (Cheng et al. 2024), and ShareGPT4Video (Chen et al. 2024a), using test clips that embed three types of harmful content: *violence*, *crime*, and *pornography*[1]. Using the unified metric of Harmfulness Omission Rate (HOR), the percentage of harmful clips that pass unmentioned, we find that, with hyperparameters ensuring the harmful content remains clearly visible and semantically recognizable to human viewers, the average HORs remain strikingly high: 99%, 91%, 100% for violence, crime and pornography content under FRA; 98%, 87%, 76% under PPA; and 93%, 82%, 93% under TOA. This phenomenon reflects the fact that most injected frames evade sampling due to temporal sparsity, and harmful content in corners is largely discarded during spatial downsampling. Even when some visual tokens survive, they are often suppressed by unbalanced modality fusion, a failure that also occurs in TOA, where transparent yet clearly visible harmful content is added to every frame. These findings reveal a fundamental weakness in state-of-the-art VideoLLMs and underscore the need for denser temporal sampling, finer spatial token retention, and a more balanced cross-modal fusion to achieve reliable safety against harmful content. Our code is available at https://github.com/yuxincao22/VideoLLM-Failures.

Contributions. Our work makes three major contributions:

- We are the first to systematically analyze the safety of VideoLLMs and uncover a novel omission vulnerability: harmful content that is clearly visible in the video can pass unmentioned in the generated textual summaries.
- Drawing on the root causes, we identify three structural flaws in contemporary VideoLLMs, including temporal sparse sampling, token under-sampling, and modality fusion imbalance, and we tailor three zero-query black-box attacks, frame replacement, picture-in-picture, and transparent overlay, that effectively exploit these flaws.
- We comprehensive test these three attacks against five representative VideoLLMs with three types of harmful videos, and the results highlight the severity of the vulnerability and underscore the urgent need for VideoLLMs to advance their design.

Background

As shown in Figure 1, a VideoLLM takes a video and a text prompt as input, and generates a textual response that reflects its semantic interpretation of the video based on the prompt. This is typically achieved through three main components: a visual encoder, a projector, and a pretrained LLM.

[1]This paper contains content that is offensive.

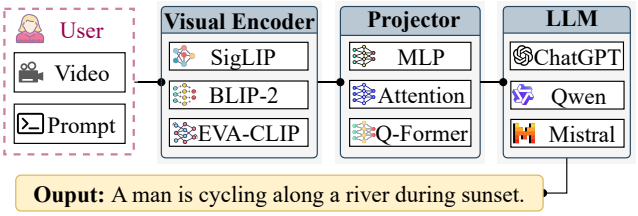


Figure 1: Pipeline of a typical VideoLLM.

Pretrained on large-scale image-text datasets, the visual encoder, such as SigLIP (Zhai et al. 2023), BLIP-2 (Li et al. 2023) and EVA-CLIP (Fang et al. 2023), extracts visual embeddings from a uniformly sampled subset of frames from the input video. These embeddings are then mapped by the projector into the same embedding space as the language input tokens. Projectors are typically implemented using Multi-Layer Perceptrons, cross-attention modules (Vaswani et al. 2017), or Q-Formers (Li et al. 2023). The LLM serves as the core reasoning engine. It receives a concatenated sequence of projected visual tokens and text tokens, and produces the textual output. Most VideoLLMs use instruction-tuned models such as LLaMA (Touvron et al. 2023), Vicuna (Chiang et al. 2023) and Qwen (Bai et al. 2023) as their backbone. This unified design enables VideoLLMs to jointly process visual and textual inputs, supporting diverse video understanding tasks including video captioning and question answering.

Formally, given a video $\mathcal{V} = \{f_1, f_2, \ldots, f_T\}$ with T frames, where each frame $f_t \in \mathbb{R}^{H \times W \times C}$, and H, W, C denotes the height, width and channel number of the frame, respectively, VideoLLMs, constrained by computation resources, first sample a subset $\mathcal{V}_s$ uniformly from $\mathcal{V}$:

$$\mathcal{V}_s = \{f_{t_1}, f_{t_2}, \ldots, f_{t_N}\}, \quad \text{where } N \ll T. \tag{1}$$

With $\mathcal{V}_s$, the visual encoder ϕ_v encodes each frame $f_{t_i} \in \mathcal{V}_s$ to extract P visual tokens $\phi_v(f_{t_i})$. These tokens are then downsampled to obtain a reduced set of P' tokens per frame ($P' < P$). The resulting output $\mathbf{v}_i \in \mathbb{R}^{P' \times d_v}$, with d_v denoting the embedding dimension of visual tokens, is aggregated to form the complete set of visual features:

$$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\} \in \mathbb{R}^{N \cdot P' \times d_v}. \tag{2}$$

