This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

TimeChat: A Time-sensitive Multimodal Large Language Model for Long Video Understanding

Shuhuai Ren[1]*, Linli Yao[1]*, Shicheng Li[1], Xu Sun[1], Lu Hou[2]
[1]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[2]Huawei Noah's Ark Lab
{shuhuai_ren, linliyao}@stu.pku.edu.cn {lisc99, xusun}@pku.edu.cn
houlu3@huawei.com

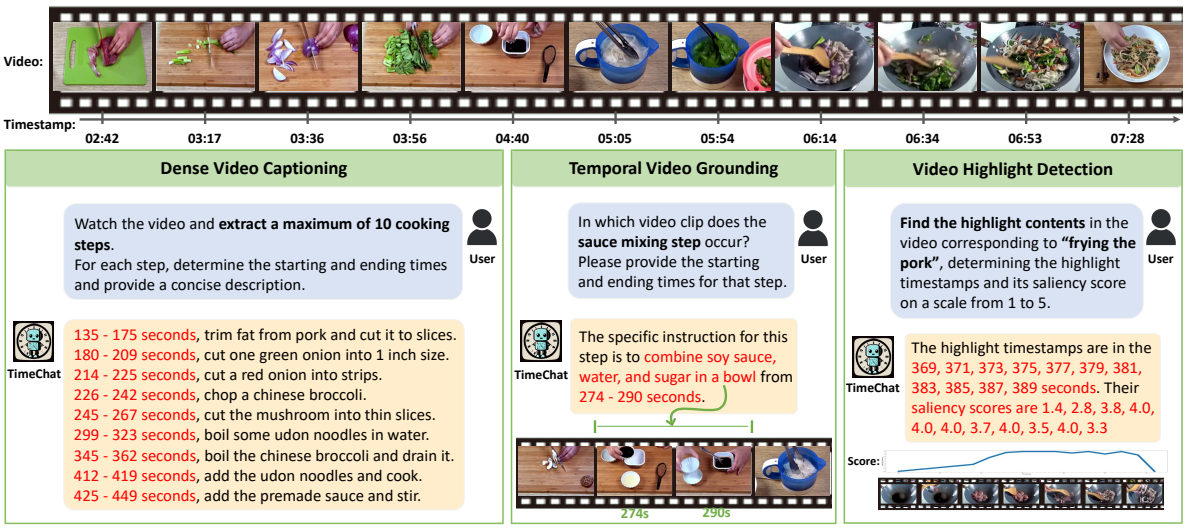

Figure 1. **An illustration of temporal localization capability of TimeChat.** TimeChat is a time-sensitive video large language model. Its capabilities extend beyond regular video captioning and Q&A. Notably, TimeChat can also follow user instructions to **(1)** summarize key events and pinpoint their moments in long videos (left block), **(2)** locate the start and end timestamps that correspond to user queries (middle block), and **(3)** detect highlight clips within the video (right block).

Abstract

This work proposes TimeChat, a time-sensitive multimodal large language model specifically designed for long video understanding. Our model incorporates two key architectural contributions: (1) a timestamp-aware frame encoder that binds visual content with the timestamp of each frame, and (2) a sliding video Q-Former that produces a video token sequence of varying lengths to accommodate videos of various durations. Additionally, we construct an instruction-tuning dataset, encompassing 6 tasks and a total of 125K instances, to further enhance TimeChat's instruction-following performance. Experiment results across various video understanding tasks, such as dense captioning, temporal grounding, and highlight detection, demonstrate TimeChat's strong zero-shot temporal localization and reasoning capabilities. For example, it achieves +9.2 F1 score and +2.8 CIDEr on YouCook2, +5.8 HIT@1 on QVHighlights, and +27.5 R@1 (IoU=0.5) on Charades-STA, compared to state-of-the-art video large language models, holding the potential to serve as a versatile video assistant for long-form video comprehension tasks and satisfy realistic user requirements.[1]

1. Introduction

From educational tutorials to feature films, long-form videos have been an essential medium in our daily lives. However, it is both time-consuming and frustrating for individuals

*Equal contribution

[1]Our code and dataset are available at https://github.com/RenShuhuai-Andy/TimeChat.

to sift through lengthy videos. Instead, human attention is consistently drawn to meaningful or highlighted visual segments such as essential steps in a cooking tutorial [60] or fantastic moments from sports events [15]. An intelligent time-sensitive video assistant to analyze long videos for users, encompassing temporal localization, timestamp detection, and key moment summarization, is a longstanding pursuit of the community. With the emergence of Large Language Models (LLMs) and their impressive capacity to execute human instructions [2, 32, 44, 53], a natural question arises, i.e. Is it feasible to develop an LLM-based assistant for long-form video comprehension tasks to satisfy realistic user requirements?

Preliminary endeavors have been made to integrate video encoders with LLMs for basic video understanding including detailed captioning and question answering [13, 18, 23, 28, 29, 40, 57]. However, existing Video LLMs (VidLLMs) can only capture global visual semantics for short clips and fail to associate the significant video content with accurate timestamps. For example, Video-LLaMA [57] and VideoChat [18] struggle to localize and describe meaningful events in untrimmed videos, leading to a low accuracy verified in Tab. 2. Two main obstacles hinder the performance of existing VidLLMs. **Firstly**, their rigid compression converting video tokens to a fixed number (e.g. 32 [57]) is unsuitable for long-form video input [37]. It neglects the video's duration and results in severe spatial-temporal semantics degradation when processing massive frames from long videos. **Secondly**, they handle video and timestamp information separately without considering the explicit time-vision association thus being unable to localize timestamps accurately.

