



# Artemis: Towards Referential Understanding in Complex Videos

Jihao Qiu<sup>1\*</sup> Yuan Zhang<sup>1\*</sup> Xi Tang<sup>1\*</sup> Lingxi Xie Tianren Ma<sup>1</sup>

Pengyu Yan<sup>2</sup> David Doermann<sup>2</sup> Qixiang Ye<sup>1</sup> Yunjie Tian<sup>1,2†</sup>

<sup>1</sup>University of Chinese Academy of Sciences <sup>2</sup>University at Buffalo

{qiujihao19, zhangyuan192, tangxi19, matianren18, tianyunjie19}@mailsucas.ac.cn  
pyan4@buffalo.edu doermann@buffalo.edu 198808xc@gmail.com qxye@ucas.ac.cn

## Abstract

Videos carry rich visual information including object description, action, interaction, *etc.*, but the existing multimodal large language models (MLLMs) fell short in referential understanding scenarios such as video-based referring. In this paper, we present **Artemis**, an MLLM that pushes video-based referential understanding to a finer level. Given a video, Artemis receives a natural-language question with a bounding box in any video frame and describes the referred target in the entire video. The key to achieving this goal lies in extracting compact, target-specific video features, where we set a solid baseline by tracking and selecting spatiotemporal features from the video. We train Artemis on the newly established VideoRef45K dataset with 45K video-QA pairs and design a computationally efficient, three-stage training procedure. Results are promising both quantitatively and qualitatively. Additionally, we show that Artemis can be integrated with video grounding and text summarization tools to understand more complex scenarios. Code and data are available at <https://github.com/qiujihao19/Artemis>.

## 1 Introduction

The past year has witnessed rapid progress of multimodal large language models (MLLMs) [33, 68, 40], offering abundant abilities of open-world image understanding with language-based dialogues. In comparison, there are fewer studies on training MLLMs for video understanding, albeit videos are much more informative than still images. Existing video-based MLLMs [29, 61, 37, 31, 38] mostly focus on superficial dialogues in which the video is encoded holistically, inevitably lacking the ability to understand fine-level video contents, *e.g.*, describing a user-specific target in the video.

We are considering a new task called video-based referential understanding to compensate for the limitation. Specifically, we are interested in complex videos that span 20–30 seconds and the target performs multiple actions during this period. Given a video, the MLLM tries to answer a question like ‘What is the target <region> doing in this video?’ where <region> refers to a bounding box in any video frame. We argue that the task is not only challenging as it requires feature extraction, tracking, summarization, *etc.*, but also important because it lays the foundation of finer-level video understanding. However, as shown in Figure 1, existing MLLMs often fell short in

\* Equal contribution. † Corresponding Author.

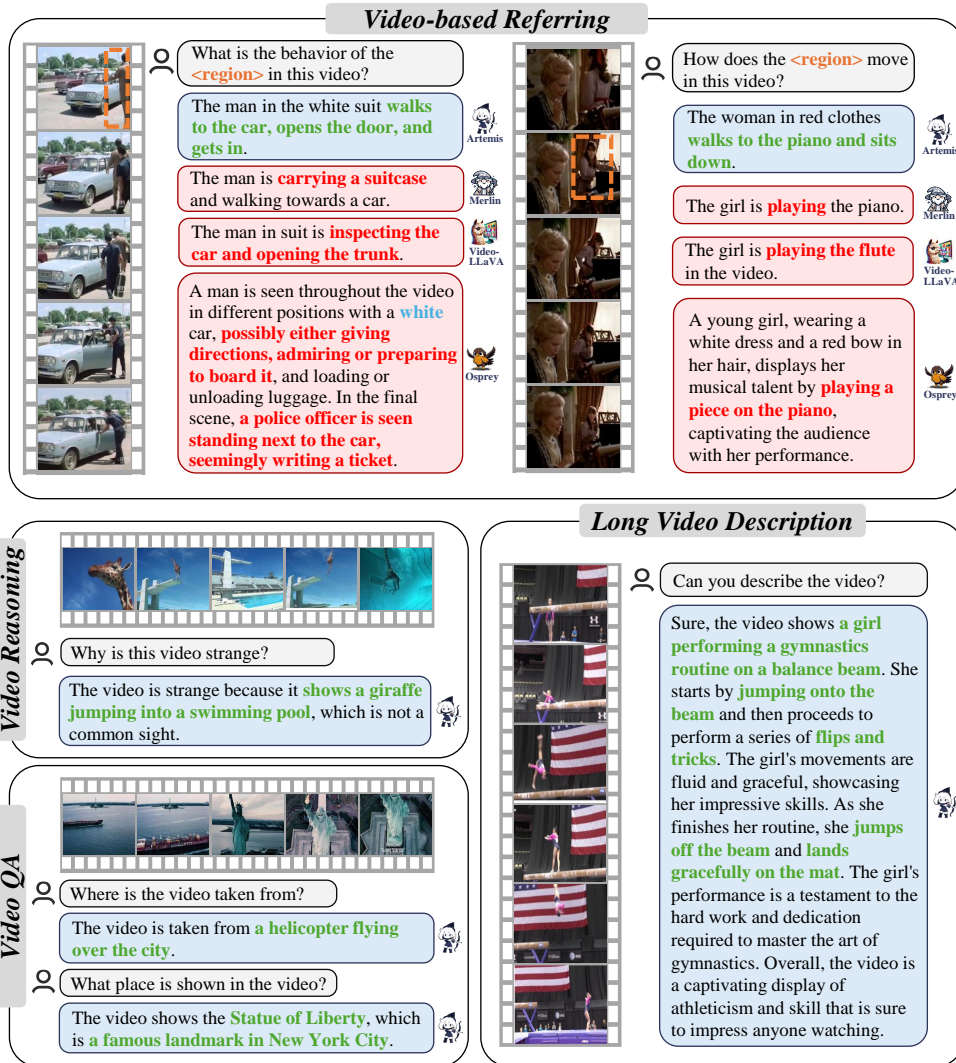


Figure 1: Artemis’ ability in video-based dialogue. Notably, Artemis excels particularly in video-based referring, outperforming the existing MLLMs including Merlin [57] and Video-LLaVA [31] lacking comprehensiveness and Osprey [59] suffering hallucination.

this seemingly easy task because they were mostly trained for image-based referential understanding; as a result, they can only perceive the action in a single moment rather than that in an entire video<sup>2</sup>.

This paper presents **Artemis**<sup>3</sup> as a solid baseline for the above task. Artemis follows the generic design of modern MLLMs (*i.e.*, visual instruction tuning [33]), but encounters a challenge in finding *sparse*, target-related information from *dense* video data. A preliminary study shows that feeding raw video features into the MLLM results in computational inefficiency and training instability. To extract target-specific video features, we propose a simple yet effective solution that involves (i) tracking the target over time and (ii) selecting informative features from a long list of regions-of-interest (RoIs). The compactness of features makes it easier to train the MLLM. We design a three-stage training schedule where the MLLM gradually learns video-text alignment from coarse to fine. This efficient design requires only 28 hours (3 hours for the final stage) on 8× NVIDIA-A800 GPUs.

To train and evaluate Artemis, we organize 7 existing video understanding datasets into the **Video-Ref45K** benchmark comprising 45K video question-answer pairs. To our knowledge, this is the first

<sup>2</sup>One can also use an image-based model for frame-wise referring and call an LLM for text summarization, but, as shown in Appendix C, this method often fails to understand the intrinsic logic in complex videos.

<sup>3</sup>The name refers to Artemis’ ability to track prey and select pivotal hunting moments.

benchmark with box-level prompts and answers spanning complex videos. Experiments show the promising results of Artemis in a wide range of quantitative metrics including the BERT score, BLEU, *etc.*. Qualitatively, Artemis also shows a clear advantage in the comprehensiveness of description meanwhile avoiding hallucination (see Figure 1 for examples). Beyond the ability of video-based referring, Artemis serves as an important building block for complex video understanding, where we integrate Artemis with off-the-shelf video grounding and text summarization tools for interactive video-based dialogue and long video understanding, respectively. We expect our work to shed light on upgrading MLLMs for fine-level and interactive video understanding.

