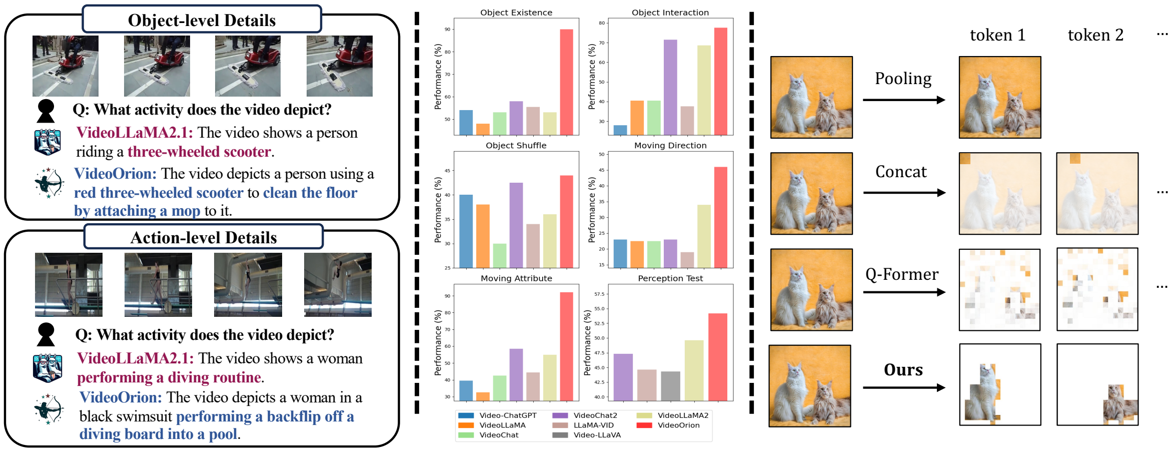
VideoOrion: Tokenizing Object Dynamics in Videos

Yicheng Feng[1]† Yijiang Li[2]† Wanpeng Zhang[1] Sipeng Zheng[4]
Hao Luo[1] Zihao Yue[3] Zongqing Lu[1,4]‡

[1]School of Computer Science, Peking University, [2]University of California, San Diego,
[3]Renmin University of China, [4]BeingBeyond



(a) VideoOrion capturing more details

(b) Performance on finer details and object tasks

(c) Entangled v.s. disentangled visual representation

Figure 1. With explicit modeling of object dynamics, *VideoOrion* can (a) grasp finer details (b) with understanding on object-related fine-grained details. (c) Comparison with prior encoding including: (1) spatial pooling the whole frame into a single token; (2) concatenating adjacent patch tokens into a single token; (3) Q-Former aggregates patch tokens with learnable queries. (4) *VideoOrion* with *object tokens* providing disentangled semantics.

Abstract

We present VideoOrion, a Video Large Language Model (Video-LLM) that explicitly captures the key semantic information in videos—the spatial-temporal dynamics of objects throughout the videos. VideoOrion employs expert vision models to extract object dynamics through a detect-segment-track pipeline, encoding them into a set of object tokens by aggregating spatial-temporal object features. Our method addresses the persistent challenge in Video-LLMs of efficiently compressing high-dimensional video data into semantic tokens that are comprehensible to LLMs. Compared to prior methods which resort to downsampling the original video or aggregating visual tokens using resamplers, leading to information loss and entangled semantics, VideoOrion not only offers a more natural and efficient way to derive compact, disentangled semantic representations but also enables explicit object modeling of video content with minimal computational cost. Moreover, the introduced object tokens naturally allow VideoOrion to accomplish video-based referring tasks. Experimental results show that VideoOrion can learn to make good use of the object tokens, and achieves competitive results on both general video question answering and video-based referring benchmarks.

1. Introduction

The remarkable performance of Large Language Models (LLMs) [16, 48, 62] has spurred interest in extending these models' capabilities beyond text, which catalyzes the development of multi-modal large language models (MLLMs) [4, 9, 36, 40, 60, 79, 81]. By integrating diverse modal-

† Equal contribution.
‡ Correspondence to <zongqing.lu@pku.edu.cn>

ities through tokenization and alignment with text tokens, MLLMs enable a broader range of real-world applications. However, a significant challenge lies in efficiently encoding the information from multi-modal inputs into a limited number of tokens, particularly for Video-LLMs, as videos encapsulate much more complex and detailed information than other input modalities, such as an image.

To compress high-dimensional visual information into a more compact representation, existing studies commonly employ downsampling or pooling techniques before tokenization [26, 43] and integrate various aggregation modules to reduce the number of visual tokens [1, 14, 30], thereby mitigating computational costs. Several limitations arise. First, due to computational constraints, Video-LLMs usually only sample a small fraction of the frames in the video for training and inference (e.g., sample 8 or 16 frames out of thousands of frames for a video of about three minutes). Despite being more efficient, they inevitably incur information loss, particularly in the fine-grained dynamics of objects and interactions between scenes. The discretized frames also fail to provide sufficient contextual information for Video-LLMs to effectively model long-range temporal dependencies within videos. We hypothesize that this limitation is a key factor preventing current video-LLMs from achieving a more detailed understanding of video content, restricting their ability to capture intricate nuances beyond a general overview. Furthermore, existing methods encode video tokens by processing image patches through a vision encoder, often overlooking the explicit semantics embedded within these visual tokens. This oversight can lead to semantically entangled representations [70]. In contrast, text tokens inherently carry clear and well-defined semantics, creating a significant disparity that complicates the alignment of ambiguous visual tokens with semantically precise text tokens in the LLM.

