Act-ChatGPT: Introducing Action Features into Multi-Modal Large Language Models for Video Understanding

Yuto Nakamizo[0009−0002−8926−3183] and Keiji Yanai[0000−0002−0431−183X]

The University of Electro-Communications, Chohu, Tokyo, JAPAN
{nakamizo-y, yanai}@mm.inf.uec.ac.jp

Abstract. In the last few years, the advancement of GPT-4 and similar extensive large language models has significantly influenced video comprehension fields, models have been developed to exploit these advances to enhance interactive video comprehension. However, existing models generally encode video using image language models or video language models with sparse sampling, overlooking the vital action features present in each video segment. To address this gap, we propose Act-ChatGPT, an innovative interactive video comprehension model that integrates action features. Act-ChatGPT incorporates a dense sampling-based action recognition model as an additional visual encoder, enabling it to generate responses that consider the action in each video segment. Comparative analysis reveals Act-ChatGPT superiority over a base model, with qualitative evidence highlighting its adeptness at recognizing actions and responding based on them.

Keywords: Multi-Modal Large Language Model · Action Features · Video Understanding · Dual-Encoder strategy

1 INTRODUCTION

The evolution of Large Language Models (LLMs) in natural language processing has led to invention of multi-modal LLMs, combining a visual encoder with LLM for enhanced video understanding. This fusion projects visual features onto LLM token spaces, facilitating interactive comprehension. Nevertheless, such models typically use an image language model as a visual encoder or a video language model that is conscious of modeling the entire video, neglecting detailed actions within video segments. Conversely, with the adoption of Transformer [20] and self-supervised learning in video domain, especially models pre-trained on extensive video data, has significantly improved action recognition. These models have high action recognition performance, and in particular, by using models that operate on individual video segments, it is possible to extract good action features from each segment of the video.

Therefore, we propose Act-ChatGPT, an advanced multi-modal LLM tailored for video understanding, which emphasizes the utilization of action features within each video segment. Act-ChatGPT enhances video comprehension by incorporating an action recognition model as an additional visual encoder. This model, designed to extract action features from each video segment, works in tandem with Video-ChatGPT's existing image-based visual encoder. Moreover, Act-ChatGPT is different from traditional models by adopting a dual-encoder strategy. This approach combines the object recognition strengths of the visual language model with the nuanced human action detection of the action recognition model, enabling a richer video understanding. Our contributions are (1) We propose Act-ChatGPT, which is the first multi-modal LLM for video understanding that introduces action features within each video segment. (2) The experimental results showed the effectiveness of our proposed method by outperforming the baseline, Video-ChatGPT.

2 RELATED WORKS

2.1 LLMs

A language model that has been pre-trained by self-supervised learning with a large corpus is called a pre-trained language model. Recently, based on the knowledge that scaling the model parameters and training data of these pre-trained language models can improve the performance of downstream tasks [10], large pre-trained language models with a very large number of parameters and trained on particularly large amounts of data have been constructed. Because these models have an emergent abilities [26] that has not been seen in small-scale pre-trained language models, and because they show tremendous ability in solving a series of complex tasks, they are distinguished from small-scale pre-trained language models and are referred to as LLMs [24]. LLMs excel in their ability to generate language and make common sense inferences, and their use has been studied in many fields, not only in the field of natural language processing but also in other fields. For example, OpenAI's GPT-4, which has been reported to have particularly excellent instruction response performance, is used for dataset creation, filtering, and data augmentation, because it can be utilized via API. Since LLaMA [8] and its successor, Llama-2 [7], are the LLMs whose models and weights are publicly available, they have become the basis for many LLMs such as Vicuna [3].

Our study delves into utilizing LLMs within the visual domain, particularly focusing on enhancing video understanding through the integration of action features, marking a significant step forward in interactive video understanding.

2.2 Multi-Modal LLMs

Current multi-modal LLMs in the visual sphere fall into two primary categories. The first involves leveraging LLMs to interlink specialized models for diverse visual tasks, exemplified by Visual ChatGPT [2], a system that integrates numerous expert models through a LLM. This setup allows the LLM to process user commands and visual inputs, activating necessary external visual models to fulfill these commands.

The second category involves the methods that merge visual models with LLMs by mapping visual encoder-extracted features onto the LLM's token space, creating a unified model capable of end-to-end learning. BLIP-2 [12] is included in this category, that employs a "Q-former" module that aimed to bridge the gap between the visual encoder's features and the LLM's tokens through end-to-end training using image-text contrast learning, image-text matching and image grounded text generation. Additionally, this category includes LLaVA [5], which introduced Instruction Tuning [22] that is used in the field of natural language processing for visual contexts as Visual Instruction Tuning. This technique enhances instruction-following abilities by fine-tuning LLMs with data composed of instructional texts and their corresponding responses, where visual features are embedded into the instructional content.

In our study, we focus on the latter method and define the latter as Vision-LLM, and the Vision-LLM focusing on the video domain is defined as Video-LLM.

2.3 Video-LLMs

Current Video-LLMs fall into two main categories based on their approach to video encoding: frame-by-frame encoding using an image language model and holistic video encoding using a video language model.

The former-type models, such as VideoChat [13], Video-LLaMA [4], Video-ChatGPT [19], and LLaMA-VID [17], encode videos frame by frame. They employ the image language model, CLIP [1], as a visual encoder to extract features from individual frames sampled across the video. These features are often condensed and temporally modeled throughout the entire video using pooling and additional modules before being integrated into the LLM's token space via a linear layer.

