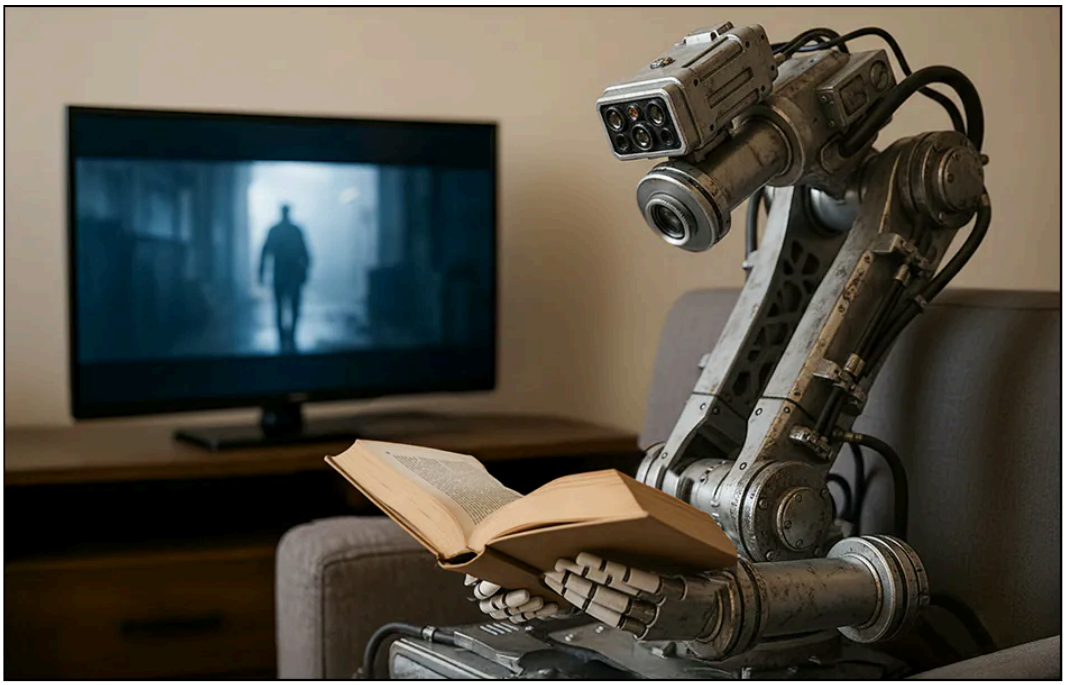
AI Would Rather Read the Book Than Watch the Movie

Originally published at https://www.unite.ai/ai-would-rather-read-the-book-than-watch-the-movie/



It's surprisingly hard to get AI models to watch and comment on actual video content, even if they are made for the task. They're more interested in the written word.

If you have ever tried to upload a small video-clip to ChatGPT, or to a similar popular vision/language model, you might have been surprised to realize that they cannot actually parse video. While models such as ChatGPT-4o+ are capable of analyzing individual frames – in the form of images, such as JPEG and PNG – they prefer the user to extract their own frames and upload them as images (which they *are* prepared to comment on).

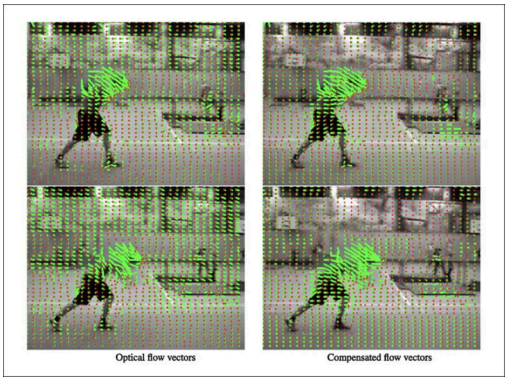
In the case of the OpenAI GPT series, one can, rather laboriously, extract a complete run of frames from a video clip and feed these to ChatGPT, for the purposes of, for instance, generating an AI-created narrative track for the video:

Images and code from an OpenAI tutorial on parsing multiple extracted frames for the purpose of developing an AI-generated commentary for a video clip. Source

But it falls to the user to make the conversion from video to frames, either by calling functions in a larger routine, as in the example above, or extracting the frames with FFMPEG or various free and paid video editing solutions.

