Supplementary Material for "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

A. Task Coverage in TimeIT

TimeIT encompasses 6 longstanding timestamp-related video tasks and incorporates 12 specific datasets derived from different domains.

Dense Video Captioning (DVC). This task unifies the event localization and event captioning subtasks. It detects a series of events in the given video and outputs the corresponding timestamps and descriptions. We gather ActivityNet Captions [8], ViTT [7], and YouCook2 [25] datasets to facilitate the narration of significant events for users when watching long videos.

Temporal Video Grounding (TVG). This task aims to predict a timestamp boundary including the start and end time in the video given a natural language query. We include DiDeMo [6], QuerYD [13], HiREST$_{grounding}$ [23], and Charades-STA [3] datasets to achieve accurate moment localization when users interact with natural language.

Step Localization and Captioning (SLC). This task is designed to automatically segment and describe significant steps in a long untrimmed video, which is useful for instructional videos. We incorporate two datasets including COIN [19] and HiREST$_{step}$ to fulfill key steps detecting when processing noisy instructional videos under the cooking, repairing, or assembling furniture scenarios.

Video Summarization (VS). The goal is to create a compressed set of frames or clip shots to represent the most informative content of the given video. TVSum [18] and SumMe [5] datasets are compiled to achieve an efficient video overview for busy stakeholders to save time.

Video Highlight Detection (VHD). Different from the video summarization, it identifies the most exciting, impressive, or emotional moments that may not cover the full scope of the original video. QVHighlights [9] dataset is utilized to evaluate the highlight moment recommendation ability of AI assistants.

Transcribed Speech Generation (TSG). The objective of this task is to predict the speech content and its corresponding start and end timestamps based on visual signals in the video. This task can be regarded as a weakly-supervised event localization and description task. We use the YT-Temporal-1B dataset [24]. The original dataset includes 18 million narrated videos collected from YouTube, while we sample 31.6K videos from it for instruction tuning. Following Vid2Seq [21], we leverage Whisper-timestamped [12, 16] to automatically transcribe speech and use it as the target answer.

Our dataset accommodates prevalent user requests involving video timestamps when interacting with AI assistants in real-world applications.

B. Instructions for Each Task

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 [15] 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 [11] that using around five instructions per task is sufficient, we generate six high-quality instructions for each task. Tab. 1 shows instruction template examples and formatted output answers for each task.

*Equal contribution

Key	Value
Dense Video Captioning	
Instruction Example	Localize a series of activity events in the video, output the start and end timestamp for each event, and describe each event with sentences.
Output Format	<timestamp_start> - <timestamp_end> seconds, <event_description> . ···
Output Example	90 - 102 seconds, spread margarine on two slices of white bread in the video. 114.0 - 127.0 seconds, place a slice of cheese on the bread. ···
Temporal Video Grounding	
Instruction Example	Detect and report the start and end timestamps of the video segment that semantically matches the given textual query <query_placeholder> .
Output Format	The given query happens in <timestamp_start> - <timestamp_end> seconds.
Output Example	The given query happens in 0.0 - 6.9 seconds.
Step Localization and Captioning	
Instruction Example	Identify and mark the video segments corresponding to a series of actions or steps, specifying the timestamps and describing the steps.
Output Format	<timestamp_start> - <timestamp_end> seconds, <step_description> . ···
Output Example	21.0 - 22.0 seconds, begin to run up. 23.0 - 24.0 seconds, begin to jump up. 25.0 - 26.0 seconds, fall to the ground.
Video Summarization	
Instruction Example	Generate a summarized version of the video, focusing on extracting key frames that best represent the overall narrative. The output should be a list of timestamps in seconds and their corresponding salient scores.
Output Format	The key timestamps are in the <timestamp_1> , <timestamp_2> , ··· seconds, Their saliency scores are <score_1> , <score_2> , ··· .
Output Example	The key timestamps are in the 8.5, 10.0, 11.0, 12.0, 23.5, 44.5, 45.0 seconds. Their saliency scores are 1.8, 3.7, 4.5, 4.2, 2.1, 4.7, 4.2.
Video Highlight Detection	
Instruction Example	Watch the provided video and mark out the scenes that stand out based on the description: <query_placeholder> . Document the timestamps of these highlights and evaluate their saliency scores.
Output Format	There are <highlight_moments_number> highlight moments in the <timestamp_1> , <timestamp_2> , ··· seconds, Their saliency scores are <score_1> , <score_2> , ··· .
Output Example	There are 16 highlight moments in the 44.0, 46.0, 48.0, 50.0, 52.0, 54.0, 56.0, 58.0, 60.0, 62.0, 64.0, 66.0, 68.0, 70.0, 72.0, 74.0 second. Their saliency scores are 2.7, 4.0, 3.7, 3.3, 2.7, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 2.7, 3.0, 3.0, 3.0.
Transcribed Speech Generation	
Instruction Example	Watch the video, transcribe the speech, and indicate when each segment starts and ends.
Output Format	Transcribed speech: <timestamp_start> - <timestamp_end> seconds, <transcribed_speech> . ···
Output Example	Transcribed speech: 4.0 - 9.3 seconds, Dolby as well as we had over 7.7 million minutes viewed. This week we visit restaurant. 9.3 - 15.4 seconds, August by Chef John Besh in New Orleans 2015. Restaurant August is currently regarded as New. ···