## 2 Related Work

**Large language models (LLMs) and multimodal LLMs (MLLMs).** LLMs [15, 5, 13, 45, 12, 64, 49, 60, 11] have opened a new era of AI, demonstrating the potential to deal with various language-based understanding and generation tasks. To unleash the power of LLMs for visual understanding, the computer vision community has been working on aligning language and vision data in the same feature space [41]. There are mainly two lines of research, where the *internal* adaptation methods [1] integrated cross-attention within an LLM for visual-language alignment, and the *external* adaptation methods [27, 14, 33] trained extra modules for this purpose. As a result, the vision foundation models, especially vision transformers [17, 35, 41, 48, 47, 66, 25], have been upgraded into MLLMs [33, 46, 63, 26] which gain the ability of language-guided visual understanding.

**MLLMs for referring and grounding.** MLLMs can be integrated with instance-level visual understanding tasks, allowing the models to (i) respond to questions targeted at specific regions of the image and (ii) identify regions corresponding to the contents in the dialogue – these functions are referred to as visual referring [63, 7] and grounding [40, 34], respectively. There are two main ways to integrate these functions into MLLMs, differing from each other in how the positional information is processed. The *explicit* methods [40, 52] introduced extra tokens to encode positions, while the *implicit* methods [9, 51, 54] used natural language to represent positions. Recently, there are also efforts [46] that used LLMs to call external vision modules for more flexible instance-level understanding quests.

**Video-based MLLMs.** Compared to the large corpus of image-based MLLMs, there are fewer video-based MLLMs for at least two reasons. First, there are fewer paired video-text data, especially for instance-level video understanding. Second, the higher dimensionality of video data poses a greater challenge to efficiently encode videos into visual features and find useful features to answer the questions. Existing efforts include VideoChat [29], Video-ChatGPT [37], Video-LLaMA [61], Video-LLaVA [31], LanguageBind [67], Valley [36], *etc.*; most of them followed the paradigm of image-based MLLMs and some of them [37] proposed a more efficient video feature. Recently, there have been some preliminary studies for instance-level video understanding, *e.g.*, LEGO [30] studied moment retrieval with the assistance of LLMs, and PG-Video-LLaVA [38] performed video grounding by employing off-the-shelf tracking and grounding modules. Merlin [57] studied video-based referring, but it was built upon three manually specified frames as visual input, incurring extra burden for users and also limiting the model’s ability to understand long and complex videos. *This paper aims to address the above two challenges, for which we set up a new formulation, establish a new benchmark named VideoRef45K, and present a solid baseline named Artemis.*

## 3 Artemis: A Baseline for Video-based Referential Understanding

### 3.1 Problem Formulation and Data Preparation

A video can be represented in the raw form of  $\mathbf{V} \in \mathbb{R}^{T \times W \times H \times C}$ , where  $T$ ,  $W$ ,  $H$ , and  $C$  stand for the number of frames, width, height, and the number of channels, respectively. In the task of video-based referential understanding (*a.k.a.* video-based referring), the model receives a question in the form of ‘What is the <region> doing in this video?’, where the concrete class of the referred object (like man or dog) is not provided, and the <region> is supplemented by a bounding box  $\mathbf{B} = (t; x_1, y_1, x_2, y_2)$  in a frame  $t \in \{1, 2, \dots, T\}$ . The expected output is a sentence describing the target’s action in the full video as detailed as possible (see Figure 1 for examples). Note that the proposed task requires a stronger ability beyond image-based referring and video captioning,

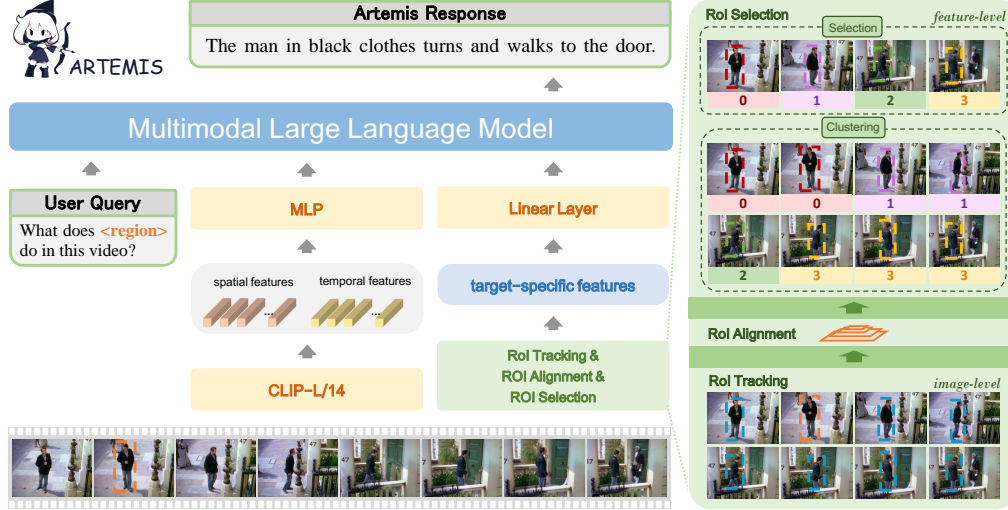


Figure 2: **Left**: the overall framework of Artemis, where an MLLM receives a text prompt together with spatial, temporal, and target-specific video features, and produces the answer. **Right**: the ROI tracking and selection mechanism to generate target-specific features. We use different IDs to show the clustering result. *This figure is best viewed in color.*

mainly in the coverage and granularity of visual understanding. Specifically, the model is expected to produce complex action descriptions for different target `<region>` specified.

We collect video data for referential understanding from 7 datasets, including HC-STVG [44], VID-Sentence [10], A2D Sentences [20], LaSOT [18], MeViS [16], GOT10K [24], and MGIT [23]. In total, there are 45K video QA pairs. We perform dataset-specific operations, including re-tracking (for HC-STVG and A2D-Sentences), clip cropping (for LaSOT and MGIT), and caption summary (for GOT10K), to convert them into the required form. Please refer to Appendix A for further details.

### 3.2 Overall Framework and Visual Features

The overall framework of Artemis, as illustrated in Figure 2, follows the pipeline of visual instruction tuning [33, 68] where a multimodal large language model (MLLM) receives video features with a text prompt and produces the desired output. We denote the function as  $\mathbf{T}_{\text{out}} = f(\mathbf{F}_{\mathbf{V}}, \mathbf{T}_{\text{in}})$ , where  $\mathbf{T}_{\text{in}}$  and  $\mathbf{T}_{\text{out}}$  are input and output texts (in tokens) and  $\mathbf{F}_{\mathbf{V}}$  is the set of features extracted from  $\mathbf{V}$ .

Compared to image-based referring, a clear difficulty of video-based referring arises from the high dimensionality of video data. Specifically, if we define  $\mathbf{F}_{\mathbf{V}}$  as the set of dense video features (e.g., using a pre-trained visual encoder such as the CLIP ViT-L model [41] to extract frame-wise visual features for  $\mathbf{V}$ ), the features often contain highly redundant information due to the similarity of neighboring frames. This brings two-fold drawbacks: (i) extra complexity for the MLLM to deal with these vision tokens, and (ii) extra difficulty for the MLLM to locate useful information, which leads to a slower convergence. To overcome this issue, we decrease the input feature dimensionality by using various slices to replace  $\mathbf{F}_{\mathbf{V}}$ , where each slice captures important yet complementary properties of the input video. Throughout this paper, we investigate three slices: the spatial, temporal, and target-specific video features.

