Understanding Long Videos with Multimodal Large Language Models

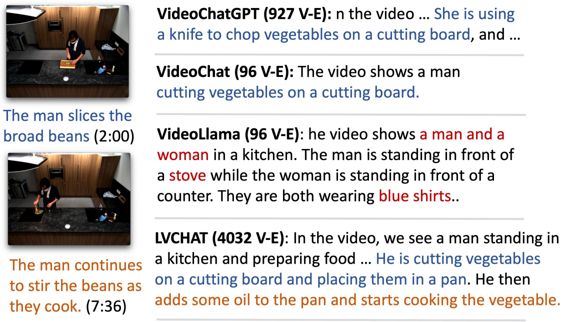
Anonymous ACL submission

Abstract

Enabling large language models (LLMs) to effectively process long videos is essential for advancing multimodal large language models (MLLMs). While current general-purpose MLLMs can handle short video clips, they often struggle with longer videos, typically failing to capture crucial information in videos over one minute. This challenge arises primarily from over-compression, where the encoded video representations are inadequate to fully represent the entire video. To address this, we introduce Long Video Chat (LVChat), a novel approach focused on long video understanding with MLLMs. In LVChat, we propose Frame-Scalable Encoding (FSE) to encode global video information, dynamically adjusting the number of video embeddings based on video duration to prevent over-compression. Additionally, we introduce Interleaved Frame Encoding (IFE), which interleaves multiple video embedding groups with shared positional embeddings across these groups. Experimental results show that LVChat significantly outperforms baselines in long-video QA and captioning tasks. Code and data will be made available upon publication.

1 Introduction

Recent efforts have focused on enhancing the multimodal capabilities of large language models, extending their power beyond text to other modalities (Touvron et al., 2021; Bao et al., 2021; He et al., 2022; Hassid et al., 2023; Borsos et al., 2023; Sicherman and Adi, 2023). Given that video is a unique medium through which humans perceive the real world (Li et al., 2023), there is growing interest in advancing multimodal large language models (MLLMs) to better understand videos. Notable examples include VideoChat (Li et al., 2023), VideoChatGPT (Maaz et al., 2023), and VideoLlama (Zhang et al., 2023b).



The man slices the broad beans (2:00)

The man continues to stir the beans as they cook. (7:36)

VideoChatGPT (927 V-E): n the video … She is using a knife to chop vegetables on a cutting board, and …

VideoChat (96 V-E): The video shows a man cutting vegetables on a cutting board.

VideoLlama (96 V-E): he video shows a man and a woman in a kitchen. The man is standing in front of a stove while the woman is standing in front of a counter. They are both wearing blue shirts..

LVCHAT (4032 V-E): In the video, we see a man standing in a kitchen and preparing food … He is cutting vegetables on a cutting board and placing them in a pan. He then adds some oil to the pan and starts cooking the vegetable.

Figure 1: An illustration from the long-video captioning task using the TACoS dataset (Rohrbach et al., 2014). The video is 671 seconds long. We highlight two keyframes, each with its corresponding timestamp and human-annotated caption. The term "V-E" refers to Video Embeddings. Our method, LVChat, leverages significantly more video embeddings compared to other approaches. This increased embedding density allows LVChat to generate detailed and accurate descriptions, in contrast to other methods that either miss critical information (marked in yellow) or produce hallucinated content (marked in red).

While previous works demonstrate promising results on short videos, improving their performance on videos *longer than one-minute* is observed to be challenging (Li et al., 2023). We believe (and empirically prove it in our experiments) that *the inability to comprehend long videos comes from the over-compression of video content.* For example, VideoChat (Li et al., 2023) and Video-Llama (Zhang et al., 2023b) convert entire video into a fixed number of video embeddings, regardless of the video's duration, resulting in inadequate representations, especially for longer videos (e.g., over 600 seconds). VideoChat-GPT (Maaz et al., 2023) compresses video content by using 256 global video embeddings along with one embedding per frame, which also suffers from over-compression as video length increases. As shown in Figure 1, these models struggle to maintain promising performance with long video.

To address this challenge, we propose LVChat, a novel framework for long-video understanding that comprises two key components: *Frame*

Scalable Encoding (FSE) and *Interleaved Frame Encoding (IFE)*. To mitigate the issue of over-compression, we design FSE, a new global feature extraction strategy that down-samples the entire video into a primary frame group, with the number of video embeddings in this group scaling with the video's length (T). To extract more detailed information from the video, we introduce IFE, which shifts the frames within the primary group to create multiple augmented video groups. Each group is processed by FSE to generate augmented sets of video embeddings. These embeddings are then interleaved and fed into the language model, with shared positional embeddings across groups. We evaluate LVCHAT on long video question answering (MVBench and EgoSchema) and caption generation (TACoS) tasks, where LVCHAT outperforms baselines by up to 27% in accuracy on video QA and 25% in ROUGE score on caption generation, demonstrating its effectiveness in long video understanding.

2 Related Work

Video Question Answering (VideoQA) VideoQA aims to evaluate the model's ability to understand videos. Typical works pretrain a video-text model and perform a successive fine-tuning on VideoQA (Zellers et al., 2021; Bain et al., 2021; Miech et al., 2019; Wang et al., 2022; Fu et al., 2021; Zeng et al., 2022; Li et al., 2022; Yu et al., 2023). These works are focused specifically on QA tasks, which may limit the applications of these models in real-world scenarios. In contrast, multimodal large language models are general-purpose models suitable for various applications.

Multimodal Large Language Models(MLLMs) Some works adopt image captioning models to generate captions for each frame to convert the video into text, which could serve as the input to the large language models (Wang et al., 2023a; Li et al., 2023). These methods depend heavily on the captioning models and may pose problems when information beyond text descriptions exists in the video. Another trend introduces adapters to convert the visual information into representations in the text embedding space, which spans from image domain (Luo et al., 2022; Chen et al., 2022; Zhu et al., 2023; Liu et al., 2023) to video domain (Li et al., 2023; Maaz et al., 2023; Zhang et al., 2023b; Yang et al., 2022; Li et al., 2024a).

