Elysium: Exploring Object-level Perception in Videos via MLLM

Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang

Bytedance Inc.

Abstract. Multi-modal Large Language Models (MLLMs) have demonstrated their ability to perceive objects in still images, but their application in video-related tasks, such as object tracking, remains understudied. This lack of exploration is primarily due to two key challenges. Firstly, extensive pretraining on large-scale video datasets is required to equip MLLMs with the capability to perceive objects across multiple frames and understand inter-frame relationships. Secondly, processing a large number of frames within the context window of Large Language Models (LLMs) can impose a significant computational burden. To address the first challenge, we introduce **ElysiumTrack-1M**, a large-scale video dataset supported for three tasks: Single Object Tracking (SOT), Referring Single Object Tracking (RSOT), and Video Referring Expression Generation (Video-REG). ElysiumTrack-1M contains 1.27 million annotated video frames with corresponding object boxes and descriptions. Leveraging this dataset, we conduct training of MLLMs and propose a token-compression model **T-Selector** to tackle the second challenge. Our proposed approach, **Elysium**: Exploring Object-level Perception in Videos via MLLM, is an end-to-end trainable MLLM that attempts to conduct object-level tasks in videos without requiring any additional plug-in or expert models. All codes and datasets are released at https://github.com/Hon-Wong/Elysium.

Keywords: Multi-modal Large Language Models · Object Tracking · Referring Single Object Tracking

1 Introduction

In recent years, Multi-modal Large Language Models [13,35,40,41,50,65,82] have gained significant attention and proven effective in various multi-modal tasks. These models [4, 13, 14, 65, 80] have showcased their ability to excel in tasks like Image Grounding [28,45] and Object Detection [39,58], demonstrating their capacity to handle object-level tasks in static images, incorporating an adapter that bridges a visual encoder with a Large Language Model (LLM) [15, 61]. However, when it comes to video scenes, certain challenges arise due to the complexity of temporal information and the abundance of motion cues, making it challenging for a unified MLLM to effectively address diverse object-level tasks in videos.

Specifically, we classify video tasks into three categories based on the granularity they address: 1) video-level tasks (*e.g.*, VideoQA [70, 77] Video Caption [11,71]), 2) frame-level tasks (*e.g.*, Video Grounding [32], Dense Video Captioning [30], and Video Highlight Detection [32]), and 3) object-level tasks (*e.g.*, Single Object Tracking (SOT) [7,34], Multi-Object Tracking (MOT) [8,67], and Video Object Segmentation (VOS) [51, 72]). Video-level tasks are primarily focused on capturing global information within a video. Previous studies [36,38,43, 44,78] have demonstrated the effectiveness of adapter architectures that employ fusion operations on the temporal axis, extracting features from all video frames, reducing the number of visual tokens at the expense of the capability for individual frame analysis. In contrast, frame-level tasks necessitate the model to differentiate and analyze each frame independently, emphasizing temporal awareness. Recent research [23, 55, 59] has explored various tasks, including Video Grounding, Dense Video Captioning, and Video Highlight Detection. These endeavors have focused on leveraging temporal-aware token compression models to extend the context frames and capture longer-term temporal dependencies. A step forward, object-level tasks require the model to differentiate and locate objects in each frame, ensuring temporal consistency with higher granularity. This capability is crucial for accurate trajectory tracking across multiple frames despite interference like occlusion and motion blur. Such tasks demand a sophisticated architecture that effectively distinguishes and tracks object identities over time while minimizing the use of visual tokens to achieve a larger context window. Moreover, training models for object-level tasks faces challenges due to the limited availability of large-scale training data, making it a challenging task for MLLMs to handle.

In this paper, we introduce Elysium, an end-to-end trainable MLLM that is capable of handling both global-level and object-level tasks in the video field. To address the issue of limited training data, we construct a large-scale dataset named **ElysiumTrack-1M**, which includes abundant trajectories and object descriptions to support tasks like Single Object Tracking (SOT), RSOT, and Video-REG. ElysiumTrack-1M is derived from the WebVid-10M dataset [5], and we employ a carefully designed processing pipeline to reduce noise in the generated labels of each video. To enable the MLLM to distinguish individual frames while reducing visual token use, we introduce a visual token compression network called **T-Selector**. This network offers a trade-off between performance and visual token consumption. We leave the exploration of additional object-level tasks, such as Multi-Object Tracking, Video Object Detection, Video Object Segmentation, and Referring Video Object Segmentation (RVOS) [21, 29, 57] as future work. Our contributions can be summarized in three key aspects:

1). We construct **ElysiumTrack-1M**, a million-scale dataset for object-level tasks in videos, supporting existing tasks like SOT and introducing benchmarks for RSOT and Video-REG.

