Can Video Large Language Models Comprehend Language in Videos?

Minjoon Jung[1,2]* **Junbin Xiao**[1] **Byoung-Tak Zhang**[2] **Angela Yao**[1]

[1]Department of Computer Science, National University of Singapore
[2]Interdisciplinary Program in Artificial Intelligence, Seoul National University
{mjjung, btzhang}@bi.snu.ac.kr, {junbin, ayao}@comp.nus.edu.sg

Abstract

Recent advancements in video large language models (Video-LLMs) have shown capabilities of temporally-grounding language queries or retrieving video moments in videos. However, such capabilities have not been thoroughly verified to be robust and trustable. In this study, we explore the consistency of Video-LLMs in grasping temporal moments within videos — a critical indicator for robust and trustworthy video language comprehension. Specifically, we devise different probes where Video-LLMs first predict temporal moments based on language queries, followed by verification questions to assess whether the predicted moments accurately reflect the queries. Our results show that current Video-LLMs respond unintuitively to such assessment; they often fail to provide consistent answers upon re-evaluation and even get near chance-level performance. This reveals the significant shortcomings in the current capabilities of Video-LLMs for reliable video temporal understanding, underscoring the need for further research and development in this field.

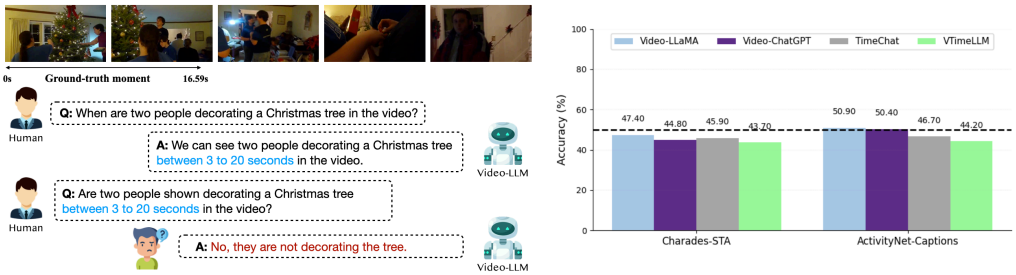

Figure 1: **Left**: An example of inconsistent behavior of Video-LLMs, where their answers contradict their initial temporal predictions. **Right**: We assess the accuracy of Video-LLMs in consistency, focusing on accurate original predictions (IoU$\geqslant$0.5). The results reveal the severe inconsistency in Video-LLMs; they perform marginally above or even below chance level (50%).

1 Introduction

The capability of video language comprehension requires the models to thoroughly understand and align a language query with a specific video moment [1, 2]. Recent advances [3–7] in video large language models (Video-LLMs) have shown promising results in time-related video understanding tasks, such as Video Temporal Grounding (VTG) [2, 8], Dense Video Captioning (DVC) [9, 10], Grounded

*Work done while the first author was an research intern at NUS@CVML.

Workshop on Video-Language Models at 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Video Question Answering [11], where models are to identify and provide specific moments and details within video sequences. Despite the increasingly high performance on standard benchmarks, it is unclear whether their predictions are truly grounded in video language comprehension, or because of other short-cuts like spurious vision-text correlations [11, 12]. To study this, we conduct a series of verification experiments to measure if the grounded temporal moments can accurately reflect the original queries.

Specifically, we consider prediction consistency as a key indicator for faithful video language comprehension, and design the following probes for consistency checking:

1. *Consistent Temporal Grounding*: A consistent Video-LLM would predict temporally close timestamps when provided with sentences that convey the same meaning. To check this, we generate rephrased sentences (*i.e.*, aligned sentences) and assess whether the Video-LLM can make consistent temporal predictions by measuring the IoU values between them.
2. *Self-Answer Verification*: Accurate timestamp prediction is important, but it's equally crucial to validate that the model correctly identifies the event in the predicted moment. We prompt the Video-LLM to localize the temporal moment for a given query and then verify if the query is truly present in that moment. To prevent the model from achieving perfect consistency by always answering "yes," we also use misaligned sentences, which intentionally distort the meaning of the original query, along with various question templates
3. *Compositional Understanding*: Since video content is often complex, compositional understanding is crucial for effective video comprehension. We evaluate the Video-LLM's ability to accurately capture the compositional details in a query sentence and apply this understanding for temporal grounding. Specifically, we break down a holistic query into a series of sub-queries (*i.e.*, compositional information) and check if the temporal predictions of the sub-queries are consistent with the original holistic query.

A model capable of faithful video language comprehension should achieve high consistency in the above probes, as illustrated in Figure 2 (Right). With the probes, we examine a series of Video-LLMs (such as Video-ChatGPT [13], Video-LLaMA [14], TimeChat [3], VTimeLLM [4]) on two popular temporal sentence grounding datasets ActivityNet-Captions [9] and Charades-STA [2]. Interestingly, we find that Video-LLMs often generates conflicting responses when being asked to verify their own answers. For instance, Video-LLM answers with "No." when being asked if the query presents in its predicted temporal moment, as shown in Figure 1. A further study shows that TimeChat, one of the state-of-the-art (SOTA) Video-LLMs, behaves below chance-level (50%) in this answer verification probe, revealing its severe deficiency of video language comprehension. Moreover, we find that higher grounding performance does not necessarily lead to higher consistency, indicating a disjoint between performance and trustworthiness in model development and alerting the urgent need for improved rationality.