3 Method

3.1 Preliminary of MLLMs

Given a video $\mathbf{V} = [\mathbf{I}_i]_{i=1,2,\cdots,F}$ consisting of F seconds, we sample F frames in the video. Then an MLLM first uses a video encoder f_{vid} to convert the video into a list of video embeddings $\mathbf{E} = f_{vid}(\mathbf{V})$ within the word embedding space of the language model f_{llm}. The language model then generates an output (e.g., the answer to a question in video QA task), given the video embeddings and a text query $\mathbf{W}$ (e.g., the question about the video):

$$\text{output} = f_{llm}(\mathbf{E}, \mathbf{W}).$$

In existing works, video embeddings $\mathbf{E}$ are typically too few relative to the number of frames, causing over-compression (as shown in Figure 1). Therefore, we propose LVCHAT, which includes Frame-Scalable Encoding (FSE) and Interleaved Frame Encoding (IFE).

3.2 FSE: Encode Global Information

FSE is designed to encode global information from a long video into a scalable set of video embeddings, with the number of embeddings increasing as the video lengthens.

As shown in Figure 2, the video $\mathbf{V}$ is first down-sampled by a factor γ into the primary group, resulting in a sequence of L frames ($[\mathbf{I}_1, \mathbf{I}_{1+\gamma}, \mathbf{I}_{1+2\gamma}, \cdots \mathbf{I}_L]$, where $L = \lceil T/\gamma \rceil$). These L frames are then segmented into n clips ($[\mathbf{C}_1, \cdots, \mathbf{C}_n]$), each containing K frames, with K as a hyper-parameter and $n = \lceil L/K \rceil$. Each clip is encoded into a fixed number of embeddings using the video encoder in the MLLM (we use $N = 96$ following Li et al. (2023)):

$$\mathbf{E}_1, \cdots, \mathbf{E}_n = f_{vid}(\mathbf{C}_1), \cdots, f_{vid}(\mathbf{C}_n). \quad (1)$$

Here, $\mathbf{E}_i \in \mathbb{R}^{N\times d}$ represent the N embeddings for the i-th clip ($i \in \{1, \cdots, n\}$). After concatenating these embeddings, we obtain $\mathbf{E}_1^{FSE} \in \mathbb{R}^{(n*N)\times d}$, corresponding to the FSE embeddings of the primary group. Since this group starts from the first frame, we denote it as $\mathbf{E}_1^{FSE}$. In IFE, where groups start from different frames, we use $\mathbf{E}_j^{FSE}$ to indicate the FSE embeddings for the group starting at the j-th frame. As the video length (T) increases, more clips are obtained (n increases), resulting in a larger number of video embeddings. This expansion helps mitigate the risk of over-compression. To enable the MLLM to understand these video embeddings, we finetune it using the FSE strategy.

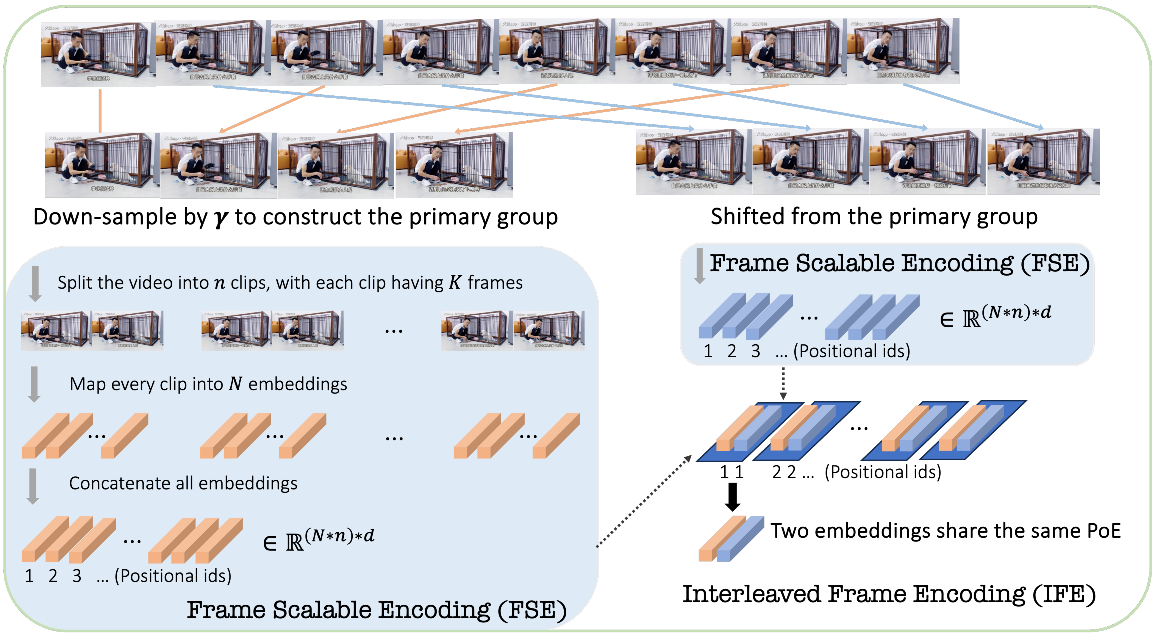

Down-sample by γ to construct the primary group
Shifted from the primary group
Split the video into n clips, with each clip having K frames
Map every clip into N embeddings
Concatenate all embeddings
$\in \mathbb{R}^{(N*n)*d}$
1 2 3 ... (Positional ids)
Frame Scalable Encoding (FSE)
Frame Scalable Encoding (FSE)
$\in \mathbb{R}^{(N*n)*d}$
1 2 3 ... (Positional ids)
1 1 2 2 ... (Positional ids)
Two embeddings share the same PoE
Interleaved Frame Encoding (IFE)

Figure 2: Illustration of Interleaved Frame Encoding (IFE) with a down-sample factor of $\gamma = 2$. First, the entire video is downsampled by $\gamma = 2$ to create the primary group, on which FSE is applied to produce $\mathbf{E}_1^{FSE}$ (shown in orange). In IFE, the primary group is augmented by shifting the frames by 1 second, followed by FSE again. The embeddings from both groups are then interleaved, with embeddings in each group sharing the same positional IDs.