**Spatial and temporal features.** The extraction of spatial and temporal video features follows the design of Video-ChatGPT [37]. Given a video clip  $V \in \mathbb{R}^{T \times W \times H \times C}$ , we use the CLIP ViT-L/14 visual encoder to cast it into frame-wise features, denoted as  $F_{\text{frame}} \in \mathbb{R}^{T \times W' \times H' \times D}$ , where the number of frames remains unchanged,  $W' = W/s$  and  $H' = H/s$  are the down-sampled resolution (e.g.,  $s = 14$  for ViT-L/14) of the visual features, and  $D$  is the feature dimensionality.) Then, these features are fed into average pooling along the  $T$  axis (into the spatial features  $\mathbf{F}_{\mathbf{V}}^{\text{S}} \in \mathbb{R}^{W' \times H' \times D}$ ) and along the  $W' \times H'$  plane (into the temporal features  $\mathbf{F}_{\mathbf{V}}^{\text{T}} \in \mathbb{R}^{T \times D}$ ), respectively.

**Target-specific features.**  $\mathbf{F}_V^S$  and  $\mathbf{F}_V^T$  have focused on the spatial and temporal features but ignored the referred target which may move or change during the video. To offer a compromise feature that captures spatiotemporal features, we propose an RoI (region-of-interest) tracking and selection mechanism (detailed in Section 3.3) and obtain a list of RoIs (represented as bounding boxes)  $\mathcal{B} = (\mathbf{B}_1, \dots, \mathbf{B}_M)$ , where  $M$  is the number of RoIs that are recognized by the algorithm to be important for referential understanding. We use the RoIAlign method [21] to extract visual features from each RoI, producing a set of target-specific features,  $\mathcal{F}_V^R = (\mathbf{F}_{V, \mathbf{B}_1}^R, \dots, \mathbf{F}_{V, \mathbf{B}_M}^R)$ .

**Instruction fine-tuning.** When the video features are ready, we feed them with the text tokens into Artemis. The MLLM follows instruction fine-tuning through three steps, gradually acquiring the ability of video-based referring. The details are described in Section 3.4.

### 3.3 RoI Tracking and Selection

Our goal is to extract compact features for video-based referring. The key lies in two factors, (i) completeness – locating the referred target in every video frame, and (ii) avoiding redundancy – not preserving too many features in the frames with similar semantics. We propose a simple solution upon RoI tracking and selection. As we shall see later, it offers a solid baseline for future work.

**Step 1: RoI tracking.** We apply HQTrack [69], an off-the-shelf tracking algorithm, to localize the RoI in each input frame. The pre-trained tracking model is not fine-tuned in the training phase. Given a RoI (a bounding box) in any video frame, the tracking algorithm outputs either a bounding box or nothing (*e.g.*, if the target is occluded) in each of the remaining frames. This step outputs a raw list of RoIs denoted as  $\mathcal{B}' = (\mathbf{B}'_1, \dots, \mathbf{B}'_{M'})$  where  $M'$  can be close to the number of frames.

**Step 2: RoI selection.** Feeding all tracked frames into the MLLM often incurs computational inefficiency and extra difficulties in model training. To avoid this, we select a subset from  $\mathcal{B}'$  containing  $M < M'$  RoIs, with the goal being to preserve diverse visual features using a limited number of RoIs. In practice, we pre-defined the target number,  $M$ , and adopt the K-means algorithm to form  $M$  clusters from the original set of  $M'$  RoIs. The final RoI list,  $\mathcal{B}$ , consists of a randomly chosen RoI from each cluster.

**Discussions.** Finding representative RoIs belongs to a generic topic of feature selection. On one hand, one can set a simple baseline by performing random or uniform sampling from the original set  $\mathcal{B}'$ . On the other hand, the information theory offers a general principle, *i.e.*, maximize the diversity of RoIs throughout the selection procedure. As demonstrated in Section 4.1, random and uniform sampling algorithms frequently fail to capture semantic changes throughout complex videos. By contrast, the simple K-means clustering used in Artemis significantly increases the diversity (see Appendix D), ensuring representative video features. We conjecture that the effectiveness of feature selection is related to the quality of video features; with stronger video foundation models, more sophisticated feature selection algorithms can make a larger difference. We leave this topic to future research.

### 3.4 Model Architecture and Training

The MLLM is built upon Vicuna-7B v1.5 [11], an open-sourced LLM<sup>4</sup>. We use CLIP ViT-L/14 [41] to extract visual features. To feed these 1024-dimensional visual tokens into the LLM, we use a learnable, two-layer MLP (1024-4096-4096) to project the visual features into the 4096-dimensional language space. We always use the auto-regressive framework to train the MLLM.

The training procedure of Artemis comprises three steps, (1) video-text pre-training, (2) video-based instruction tuning, and (3) video-based referring. The first two stages are similar to Video-LLaVA [31] but different training data are used. We set a unified template,

**User:** <video-tokens> <instruction> **Assistant:**  
guiding the model to output the desired answer. Here, <video-tokens> contains the spatial and temporal video features ( $\mathbf{F}_V^S$  and  $\mathbf{F}_V^T$ , projected by MLP), and <instruction> contains the language tokens of the task description (see below).

In the first stage, <instruction> has the form of ‘Write a terse but informative summary of the following video clip.’ and the model outputs the overall description of the video.

<sup>4</sup>A stronger LLM (*e.g.*, with a larger number of parameters) brings marginal improvement, because at the current stage, video understanding does not rely on strong language modeling abilities.

The training data includes image-text and video-text pairs, using images as still videos. We use a subset of 558K LAION-CCSBU image-text pairs with BLIP [28] captions, sourced from CC3M [42] and refined by LLaVA [33]. Additionally, we use the 702K video-text pairs provided by Video-LLaVA [31], derived from the 703K pairs constructed by Valley [36] using WebVid [3]. Only the MLP is trained (from scratch) in this stage, initializing the alignment of vision and language. The training elapses one epoch with a learning rate of  $1 \times 10^{-3}$ , taking about 5 hours on  $8 \times \text{A800 GPUs}$ .

In the second stage, `<instruction>` contains specific task descriptions like ‘Where is the person in the image?’ and ‘What is the person doing in the video?’, and the model follows the instruction to produce the answer. The training data comprises the 665K image-text instruction dataset from LLaVA-1.5 [33] and the 100K video-text instruction set from Video-ChatGPT [37]. Both the LLM and MLP are fine-tuned in this stage. The training elapses one epoch with a learning rate of  $2 \times 10^{-5}$ , taking about 20 hours on  $8 \times \text{A800 GPUs}$ .

In the third stage, we use the curated VideoRef45K dataset to endow the model with the ability of video-based referring. The template is modified as follows,

**User:** `<video-tokens> <refer-instruction> <track-instruction>` **Assistant:** Here, `<refer-instruction>` is formulated as ‘What is the `<region>` doing during this video?’ where the `<region>` token is replaced by the visual features extracted from the bounding box in the specified input frame, and `<track-instruction>` contains additional information, ‘This is the tracking list: `<region>`, ..., `<region>`’ where the `<region>` tokens are the target-specific features ( $F_{V, B_1}^R, \dots, F_{V, B_M}^R$ , projected by a Linear) extracted from the selected RoIs<sup>5</sup>, and the number of `<region>` token is  $M$ . In this stage, we fine-tune the LLM (with LoRA [22]), MLP and the RoI Align module. The training procedure elapses 3 epochs with a learning rate of  $4 \times 10^{-5}$ , taking about 3 hours on  $8 \times \text{A800 GPUs}$ .

## 4 Experiments

### 4.1 Artemis Is a Strong Baseline for Video-based Referential Understanding

**Setting and metrics.** We evaluate the ability of Artemis in video-based referring on the test set of HC-STVG [44]. The video and text data are pre-processed using the same method as in the training set. The test procedure uses the same instruction as in the third training stage and applies HQTrack [69] to localize the RoIs in video frames. We use the standard evaluation metrics including BERTScore [65], BLEU@4 [39], METEOR [4], ROUGE\_L [32], CIDEr [50], and SPICE [2].

Table 1: A comparison of video-based referring metrics on the HC-STVG test set. †: We use 5 key frames while using 8 frames leads to worse results.