To align visual tokens with text embeddings, they are passed through a projector ϕ_p, which maps them into the LLM token space of dimension d_t:

$$\mathbf{V}' = \phi_p(\mathbf{V}) \in \mathbb{R}^{N \cdot P' \times d_t}. \tag{3}$$

At the same time, the user-provided textual prompt is tokenized as a sequence of tokens: $\mathcal{Q} = \{w_1, w_2, \ldots, w_L\}$, where L is the token length of the prompt. These tokens are then embedded by the language encoder ϕ_q:

$$\mathbf{Q} = \phi_q(\mathcal{Q}) \in \mathbb{R}^{L \times d_t}. \tag{4}$$

Finally, the projected visual tokens $\mathbf{V}'$ and text embeddings $\mathbf{Q}$ are concatenated into a unified sequence $\mathbf{Z} = [\mathbf{V}'; \mathbf{Q}] \in \mathbb{R}^{(N \cdot P' + L) \times d_t}$, which will be fed into the LLM $\mathcal{F}$ and produce the final textual output $\hat{\mathcal{Y}} = \mathcal{F}(\mathbf{Z})$.

Related Work

Video Large Language Models. Early progress in multimodal LLMs (MLLMs), such as Flamingo (Alayrac et al. 2022) and BLIP-2 (Li et al. 2023), shows that pairing LLMs with visual encoders can excel at image-based tasks such as captioning and visual question answering (Radford et al. 2018; Touvron et al. 2023). The same demand for rich, multimodal reasoning now extends to the temporal domain, spurring the rise of VideoLLMs that aim to interpret dynamic, semantically complex video content. To cope with the higher spatio-temporal complexity, modern VideoLLMs extract spatial-temporal features from video frames, align them with language embeddings, and feed the combined embeddings into a pretrained LLM (Li et al. 2024). Since processing full videos is prohibitively expensive on GPU memory and compute limits, most existing systems (*e.g.,* Video-LLaMA2 (Cheng et al. 2024), InternVL2.5 (Chen et al. 2024c) and NVILA (Liu et al. 2025)) resort to sparse uniform sampling. ViLaMP (Cheng et al. 2025) suggests that sampling 16 frames for short videos and 32 for long ones provides a good trade-off between efficiency and performance. However, fixed low sampling rate regardless of video length results in uneven temporal spacing (the interval grows with video duration), and causes critical segments to be skipped, leaving large portions of the video unexamined. A few models, such as ShareGPT4Video (Chen et al. 2024a), VideoAgent (Fan et al. 2024) and AKS (Tang et al. 2025a), further apply key frame selection, yet the final number of frames remains small.

Moreover, token-level compression techniques are also widely used. Typically, the visual tokens are downsampled using a 2×2 bilinear interpolation in LLaVA-OneVision (Li et al. 2024) and VideoLLaMA3 (Zhang et al. 2025a), or average pooling in LLaVA-Video (Zhang et al. 2024b). Some other models reduce visual tokens through various compression strategies. For instance, LLaMA-VID (Li, Wang, and Jia 2024) fixes the number of tokens per frame to two, NVILA (Liu et al. 2025) scales up spatial and temporal resolution before pooling, and Chat-UniVi (Jin et al. 2024) performs k-nearest-neighbor based clustering to reduce redundancy. However, such token compression strategies result in extremely limited information per frame, leading to the loss of fine-grained visual details. More details of existing mainstream VideoLLMs are provided in Appendix.

MLLM Safety. The safety of MLLMs has become a pressing concern for surveillance, content moderation, and educational applications, where models must reliably recognize and react to harmful material such as violence, nudity, or abuse. Recent studies (Fu et al. 2025) reveal that image MLLMs often underutilize visual features during decoding. Even when meaningful signals are extracted by the visual encoder, the fusion and decoding stages tend to favor linguistic priors. Despite growing attention to similar safety risks in image MLLMs (Liu et al. 2024; Ying et al. 2024) and generative video models (Wang et al. 2024; Chen et al. 2024b), safety vulnerabilities in VideoLLMs remain largely underexplored. To bridge this gap, we systematically characterize omission failures in VideoLLMs and introduce corresponding attacks that expose fundamental flaws in their design.

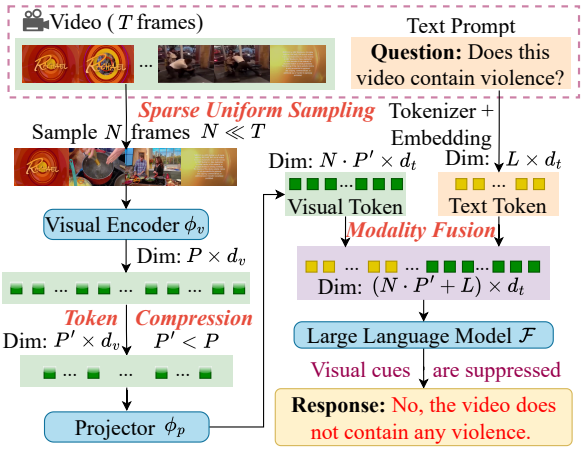


Figure 2: Inherent design flaws in VideoLLMs.

