Divide and Conquer Long Video Understanding by LLMs

Paul Yang
MSCS
pwy@unc.edu

Hao Chen
MSCS
chhao@unc.edu

Dan Peng
BSCS
danpeng@unc.edu

Abstract

Current video recognition models are limited to processing approximately 2-minute clips, making it challenging to effectively understand longer content. Our goal is to extend this limit to 10 minutes by segmenting videos and leveraging existing video understanding models such as VideoChat and Video-ChatGPT. Instead of increasing computational costs by expanding model limits, we want to propose an efficient approach: dividing long videos into shorter segments, using LLM-based models to summarize each segment, and then combining these summaries for a complete understanding. This method allows us to efficiently extend long video understanding while extracting both high-level and detailed information as needed. Our code is available at https://github.com/GreatDanPeng/Divide-Conquer-VideoQA.

1. Introduction

The motivation for this work stems from challenges encountered in summarizing long-form content, particularly in the context of video understanding. This idea draws inspiration from prior experiments in natural language processing (NLP), where token limits in language models posed significant obstacles to summarizing lengthy texts, such as entire books or detailed reports. Inspired by the effectiveness of the divide-and-conquer strategy [9], which has been widely used in computational tasks, this approach was adapted to handle long-form content by breaking it into manageable segments, summarizing each, and then merging the summaries into a cohesive final output. While this specific adaptation for video summarization is novel to this work, the general strategy aligns with prior applications in areas such as text summarization and hierarchical data processing

Recognizing the parallels between summarizing long texts and videos, we applied a similar approach to tackle the inherent complexity of video summarization. This method draws inspiration from the divide-and-conquer strategies employed in long document summarization, where documents are segmented into smaller parts, each summarized individually, and then combined into a cohesive summary. For instance, Gidiotis and Tsoumakas [4] introduced such a technique for long document summarization. Videos, like long texts, often exceed the processing limits of current models due to their sequence length and multimodal nature. Furthermore, the lack of straightforward methods to condense visual and textual data from videos into concise summaries presented an additional challenge. Inspired by our success in NLP, we adapted the divide-and-conquer method to segment videos, summarize each clip, and iteratively merge the summaries. This approach aims to balance computational efficiency, scalability, and the retention of essential information.

Current video recognition and understanding models are limited by their processing capacity, typically handling up to approximately two minutes of video content at a time. While this aligns with the human ability to focus on short segments, it poses significant challenges for understanding longer videos, such as movies or extended content, where critical information often spans beyond this duration. This limitation hinders comprehensive video analysis, especially when the narrative or context unfolds over longer periods. To address this issue, our work extends the maximum processing limit to 10 minutes by segmenting longer videos into manageable parts. This segmented approach not only accommodates the inherent constraints of existing models but also ensures a cohesive understanding of extended video content by iteratively summarizing and merging segments into a unified output.

2. Related Work

Recent advancements in video understanding have led to the development of chat-centric systems that integrate video foundation models with large language models (LLMs) to enhance spatiotemporal reasoning and conversational capabilities. For instance, VideoChat [8] employs a learnable neural interface to connect these models, excelling in event localization and causal relationship inference. It utilizes a video-centric instruction dataset comprising thousands of videos with detailed descriptions and conversations to fine-tune the system, demonstrating potential across various video applications.

Similarly, Video-ChatGPT [12] merges a video-adapted visual encoder with an LLM to facilitate detailed video-based conversations. Trained on a dataset of 100,000 video-instruction pairs acquired through a scalable manual and semi-automated pipeline, it effectively captures temporal dynamics and frame-to-frame consistency. Additionally, it introduces a quantitative evaluation framework to objectively assess the strengths and weaknesses of video-based dialogue models.

While these approaches focus on enhancing model capacities and integrating conversational elements, they often involve increasing model complexity and computational costs to handle longer video sequences. In contrast, our method adopts a divide-and-conquer strategy by segmenting videos into lengths manageable by current models and iteratively summarizing these segments. This approach emphasizes efficiency and simplicity, aiming to merge content effectively without substantially escalating computational expenses. By concentrating on the aggregation of segmented video content, our method offers a streamlined alternative to existing techniques, providing a practical solution for comprehensive video understanding without the need for extensive model scaling.