In this paper, we present *VideoOrion*, a Video-LLM with a novel vision encoding method that explicitly captures the key semantic information in videos. Drawing inspiration from the way humans naturally identify object semantics from visual observations before integrating contextual information into cognitive processes, we argue that a tokenizer for the visual modality should provide semantically rich tokens, akin to tokenizers in text processing, to enhance a model's understanding of visual content. A key aspect of visual semantics lies in the object dynamics, including their appearance, interactions, and temporal variations.

To effectively capture these dynamics, *VideoOrion* introduces a detect-segment-track pipeline that utilizes expert models to extract objects and their evolving characteristics across a sequence of frames. This information is then fused into a set of *object tokens*, representing the objects and their spatial-temporal dynamics throughout the video. Beyond object dynamics, contextual visual information is also essential, complementing object tokens for a more comprehensive understanding of videos. To address this, we propose to also supplement the object tokens with a set of *context tokens* produced by a Video Projector. By incorporating both encoding branches, *VideoOrion* effectively captures overarching contextual elements (e.g., static scene information) while also preserving fine-grained details about specific objects or instances through object tokens. This disentangled object representation enables the subsequent LLM to more accurately model spatial and temporal dynamics of objects, ultimately improving video comprehension.

Through extensive experiments and visualization, *VideoOrion* demonstrates superior performance in video understanding tasks across multiple benchmarks. Moreover, the proposed object tokens naturally facilitate video-based referring, i.e., visual question-answering involving a specific object or instance in the videos (provided in the first frame)[54, 66, 75]. Notably, *VideoOrion* exhibits remarkable capabilities on video-based referring tasks, achieving substantial gains compared to previous methods. Our main contributions can be summarized as follows:

- We present *VideoOrion*, featuring a novel object branch that encodes the spatial-temporal dynamics of objects and instances in the videos through a set of *object tokens*.
- To effectively capture object dynamics, we propose a detect-segment-track pipeline that leverages knowledge from expert vision models to extract object masks across frames, thereby explicitly generating disentangled objects representations.
- We conduct extensive experiments and ablation studies on multiple benchmarks, showcasing consistent improvements with object tokens and achieving competitive results on general VQA and video-based referring tasks.

2. Related Work

Video-LLMs. Most existing Video-LLMs extend the framework of image-based MLLMs by introducing various adapters to handle the large number of visual tokens generated by image-level pretrained visual encoders such as CLIP [55]. Prominent models include VideoChatGPT[43], which employs spatial-temporal pooling; VideoChat[30], VideoChat2[32], and VideoLLaMA[79], which leverage Q-Former [29, 84]; MiniGPT4-Video[3], which concatenates adjacent visual tokens and applies linear mapping; VideoLLaMA2[14], which incorporates 3D convolution blocks; and LLaMA-VID [34], which utilizes cross-attention between visual and textual embeddings. While these approaches have achieved promising results, they predominantly emphasize adapting visual features for integration into large language models (LLMs), often neglecting explicit semantic representation within visual tokens. Consequently, the task of interpreting token semantics is largely delegated to the LLMs themselves.

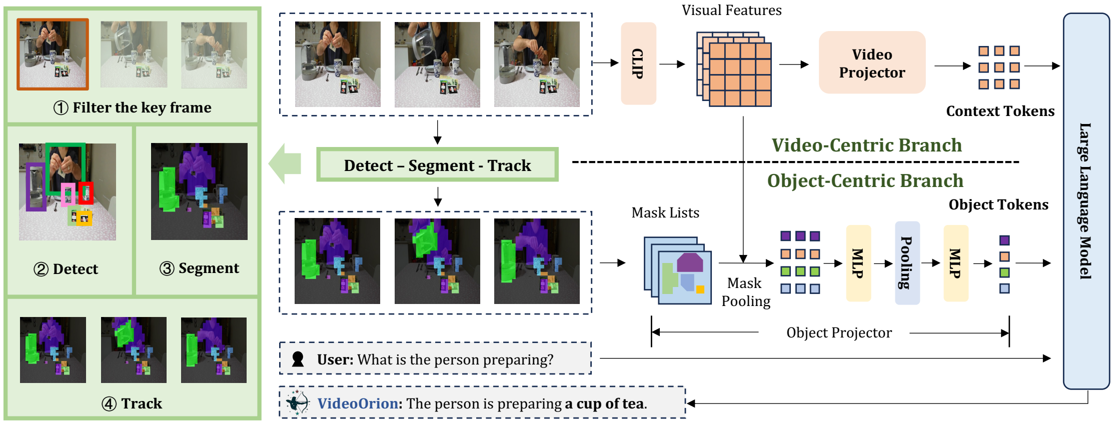


Figure 2. The overall architecture of *VideoOrion*. Two branches are employed to encode the video content into tokens: the Video-Centric Branch encodes the general information with *context tokens*, while the Object-Centric Branch encodes the dynamics of objects through the detect-segment-track pipelines in the video into a set of *object tokens*. All these tokens are fed together to the LLM for integrating information from both branches and generating responses to the text inputs.