Analyses

Recent advances in VideoLLMs have enabled impressive performance across a wide range of video understanding tasks. However, their ability to handle safety-critical content remains largely unexamined. In this paper, we uncover three inherent design flaws in current VideoLLMs (illustrated in Figure 2) that allow harmful content to pass undetected, and therefore unreported in their textual outputs.

Flaw 1: Sparse Uniform Sampling. To conserve computation, most current VideoLLMs uniformly sample only a few frames (*e.g.,* 8, 16 or 32) from a video, leaving most of the segment unchecked. Even when a sampled frame does contain harmful content, it usually differs sharply from neighboring frames; this abrupt, high-frequency signal is dampened by frequency aliasing, so its semantics may never reach the model's output. This sampling mechanism employed by VideoLLMs leaves broad temporal gaps that adversaries can exploit, allowing harmful segments to slip past detection.

Flaw 2: Token Under-Sampling. Modern VideoLLMs inherit the input token limit of their host LLMs. For example, GPT-4 allows at most 8,192 tokens per input (Achiam et al. 2023). Since this token budget is shared between visual tokens and textual tokens, VideoLLMs must compress the tokens of each frame to meet the token budget limit. Formally, given a video of N sampled frames and per-frame token number P, the total token number should satisfy:

$$N \cdot P + L \leq B \tag{5}$$

where B denotes the LLM's input token limit. Therefore, the token number is reduced to P' ($P' < P$) through token compression. Many recent works adopt simple downsampling techniques, such as bilinear interpolation (Zhang et al. 2025a; Li et al. 2024) or average pooling (Zhang et al. 2024b), which aggregates visual tokens on a 2D spatial token grid. For example, an original 14×14 token grid is downsampled to 7×7, retaining only 25% of the spatial tokens.

While effective in reducing token count, this compression process inevitably leads to the loss of local spatial details, especially from peripheral or low-saliency regions such as small objectionable content in a corner. The influence of

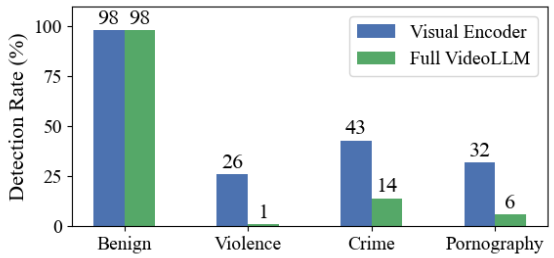


Figure 3: Comparison in harmful video detection.

such harmful patches becomes significantly weakened after token compression and may not survive downstream processing. Moreover, harmful patches often introduce sharp local changes in otherwise smooth regions, manifesting as high-frequency signals. Since token compression acts as a low-pass filter, these high-frequency components are suppressed or diffused across multiple tokens, weakening their influence and leading to spatial aliasing. As a result, the harmful content is unlikely to be retained in the final visual representation.

Flaw 3: Modality Fusion Imbalance. After projection into the language model's embedding space, visual tokens are often underutilized during decoding. As a result, the LLM tends to prioritize textual information while downplaying or even ignoring signals from the visual encoder, preventing harmful cues from being reflected in the final response. Prior studies (Fu et al. 2025) show that the standalone visual encoder surpasses the fully fused image-text model on vision-centric benchmarks, underscoring a structural imbalance where visual information loses influence after fusion and barely shapes the final representation. The problem persists in existing VideoLLMs, which reuse image-based encoders to generate visual tokens. Even when these encoders flag harmful content in the sampled frames, their signals are diminished during decoding, hindering the model from faithfully reporting it.

To validate this, we conduct a comparative experiment using LLaVA-Video-7B-Qwen2 and its underlying visual encoder, SigLIP. Following (Fu et al. 2025), we examine the effectiveness of the visual encoder through a visual probing strategy. Specifically, we construct evaluation videos by inserting harmful content into benign source videos (details in the next section) and randomly sample 100 benign and 100 harmful examples for each of three categories: *violence*, *crime*, and *pornography*. For each video, we examine the binary classification accuracy of both the standalone visual encoder and the full VideoLLM under identical inputs. Figure 3 reports the proportion of correctly identified videos for each category. For benign videos, both the visual encoder and the full VideoLLM achieve comparably high detection rates. However, for the three harmful categories, the full VideoLLM exhibits a significant performance drop compared to the visual encoder. This discrepancy provides concrete empirical support for this flaw, confirming that modality fusion does suppress visual signals even when they are preserved at the visual encoder level.