3. Methods

3.1. Models

We will use the pretrained models (VideoChat [8] and VideoChatGPT [12]) for video understanding. The recent research focuses on understanding long videos by improving the model's max limit and adding more expensive costs, but we try to focus on how the content from the videos can be merged and conquered with a simple and efficient method.

After tracking where the information is in the video from the responses generated by LLM, we consider looking back to the video and extracting information from the segment of the video. Thus, based on Fig.1, we could know what the whole video is talking about and the specific details the user want to know.

3.2. Divide-and-Conquer (DAC)

Divide the video into the length that the current model can accept and conquer the summary of the short videos into the long video to understand what the full video is talking about.

Divide the video into segments. Dealing with over 10-minute videos with text summarization, LLM has a limit of tokens in one-time input. It's very difficult to generate a summary from such long context. To deal with this issue, we could divide a long video into segments and conquer it after we summarize it. We plan to use the overlapping strategy of dividing (Fig.2), where the information could be regarded as logically connected. Overlapping segments may help retain cohesive information across boundaries.

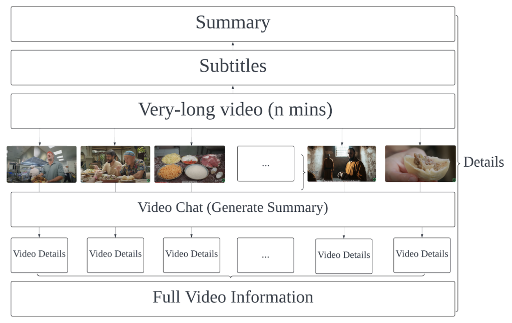


Figure 1. High-Level Overview of the Proposed Approach for Summarizing Long Videos

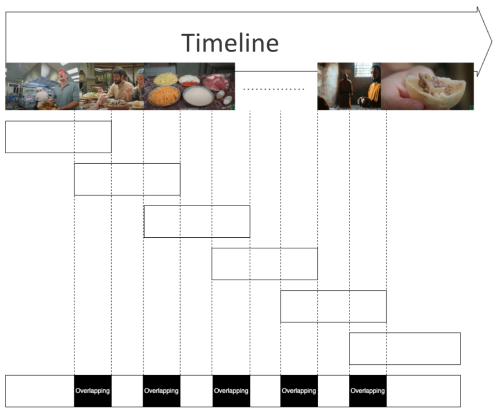


Figure 2. Divide the Video Into Overlapping Segments.

For a long video, we use 15s, 30s, 45s, 60s as time length to divide it into clips. For each clip, we sample 8, 16, 24, 32 frames to represent the contents of the video segments. We consider overlap as a setting to change the basic length of the frames selections. If the overlap is set, the frames selection of a segment would add 20% frames of its previous segment's tail to its head.

For time selection, we also tested clips shorter than 10s, but the results are not so good. Too short clips would generate comparably independent segments to make models misunderstand the meaning of these clips. For example, a 5s clip about forest fire only contains the elements like red color, fire, Chinese words subtitles. So the model thought the scene is about Chinese spring festival and fireworks, which are totally unrelated with the actual contents.

After diving the videos into small segments and extract frames, we leverage model VideoChat with prompt ”This is a segment of a long video. Please describe it detailedly.” as our question to generate corresponding text results of every video clip. We then store these text results in different json files for the later Conquer part and Evaluation part.

Conquer and merge the summarization by LLM. The core innovation of our approach lies in the iterative merging process, which we term the "conquer" stage. This stage is designed to progressively merge segment-level summaries into a single, cohesive representation of the entire video. After segmenting the video into smaller clips and generating individual summaries for each clip, adjacent summaries are merged iteratively. This process allows for scalable summarization of videos, regardless of their length.