In this paper, we propose **TimeChat**, a time-sensitive multimodal large language model for long video understanding and accurate temporal localization. To handle long video input, we propose a sliding video Q-Former to accommodate adaptive video token length during the extraction and compression of video features. Specifically, the video Q-Former compresses the frames within a sliding window into video tokens. By temporally moving the window, we can dynamically create a video token sequence of varying lengths to accommodate videos of various durations. It preserves the significant visual semantics of long videos and leads to more expressive and scalable video representation. Furthermore, to enhance the vision-timestamp association, we propose a time-aware frame encoder, which explicitly binds the visual context with the timestamp description of each frame.

To stimulate the intrinsic timestamp localization capability of TimeChat and enhance its instruction-following ability, we construct a novel instruction tuning dataset **TimeIT** involving diverse timestamp-related user instructions. This dataset is compiled from a variety of timestamp-associated long-video datasets with an average video length of 190.8 seconds. It is composed of 6 diverse tasks, 12 widely-used academic benchmarks, and a total of 125K instances. We reformat the original academic datasets into dialog style with manually written high-quality instructions. To our best knowledge, TimeIT is the first time-sensitive video-centric dataset designed for instruction tuning, with the aim to facilitate the development of VidLLMs.

Utilizing the TimeIT dataset, We perform instruct tuning on TimeChat and then assess its performance across various downstream tasks including dense video captioning, temporal video grounding, and video highlight detection. Experimental results show that our model substantially outperforms previous VidLLMs under zero-shot settings, with +9.2 F1 score and +2.8 CIDEr on YouCook2 [60], +5.8 HIT@1 on QVHighlights [15], and +27.5 R@1 (IoU=0.5) on Charades-STA [6], respectively. Furthermore, qualitative results in new domains such as movie [54] and egocentric videos [7] demonstrate the generalization of TimeChat towards a versatile and practical video assistant.

2. Related Work

2.1. Video Large Language Models

With advancements in Large Language Models (LLMs), numerous studies have endeavored to integrate LLMs with a video encoder, thereby harnessing the powerful comprehension and generation capabilities of LLMs for video tasks [13, 18, 23, 28, 29, 40, 57]. These studies typically employ open-source LLMs such as Vicuna [2] and LLaMA [44]. Their key difference lies in how they encode the video into vision tokens compatible with the LLMs. Representative work like VideoChat [18] utilizes a video transformer to encode video features and subsequently implement a Query Transformer (Q-Former) [17] to compress video tokens. Video-LLaMA [57] first uses a vision transformer (ViT) [4] with an image Q-Former to encode individual frames, and then employs a video Q-Former for temporal modeling. However, these methods compress video tokens to a fixed number, resulting in visual semantic degradation when handling lengthy videos. In contrast, our model TimeChat offers adjustable compression rates for visual tokens, increasing adaptability to varying video lengths. Moreover, our model explicitly establishes a frame-level vision-timestamp relationship to improve temporal localization capabilities.

2.2. Vision-Language Instruction Tuning

Inspired by the recent success of instruction tuning on LLMs [34, 47, 48], researchers have adopted vision-language instruction tuning to improve instruction following capabilities of Multimodal LLMs [3, 18, 28, 29, 57, 61]. This primarily entails producing high-quality data with human instructions, which can be categorized into two technical branches. The first branch [3, 19, 51] inte-

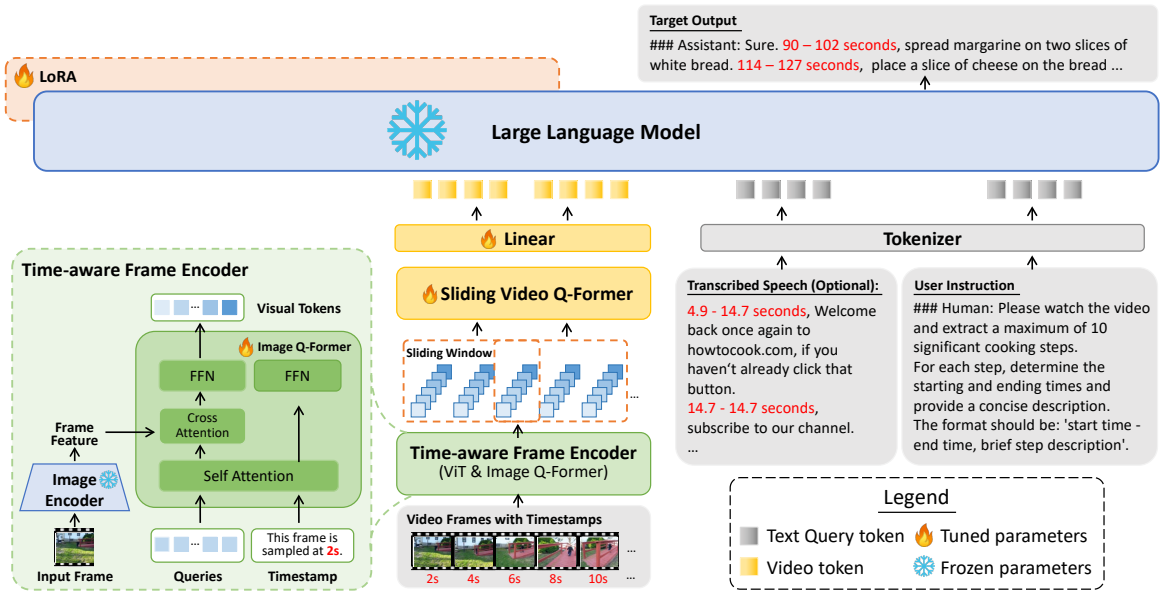

Figure 2. The overall architecture of TimeChat. Input a sequence of video frames along with their timestamps, **(a) Time-aware Frame Encoder** firstly extracts spatial tokens of each frame and binds the visual tokens with the corresponding timestamp description in frame-level. Then **(b) Sliding Video Q-Former** establishes temporal relations across frame tokens with a moving sliding window which produces varied-length video tokens. Finally, the video tokens are concatenated with the optional transcribed speech and the user query as input for a **(c) Large Language Model**, which produces appropriate responses.