Attack Approaches

The architecture-level flaws mentioned above significantly undermine the reliability and safety of VideoLLMs in security-critical applications. Exploiting these flaws, we design three attacks targeting current VideoLLMs by inserting harmful content into videos in three distinct ways, each intended to make VideoLLMs omit the harmful content in their outputs.

Threat Model

We assume a strict zero-query black-box setting: the adversary has no knowledge of the targeting VideoLLM's internals, such as architecture, weights, training data, temporal sampling rate, token compression strategy, or modality fusion scheme, and cannot repeatedly query the model for optimization. The only prior is the knowledge of the three architectural flaws identified previously. The adversary may insert self-sourced harmful clips, but these must remain visible to a human *i.e.,* not single-frame flashes or imperceptible perturbations, so that any detection failure reflects a true omission. This zero-query, training-free attacking setup enables efficient, real-time deployment without per-video adaptation.

Three Attacks

We devise the following three attacks, each exploiting one or more flaws. Figure 4 summarizes these attacks.

Frame-Replacement Attack (FRA $\rightarrow$ Flaws 1, 3). This attack replaces a segment of the original video with a harmful video clip at a randomly chosen position. Specifically, we select a random insertion point and overwrite the subsequent t_r seconds ($t_r > 1$) with a preselected harmful video clip. Since VideoLLMs employ the sparse uniform sampling, a short replacement window t_r is seldom sampled. For instance, in a 2-minute video at 30 FPS, *i.e.,* 3,600 frames, taking only 16 evenly spaced frames gives a stride of 8 seconds (240 frames). A 4-second harmful clip can therefore fit entirely between two sampled frames, leaving the model with no evidence of it, even though human viewers see this harmful segment clearly.

Picture-in-Picture Attack (PPA $\rightarrow$ Flaws 2, 3). We particularly embed a harmful clip in a fixed Picture-in-Picture (PiP) region within each frame, *e.g.,* the bottom-right corner, while the rest of the frame remains unchanged. The PiP region occupies $\eta H \times \eta W$ pixels of each frame, where $\eta \in (0, 1)$ is chosen not too small to ensure the malicious content is visible to humans. Because VideoLLMs compress the tokens of each frame to fit the token budget, small peripheral regions are often discarded, failing to influence the model's output despite being clearly visible.

Transparent-Overlay Attack (TOA $\rightarrow$ Flaw 3). To conduct this attack, we resize the harmful video clip to match the original video's resolution, loop it if shorter, and blend it into every frame with a fixed opacity parameter $\alpha \in (0, 1)$. In addition, α is set large enough to make the overlaid harmful content clearly visible to humans. Although this guarantees that every sampled frame carries the malicious signal, the modality fusion imbalance can still suppress these visual cues, causing the VideoLLM to omit them in its textual response—a failure mode shared with FRA and PPA whenever their harmful segments are sampled.