Conversely, the latter-type models, such as VideoChat2 [14] and Video-LLaVA [18], encode videos as a whole. Some video language models such as UMT [16] and LanguageBind [25] capture video-wide features from a limited sampling of 4-16 frames for efficiency. These features are especially focusing on the video's overall context rather than the detailed temporal elements contained in each segment of the video.

Therefore, the existing Video-LLMs do not explicitly model the temporal features of the video or focus on modeling throughout the entire video and do not focus on the action in each segment of the video. Our study differs from the existing methods in that we introduce action features in each segment of the video to Video-LLM.

2.4 Action Recognition Model

The recent advancements in self-supervised learning have underscored its effectiveness, particularly with transformer-based models such as VideoMAEv2 [21] and UMT [16]. These models, pre-trained on extensive video datasets, have shown remarkable efficacy in action recognition tasks by fine-tuning.

Current action recognition models predominantly fall into two categories based on their frame sampling techniques. The first employs dense sampling, a method that extracts multiple video segments of a set frame length throughout the video, exemplified by VideoMAE v2. The second utilizes sparse sampling, a strategy that selects a fixed number of frames about 4 or 16 from the entire video, regardless of its length, as seen in models like UMT [16]. Dense sampling is suitable for capturing detailed features within individual video segments, while sparse sampling is suitable for providing a broader overview of features across the entire video. Those approach, therefore, offers different unique advantages for modeling action content, in that they either focus on specific segments or the video as a whole.

3 METHOD

3.1 Overview

We introduce a novel Video-LLM into Video-ChatGPT [19] by integrating action features. Fig. 1 provides an overview of our method.

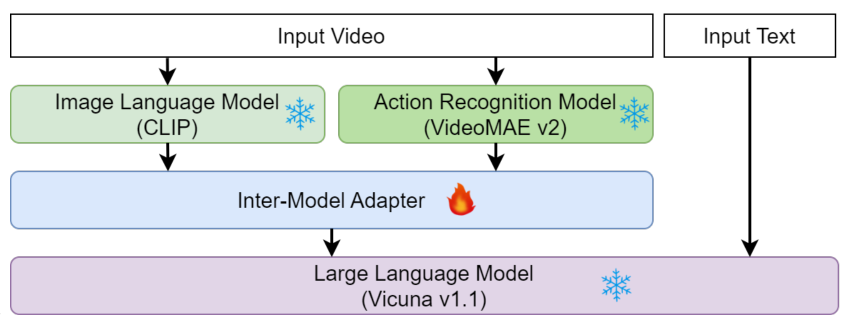

Fig. 1: The overview of Act-ChatGPT

We employ a dual-encoder strategy for the visual encoder, combined using an image language model for frame-based image feature extraction with an action recognition model dedicated to capturing action features from video segments. Initially, we sample T frames, $F \in \mathbb{R}^{T\times W\times H\times C}$, and T sets of 16-frame video segments, $S \in \mathbb{R}^{T\times 16\times W\times H\times C}$, from the input video. Then, from these samples image features, $V_f \in \mathbb{R}^{T\times N\times D_f}$, and action features, $V_s \in \mathbb{R}^{T\times D_s}$, are extracted via their respective encoders. Here, D_f and D_s represent the dimensional of the embedded features from the image language model and the action recognition model, respectively. N denotes the number of the image language model's patches, calculated as $N = W/p \times H/p$ based on the patch size p of the image language model where W, H, and C represent the width, the height, and the channel of the input video.

Subsequently, the extracted image and action features, V_f and V_s, are converted into visual tokens, $Q_v \in \mathbb{R}^{(2T+N)\times D_h}$. Here, D_h represents the dimension of the LLM's token space. This is achieved through an Inter-Model Adapter that projects each feature set into the LLM's token space and merges them. The specifics of this conversion process within the Inter-Model Adapter are detailed further in Section 3.3.

In the final step, the next tokens are predicted from a visual token, Q_v, and a linguistic token, Q_t, tokenized from the input text, and then a response text is generated by LLM. To optimize training efficiency, in our proposed method, we leverage pre-trained models for both two visual encoders and the LLM and train only the Inter-Model Adapters.

3.2 Using Trained Models

Our method incorporates several pre-trained models across a visual language model, an action recognition model, and LLM components. Initially, for the visual language model, we utilize the OpenAI CLIP [1] ViT-L/14 model. Here, the outputs from the penultimate layer are harnessed as the image features. Secondly, as an action recognition component, we employ the VideoMAEv2 [21] ViT-g/14 model, which has been fine-tuned on the Kinetics-710 dataset [15]. For this model, the action features are derived by applying Layer Normalization to the final layer's output and calculating the mean value. Lastly, for the LLM, we use Vicuna v1.1 [3], a 7B model fine-tuned for the multi-modal model LLaVA [5].

3.3 Inter-Model Adapter

Fig. 2 provides an overview of our method's Inter-Model Adapter. The Inter-Model Adapter is structured from three modules: the Image Feature Conversion Module, the Action Feature Conversion Module, and the Features Fusion Module. Below, we detail the components of each module and outline the processing procedure.