3.3 IFE: Augment to Provide More Details

While the primary $\mathbf{E}_1^{FSE}$ embeddings cover the whole video, they can be sparse and lose finer details. To address this, we propose IFE, which shifts the frames in the primary group by a small interval (e.g., 1 sec) and repeats the downsampling process, generating a new group of frames (e.g., $[\mathbf{I}2, \mathbf{I}_{2+\gamma}, \cdots]$). We then apply FSE to this new group to produce a new set of video embeddings (e.g., $\mathbf{E}_2^{FSE}$). By shifting the frames $(\gamma - 1)$ times, we obtain $(\gamma - 1)$ additional sets of video embeddings $\{\mathbf{E}_2^{FSE}, \cdots, \mathbf{E}_\gamma^{FSE}\}$, which augment the primary $\mathbf{E}_1^{FSE}$ and provide comprehensive coverage of all frames in the video.

To feed these embeddings into f_{llm}, we assign the same position ids $[1, \cdots, N * n]$ to each group (as noted in Section 3.2, each group contains $N * n$ embeddings). This approach is based on the intuition that embeddings within each group share similar temporal orders, and embeddings across different groups with the same positions are also temporally aligned. When inputting these embeddings into the language model, we interleave the embeddings from each group, reorganizing them according to their positional ids to ensure they are processed in the correct temporal order (as illustrated in Figure 2). The IFE strategy intuitively adds more detail on top of FSE, addressing the potential sparsity of FSE embeddings.

4 Experiments

4.1 Experimental Setups

We use VideoChat2 (Li et al., 2023) as the backbone of LVChat and conduct training with FSE (see more training details in Appendix §C.1). We compare LVChat with **VideoChat2** (Li et al., 2023), **Video-Llama** (Zhang et al., 2023b), **Video-ChatGPT** (Maaz et al., 2023), **LLoVi** (Zhang et al., 2023a). We refer the readers to Appendix § C.2 for more details about these baselines.

Evaluation Benchmarks We evaluate LVChat on the following long-video QA and caption generation tasks: (1) **MVBench-Extended**: we take MVBench (Li et al., 2024b) and extend its original videos by concatenating distracting videos from Ramachandra and Jones (2020) to 100s, 300s, and 600s. (2) **EgoSchema** (Mangalam et al., 2023): a long video QA dataset with an average length of 180s, derived from Ego4D (Grauman et al., 2022). We use the official validation set with publicly released answers for evaluation. (3) **TACoS** (Rohrbach et al., 2014): a long-video caption generation dataset whose videos are 287s on average with human-annotated captions of critical timestamps in the video. We use OpenAI's GPT-4 (OpenAI, 2024) to generate a reference summary

	AS			AP			UA			OI		
	100s	300s	600s	100s	300s	600s	100s	300s	600s	100s	300s	600s
VideoChatGPT	30	27.5	26	23	25.5	27	34	28	30	27.5	26	26.5
VideoLlama	24	25.5	23.5	23.5	23.5	25	39	38	37.5	27	26	26
VideoChat2	38.5	30.5	28.5	33	29	23	46.5	**45**	**41.5**	57.5	39.5	38.5
LVChat w/o IFE	-	41	34.5	-	**38.5**	30.5	-	38.5	38.5	-	47	46
LVChat	**53.5**	**42.5**	**37**	**45.5**	37.5	**34**	**47**	37	38.5	**66**	**52.5**	**48.5**

Table 1: Results on QA datasets extended from MVBench. The interleaving factor γ is set to be 2 for videos of length 5 min and 4 for videos of length 10 min. All models are evaluated using MVBench's protocol.

	Rouge1	Rouge2	RougeL	RougeSum
VideoLlama	0.269	0.0490	0.196	0.193
VideoChatGPT	0.263	0.0567	0.188	0.188
VideoChat2	0.261	0.0675	0.195	0.196
LVChat	**0.360**	**0.0920**	**0.244**	**0.246**

Table 2: Evaluation on long-video caption generation dataset TACoS (Average 287s). **Bold**: best results.

from the labeled captions and conduct the human inspection. We report ROUGE scores on its official dataset. For more details on each dataset, please refer to §C.5 and §C.6.

4.2 Comparisons on MVBench-Extended

We report the QA performance of various models on MVBench-extended across different video lengths in Table 1. The results indicate that LVChat consistently outperforms previous methods on nearly all datasets and in almost all settings, demonstrating its superior ability to extract critical information from videos, even those up to 600 seconds long. Additionally, we present results without the IFE strategy, where a noticeable performance drop occurs, highlighting the effectiveness of IFE.

4.3 Comparisons on TACoS and EgoSchema

In this section, we evaluate the performance of LVChat on two datasets, EgoSchema and TACoS, both of which feature naturally long videos. For TACoS caption generation task, all models generate up to 100 new tokens using greedy search, and the results are reported using the ROUGE score in Table 2. For EgoSchema QA task, we maintain the same settings and report the accuracy in Figure 3. It's important to note that our focus is on zero-shot performance. As shown in the tables, LVChat outperforms all other baselines by a significant margin. We provide the case study in Table 1, where we demonstrate that existing methods either fail to capture detailed information in contiguous frames or suffer from hallucinations, whereas LVChat effectively captures correlations within the long video, providing detailed and accurate descriptions.

VideoLlama	28.4
VideoChatGPT	26.0
VideoChat2	50.0
LLoVi	51.2
LVChat	**55.4**

Figure 3: Accuracy on EgoSchema.

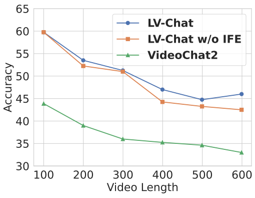


Figure 4: Comparison with different lengths.

4.3.1 Ablation Study

We present an ablation study on the impact of FSE and IFE in LVChat, with VideoChat2 as our backbone model. We evaluated various model variants on videos ranging from 100 to 600 seconds in length, reporting the average accuracy on OI and AS. The downsample factor γ varies from 1 to 6 as the video length increases from 100 to 600 seconds. When IFE is not enabled, only the primary group with 96 video embeddings is used. However, with IFE enabled, we interleave multiple groups of embeddings. For example, videos of 600 seconds are downsampled into 6 groups, with the positional IDs of the obtained video embeddings repeated 6 times across all groups. As shown in Figure 4, there is a clear trend: incorporating FSE significantly enhances the model's capability to handle long videos, and as video length increases, the additional benefit of IFE augmentation becomes even more pronounced.