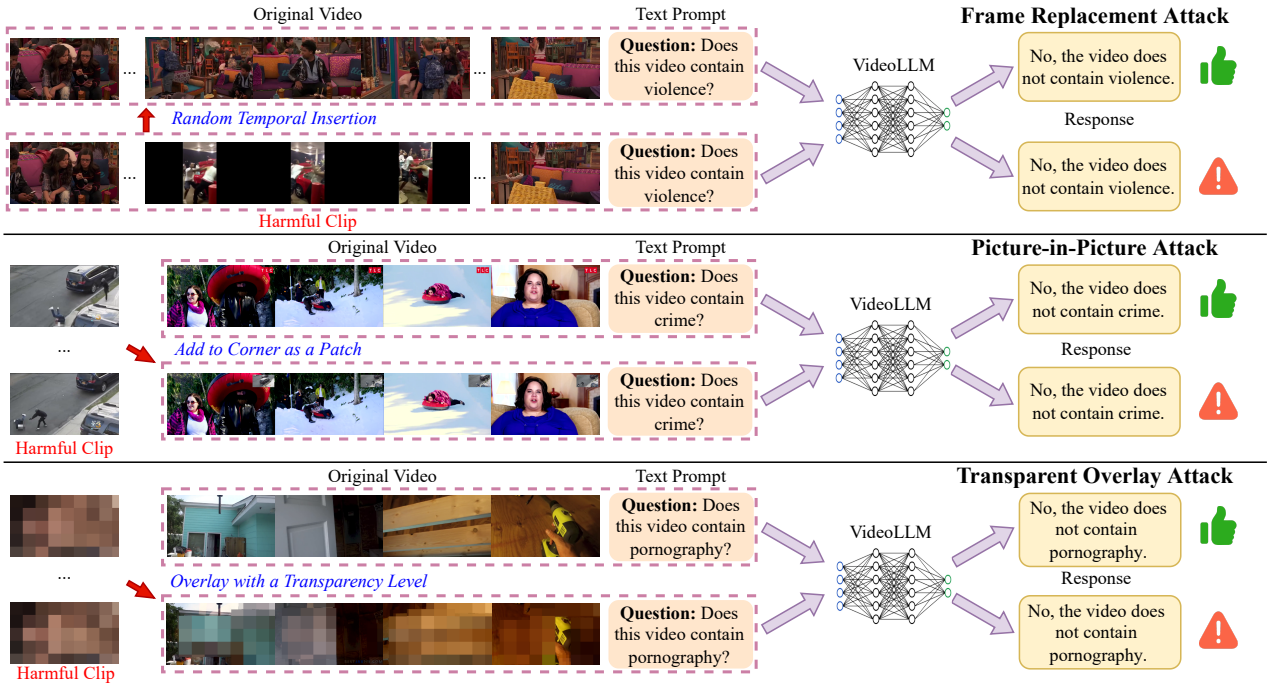


Figure 4: Overview of three proposed attacks which exploit VideoLLMs' design flaws.

Attack	Model	Harmful Category			Avg
		Violence	Crime	Pornography	
FRA ($t_r = 4$)	L-7B	100	85	100	95
	LN-7B	100	100	100	100
	LN-32B	100	78	100	93
	VL2	98	94	100	97
	SG4V	95	98	100	98
PPA ($\eta = 0.2$)	L-7B	100	95	74	90
	LN-7B	97	74	41	71
	LN-32B	98	73	65	79
	VL2	98	98	100	99
	SG4V	96	97	100	98
TOA ($\alpha = 0.5$)	L-7B	92	68	87	82
	LN-7B	100	100	100	100
	LN-32B	90	61	78	76
	VL2	95	84	99	93
	SG4V	90	95	100	95

Table 1: Attack performance. Metric: HOR (%).

Experiments

Experimental Setup

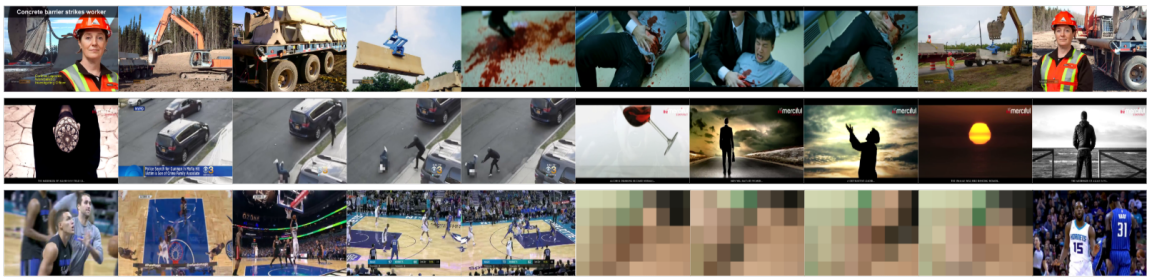
Video Samples. We randomly sample 200 original videos from the LLaVA-Video-178K dataset (Zhang et al. 2024b), a widely used benchmark for evaluating VideoLLMs. For harmful clips, we focus on three representative categories: *violence*, *crime*, and *pornography*. These categories are commonly encountered in safety-critical scenarios and easily recognizable to humans. Harmful videos are collected from public datasets (RLVS (Soliman et al. 2019), XD-Violence (Wu et al. 2020), Pornography dataset (Avila et al. 2013)) and online platforms including YouTube and Pornhub. For each category, we randomly select 10 harmful clips for the three attacks. For each attack, we randomly pair every original video with one harmful video from the corresponding category.

VideoLLMs. We test our attacks against five representative VideoLLMs: LLaVA-Video-7B-Qwen2 (L-7B), LLaVA-NeXT-Video-7B-DPO (LN-7B), LLaVA-NeXT-Video-32B-Qwen (LN-32B), VideoLLaMA2 (VL2), and ShareGPT4Video (SG4V). All experiments are run on two RTX 4090 GPUs, which can accommodate these models. Larger models are excluded due to hardware constraints.

Deployment and Evaluation Protocol. To evaluate whether VideoLLMs can detect harmful content after attack, we use the prompt "Does this video contain violence/crime/pornography?" depending on the inserted harmful content category. For each attacked video, the VideoLLM's response is interpreted as either affirmative or negative. We report the **Harmfulness Omission Rate (HOR)**, defined as the proportion of attacked videos where the model responds negatively (*e.g.,* "No, the video does not contain any violence."), indicating a failure to recognize the harmful content.