grates available multi-modal benchmark datasets and converts them to instruction format, with efforts like MultiInstruct [51], InstructBLIP [3], and M^{3}IT [19]. The second branch [16, 18, 24, 28, 61] leverages LLMs such as ChatGPT [32] and GPT-4 [33] to create more diverse dialog-style data. Approaches like MiniGPT4 [61], LLaVA [24], MIMIC-IT [16], VideoChat [18], and Valley [28] obtain detailed visual descriptions and build image-centric or video-centric conversation data from LLMs. However, they all neglect the time-aware user requests for video understanding. To address this, we propose a time-aware instruction tuning dataset to enhance the time-vision association ability of Multimodal LLMs.

2.3. Video Temporal Localization

Temporal localization is a foundational capability in video understanding tasks, particularly for untrimmed long videos. There have been miscellaneous time-sensitive video tasks, including temporal video grounding [27, 49], dense video captioning [46, 52, 62], video summarization [1, 8, 41, 58], video highlight detection [15, 30], and step localization [5, 39, 55], etc. These tasks necessitate explicit associations between video semantics and the corresponding timestamps. Previous studies [5, 46, 49, 58] tend to settle each task separately on specialized downstream datasets. Although recent works [12, 15, 22, 25, 59] make preliminary attempts to bridge several tasks, a generalist paradigm based on LLMs is under exploration. In this paper, we unify a wide range of time-sensitive video tasks in language modeling format and take a first step to leverage LLMs.

3. Method

In this section, we present TimeChat, a VidLLM featuring two novel modules: a timestamp-aware frame encoder and a sliding video Q-Former. These modules enhance our TimeChat's ability to localize temporally and understand long videos (§ 3.1). To further empower TimeChat to follow human instructions across time-sensitive video tasks, we collect an instruction-tuning dataset named TimeIT (§ 3.2). This dataset comprises 6 tasks and 125K instances. Based on TimeIT, we perform instruction tuning on our model to unlock its full potential.

3.1. TimeChat Architecture

3.1.1 Overview

TimeChat is composed of a time-aware frame encoder, a sliding video Q-Former, and a large language model, as depicted in Fig. 2. Given an input video, the frame encoder first extracts visual and timestamp features for each frame independently. Next, the video Q-Former models temporal relations across frames within a sliding window to produce video tokens. Finally, these video tokens are concatenated with optional transcribed speech and user instructions, which are then fed into the LLM to generate responses.

3.1.2 Timestamp-aware Frame Encoder

Previous studies typically separate the modeling of visual semantics and their respective timestamp information of input frames [18, 57]. For example, VideoChat [18] utilizes a visual encoder to process visual frame semantics but an LLM to receive timestamp information, e.g. "`This video contains 8 frames sampled at 2s, 4s, ..., 16s`". As a result, this method fails to directly capture the time when a visual event occurs. Some alternative approaches add learnable position (time) embeddings to the visual tokens [57]. However, this only enables models to discern frame order, lacking precision in determining the exact temporal moment.

To mitigate these issues, we introduce a time-aware frame encoder (green block in Fig. 2) inspired by InstructBLIP [3]. Given a video input $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$, the encoder first uses a pre-trained image encoder, i.e. ViT [35, 42], to encode each frame and obtain frame features. Subsequently, an image Q-Former further compresses the frame tokens. As Fig. 2 illustrates, the Q-Former takes N_I learnable queries in dimension D_Q as input. These queries interact with the frame features via cross-attention and update the initial queries to final N_I visual tokens in dimension D_Q [3, 17]. Importantly, during visual token extraction, we add the frame's timestamp[2], e.g., "`This frame is sampled at 2s.`", as a condition to the Q-Former to fuse the visual and timestamp information. While our approach shares structural similarities with InstructBLIP's Q-Former, the motivations differ. InstructBLIP uses instructions (e.g., "`Choose the correct option to the following question: ...`" for VQA tasks) as additional input to extract task-relevant visual tokens, whereas our Q-Former takes timestamp descriptions to bind time information to the visual tokens.

3.1.3 Sliding Video Q-Former

After applying the time-aware frame encoder, we obtain $T \times N_I$ visual tokens for a T-frames video input. Since frames are encoded independently, the temporal relationship across frames has not been modeled yet. To this end, we incorporate a sliding video Q-Former (yellow block in Fig. 2) to enhance the feature fusion in the temporal dimension. The video Q-Former mirrors the structure of the image Q-Former, it takes N_V learnable queries in dimension D_Q as input without timestamps. We design a sliding window of length L_W and within each window utilizing the video Q-Former extract N_V video tokens from L_W frames. By sliding the video Q-Former in strides of S, we can represent the input video as $(T/S) \times N_V$ video tokens.