To an extent, perhaps even a large extent, the limitations on video analysis in high-scale products such as ChatGPT hinge on *resource usage*: just tooling *one* AI instance with a selection of the most popular video codecs, and committing compute resources to the disk-heavy and CPU-throttling process of extraction is no minor consideration, should hundreds of millions of users decide to start using these facilities every day.

Additionally, temporal analysis can paint a very different picture than a single frame (imagine someone entering a house in a happy mood and then discovering a body); therefore considering the entire temporal 'checksum' of even a short video clip is a demanding and resource-intensive task – as well as a specialized area of the research literature, for instance with the ongoing development of frameworks such as Optical Flow –which essentially 'unfolds' a length of video so that it can be regarded and acted upon as if it were a static document:



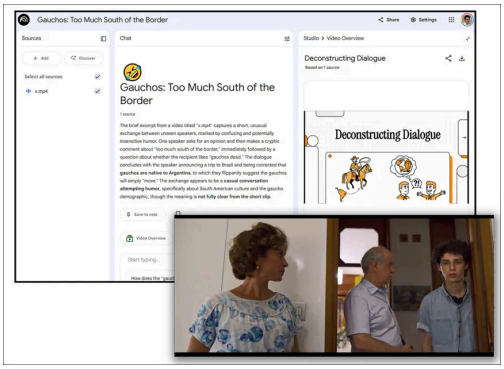
Optical flow diagrams highlight how motion is tracked across frames in a video sequence, with green vectors showing movement direction and intensity. These mappings provide the temporal continuity needed for VLMs and can also serve as structural guides in VFX workflows. Source

Settling for the Cliff's Notes

Nonetheless, since models such as Google's Notebook LM and the more recent ChatGPT entries are able to read associated metadata (i.e., embedded text content that contextualizes the video in some way), they do not ban video file uploads; and sometimes, they will even endeavor to interpret a video that has no such data.

In the following case, I uploaded a 6-second random clip from the Italian movie *The Hand of God* (2021) to NotebookLM, ensuring that the clip contained zero useful text, either in metadata or in the file name.

NotebookLM then proceeded to elaborately hallucinate material totally unrelated to the video*, complete with a nonsensical and unrelated five-minute head-to-head podcast:



An everyday moment in a six-second clip from an Italian movie is wildly misinterpreted by NotebookLM. Source: Google NotebookLM

Though Notebook, like ChatGPT, will accept a YouTube video as input, it will only do so if the video features an interpretable text-layer annotation and/or subtitles (not rasterized subtitles that are burned into the video).

In this way, the hard work of actually *looking* at and listening to the content of the video and performing semantic interpretation of it (a legal necessity for YouTube, due to its copyright protection measures, and its pending identity protection system), has been done at leisure after the user upload, as and when the clip could be allocated the necessary processing resources.

Real video interpretation is expensive and exhausting, and, it transpires, even models which have been trained specifically to perform this task would rather read text than actually look at video.

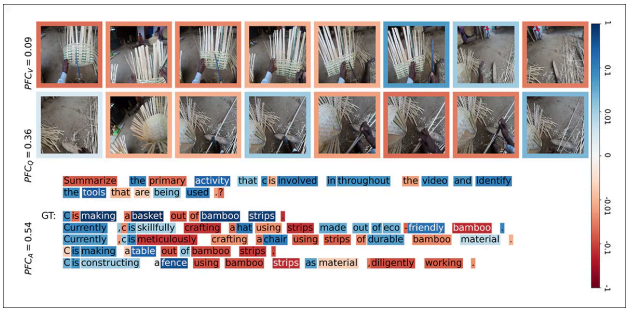
TL;DW

This, according to a new paper from the UK's University of Bristol, titled *A Video Is Not Worth a Thousand Words*, wherein the two authors conclude that current state-of-the-art vision language models (VLMs) – models specifically intended to be able to analyze video in a more effortful way, and to participate in *video question answering* (VQA) – also default to text-based information whenever they can.