2). We introduce **Elysium**, an end-to-end trainable MLLM, equipped with a carefully designed token compression network named **T-selector**. This ap-

proach extends the object-perception capabilities of MLLMs to encompass multiple frames, specifically videos.

3). Extensive experiments have shown the effectiveness of Elysium in downstream tasks such as Image Grounding, Video QA, SOT, RSOT, and Video-REG.

2 Related Works

2.1 MLLMs

In recent years, Multimodal Large Language Models have made remarkable advancements in a wide range of multi-modal tasks. Vision-language models [35,40], in particular, have achieved significant success in tasks such as image captioning and image question answering. These models have demonstrated their ability to understand the visual content of images and videos and follow the instructions given by humans. This is achieved through the use of a lightweight adapter that connects the visual encoder with the language model. Typically, these models are pretrained on large-scale datasets and then finetuned on downstream task-aware datasets. Furthermore, researchers have extended the capabilities of MLLMs [13,14,50,65,80] to tackle object-level tasks in images, *e.g.*, Object Detection and Image Grounding. These tasks require MLLMs to provide object coordinates as outputs, enabling accurate localization and identification of specific objects. However, there has been a relative lack of research in the object-level tasks in videos, *e.g.,* object tracking, except early exploration [49] that employs multiple external expert models. Therefore, in this paper, we aim to extend the research to the domain of videos using an MLLM only.

2.2 Object-level Tasks in Videos

In the domain of videos, there exist numerous object-level tasks, including Single Object Tracking, Multi-Object Tracking, Video Object Detection, Video Object Segmentation, and Referring Video Object Segmentation. Among these, Single Object Tracking stands as a fundamental task aimed at predicting the location of a specific object in consecutive frames by referencing its initial position in the first frame. Existing models [7,16,22,33,73] commonly employ two inputs: a template extracted from the region of interest in the first frame and a dynamic search area that adjusts based on the tracking result from the previous frame. Typically, the search area takes the form of a larger square area centered around the predicted tracking box of the previous frame. This approach effectively narrows down the region where the object is likely to be located in subsequent frames. To enhance tracking performance, some techniques involve periodically updating the template [16,81] to adapt to appearance changes, while others utilize a cosine window [33,73] to constrain the heatmap and improve localization accuracy. However, these methods often require handcrafted parameter tuning to adapt to various scenes with different frame rates or fast-moving objects. Moreover, Referring Video Object Segmentation [29] is a task that leverages

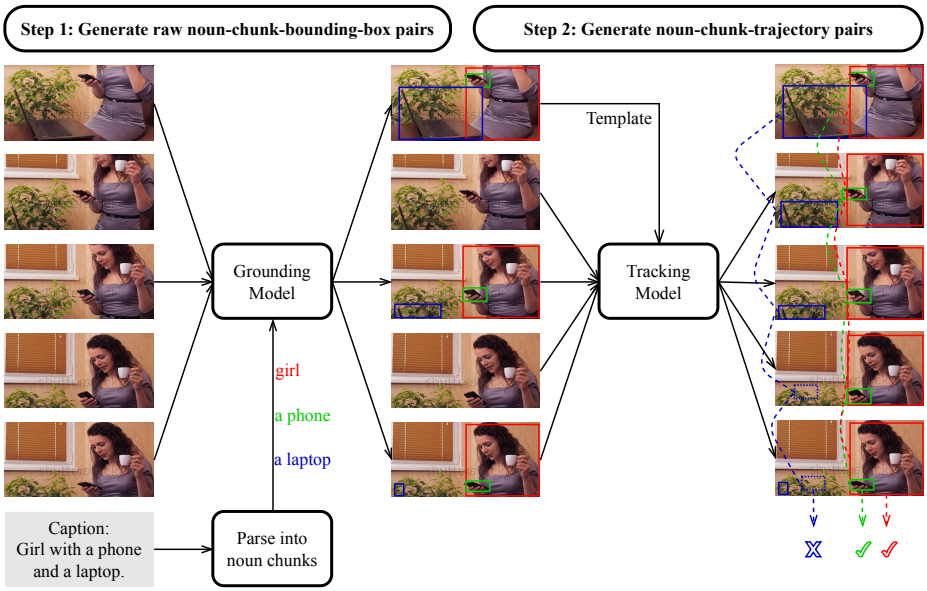

Fig. 1: The pipeline to construct ElysiumTrack-1M dataset.