Our contributions can be summarized as follows:

- We systematically analyze Video-LLMs' capabilities of temporal video grounding from a perspective of consistency. We construct the temporal comprehension evaluation sets and design three kinds of probes for consistency checking: Consistent Temporal Grounding, Self-Answer Verification, and Compositional Understanding.
- Given our results, Video-LLMs often struggle to provide consistent moment predictions even though their initial predictions are accurate. Also, higher performance does not necessarily guarantee consistent moment predictions.
- We reveal that most Video-LLMs achieve near chance- or random-level consistency in video language comprehension. Additionally, Video-LLMs that are specially instruction-tuned for time understanding do not necessarily behave better than those Video-LLMs that are tuned for naive QA or conversation.

2 Evaluation

We formally define a video as v, a query sentence as q, and the corresponding temporal moment as m (m is given by the start and end timestamps). Thus, the process of temporal sentence grounding can be denoted as $m = \text{Temp}_G(v, q)$, and the process of verifying within this moment can be denoted

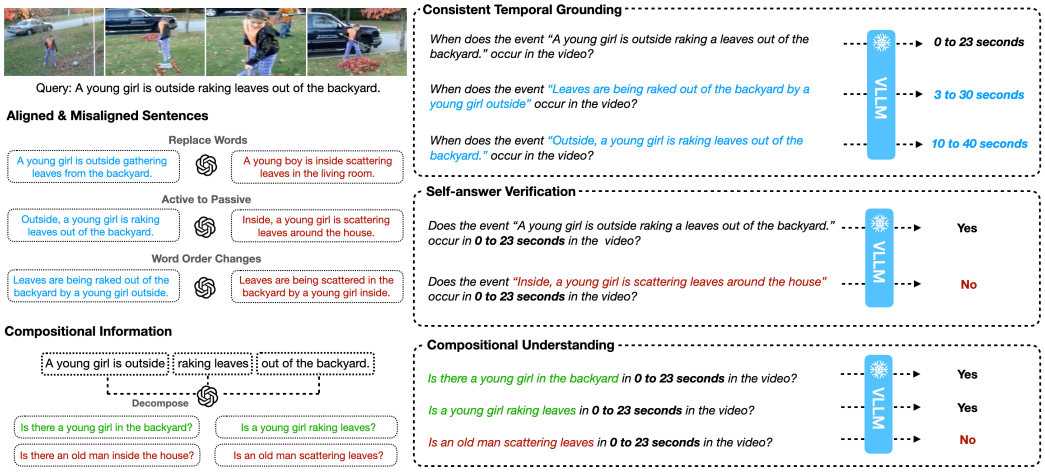

Figure 2: **Left:** An example of aligned and misaligned sentences and compositional information. **Right:** The evaluation assesses the Video-LLMs' consistency and compositional understanding. We first test whether the model can consistently ground a video moment by giving queries that have the same meaning but are phrased differently. Next, we evaluate whether the Video-LLMs can confirm its predictions and accurately understand the predicted moment's contents and its components.

Datasets	Samples	Aligned & Misaligned Sentences	Compositional Information
Charades-STA [2]	707	2121 (3.0)	2827 (3.9)
ActivityNet-Captions [9]	1422	4266 (3.0)	6222 (4.3)

Table 1: Statistics of evaluation datasets. The number in parenthesis represents the average number for each sample.

as $a = \text{Temp}_V(v, q, m) \in \{\text{Yes}, \text{No}\}$. With this definition, we consider a series of experiments for checking video language comprehension: (1) Consistent Temporal Grounding, (2) Self-answer Verification, and (3) Compositional Understanding.

Consistent Temporal Grounding. We begin by evaluating the consistency of Video-LLMs' moment predictions based on their grounding abilities. Our hypothesis is that if the model is consistent, the predictions from sentences conveying the same meaning should be close. To do this, we generate an aligned sentence $\tilde{q}$, which has the same meanings as q but is expressed differently. We measure how close between moment predictions $m = \text{Temp}_G(v, q)$ and $\tilde{m} = \text{Temp}_G(v, \tilde{q})$.

Self-Answer Verification. Based on the model's moment prediction $m = \text{Temp}_G(v, q)$, we then ask the model whether q occurs in m in the video. However, the model could achieve perfect consistency by simply answering "Yes" to all questions. To address this, we generate a misaligned sentence $\bar{q}$ to evaluate whether the model can correctly respond to contradictory query sentences. The $\tilde{q}$ conveys the same meaning with q, while $\bar{q}$ either contrasts with q or contains incorrect information. Consequently, the answers $a = \text{Temp}_V(v, q, m)$ and $\tilde{a} = \text{Temp}_V(v, \tilde{q}, m)$ should be the same, but the answer $\bar{a} = \text{Temp}_V(v, \bar{q}, m)$ should be different. Additionally, we randomly select question templates requiring varied responses to prevent the model from always defaulting to "Yes." As shown in Table 2, while the correct answer to the question template "`Does` q `occur from` m `in the video?`" is "Yes," the answer to the question template "`Is` q `missing from` m `in the video?`" would be "No." This challenge rigorously tests the model's reasoning capabilities.