When given both moving pictures and written questions and multiple-choice answers, the paper's authors found that models usually based their choices on patterns in the text, rather than anything happening on screen – in many cases performing just as well even when the question was *taken away entirely*.

In what appears to be a habitual form of shortcutting or cheating, what mattered most to the majority of the models was being able to spot patterns in the possible answers; only when the task was made harder, by adding more answer options, did the AIs begin to pay closer attention to the video.

The authors gave VQA tests under a range of conditions to six VLM models of diverse context lengths, on four suitable datasets; and found that the results indicated the models' dependence on text over video content.



Example from the study showing how a video-analysis model weighs what it sees versus what it reads. The clip shows a person weaving bamboo, yet the model assigns much more importance to the written question and answer text than to the video frames themselves. Blue highlights mark elements that support the chosen answer, while red marks those that pull it in the opposite direction, illustrating how the model's reasoning centers on the wording rather than on the moving images. Source

The new paper is accompanied by a GitHub repository with runtime and data.

Method

To understand how much each input contributes to a model's decision, the new work uses a method from game theory called Shapley Values. Originally designed to fairly divide a payout among players in a coalition, Shapley Values assign credit to each 'player' based on their individual impact.

Effectively then, the players in this scenario are either the video frames or the text components (annotations, subtitles, captions, etc.) of a VQA task; and the 'payout' is the model's final answer. By systematically testing what occurs when each part is added or removed, the technique reveals how important that element was in getting to the chosen answer.

In the case of the new project, in order to extend the method across multiple data types, Shapley Values were adapted to handle mixed modalities, with video and text components treated discretely, and their varying influence on the models' outputs measured, revealing whether the video content was being truly interpreted, or whether written clues were being used as shortcuts.

Metrics

Two simple metrics were defined to compare how much each modality (i.e., video, question, or answer) contributed to the model's decision: *Modality Contribution* measures how much of the total explanation came from each type of input; here, all available Shapley values are summed up, and the share belonging to each modality calculated as a percentage of the total.

Secondly, *Per-Feature Contribution* corrects for the fact that some modalities, like video, contain many more features than others. Instead, the average Shapley value is calculated for each feature, and those averages compared to determine which modality's influence is dominant.

Data and Tests

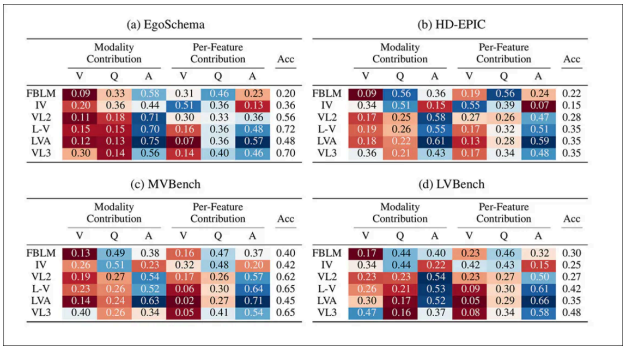
The authors tested the approach on six VLMs with a variety of characteristics designed to ensure that the tests' principles were broadly applicable and generalizable. Therefore the models were chosen for different context lengths, different ages (i.e., how long since the framework was released), and different architecture configurations.

The contenders were FrozenBiLM; InternVideo; VideoLLaMA2; VideoLLaMA3; LLaVa-Video (which leverages Qwen2); and LongVA (also using Qwen2).

With the same aim of diverse variety, the four target datasets chosen were EgoSchema, a VQA dataset designed to be impossible to complete without full viewing of the associated videos; HD-EPIC, a kitchen-focused dataset featuring some unusually long videos; MVBench, a curated assembly of contributions from other datasets; and LVBench, which poses VQA queries for very long videos.

From these, the authors devised 60 questions – ten from each question type.