5 Conclusion

In this study, we introduced LVChat, a novel approach for long video understanding, consisting of Frame-Scalable Encoding (FSE) and Interleaved Frame Encoding (IFE) to deal with the over-compression problem. We evaluated LV-Chat's performance in long-video qustion answering and captioning tasks. The experimental results demonstrate that LVChat consistently surpasses previous methods on videos with varying lengths.

6 Limitations

One limitation is that LVChat uses VideoChat2 as the backbone model, whose LLM inside is Vicuna-7B-v1.0. We acknowledge that there are LLMs that are more advanced such as Vicuna-7B-v1.5, released during the development of LVChat. We plan to update the backbone in LVChat by training more advanced LLMs or at larger size, to further boost the long-video understanding capabilities of LVChat.

References

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2022. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *CVPR*, pages 18009–18019. IEEE.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. 2021. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. 2022. Ego4d: Around the world in 3, 000 hours of egocentric video. In *CVPR*, pages 18973–18990. IEEE.

Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. 2023. Textually pretrained speech language models. *arXiv preprint arXiv:2305.13009*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. 2022. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024a. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

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

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. 2022. VC-GPT: visual conditioned GPT for end-to-end generative vision-and-language pre-training. *CoRR*, abs/2201.12723.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Antoine Miech, Dimitri Zhukov, Jean Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*.

OpenAI. 2024. ChatGPT (4) [Large language model]. `https://chat.openai.com`.

Bharathkumar Ramachandra and Michael Jones. 2020. Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2569–2578.

Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. *Coherent Multi-sentence Video Description with Variable Level of Detail*, page 184–195. Springer International Publishing.

Amitay Sicherman and Yossi Adi. 2023. Analysing discrete self supervised speech representation for spoken language modeling. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR.

Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*.

Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. 2023a. Chatvideo: A tracklet-centric multimodal and versatile video understanding system. *arXiv preprint arXiv:2304.14407*.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. 2023b. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, pages 9777–9786. Computer Vision Foundation / IEEE.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. 2020. CLEVRER: collision events for video representation and reasoning. In *ICLR*. OpenReview.net.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. 2023. Self-chained image-language model for video localization and question answering. In *NeurIPS*.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. Merlot: Multimodal neural script knowledge models. In *NeurIPS*.

Yan Zeng, Xinsong Zhang, Hang Li, Jiawei Wang, Jipeng Zhang, and Wangchunshu Zhou. 2022. X^2-vlm: All-in-one pre-trained model for vision-language tasks. *arXiv preprint arXiv:2211.12402*.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. 2023a. A simple LLM framework for long-range video question-answering. *CoRR*, abs/2312.17235.

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

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

A Notations

All the notations are provided in Table 3.

Symbols	Meanings
T	duration
F	total number of frames
K	number of frames in one clip
N	number of tokens per clip
F_s	number of sampled frames
n	number of clips
n_m	max number of clips
γ	number of interleaved times
n_i	number of clips in interleaved setting

Table 3: Notations

B Additional Experimental Results

B.1 Model performance on all subsets of MVBench

Table 6 shows the results on the original MVBench and Table 7 shows the results on the augmented MVBench with Street-Scene.

B.2 Additional Ablation Study of IFE

As shown in Figure 2, we interleave the embeddings encoded from different clips. A more straightforward way is shown in Figure 5. However, this strategy shows inferior performance compared to IFE, with the results shown in Table 4. We provide some potential insights behind the table: Imagine there are eight frames in total, and IFE encoding would be split them into [1,2,1,2,1,2,1,2], where "1" and "2" refer to the indices of two groups. The other strategy in Figure 5 would be [1,1,1,1,2,2,2,2]. Then the major reason may lie in the positional encoding. When inputted into the language model, the positional encoding would be [1,1,2,2,3,3,4,4]. Then for the IFE split, all the frames in group one would have the positional encoding [1,2,3,4], with the same case for group two. Whereas for the other strategy, the positional encoding for the first group is [1,1,2,2], and for the second group, the positional encoding would be [3,3,4,4]. Note that during FSE training, the model has only seen the embeddings equipped with positional encoding [1,2,3,4], thus the positional encodings such as [1,1,2,2] and [3,3,4,4] may potentially make the model confused and yield inferior performances.

B.3 Case Study

We compare LVCHAT against the baselines on the TACoS and show two cases in Table 5. For each video, we choose two representative scenes and match them with the captions from the TACoS. In many cases, VideoChatGPT and VideoChat2 can only summarize the whole video in one sentence without any further detail. While VideoLlama generates longer answers generally, it often has strong hallucinations on the details of the video and gives far-off descriptions. In contrast, our model captures much more detail, including the actions of the subject and the environment where the video was shot. Intuitively this is due to the use of more embeddings (as shown in the captions of Table 5) in LVCHAT. In the cases we show, we also highlight the correct action descriptions that these models generate. All three baselines fail to correctly capture the actions of the person from both two scenes while LVCHAT succeeds in describing both. LVCHAT benefits from longer encodings using FSE; and with the IFE technique, even more frames can be compressed into encodings of limited lengths.

C Experiment Settings

C.1 Additional Implementation Details

We initialize our model from VideoChat2 (Li et al., 2023). For LVCHAT, we use the equation $n = \lceil T/K \rceil$ to determine the number of frames to sample, and encode every K frames into N embeddings, where $K = 16, N = 96$. During the training, we specify $n_m = 10$. Thus if the video length T is shorter than $n_m * K = 160$, we do not need IFE and only FSE is turned on, whereas if the video length T is longer than 160, we determine the interleaving factor γ with $\gamma = \lceil \lceil T/K \rceil / n_m \rceil$ and then perform the IFE process. We set the learning rate as 2e-6, with warmup epochs=0.3, num_epochs=1, scheduler=cos, optimizer=AdamW. The fine-tuning is performed on 4 NVIDIA-RTX-A6000 GPUs. For FSE, we fine-tune our model on the instruction dataset collected for training VideoChat2 (Li et al., 2023) with the detailed datasets shown in Appendix §C.3. We use `evaluate` package to calculate ROUGE score for TACoS dataset.