There are also studies emphasizing the semantic aggregation of visual tokens. For instance, Chat-univi [25] aggregates visual tokens into dynamic clusters using DPC-KNN [15], a clustering algorithm aimed at reducing redundancy by merging visually similar tokens. Unlike Chat-univi, *VideoOrion* aggregates visual tokens explicitly in an object-centric manner, thus ensuring clearer semantic representation. Similarly, Video-LaVIT [26] employs the MPEG-4 compression algorithm to distill video information into key frames and motion features, subsequently encoding these using VQ-VAE. While our approach also segments videos via key frames, we prioritize the semantic representation of objects rather than general motion information. The closest related study is Slot-VLM [70], which shares conceptual similarities with ours by employing slot attention mechanisms to cluster tokens into object-centric and event-centric representations. Our methodology substantially diverges by deploying specialized vision models to precisely detect and extract object representations and incorporating temporal fusion techniques to capture the dynamics of objects across frames. This ensures that our resulting object tokens encapsulate richer, temporally integrated semantic information. Artemis [54] also incorporates ROI tracking to derive target-specific features from visual tokens but predominantly targets video-based referential understanding tasks with exclusive emphasis on individual objects. In contrast, our research aims to enhance the object-centric comprehension of Video-LLMs by systematically aggregating and encoding semantic tokens for all identified objects within the observed video sequences.

Vision Foundation Models. Humans naturally incorporate visual information into reasoning by semantically segmenting and extracting meaningful visual content, particularly when identifying distinct objects or entities. However, existing Multimodal Large Language Models (MLLMs) often neglect this crucial semantic abstraction. Notably, this semantic-level processing aligns closely with several fundamental computer vision tasks, including object detection [8, 49, 68, 73, 78], instance segmentation [12, 21, 33, 35], and video object segmentation [52, 67, 72].

To bridge this gap, we propose to leverage specialized vision foundation models to more effectively encode object dynamics, rather than exclusively relying on video-text paired approaches such as CLIP [55]. For instance, GroundingDINO [41] introduces open-vocabulary object detection capabilities by marrying DINO [49, 78] with grounding abilities by contrastive training on object region-text pairs [8, 73]. Moving beyond text-based prompts, another line of research proposes SAM [13, 57] with the ability to interactive segment anything with points, scribbles, and bounding boxes. Further, beyond understanding a single image, multi-object tracking [44, 53, 82] and object segmentation [52, 67, 72] in video contexts have been extensively studied, resulting in the emergence of several specialized vision foundation models [13, 46, 57, 85].

3. Methodology

Building upon the discussion presented in Section 1, we introduce *VideoOrion*, an instantiation of an object-centric representation characterized by enriched semantics, increased compactness, and greater efficiency—achieving comprehensive information encoding with fewer tokens. It is important to emphasize that one of our primary contributions lies in this conceptual idea of object-centric representation, with *VideoOrion* serving merely as a specific realization of this broader concept, detailed further below. *VideoOrion* demonstrates the effectiveness of utilizing a more disentangled and compact representation of object dy-

Table 1. Data mixture of *VideoOrion* training. For video-centric branch, we directly utilized open-source base models.

Stage	Modality	# Samples	Object Tokens	Source
Video pretraining	VideoLLaMA2 Base Model			
Object pretraining	video-text	700K	√	InternVid-10M, OpenVid-1M
MM instruction tuning	video-text	2.7M	√	Video-ChatGPT, VideoChat2, LLaVA-Video
	video-text	1M	√	Ego4D, EgoExo4D
	image-text	625K	√	LLaVA
	text-only	40K		Video-LLaVA

namics, which enhances and complements general video features, thereby significantly improving video modeling and comprehension.

3.1. Overall Architecture

VideoOrion features on an Object-Centric Branch complemented by a Video-Centric Branch, designed for comprehensive tokenization of both general video context and specific object dynamics, as depicted in Figure 2. These two branches function independently, with the downstream large language model (LLM) tasked with integrating semantic information derived from each branch. This decoupled design, naturally enables *VideoOrion* with high flexibility, readily accommodating various architectural configurations, i.e. different video projectors [14, 29, 39, 43, 79] and LLMs [24, 71].

3.2. Video-Centric Branch

Given a video $V_i \in \mathbb{R}^{T \times H \times W \times C}$, where T denotes the number of frames. The Video-Centric Branc first utilizes a vision encoder, e.g. CLIP to extract the features before employing a Video Projector to project the high-dimensional per-frame features into a set of *context tokens*. This procedure usually incurs large computation due to the large number of *context tokens* (e.g. 576 per frame) which limits the number of frames to be encoded. Consequently, these tokens can only encode the general and static information, such as the background and scene. *VideoOrion* supports diverse Video Projector designs; here, we employ the STC Connector introduced by VideoLLaMA2 [14] for its superior preservation of spatial-temporal details, enhancing the association between visual and object tokens. Specifically, we first sample a fixed number of frames from the video, and use a vision encoder such as CLIP (ViT-L/14) [55] to encode each frame, resulting in the embedding $x_i \in \mathbb{R}^{t_v \times h \times w \times D}$, where t_v is the number of sampled frames, and $h = H/p, w = W/p$ are the resolution of visual embeddings, $p = 14$ for ViT-L/14. Then, we use STC Connector, which is composed of two RegStage blocks [56] and a 3D convolution block, followed by an MLP projector, to transform x_i into context tokens $v_i \in \mathbb{R}^{N_v \times d}$, where N_v is the number of context tokens and d is dimension of the LLM decoder's token embedding space.

3.3. Object-Centric Branch

In the Object-Centric Branch, our objective is to extract a concise set of *object tokens* from the entire video, each encapsulating both semantic and spatial-temporal information related to individual objects or instances and their dynamics throughout the video. This process begins with object proposal generation, where instances of interest are identified, followed by dynamic tracking to establish their trajectories across frames. Finally, an Object Projector aggregates the encoded features of these dynamics and trajectories to derive compact *object tokens*.