Method	BERT Score	BLEU@4	METEOR	ROUGE_L	CIDEr	SPICE
Osprey [59]	0.8698	0.7	12.0	18.0	1.2	15.6
Ferret-13B [56]	0.8632	0.5	10.2	17.0	1.2	11.2
Shikra-7B [9]	0.8742	1.3	11.5	19.3	3.1	13.6
Video-ChatGPT [37]	0.8718	1.3	10.1	20.2	5.5	11.7
Video-LLaVA [31]	0.8639	1.7	9.8	20.8	2.6	9.1
Merlin [57]†	0.8829	3.3	11.3	26.0	10.5	20.1
Artemis (Ours)	0.9135	15.5	18.0	40.8	53.2	25.4

**Adapting existing MLLMs for video-based referring.** Due to the limited availability of research for video-based referring, we compare our model to a few recent MLLMs trained for image-based or multi-frame based referring<sup>6</sup>. The image-based referring models include Osprey [59], Ferret [56],

<sup>5</sup>During the training phase, we randomly select a frame with an annotated region bounding box as the input and employ the tracking module to locate the bounding box of the referred object in the sampled frames. During inference, we track the given region to generate the tracking list.

<sup>6</sup>The comparison against these methods is not totally fair because they have not been trained for video-based referring. We mainly use the comparison to claim that video-based referring is important and challenging, yet image-based MLLMs cannot do it well.

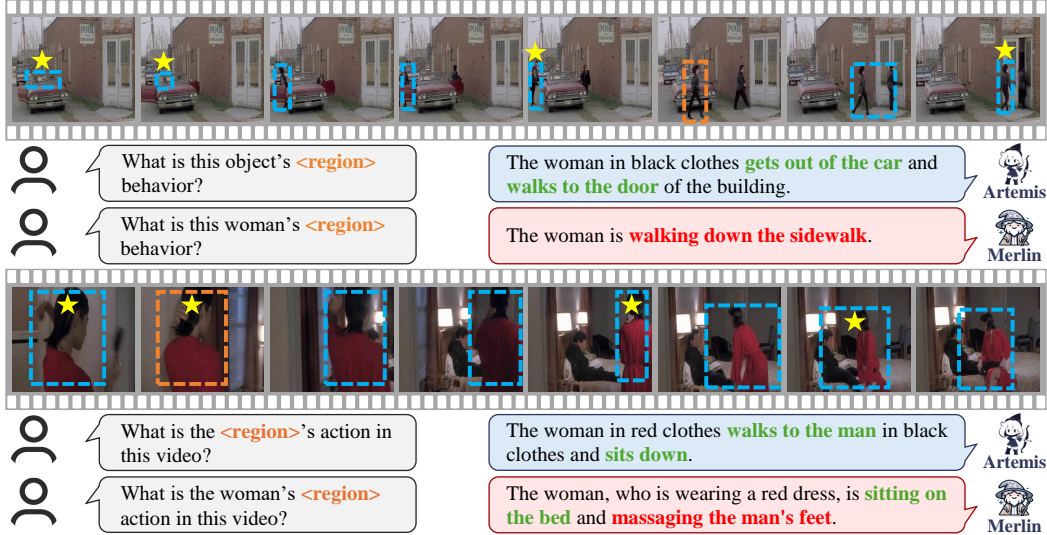


Figure 3: Artemis and Merlin for video-based referring. Note that Merlin needs the semantic class of `<region>` to be provided while Artemis does not. In each case, the orange rectangle indicates the input `<region>`, blue rectangles are the tracked RoIs, and yellow stars label the selected RoIs. Red and green texts indicate incorrect and correct answers, respectively. *This figure is best viewed in color.*

and Shikra [9]. For each video, we extract 5 key frames with RoIs produced by HQTrack and ask the trained model “What is the target `<region>` doing?” in the way the models are familiar with. Finally, we use GPT-3.5-Turbo to summarize the 5 answers into the overall description of the target. The multi-frame based reference model is Merlin [57] which receives 5 key video frames and RoIs and produces the overall description. The selection of key frames is consistent with Artemis. To compare with MLLMs that are trained for video understanding, such as Video-ChatGPT [37] and Video-LLaVA [31], we follow [43] to draw a red rectangle to mark the referred object in each key frame of the video. Then, we feed the rendered video to the models and ask the question “What is the target indicated by the red rectangle doing?”.

#### Quantitative results, and necessity of native video-based referring.

The numbers are summarized in Table 1. Artemis outperforms other MLLMs in each single evaluation metric. Note that the advantage is significant for some metrics, *e.g.*, BLEU4. Please refer to Figure 1 for representative examples. In Figure 9 (see Appendix C), we show the behavior of the methodology using a standalone LLM (*e.g.*, GPT-3.5-Turbo) upon image-based referring outputs. The image-based models tend to describe individual moments rather than an entire video; based on these inputs, the LLM cannot realize video descriptions and is sometimes confused to hallucinate what never happens in the video. The comparison validates the necessity of training a **native** model (*i.e.*, directly on the instruction data for video-based referring) like what Artemis has done. Equipping with such a fundamental ability of video understanding at a finer level, Artemis can perform even more complex video understanding tasks, as shown in Section 4.2.

**Qualitative results.** We display several representative examples of video-based referring in Figure 3. The output of Artemis is comprehensive (especially compared to other MLLMs, see Figure 1), often containing fine-grained actions of the target. This mainly concerns the compact video features extracted by the RoI tracking and selection algorithm that extracts key features for understanding.

**Ablative studies for target-specific video features.** The key to extracting compact target-specific features lies in RoI tracking and selection. To validate this, we ablate two key parameters: the strategy of RoI selection and the number of preserved RoIs. In Table 2, we define a baseline for region of interest. For each frame of a video containing an object of interest, we enclose the object’s location

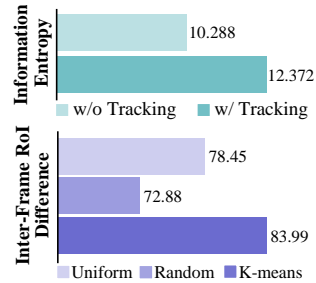


Figure 4: RoI manipulation increases the informativeness and diversity of RoIs. See Appendix D for details.

with a red rectangle and encode the video using Video-ChatGPT. The system is then queried with “What is the object in the red rectangle doing in this video?”. As illustrated in Table 2, K-means clustering emerges as a simple yet effective approach for RoI selection, whereas random or uniform sampling fails to consistently capture representative RoIs. To validate the effectiveness of RoI features in representing the object, we replace the <region> features with visual features (the [CLS] token from CLIP-ViT-L/14) in key frames, a variation we refer to as “w/<track-instruction>”. The performance decline compared to Artemis indicates that RoI features are of higher quality, as whole-frame features are more susceptible to background noise. Additionally, Table 3 demonstrates the importance of using multiple RoIs for understanding the full video content. While retaining more keyframes slows down both training and inference, it also introduces redundant information, leading to a slight drop in performance metrics. Empirically, using 4 RoIs provides the optimal balance on the HC-STVG test set, although increasing the number of RoIs may be beneficial for more complex videos.

From the information theory, we show that RoI tracking and selection improve informativeness (in terms of entropy) and diversity (in terms of frame-level difference) of the target-specific features in Figure 4. As shown in Figure 5, RoI tracking and selection gradually improve the comprehensiveness of the referring results.

Table 2: Ablation on different RoI selection methods. Results are reported on HC-STVG.

Method	BLEU@4	METEOR	ROUGE_L	CIDEr	SPICE
baseline	11.2	16.3	34.9	23.8	21.4
w/o	13.9	16.9	39.1	43.7	23.2
w/<track-instruction>	13.9	16.9	38.2	42.1	23.1
Uniformly	14.2	17.2	39.4	44.5	23.6
Randomly	14.3	17.1	40.0	46.5	24.2
Clustering	14.6	17.4	40.2	47.2	23.9

Table 3: Ablation on the number of selected RoIs. Results are reported on HC-STVG.