Compositional Understanding. Given the complexity and intricacy of the video's content, compositional understanding is crucial for accurately predicting timestamps in videos. Hence, it's important to verify whether the model predicts moments based on a genuine understanding of the compositional components within those moments. To achieve this, we decompose query sentences into their key components and assess whether the model captures this sub-information effectively. For example, if the model answers "0 to 5 seconds." to the question "A young girl is outside raking leaves out of the backyard.", we verify whether the model correctly identifies and understands key components of the

Answer Type	Examples
Yes	Is the event q present from m in the video? Is the event q occurring from m in the video? Does the event q happen from m in the video? Is the event q included from m in the video?
No	Is the event q absent from m in the video? Is the event q not present from m in the video? Does the event q not happen from m in the video? Is the event q missing from m in the video?

Table 2: Examples of question templates. Given a query sentence q and the model's predicted timestamp m, the above question templates can be used for self-answer verification.

scene, such as the presence of the young girl, the outdoor setting, and the action of raking leaves in its moment predictions. This ensures the model's prediction is based on genuine comprehension of the video's compositional elements, rather than shallow pattern recognition.

2.1 Constructing Evaluation Datasets

We first curate a test set based on existing benchmarks: Charades-STA [2] and ActivityNet Captions [9]. We sample 500 videos for each dataset and filter out annotations in the dataset where the timestamp is too long (*e.g.*, over 70% of the total video length), too short, or where the query sentence is too short, which may cause inaccurate evaluation. This results in 707 and 1,422 query-moment pairs for Charades-STA and ActivityNet-Captions, respectively. Below, we detail the process of generating aligned and misaligned sentences, as well as compositional information using a closed-source LLM (*i.e.*, gpt-4o-mini [15]) for each query-moment pair.

Aligned & Misaligned Sentences. We generate aligned and misaligned sentences using several key techniques: 1) Replace words: Replace key nouns and verbs. 2) Active to Passive: Convert active sentences to passive. 3) Word Order Changes: Rearrange the word order in the sentence. Following the above techniques, we generate three aligned and misaligned sentences for each sentence.

Compositional Information. To extract the compositional information, we break query sentences down into their key components. Specifically, we extract them based on two key attributes: 1) Subject Identification: Identifying the main entities involved in the sentence. 2) Actions: Describing what the subjects are doing or what is happening to them. The number of compositional information may vary for each sentence.

We give a summary of generated evaluation sets in Table 1. To ensure the quality of evaluation sets, we conduct a human evaluation to verify that the sentences are appropriately generated as we intended. The generated sentences exhibit high agreement with human assessments. More details are specified in Appendix A.

2.2 Evaluating Video-LLMs

Consistency in predictions is important, but if those predictions are based on incorrect or irrelevant moments, the consistency metric becomes less meaningful and may lead to misleading conclusions. With this in mind, we want to test the consistency of the Video-LLMs' accurate predictions—those with an IoU greater than 0.5 between the ground-truth and predicted moments—in the evaluations.

We utilize four state-of-the-art Video-LLMs: Video-LLaMA [16], Video-ChatGPT [13], TimeChat [3], and VTimeLLM [4] for our experiments. Video-LLaMA enables video comprehension by cross-modal training for both vision and audio modalities in the video. Video-ChatGPT designs spatiotemporal video modeling and constructs video instruction tuning upon LLaVA [17]. Unlike these Video-LLMs, TimeChat and VTimeLLM have been proposed to tackle time-related video understanding tasks, such as VTG and DVC. TimeChat introduces TimeIT, a time-aware instruction-tuning dataset, and develops the video encoder to learn temporal information in the video. VTimeLLM implements a three-stage temporal-aware training method that requires the model to identify events in the video and provide their corresponding timestamps. One may note that the performance gap

Methods	Charades-STA				ActivityNet-Captions			
	R@1,0.5	R@1,0.7	R_{con}@1,0.5	R_{con}@1,0.7	R@1,0.5	R@1,0.7	R_{con}@1,0.5	R_{con}@1,0.7
Video-LLaMA [16]	10.04	2.55	53.52	48.36	10.62	4.01	56.51	54.53
Video-ChatGPT [13]	14.43	7.64	89.22	87.90	6.68	2.95	64.56	63.86
TimeChat [3]	30.69	13.15	80.49	64.06	4.64	2.04	64.14	58.59
VTimeLLM [4]	27.72	11.88	83.16	80.61	31.43	17.16	83.30	78.82

Table 3: Performance of the Video-LLMs on Charades-STA and ActivityNet-Captions datasets. R_{con} is considered only when the model's initial prediction has an IoU higher than 0.5, comparing IoU values for the predictions between original and aligned sentences. Despite the Video-LLMs' accurate initial predictions, they often struggle to maintain consistent timestamp predictions.