Table 1. Instruction template examples and formatted output answer for each task.

C. Contribution Analysis of Each Task to Model Performance

We examine the impact of individual tasks within the TimeIT dataset on model performance. Initially, we construct the TimeIT dataset with only DVC and TVG tasks, then gradually integrating additional tasks such as SLC, VS, and TSG, to assess their influence on model performance. As shown in Tab. 2, introducing similar tasks (e.g., SLC to DVC) yields a positive impact (e.g., increasing F1 score on YouCook2 from 5.9 to 12.1). Overall, all 6 tasks are beneficial.

D. Hyper-parameters for Instruction Tuning

Tab. 3 lists hyper-parameters for instruction tuning. We also conduct an ablation on sliding window hyper-parameters. The results are on Tab. 4. We adopt a window size=stride=32 for efficiency (higher compression rate [17]

Tasks in TimeIT	Dense Captioning YouCook2			Highlight Detection QVHighlights		Temporal Grounding Charades-STA	
	SODA_c	CIDEr	F1	mAP	HIT@1	R@1$_{(IoU=0.5)}$	R@1$_{(IoU=0.7)}$
DVC+TVG	0.6	1.9	5.9	11.2	15.5	**34.9**	13.6
DVC+TVG+**SLC**	1.1	3.2	12.1	11.8	16.5	32.7	13.9
DVC+TVG+SLC+**VS**	1.1	3.0	12.2	13.0	19.0	33.2	**14.3**
DVC+TVG+SLC+VS+**TSG**	**1.2**	**3.4**	**12.6**	**14.5**	**23.9**	32.2	13.4

Table 2. Contributions (positive / negative) of tasks in TimeIT to model performance. The tasks include Dense Video Captioning (DVC), Temporal Video Grounding (TVG), Step Localization and Captioning (SLC), Video Summarization (VS), and Transcribed Speech Generation (TSG).

Hyper-parameter	Value
Patch size	14×14
Frame resolution	224×224
Fine-tuning epochs	3
Batch size	32
Learning rate	3e-5
Warm-up learning rate	1e-6
Weight decay	0.05
AdamW β	(0.9, 0.999)
Window size L_W	32
Stride S	32
Number of video tokens per window N_V	32
Number of input frames T	96
Max text length	2048
Number of layers in video Q-Former	2
Number of layers in image Q-Former	12
Hidden size of image/video Q-Former (D_Q)	768
Hidden size of LLaMA-2 (D_{LLM})	4096

Table 3. Hyper-parameters for instruction tuning.

and fewer video tokens). However, a more thorough search may improve performance (window size=stride=16). Besides, non-overlapping windows outperform overlapping ones.

E. Details of Evaluation Datasets and Metrics

TimeIT's 6 tasks can be grouped based on format similarity: **(1)** dense-captioning-formatted: DVC, SLC, and TSG; **(2)** highlight-detection-formatted: HD and VS; **(3)** temporal-grounding-formatted: TVG. For practicality and representation, we select the most relevant tasks from each group—DVC, HD, and TVG—for evaluation.

(1) For dense video captioning, we use the YouCook2 dataset [25], which has 1,790 untrimmed videos of cooking procedures. On average, each video lasts 320s and is annotated with 7.7 temporally-localized imperative sentences. The dataset is split into 1,333 videos for training and 457 videos for validation. We evaluate caption quality using CIDEr [20]. For an overall evaluation at the story level, we use the SODA_c metric [2]. We also report the F1 score, which is the harmonic mean of the average precision and recall across IoU thresholds of 0.3, 0.5, 0.7, 0.9, to measure event localization performance.