Considering the 3D nature of videos and the redundancy in space-time information, the original sequence of visual tokens (i.e., patches in all frames) can be extremely long [37]. Thus, it's crucial to condense video information to a reduced number of video tokens, thereby decreasing the computation burden on the LLM. However, previous work [13, 18, 23, 57] usually set a fixed number of video tokens N_V, such as 32, which can result in severe visual semantics degradation when the number of input frames T is large. Concretely, we define the compression rate R as the ratio of the number of original visual tokens to the number of final video tokens. The compression rate for previous work like Video-LLaMA [57] is:

$$R = (T \times N_P)/N_V, \tag{1}$$

where N_P is the number of patches of each frame. This ratio increases with the number of input frames T and can cause excessive compression for long videos. With our sliding video Q-Former, our compression rate $R^{'}$ becomes a constant value:

$$R^{'} = \frac{T \times N_P}{(T/S) \times N_V} = \frac{S \times N_P}{N_V}, \tag{2}$$

retaining richer semantics for long videos. By adjusting the stride S, we can control the final number of video tokens according to the computation budget. Finally, we use a linear layer to transform the dimension D_Q of video tokens to match the dimension D_{LLM} of the LLM embedding space.

3.1.4 Large Language Model

Ultimately, we concatenate inputs from various modalities, e.g., the video tokens $\mathbf{X}_v$, text query tokens $\mathbf{X}_q$ (including optional transcribed speech and user instruction), and feed these into a large language model to generate reasonable and coherent responses (answers) $\mathbf{X}_a$. Here, $\mathbf{X}_v$, $\mathbf{X}_q$, and $\mathbf{X}_a$ have the same token embedding dimension D_{LLM}. The training of the VidLLM typically utilizes a two-stage training framework. The first stage pre-trains the model using large-scale image/video-text pairs for vision-language alignment [17, 20, 26, 36, 38, 50]. The second stage finetunes the model with instruction data for instruction following. Considering computing efficiency, we reuse the checkpoints of the existing open source models after the first stage training (see § 4.1), conducting only instruction tuning. During the training procedure, we utilize the language modeling loss for generating target answers $\mathbf{X}_a$ with length L_T, which serves as the objective function:

$$\begin{aligned} \mathcal{L} &= -\log P_\theta(\mathbf{X}_a \mid \mathbf{X}_v, \mathbf{X}_q) \\ &= -\sum_{i=1}^{L_T} \log P_\theta\left(x_i \mid \mathbf{X}_v, \mathbf{X}_q, \mathbf{X}_{a,<i}\right), \end{aligned} \tag{3}$$

where θ is the trainable parameters, and $\mathbf{X}_{a,<i}$ refers to the answer tokens preceding the current prediction token x_i. To better adapt the LLM to video tasks, we apply the parameter-efficient fine-tuning method, LoRA [10].

[2] Vid2Seq [52] shows comparable performance using relative and absolute timestamps (Tab.13, N=500). For chat-friendliness and simplicity, we opt for the absolute ones.

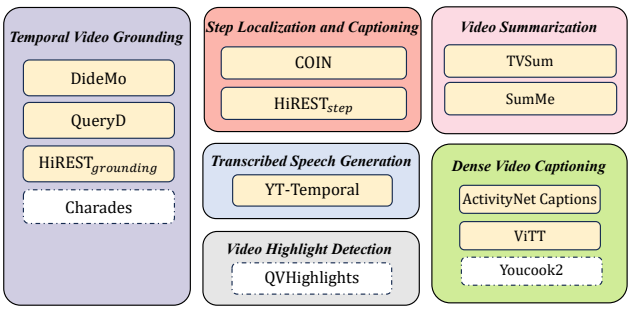

Figure 3. Involved tasks and datasets in the time-aware instruction tuning dataset. The held-in datasets are colored with yellow blocks while the held-out datasets are in dashed white blocks.

3.2. Instruction Data TimeIT

To boost TimeChat's ability to understand time-sensitive human instructions, we introduce TimeIT, a video-centric instruction-tuning dataset involving timestamps. This dataset integrates a wide range of timestamp-associated video datasets and is characterized by long-form videos.

3.2.1 Task Coverage

TimeIT encompasses 6 longstanding timestamp-related video tasks, i.e., (1) dense video captioning, (2) temporal video grounding, (3) step localization and captioning, (4) video summarization, (5) video highlight detection, as well as (6) transcribed speech generation. It also incorporates 12 specific datasets [6, 8, 9, 11, 14, 15, 31, 41, 43, 55, 56, 60] derived from different domains as illustrated in Fig. 3. Please refer to Appendix A for details. Our dataset accommodates prevalent user requests involving video timestamps when interacting with AI assistants in real-world applications.

3.2.2 Data Construction

We convert the above datasets into an instruction-following format to obtain high-quality video-centric instruction data. The construction process comprises two main steps including (1) instruction writing and (2) answer formatting.

Step I: Instruction Writing. The quality and diversity of instructions are essential in the construction process. We manually write well-designed instructions for each task as a good starting. Then we utilize GPT-4 [33] to extend more diverse and flexible expressions based on the manual initialization. Eventually, we manually select and refine the LLM-generated instructions to obtain the final version. Inspired by the observation in M^{3}IT [19] that using around five instructions per task is sufficient, we generate six high-quality instructions for each task. Specific instructions designed for each task are depicted in Appendix B.

Dataset	Time-aware	# Tasks	# Samples	Avg. video len
VideoChat [18]	✗	3	11K	18.0s
Valley [28]	✗	2	73K	21.9s
Video-ChatGPT [29]	✗	1	100K	115.2s
TimeIT (Ours)	✓	6	125K	190.8s

Table 1. Comparison with existing video instruction tuning data.

Step II: Answer Formatting. Based on the written instructions, we further reformulate the task outputs into a user-friendly natural language response paradigm (format details are provided in Appendix B). Considering the involved video datasets are manually collected, the overall quality of TimeIT data is guaranteed.