Methods	Charades-STA		ActivityNet-Captions	
	Self-answer Verification	Compositional Understanding	Self-answer Verification	Compositional Understanding
Random	50.0	50.0	50.0	50.0
Video-LLaMA [16]	50.6	49.7	49.4	53.4
Video-ChatGPT [13]	52.0	51.8	51.0	49.4
TimeChat [3]	53.0	55.7	49.9	51.9
VTimeLLM [4]	52.0	51.7	50.8	52.4

Table 4: ROC-AUC scores of the Video-LLMs for the Answer Verification and Compositional Understanding tasks on datasets. Most Video-LLMs show poor video comprehension, demonstrating their performances are close to random.

between Video-LLMs may be influenced by the fact that TimeChat includes Charades videos and VTimeLLM incorporates ActivityNet-Captions as part of their instruction tuning.

In the evaluation, the question and template formats may vary depending on the model. We adhere as closely as possible to the prompts used in their instruction tuning to reproduce their performance faithfully. More details on these models and experiment settings can be found in Appendix B.

Consistency and accuracy in temporal grounding are not necessarily proportional. In Table 3, we use the R@1 metric with IoU thresholds to present the performance of Video-LLMs and their consistency in temporal grounding (*e.g.*, $R_{con}@1, 0.5$). Note that the consistency is measured only for initial predictions with an IoU higher than 0.5. While TimeChat and VTimeLLM have been specifically designed for time-related video understanding tasks, their high overall performance does not necessarily translate into consistent temporal grounding abilities. For instance, although TimeChat outperforms Video-ChatGPT on Charades-STA in terms of accuracy, it exhibits weaker consistent temporal grounding. Similarly, while Video-LLaMA achieves higher performance than Video-ChatGPT on ActivityNet-Captions, it fails to deliver consistent predictions. Even though Video-ChatGPT shows the best consistency on Charades-STA, its limited number of accurate predictions undermines its reliability, making it difficult to draw strong conclusions.

Video-LLMs struggles with self-answer verification and compositional understanding: As shown in Figure 1, most Video-LLMs show near or below chance level performance. This tendency persists in Table 4, which shows ROC-AUC scores of the Video-LLMs in both self-answer verification and compositional understanding. The fact that these scores are generally close to random suggests poor comprehension of video content. While VTimeLLM demonstrates relatively better grounding performance compared to other Video-LLMs on ActivityNet-Captions in Table 3, this improvement does not extend to consistency or compositional understanding. Overall, Video-LLMs demonstrate significant deficiencies in both self-answer verification and compositional understanding, highlighting their moment predictions may not be truly grounded in video comprehension.

Inconsistent answers between self-answer verification and compositional understanding: In Figure 3, we give the responses of Video-LLMs during our evaluation. We can see that TimeChat provides an answer that contradicts the predicted timestamp in self-answer verification. Also, we find inconsistencies in the Video-LLMs' answers when performing self-answer verification and compositional understanding. For instance, Video-ChatGPT confirms that the event "A small group of people are seen standing together and speaking" occurs at its predicted moment. However, Video-ChatGPT also confirms that the misalignment sentence of the given event occurs at the same time,

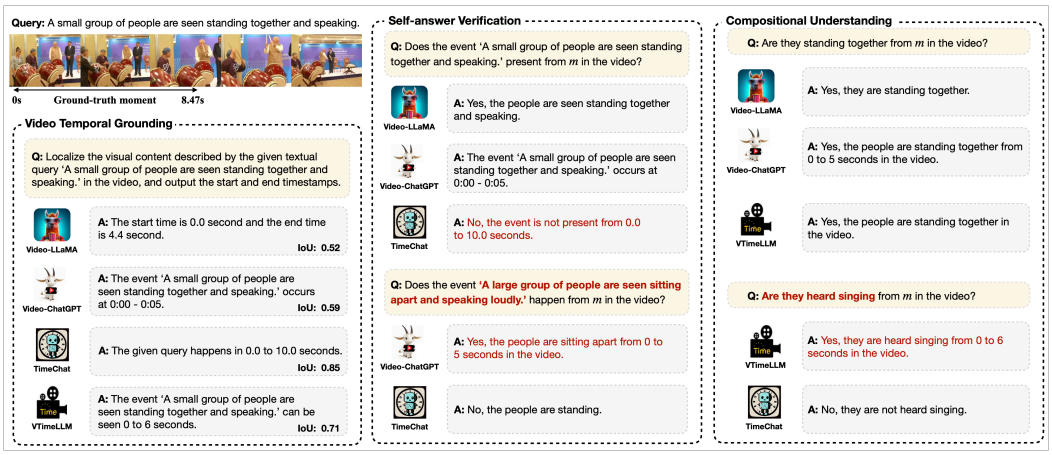

Figure 3: Visualization of qualitative examples with four Video-LLMs. We first ask Video-LLMs to localize the given sentence. We then replace m in the questions with the predicted timestamp for each model during self-answer verification and compositional understanding. The red text represents using the misaligned sentence in the question and the Video-LLMs' inconsistent answer.

which contradicts the earlier answer. These inconsistencies between the tasks highlight significant gaps in the model's ability to provide reliable predictions based on actual video comprehension.