# RoI	Bert Score	BLEU@4	METEOR	ROUGE_L	CIDEr	SPICE
1	0.9114	13.9	16.9	39.0	43.3	23.6
2	0.9113	14.6	17.5	39.5	43.5	23.7
4	0.9125	14.3	17.1	40.0	46.5	24.2
6	0.9122	14.3	16.9	39.5	46.2	23.9
8	0.9110	14.0	17.0	39.0	43.1	23.6



Figure 5: How RoI tracking and selection gradually improves the quality of video-based referring. In each example, the orange rectangle indicates the input <region>, blue rectangles are the tracked RoIs, and green and yellow stars label the uniformly sampled and K-means selected RoIs, respectively. Red and green texts highlight the incorrect and correct outputs. *This figure is best viewed in color.*



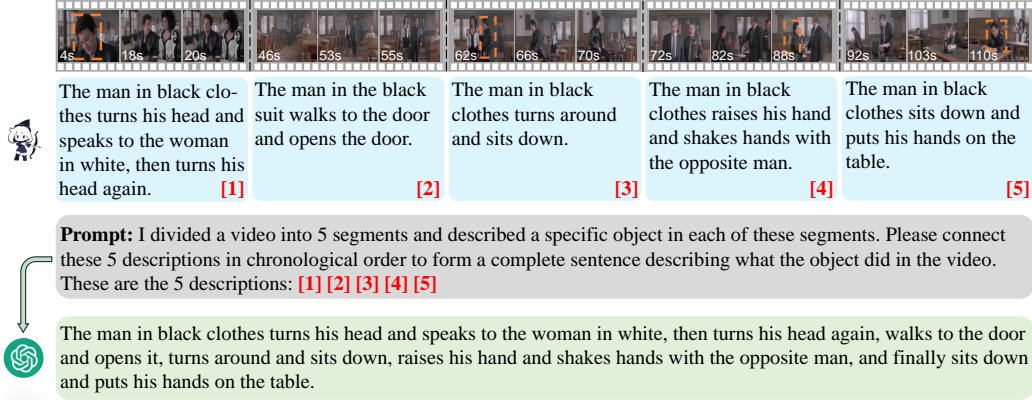


Figure 7: Example of long video understanding. We apply Artemis to output descriptions for segmented video clips and integrate them using an LLM (GPT-3.5-Turbo in this example).

## 4.2 Artemis Is a Building Block for Complex Video Understanding

With a strong ability of video-based referring, Artemis serves as a building block that strengthens the existing video-based MLLMs in complex video understanding.

**Multi-round video understanding with grounding.** Multi-round dialogues, especially answering logically related chain-of-questions [46], is an important yet challenging topic for MLLMs. In Figure 6 and Figure 14 in Appendix E, we show that Artemis’s referential understanding ability can be combined with image-based grounding models (e.g., GroundingDINO [34]) to answer multi-round chain-of-questions, where the entities mentioned in the video-based referring result is located and fed into the next round of video-based referring quest, allowing for more complex interactions.

### Long video understanding with text summarization.

Target-centric understanding of long videos is a major challenge for existing video-based MLLMs. The difficulty mainly lies in extracting compact video features (to feed into the MLLM) and tracking the target throughout the long video. We offer a simple solution that first segments the video into shorter clips, applies Artemis for understanding these clips, and applies an off-the-shelf LLM (e.g., GPT-3.5-Turbo) for summarization. As shown in Figure 7 and Figure 12 in Appendix E, the final output offers a comprehensive understanding. To our knowledge, this function was not achieved by existing MLLMs.

**Video question answering.** Lastly, we show that Artemis can perform general video question answering. We test the trained model on the Video-ChatGPT test set [37] and the other three benchmarks (i.e., MSVD-QA [8], MSRVT-QA [53], and ActivityNet-QA [6, 58]) where their training sets was not used to train Artemis. Results are summarized in Tables 4. Artemis shows competitive performance among a few recent MLLMs. These results inspire us that (i) an MLLM trained for finer-level video understanding can seamlessly transfer to coarser-level tasks, and (ii) extracting compact video features also benefits video question answering.

## 5 Conclusion

This paper proposes a challenging setting for video-based referring and establishes an effective MLLM named Artemis. Compared to existing methods, Artemis can understand human intention from simpler inputs (a text prompt and a single-frame bounding box) and comprehensively describe the target’s action in a complex video. At the core of Artemis is an RoI tracking and selection

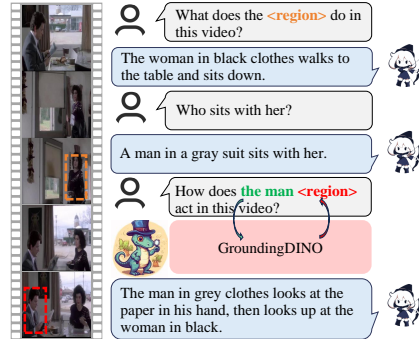


Figure 6: An example of multi-round, video-based referring by integrating Artemis with GroundingDINO [34].

Table 4: **Left:** Video QA on Video-ChatGPT. Metrics: correctness (CN), detail orientation (DO), contextual understanding (CU), temporal understanding (TU), consistency (CC). **Right:** Zero-shot video QA on MSVD-QA, MSRVTT-QA, and ActivityNet-QA. Metrics: accuracy (Acc.), score (Sc.).

Method	CN	DO	CU	TU	CC	Method	MSVD Acc.	MSRVTT Sc.	ActivityNet Acc.	ActivityNet Sc.
Video-Chat [29]	2.23	2.50	2.53	1.94	2.24	FrozenBiLM [55]	32.2	-	16.8	-
LLaMA-Adapter [62, 19]	2.03	2.32	2.30	1.98	2.15	Video-Chat [29]	56.3	2.8	45.0	2.5
Video-LLaMA [61]	1.96	2.18	2.16	1.82	1.79	LLaMA-Adapter [62, 19]	54.9	3.1	43.8	2.7
Video-ChatGPT [37]	2.40	2.52	2.62	1.98	2.37	Video-LLaMA [61]	51.6	2.5	29.6	1.8
Valley-v3 [36]	2.43	2.13	2.86	2.04	2.45	Video-ChatGPT [37]	64.9	3.3	49.3	2.8
						Valley-v3 [36]	60.5	3.3	51.1	2.9
<b>VideoRoI (Ours)</b>	<b>2.69</b>	<b>2.55</b>	<b>3.04</b>	<b>2.24</b>	<b>2.70</b>	<b>VideoRoI (Ours)</b>	<b>72.1</b>	<b>3.9</b>	<b>56.7</b>	<b>3.2</b>

mechanism to extract compact video features. Artemis shows advantages in video-based referring in VideoRef45K and transfers the ability to general video understanding, including being integrated with other modules for more complex tasks. We hope that Artemis can serve as a solid baseline to facilitate the research in fine-level video understanding.

**Limitations.** First, Artemis relies on a tracking algorithm to generate the RoIs; however, the tracking algorithm may produce inaccurate results and can confuse Artemis— see an example in Figure 13 (top) in Appendix E. Second, Artemis also suffers from the issues of general video-based understanding, such as the spatial-temporal aliasing problem, which can affect the model’s ability to describe the visual content accurately – see an example in Figure 13 (bottom) where Artemis accurately predicts the movement of the target but reverses the temporal order.

## 6 Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) under Grant No.62225208 and CAS Project for Young Scientists in Basic Research under Grant No.YSBR-117.

## References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.
- [2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016.
- [3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.
- [4] Satyanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Nieves. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.
- [7] Chi Chen, Ruoyu Qin, Fuwen Luo, Xiaoyue Mi, Peng Li, Maosong Sun, and Yang Liu. Position-Enhanced Visual Instruction Tuning for Multimodal Large Language Models. *arXiv preprint arXiv:2308.13437*, August 2023.
- [8] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

- [9] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing Multimodal LLM’s Referential Dialogue Magic. *arXiv preprint arXiv:2306.15195*, July 2023.
- [10] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee K Wong. Weakly-supervised spatio-temporally grounding natural sentence in video. *arXiv preprint arXiv:1906.02549*, 2019.
- [11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2023.
- [12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [13] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.
- [14] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [16] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2694–2703, 2023.
- [17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [18] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5374–5383, 2019.
- [19] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.
- [20] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5958–5966, 2018.
- [21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [23] Shiyu Hu, Dailing Zhang, Xiaokun Feng, Xuchen Li, Xin Zhao, Kaiqi Huang, et al. A multi-modal global instance tracking benchmark (mgit): Better locating target in complex spatio-temporal and causal relationship. *Advances in Neural Information Processing Systems*, 36, 2024.
- [24] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1562–1577, 2019.
- [25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. *arXiv preprint arXiv:2304.02643*, April 2023.
- [26] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning Segmentation via Large Language Model. *arXiv preprint arXiv:2308.00692*, August 2023.
- [27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv preprint arXiv:2301.12597*, May 2023.