Tab. 1 compares our TimeIT data with existing video-centric instruction tuning data, revealing our significant advantages across data scale, task diversity, and video length. Appendix C provides contribution analysis of each task to model performance. Overall, all 6 tasks are beneficial.

4. Experiments

4.1. Implementation Details

We take ViT-G/14 from EVA-CLIP [42] as the image encoder and LLaMA-2 (7B) [45] as the language foundation model. The parameters of the image Q-Former are initialized from InstructBLIP's checkpoint, while the video Q-Former is initialized from Video-LLaMA's checkpoint. We finetune our TimeChat on TimeIT and Valley [28] for 3 epochs, using a batch size of 32, with a single 8-V100 (32G) machine. As shown in Fig. 2, the parameters of ViT and LLM are frozen, while those of image Q-Former, video Q-Former, and linear layer are tuned. The rank in LoRA is 32. The window size L_W, stride S, and the number of video tokens N_V per window are 32. The number of input frames is 96. Please refer to Appendix D for additional hyper-parameters.

4.2. Evaluation Setups

Tasks, Datasets and Evaluation Metrics. We evaluate our model on three tasks of long video understanding, i.e., dense captioning, temporal grounding, and highlight detection, in a zero-shot setting. The evaluation datasets include YouCook2 [60], Charades-STA [6], and QVHighlights [15]. See Appendix E for details of evaluation metrics.

Heuristic Rules for Parsing LLM Outputs. It is important to note that the outputs generated by LLMs may include colloquial expressions, leading to a wide range of response variations. Accordingly, we carefully devise a considerable number of heuristic rules to guarantee that predicted answers can be accurately extracted from the model's responses for the computation of final metrics.

Model	LLM Size	Dense Video Captioning YouCook2 [60]			Highlight Detection QVHighlights [15]		Temporal Grounding Charades-STA [6]	
		SODA_c	CIDEr	F1 Score	mAP	HIT@1	R@1 (IoU=0.5)	R@1 (IoU=0.7)
Multi-model Pipelines								
InstructBLIP [3]+ChatGPT [32]	-	0.5	1.0	8.2	25.6	53.0	7.0	1.0
VideoChat-Text (w/ ChatGPT) [18]	-	0.4	0.9	8.4	17.5	31.0	9.0	3.0
End2end VidLLMs								
Video-LLaMA [57]	13B	0.0	0.0	0.1	11.1	15.6	2.1	0.6
VideoChat-Embed [18]	13B	0.5	1.0	7.0	14.2	18.9	2.6	0.8
Valley [28]	7B	0.1	0.0	1.5	10.9	15.2	4.7	1.6
Video-LLaMA [57]	7B	0.0	0.0	0.1	11.3	15.6	2.7	1.2
VideoChat-Embed [18]	7B	0.2	0.6	3.4	13.1	18.1	3.2	1.4
TimeChat (Ours)	7B	**1.2** (+1.0)	**3.4** (+2.8)	**12.6** (+9.2)	**14.5** (+1.4)	**23.9** (+5.8)	**32.2** (+27.5)	**13.4** (+11.8)

Table 2. Zero-shot performance on three video tasks of dense captioning (YouCook2), highlight detection (QVHighlights), and temporal grounding (Charades). For all metrics, higher is better. We gray out methods that use 13B LLMs for fair comparison. The performance gain in green is compared to the best-performing 7B model. We do not use transcribed speech for evaluation.

Model	SODA_c	CIDEr	F1 Score
TimeChat	3.1	10.3	19.5
w/o sliding video Q-Former	2.1 (-1.0)	7.5 (-2.8)	16.5 (-3.0)
w/o timestamp-aware frame encoder	1.8 (-0.3)	6.1 (-1.4)	14.2 (-2.3)

Table 3. Ablation study. We remove the two key modules respectively, then finetune and evaluate our model on YouCook2.

Model	SODA_c	CIDEr	F1 Score
Video-LLaMA [57]	1.8	6.1	14.2
VideoChat-Embed [18]	2.2	8.1	15.6
TimeChat (Ours)	**3.1**	**10.3**	**19.5**

Table 4. Performance of various VidLLMs (7B) after fine-tuneing on YouCook2 (w/o using the TimeIT dataset).

Compared Methods. We compare our model with two branches of baselines. **(1) Multi-model Pipielines**, including VideoChat-Text [18], InstructBLIP [3]+ChatGPT [32]. These pipelines integrate specialized visual models with ChatGPT, which firstly convert video semantics (e.g. frame descriptions, clip captions or action tags) into textual descriptions and then leverage ChatGPT to process all inputs to solve the target task. See Appendix F for more details. **(2) End2end Models**, including Valley [28], VideoChat-Embed [18], Video-LLaMA [57] with 7B LLMs. These models directly take videos as inputs and generate responses in an end2end manner.

4.3. Zero-shot performance

Tab. 2 shows the zero-shot performance of TimeChat (7B), which outperforms previous VidLLMs (7B/13B) in all tasks.

Dense Video Captioning. This task on YouCook2 is quite challenging. The model is required to accurately identify roughly 8 essential cooking steps within the average video duration of 320 seconds, alongside providing faithful descriptions that match the visual content. Moreover, the specialized nature of cooking amplifies the task complexity, thereby challenging the model's generalizability. Existing end-to-end VidLLMs struggle with precise moment localization, as evidenced by the low F1 score of 3.4 achieved by the top-performing VideoChat-Embed model. Such imprecision in moment localization significantly impacts the captioning evaluation, with both SODA_c and CIDEr metrics approaching zero. Compared to them, our model achieves remarkable performance gains exceeding the previous SOTA by +1.0 SODA_c, +2.8 CIDEr, and +9.2 F1 score. This reveals that TimeChat effectively processes lengthy videos with precise temporal localization capability. Moreover, our performance also significantly surpasses the multi-model pipelines powered by ChatGPT (F1 score from 8.4 to 12.6), which demonstrates both the challenging nature of this task and the superiority of our model in processing long videos.