The merging process offers two primary approaches: 2-by-2 merging and 3-by-3 merging. In the 2-by-2 approach, summaries of two adjacent segments are combined to form a higher-level summary. For instance, summaries of Clip A and Clip B are merged to produce a new summary, Summary AB, which encapsulates the content of both segments. This process is repeated across multiple levels until only one comprehensive summary remains, effectively representing the entire video. On the other hand, the 3-by-3 approach merges summaries from three segments at a time, allowing for faster processing by reducing the total number of iterations. While the 2-by-2 method retains finer granularity and is better suited for shorter or highly detailed videos, the 3-by-3 method prioritizes efficiency, making it ideal for extremely long videos.

We leverage both GPT-3.5 [13] and 4o-mini [14] models during the "conquer" stage to optimize the merging process. GPT-3.5 is employed for its speed and capability to process segment summaries efficiently, while 4o-mini is utilized for its enhanced ability to provide more nuanced and detailed summaries. This combination ensures that summaries at each merging step remain both concise and contextually rich, striking a balance between computational efficiency and information retention. Furthermore, designing effective prompts for GPT-3.5 is a critical step in ensuring that the model accurately captures essential visual and textual features during each summarization phase.

The prompt "Merge the following two segments into a single cohesive summary. Ensure the content flows naturally and feels like one unified narrative, without any clear separation between the original parts." is designed to guide the language model in combining two segment summaries into a seamless and unified output. The goal is to eliminate any sense of disjointedness or segmentation, ensuring that the final merged summary reads as though it were generated from a single, continuous piece of content rather than from two distinct parts.

In this approach, each segment's pre-generated textual summary is "conquered" by being iteratively summarized. Instead of directly analyzing video features, we focus on refining and merging textual summaries to preserve essential information from the video. The process involves analyzing potentially thousands of pre-generated summaries and merging them two by two or three by three to create progressively broader context representations. This strategy ensures that the critical information is preserved at every level of summarization, overcoming the token and sequence length limitations of large language models. By the end of the process, a single comprehensive summary encapsulates the full content of the video.

Through this iterative conquer process, we effectively reconstruct the full context of a long video by integrating and refining the essential information from its individual segments. Figure 3 provides a visual representation of the 2-by-2 merging workflow, illustrating how smaller segment summaries are progressively combined into a cohesive whole. This method highlights the adaptability of large language models like GPT-3.5 and 4o-mini in handling large-scale video summarization tasks, showcasing the potential of such models in efficiently summarizing and understanding long-form content.

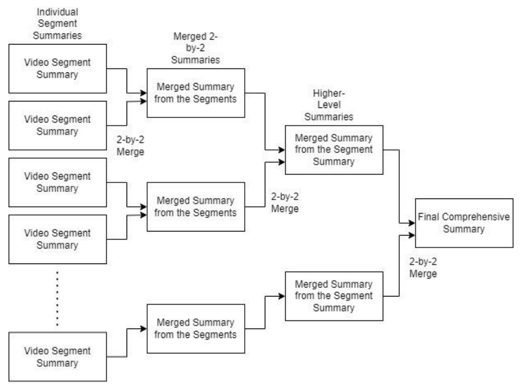

Individual Segment Summaries
Merged 2-by-2 Summaries
Higher-Level Summaries
Video Segment Summary
Video Segment Summary
Video Segment Summary
Video Segment Summary
Video Segment Summary
Merged Summary from the Segments
Merged Summary from the Segments
Merged Summary from the Segments
2-by-2 Merge
Merged Summary from the Segment Summary
Merged Summary from the Segment Summary
2-by-2 Merge
Final Comprehensive Summary
2-by-2 Merge

Figure 3. Conquer two by two

4. Experiments

4.1. Model Deployment

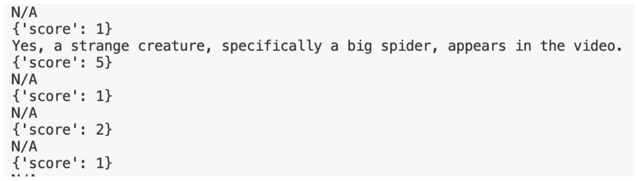
In our experiments, we implemented previous methods, VideoChat [8] with Vicuna-7B weights [19] and VideoChat-GPT [12] with LLaVA-7B weights [10] by deploying a single V100 GPU with 32G memory. Since they don't officially support long videos with over 10k frames, we uniformly sample frames from the original video up to their max frames in Table 1. We also implemented our DAC method based on VideoChat pretrained 7B model and test them with long video QA pairs on Moviechat dataset [17].