(2) For video highlight detection, we use the QVHighlights dataset [9]. It consists of over 10,000 videos annotated with human-written text queries. The evaluation metrics are mAP (mean average precision) with IoU thresholds of 0.5 and 0.75, and HIT@1 (the hit ratio of the highest-scored clip).

(3) For temporal video grounding, we use the Charades-STA [3] dataset. The dataset contains 6,670 videos and involves 16124 queries, where 12,404 pairs are used for training and 3,720 for testing. The average duration of the videos is 30.59 seconds and each video contains 2.41 annotated moments, and the moment has an average duration of 8.09 seconds. The evaluation metric is "R@1, IoU = μ", which denotes the percentage of retrieved moments with an intersection over union (IoU) greater than μ compared to the ground truth, given language queries.

F. Details of Multi-model Pipelines

We take VideoChat-Text [10] and InstructBLIP [1]+ChatGPT [14] as the baselines of Multi-model Pipielines. These pipelines integrate specialized visual models with ChatGPT, which firstly convert video semantics into textual descriptions and then leverage ChatGPT to process all inputs to solve the target task.

VideoChat-Text utilizes `ffmpeg` to extract key frames from the video at FPS=1. Then it leverages visual tools to obtain rich video information including action labels, frame summaries, video tags, comprehensive descriptions, object positional coordinates, video narratives, timestamps, and segment-related details. The overall visual information will

window size	stride	window overlap	#video tokens	SODA_c	CIDEr	F1 Score
32	32	✗	96	2.9	9.6	19.0
32	16	✓	192	2.9	10.0	19.6
16	16	✗	192	**3.2**	**11.7**	**19.8**
16	8	✓	384	3.1	10.8	19.5
8	8	✗	384	3.1	11.2	19.7

Table 4. Sliding window hyper-parameters sweep on YouCook2.

You are an AI visual assistant, and you are seeing successive frames from the same video to tackle a task called Dense Video Captioning. The task goal is to locate a series of activity events and describe them with a sentence based on the video frames. I will give you descriptions of all extracted frames with their timestamps. You can get and understand the video context based on the given detailed visual descriptions. The task output should be in a tone that a visual AI assistant is seeing the video and is time-sensitive.

Guidelines:

- **In the context of dense video captioning, an "event" can be defined as a specific activity or series of related activities having similar semantics within the video. Dense video captioning would aim to integrate frames with similar semantics/actions to an event and provide the description. It is important to focus on the change of human actions, related objects and environment/background in the given video descriptions to perceive the temporal semantics and recognize the successive events.**
- **Note that the frame descriptions are from a visual captioning model and may have tiny errors like describing misidentified objects, you can modify and fix the misidentified content reasoning from the successive and global video semantics.**

Examples:

Task Input

- ***Frame at 5.7 second shows a man wearing a chef's hat is preparing food in a kitchen. He is standing in front of a stove, holding a knife and cutting ingredients for a sandwich. There are various kitchen appliances visible in the scene, including an oven, a microwave, and a refrigerator. The kitchen appears to be well-equipped for cooking and preparing meals.***
- ***Frame at 17.0 second shows ……***
- ***……***

Task Output

12.0 - 23.0 seconds, add carrots radishes sugar salt to a vinegar to a bowl. 26.0 - 34.0 seconds, mix fish sauce oil and soy sauce in a bowl. 35.0 - 41.0 seconds, pour the sauce over the bread. 43.0 - 47.0 seconds, spread mayonnaise on the bread. 49.0 - 72.0 seconds, place lettuce onions chicken jalapenos basil on top of the bread. 77.0 - 84.0 seconds, add vegetable mixture on top of the sandwich".

New Inputs:

Now I need your help to handle the following video:

- ***Frame at 1 second shows A woman in a kitchen waving at the camera with a smile. She is wearing an orange top and a black apron with the words "Titli' s Busy Kitchen" on it. Behind her, there are wooden cabinets, a tiled backsplash, and a microwave oven on the wall. On the counter, there are various cooking utensils, such as a knife, a cutting board, a bowl, and a spatula.***
  ……
- ***Frame at 171 second shows ……***

Based on the above time-sensitive frame descriptions, please solve the dense video captioning task and output a series of events following the above example format: "start - end seconds, occurred event description".