language to locate and segment objects within videos, thereby combining visual and linguistic modalities to achieve precise object localization and segmentation. These object-level tasks play crucial roles in video analysis and understanding, enabling various applications.

3 Construct ElysiumTrack-1M dataset

In this section, we will give a clear definition about two tasks: RSOT and Video-REG. We will provide details on constructing the ElysiumTrack-1M dataset and discuss its scale and the tasks it can support. Additionally, we will address the evaluation of model performance and the metrics used on the ElysiumTrack-1M dataset.

3.1 Introduce RSOT and Video-REG Tasks

RSOT. Motivated by Referring Video Object Segmentation, we define the task of RSOT as identifying and locating a specific object within an entire video by the given language expression only (no positional cue is used). This task offers a more flexible format for tracking compared to traditional Single Object Tracking methods and establishes a valuable connection between language and tracking.
Video-REG. In addition to RSOT, we further extend the concept of Referring Expression Generation (REG) to the field of videos. We define this task as Video-REG, which involves predicting a description of an object given its coordinates

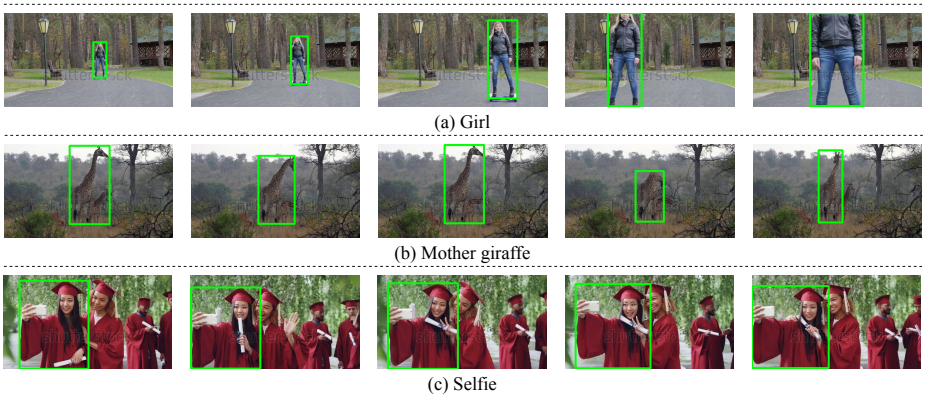

Fig. 2: Samples from ElysiumTrack-1M dataset.

Table 1: Dataset comparisons on the aspects of 1) number of trajectories. 2) number of expressions. 3) total duration.

Datasets	#Trajectories	#Expressions	Duration
OTB15 [69]	51	0	16.4 minutes
VOT14 [2]	25	0	5.7 minutes
VOT16 [56]	60	0	11.9 minutes
VOT17 [3]	60	0	11.9 minutes
UAV20L [47]	20	0	32.6 minutes
UAV123L [47]	91	0	1.1 hours
GOT-10K [24]	10K	0	1.7 days
LaSOT [18]	1.4K	1.4K	1.4 days
TrackingNet [48]	30.6K	0	5.6 days
ElysiumTrack-1M	**1.27M**	**1.27M**	**9.2 months**

in any frame of a video. Unlike conventional REG tasks, Video-REG demands the model to possess temporal awareness. This is because the appearance of the object in the current frame might be affected by occlusion or motion blur, but it can be identified in other frames provided. Thus, the model needs to consider the temporal context to generate precise descriptions in Video-REG.