Detect-segment-track pipeline. Identifying and tracking the dynamics of objects in videos present significant challenges. A fundamental question is: How can we efficiently represent an object along with its temporal dynamics? To achieve this, an ideal representation should capture both the spatial location and fine-grained details of objects throughout a video's progression. Using masks to refer to objects' representations naturally emerge as a superior choice as masks can not only precise spatial information, but also encapsulate precise object contours, free-form shapes, and other fine-grained details (compared to coarse-grained bounding boxes), which have been widely adopted in referring and grounding methods [57, 74] for object representation. By associating a sequence of masks with an object across multiple frames, we can effectively characterize its motion and evolution within the video.

To identify objects of interest, we employ GroundingDINO [41] in its generic mode during the *detect* stage, as it offers both efficiency and strong performance in generating object region proposals in specific frames. In the subsequent *segment* stage, we leverage SAM [27] to refine these bounding box proposals into precise object masks. However, due to the dynamic nature of objects and scene changes, objects may enter or exit the frame over time. A naive approach that tracks only the objects present in the first frame risks substantial information loss. A possible strategy is to uniformly sample multiple frames and segment the video into clips for object identification and tracking. However, this approach may fail to capture newly appearing objects accurately. Another alternative is to use tracking algorithms capable of detecting new objects automatically; however, these methods are typically limited to closed-set object categories, thus unsuitable for general video analysis. To address these challenges, we propose a novel segmentation method that dynamically partitions the video based on variations in object presence across frames. Specifically, we sample a sequence of frames, $f_1, f_2, ..., f_n$, and apply RAM++ [23], an open-world image recognition model, to annotate each frame with a set of object-related tags. We then utilize the NLTK toolkit [6] to filter these tags, extracting concrete nouns and their synonyms to construct a tag set $l_1, l_2, ..., l_n$, which serves as an indicator of object presence in each frame. To refine this process, we introduce two thresholds, θ_a and θ_b. The first threshold, θ_a, is used to filter out frames with a low number of object tags, as such frames are often noisy: $\mathcal{F}_1 = \{f_u \mid u \in \{1, \ldots, n\}, |l_u| > \theta_a\}$. The first frame in $\mathcal{F}_1$ is designated as the initial key frame. Subsequently, for each subsequent frame, we compute the overlap between its tag set and that of the most recent key frame. If the overlap falls below the threshold θ_b, i.e., $|l_u \cap l_{key}| < \theta_b$. we consider this a significant object transition and designate the frame as a new key frame. The values of θ_a and θ_b are empirically determined based on the performance characteristics of RAM++, with our experiments using $\theta_a = 3$ and $\theta_b = 2$.

Through this process, we obtain a sequence of key frames that segment the video into distinct clips based on object transitions. For each key frame, we apply the object proposal and segmentation steps to generate object masks.

In the tracking stage, we sample t_o frames from the video and partition them into clips based on the key frames. Notably, t_o is typically much larger than t_v since it does not affect the number of tokens, thereby keeping computational costs manageable. In our experiments, we set $t_v = 8$ and use $t_o = 64$ for short videos, increasing to $t_o = 128$ for videos longer than one minute. A key advantage of this approach is that object tokens retain richer temporal information compared to context tokens. Finally, we utilize XMem [13], a multi-object tracking algorithm, to track all object masks across the entire video, starting from each key frame. This results in a set of object mask lists: $\mathcal{M} = \{M_1, M_2, ..., M_{N_{oi}}\}$ where N_{oi} represents the total number of identified objects in video V_i, and M_j corresponds to the mask sequence for object j across frames.

Object Projector. We introduce Object Projector to translate the list of object masks into *object tokens*. Each object j, appearing in k frames, has an associated mask list $M_j = \{m_j^{f_{j1}}, m_j^{f_{j2}}, \ldots, m_j^{f_{jk}}\}$. Mask pooling is performed on each mask to fuse the features from the corresponding frames $f_{j1}, \ldots, f_{jk}$. Subsequently, temporal pooling, followed by an MLP projector, integrates these spatially pooled features into a unified representation. Consequently, for each V_i, we obtain N_{oi} compact *object tokens* $o_j \in \mathbb{R}^d$, where $j \in \{1, \ldots, N_{oi}\}$. These object tokens encapsulate both spatial and temporal dynamics effectively.

3.4. Training

Instruction Template. To integrate context tokens and object tokens with text tokens for input into the LLM, we define the following input template:

> **User:** $<v>$ Here is a list of objects and instances in the video: $<o_1>$,$<o_2>$,...,$<o_N>$ `<Instruction>`
> **Assistant:** `<Answer>`

where $<v>$ represents the context tokens of the video sample, and $<o_1>$, $<o_2>$, ..., $<o_N>$ denote the object tokens. Following the two-branch design of *VideoOrion*, the training procedure is divided into three stages: 1) Video-Centric Branch pretraining; 2) Object-Centric Branch pretraining; 3) Multi-modal instruction tuning.

Video-Centric Branch pretraining. At this stage, only the Video Projector is optimized using video-text and image-text pairs for vision-language alignment, without incorporating object tokens. To ensure simplicity and computational efficiency, we directly utilize the open-source base model of the STC-Connector from VideoLLaMA2[14], which has been pretrained on 12.2M vision-language data for vision-language alignment [14].

Object-Centric Branch pretraining. At this stage, the pretrained Video Projector remains frozen, and only the Object Projector is optimized using video-text pairs. We construct a training dataset by sourcing captioned videos from InternVid-10M [69] and OpenVid-1M [47]. To refine the dataset, we first filter InternVid-10M using aesthetic scores[59] and UMT-SIM scores[31], which assess video quality and video-caption similarity, respectively, yielding a 1.4M-sample subset. This subset is then combined with OpenVid-1M, and the resulting dataset undergoes further filtering based on noun-phrase concept balance, following [40]. Ultimately, we obtain a pretraining dataset comprising 700K video-text pairs. To enhance training efficiency, we preprocess this dataset using a detect-segment-track pipeline to generate object mask lists.