C.2 Links to the baselines

We put the details and the links to the baselines as below:

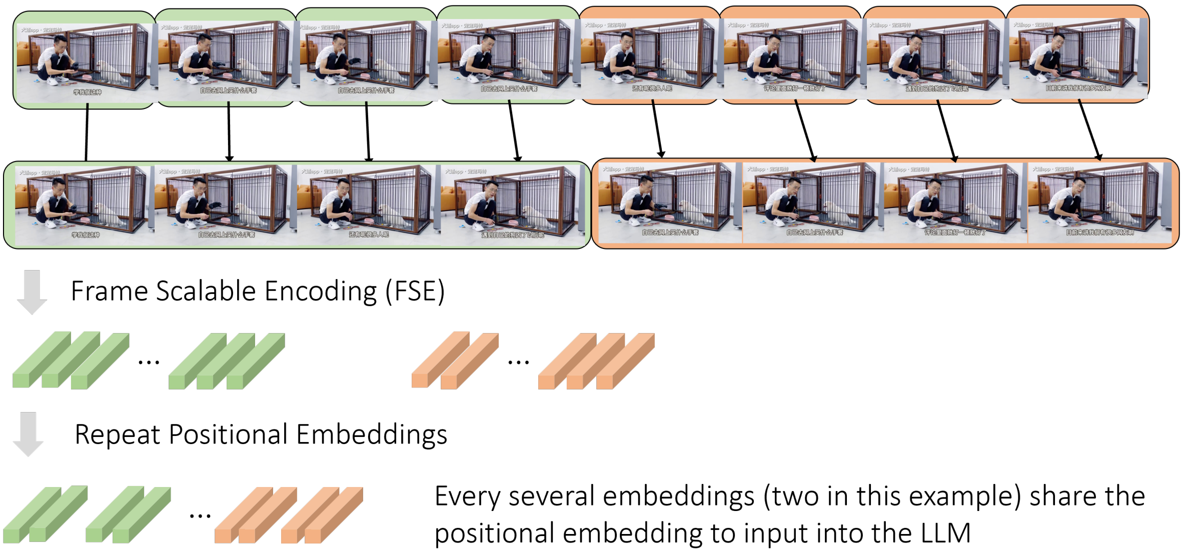

Frame Scalable Encoding (FSE)
...
...
Repeat Positional Embeddings
...
Every several embeddings (two in this example) share the positional embedding to input into the LLM

Figure 5: A more straightforward encoding strategy to allow longer context input to the LLM.

	300s				600s			
	AS	AP	UA	OI	AS	AP	UA	OI
LVChat	**42.5**	**38.5**	37	**52.5**	**37**	34	**38.5**	**48.5**
LVChat (IFE replaced)	31	35.5	**39**	48	31	**35**	33.5	43

Table 4: Ablation study of IFE. "(IFE replaced)" means using the encoding strategy in Figure 5. **Bold**: best results.

VideoChat2 (Li et al., 2023): The backbone of our model without FSE and IFE. We follow the implementation in VideoChat2 and sample 16 frames from the given video regardless of the video length. The link is `https://github.com/OpenGVLab/Ask-Anything/blob/main/video_chat2/mvbench.ipynb`

Video-Llama (Zhang et al., 2023b): We exclude the audio modality here for fair comparison. Following the setting from the original implementation model, we use the Video-LLaMA-2-7B-Finetuned checkpoint and sample 16 frames from each video. The link is `https://github.com/DAMO-NLP-SG/Video-LLaMA`

Video-ChatGPT (Maaz et al., 2023): We use the same setup as in the official demo and samples 100 frames from each video. The link is `https://github.com/mbzuai-oryx/Video-ChatGPT/blob/main/docs/offline_demo.md`

LLoVi (Zhang et al., 2023a): (On EgoSchema benchmark) A framework that can generate dense captions for short video clips, where the captions are aggregated for long videos to be inputted into LLMs. We report the results in the paper (Zhang et al., 2023a) with Captioner being LaViLa and LLM being GPT-3.5-Turbo. the link is `https://github.com/CeeZh/LLoVi`

C.3 Instruction Tuning Dataset Details

To fine-tune our model with FSE, we adopt the dataset collected by VideoChat2 (Li et al., 2023), where there is 1.9M video instruction data in total[1]. However, due to that some datasets are not accessible, we use a subset of this dataset:

- VideoChat (Li et al., 2023), collected from InternVid (Wang et al., 2023b).
- VideoChatGPT (Maaz et al., 2023), the original caption data is converted into conversation data by (Li et al., 2023).
- NExTQA (Xiao et al., 2021), a multi-choice question answering dataset.
- CLEVRER (Yi et al., 2020), an action prediction, multi-choice question answering dataset.

---

[1] `https://github.com/OpenGVLab/Ask-Anything/blob/main/video_chat2/DATA.md`

Captioned scenes	Results
 He cut off ends of cucumbers. He put cucumber on plate.	LVCHAT: “The video shows a man standing in a kitchen and cutting a cucumber on a cutting board. He then places the cucumber on a plate and walks away. The kitchen has a counter with a sink and a refrigerator. The man is wearing a black shirt and is standing in front of a black and white fridge. The video is shot from an overhead angle, giving a bird’s eye view of the kitchen...”
	VideoChatGPT: “The video shows a person cutting vegetables on a countertop in a kitchen.”
	VideoChat2: “The video shows a man cutting a cucumber on a cutting board.”
	VideoLlama: “1. A man is standing in a kitchen with a black countertop and a wooden table. 2. He is wearing a black shirt and a black jacket. 3. He is preparing food in the kitchen. 4. He is using a knife to cut a piece of meat. 5. He is cutting the meat on a cutting board. 6. He is using a spoon to stir the food. 7. He is standing in the kitchen”
 The man slices the broad beans. The man continues to stir the beans as they cook.	LVCHAT: “In the video, we see a man standing in a kitchen and preparing food. He is wearing a blue shirt and a white scarf around his neck. He is cutting vegetables on a cutting board and placing them in a pan. He then adds some oil to the pan and starts cooking the vegetables. The man is standing in front of a counter with a black and white marble top. There is a microwave oven on the counter, and a white bowl...”
	VideoChatGPT: “In the video, a woman is standing in a kitchen and preparing food. She is using a knife to chop vegetables on a cutting board, and she is also using a spoon to stir something in a bowl. The video also shows her using a spoon to stir something in a bowl.”
	VideoChat2: “The video shows a man cutting vegetables on a cutting board.”
	VideoLlama: “The video shows a man and a woman in a kitchen. The man is standing in front of a stove while the woman is standing in front of a counter. They are both wearing blue shirts. The kitchen has a black countertop and a black stove. There is a brown wooden table in the kitchen. The man is holding a knife and the woman is holding a plate. They are both preparing food.”