The contribution metrics made clear that most models relied less on the video than on the text, especially when judged frame by frame. Even where video made a reasonable showing in overall contribution, its per-feature influence was often minimal, suggesting that while the model might be using the video in aggregate, it was paying little attention to individual frames. VideoLLaMA3 was the main outlier here, with a heavier visual dependence, especially on the longer sequences in LVBench:



(a) EgoSchema

	Modality Contribution			Per-Feature Contribution			Acc
	V	Q	A	V	Q	A	
FBLM	0.09	0.33	0.58	0.31	0.46	0.23	0.20
IV	0.20	0.36	0.44	0.51	0.36	0.13	0.36
VL2	0.11	0.18	0.71	0.30	0.33	0.36	0.56
L-V	0.15	0.15	0.70	0.16	0.36	0.48	0.72
LVA	0.12	0.13	0.75	0.07	0.36	0.57	0.48
VL3	0.30	0.14	0.56	0.14	0.40	0.46	0.70

(b) HD-EPIC

	Modality Contribution			Per-Feature Contribution			Acc
	V	Q	A	V	Q	A	
FBLM	0.09	0.56	0.36	0.19	0.56	0.24	0.22
IV	0.34	0.51	0.15	0.55	0.39	0.07	0.15
VL2	0.17	0.25	0.58	0.27	0.26	0.47	0.28
L-V	0.19	0.26	0.55	0.17	0.32	0.51	0.35
LVA	0.18	0.22	0.61	0.13	0.28	0.59	0.35
VL3	0.36	0.21	0.43	0.17	0.34	0.48	0.35

(c) MVBench

	Modality Contribution			Per-Feature Contribution			Acc
	V	Q	A	V	Q	A	
FBLM	0.13	0.49	0.38	0.16	0.47	0.37	0.40
IV	0.26	0.51	0.23	0.32	0.48	0.20	0.42
VL2	0.19	0.27	0.54	0.17	0.26	0.57	0.62
L-V	0.23	0.26	0.52	0.06	0.30	0.64	0.65
LVA	0.14	0.24	0.63	0.02	0.27	0.71	0.45
VL3	0.40	0.26	0.34	0.05	0.41	0.54	0.65

(d) LVBench

	Modality Contribution			Per-Feature Contribution			Acc
	V	Q	A	V	Q	A	
FBLM	0.17	0.44	0.40	0.23	0.46	0.32	0.30
IV	0.34	0.44	0.22	0.42	0.43	0.15	0.25
VL2	0.23	0.23	0.54	0.23	0.27	0.50	0.27
L-V	0.26	0.21	0.53	0.09	0.30	0.61	0.42
LVA	0.30	0.17	0.52	0.05	0.29	0.66	0.35
VL3	0.47	0.16	0.37	0.08	0.34	0.58	0.48

Modality Contribution (MC) and Per-Feature Contribution (PFC) scores across models and datasets, showing the relative weight of video (V), question (Q), and answer (A) inputs. Cooler colors indicate stronger contributions; warmer colors indicate weaker or negligible influence. Across most settings, language is clearly dominant, with video often sidelined – especially in per-frame influence.

As for the text, the question tended to matter more than the answer, particularly in stronger models. This was most obvious in datasets like EgoSchema, where questions were longer and more naturalistic, while the answers were short and sometimes schematic. MVBench reversed this somewhat, since its binary-answer structure inflates the apparent importance of the answer tokens.

Across all models and datasets, though, vision was consistently sidelined, with language doing most of the heavy lifting.

The paper states:

> *'[For] long context models, video shows vastly reduced contributions, meaning that per-frame the Shapley values are much smaller than their text feature counterparts.*
>
> *'Video as a whole modality is clearly still highly relevant, but this is evidence that the Shapley values of its individual frames are more centered around zero, and that the model's attention to them is much less guided than for the text.'*

To test how each part of the input (video, text, etc.) contributes to model accuracy, the researchers conducted additional tests using *masking* – deliberately hiding one or more parts of the input, and seeing how much the model's accuracy changes as a result.

If performance drops sharply when a particular input is removed, that input is probably important; f the model performs about the same, it suggests that the missing piece wasn't heavily relied upon. In this sense, masking tests are a kind of iterative ablation study.