- [28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.
- [29] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding, 2024.
- [30] Zhaowei Li, Qi Xu, Dong Zhang, Hang Song, Yiqing Cai, Qi Qi, Ran Zhou, Junting Pan, Zefeng Li, Van Tu Vu, Zhida Huang, and Tao Wang. Groundinggpt:language enhanced multi-modal grounding model, 2024.
- [31] Bin Lin, Yang Ye, Bin Zhu, Jiayi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2023.
- [32] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- [33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. *arXiv preprint arXiv:2304.08485*, April 2023.
- [34] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [36] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability, 2023.
- [37] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models, 2023.
- [38] Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, Mubarak Shah, and Fahad Khan. Pg-video-llava: Pixel grounding large video-language models, 2023.
- [39] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [40] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding Multimodal Large Language Models to the World. *arXiv preprint arXiv:2306.14824*, July 2023.
- [41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [42] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.
- [43] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997, 2023.
- [44] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12):8238–8249, 2021.
- [45] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
- [46] Yunjie Tian, Tianren Ma, Lingxi Xie, Jihao Qiu, Xi Tang, Yuan Zhang, Jianbin Jiao, Qi Tian, and Qixiang Ye. Chatterbox: Multi-round multimodal referring and grounding. *arXiv preprint arXiv:2401.13307*, 2024.

- [47] Yunjie Tian, Lingxi Xie, Jihao Qiu, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Fast-itpn: Integrally pre-trained transformer pyramid network with token migration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [48] Yunjie Tian, Lingxi Xie, Zhaozhi Wang, Longhui Wei, Xiaopeng Zhang, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Integrally Pre-Trained Transformer Pyramid Networks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18610–18620. IEEE, June 2023.
- [49] 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 arXiv:2302.13971*, 2023.
- [50] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.
- [51] Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. CogVlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.
- [52] Wenhao Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023.
- [53] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.
- [54] Shiyu Xuan, Qingpei Guo, Ming Yang, and Shiliang Zhang. Pink: Unveiling the Power of Referential Comprehension for Multi-modal LLMs. *arXiv preprint arXiv:2310.00582*, October 2023.
- [55] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141, 2022.
- [56] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and Ground Anything Anywhere at Any Granularity. *arXiv preprint arXiv:2310.07704*, October 2023.
- [57] En Yu, Liang Zhao, Yana Wei, Jinrong Yang, Dongming Wu, Lingyu Kong, Haoran Wei, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Wenbing Tao. Merlin:empowering multimodal llms with foresight minds, 2023.
- [58] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering, 2019.
- [59] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning, 2024.
- [60] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.
- [61] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023.
- [62] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.
- [63] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. GPT4RoI: Instruction Tuning Large Language Model on Region-of-Interest. *arXiv preprint arXiv:2307.03601*, July 2023.
- [64] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

- [65] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.
- [66] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *The Eleventh International Conference on Learning Representations*, 2022.
- [67] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiayi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Wancai Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment, 2024.
- [68] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.
- [69] Jiawen Zhu, Zhenyu Chen, Zeqi Hao, Shijie Chang, Lu Zhang, Dong Wang, Huchuan Lu, Bin Luo, Jun-Yan He, Jin-Peng Lan, et al. Tracking anything in high quality. *arXiv preprint arXiv:2307.13974*, 2023.

## A Data Curation

We organized 7 existing video datasets and performed a careful data curation procedure, resulting in the VideoRef45K benchmark, which comprises 45K video question-answer pairs. The 7 datasets include HC-STVG [44], VID-Sentence [10], A2D Sentences [20], LaSOT [18], MeViS [16], GOT10K [24], and MGIT [23].

HC-STVG is a movie clip dataset that provides tracking sequences and textual annotations describing the person’s actions during a certain period. We use the training portion, which contains approximately 10K video clips as our training data. The validation portion, containing 3,400 video clips, evaluates Artemis’s ability. The original tracking sequences in HC-STVG are of poor quality, so we use the off-the-shelf tracking model HQTrack [69] to regenerate the tracking sequences and remove some low-quality bounding boxes. To prevent tracking target deviation caused by cross-frame tracking, we select the first and middle frames with ground truth bounding boxes annotated by the HC-STVG dataset as the referring frames for HQTrack to generate tracking lists for the whole video. We then compare these two generated tracking lists with the HC-STVG annotations and exclude the frames with low IoU between the generated and annotated bounding boxes from the tracking lists.

A2D Sentences provides tracking sequences and captions for different objects, but the tracking sequences are only 3 frames long. To address this, we use HQTrack to regenerate the sequences and obtain longer tracking frames, extending them to 20 frames.

LaSOT provides a caption for an object along with its tracking sequence. However, LaSOT videos are usually long, and the captions for the entire video are generic. To address this, we extract three segments of 10 seconds each from the entire video for our training data.

GOT-10K is a tracking dataset that provides tracking sequences of objects and their categories, actions, and adverbs describing the actions. We concatenate these elements to describe the object’s action in the video, *e.g.*, “bear is slowly walking.”

MGIT videos are typically long, with annotations indicating the object’s actions at different time intervals. We extract these segments as our training data.

For MeViS and VID-Sentence, we did not perform any special processing. We converted the mask annotations of MeViS into bounding boxes.

Figure 8 shows some examples of VideoRef45K.



Figure 8: Some examples of VideoRef45K.

The total dataset composition is summarized in Table 5.

Table 5: Curated datasets for Video-based Referring.

Dataset	Video Clips	Q&A Pairs
HC-STVG [44]	10105	10105
MeViS [16]	1644	4489
A2D Sentences [20]	3017	5359
LaSOT [18]	2640	7920
VID Sentence [10]	4045	7654
GOT-10K [24]	9090	9090
MGIT [23]	105	614
<b>VideoRef45K</b>	<b>30646</b>	<b>45231</b>

Through the above processing steps, we obtained the VideoRef45K dataset. This dataset includes captions describing the actions of objects in videos, along with their corresponding tracking sequences. To utilize VideoRef45K, we created a template of questions to prompt the language model (LLM) to answer what the referred object did in a video. The template consists of ‘<refer-instruction>’ and ‘<track-instruction>’. The ‘<refer-instruction>’ is formulated as ‘what is the <region> doing during this video?’, and we utilized GPT-3.5-Turbo to generate the refer instruction template. The ‘<track-instruction>’ contains additional tracking lists to help the LLM perceive the referred object, such as ‘This is the region’s tracking list: <region> ... <region>’. We created several ‘<track-instruction>’ options as the track instruction template. During training, we randomly sampled a ‘<refer-instruction>’ from the refer instruction template and a ‘<track-instruction>’ from the track instruction template. The ‘<refer-instruction>’ and the ‘<track-instruction>’ are then concatenated as ‘<refer-instruction><track-instruction>’ to formulate the text prompt.

## B Implementation Details

We report the detailed training hyper-parameters of Artemis in Table 6.

Table 6: Training hyper-parameters of Artemis.