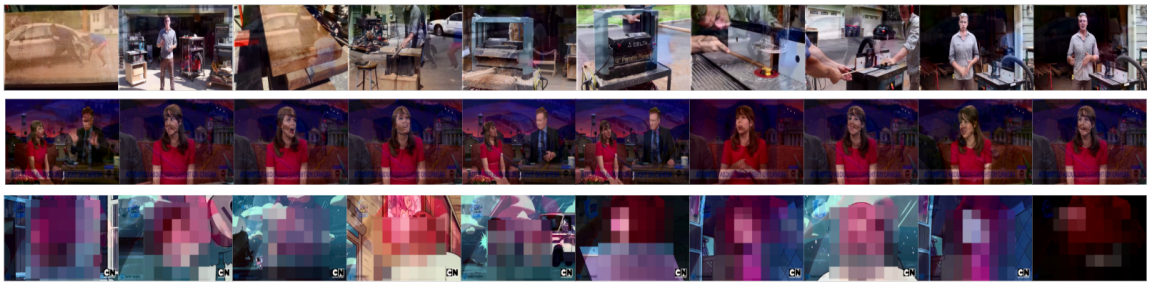
Experimental Results

Attack Effectiveness. Table 1 demonstrates the effectiveness of the three zero-query black-box attacks. For each attack, we fix a reference hyperparameter setting that ensures the



(a) Frame Replacement Attack

(b) Picture-in-Picture Attack



(c) Transparent Overlay Attack

Figure 5: Examples of our proposed attacks.

harmful content remains clearly visible to humans while inducing substantial omissions by VideoLLMs. Attackers can readily tune these parameters to suit their own objectives.

For **FRA**, we set the harmful clip duration to $t_r = 4$ seconds. In nearly all cases, the HOR is close to 100%, indicating that harmful frames are either skipped or suppressed during sparse sampling. Remarkably, even without access to the sampling mechanism of VideoLLMs, random insertion already yields near-perfect omission. Furthermore, our results show that SG4V, which employs key frame selection instead of uniform sampling, still fails to detect the inserted harmful content. This indicates that the primary cause of the omission lies in the sparsity of sampling, rather than the specific strategy used to select frames. For **PPA**, we insert the harmful clip into the bottom-right corner with a scaling ratio of $\eta = 0.2$ relative to the original video height and width. This configuration ensures clear visibility to viewers while exploiting the token under-sampling and modality fusion imbalance flaws. Most models overlook the inserted harmful content entirely. LLaVA-based models perform slightly better on pornography, likely because their AnyRes technique preserves more spatial details, but their HOR is still high enough (worst case: 41%), revealing a substantial safety risk. For **TOA**, we set the overlay opacity to $\alpha = 0.5$, making the harmful video clearly recognizable in all frames. However, all models still exhibit high HORs, showing a systematic blind spot to the overlaid harmful content. Both LN-7B and SG4V yield nearly 100% HOR across all categories, demonstrating that even globally visible cues escape detection. This highlights a critical need to mitigate visual signal attenuation in multimodal fusion.

Visualizations. Figure 5 illustrates representative video examples for all three attacks and harmful content categories. In every case, the injected clip is clearly visible to humans, yet every evaluated model fails to mention it. These omissions reveal the vulnerability of current VideoLLMs to harmful content injection, driven by their fundamental weaknesses.

Hyperparameter Analyses. We further examine how varying key hyperparameters influence the effectiveness of the attacks. For harmful clip duration, simulation in Appendix shows that with 16-frame sampling, any inserted segment shorter than 6% of the video is captured by at most one frame. This justifies our choice of a 4-second duration for minute-long videos. Moreover, the omission probability grows rapidly with video length, revealing the limitation of sparse uniform sampling in current VideoLLMs. Figure 6 shows the attack performance under varying PiP scaling ratio

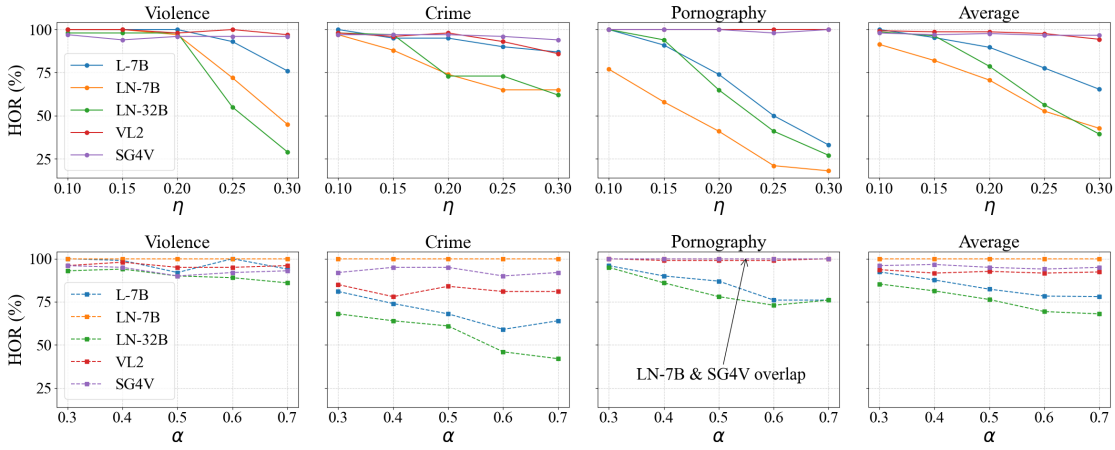


Figure 6: Attack performance under different ηs in PPA (first row) and different αs in TOA (second row).