Model	Masking	EgoSchema	HD-EPIC	MVBench	LVBench
FrozenBiLM	None	0.20	0.22	0.40	0.30
	All	-0.02	+0.00	-0.23	+0.03
	Video	+0.00	-0.03	-0.05	-0.02
	Question	+0.02	-0.02	-0.03	+0.00
	Answer	-0.02	+0.00	-0.23	+0.03
InternVideo	None	0.36	0.15	0.42	0.25
	All	-0.18	+0.07	-0.25	+0.08
	Video	-0.14	+0.07	-0.10	+0.05
	Question	+0.06	+0.02	+0.02	+0.03
	Answer	-0.18	+0.07	-0.25	+0.08
VideoLLaMA2	None	0.56	0.28	0.62	0.27
	All	-0.36	+0.00	-0.30	-0.02
	Video	-0.18	+0.00	-0.23	+0.00
	Question	+0.02	-0.13	-0.13	+0.05
	Answer	-0.34	+0.03	-0.22	-0.03
LLaVA-Video	None	0.72	0.35	0.65	0.42
	All	-0.54	-0.17	-0.48	-0.08
	Video	-0.26	-0.12	-0.23	-0.07
	Question	+0.04	-0.05	-0.07	-0.12
	Answer	-0.58	-0.17	-0.28	-0.32
LongVA	None	0.48	0.35	0.45	0.35
	All	-0.28	-0.17	-0.15	-0.05
	Video	-0.10	-0.08	-0.05	-0.05
	Question	-0.06	-0.12	-0.02	+0.05
	Answer	-0.30	-0.17	-0.12	-0.17
VideoLLaMA3	None	0.70	0.35	0.65	0.48
	All	-0.52	-0.23	-0.48	-0.15
	Video	-0.26	-0.05	-0.33	-0.20
	Question	-0.06	-0.07	-0.08	-0.08
	Answer	-0.48	-0.23	-0.40	-0.18

Performance impact of masking video, question, or answer inputs across four VQA benchmarks. Scores show change from the unmasked baseline. Red means lower accuracy, green higher. Models often retained high scores without video, but lost more when the (text) answer was removed. The question could usually be masked with minimal effect.

The results (visualized above) indicate that answers (text answers in the multiple choice data) carry the most weight across the board. Masking the answer typically caused the biggest drop in accuracy, often reducing models to near-random performance.

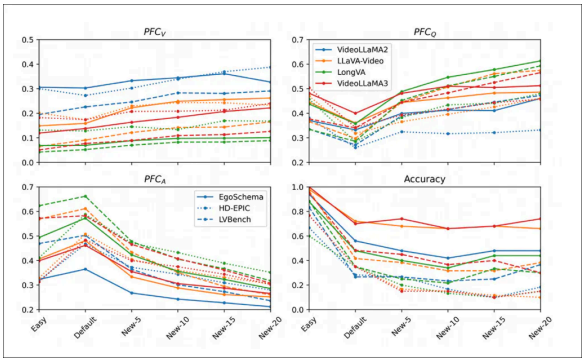
However, masking the *question* usually had much less effect, which supports the earlier finding that models often undervalue the question.

In some cases, accuracy even improved *when the question was removed*, which implies that models were sometimes just matching answers to visual or textual cues rather than properly evaluating the question.

Models also varied in their dependence on video: some maintained reasonable accuracy without it, further confirming the limited contribution of video features in many current setups.

The authors subsequently tested whether models could be forced to rely on video by adding extra *wrong answers* to the multiple-choice options.

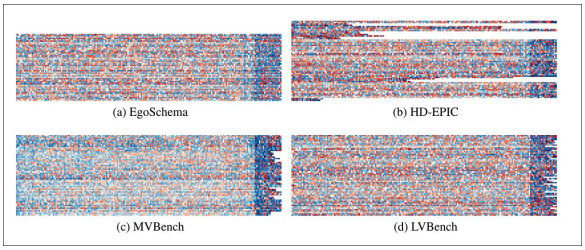
When the distractors were easy and recycled from other questions, performance improved, as models matched text patterns without much reasoning. But with ten or more unrelated answers, they began to depend more on the video and question:



Per-feature contribution and accuracy for video, question, and answer inputs, as additional wrong answers, are added to each VQA test, showing that increasing the number of distractors reduces text dominance and raises the relative influence of visual and question features.