Highlight Detection. While the dense video captioning task focuses on localizing events at the clip level, this task requires a more fine-grained video comprehension at the frame level. For an input video, it aims to output the times and the related salient scores of highlight frames. Overall, our model achieves a 14.5 mAP and 23.9 HIT@1 on QVHighlights, surpassing the previous VidLLMs by +1.4 and +5.8 points, respectively. This highlights the contribution of our timestamp-aware frame encoder in identifying the salient semantics of each frame. Moreover, this task is a held-out task in TimeIT, indicating the generalization ability of our model on novel tasks. As for multi-model pipeline approaches, they achieve even stronger performance. We speculate that this is due to the format of highlight detection being more compatible with their methods, as the model receives a series of joint timestamp-visual descriptions for

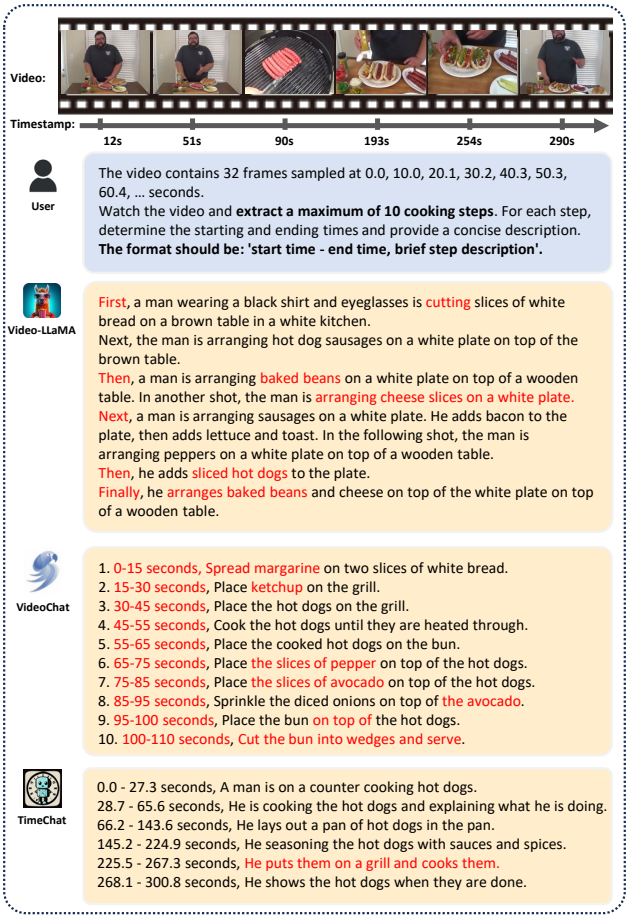

Figure 4. Case study (in zero-shot). The red text highlights inaccuracies in the model-generated timestamp or description.

input frames. This enables the frame-by-frame assessment by the LLM, facilitating more accurate judgments.

Temporal Grounding. This task aims to identify the corresponding timestamp described by a query sentence. TimeChat achieves 32.2 points on "R@1, IoU=0.5" of the Charades-STA dataset, which surpasses the previous SOTA end2end VidLLM, i.e. Valley, by a substantial margin (+27.5). This demonstrates that our model excels at accurately localizing the video moment content referred to a given text query. Notably, TimeChat gains the most improvements on the temporal grounding task, we argue that this task mainly emphasizes the temporal localization capability of long videos which is exactly the best advantage of TimeChat.

4.4. Qualitative Evaluation

Compared to Existing VidLLMs. Fig. 4 presents qualitative comparisons between TimeChat and other VidLLMs in zero-shot settings. Video-LLaMA falls short in fully adhering to the user instruction, as it only describes the cooking steps without the corresponding start and end timestamps for each step. VideoChat, on the other hand, produces captions that fit the requested format but misplaces the timing of all the steps. Despite this, the generated description from VideoChat includes several hallucinations [21], such as references to "pepper" and "avocados" that are not present in the video. In contrast, TimeChat demonstrates improved temporal localization and summarization capabilities compared to the previous models. It successfully matches the content of the video for almost all extracted clips. Furthermore, the occurrence of hallucinations is significantly reduced. However, there is still room for improvement in terms of enhancing the richness and details in the summarization generated by our model.

Generalized to New Domains. In Appendix G, we show qualitative results in new domains such as movie [54] and egocentric videos [7], demonstrating the generalization of TimeChat to novel scenarios. This generalization is a key characteristic towards a practical video assistant and represents a fundamental difference between LLM-based TimeChat and the current specialized models tailored for specific downstream datasets. More cases can be found in Appendix H.

4.5. Ablation Study

We conducted an ablation study based on YouCook2 to assess the efficacy of key designs in our TimeChat. As illustrated in Tab. 3, when the sliding video Q-Former is removed, the number of final visual tokens decreases from 96 to 32, resulting in a $3\times$ information compression rate. This reduction in semantic information leads to a decrease in the alignment between the generated descriptions and the video content. Specifically, the SODA_c metric decreases by 1.0, while the CIDEr metric decreases by 2.8. Additionally, the accuracy of timestamps (measured by F1 score) decreases by 3.0. In the case of the removal of the timestamp-aware frame encoder, the model's ability to temporally ground the descriptions diminishes dramatically, as indicated by a decrease of 2.3 in the F1 score. These results highlight the effectiveness of the two novel modules in the model.