4.2. Dataset & Data Loader

We designed new data loaders for VideoChat and VideoChatGPT to enable them to divide longer video (over 8 mins) into small clips. For each clip, we fixed frame-per-second (FPS) rate at 8/15s in Table 1.

Methods	Max Frames
VideoChat	32
VideoChatGPT	100
VideoChat-DAC	8/15s

Table 1. Max frames across VideoChat-DAC and other methods



```
N/A
{'score': 1}
Yes, a strange creature, specifically a big spider, appears in the video.
{'score': 5}
N/A
{'score': 1}
N/A
{'score': 2}
N/A
{'score': 1}
```

Figure 4. Evaluation example by GPT3.5-turbo.

For each test video, we have 5 questions and corresponding ground truth answers. We have general question about the movie style and detailed question about the number of people appearing in the video. The average video size is around 130MB and we randomly selected only 20 videos for testing due to the limited cloud storage.

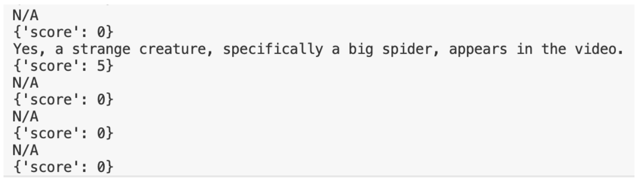
4.3. Evaluation Metrics

We use the following five benchmarks to evaluate long video understanding performance:

- **Correctness of Information (CI):** Measures the factual accuracy of the model's responses.
- **Detailed Orientation (DO):** Evaluates how comprehensively the model captures fine-grained details.
- **Contextual Understanding (CU):** Assesses the model's ability to grasp the broader video context.
- **Temporal Understanding (TU):** Tests the model's ability to understand the event sequences in the video.
- **Consistency (CO):** Evaluates the logical coherence of the model's answers across multiple questions.

During the former evaluation, we followed the long video benchmark table from MovieChat [18], which evaluate the previous methods by using the average score of GPT-3.5 [13], claud[2], and human blind rating. Since we only have access to OPENAI api keys and it's hard for us to do human blind rating, we tried only using GPT-3.5 [13] to rate our models first. We found our model achieved insanely high correctness score over 4.5/5 led by incorrect evaluation of GPT-3.5 [13] in Fig 4.

To better evaluate the long video understanding performance with five metrics (CI, DO, CU, TU, CO), we leveraged GPT-4omini [14] which gives accurate scores in Fig 6. We designed prompts that require GPT 4omini to rate 0 points first if they can't find any answer from the QA results.



```
N/A
{'score': 0}
Yes, a strange creature, specifically a big spider, appears in the video.
{'score': 5}
N/A
{'score': 0}
N/A
{'score': 0}
N/A
{'score': 0}
```

Figure 5. Evaluation example by GPT4o-mini.

4.4. Short Video Understanding

For the project milestone, we tested VideoChat and Video-ChatGPT with and without Divide-and-Conquer (DAC) on a 1-min video about polo game curated from the ActivityNet-200 videos (Table. 9) for their basic understanding about a polo game.

4.5. Long Video Understanding

For our base VideoChat-DAC model in Table 2, which divides video into 30s non-overlap clips, and conquers by GPT-3.5, it doesn't perform as well to capture factually correct information, figure out the temporal sequence of correct answer, and generate consistent correct answers as videochat does. To our expectation, it does well in Detailed orientations by capturing more rich details. However, our model VideoChat-DACo by summarizing overlapping clips works better than base model with almost the same settings. It achieves highest scores among 4 metrics.

Method	CI	DO	CU	TU	CO
VideoChat	3.05	2.64	2.84	2.89	**3.02**
VideoChatGPT	2.34	2.17	2.26	2.31	2.45
VideoChat-DAC	2.66	2.78	2.83	2.71	2.64
VideoChat-DACo	**3.08**	**2.99**	**3.00**	**3.02**	2.89

Table 2. Long Video Understanding Benchmark: CI, DO, CU, TU, and CO across different methods. Our default VideoChat-DAC uses 30s non-overlap segment time with GPT-3.5 for conquering.