Image Feature Conversion Module The Inter-Model Adapter of Video-ChatGPT converting image features into tokens is used for this module. This process starts by applying both temporal and spatial mean pooling to the image features, $V_f \in \mathbb{R}^{T\times N\times D_f}$, extracted from each frame by the image language model. This process results in temporal features, $V_t \in \mathbb{R}^{T\times D_f}$, and spatial features, $V_n \in \mathbb{R}^{N\times D_f}$. Subsequently, these features are concatenated and then mapped to the LLM's token space through a single linear layer, f_f, resulting in the converted image feature tokens, $Q_f = f_f([V_t, V_n]) \in \mathbb{R}^{(T+N)\times D_h}$. Here, the notation $[a, b]$ signifies the concatenation of vectors a and b.

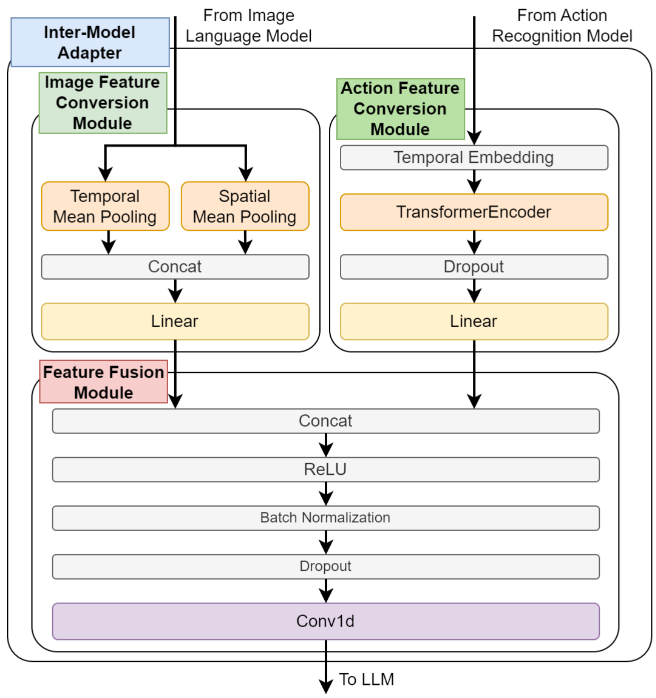

Fig. 2: The overview of Inter-Model Adapter.

Action Feature Conversion Module This module is designed to analyze the interplay among action features within video segments and to map these features into the LLM's token space effectively. The first function of this module is to capture global features that cannot be captured by segment-by-segment feature extraction by modeling the features in the temporal direction. To achieve this, it incorporates time embedding and a TransformerEncoder, with the TransformerEncoder set to a single layer featuring two heads mechanisms. Also, a single linear layer is utilized to map these analyzed features into the LLM's token space. During the conversion of action features into action feature tokens in this module, the process starts with adding temporal embedding to the action features extracted per video segment by the action recognition model through the TransformerEncoder. This step produces an enhanced set of action features, $V'_s = \text{TransformerEncoder}(V_s + TE \in \mathbb{R}^{T \times D_s})$, reflecting the temporal relationships between segments. Here, TE represents the temporal embedding that is the positional encoding in the temporal direction. Finally, a Dropout layer followed by a single linear layer f_s is applied, projecting the refined action features V'_s into the LLM's token space, resulting in converted action feature tokens $Q_s = f_s(\text{Dropout}(V'_s)) \in \mathbb{R}^{T \times D_h}$.

Features fusion module To merge the two distinct sets of features effectively, this module utilizes a one-dimensional convolution with a kernel size of one. The process starts by concatenating the image feature tokens, Q_f, and the action feature tokens, Q_s, from the feature conversion modules. This concatenated set then is processed by sequentially adapting ReLU, Batch Normalization, Dropout, and finally, the 1D convolution, resulting in the combined visual token, $Q_v = (\text{Conv1d}(\text{Dropout}(\text{BN}(\text{ReLU}([Q_f, Q_s])))) \in \mathbb{R}^{(2T+N)\times D_h}$, merged visual information of image and action information tailored for the LLM.

3.4 Data Augmentation

To address the challenge of insufficient training data our proposed method incorporates data augmentation techniques applied to the Video Instruction Dataset utilized for training. This augmentation process involves rephrasing existing instruction response texts, executed with the aid of Vicuna v1.5 [3] 13B. Specifically, paraphrases of the instructions are generated by instructing Vicuna to use synonyms and thesauruses extensively, avoid incorporating external information, and ensure the paraphrased instructions remain faithful to the original instruction-response relationship. This preserves the relationship between the instructions provided and the response, and extends the dataset without significantly deteriorating data quality.

3.5 Training

Our training approach follows Vision Instruction Tuning, utilizing a dataset comprised of video and corresponding instruction response text pairs, similar to Video-ChatGPT. The training objective is to minimize the token-by-token cross-entropy error between the actual responses and the model's predictions.

The training process is divided into two distinct stages. In the first stage, only one visual encoder is active, and the feature conversion module corresponding is trained independently. The model structure at this stage of training is shown in Fig. 3a and Fig. 3b. This stage's model architecture, when training the Image Feature Conversion Module, is similar to Video-ChatGPT [19], with the Image Feature Conversion Module being initialized using the inter-model adapter of Video-ChatGPT. The weights of the model-to-model adapter of Video-ChatGPT are equivalent to the weights of the Image Feature Conversion Module of the proposed method initialized with LLaVA [5] and then trained with the architecture shown in Fig. 3a using the non-augmented Video Instruction Dataset. Subsequently, in the second stage, both feature conversion modules are initialized with the weights trained in the first stage, and the entire Inter-Model Adapter, including the features fusion module, then are trained.