Figure 1. Examples of designed prompts for the InstructBLIP+ChatGPT pipeline. The input prompts encompass **(1)** the task definition, **(2)** specific guidelines, **(3)** an in-context example, and **(4)** the new instance input. The video information includes detailed frame descriptions from InstrucBLIP with the corresponding timestamp. We uniformly sample 50 frames for each video to obtain visual semantics.

be processed by the ChatGPT to respond to user instructions. We design task-related prompts to endow VideoChat-Text with the capability to solve timestamp-sensitive tasks.

InstructBLIP+ChatGPT endows a more powerful visual expert model, i.e. InstructBLIP, to describe each frame with exhaustive paragraphs containing detailed video semantics. We employ well-designed prompts (illustrated in Fig. 1) for ChatGPT to solve each task. For video input, we uniformly sample 50 frames to obtain frame descriptions.

G. Generalized to New Domains

In Fig. 2, we show qualitative results in new domains such as movie [22] and egocentric videos [4], 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.

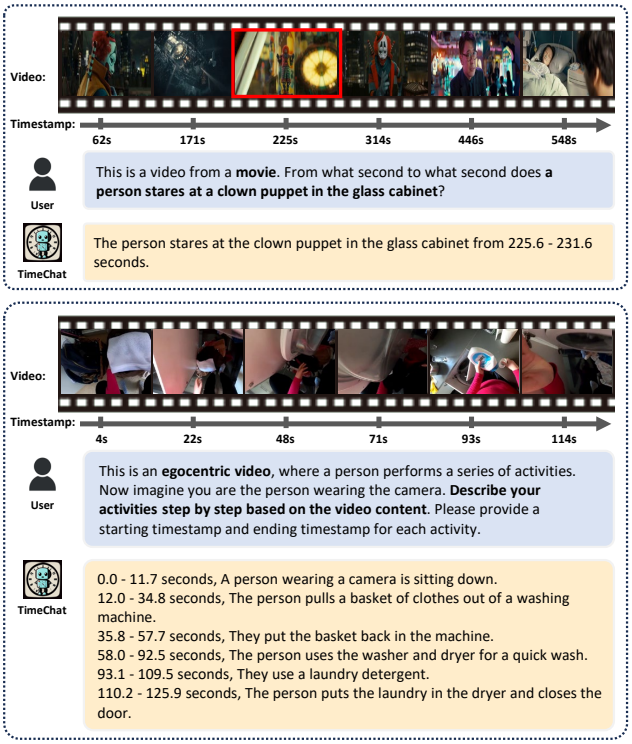

Figure 2. Zero-shot transfer to new domains such as movie (upper) and egocentric videos (bottom).

H. More Qualitative Results

Within Figures 3-5, we present an extended range of qualitative results, encompassing dense video captioning, temporal video grounding, and video highlight detection tasks. Overall, our model demonstrates proficiency in executing a diverse array of intricate temporal localization tasks.

References

[1] 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. 3

[2] Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. Soda: Story oriented dense video captioning evaluation framework. In *European Conference on Computer Vision*, 2020. 3

[3] 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. 1, 3

[4] 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. 4

[5] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision*, 2014. 1

[6] 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. 1

[7] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In *AACL*, 2020. 1

[8] 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. 1

[9] Jie Lei, Tamara L. Berg, and Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries. *ArXiv*, abs/2107.09609, 2021. 1, 3

[10] 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. 3

[11] 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. 1

[12] Jérôme Louradour. whisper-timestamped. https://github.com/linto-ai/whisper-timestamped, 2023. 1