4.6. Ablation Studies

In our ablation studies, we evaluated how the length of video clips influences model performance in Table 3. Despite keeping the FPS rate fixed, the total information captured by the model varied due to the attention-based mechanism by VideoChat, which processes frames hierarchically. When sampling shorter 15s clips, the model achieved superior performance in terms of correctness of information, indicating that shorter clips may provide more focused and precise representations for factual accuracy. However, when using 30s clips, the model demonstrated the best overall performance across benchmarks, suggesting that this clip length achieves a balance between maintaining detail and contextual coverage. According to Table 3, we think the temporal understanding appeared to decay for clips longer than 30 seconds with 16 frames.

SegTime	CI	DO	CU	TU	CO
15s	**2.74**	2.72	2.73	2.68	2.55
30s	2.66	**2.78**	**2.83**	**2.71**	**2.64**
45s	2.51	2.61	2.66	2.63	2.47
60s	2.64	2.57	2.61	2.61	2.45

Table 3. Long Video Understanding Benchmark: CI, DO, CU, TU, and CO across different segment time.

According to Table 4, sampling overlapping clips can improve the VideoChat-DAC's performance over all metrics of long video understanding by average 0.2.

w/o Overlap	CI	DO	CU	TU	CO
No Overlap (30s)	2.66	2.78	2.83	2.71	2.64
Overlap (30s)	**3.08**	**2.99**	**3.00**	**3.02**	**2.89**

Table 4. Long Video Understanding Benchmark: CI, DO, CU, TU, and CO across w/o overlapp.

We evaluated the effectiveness of dividing video clips with segmented captions while relying on the LLM agent to extract answers directly, without proceeding to the conquer step in the DAC method in the Table 5. This resulted in only marginal differences in performance between the Divide and DAC methods for benchmarks like correctness, detail orientation, and consistency, indicating that the divide step alone can achieve comparable accuracy in these metrics. However, including summarization mechanism in DAC significantly improved the model's performance in contextual and temporal understanding. This demonstrates the value of incorporating LLM summarization in long video understanding tasks.

w/o Conquer	CI	DO	CU	TU	CO
Divide	2.62	2.74	2.63	2.61	**2.65**
Divide-and-Conquer	**2.66**	**2.78**	**2.83**	**2.71**	2.64

Table 5. Long Video Understanding Benchmark: CI, DO, CU, TU, and CO across w/o conquer.

In contrast to the findings of our LLM-assisted evaluation, GPT-3.5 [13] performs better in summarizing captions than GPT-4omini [14]. As shown in Table 6, GPT-3.5 consistently outperformed GPT-4omini across several benchmarks, which indicates its superior ability to summarize segmented captions and extract key information to answer the test questions. Based on the previous arguments, GPT-3.5 showcases its powerful summarization including key features requested from questions one by one, but fails to figure out the priority of multiple requirements.

Our last ablation study in Table 7 compares 2by2 and 3by3 summarization strategies, where 2by2 performs better overall. This strategy balances detail preservation

ConquerLLM	CI	DO	CU	TU	CO
GPT-3.5	2.66	**2.78**	**2.83**	**2.71**	**2.64**
GPT-4omini	**2.68**	2.63	2.69	2.54	2.40

Table 6. Long Video Understanding Benchmark: CI, DO, CU, TU, and CO across different LLMs for conquering (summarization).

and context merging, ensuring higher accuracy and coherence. While 3by3 slightly improves temporal understanding (TU), it sacrifices performance in other metrics like DO and CU, likely due to information loss or contradictions when merging more chunks. These results demonstrate 2by2 as the right default setting to handle sufficient token and sequence while maintaining key information.

ConquerSeg	CI	DO	CU	TU	CO
2by2	**2.66**	**2.78**	**2.83**	2.71	**2.64**
3by3	2.56	2.61	2.72	**2.74**	2.62

Table 7. Long Video Understanding Benchmark: CI, DO, CU, TU, and CO across different summarization methods of segments.