Configuration	Pre-training	Instruction Tuning	Referring Instruction Tuning
ViT init.	CLIP-L/14	CLIP-L/14	CLIP-L/14
LLM init.	Vicuna-7B-v1.5	Vicuna-7B-v1.5	Artemis-Finetune
Projection init.	random	Artemis-Pretrain	Artemis-Finetune
Image resolution	224	224	224
Video feature length	356	356	356
LLM sequence length	2048	2048	2048
Optimizer	AdamW	AdamW	AdamW
Peak learning rate	$1 \times 10^{-3}$	$2 \times 10^{-5}$	$4 \times 10^{-5}$
Minimum learning rate	0	0	$4 \times 10^{-5}$
Learning rate schedule	cosine decay	cosine decay	constant
Weight decay	0	0	0
LoRA rank	None	None	16
Number input trackbox	None	None	8
Number choosen bbox	None	None	4
Training steps	4927	5979	142
Global batch size	256	128	48
Numerical precision	bfloat16	bfloat16	float16

## C Image-based MLLMs for Video-based Referring

Existing multimodal language models (MLLMs) for video understanding rarely possess video-based referring capabilities. To address this limitation, we leverage image-based MLLMs (such as Osprey [59] and Ferret [56]) to perform image referring on 5 keyframes within the video independently. Subsequently, we integrate these outputs using GPT-3.5-Turbo to achieve video-based referring, as



depicted in Figure 9. However, these models struggle to perceive the action and behavior of the given region of interest (RoI) as effectively as Artemis.

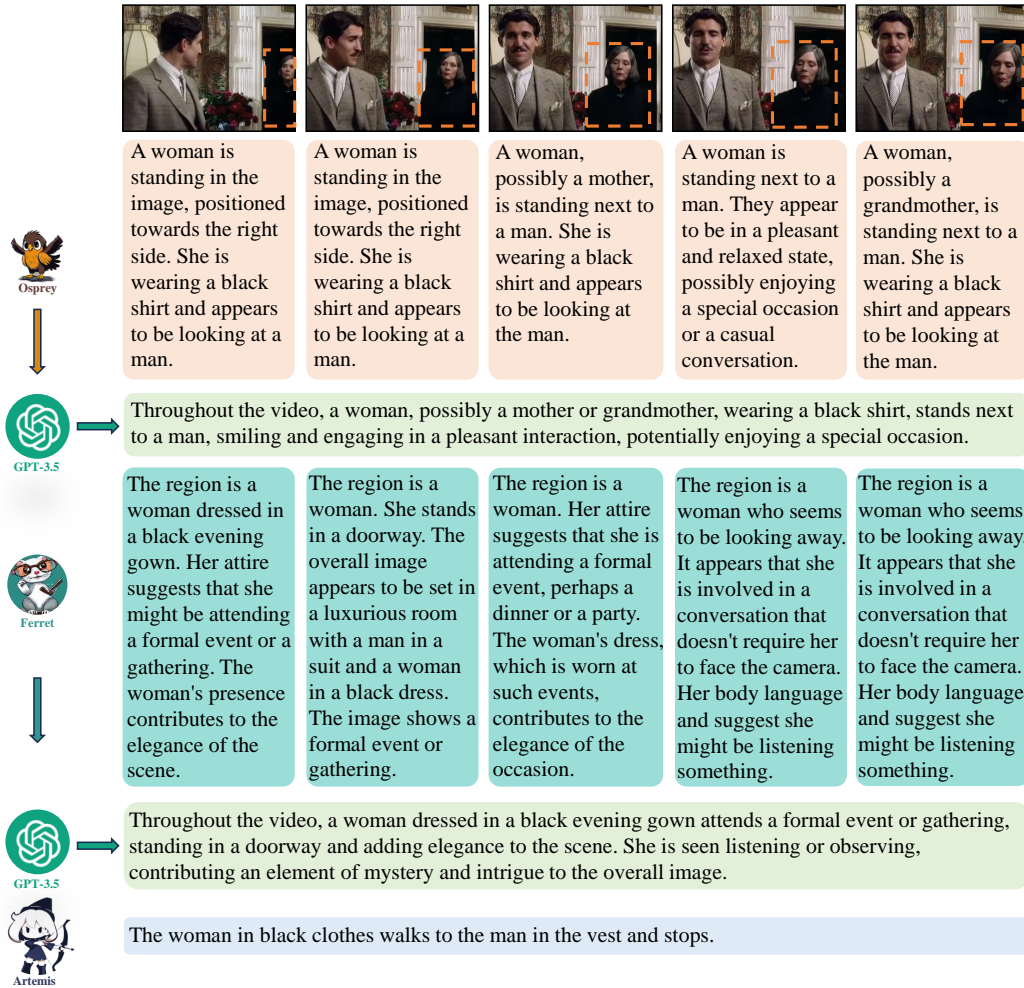


Figure 9: A comparison of video-based referring between image-based MLLMs and Artemis. GPT-3.5-Turbo is used to integrate the 5 independent outputs from the image-based MLLMs.

## D How RoI Tracking and Selection Improve the Results

To better demonstrate the effectiveness of our method, we conducted the following experiments. Firstly, we computed the attention values between different RoI tokens and temporal tokens. We observed that the tracking tokens added through tracking compensated for the weak perception aspect of the initial video tokens, as illustrated in Figure 10. Subsequently, as depicted in Figure 4 (top), we calculated the information entropy before and after adding the tracking list. Upon adding the tracking list, the overall amount of RoI information fed into the MLLM increased by 20.3%. Additionally, we computed the inter-frame differences of the boxes chosen from the tracking list. As shown in Figure 4 (bottom), the K-means clustering method selects RoIs with greater differences than random and average selection. This enables the MLLM to better perceive the action changes of RoIs throughout the entire video.

## E More Qualitative Examples

Figure 11 shows more examples of Artemis’s outputs of video-based referring. Figure 12 shows more examples of target-centric video understanding with text summarization. Figure 13 shows

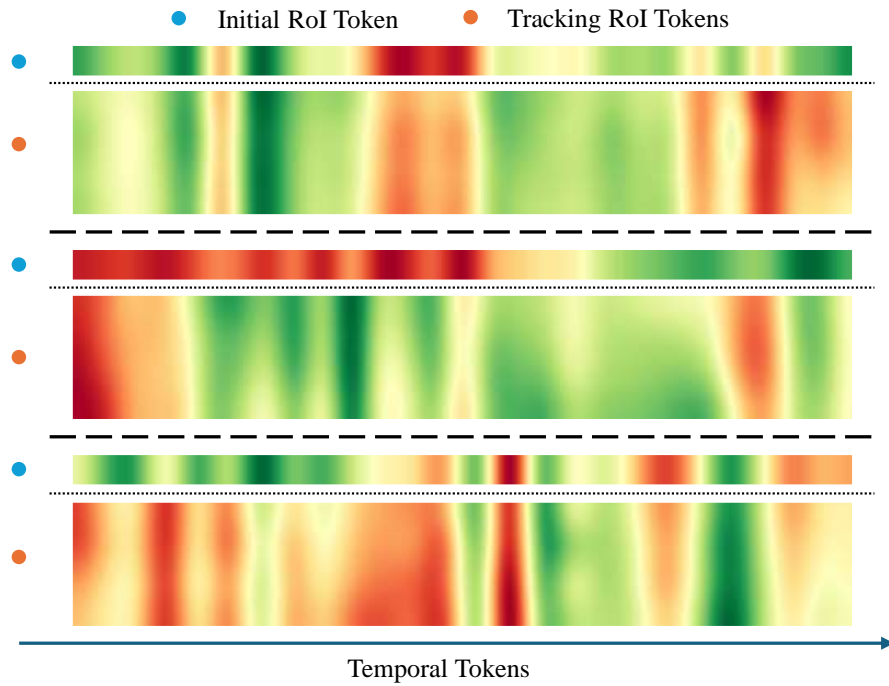


Figure 10: Attention map between RoI tokens and temporal tokens.