3 Disconnection Between Temporal Grounding and True Comprehension

While Video-LLMs have demonstrated promising results in various video understanding tasks, our findings suggest that their temporal predictions may not be genuinely based on video comprehension. In this section, we explain why video comprehension is not essential for Video-LLMs to achieve high temporal grounding performance. First, we show that while Video-LLMs achieve high performance through instruction tuning on target datasets, they still lack true video comprehension, as evidenced by self-answer verification. Next, we explore the video instruction tuning in existing Video-LLMs, and argue that it fails to ensure consistency.

3.1 Video-LLMs with Instruction Tuning

But perhaps consistency may be affected by the subjectivity of the target dataset, and stronger grounding capabilities in Video-LLMs could potentially lead to better consistency. To confirm this, we utilize two Video-LLMs, Video-LLaMA and TimeChat, and conduct instruction tuning on Charades-STA and ActivityNet-Captions for the VTG task, respectively. Specifically, we fine-tune the Video-LLMs using annotations from the training splits of each dataset. We measure the R@1, 0.5 metric for temporal grounding, as well as ROC-AUC scores for self-answer verification, comparing the results before and after conducting instruction tuning. As shown in Figure 5, instruction tuning with the target datasets improves the performance of both Video-LLMs, with TimeChat achieving a 4.6-fold improvement in the R@0.5 metric on ActivityNet-Captions. However, both Video-LLMs still struggle to confirm their initial predictions accurately. Surprisingly, TimeChat shows declined ROC-AUC scores even after conducting instruction tuning. In summary, while instruction tuning with target datasets improves the Video-LLMs' temporal grounding capabilities, it does not lead to improved consistency.

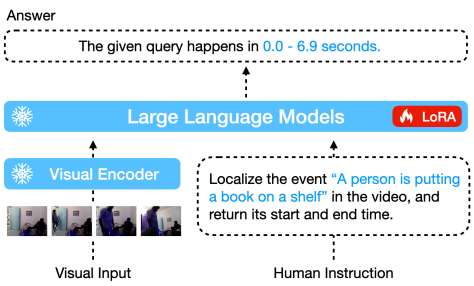

Figure 4: An example of video instruction tuning.

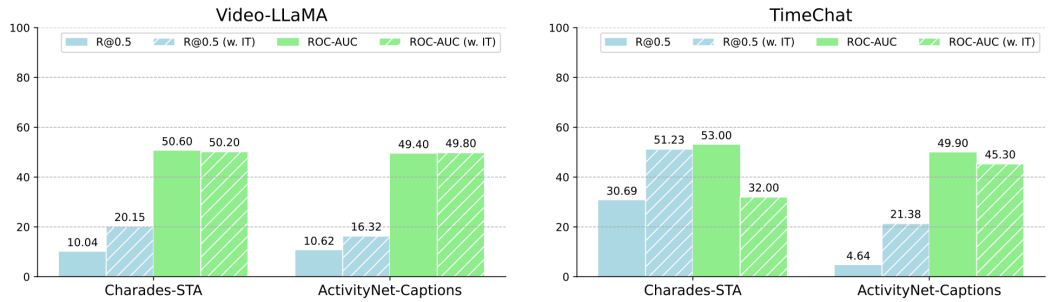

Figure 5: Comparison of the results of Video-LLMs in temporal grounding and self-answer verification before and after conducting instruction tuning with the target datasets (*i.e.*w. IT). After instruction tuning with the target datasets, both Video-LLMs demonstrate enhanced grounding abilities. However, this improvement does not translate into better consistency, as there are no significant changes in the self-answer verification scores.

3.2 Limitations of Naive Video Instruction Tuning

As illustrated in Figure 4, most Video-LLMs [3, 5, 4, 16, 18, 19, 13, 20, 21] primarily focus on constructing a large scale video dataset for instruction tuning to generate answers in a specific format, often using task-specific templates. Specifically, to perform temporal grounding tasks, templates like "`The given event occurs from start to end seconds.`" are pre-defined to facilitate easy extraction of timestamps from the model's response. However, this approach emphasizes generating answers over developing the model's ability to engage in nuanced reasoning or providing a rationale based on true understanding. Specifically, Video-LLMs might be overfitted and rely on shortcut strategies, like identifying certain frames or using common objects as cues, rather than understanding the temporal or compositional structure of the video. In conclusion, while Video-LLMs might appear to perform well according to traditional metrics and excel at specific tasks, this does not necessarily mean their predictions are based on genuine video comprehension.