5. Discussion

In this study, we designed a specific prompt to guide the merging process of segment-level summaries, ensuring logical cohesion and narrative continuity. The results demonstrate the effectiveness of this approach in producing natural and unified summaries. However, the impact of prompt variations on the quality of the merged summaries remains an open question.

Recent research has explored the impact of prompt engineering on the performance of OpenAI's large language models (LLMs). For instance, Clavié et al. [3] investigated job classification tasks using GPT-3.5-based models, finding that well-designed prompts significantly enhance model performance, with specific wording critically influencing outcomes. Similarly, Shin et al. [16] assessed GPT-4 in automated software engineering tasks, concluding that task-specific prompts can outperform fine-tuned smaller models in certain scenarios. These studies underscore the importance of prompt design in optimizing LLM outputs.

In Table 8, the inference time of our models is too high. Longer segments required processing a greater number of frames, which increased computational overhead and led to performance bottlenecks. This constraint limits the scalability of our model for large-scale datasets or real-time applications.

To address this issue, we propose the following improvements based on related work:

1. Asynchronous Processing: By enabling parallel processing of multi-modal data such as text, images, audio, and video, we can efficiently utilize computational resources and reduce processing delays. Non-blocking task execution allows simultaneous summarization of different video segments, significantly lowering inference time [7].

2. Early Stopping Criteria: We propose implementing a quality-based stopping mechanism that halts the inference process when a summary meets predefined evaluation standards. This reduces unnecessary computation and prevents redundant iterations during the merging process [15].

3. Video Segmentation Improvement: Enhancing video segmentation accuracy with real-time adaptation techniques such as cross-unit deployment ensures that only relevant segments are processed. This reduces the number of irrelevant frames, improving both computational efficiency and inference time [11].

4. Adaptive Frame Sampling: To optimize frame selection, we propose using adaptive frame sampling based on scene detection, motion analysis, or query relevance. This strategy would dynamically sample only the most informative frames, reducing computational load while preserving essential context [5].

5. Sparse Attention Mechanism: Incorporating sparse-dense attention can enhance summarization by focusing on relevant frames and temporal features while ignoring less significant sections of the video. This approach effectively reduces processing time while maintaining high-quality summaries [6].

Another limitation observed during evaluation is related to the dataset's question-answer (QA) pairs. The current questions tend to target general information, which can be found in several minutes of the video. This reduces the effectiveness of the evaluation process when testing the system's ability to capture fine-grained details.

To enhance the evaluation process, we propose the following improvement:

Redesigning Detailed Questions: We suggest rewriting the QA pairs to include more context-specific and detailed questions targeting specific video events or moments that occur within a few seconds. This adjustment would create a more challenging evaluation benchmark and better assess the model's ability to capture granular information and contextual understanding [1].

This study emphasizes the importance of prompt design and computational efficiency in the context of video summarization using a divide-and-conquer (DAC) approach. The overall inference time of our VideoChat-DAC methods remains significantly higher compared to baseline models. This is particularly evident in shorter segment sizes, such as 15 seconds, where computational overhead becomes a critical limitation for scalability and practical deployment.

To address these challenges, future work will focus on optimizing the balance between segment lengths, computational cost, and summarization quality. By exploring diverse prompt designs and strategies such as asynchronous processing, adaptive segment merging, we aim to reduce the computational burden while maintaining the quality of generated summaries. These efforts will provide deeper insights into the interaction between prompt engineering and large language model outputs, enabling more efficient and scalable solutions for long video summarization tasks.

Methods	**InferenceTime**
VideoChat	17.6s
VideoChatGPT	20.3s
VideoChat-DAC(60s)	97.9s + 45.0s
VideoChat-DAC(45s)	106.5s + 68.8s
VideoChat-DAC(30s)	278.4s + 87.7s
VideoChat-DACo(30s)	259.2s + 93.2s
VideoChat-DAC(15s)	461.6s + 163.6s

Table 8. Inference time across VideoChat-DAC and other methods), where n stands for the number of questions.