3.6 Prompts

The prompts for the LLM are crafted following the format established by Video-ChatGPT, structured as follows:

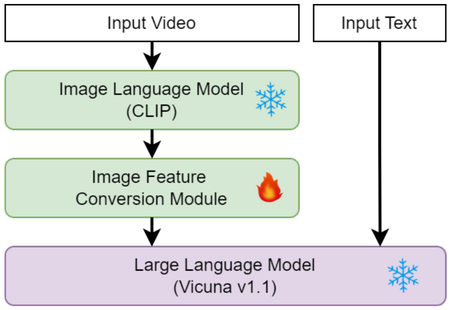

Fig. 3a: The model structure when training only Image Feature Conversion Module.

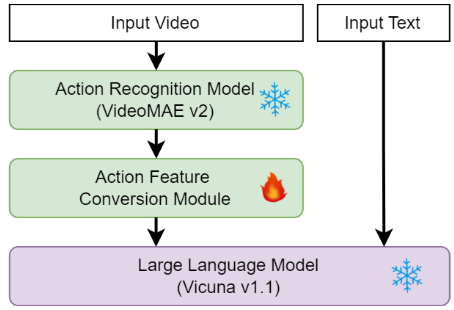

Fig. 3b: The model structure when training only Action Feature Conversion Module.

$$\text{USER: } \langle\text{Instruction}\rangle\ \langle\text{Video-token}\rangle\ \text{ASSISTANT:}$$

Here, ⟨Instruction⟩ denotes the instructions to the LLM, such as queries about the video, while ⟨Video-token⟩ symbolizes the visual features converted to tokens. The designations, "USER: and ASSISTANT:", distinguish between user instructions and LLM responses, facilitating the LLM's comprehension of dialogue progression, particularly in extended conversations. In our method, ⟨Instruction⟩ within the template is replaced by the actual instruction text and tokenized. Subsequently, the token for ⟨Video-token⟩ is substituted with the visual token Q_v, obtained by the Inter-Model Adapter, before being fed into the LLM.

4 EXPERIMENTS

4.1 Experimental Settings

In our experiments, we follow the sampling parameters of Video-ChatGPT [19], setting the number of frames and video segments, T, to 100. The Dropout layer's probability parameter, p, was adjusted to 0.0 during the first training stage and increased to 0.5 in the second stage. Additionally, the temperature parameter, τ, pivotal in controlling the probability distribution of LLM's token generation during inference and thus influencing the model's creativity, was fixed at 0.2, except where specified otherwise.

The training for both stages utilizes the same dataset and settings, employing the Video Instruction Dataset [19] derived from a subset of the ActivityNet dataset [6]. This dataset contains around 100,000 video pairs coupled with single-turn instruction-response texts. It is created by making instruction-response texts pertinent to video content using GPT-3.5 from human-crafted captions being included in a subset of ActivityNet dataset and frame-level captions from BLIP-2 [12]. As mentioned above, to address the scarcity of training data, our approach includes a data augmentation strategy, rephrasing instructions via a LLM, unlike Video-ChatGPT. Optimization is conducted using AdamW, with

Table 1: The number of questions in each category

Question Set	Action	Object	Total
GENERIC	1466	530	1996
TEMPORAL	481	18	499
CONSISTENCY	231	268	499

a learning rate schedule using linear warmup with a warmup rate of 0.03 and cosine decay with a peak at 2×10^{-5}. Each training stage is trained for three epochs, following the training of the inter-model adapter in Video-ChatGPT.

The quantitative evaluation is carried out by Video-based Generative Performance Benchmarking [19] and AutoEval-Video [23]. For the Video-based Generative Performance Benchmarking, a test set based on a subset of ActivityNet dataset [6] as well as the Video Instruction Dataset is used. The evaluation of each response is performed by GPT-3.5 (the checkpoints used is gpt-3.5-turbo-0125) to score a relative score on 0 to 5, based on comparison with the correct answers, in terms of perspective assigned to each data from five perspectives that are Correctness of Information (CI), Detail Orientation (DO), Contextual Understanding (CU), Temporal Understanding (TU) and Consistency (C). In the following, all questions are evaluated three times, and the means and standard deviations are reported for each item, except where specified otherwise. In addition, the evaluation questions in the Generative Performance Benchmarking dataset is divided into two types of questions using GPT-4o (the checkpoints used are gpt-4o-2024-05-13): action-oriented questions and object-oriented questions. Action-oriented questions mean the questions on dynamics in the videos where action features are expected to help to answer, while object-oriented questions mean the questions on objects and scenes for which image features are expected to be helpful. The number of both types are shown in Table 1. Note that, GENERIC is a split of the dataset used evaluating Correctness of Information, Detail Orientation, and Contextual Understanding. TEMPORAL is a split of the dataset used evaluating Temporal understanding, and CONSISTENCY is a split of the dataset used evaluating Consistency. The evaluation results for each of the two types of questions are reported as well.

For the AutoEval-Video, a uniquely collected and annotated dataset for the benchmark from YouTube across multiple capability domains and topics is used. The evaluation of each response was performed by GPT-4 (the checkpoints used are gpt-4-1106-preview) to judge right and wrong based on the specific evaluation rules defined for each sample, in terms of the perspective assigned to each sample from nine perspectives: Dynamic Perception, State Transition Perception, Comparison Reasoning, Reasoning with External Knowledge, Explanatory Reasoning, Predictive Reasoning, Description, Counterfactual Reasoning and Camera Movement Perception. In the following, the means of the accuracy of overall and each item of three times evaluations conducted are reported.