3.2 Dataset Construction

We present ElysiumTrack-1M, a million-scale object perception video dataset, designed to support SOT, RSOT, and Video-REG tasks. As depicted in Figure 1, the dataset construction pipeline consists of two essential steps: 1) Generating raw noun-chunk-bounding-box pairs and 2) Extending noun-chunk-bounding-box pairs to noun-chunk-trajectory pairs:

Step 1: Generating raw noun-chunk-bounding-box pairs. Similar to prior

research [50], we employ spaCy [46] to parse the video captions in the WebVid-10M [5] dataset into noun chunks. To minimize noise, we remove chunks that contain virtual words (e.g., time, love, wind) or plural words (e.g., family, people, two dogs). Next, we utilize a pretrained grounding model, Grounding DINO [42] to generate bounding boxes for each noun chunk in the first frame, middle frame, and last frame of a video. We retain noun-chunk-bounding-box pairs with predicted confidence scores higher than 0.6.

Step 2: Extending noun-chunk-bounding-box pairs to noun-chunk-trajectory pairs. We utilize a pretrained Tracking Model (i.e., MixFormer [16]) to generate trajectories given the bounding boxes in the first frame, thus extending raw noun-chunk-bounding-box pairs to raw noun-chunk-trajectory pairs. We retain noun-chunk-trajectory pairs that achieve a tracking confidence score higher than 0.8 consistently across all frames in a video. We also utilize the Kalman Filter [8, 67] to discard trajectories with extreme drift. Subsequently, we calculate the Intersection over Union (IoU) score between the grounding box and the tracking box in the middle frame and the last frame. Cases where either of these scores is lower than 0.3, indicating potential tracker drift, are discarded to ensure the removal of inaccurate tracking instances.

In the end, 1.27 million noun-chunk-trajectory pairs in total are generated for ElysiumTrack-1M dataset, samples are shown in Figure 2. The whole process takes about 6 days on 24 A100-80G GPUs. As shown in Table 1, it contains a significantly larger number of trajectories compared to existing tracking datasets, with each trajectory accompanied by an expression that refers to the corresponding object. The dataset is split into two subsets: around 1.27 million videos are used for training, while 500 videos are retained for evaluation. For the evaluation split, we manually check each case and ensured that the referring object in the initial frame of each video is unique.

3.3 Evaluations

For evaluation purposes on RSOT and SOT, we utilize the widely adopted One-Pass Evaluation (OPE) strategy as described in [68]. We use commonly used metrics such as success and precision, which serve as the evaluation criteria for both traditional SOT tasks and the proposed RSOT task on the ElysiumTrack-1M dataset. By employing these established metrics, we ensure a consistent and standardized evaluation process, enabling fair comparisons and accurate assessments of tracking performance across different tasks. For Video-REG tasks, we use Meteor [6] and CIDEr [62] for evaluation following [50].

4 Elysium

4.1 Architecture

Overall architecture. The primary goal of architectural designs in the context of MLLMs encompasses two key aspects. Firstly, it aims to enhance the

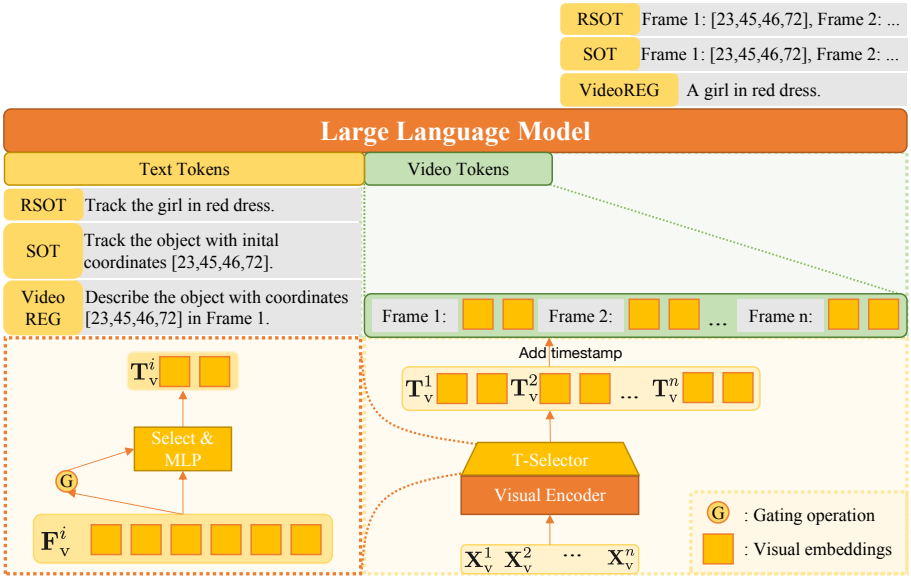

Fig. 3: The architectures of Elysium. Elysium combines a visual encoder, an LLM, and a T-Selector to connect the visual encoder with the LLM.