Table 5: Two cases (s23-d21, s28-d51) on the TACoS dataset of LVCHAT compared with the baselines. The lengths of the two videos are 2 min 46 s and 11 min 11 s respectively. VideoChatGPT uses 100 frames; VideoChat2 and VideoLlama use 16 frames. LVCHAT uses 192/672 frames from the two videos and compresses them into 96 frame encodings. The highlighted parts are correct descriptions of actions.

C.4 Datasets Selection Criteria

By manually looking at the examples, we compiled a few rules that a valid set of data should satisfy:

1. The baseline’s performance drops as the target length of the extended video increases.
2. The baseline’s performance should be better than random guesses.
3. Questions in the subset should not be greatly affected by video from Street-Scene.
4. Video should not be too short compared to our target length.
5. The questions in the subset should be answerable by a visual-only model. (i.e., the answers should not be all in the subtitles or the captions, leading to unanswerable questions based on visual data only)

By applying these rules, we select four datasets (Action Sequence, Action Prediction, Unexpected Action, Object Interaction) that are valid for testing long video-language models.

C.5 Dataset Extension

Despite the variety of videos that MVBench(Li et al., 2024b) has. The average length of the four selected datasets are merely 25.5s, which can barely benefit from the capability of long-video models. To make use of these videos, we extend them with a second video sampled from the Street-Scene dataset(Ramachandra and Jones, 2020). The Street-Scene dataset contains 91 videos with 15 frames per second, and we select the first 54000 frames from the dataset, totaling an 1 hour video from which we sample the second video.

The extension process is as follows:

1. Set a target length of video T that the model should see.
2. For a original video v of length $\mathcal{L}(v) < T$, we applies a hash function $\mathcal{H}$ (see below) to the file name N_v of the video v to get a integer t_0 that is between 0 and 3600, which will be used as the starting time of the second video. The hash function in python is:
   ```
   def hashstr(s: str) -> int:
     return sum(ord(c) * 31 ** (i % 3)
         for i, c in enumerate(s))
   ```
3. Draw a second video from the Street-Scene dataset that starts at $t_0 = \mathcal{H}(N_v)$ and ends at $t_0 + (T - \mathcal{L}(v))$.
4. Choose a time point $t_1 = \mathcal{H}(N_v + \text{":insert"})$ in the second video where we will insert the original video.
5. Insert the original video at t_1 of the second video and returns the extended video.

C.6 GPT-4 TACoS summarization

We use the following content to query the "GPT-4" API from OpenAI on Oct.9th, 2023. The context is composed of human-labelled captions and their starting times. The template we use for prompting GPT-4 is:

```
You are an assistant answering questions
     based on video contexts. Your
    answer should be based on the given
    contexts, but you can also infer the
     actual video content from the tag
    information and your common sense.
    The timed description is a
    description for the video at the
    given second. When describing,
    please mainly refer to the timed
    description. Don't create a video
    plot out of nothing.
Contexts for the video: \{context\}
Question: Could you please describe what
     is happening in the video?
```

Here is an example of video s13-d21. The prompt for GPT-4 is:

```
 You are an assistant answering
    questions based on video contexts.
    Your answer should be based on the
    given contexts, but you can also
    infer the actual video content from
    the tag information and your common
    sense. The timed description is a
    description for the video at the
    given second. When describing,
    please mainly refer to the timed
    description. Don't create a video
    plot out of nothing.
Contexts for the video: """
Second 9: He took out cutting board
Second 17: He took out knife
Second 22: He took out cucumber
Second 35: He took out plate
Second 47: He washed cucumber
Second 57: Cut off ends of cucumbers
Second 72: He sliced cucumbers
Second 90: He put cucumbers on plate
Second 9: person takes chopping board
    out
Second 17: person removes knife from
    draw
Second 22: person removes cucumber out
    of refrigerator
Second 35: person removes plate out of
    cabinet
Second 47: person then washes cucumber
Second 57: person then places cucumber
    on plate
Second 64: perosn then cuts ends off
    cucumber
Second 72: person then cuts cucumber in
    slices
Second 90: person then places cucumber
    on plate.
Second 9: The person gets out a cutting
    board.
Second 17: The person gets out a knife.
Second 22: The person gets out a
    cucumber.
Second 35: The person gets out a plate.
Second 47: The person rinses the
    cucumber.
Second 57: The person cuts the tips off
    the cucumber.
Second 96: The person slices the
    cucumber and puts the slices on the
    plate.
Second 9: The person gets out a cutting
    board.
Second 17: The person gets out a knife.
Second 25: The person gets out a
    cucumber.
Second 35: The person gets out a plate.
Second 47: The person rinses the
    cucumber.
Second 57: The person cuts off the tips
    of the cucumber.
Second 72: The person cuts up the
    cucumber.
Second 90: The person puts the cucumber
    slices on the plate.
Second 9: The person takes out a cutting
     board from the drawer.
Second 17: The person takes out a knife
    from the drawer.
Second 25: The person procures a
    cucumber from the fridge.
Second 35: The person procures a plate
    from the cabinet.
Second 47: The person washes the
    cucumber in the sink.
Second 57: The person cuts the ends off
    the cucumber then cuts the body into
     slices.
```