4.6. Further Analysis

We provide further analysis to validate the superiority of our model. To demonstrate that the performance gain of our model is not solely attributed to the new TimeIT dataset, but also to the improvements in our model architecture, we conduct fine-tuning and evaluation using only the YouCook2 dataset. In this setup, we initialize our model with existing open-source checkpoints (see § 4.1). For all the models, we finetune their Q-Formers and apply LoRA [10] for their LLMs. Tab. 4 presents the results, showing that our model consistently outperforms previous models across all metrics, with increases of +2.2 CIDEr and +3.9 F1 score.

Model	Generalist Model	Finetune Epochs	Dense Video Captioning YouCook2 [60]			Highlight Detection QVHighlights [15]		Temporal Grounding Charades-STA [6]	
			SODA_c	CIDEr	F1 Score	mAP	HIT@1	R@1 (IoU=0.5)	R@1 (IoU=0.7)
Vid2Seq [52]	✗	40	7.9	47.1	27.3	-	-	-	-
QD-DETR [30]	✗	200	-	-	-	38.9	62.4	-	-
QD-DETR$_{w/Audio}$ [30]	✗	200	-	-	-	39.0	62.9	-	-
MMN [49]	✗	18	-	-	-	-	-	50.5	29.7
VDI [27]	✗	18	-	-	-	-	-	52.3	31.4
TimeChat (Ours)	✓	3	3.4	11.0	19.5	21.7	37.9	46.7	23.7

Table 5. Supervised performance compared to task-specific models. They use specific objective functions and extra fine-tuning epochs to better fit the downstream dataset. In contrast, TimeChat exhibits a broad versatility and generalization across various tasks and domains.

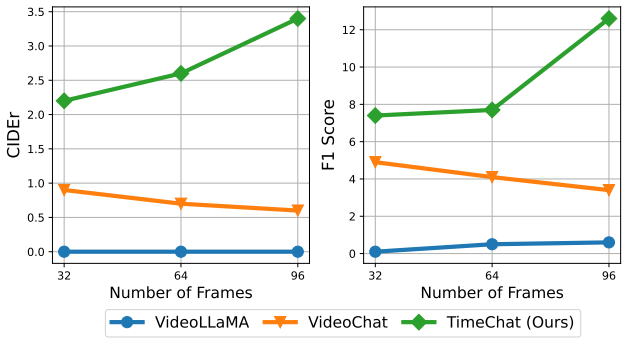

Figure 5. Comparison of CIDEr and F1 score w.r.t the number of input frames on the YouCook2 dataset (zero-shot).

In Fig. 5, we examine the performance scalability of our model with respect to the number of input frames [37]. As mentioned in § 3.1.3, previous models like Video-LLaMA and VideoChat compress excessive information for long videos, resulting in minimal performance impact when increasing the number of input frames from 32 to 96. In contrast, our TimeChat decouples the number of frames T and the compression rate R' using the sliding video Q-Former. Our curve exhibits linear improvement in performance as the number of frames increases, showcasing superior scalability.

4.7. Comparison with Specialized Models

In this subsection, we compare our generalist model, TimeChat, with state-of-the-art specialized models on the three tasks, respectively. Given that all specialized models have been fine-tuned on specific datasets, we also finetune our model for a fair comparison. As shown in Tab. 5, after fine-tuning, TimeChat has made further performance gains, e.g., +6.9 F1 score on YouCook2, +16.9 HIT@1 on QVHighlights, and +16.4 R@1 (IoU=0.5) on Charades-STA. Nonetheless, there is still much room for boosting our approach compared to specialized models, whose superior performance arises from task-specific designs. For example, Vid2Seq [52] pretrains on YT-Temporal-1B [56], which contains much more high-quality long videos than the tuning dataset we used. QD-DETR [30] employs a special saliency token for saliency prediction and introduces 4 loss functions for training, while our model trains purely through language modeling. In addition, these models also used much more fine-tuning steps to better fit the downstream dataset. However, as a generalized model, TimeChat exhibits a strong generalization ability in zero-shot scenarios, multi-task, and multi-domain settings, which is not present in those expert models. Achieving state-of-the-art performance on every task is not the major goal of this paper, and we leave this as future work.

5. Discussion and Conclusion

We present TimeChat, a time-sensitive VidLLM for long video understanding. Benefiting from the novel time-aware frame encoder, sliding video Q-Former, and instruction tuning on TimeIT, our model demonstrates strong temporal localization capabilities that were absent in previous VidLLMs. Through its ability to identify significant events within lengthy videos, pinpoint events' start and end times, and generate concise summarization, TimeChat makes a crucial step toward an intelligent video assistant. In the future, we will make architectural advances to improve video semantic density while reducing spatial-temporal redundancy. We will also collect more diverse and high-quality instruction-tuning data to broaden the time-related applications.

Acknowledgements

This work is supported in part by a Huawei Research Grant and National Natural Science Foundation of China (No. 62176002). Xu Sun is the corresponding author.