[13] 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. 1

[14] OpenAI. Introducing chatgpt. 2022. 3

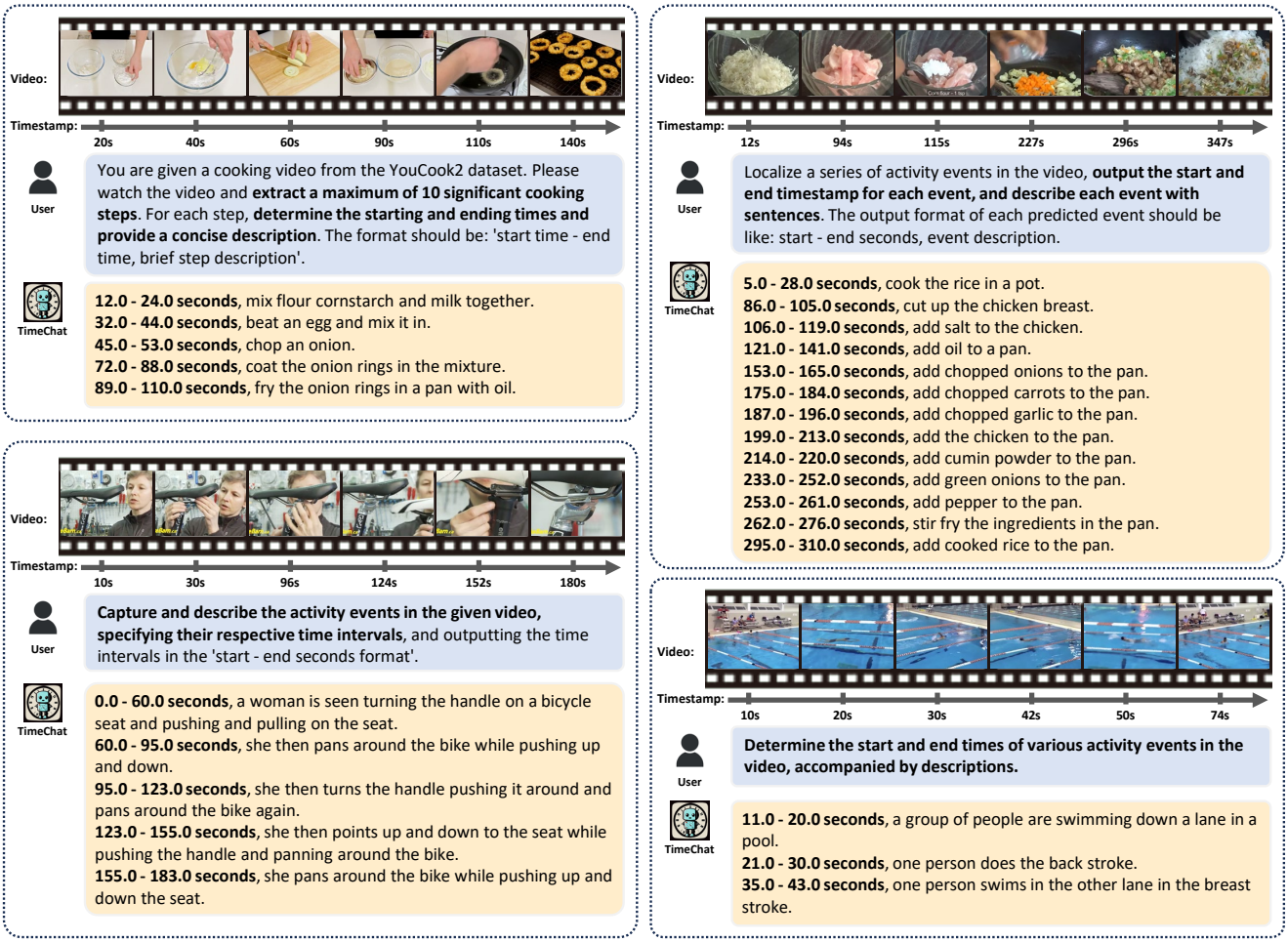

Figure 3. Qualitative results on video dense captioning task. For each video, we ask TimeChat to detect a series of events in the given video and output the corresponding timestamps and descriptions.

[15] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 1

[16] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022. 1

[17] 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

[18] 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. 1

[19] 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. 1

[20] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2014. 3

[21] 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. 1

[22] 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. 4

[23] 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. 1

[24] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yan-

Figure 4. Qualitative results for temporal video grounding task. For each video, we prompt our model to estimate the starting and ending timestamps for two specified queries (highlighted in bold). The predicted start and end timestamps and their corresponding segments are displayed in **green** and **red**, respectively.

peng 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. 1

[25] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2017. 1, 3

Figure 5. Qualitative results for video highlight detection task. In each video, we instruct TimeChat to pinpoint the most thrilling, remarkable, or emotive moment based on a specified query. The model is also required to assess the saliency score for each identified moment. We graph the saliency scores in relation to the moment's timestamp. The **orange** curve denotes the ground truth, while the **blue** curve signifies the predictions made by our model.