Multi-modal instruction tuning. In the final stage, we only freeze the vision encoder, and optimize the Video Projector, the Object Projector, and the LLM backbone together, with multi-modal (MM) instruction tuning datasets. We adopt the training data from Video-LLaVA[37], VideoChat2[32] and LLaVA-Video[83]. We also incorporate an ego-centric video question-answering dataset built with videos and annotations from Ego4D[19] and EgoExo4D[20]. We sampled 1M ego data samples to make up to the 4M dataset size. We segment egocentric videos into shorter clips and generate question-answer pairs based on the original human annotations as well as additional annotated visual information. These question-answer pairs encompass multi-level knowledge, ranging from low-level visual basic facts to high-level behavior-centric understanding, and feature diverse types of questions, spanning from descriptive to deductive in nature. We preprocess the video-text samples and image-text samples with the detect-segment-track pipeline to obtain the object mask lists here. For images, we treat them as single-frame videos, and the tracking model is not used in the pipeline.

We use auto-regressive cross-entropy loss in all three stages. The data mixture is summarized in Table 1.

4. Experiments

4.1. Experimental Setup

We present two variants of *VideoOrion*: *VideoOrion*, which uses CLIP (ViT-L/14) [55] as the vision encoder and Mistral-Instruct-7B [24] as the LLM backbone; and *VideoOrion*+, which uses SigLIP (so400m-patch14-384) [77] as the vision encoder and Qwen2-7B [71] as the LLM backbone. We sample $t_v = 8$ frames from each video for the Video-Centric Branch for *VideoOrion*, and set $t_v = 16$ for *VideoOrion*+. We sample $t_o = 64$ frames from short videos and $t_o = 128$ frames from videos longer than 1 minute for the Object-Centric Branch. We set $\theta_a = 3$ and $\theta_b = 2$ when extracting key frames to split videos with RAM++, which we find have the best performance, and the number of sampled frames for tagging is set to $n = 16$ for Object-Centric Branch pretraining and $n = 64$ for MM instruction tuning. We resize and crop each frame to 336×336 for CLIP [55] and 384×384 for SigLIP [77], following their implementations. We use a learning rate of $1e-4$ and $5e-6$ for the two stages, with a warm-up ratio of 0.03. The global batch size for Object-Centric Branch pertaining is 256 and 128 for the MM instruction tuning stage. In both stages, we train for only one epoch, with $8\times$ A800 GPUs, for *VideoOrion* and $64\times$ A800 GPUs for *VideoOrion*+.

4.2. Video Question Answering

Evaluation benchmarks. We comprehensively evaluate the video understanding capabilities of *VideoOrion* and *VideoOrion*+ on four multi-choice video question answering (MC-VQA) datasets including MVBench [32], EgoSchema [45], Perception-Test [51] and VideoMME [18]. Accuracies are reported for each of the benchmarks. We also test on an open-ended video question answering (OE-VQA) benchmark ActivityNet-QA [76], where ChatGPT3.5 is employed to evaluate the answer following Maaz et al. [43], by two metrics: a yes/no indicator that signifies whether the predicted answer matches the correct answer, and a score ranging from 0 to 5 that reflects the degree of alignment between the model output and the correct answer. We report both the accuracy and the average score.

Zero-shot performance. We compare the performance of *VideoOrion* and *VideoOrion*+ with prior state-of-the-art Video-LLMs in Table 2. *VideoOrion* achieves competitive performance consistently surpassing the second-best methods by a large margin. Notably, compared to the baselines VideoLLaMA2 and VideoLLaMA2.1 which has the same Video-Centric Branch, *VideoOrion* and *VideoOrion*+ consistently outperforms them by 10.1%, 14.6%, 15.6%, 8.7%, 7.8% and 10.1%, 11.9%, 11.0%, 5.1%, 7.3% on MVBench, EgoSchema, Perception-Test, VideoMME and ActivityNet-QA respectively. This result demonstrates the effectiveness of Object-Centric Branch for general video understanding, serving as a proof of concept that explicit disentangled object presentation can efficiently and effective encode the rich information in videos for LLMs to comprehend.

4.3. Video-based Referring

With the proposed Object-Centric Branch, *VideoOrion* inherently supports video referring—a capability often absent in conventional Video-LLMs. To enable video referring, we simply structure the input prompt template as follows:

User: <*v*> What is <*o*> doing in this video?
Assistant: `<Answer>`

where we insert the object token corresponding to the referring target in the video with <*o*> in the instruction. We evaluate performance on the VideoRef45K benchmark [54], which comprises video question-answer data with box-level prompts in the first frame to specify the referring target. To encode object tokens, we apply SAM to the bounding box prompts to extract target masks.

We train two variants of models with a subset of VideoRef45K including data from VID-Sentence [11] (8K), HC-STVG [61] (10K) and LaSOT [17] (8K). For *VideoOrion-Ref*, we integrate this data into the MM instruction tuning stage with instruction following data from VideoLLaVA to train *VideoOrion*. For *VideoOrion-Ref-FT* and *VideoOrion-Ref-FT*+, we finetune the *VideoOrion* and *VideoOrion*+ with the referring data for 3 epochs, following [54]. We evaluate the models with metrics BLEU@4 [50], METEOR [5], ROUGE _L[38], CIDEr [65] and SPICE [2]. We compare the results with Artemis, a video-based referring model, and Merlin [75], a multi-frame-based referring model, in Table 3.