	AS	AP	AA	FA	UA	OE	OI	OS	MD	AL	ST	AC	MC	MA	SC	FP	CO	EN	ER	CI	Average
VideoChat2	66	47.5	83.5	49.5	60	58	71.5	42.5	23	23	88	39	42	58.5	44	49	36.5	35	40.5	65.5	51.125
VideoChatGPT	23.5	26	62	22.5	26.5	54	28	40	23	20	31	30.5	25.5	48.5	29	39.5	33	29.5	26	35.5	32.675
VideoLlama	27.5	25.5	51	29	39	48	40.5	38	22.5	22.5	43	34	22.5	45.5	32.5	32.5	40	30	21	37	34.075
LV-Chat	62.5	47	79.5	44	61.5	56	74	40.5	23.5	27	82	47.5	39.5	69.5	47	48.5	40	34.5	38.5	60	51.125

Table 6: Model Performance on the original MVBench. The results of VideoChat2, VideoChatGPT and VideoLlama are from the MVBench repository (`https://github.com/OpenGVLab/Ask-Anything/blob/main/video_chat2/MVBENCH.md`).

```
Second 90: The person sets cucumber slices on the plate.
Second 9: The person takes out a cutting board from the drawer.
Second 17: The person takes out a knife from the drawer.
Second 22: The person procures a cucumber from the fridge then takes a plate from the cabinet.
Second 47: The person washes the cucumber in the sink.
Second 57: The person cuts the ends from the cucumber.
Second 72: The person chops the cucumber into slices on the cutting board.
Second 90: The person sets the cucumber slices on the plate.
Second 9: The person takes out a cutting board from the drawer.
Second 17: The person takes out a knife from the drawer.
Second 22: The person procures a cucumber from the fridge.
Second 35: The person procures a plate from the cabinet.
Second 47: The person washes the cucumber in the sink.
Second 57: The person cuts the ends off the cucumber.
Second 72: The person slices the cucumber on the cutting board.
Second 90: The person sets the sliced cucumber on the plate.
Second 9: He goes to the drawer and takes out a cutting board and knife.
Second 25: He goes to the refrigerator and takes out a cucumber.
Second 35: He goes to the cupboard and takes out a plate and places it on the counter.
Second 50: He goes to the sink and washes the cucumber.
Second 57: He then cuts off the ends of the cucumber and then slices the cucumber.
Second 72: He picks up the cucumber and places it on the plate.
Second 9: He opens the drawers and takes out a cutting board and a knife.
Second 25: He gets a cucumber from the refrigerator and a plate from the cabinet.
Second 47: He sets the plate down and washes the cucumber in the sink.
Second 57: He puts the cucumber on the plate and dries off his hands.
Second 64: He uses the knife to cut off the ends of the cucumbers.
Second 72: He uses the knife to slice the cucumber into smaller pieces.
Second 96: He picks up the pieces of cucumber and places them on the plate.
Second 9: The person takes out a cutting board from the drawer.
Second 17: The person takes out a knife from the drawer.
Second 22: The person procures a cucumber from the fridge.
Second 35: The person procures a plate from the cabinet.
Second 47: The person washes the cucumber in the sink.
Second 57: The person chops the ends off the cucumber on the cutting board.
Second 72: The person slices the cucumber on the cutting board.
Second 90: The person sets the sliced cucumber on the plate.
Second 9: He gets out the cutting board, knife, plate, and cucumber from drawers and the refrigerator.
Second 50: He washes the cucumber in the sink and puts it on the plate.
Second 60: He wipes his hand on the rag.
Second 65: He chops the ends off the cucumber.
Second 72: He chops the cucumber into pieces with the knife.
Second 96: He gathers the chopped cucumbers together on a plate.
Second 9: the man is opening the drawer
Second 17: the man is placing a knife on the cutting board he got from the
```

```
    drawer
Second 22: the man is opening the
    refridgerator
Second 25: he got a cucumber from the
    refridgerator
Second 35: the man is washing the
    cucumber
Second 50: the man is slicing the
    cucumber width-wise
Second 96: then the man arranges the
    cucumbers on a plate
Second 9: The man places a cutting board
     on the counter.
Second 17: The man places a knife on the
     cutting board.
Second 22: The man gets a cucumber out
    of the refrigerator.
Second 35: The man gets a plate out of
    the cupboard.
Second 47: The man rinses the cucumber.
Second 57: The man dries his hands.
Second 64: The man cuts the ends off of
    the cucumber.
Second 72: The man cuts the cucumber
    into slices.
Second 96: The man arranges the cucumber
     slices on the plate.
Second 9: The person takes a cutting
    board from the drawer.
Second 17: The person removes a large
    knife from a drawer and puts it on
    the cutting board.
Second 25: The person takes a cucumber
    from the fridge.
Second 35: The person takes a plate from
     the cabinet.
Second 47: The person washes the
    cucumber in the sink.
Second 57: The person dries his hands
    with the towel.
Second 64: The person slices the
    cucumber.
Second 90: The person places the sliced
    cucumber onto the plate.
Second 9: He places the cutting board on
     the counter
Second 17: He placed the knife on the
    cutting board
Second 25: He got a cucumber from the
    refrigerator
Second 35: The man washed the cucumber
    in the sink
Second 57: The man dried his hands on
    the towel
Second 64: The man cut the cucumber into
     slices
Second 90: The man placed the cucumber
    slices on the plate
"""
Question:Could you please describe what
    is happening in the video?
```

And the response is:

```
The video is showing a person preparing
    a cucumber. They start by taking out
     a cutting board and a knife from a
    drawer. Then, they get a cucumber
    from the refrigerator and a plate
    from the cabinet. After that, they
    wash the cucumber in the sink. They
    then cut off the ends of the
    cucumber and slice it into pieces on
     the cutting board. Finally, they
    place the sliced cucumber on the
    plate.
```

We manually inspect the generated responses and query GPT-4 again if there are unreasonable responses or clear hallucinations.