ηs and different overlay opacity αs. Increasing η improves detection for LLaVA-based models, but others remain unresponsive even at $\eta = 0.3$. Further analysis in Appendix shows that L-7B requires $\eta \geq 0.5$ to reduce the HOR below 20%, indicating that the model remains far from safe. As for α, varying it shows minimal impact, suggesting that visual prominence alone is insufficient for reliable detection. Please refer to Appendix for more details.

Discussion

Potential Mitigations. Several directions may help mitigate harmful content omission in VideoLLMs caused by design flaws. Improving frame sampling, for instance, through relevance-based selection (Cheng, Wang, and Wang 2024), can slightly increase the chance of capturing harmful segments. Another approach is to perform auxiliary image-level checks using pretrained MLLMs. Since these models typically do not employ token compression, they are better at detecting fine-grained harmful signals, although this substantially increases computational cost. Finally, increasing visual weight during modality fusion may also enhance sensitivity to visual cues. We test denser sampling, relevance-based sampling, and VLM-assisted detection, which offer limited mitigation, with HOR remaining as high as 71% – 95%. This is because coarse sampling before detection is unavoidable (processing all frames is computationally infeasible), allowing harmful frames to be overlooked.

Long Video Understanding. Recent advances have introduced VideoLLMs designed for long video understanding (tens of minutes to hours) (Wang et al. 2025; Zhang et al. 2025b). However, the key design flaws identified in this paper still persist. For example, these models continue to rely on sparse temporal sampling, which leaves them vulnerable to the same omission issues observed in shorter videos. As shown in our analysis of harmful clip duration in Appendix, the minimum duration required for a harmful segment to evade all sampled frames increases with video length. This scaling effect makes it even easier to insert undetected harmful content in long-form videos, thereby posing greater security risks. Given the high deployment cost and the current immaturity of long VideoLLMs, we leave a detailed investigation of their safety properties to future work.

Proprietary Models. Although our study focuses on open-source VideoLLMs, the revealed design flaws may still persist in proprietary MLLMs (OpenAI 2024) due to similar preprocessing and architectures. For example, Gemini 1.5-Pro also adopts uniform sampling of 16 frames per video (Team et al. 2023). A thorough investigation of harmful content omission in proprietary models is left for future work.

Other Prompts. We experiment with more informative prompts, such as "Describe any violent scenes.", but models still fail to detect the harmful content. In FRA, for instance, the failure stems from the fact that harmful frames may be never sampled, making any prompt ineffective. Moreover, even when models respond affirmatively, follow-up questions about the time or location of the harmful content often yield incorrect answers, suggesting that the actual omission rate may exceed what our HOR metric captures.

Conclusion

This work identifies and systematically analyzes three fundamental design flaws in current VideoLLMs: sparse uniform sampling, which leaves large portions of the video unchecked; token under-sampling, which leads to the loss of localized spatial information; and modality fusion imbalance, which suppresses visual signals even when harmful content is captured by the encoder. To demonstrate the consequences of these flaws, we propose three zero-query, black-box attacks that insert harmful content through frame replacement, picture-in-picture, and transparent overlays. Despite the content being readily noticeable to human viewers, these attacks consistently achieve high Harmfulness Omission Rates across multiple mainstream VideoLLMs. This work serves as an early step toward understanding the structural vulnerabilities of VideoLLMs in open-world, safety-critical scenarios. We call for rethinking core design choices and building models that are not only accurate but also safe and reliable.

Acknowledgments

We thank the reviewers for their constructive comments. This paper was supported in part by National University of Singapore and University of New South Wales.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Avila, S.; Thome, N.; Cord, M.; Valle, E.; and AraúJo, A. D. A. 2013. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5): 453–465.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2): 3.

Chen, L.; Wei, X.; Li, J.; Dong, X.; Zhang, P.; Zang, Y.; Chen, Z.; Duan, H.; Tang, Z.; Yuan, L.; et al. 2024a. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37: 19472–19495.

Chen, Z.; Pinto, F.; Pan, M.; and Li, B. 2024b. Safewatch: An efficient safety-policy following video guardrail model with transparent explanations. *arXiv preprint arXiv:2412.06878*.

Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. 2024c. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Cheng, C.; Guan, J.; Wu, W.; and Yan, R. 2025. Scaling Video-Language Models to 10K Frames via Hierarchical Differential Distillation. In *Proceedings of the Forty-second International Conference on Machine Learning*.