We can see that all our models outperform the baselines on all evaluation metrics. Notably, *VideoOrion-Ref* shows good zero-shot performance, with only a small amount of referring data involved in the MM instruction tuning stage. With additional finetuning following [54], *VideoOrion-Ref-FT* and *VideoOrion-Ref-FT*+ achieve significantly better results. This result validates that object tokens effectively encode accurate object semantics, enabling the model to identify the target object. Moreover, our approach equips Video-LLMs with a unified interface for improved general video understanding and referring capabilities.

Table 2. Performance comparison with the state-of-the-art Video-LLMs. All models except ShareGPT4Video use a 7B LLM backbone.

Model	Frame Number	MVBench Acc.	Egoschema Acc.	Perception-Test Acc.	Video-MME w/o / w subs	ActivityNet-QA Acc. / Score
LLaMA-VID[34]	1fps	41.9	38.5	44.6	25.9/ -	47.4/3.3
TimeChat[58]	-	38.5	33.0	-	-	-
Chat-UniVi[25]	64	-	-	-	40.6/45.9	46.1/3.3
LLaVA-NeXT-Video[83]	32	46.5	43.9	48.8	33.7/ -	53.5/3.2
ShareGPT4Video-8B[10]	16	51.2	-	-	39.9/43.6	-
VideoChat2[32]	16	60.4	54.4	47.3	39.5/43.8	49.1/3.3
Video-LLaVA[37]	8	41.0	38.4	44.3	39.9/41.6	45.3/3.3
VideoLLaMA[79]	8	34.1	-	-	-	12.4/1.1
VideoLLaMA2[14]	8	53.4	50.5	49.6	45.1/46.6	49.9/3.3
VideoOrion	8	**63.5**	**65.1**	**65.2**	**54.6/55.3**	**57.7/3.7**
Models with Qwen-2-7B LLM backbone						
LongVA[80]	64	-	-	-	52.4/ -	- /2.8
LLaVA-OneVision[28]	32	56.7	60.1	57.1	58.2/**61.5**	56.6/ -
VideoLLaMA2.1[14]	16	57.3	53.1	54.9	54.9/56.4	53.0/3.4
VideoOrion+	16	**67.4**	**65.0**	**65.9**	**58.9 / 61.5**	**60.3/3.7**

Table 3. Performance on the video referring task Qiu et al. [54].

Model	BLEU@4	METEOR	ROUGE_L	CIDEr	SPICE
Merlin[75]	3.3	11.3	26.0	10.5	20.1
Artemis[54]	15.5	18.0	40.8	53.2	25.4
VideoOrion-Ref	17.5	19.5	43.0	69.7	28.4
VideoOrion-Ref-FT	19.0	21.0	43.8	79.6	30.4
VideoOrion-Ref-FT+	**19.7**	**21.5**	**45.4**	**90.6**	**31.4**

4.4. Ablation Study

The objective of this section is to understand the effectiveness of each component and how they contribute to the improved performance of *VideoOrion*. Due to limited computation, we conduct all the ablation studies with a subset of the data used in *VideoOrion*. For Video-Centric Branch pertaining, we use 702K video-text pairs provided by Valley [42] and 558K image-text pairs provided by LLaVA [40]. We use our filtered 700k samples for the Object-Branch pertaining, and use 765K samples form Video-LLaVA[37] for the MM instruction tuning.

Object Tokens. To validate the effectiveness of Object-Centric Branch, we compare *VideoOrion* with baseline model VideoLLaMA2, with the same amount of data. Since the baseline model VideoLLaMA2 does not have the Object-Branch pertaining stage, the 700K data is added to the Video-Branch pertaining stage, for a fair comparison. As per Table 4, *VideoOrion* with object tokens consistently improves over the baseline on all the benchmarks, showing the effectiveness of explicit object-centric representation.

Table 4. Ablation study on the Object-Centric Branch.

Model	MVBench	Egoschema	Perception	VideoMME	ActNet
video-only	41.9	41.3	43.6	44.1	43.0
VideoOrion	**44.2**	**44.5**	**46.3**	**46.1**	**43.3**

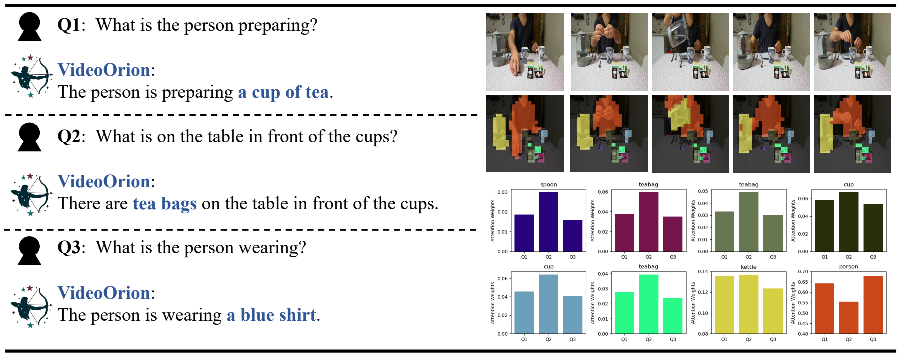
Design choices of detect-segment-track pipeline. Table 5 analyzes the design choices of the detect-segment-track pipeline. By default (Section 3.3), we use GroundingDINO (generic mode) for object proposals, RAM++ for adaptive segmenting the video, and XMem for tracking. To evaluate alternatives, we replace GroundingDINO (generic mode) with RAM++ and Mask2Former [12] for object proposal. For segmenting videos, we explore alternatives of segmenting videos uniformly into four parts or using the entire video without any segmentation. For tracking, we substitute Xmem with SAM2 as an alternative. As per Table 5, we show that our detect-segment-track pipeline remains robust across variations. Notably, uniform segmentation slightly underperforms RAM++, offering a trade-off between efficiency and performance. All configurations outperform the video-only baseline (Table 4), highlighting the strength of our object representation.