MLLMs' ability to tackle object-level tasks in videos, such as object tracking. This entails addressing challenges related to incorporating temporal information and effectively handling inter-frame relationships within the MLLM framework. Secondly, architectural designs need to strike a balance between minimizing the number of tokens consumed by visual features and optimizing the overall performance of the MLLM on downstream tasks. Thus, we choose CLIP-ViT-L [52] as our visual encoder, Vicuna [15] as our LLM, and a specially designed token compressor (T-selector) to connect visual encoder and LLM.

As shown in Figure 3, for each frame $\mathbf{X}_\mathrm{v}^i$ in a video, we utilize a visual encoder to obtain its feature $\mathbf{F}_\mathrm{v}^i \in \mathbb{R}^{N\times C}$. Subsequently, a token-compressing model, denoted as T-Selector, compresses $\mathbf{F}_\mathrm{v}^i \in \mathbb{R}^{N\times C}$ into $\mathbf{T}_\mathrm{v}^i \in \mathbb{R}^{\alpha N\times D}$, where $\alpha \in (0, 1]$ is a hyperparameter that indicates the compression ratio of the token count, and D represents the hidden dimension of the Large Language Model (LLM). The process can be expressed as follows:

$$\begin{aligned} \mathbf{F}_\mathrm{v} &= g\left(\mathbf{X}_\mathrm{v}\right), \mathbf{F}_\mathrm{v} \in \mathbb{R}^{N\times C}; \\ \mathbf{T}_\mathrm{v} &= h\left(\mathbf{F}_\mathrm{v}\right), \mathbf{T}_\mathrm{v} \in \mathbb{R}^{\alpha N\times D}, \end{aligned} \tag{1}$$

where g indicates the visual encoder (*i.e.*, CLIP-ViT-L) and h is the T-Selector.

T-Selector. Given the hypothesis that videos contain redundant information [20, 60], our research focuses on investigating effective methods to compress visual tokens. The goal is to enable the MLLM to handle a larger number of frames without suffering a significant drop in performance. We have explored

various visual token compression techniques, including commonly used architectures such as one-layer cross-attention and concatenation. Notably, we have observed that fusion operations on the spatial dimension result in a drastic performance decline. To mitigate this issue, we propose a novel approach called T-selector. The T-selector aims to strike a balance between the visual token count and performance. It comprises two parts: a gating operation conducted by an MLP layer and a softmax layer, responsible for determining which tokens to select, and an MLP layer that transforms the hidden dimension to match that of the LLM. The gating process can be written as:

$$\begin{aligned} \mathbf{G}_{\mathrm{v}} &= \mathrm{KeepTopK}(\mathrm{Softmax}(\mathrm{MLP}(\mathbf{F}_{\mathrm{v}})), k, \mathbf{F}_{\mathrm{v}}), \\ \mathbf{T}_{\mathrm{v}} &= \mathrm{MLP}(\mathbf{G}_{\mathrm{v}}), \end{aligned} \tag{2}$$

where $k = \alpha N$, indicating the compression ratio of T-Selector. With a gating operation denoted as KeepTopK, we select the tokens with scores among the top k, the selected tokens are represented as $\mathbf{G}_{\mathrm{v}}$. This selection strategy helps to reduce the number of tokens while maintaining the most relevant and informative ones according to their scores. Visual tokens after MLP layer are denoted as $\mathbf{T}_{\mathrm{v}}$.

Input & output format. As depicted in Figure 3, we have incorporated timestamps into each visual token $\mathbf{T}_{\mathrm{v}}^{i}$ to enable Elysium to differentiate between consecutive frames. To optimize token use, we represent a location simply using the coordinates of the top-left and bottom-right corners. Each value is in the range of $[0, 100)$, separated by a comma without extra space. For instance, a coordinate like "[23,45,46,72]" requires 13 tokens only in LLaMA [61] tokenizer, which is much fewer compared to "[0.231, 0.452, 0.458, 0.783]" used by Shikra [14] (28 tokens). Additionally, we have devised distinct prompts tailored to different types of questions and expanded the question set to include multiple queries for each task. This approach enhances Elysium's robustness when dealing with diverse question formats.