C.7 Prompt for different models

C.7.1 System Prompt

For Video-ChatGPT, we use the system prompt (as used in the original paper):

```
You are Video-ChatGPT, a large vision-
    language assistant. You are able to
    understand the video content that
    the user provides, and assist the
    user with a variety of tasks using
    natural language. Follow the
    instructions carefully and explain
    your answers in detail based on the
    provided video.
```

For VideoChat2, Video-Llama, and our own model, we use the same system prompt from MVBench(Li et al., 2024b):

```
Carefully watch the video and pay
    attention to the cause and sequence
    of events, the detail and movement
    of objects, and the action and pose
    of persons.
```

C.7.2 Dataset-specific prompt

In TaCoS generation, the user asks the assistant:

```
Based on your observations, describe
    what is happening in the video as
    detailed as possible.
```

In QA datasets (MVBench and EgoSchema), we use the same format as in MVBench. Following is an example:

```
Question: What happened after the person
     took the food?
Options:
(A) Ate the medicine.
(B) Tidied up the blanket.
(C) Put down the cup/glass/bottle.
(D) Took the box.
Only give the best option.
```

Length 100s

	AS	AP	AA	FA	UA	OE	OI	OS	MD	AL
VideoChat2(16*1)	38.5	33	64.5	34	46.5	53	57.5	31.5	23.5	29
VideoChat2(16*10)	35.5	33.5	41.5	29.5	36.5	54.5	43	38	19.5	22
VideoChat2(8*10)	36.5	33	43	28	34.5	54	41.5	38	18.5	23
VideoChatGPT	30	23	54.5	24	34	53.5	27.5	41	24.5	26.5
VideoLlama	24	23.5	42.5	27	39	52.5	27	33	23.5	21
LVCHAT (8*10)	48.5	44	52.5	28.5	42.5	55	61	34	20.5	29
LVCHAT (16*10)	53.5	45.5	59.5	30	47	53	66	36.5	20.5	28

	ST	AC	MC	MA	SC	FP	CO	EN	ER	CI
VideoChat2(16*1)	72	43.5	30.5	57.5	54	29	40	31	39.5	43.5
VideoChat2(16*10)	40	39.5	22.5	37.5	58.5	26.5	38	24.5	30.5	39.5
VideoChat2(8*10)	40	38	22.5	37	57.5	27	41	25.5	32	44.5
VideoChatGPT	40	30	29	36.5	48.5	21	36	28.5	29	39
VideoLlama	32.5	29	28	41.5	45.5	29	34.5	30	25	35.5
LVCHAT (8*10)	55	39.5	26	46.5	48.5	31.5	39	37.5	35	39
LVCHAT (16*10)	62	41.5	27	49.5	47.5	28	36	38	37	38

Length 300s

	AS	AP	AA	FA	UA	OE	OI	OS	MD	AL
VideoChat2(16*1)	30.5	29	63	31.5	45	53	39.5	32	23	28.5
VideoChat2(16*10)	32	28.5	40.5	24	28.5	55.5	39	39	19	25
VideoChat2(8*10)	32	28.5	40.5	24	28.5	55.5	39	39	19	25.5
VideoChatGPT	27.5	25.5	54	23.5	28	53.5	26	43.5	24.5	29
VideoLlama	25.5	23.5	41.5	26.5	38	52	26	33	21.5	21
LVCHAT (8*10)	42.5	35.5	50	26.5	36	54	49.5	33.5	21.5	29
LVCHAT +IFE(8*10)	43.5	37	48.5	26.5	33.5	56	50	33	21	29.5
LVCHAT (16*10)	41	38.5	54	26.5	38.5	53.5	47	32.5	20.5	28.5
LVCHAT +IFE(16*10)	42.5	37.5	54	25	37	53.5	52.5	32.5	20	29

	ST	AC	MC	MA	SC	FP	CO	EN	ER	CI
VideoChat2(16*1)	60	44.5	28.5	58	57.5	27.5	41	33	35	42
VideoChat2(16*10)	36.5	38.5	22.5	37	58	25.5	38.5	25	26	39
VideoChat2(8*10)	36.5	38.5	22.5	37	58	25.5	38.5	25	26	39
VideoChatGPT	38.5	29.5	23.5	28	52	27	38	27	28.5	40.5
VideoLlama	30.5	29	28.5	41.5	47	29	33	32	22.5	34.5
LVCHAT (8*10)	51.5	39	25.5	45	48	29.5	34.5	36.5	30	34
LVCHAT +IFE(8*10)	46	40	28	46	48	29.5	35.5	36.5	29	33
LVCHAT (16*10)	49	37.5	29.5	45	48.5	27	34.5	36.5	35	34
LVCHAT +IFE(16*10)	48.5	39	29	47	48.5	29.5	30	35	32	35

Length 600s

	AS	AP	AA	FA	UA	OE	OI	OS	MD	AL
VideoChat2(16*1)	28.5	23	63	32	41.5	53	39	30.5	21.5	28.5
VideoChat2(16*10)	27	28	39	26.5	28	53	35.5	39	19	22.5
VideoChat2(8*10)	30	28	40	24.5	28.5	51	35.5	39	20.5	21.5
VideoChatGPT	26	27	56	25	30	52.5	26.5	40	24.5	25.5
VideoLlama	23.5	25	40	27	37.5	52.5	26	33	21.5	20
LVCHAT (8*10)	34	32	49	27.5	34.5	54	49	33	21.5	30
LVCHAT +IFE(8*10)	34	32	49	27.5	34.5	54	49	33	21.5	30
LVCHAT (16*10)	34.5	30.5	54	24	38.5	54	46	33.5	19	29.5
LVCHAT +IFE(16*10)	37	34	50.5	24.5	38.5	53.5	48.5	32.5	19.5	28.5

	ST	AC	MC	MA	SC	FP	CO	EN	ER	CI
VideoChat2(16*1)	51	45.5	28	59.5	56.5	30.5	36.5	33	32.5	43.5
VideoChat2(16*10)	38.5	38.5	22.5	36	57	26	39.5	25.5	25	38
VideoChat2(8*10)	35.5	38.5	23	33.5	59	26	37.5	24.5	25	36.5
VideoChatGPT	38	29.5	31	36.5	49	25.5	38.5	28.5	26.5	39
VideoLlama	28	29	29.5	42.5	47.5	29	33	31	22	33.5
LVCHAT (8*10)	42.5	42.5	26	43	48	30	33	36	29.5	35.5
LVCHAT +IFE(8*10)	42.5	42.5	26	43	48	30	33	36	29.5	35.5
LVCHAT (16*10)	44.5	37	24.5	46.5	48.5	27.5	35.5	36.5	33	35
LVCHAT +IFE(16*10)	47	41.5	24	47	47.5	27.5	37	36	35	33.5

Table 7: Model performance on extended MVBench