4 Related Work

Video-LLMs. Recent studies have integrated visual information into Large Language Models (LLMs) [15, 22–24] to augment their multimodal reasoning capabilities. Beyond considering a single image input [17, 25, 26], video large language models (Video-LLMs) [18, 16, 19, 13, 20] have seamlessly connected between visual perception ability in vision encoders (*e.g.*, ViT [27]) and the powerful capabilities of LLMs through instruction tuning with massive video datasets. This enables Video-LLMs to perform various video understanding tasks, including Video Question Answering [28, 29], yet they still struggle to perform temporal reasoning and capture details in specific video moments. As a result, recent Video-LLMs [3–7] have been proposed to address this limitation. Unlike previous Video-LLMs, they can accurately localize the timestamps of events within a given video (*i.e.*, Video Temporal Grounding [2]) and also provide video segment-level details with corresponding timestamps in the video (*i.e.*, Dense Video Captioning [9, 10]). They develop effective temporal representation and newly construct video instruction tuning datasets for time-related video understanding tasks. For instance, TimeChat [3] proposes TimeIT, a time-aware instruction tuning dataset, and encodes video frames with the corresponding timestamp descriptions. VTimeLLM [4] designs a three-stage training framework to enhance the model's capabilities to understand sequential video frames. Momentor [5] designs a temporal perception module to express precise temporal positions and conducts a grounded event sequence modeling to facilitate multi-event comprehension in the video.

Benchmarking Video-LLMs. While Video-LLMs have shown remarkable advancements in various video understanding tasks, several studies [13, 30–32] raised concerns that evaluating existing benchmarks fail to assess temporal perception ability of Video-LLMs in various aspects. For instance, MVBench [30] introduces 20 challenging tasks that require a wide range of temporal understanding skills from perception to cognition. TempCompass [31] constructs a benchmark to comprehensively evaluate the temporal perception ability of Video LLMs, ensuring they cannot rely on shortcuts,

such as single-frame bias or language priors, to provide answers. Although these benchmarks have challenged the capabilities of Video-LLMs, they are primarily limited to multi-choice question-answering, asking directions or objects in the video. While similar tendencies of inconsistent behavior in LLMs have been discussed in previous studies, they focused on either the text [33–36] or the image-text [37, 38] levels. Unlike these benchmarks, our goal is to investigate whether Video-LLMs can accurately respond to queries requiring timestamps in videos, confirming that their answers are genuinely grounded in video comprehension.

Video Temporal Grounding. Video temporal grounding (VTG) [2, 8], which is one of the challenging video understanding tasks, aims to retrieve specific video moments corresponding to a given query sentence. This task requires sensitivity to temporal dynamics and fine-grained understanding skills, such as dense video captioning [39, 40], video corpus moment retrieval [41, 42], and highlight detection [43]. Previous methods [44–47] have been proposed to address VTG, but they are specialized and hard to generalize across multiple tasks. Some studies [39, 40, 48] have tackled several video understanding tasks by pre-training on large video datasets, yet they do not integrate LLM's capabilities to perform video understanding tasks. In this work, we delve into analyzing whether Video-LLMs faithfully perform the VTG task and comprehend videos, rather than proposing new architectures or training methods to improve performances.

5 Conclusion

In this paper, we investigate whether the temporal predictions of existing Video-LLMs are truly based on video language comprehension. To achieve this, we construct evaluation sets and design a series of tasks to assess the consistency of Video-LLMs. Our evaluation reveals that most Video-LLMs exhibit inconsistent answers, indicating that their predictions are not genuinely grounded in video comprehension. Furthermore, while Video-LLMs show improved temporal grounding performance after instruction tuning with target data, they continue to struggle with providing consistent answers based on their initial predictions. We conjecture that this is due to instruction tuning in existing Video-LLMs, which drives models to follow specific answer formats rather than improving consistency. For future work, we will further develop more comprehensive evaluation probes and analyze more Video-LLMs. Also, we will explore method to improve the inconsistent behavior of Video-LLMs. We hope this work can spark more future Video-LLMs that focus on making reliable predictions rooted in faithful video comprehension, and also more related benchmarks will be developed to support this goal.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its NRF Fellowship for AI (NRF-NRFFAI1-2019-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. Additionally, this work is partly supported by the IITP (RS-2021-II212068-AIHub/10%, RS-2021-II211343-GSAI/15%, RS-2022-II220951-LBA/15%, RS-2022-II220953-PICA/20%), NRF (RS-2024-00353991-SPARC/20%, RS-2023-00274280-HEI/10%), and KEIT (RS-2024-00423940/10%) grant funded by the Korean government.

References

[1] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[2] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.

[3] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024.

[4] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024.

[5] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv preprint arXiv:2402.11435*, 2024.

[6] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *arXiv preprint arXiv:2405.13382*, 2024.

[7] Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. Hawkeye: Training video-text llms for grounding text in videos. *arXiv preprint arXiv:2403.10228*, 2024.

[8] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pages 184–195. Springer, 2014.

[9] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, 2017.

[10] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[11] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13204–13214, 2024.

[12] Junbin Xiao, Nanxin Huang, Hangyu Qin, Dongyang Li, Yicong Li, Fengbin Zhu, Zhulin Tao, Jianxing Yu, Liang Lin, Tat-Seng Chua, and Angela Yao. Videoqa in the era of llms: An empirical study. *arXiv preprint arXiv:2408.04223*, 2024.

[13] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.

[14] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[15] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[16] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL `https://arxiv.org/abs/2304.08485`.

[18] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[19] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[20] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. 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*, pages 13700–13710, 2024.