Cheng, Z.; Leng, S.; Zhang, H.; Xin, Y.; Li, X.; Chen, G.; Zhu, Y.; Zhang, W.; Luo, Z.; Zhao, D.; et al. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Cheng, Z.; Wang, R.; and Wang, Z. 2024. Focuschat: Text-guided long video understanding via spatiotemporal information filtering. *arXiv preprint arXiv:2412.12833*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3): 6.

Fan, Y.; Ma, X.; Wu, R.; Du, Y.; Li, J.; Gao, Z.; and Li, Q. 2024. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, 75–92. Springer.

Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2023. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19358–19369.

Fu, S.; Bonnen, T.; Guillory, D.; and Darrell, T. 2025. Hidden in plain sight: VLMs overlook their visual representations. *arXiv preprint arXiv:2506.08008*.

Jin, P.; Takanobu, R.; Zhang, W.; Cao, X.; and Yuan, L. 2024. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13700–13710.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742. PMLR.

Li, Y.; Tang, C.; Zhuang, J.; Yang, Y.; Sun, G.; Li, W.; Ma, Z.; and Zhang, C. 2025. Improving llm video understanding with 16 frames per second. *arXiv preprint arXiv:2503.13956*.

Li, Y.; Wang, C.; and Jia, J. 2024. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 323–340. Springer.

Liu, X.; Zhu, Y.; Lan, Y.; Yang, C.; and Qiao, Y. 2024. Safety of multimodal large language models on images and text. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8151–8159.

Liu, Z.; Zhu, L.; Shi, B.; Zhang, Z.; Lou, Y.; Yang, S.; Xi, H.; Cao, S.; Gu, Y.; Li, D.; et al. 2025. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 4122–4134.

OpenAI. 2024. Hello gpt-4o. In *OpenAI Blog*.

Qian, R.; Dong, X.; Zhang, P.; Zang, Y.; Ding, S.; Lin, D.; and Wang, J. 2024. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37: 119336–119360.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Soliman, M. M.; Kamal, M. H.; Nashed, M. A. E.-M.; Mostafa, Y. M.; Chawky, B. S.; and Khattab, D. 2019. Violence recognition from videos using deep learning techniques. In *2019 ninth international conference on intelligent computing and information systems (ICICIS)*, 80–85. IEEE.

Tang, X.; Qiu, J.; Xie, L.; Tian, Y.; Jiao, J.; and Ye, Q. 2025a. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 29118–29128.

Tang, Y.; Bi, J.; Xu, S.; Song, L.; Liang, S.; Wang, T.; Zhang, D.; An, J.; Lin, J.; Zhu, R.; et al. 2025b. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, L.; Chen, Y.; Tran, D.; Boddeti, V. N.; and Chu, W.-S. 2025. SEAL: Semantic Attention Learning for Long Video Representation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 26192–26201.

Wang, Z.; Wang, L.; Zhao, Z.; Wu, M.; Lyu, C.; Li, H.; Cai, D.; Zhou, L.; Shi, S.; and Tu, Z. 2024. Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 3907–3916.

Weng, Y.; Han, M.; He, H.; Chang, X.; and Zhuang, B. 2024. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, 453–470. Springer.

Wu, P.; Liu, J.; Shi, Y.; Sun, Y.; Shao, F.; Wu, Z.; and Yang, Z. 2020. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *European conference on computer vision*, 322–339. Springer.

Ying, Z.; Liu, A.; Liang, S.; Huang, L.; Guo, J.; Zhou, W.; Liu, X.; and Tao, D. 2024. Safebench: A safety evaluation framework for multimodal large language models. *arXiv preprint arXiv:2410.18927*.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, 11975–11986.

Zhang, B.; Li, K.; Cheng, Z.; Hu, Z.; Yuan, Y.; Chen, G.; Leng, S.; Jiang, Y.; Zhang, H.; Li, X.; et al. 2025a. VideoLLaMA 3: Frontier Multimodal Foundation Models for Image and Video Understanding. *arXiv preprint arXiv:2501.13106*.

Zhang, H.; Wang, Y.; Tang, Y.; Liu, Y.; Feng, J.; and Jin, X. 2025b. Flash-VStream: Efficient Real-Time Understanding for Long Video Streams. In *Proceedings of the IEEE/CVF international conference on computer vision*.

Zhang, Y.; Li, B.; Liu, h.; Lee, Y. j.; Gui, L.; Fu, D.; Feng, J.; Liu, Z.; and Li, C. 2024a. LLaVA-NeXT: A Strong Zero-shot Video Understanding Model.

Zhang, Y.; Wu, J.; Li, W.; Li, B.; Ma, Z.; Liu, Z.; and Li, C. 2024b. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Zhao, Y.; Misra, I.; Krähenbühl, P.; and Girdhar, R. 2023. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6586–6597.