In our study, emphasis was placed on the results of Video-based Generative Performance Benchmarking, as this is the most commonly used method in exist-

Table 2: Results of Video-based Generative Performance Benchmarking.

	CI↑	DO↑	CU↑	TU↑	C↑
Video-LLaMA	2.23 ± 1.25	2.16 ± 0.79	2.52 ± 1.13	1.93 ± 1.09	2.02 ± 1.09
Video-ChatGPT	2.50 ± 1.33	2.31 ± 0.85	2.87 ± 1.18	2.10 ± 1.15	2.20 ± 1.24
Video-ChatGPT (scratch)	2.44 ± 1.31	2.29 ± 0.83	2.82 ± 1.17	2.10 ± 1.11	2.06 ± 1.19
Act-ChatGPT (scratch)	2.53 ± 1.34	2.33 ± 0.82	2.89 ± 1.21	2.19 ± 1.15	2.17 ± 1.23
Act-ChatGPT (w/o data aug.)	2.53 ± 1.36	2.33 ± 0.86	2.92 ± 1.19	2.13 ± 1.14	2.17 ± 1.23
Act-ChatGPT	$\mathbf{2.62 \pm 1.35}$	$\mathbf{2.37 \pm 0.85}$	$\mathbf{3.00 \pm 1.17}$	$\mathbf{2.20 \pm 1.14}$	$\mathbf{2.28 \pm 1.25}$
GPT-4o	4.02 ± 1.13	3.46 ± 0.92	4.19 ± 0.92	3.30 ± 1.30	3.54 ± 1.23

ing Video-LLM assessments. The results of AutoEval-Video, on the other hand, were used to check the generalisation performance of the model, as they were based on dataset collected and annotated in a completely different way to the training data.

4.2 Comparison with Baseline

A quantitative comparative analysis by Video-based Generative Performance Benchmarking between our proposed method and Video-ChatGPT [19], Video-LLaMA [4] is shown in Table 2. Table 2 also includes the results of the evaluation of responses by GPT-4o for reference. GPT-4o generated the responses based on the following instruction, using 20 frames sampled from the video: "These images are frames cut from a single video. Referring to these images, answer the following questions. However, the actual answers do not require frame-by-frame explanations, please generate the actual answer to the aggregated video." Note that the evaluation of GPT-4o was conducted only once. To make fair comparison, we show the results excluding data augmentation (denoted as w/o data aug.) and the results training inter-model adapters with only augmented Video Instruction Dataset without pre-training with such as LLaVA[5] dataset (denoted as scratch). Also, the results for action-oriented questions and object-oriented questions are shown in Table 3 and Table 4.

Our method superior performance across all metrics when compared to existing models. Also,within the same metrics, there is no significant difference in standard deviations between different methods. Notably, even in the absence of data augmentation, our approach surpassed Video-ChatGPT in all but Consistency. This underscored the significant impact of integrating action features on enhancing theresponse performance of Video-LLM responses. Note that our method is still clearly inferior to the response by GPT-4o, indicating room for further development of the open source Video-LLM.

On the other hand, Table 3 and Table 4 show that Act-ChatGPT outperforms Video-ChatGPT, especially for action-oriented questions, while conversely the performance of Act-ChatGPT is slightly less than the baseline in the evaluations for only object-oriented questions. Therefore, our proposed method can be regarded as focusing on action-oriented questions more.

Table 3: Results of Video-based Generative Performance Benchmarking for the action-oriented questions.

	CI↑	DO↑	CU↑	TU↑	C↑
Video-LLaMA	2.16 ± 1.11	2.08 ± 0.68	2.41 ± 1.04	1.90 ± 1.05	2.13 ± 1.14
Video-ChatGPT	2.51 ± 1.23	2.25 ± 0.78	2.85 ± 1.13	2.09 ± 1.12	2.49 ± 1.24
Video-ChatGPT (scratch)	2.50 ± 1.22	2.28 ± 0.78	2.86 ± 1.13	2.10 ± 1.07	2.43 ± 1.21
Act-ChatGPT (scratch)	2.65 ± 1.23	2.35 ± 0.76	3.00 ± 1.15	$\mathbf{2.20 \pm 1.14}$	2.60 ± 1.24
Act-ChatGPT (w/o data aug.)	2.62 ± 1.25	2.32 ± 0.80	2.97 ± 1.14	2.13 ± 1.12	2.54 ± 1.24
Act-ChatGPT	$\mathbf{2.72 \pm 1.24}$	$\mathbf{2.36 \pm 0.78}$	$\mathbf{3.08 \pm 1.11}$	2.19 ± 1.10	$\mathbf{2.72 \pm 1.23}$

Table 4: Results of Video-based Generative Performance Benchmarking for the object-oriented questions.