6. Conclusion

Our approach breaks through the time limits of video understanding and can help expand models designed for short videos to 8 minutes or even more. At the same time, the different length of time of video segment and the setting of overlap or conquer would influence the accuracy of content understanding, among which the 30-second with overlap test is the best. However, our approach also brings the problem of computing time, although it does not require much GPU resources, but requires a lot of time for videos and texts processing compared with other models able to process long videos. Nevertheless, other models in video understanding [17] [12] would review the video for each QA prompt, while our method only need to do once and answer multiple questions.

References

[1] Toqa Alaa, Ahmad Mongy, Assem Bakr, Mariam Diab, and Walid Gomaa. Video summarization techniques: A comprehensive review, 2024. 6

[2] Anthropic. Claude: A next-generation ai assistant, 2023. 4

[3] Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. Large language models in the workplace: A case study on prompt engineering for job type classification, 2023. 5

[4] Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents, 2020. 1

[5] Sullam Jeoung, Goeric Huybrechts, Bhavana Ganesh, Aram Galstyan, and Sravan Bodapati. Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning, 2024. 6

[6] Hyun-Woo Kim and Yong-Suk Choi. Fusion attention for action recognition: Integrating sparse-dense and global attention for video action recognition. *Sensors*, 24(21), 2024. 6

[7] Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. Multi-modal summarization for asynchronous collection of text, image, audio and video. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1092–1102, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. 6

[8] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding, 2024. 1, 2, 3

[9] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In Ron Kaplan, Jill Burstein, Mary Harper, and Gerald Penn, editors, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920, Los Angeles, California, June 2010. Association for Computational Linguistics. 1

[10] Haotian Liu. Llava-7b-lightening-v1-1, 2023. 3

[11] Cheng Ma, Jingyi Zhang, Jie Zhou, and Jiwen Lu. Learning series-parallel lookup tables for efficient image super-resolution, 2022. 6

[12] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models, 2024. 2, 3, 6

[13] OpenAI. Gpt-3.5 technical documentation, 2023. Accessed: 2024-12-05. 3, 4, 5

[14] OpenAI. Gpt-4o mini model documentation, 2024. Accessed: 2024-12-05. 3, 4, 5

[15] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkilä. Rethinking the evaluation of video summaries, 2019. 6

[16] Jiho Shin, Clark Tang, Tahmineh Mohati, Maleknaz Nayebi, Song Wang, and Hadi Hemmati. Prompt engineering or fine tuning: An empirical assessment of large language models in automated software engineering tasks, 2023. 5

[17] Enxin Song. Moviechat-1k test dataset, 2023. 3, 6

[18] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding, 2024. 4

[19] LMSys Team. Vicuna-7b delta v0, 2023. 3

7. Appendix

Table 9. Prompting.

System Message You are a helpful assistant.
Prompting What is the video about?
Response **VideoChat:** The video shows a group of people riding horses and playing polo. The video shows several shots of people playing polo, as well as some shots of the fields where they play. It appears to be a leisurely, casual game with the people riding horses in a grass field. The horses are playing with each other and seem to be having a good time. **VideoChat (DAC):** The video descriptions collectively depict scenes of a large outdoor gathering with people seated on white chairs, watching events in a relaxed and casual setting, including a polo match. The polo scenes feature riders in white and red attire, competing on a green field surrounded by trees and buildings, giving a glimpse into competitive polo and horseback riding. The visuals highlight various aspects of the game, such as galloping horses, attempts to score goals, and close-ups of the polo balls, set against a green, open field backdrop with trees in the distance. **VideoChatGPT:** The video shows a game of polo being played on a field, with horses and players riding around and using long-necked bamboo sticks called ”quebradas” to hit a small ball in the air. It ends with the horses galloping by. **VideoChatGPT (DAC):** The video portrays two distinct scenes: first, a group of men watching a cricket game from a balcony, seated in white chairs; and second, a unique polo match where a man and a woman, both wearing masks and blue shirts, play polo within an enclosure for horses. The match concludes with the man scoring a goal, marking it as a successful game.