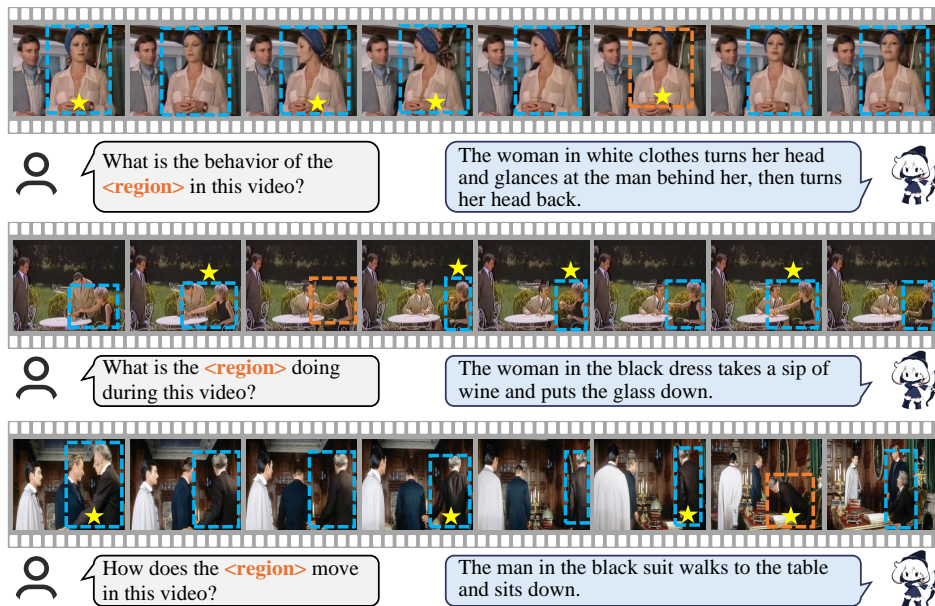


Figure 11: Examples of video-based referring generated by Artemis.

Artemis’s failure cases, revealing the limitation of Artemis. Figure 14 shows more example of combining Artemis with off-the-shelf grounding model, *e.g.* GroundingDINO to answer multi-round chain-of-questions conveniently.

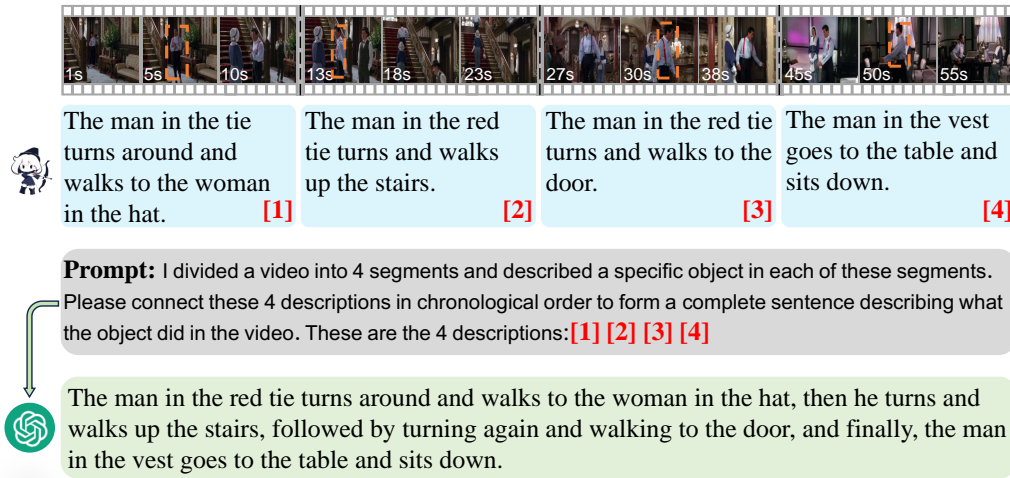


Figure 12: The example of long video understanding generated by Artemis.

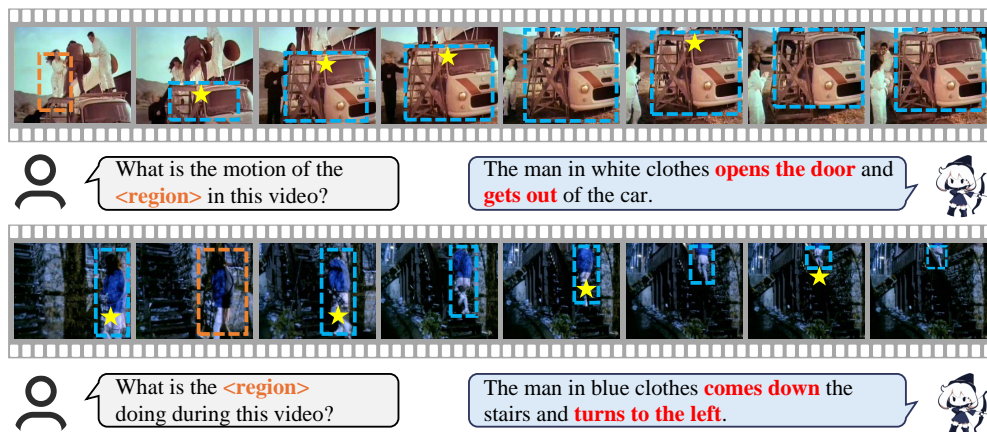


Figure 13: Failure cases of Artemis. **Top:** The tracking module generates inaccurate RoI list, misleading Artemis’s understanding. **Bottom:** Spatial-temporal aliasing in video hinders Artemis to perceive objects.



How does the <region> act in this video?

The woman in the blue coat opens the door and gets out of the car.

Who is also in the car?

A man is also in the car.

What is the action of the man <region> in this video?

The man in the car turns his head and looks at the woman in the blue coat, then turns his head again.



GroundingDINO



What is the motion of the <region> in this video?

The man in the red tie walks up the stairs.

Who stands with the man?

A woman stands with the man.

How does the woman <region> move in this video?

The woman in the hat turns and walks up the stairs.



GroundingDINO

What is in the woman's hand?

The woman is holding a bag in her hand.

What is the color of the bag?

The color of the bag is white.



What is the <region> do in this video?

The woman in white clothes turns around and sits on the bed.

Who is also in this room?

A man is also in this room.

What action is the man <region> perform in this video?

The man in short sheeves goes to the bed and sits down.



GroundingDINO

Figure 14: Examples of multi-round video understanding with grounding generated by Artemis.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction provide a comprehensive overview of the background and motivation of this study, effectively outlining its main contributions, thus accurately reflecting the paper's scope and significance.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We primarily focused on discussing the limitations associated with this study in Sec. 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper includes the theoretical result. Notably, it covers the results of information entropy and inter-frame RoI difference, ensuring completeness and accuracy in theoretical presentation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All information regarding the key contribution of this paper including architectural, data, and experimental configurations, have be fully disclosed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The supplementary material submitted with the manuscript includes scripts necessary to faithfully reproduce the main experimental results and the anonymous link includes the code and data. Instructions for running the code are also provided within the anonymous link.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies detailed experimental configurations in Section 4.1 in main manuscript and Section A and Section B in Appendix, providing readers with essential information to comprehend the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Generally, papers about MLLMs do not include error bars, and we have found that the MLLM’s training is quite stable with little variation across multiple runs. However, we have provided the code, hyperparameters, and random seeds used in our experiments to facilitate the reproducibility of our findings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All experiments were carried out on an 8×A800 GPU server, as detailed at Section 3.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: After carefully reviewing the referenced document, we certify that the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]



Justification: The paper primarily focuses on multimodal large language models about video understanding using publicly available datasets that have undergone thorough validation. This study only serves as a foundation model for video-based referring, which is not directly applicable for practical scenarios until now.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The proposed models are trained on benchmark datasets such as HC-STVG, VID-Sentence, and S2D Sentences, *etc.* These datasets have been extensively used in the computer vision community and have undergone comprehensive safety risk assessments.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In the paper, we clearly specified the datasets and code sources used, and provided appropriate citations in the reference section. Additionally, we ensured transparency by including the original sources of any modified code files, making the changes traceable.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have included the code, data, along with detailed usage instructions, in a anonymous link. After the review process is completed, we will make the code and data publicly available to the community.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This study does not involve any crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No crowdsourcing experiments or research with human subjects were involved in this study. All experiments were conducted using code and GPU servers.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.