Design choices of Object Projector The Object Projector aggregates object dynamics captured by mask-pooled features. We explore various design choices, including a simple multi-layer perceptron (MLP), a single linear layer, average pooling, and two temporal modeling approaches—attention[64] and LSTM[22]. Although we hypothesized that temporal modeling would enhance object tokens, the results in Table 6 unexpectedly reveal that a simple linear layer performs just as effectively. Additionally, we observe that LSTM ranks second on average, with only a minimal gap compared to the MLP.

We further investigate the use of the DINOv2 vision encoder as an alternative to the CLIP encoder for encoding object features. Since DINOv2 is known for its ability to capture objectness, finer details, and low-level features, it could potentially enhance the modeling of object dynamics in videos, albeit at the cost of an additional vision encoder. However, as shown in the last row of Table 6, its performance falls short of CLIP, likely due to a lack of language-aligned semantics.

Choices	MVB	EGO	PER	V-MME	ACT	Avg.
how to segment the video?						
RAM++	**44.2**	**44.5**	46.3	46.1	43.3	**44.9**
no-split	43.8	41.2	45.3	**47.4**	41.9	43.9
uniform	43.6	43.0	45.3	**47.4**	**44.5**	44.7
object proposals						
generic	**44.2**	**44.5**	46.3	46.1	43.3	**44.9**
M2F	43.2	41.6	43.6	46.4	42.3	43.4
RAM++	**44.2**	42.9	45.5	45.2	42.9	44.1
tracking model						
Xmem	**44.2**	**44.5**	46.3	46.1	43.3	**44.9**
SAM2	44.0	41.8	**46.7**	46.3	43.3	44.4

Table 5. Ablation study on design choices of the detect-segment-track pipeline. *no-split* refers to not segmenting the video at all; *uniform* refers to uniformly sampling 4 frames as key frames; *M2F* refers to using Mask2Former as object proposer. We color the default choice in grey.



Q1: What is the person preparing?

VideoOrion:
The person is preparing **a cup of tea**.

Q2: What is on the table in front of the cups?

VideoOrion:
There are **tea bags** on the table in front of the cups.

Q3: What is the person wearing?

VideoOrion:
The person is wearing **a blue shirt**.

Figure 3. Case studies showing how *VideoOrion* utilizes object tokens to generate responses based on different instructions.

Table 6. Performance of *VideoOrion* with different Object Projectors. We explore mlp (default), attention, linear, lstm layers and plain average pooling.

Architecture	MVBench	Egoschema	Perception	VideoMME	ActNet	Avg.
mlp (ours)	48.0	**51.5**	46.9	**45.6**	41.6	**46.73**
Attention	48.8	45.9	47.6	44.3	41.0	45.51
Linear	47.3	47.2	46.1	44.1	**42.9**	45.52
avg pooling	47.3	47.5	46.0	45.4	41.9	45.61
LSTM	**49.5**	47.0	**47.8**	45.1	**42.9**	46.45
DINOv2-L	44.1	41.7	45.1	45.8	41.5	43.6

4.5. Case Study

This section presents a case study demonstrating how *VideoOrion* utilizes object tokens. Given an input video (Figure 3), we ask three different questions and visualize the corresponding changes in attention weights from the last decoder layer of *VideoOrion*'s LLM backbone. The attention weights are normalized so that the sum across all eight object tokens equals one, and we also display the object masks extracted through the detect-segment-track pipeline. The charts illustrate that attention to object tokens varies with different input questions, with higher attention weights assigned to tokens corresponding to more relevant objects. For instance, the object token for the *person* (last chart) receives the least attention in $Q2$ as it is irrelevant to the question. In $Q1$, attention increases since the question mentions *person*, but it peaks in $Q3$, which focuses solely on the person's features. Conversely, tokens for teabags and cups gain significantly higher attention in $Q2$, directly contributing to the answer. Taking another perspective, in $Q1$, which concerns the video's general content, attention weights obtained by object tokens are in the middle among the three questions. For the other two questions, which focus on details, attention shifts more toward relevant objects. This suggests that *VideoOrion* adapts its focus dynamically, demonstrating that object tokens effectively enhance video comprehension.

5. Conclusion

We propose *VideoOrion*, a novel Video-LLM, with explicit disentangled representation for object dynamics in the video. *VideoOrion* has a Video-Centric Branch and an Object-Centric Branch, through a detect-segment-track pipeline with an Object Projector to extract and aggregate the object tokens. Empirical results across multiple benchmarks demonstrate the capability of *VideoOrion* for improved general video understanding and the inherent ability of referring tasks in videos.

Limitations

Despite its effectiveness, our method has certain limitations. The detect-segment-track pipeline relies on multiple vision models, introducing additional computational costs (see Appendix F for a detailed analysis) and potentially leading to inaccurate mask extraction, particularly for low-quality videos (see Appendix E.1 for a detailed analysis). However, the explicit and disentangled object representation allows the opportunities to diagnose and interpret the mistakes (see Appendix E.2 for a detailed analysis). Moreover, we believe future advancements in vision models could mitigate these issues, and this work serves as a proof-of-concept that explicit disentangled object representation can enhance general video understanding with inherent referring abilities. Additionally, our framework still relies on the Video-Centric Branch for contextual information, and the alignment between the two branches remains an open area for further investigation.