Figure 6. Sample Test Movie

Table 10. Sample Long Video QA Test Pairs

Sample Global Questions and Answers:
Question: Does the clip happen in the 21st century? **Answer:** No.
Question: Does it happen during day or night? **Answer:** Night.
Question: Do more than 5 people appear in the video? **Answer:** Yes.
Question: Is this clip a cartoon? **Answer:** No.
Question: Are there more than three different backgrounds appearing in the video? **Answer:** Yes.

Table 11. Evaluation Prompt of CI

System Message
”You are an intelligent chatbot designed for evaluating the factual accuracy of ” ”generative outputs for video-based question-answer pairs. Your task is to compare ” ”the predicted answer with the correct answer and determine if they are factually consistent. ” ”Here’s how you can accomplish the task:” ”- Focus on the factual consistency between the predicted answer and the correct answer.” ”- The predicted answer must be factually accurate and align with the video content.” ”- Consider synonyms or paraphrases as valid matches.” ”- Evaluate the factual accuracy of the prediction compared to the answer.”
Prompting
f”Please evaluate the following video-based question-answer pair:” f”Question: {`question`}” f”Correct Answer: {`correct_answer`}” f”Predicted Answer: {`pred_answer`}” ”Provide your evaluation only as a detail orientation score where the detail orientation score is an integer value between 0 and 5, with 5 indicating the highest level of detail orientation. ” ”Please generate the response in the form of a Python dictionary string with keys ’score’, where its value is the detail orientation score in INTEGER, not STRING.” ”DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string. ” ”For example, your response should look like this: ”score’: 4.8.”
Response
`{'score': 4}`

Method	**CI**	**DO**	**CU**	**TU**	**CO**
videochat	3.05	2.64	2.84	2.89	**3.02**
15_segments	2.88	2.71	2.94	**3.06**	2.76
15_summary	2.74	2.72	2.73	2.68	2.55
15_overlap_segments	2.63	2.73	2.76	2.66	2.61
15_overlap_summary	2.96	2.66	2.68	2.75	2.64
30_segments	2.62	2.74	2.63	2.61	2.65
30_summary	2.66	2.78	2.83	2.71	2.64
30_overlap_segments	3.02	2.84	2.87	2.79	2.76
30_overlap_summary	**3.08**	**2.99**	**3.00**	3.02	2.89
45_segments	2.66	2.46	2.53	2.57	2.44
45_summary	2.51	2.61	2.66	2.63	2.47
45_overlap_segments	2.45	2.48	2.54	2.47	2.47
45_overlap_summary	**3.08**	2.84	2.83	2.85	2.58
60_segments	2.65	2.49	2.55	2.53	2.48
60_summary	2.64	2.57	2.61	2.61	2.45
60_overlap_segments	2.56	2.41	2.49	2.42	2.40
60_overlap_summary	2.85	2.69	2.64	2.55	2.52

Table 12. DAC-GPT3.5 Scores for CI, DO, CU, TU, and CO across different methods.

Method	**CI**	**DO**	**CU**	**TU**	**CO**
videochat	**3.05**	2.64	2.84	2.89	**3.02**
60_overlap_segments	2.48	2.45	2.60	2.35	2.26
60_overlap_summary	2.67	2.60	2.76	2.74	2.49
45_overlap_segments	2.52	2.52	2.60	2.56	2.47
45_overlap_summary	2.57	2.57	2.55	2.68	2.55
15_overlap_segments	2.72	2.74	2.79	2.68	2.66
15_overlap_summary	2.94	2.83	**2.90**	**2.99**	2.95
15_segments	2.91	**3.03**	2.84	2.86	2.81
15_summary	2.73	2.61	2.61	2.61	2.58
45_segments	2.48	2.44	2.63	2.51	2.39
45_summary	2.57	2.66	2.73	2.76	2.44
60_segments	2.58	2.51	2.48	2.48	2.50
60_summary	2.70	2.63	2.81	2.71	2.69
30_overlap_segments	2.79	2.78	2.86	2.91	2.73
30_overlap_summary	2.95	2.94	2.75	2.65	2.70
30_segments	2.55	2.66	2.67	2.61	2.53
30_summary	2.68	2.63	2.69	2.54	2.40

Table 13. DAC-GPT4omini Scores for CI, DO, CU, TU, and CO across different experiments