	CI↑	DO↑	CU↑	TU↑	C↑
Video-LLaMA	2.43 ± 1.56	2.39 ± 1.00	2.81 ± 1.29	$\mathbf{2.61 \pm 1.76}$	1.92 ± 1.03
Video-ChatGPT	$\mathbf{2.49 \pm 1.58}$	$\mathbf{2.48 \pm 1.01}$	$\mathbf{2.90 \pm 1.29}$	2.26 ± 1.89	$\mathbf{1.94 \pm 1.19}$
Video-ChatGPT (scratch)	2.27 ± 1.51	2.32 ± 0.95	2.73 ± 1.26	2.15 ± 1.84	1.75 ± 1.09
Act-ChatGPT (scratch)	2.20 ± 1.56	2.27 ± 0.96	2.60 ± 1.32	1.83 ± 1.53	1.80 ± 1.10
Act-ChatGPT (w/o data aug.)	2.28 ± 1.59	2.36 ± 1.00	2.77 ± 1.30	2.33 ± 1.63	1.84 ± 1.13
Act-ChatGPT	2.33 ± 1.58	2.37 ± 1.02	2.79 ± 1.28	2.35 ± 1.89	1.89 ± 1.14

In addition, a quantitative comparative analysis by AutoEval-Video between our proposed method and Video-ChatGPT, Video-LLaMA, is detailed in Table 5 and Table 6. As with the Video-based benchmark, we show the results excluding data augmentation (denoted as w/o data aug.) and the results training inter-model adapters with only augmented Video Instruction Dataset (denoted as scratch). In this evaluation, by contrast, our method underperformed the base model on almost all items. Thus, it can be said that our proposed method has poorer generalization performance than the Video-ChatGPT. The poor performance of Act-ChatGPT for AutoEval-Video mainly comes from the differences with and without pre-training of inter-model adapters.

The inter-model adapter of Video-ChatGPT is pre-trained with 753k LLaVA [5] training images and fine-tuned with non-augmented 100k Video Instruction Dataset, whereas the one for Act-ChatGPT is trained with only augmented 200k Video Instruction Dataset from scratch, except for the Image Feature Conversion Module, which is initialized with the weights of the inter-model adapter of Video-ChatGPT. This means that the Image Feature Conversion Module in the Act-ChatGPT was pre-trained with 753k LLaVa training images as well, whereas the action feature conversion module was not pre-trained with any dataset. This is due to the fact that the proposed method uses a segment-based action recognition model for one of the visual encoders in which not image data but video data is used for training. In fact, when comparing Video-ChatGPT (scratch) and Act-ChatGPT (scratch) from Table 5, which were trained inter-model adapters only on the augmented Video Instruction Dataset without pre-training with image data, Act-ChatGPT outperforms Video-ChatGPT on overall accuracy. This shows that the performance deterioration in the evaluation with AutoEval-Video

Table 5: Results of AutoEval-Video (overall)

	All↑
Video-LLaMA	0.070
Video-ChatGPT	**0.101**
Video-ChatGPT (scratch)	0.045
Act-ChatGPT (scratch)	0.049
Act-ChatGPT (w/o data aug.)	0.064
Act-ChatGPT	0.064

Table 6: Results of AutoEval-Video (each item)

	Dynamic ↑	State Transitions ↑	Comparison ↑
Video-LLaMA	0.059	0.073	0.140
Video-ChatGPT	**0.088**	**0.115**	**0.246**
Video-ChatGPT (scratch)	0.044	0.094	0.176
Act-ChatGPT (scratch)	0.050	0.041	0.123
Act-ChatGPT (w/o data aug.)	0.036	0.083	0.123
Act-ChatGPT	0.029	0.073	0.193
	External Knowledge ↑	Explanatory ↑	Predictive ↑
Video-LLaMA	**0.084**	0.040	0.041
Video-ChatGPT	**0.084**	**0.086**	**0.135**
Video-ChatGPT (scratch)	0.016	0.035	0.031
Act-ChatGPT (scratch)	0.042	0.045	0.000
Act-ChatGPT (w/o data aug.)	0.062	0.066	0.062
Act-ChatGPT	0.050	0.066	0.052
	Description ↑	Counterfactual ↑	Camera Movement ↑
Video-LLaMA	0.056	0.140	0.000
Video-ChatGPT	0.044	0.123	**0.111**
Video-ChatGPT (scratch)	0.022	0.035	**0.111**
Act-ChatGPT (scratch)	0.011	**0.176**	0.000
Act-ChatGPT (w/o data aug.)	**0.067**	0.053	0.000
Act-ChatGPT	0.044	0.123	0.000

observed in Act-ChatGPT is due to the lack of pre-training data, not to the introduction of the proposed action features. This does not negate the effectiveness of the proposed method.

Fig. 4 shows qualitative comparisons between Act-ChatGPT and Video-ChatGPT. The observations from the top and middle response results in Fig. 4 illustrate that Act-ChatGPT enhances responses over Video-ChatGPT by improving action recognition as well as the identification of objects involved in these actions. It is conceivable that this improvement in object recognition is due to the fact that the large language model recognizes action features in a different way to spatial features, allowing the consistency of object and action elements as sentences to be taken into account when generating responses. Furthermore, the bottom response result in Fig. 4 demonstrates that Act-ChatGPT retains the capability to recognize unique objects, as observed in Video-ChatGPT.