[21] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023.

[22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[24] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[25] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL `https://arxiv.org/abs/2305.06500`.

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[27] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[28] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.

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

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

[31] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024.

[32] Shicheng Li, Lei Li, Shuhuai Ren, Yuanxin Liu, Yi Liu, Rundong Gao, Xu Sun, and Lu Hou. Vitatecs: A diagnostic dataset for temporal concept understanding of video-language models. *arXiv preprint arXiv:2311.17404*, 2023.

[33] Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. Benchmarking and improving generator-validator consistency of language models. *arXiv preprint arXiv:2310.01846*, 2023.

[34] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[35] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

[36] Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Prompt consistency for zero-shot task generalization. *arXiv preprint arXiv:2205.00049*, 2022.

[37] Tongtian Yue, Jie Cheng, Longteng Guo, Xingyuan Dai, Zijia Zhao, Xingjian He, Gang Xiong, Yisheng Lv, and Jing Liu. Sc-tune: Unleashing self-consistent referential comprehension in large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13073–13083, 2024.

[38] Yuan Zhang, Fei Xiao, Tao Huang, Chun-Kai Fan, Hongyuan Dong, Jiawen Li, Jiacong Wang, Kuan Cheng, Shanghang Zhang, and Haoyuan Guo. Unveiling the tapestry of consistency in large vision-language models. *arXiv preprint arXiv:2405.14156*, 2024.

[39] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022.

[40] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726, 2023.

[41] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 447–463. Springer, 2020.

[42] Minjoon Jung, SeongHo Choi, JooChan Kim, Jin-Hwa Kim, and Byoung-Tak Zhang. Modal-specific pseudo query generation for video corpus moment retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[43] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

[44] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12870–12877, 2020.

[45] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020.

[46] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2613–2623, 2022.

[47] Minjoon Jung, Youwon Jang, Seongho Choi, Joochan Kim, Jin-Hwa Kim, and Byoung-Tak Zhang. Background-aware moment detection for video moment retrieval. *arXiv preprint arXiv:2306.02728*, 2024.

[48] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2794–2804, 2023.

[49] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016.

[50] Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *Proceedings of the 2nd international workshop on human-centric multimedia analysis*, pages 13–21, 2021.

[51] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A Proposed Evaluation Sets

In this section, we describe how we construct evaluation sets for our experiments. To construct our evaluation sets, we follow the steps below:

1. First, we collect videos from two benchmarks: Charades-STA [2] and ActivityNet-Captions [9]. Charades-STA is based on the Charades dataset [49], focusing on indoor human activities. The average length of the videos is 30 seconds. ActivityNet-Captions features much longer videos than Charades-STA, averaging 2 minutes. The videos are sourced from YouTube and show outdoor human activities.
2. However, there are some noisy annotations in datasets that might cause misleading evaluation, as pointed out in [50]. Therefore, we filter out an annotation if the timestamp is too long (*i.e.*, covering longer than 70% of the total length of the video), too short (*i.e.*, the length is less than 5 seconds), or where the query sentence is too short (*i.e.*, the number of words is less than 5). After filtering the annotations, we curate 500 videos for each dataset and this leads to 707 and 1422 query-moment pairs for Charades-STA and ActivityNet-Captions, respectively.
3. As shown in Figure 6, we then generate aligned and misaligned sentences and compositional information using GPT-4o-mini [15] from the annotations in the test split for each dataset.
4. After generating aligned and misaligned sentences and compositional information, we conduct a human evaluation on 700 samples ($\approx$ 30% of the whole dataset) to ensure that they are appropriately generated for a reliable evaluation. To do this, we ask the annotators to decide whether the generated sentence is correctly generated from the original sentence. We found that 94% of generated sentences are well generated as we intended.

B Details of Models and Experiment Settings

In this section, we explain each Video-LLM and experiment setting details.

1. **Video-LLaMA** [16] is a Video-LLM that understands both visual and audio information in the video. Video-LLaMA exhibits two branches: Vision-Language and Audio-Language in its modeling and utilizes cross-modal training from both the frozen pre-trained visual and audio encoder. Video-LLaMA shows a remarkable zero-shot audio understanding capability and also generates responses in the visual and audio information presented in the videos. We use the checkpoints released at `https://github.com/DAMO-NLP-SG/Video-LLaMA`. Specifically, we select the fine-tuned checkpoints, which are trained on the instruction tuning data from Mini-GPT-4 [51], LLaVA [17], and VideoChat [18].
2. **Video-ChatGPT** [13] designs spatiotemporal video modeling and constructs video instruction tuning upon LLaVA [17]. It introduces a new dataset for video instruction tuning, containing 100,000 high-quality video-instruction pairs. Video-ChatGPT outperforms previous Video-LLMs in Zero-shot VQA across several benchmarks. Additionally, Video-ChatGPT proposes a video conversation evaluation framework. We use the checkpoints released at `https://github.com/mbzuai-oryx/Video-ChatGPT`.
3. **TimeChat** [3] is a time-sensitive multimodal LLM, specifically developed to accurately localize and understand specific video moments from long videos. TimeChat designs two key architectural: (1) A time-aware frame encoder that explicitly encodes video frames along with timestamps, (2) A sliding video Q-Former to accommodate sequential information in video frames. Additionally, TimeChat constructs 125K video instruction tuning datasets to perform time-related video understanding tasks, such as VTG and DVC. We use the checkpoints released at `https://github.com/RenShuhuai-Andy/TimeChat`.
4. **VTimeLLM** [4] proposes a three-stage temporal-aware method, including image-text training and understanding events within the video, enabling more precise video temporal understanding. VTimeLLM devises two types of QA dialogue templates, including single-turn and multi-turn, to prompt questions requiring a comprehensive description of all events and their corresponding timestamps. We use the checkpoints released at `https://github.com/huangb23/VTimeLLM`.