Acknowledgments

This work was supported by NSFC under Grant 62450001. The authors would like to thank the anonymous reviewers for their valuable comments and advice.

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. In *NeurIPS*, 2022. 2

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 6

[3] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024. 2

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1

[5] Satanjeev 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*, 2005. 6

[6] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 5

[7] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. Temporalbench: Towards fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024. 13

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[9] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023. 1

[10] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*, 2024. 7

[11] 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. 6

[12] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 3, 7

[13] Ho Kei Cheng and Alexander G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 3, 5

[14] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2, 4, 5, 7

[15] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl. Based Syst.*, 2016. 3

[16] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[17] 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 *CVPR*, 2019. 6

[18] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 6

[19] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 5

[20] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, 2024. 5

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 7

[23] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-set image tagging with multi-grained text supervision. *arXiv e-prints*, 2023. 5

[24] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 4, 6

[25] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024. 3, 7

[26] Yang Jin, Zhicheng Sun, Kun Xu, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang,

Yang Song, Kun Gai, and Yadong Mu. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. In *ICML*, 2024. 2, 3

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *ICCV*, 2023. 4

[28] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 7

[29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2, 4

[30] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2

[31] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *ICCV*, 2023. 5, 13

[32] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024. 2, 5, 6, 7

[33] Yijiang Li, Wentian Cai, Ying Gao, Chengming Li, and Xiping Hu. More than encoder: Introducing transformer decoder to upsample. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 1597–1602. IEEE, 2022. 3

[34] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023. 2, 7

[35] Yijiang Li, Xinjiang Wang, Lihe Yang, Litong Feng, Wayne Zhang, and Ying Gao. Diverse cotraining makes strong semi-supervised segmentor. *arXiv preprint arXiv:2308.09281*, 2023. 3

[36] Yijiang Li, Qingying Gao, Tianwei Zhao, Bingyang Wang, Haoran Sun, Haiyun Lyu, Robert D Hawkins, Nuno Vasconcelos, Tal Golan, Dezhi Luo, et al. Core knowledge deficits in multi-modal language models. In *Forty-second International Conference on Machine Learning*, 2025. 1

[37] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 5, 7

[38] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004. 6

[39] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 4

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 5, 7

[41] 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. 3, 4

[42] 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. *arXiv preprint arXiv:2306.07207*, 2023. 7

[43] Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, 2024. 2, 4, 6

[44] Gianluca Mancusi, Aniello Panariello, Angelo Porrello, Matteo Fabbri, Simone Calderara, and Rita Cucchiara. Trackflow: Multi-object tracking with normalizing flows. In *ICCV*, 2023. 3

[45] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023. 6

[46] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, 2022. 3

[47] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024. 5

[48] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[49] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3

[50] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[51] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adrià Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video models. In *NeurIPS*, 2023. 6

[52] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 3

[53] Zheng Qin, Le Wang, Sanping Zhou, Panpan Fu, Gang Hua, and Wei Tang. Towards generalizable multi-object tracking. In *CVPR*, 2024. 3

[54] Jihao Qiu, Yuan Zhang, Xi Tang, Lingxi Xie, Tianren Ma, Pengyu Yan, David Doermann, Qixiang Ye, and Yunjie Tian. Artemis: Towards referential understanding in complex videos. *arXiv preprint arXiv:2406.00258*, 2024. 2, 3, 6, 7

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3, 4, 6

[56] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020. 4

[57] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 4

[58] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 7

[59] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NIPS*, 2022. 5

[60] Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. Audio-visual llm for video understanding. *arXiv preprint arXiv:2312.06720*, 2023. 1

[61] 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*, 2021. 6

[62] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[63] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NIPS*, 2022. 13

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 2017. 7

[65] 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*, 2015. 6

[66] Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang. Elysium: Exploring object-level perception in videos via mllm. In *ECCV*, 2024. 2

[67] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 3

[68] Xinjiang Wang, Xingyi Yang, Shilong Zhang, Yijiang Li, Litong Feng, Shijie Fang, Chengqi Lyu, Kai Chen, and Wayne Zhang. Consistent-teacher: Towards reducing inconsistent pseudo-targets in semi-supervised object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3240–3249, 2023. 3

[69] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2023. 5

[70] Jiaqi Xu, Cuiling Lan, Wenxuan Xie, Xuejin Chen, and Yan Lu. Slot-vlm: Slowfast slots for video-language modeling. *arXiv preprint arXiv:2402.13088*, 2024. 2, 3

[71] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 4, 6

[72] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021. 3

[73] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pretraining for open-world detection. In *NeurIPS*, 2022. 3

[74] 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*, 2023. 4

[75] En Yu, Liang Zhao, Yana Wei, Jinrong Yang, Dongming Wu, Lingyu Kong, Haoran Wei, Tiancai Wang, Zheng Ge, Xiangyu Zhang, et al. Merlin: Empowering multimodal llms with foresight minds. In *ECCV*, 2025. 2, 6, 7

[76] 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. In *AAAI*, 2019. 6

[77] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 6

[78] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023. 3

[79] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023. 1, 2, 4, 7

[80] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 7

[81] Wanpeng Zhang, Zilong Xie, Yicheng Feng, Yijiang Li, Xingrun Xing, Sipeng Zheng, and Zongqing Lu. From pixels to tokens: Byte-pair encoding on quantized visual modalities. *arXiv preprint arXiv:2410.02155*, 2024. 1

[82] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, 2022. 3

[83] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 5, 7

[84] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024. 2

[85] 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. 3