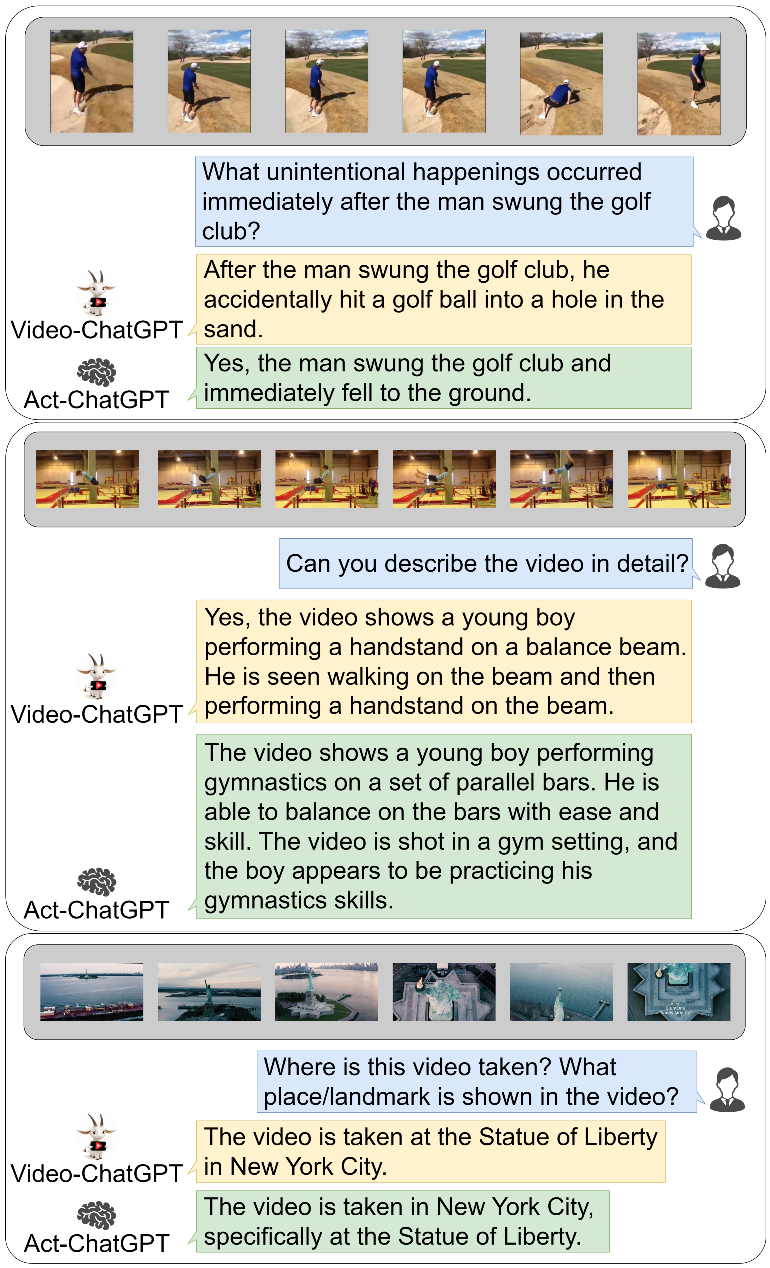

Fig. 4: Examples of responses.

4.3 Ablation Studies

In the ablation studies, Video-based Generative Performance Benchmarking is used. Table 7 displays the quantitative results for Act-ChatGPT under various settings. Specifically, (w/o Stage1) denotes results from training solely in the second stage, (w/o Fusion) refers to scenario not using a features fusion module, while (w/o Image) and (w/o Action) refer to scenarios where only the action recognition model or the image language model is employed as a vision encoder, respectively. Note that the evaluation was conducted only once in each setting and the results are reported. As a side note, even when only one visual encoder is used, the feature fusion module was applied by adjusting the number of dimensions. The findings reveal a notable decline in performance metrics when Act-ChatGPT is trained solely during the second stage, underscoring the critical role of multi-stage learning. Moreover, utilizing only one type of visual encoder, whether for actions or images, leads to significant drops in all metrics. These outcomes suggest that image and action features play complementary roles in video understanding and emphasize the benefits of action features utilized in video understanding. In addition, focusing on feature fusion, the performance significantly deteriorated is found when features were not fused. This shows that in this research, where a LLM is frozen, a mechanism for explicitly fusing features is important for improving the performance.

Table 7: Results of Video-based Generative Performance Benchmarking under various settings.

	CI↑	DO↑	CU↑	TU↑	C↑
Act-ChatGPT (w/o Stage1)	2.28 ± 1.32	2.20 ± 0.89	2.66 ± 1.24	2.00 ± 1.16	2.01 ± 1.46
Act-ChatGPT (w/o Image)	2.17 ± 1.34	2.03 ± 0.85	2.46 ± 1.22	1.86 ± 1.07	1.99 ± 1.12
Act-ChatGPT (w/o Action)	2.41 ± 1.31	2.21 ± 0.82	2.74 ± 1.20	2.19 ± 1.16	1.97 ± 1.23
Act-ChatGPT (w/o Fusion)	2.39 ± 1.32	2.23 ± 0.89	2.74 ± 1.24	2.12 ± 1.16	2.26 ± 1.21
Act-ChatGPT (w/ all)	$\mathbf{2.62 \pm 1.35}$	$\mathbf{2.37 \pm 0.85}$	$\mathbf{3.00 \pm 1.17}$	$\mathbf{2.20 \pm 1.14}$	$\mathbf{2.28 \pm 1.25}$

5 LIMITATIONS

In our study, a new Act-ChatGPT with a newly introduced action feature was proposed. Several limitations still remain. The first major limitation relates to training data. In the recent trends in Video-LLMs, as Peng Jin *et al.* [9] have shown the advantages of joint learning of images and videos, it has become mainstream to learn various visual representations by also utilizing a large amount of image data in addition to video data. However, in our work, we used an action recognition model that operated on a video segment basis as part of the visual encoder. This design choice made it difficult to utilize image data for training. Therefore, to keep up with these trends and achieve better performance, it is necessary to create extensive video datasets to compensate for the lack of data or develop methods to utilize images for training.