References

[1] Evlampios Apostolidis, E. Adamantidou, Alexandros I. Metsai, Vasileios Mezaris, and I. Patras. Video summarization using deep neural networks: A survey. *Proceedings of the IEEE*, 109:1838–1863, 2021. 3

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[3] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv*, abs/2305.06500, 2023. 2, 3, 4, 6

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 2

[5] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18952–18961, 2023. 3

[6] J. Gao, Chen Sun, Zhenheng Yang, and Ramakant Nevatia. Tall: Temporal activity localization via language query. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5277–5285, 2017. 2, 5, 6, 8

[7] 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 K. Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina González, James M. 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 K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, 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. Ego4d: Around the world in 3,000 hours of egocentric video. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990, 2021. 2, 7

[8] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision*, 2014. 3, 5

[9] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5804–5813, 2017. 5

[10] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. 4, 7

[11] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In *AACL*, 2020. 5

[12] Hao Jiang. Joint video summarization and moment localization by cross-task sample transfer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16367–16377, 2022. 3

[13] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023. 2, 4

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 706–715, 2017. 5

[15] Jie Lei, Tamara L. Berg, and Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries. *ArXiv*, abs/2107.09609, 2021. 2, 3, 5, 6, 8

[16] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, C. Li, and Ziwei Liu. Mimic-it: Multimodal in-context instruction tuning. *ArXiv*, abs/2306.05425, 2023. 3

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597, 2023. 2, 4

[18] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wen Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *ArXiv*, abs/2305.06355, 2023. 2, 3, 4, 5, 6

[19] Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. M3it: A large-scale dataset towards multi-modal multilingual instruction tuning. *ArXiv*, abs/2306.04387, 2023. 2, 3, 5

[20] 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*, abs/2311.17404, 2023. 4

[21] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji rong Wen. Evaluating object hallucination in large vision-language models. *ArXiv*, abs/2305.10355, 2023. 7

[22] 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. 3

[23] Haogeng Liu, Qihang Fan, Tingkai Liu, Linjie Yang, Yunzhe Tao, Huaibo Huang, Ran He, and Hongxia Yang. Video-

teller: Enhancing cross-modal generation with fusion and decoupling. *ArXiv*, abs/2310.04991, 2023. 2, 4

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *ArXiv*, abs/2304.08485, 2023. 3

[25] Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3042–3051, 2022. 3

[26] Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation. *ArXiv*, abs/2311.01813, 2023. 4

[27] Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Towards generalisable video moment retrieval: Visual-dynamic injection to image-text pre-training. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23045–23055, 2023. 3, 8

[28] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Ming-Hui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *ArXiv*, abs/2306.07207, 2023. 2, 3, 5, 6

[29] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ArXiv*, abs/2306.05424, 2023. 2, 5

[30] WonJun Moon, Sangeek Hyun, Sang shin Paldal-gu Suwon-city Park, Dongchan Park, and Jae-Pil Heo. Query - dependent video representation for moment retrieval and highlight detection. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23023–23033, 2023. 3, 8

[31] Andreea-Maria Oncescu, João F. Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. Queryd: A video dataset with high-quality text and audio narrations. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2265–2269, 2020. 5

[32] OpenAI. Introducing chatgpt. 2022. 2, 3, 6

[33] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 3, 5

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 2

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 4

[36] Shuhuai Ren, Junyang Lin, Guangxiang Zhao, Rui Men, An Yang, Jingren Zhou, Xu Sun, and Hongxia Yang. Learning relation alignment for calibrated cross-modal retrieval. In *Annual Meeting of the Association for Computational Linguistics*, 2021. 4

[37] Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. TESTA: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 2023. 2, 4, 8

[38] Shuhuai Ren, Lei Li, Xuancheng Ren, Guangxiang Zhao, and Xu Sun. Delving into the openness of CLIP. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2023. 4

[39] Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2021. 3

[40] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tianbo Ye, Yang Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. *ArXiv*, abs/2307.16449, 2023. 2

[41] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187, 2015. 3, 5

[42] Quan Sun, Yuxin Fang, Ledell Yu Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *ArXiv*, abs/2303.15389, 2023. 4, 5

[43] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1207–1216, 2019. 5

[44] 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*, abs/2302.13971, 2023. 2

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5

[46] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6847–6857, 2021. 3

[47] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 2

[48] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv e-prints*, pages arXiv–2204, 2022. 2

[49] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric

learning for temporal grounding. In *AAAI Conference on Artificial Intelligence*, 2021. 3, 8

[50] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, and Florian Metze Luke Zettlemoyer Christoph Feichtenhofer. Videoclip: Contrastive pretraining for zero-shot video-text understanding. In *Conference on Empirical Methods in Natural Language Processing*, 2021. 4

[51] Zhiyang Xu, Ying Shen, and Lifu Huang. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. In *Annual Meeting of the Association for Computational Linguistics*, 2022. 2, 3

[52] 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. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10714–10726, 2023. 3, 4, 8

[53] Linli Yao, Yuanmeng Zhang, Ziheng Wang, Xinglin Hou, Tiezheng Ge, Yuning Jiang, and Qin Jin. Edit as you wish: Video description editing with multi-grained commands, 2023. 2

[54] Zihao Yue, Qi Zhang, Anwen Hu, Liang Zhang, Ziheng Wang, and Qin Jin. Movie101: A new movie understanding benchmark. In *Annual Meeting of the Association for Computational Linguistics*, 2023. 2, 7

[55] Abhaysinh Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Ouguz, Yasher Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23056–23065, 2023. 3, 5

[56] 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. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16354–16366, 2022. 5, 8

[57] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *ArXiv*, abs/2306.02858, 2023. 2, 4, 6

[58] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 766–782. Springer, 2016. 3

[59] Qi Zhang, Yuqing Song, and Qin Jin. Unifying event detection and captioning as sequence generation via pre-training. In *European Conference on Computer Vision*, pages 363–379. Springer, 2022. 3

[60] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2017. 2, 5, 6, 8

[61] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *ArXiv*, abs/2304.10592, 2023. 2, 3

[62] Wanrong Zhu, Bo Pang, Ashish V Thapliyal, William Yang Wang, and Radu Soricut. End-to-end dense video captioning as sequence generation. *arXiv preprint arXiv:2204.08121*, 2022. 3