For VideoLLaMA3, masking the video reduced accuracy by 40% on EgoSchema and 15% on LVBench, indicating that a simple increase in answer count can shift models away from text shortcuts, and toward genuine multimodal reasoning.

The researchers also explored how attribution is distributed across inputs, and below we see heatmaps of Shapley values for each model input:



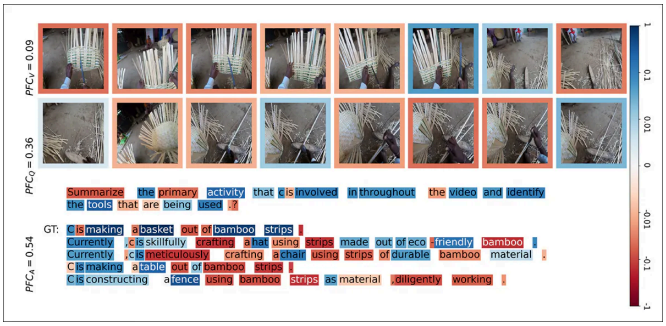
Shapley value heatmaps for four datasets, where each row shows one VQA-tuple, and each column a single feature. Video features appear on the left, followed by text. The much stronger values in the (red) text regions confirm that models rely far more on language than video.

Commenting on the results above, the authors state:

> *'The magnitude of the Shapley values are much larger towards the right hand side of each heatmap, representing the question and answer attributions. This stark boundary is where the video frames end and the text features begin, demonstrating that the video modality contribution is much less than the question/answer. '*

In summary, across all datasets, the values are far stronger toward the text end of the spectrum, strongly indicating that models rely far more on language than visual cues. Even where the video *is* used, the contribution is spread thinly across many frames, often with no consistent pattern.

Below we see an annotated example from EgoSchema. The 16 most 'important' frames were selected using Shapley values and colored according to their influence, with blue indicating a positive contribution and red a negative one:



Shapley attributions for a single EgoSchema example, showing the 16 most influential frames and all text inputs. Video contributions are minimal compared to text, which dominates the model's reasoning. Blue and red indicate positive and negative influence on the selected answer.

The result is that almost every frame has only weak influence compared to the words in the question and answers. Visual clues are sparse and inconsistent, while nouns such as 'chair' and 'fence' steer the model toward the correct choice – or away from it, depending on context.

Conclusion

Anyone who has ever dabbled in video-editing or video analysis will already know how resource-intensive these processes are, and understand why companies parsing millions of AI-based requests a day cannot afford to casually allow *ad hoc* editing and interpretive video processes to be run by users.

One thing to remember in this regard is that nearly every API AI interface you'll ever try (with the possible exception of a brand-new and short-lived demo in support of new scientific research) is looking to accomplish users' wishes at the minimum possible level of resource expenditure.

That means relying on existing metadata from user-supplied data or RAG retrievals, if at all possible; and extracting (if absolutely necessary) metadata for more parseable formats such as PDF, documents and single images.

What's not on the table is running your uploaded video through CLIP or the latest YOLO release, or through any power-guzzling and time-consuming VLM that can actually identify what's in the frames, and understand what's happening in the supplied video, accounting for temporality.

However, this does not mean the phenomena that the current paper documents necessarily proceeds from parsimonious architectural approaches. The authors observe that text dominates state-of-the-art multimodal training paradigms in any case, indicating that 'visual language' is either less developed, less important or informative within a multimodal context, or (at least at the moment) less well-understood,

* *Interestingly, the material the NotebookLM came up with appears to be either entirely original or totally unindexed by Google, as I am unable to find any we results that could have sneaked into the training data and prompted this output.*

First published Friday, October 31, 2025; edited 14:20 for formatting

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More