The second limitation is in the computational cost. The computational cost of our proposed method is relatively high because it employs a large action recognition model. Additionally, our dual-encoder approach, which processes a certain amount of object features in the action branch, further contributes to these costs. The action recognition model used in our proposed method, trained on the Kinetics dataset [11] that are considered relatively easy to classify even with only scene information, includes somewhat object recognition capabilities. However, these capabilities are sometimes redundant in our method since the image branch already handles object recognition. This redundancy suggests a need for the more focused and compact model that extracts only movement features, which could help in reducing computational expenses.

6 CONCLUSIONS

In this paper, we proposed Act-ChatGPT, a Video-LLM designed to use action features from individual video segments to enrich response generation with insight into the action depicted. Act-ChatGPT enhanced both action and their associated object recognition capabilities, outperforming the Video-ChatGPT used as a base model. In addition, it also retains a certain level of object recognition capabilities, such as identifying unique objects in the video, demonstrating the improvement in video understanding over the base approaches overall.
Acknowledgments: This work was supported by JSPS KAKENHI Grant Numbers, 22H00540 and 22H00548.

References

1. Alec, R., Jong, Wook, K., et al.: Learning transferable visual models from natural language supervision. In: ICML. vol. 139, pp. 8748–8763 (2021)
2. Chenfei, W., Shengming, Y., Weizhen, Q., Xiaodong, W., Zecheng, T., Nan, D.: Visual ChatGPT: Talking, drawing and editing with visual foundation models. In: arXiv:2303.04671 (2023)
3. Chiang, W.L., Li, Z., et al.: Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality (2023), https://lmsys.org/blog/2023-03-30-vicuna/
4. Hang, Z., Xin, L., Lidong, B.: Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In: arXiv:2306.02858 (2023)
5. Haotian, L., Chunyuan, L., Qingyang, W., Yong, Jae, L.: Visual instruction tuning. In: NeurIPS (2023)
6. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: ActivityNet: A large-scale video benchmark for human activity understanding. In: CVPR. pp. 961–970 (2015)
7. Hugo, T., Louis, M., et al.: Llama 2: Open foundation and fine-tuned chat models. In: arXiv:2307.09288 (2023)
8. Hugo, T., Thibaut, L., Gautier, I., Xavier, M., Marie-Anne, L., Timothée, L., Baptiste, R., Naman, G., Eric, H., Faisal, A., Aurelien, R., Armand, J., Edouard, G., Guillaume, L.: LLaMA: Open and efficient foundation language models. In: arXiv:2302.13971 (2023)

9. Jin, P., Takanobu, R., Zhang, C., Cao, X., Yuan, L.: Chat-UniVi: Unified visual representation empowers large language models with image and video understanding. In: arXiv:2311.08046 (2023)
10. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. In: arXiv:2001.08361 (2020)
11. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. In: arXiv:1705.06950 (2017)
12. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML. pp. 19730–19742 (2023)
13. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: VideoChat: Chat-centric video understanding. In: arXiv:2305.06355 (2023)
14. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., Wang, L., Qiao, Y.: MVBench: A comprehensive multi-modal video understanding benchmark. In: arXiv:2311.17005 (2023)
15. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Wang, L., Qiao, Y.: UniFormerV2: Unlocking the potential of image vits for video understanding. In: ICCV. pp. 1632–1643 (2023)
16. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked Teacher: Towards training-efficient video foundation models. In: ICCV. pp. 19948–19960 (2023)
17. Li, Y., Wang, C., Jia, J.: LLaMA-VID: An image is worth 2 tokens in large language models. In: arXiv:2311.17043 (2023)
18. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-LLaVA: Learning united visual representation by alignment before projection. In: arXiv:2311.10122 (2023)
19. Muhammad, M., Hanoona, R., Salman, K., Fahad, Shahbaz, K.: Video-ChatGPT: Towards detailed video understanding via large vision and language models. In: arXiv:2306.05424 (2023)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. vol. 30 (2017)
21. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: VideoMAE V2: Scaling video masked autoencoders with dual masking. In: CVPR. pp. 14549–14560 (2023)
22. Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. In: ICLR (2022)
23. Xiuyuan, C., Yuan, L., Yuchen, Z., Weiran, H.: AutoEval-Video: An automatic benchmark for assessing large vision language models in open-ended video question answering. In: arXiv:2311.14906 (2023)
24. Zhao, W.X., Zhou, K., et al.: A survey of large language models. In: arXiv:2303.18223 (2023)
25. Zhu, B., Lin, B., Ning, M., Yan, Y., Cui, J., HongFa, W., Pang, Y., Jiang, W., Zhang, J., Li, Z., Zhang, C.W., Li, Z., Liu, W., Yuan, L.: LanguageBind: Extending video-language pretraining to n-modality by language-based semantic alignment. In: arXiv:2310.01852 (2023)
26. Zoph, B., Raffel, C., Schuurmans, D., Yogatama, D., Zhou, D., Metzler, D., Chi, E.H., Wei, J., Dean, J., Fedus, L.B., Bosma, M.P., Vinyals, O., Liang, P., Borgeaud, S., Hashimoto, T.B., Tay, Y.: Emergent abilities of large language models. In: Proc. of Transactions on Machine Learning Research (2022)