We employ Vicuna-v1.5 [24] for the language model backbone. When performing the VTG task for Video-LLaMA and Video-ChatGPT, we use the question template "Please answer when the q occurs in the video. The output format should be: 'start - end seconds.' Please return its start time and end time." For TimeChat and VTimeLLM, we follow the same question templates, which are used in their instruction tuning, and also use the official codes to extract the timestamps from their predictions.

You are an intelligent chatbot designed for generating and decomposing sentences. You are an intelligent chatbot designed for generating and decomposing sentences. Your task is to generate aligned and misaligned sentences, as well as compositional information, based on the input sentence. Ensure that the generated sentences stay within the context of the provided sentence without introducing any new information.

INSTRUCTIONS: Both aligned and misaligned sentences should be natural and realistic, without introducing any implausible or unrealistic information. If you find that the generated sentences using the above techniques are unnatural, you may skip generating them. Ensure that generate at least three aligned and misaligned sentences and the number of aligned and misaligned sentences should be equal.

- Aligned Sentences (A): Follow below tecniques.
  1. Replace Words: Replace key nouns and verbs while maintaining the sentence's meaning.
  2. Active to Passive: Convert active sentences to passive voice.
  3. Word Order Changes: Rearrange the word order while preserving the original meaning.
- Misaligned Sentences (M): Use the same techniques as above, but alter the meaning so that the sentence no longer releavnt to the original context.
- Compositional Information (C): Your task is analyze the given query sentence and decompose it into its fundamental components:
    1. Subject: Identify the primary entities or characters mentioned in the sentence.
    2. Action: Describe the actions being performed by the subjects or what is happening to them.
    3. Relation: Determine the relationship between the subjects and other elements in the sentence, such as objects, locations, or other entities.

Based on the extracted components, create a series of yes/no questions:
    1. Positive Questions (Y): Ensure these questions are directly aligned with the original sentence, capturing its true meaning.
    2. Negative Questions (N): Modify the sentence meaningfully to generate misaligned questions, ensuring the correct answer is 'no.'.
Ensure that all generated questions are coherent, natural, and contextually appropriate.

The output should be formatted as Python dictionary style as follows in example:
Input sentence: "She is surrounded by two other accordions as she instructs on how to play the instrument."
Output:
```
{
  "A": [
    "She is encircled by two other accordions while she teaches how to play the instrument.",
    "As she instructs on how to play the instrument, two other accordions surround her.",
    "Two accordions are near her as she gives instructions on playing the instrument.",
  ],
  "M": [
    "He is surrounded by two guitars as she instructs on how to play the instrument.",
    "She is alone on stage as she instructs on how to play the piano.",
    "Two accordions are lying on the floor while she plays a guitar.",
  ],
  "C": {
    "Y": ["Is she surrounded by accordions?", "Does she instruct on playing the instrument?"],
    "N": ["Is she surrounded by guitars?", "Does she assemble instruments?"],
  }
}
```

Now it's your turn.

Please write the answer based on the given sentence.

DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string.
Input sentence: "{sentence}"
Output:

Figure 6: GPT-4o-mini APT prompt to generate aligned and misaligned sentences and compositional information.

C Fine-tuning Details

To conduct fine-tuning Video-LLaMA and TimeChat on Charades-STA and ActivityNet-Captions, we first collect the annotations in the train split for each dataset and convert the annotations into a task-specific template to derive Video-LLMs can predict the timestamps. For example, if the query "The person closes the laptop." is grounded in 0 to 5 seconds in the video, we prompt the Video-LLMs "Localize the visual content described by the given textual query 'The person closes the laptop.' in the video, and output the start and end timestamps in seconds.". Then the model's answer should be "The given query happens in 0 - 5 seconds." We use the official codes and configurations to conduct instruction tuning upon their official checkpoints using four Quadro RTX 8000 GPUs.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly addressed our contributions in the introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of our work in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our work does not introduce hypotheses and proofs of theory.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have detailed the experiment settings of Video-LLMs in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We plan to release our codes upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have mentioned which checkpoints and configurations are used in our experiments for each Video-LLM.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conduct experiments with five different random seeds and report the mean score.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have addressed the details of reproducing the Video-LLMs in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Cods of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Our work does not have those impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: Our work does not posses such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all references in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

Justification: We have addressed how we construct our evaluation sets in the Evaluation section and Appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.