4.2 Training Setups

Datasets Preparation. Following previous works [4, 13, 14, 50], we use large-scale data to train our model. We classify the datasets we use into three classes: 1) Image data: LLaVA-558K [41], GRIT-20m [50], CogVLM-SFT-311K [65], COCOVQA [54], GQA [25], Visual7W [83], Flickr [76], VisualGenome [31], RefCOCO [28], RefCOCO+ [28], RefCOCOg [45]; 2) Data for video-level tasks: VideoChat [36]; 3) Video data for object-level tasks: ElysiumTrack-1M.

To achieve comprehensive training for an MLLM, we employ progressive training strategies. The overall training process involves two distinct stages: pretraining and finetuning stage.

Stage 1: Pretraining on large-scale image data. For pretraining, we exclusively utilize image data due to their smaller token count, resulting in faster training speeds. To ensure stable training, we adopt a two-step process. Initially, we employ LLaVA-558K to exclusively train the T-Selector. During this phase, the ViT and LLM components remain frozen. The learning rate is set to 2×10^{-3},

the batch size is 32, and the training is performed on 8 GPUs. This step aims to initialize the T-Selector's parameters, enhancing the stability of subsequent training procedures. Following the T-Selector training, we proceed to train Elysium end-to-end, unfreezing all parameters. This training involves a mixture of image data and is conducted for 30,000 steps on 32 GPUs. The learning rate is set to 5×10^{-5}, and the batch size is 16.

Stage 2: Finetuning on high-quality data. After the pretraining phase, we proceed to train Elysium using high-quality data to enhance its performance in object-level tasks and downstream applications like image grounding. This training involves a mixture of datasets comprising high-quality image data and video data, *i.e.*, VideoChat, ElysiumTrack-1M, and all the image datasets mentioned above except GRIT-20M and LLaVA-558K. To optimize the training process, we conduct training for 22,000 steps on 32 GPU cards, utilizing a learning rate of 5×10^{-5} and a batch size of 16. For the video dataset, we adopt a random sampling approach, randomly selecting 2 to 8 frames per video with a random interval ranging from 1 to 60. This selection process helps simulate varying frame rates and motion speeds. This configuration is used for the first 20,000 steps. In the subsequent 2,000 steps, we extend the frame count to 32 frames per video, utilizing a batch size of 1. By employing this training methodology, we aim to optimize the performance of Elysium in object-level tasks and ensure its adaptability to different video scenarios with varying frame rates and motion speeds.

During the training process, we resize the images to a size of 336×336 without padding. Data augmentations are not applied. To optimize training, we utilize a cosine learning rate schedule. Additionally, we employ deepspeed [53] to leverage larger batch sizes, enhancing computational efficiency. All experiments are conducted on A100-80G GPUs.

4.3 Evaluation Setups

For video-level tasks like VideoQA, we uniformly sample 16 frames from a video for inference. However, for object-level tasks such as RSOT and SOT, if the video has more than 32 frames, we divide it into smaller clips. Each clip consists of 8 frames, with an overlap of one frame in the preceding and subsequent clips. Thus, by utilizing the predicted location of the last frame in the previous clip, we can initialize the tracking process in the subsequent clip. This process aligns with the commonly used updating template trick in SOT.

5 Experiments

5.1 State-of-the-art Comparisons

In this section, we assess the capabilities of our proposed model, Elysium. All the experiments are conducted using the checkpoint after the second stage of training without any further finetuning. Firstly, we evaluate Elysium's object-level perception ability in images by examining its performance on image grounding

Table 2: Results on Referring Expression Comprehension. "*Res.*" indicates the resolution of the input image. "*T.*" denotes the visual token count per image. "*" represents this method utilizes a much larger ViT [19] than CLIP-ViT-L.

Model			RefCOCO			RefCOCO+			RefCOCOg	
Model	*Res.*	*T.*	val	test-A	test-B	val	test-A	test-B	val	test
OFA-L [64]	-	-	79.96	83.67	76.39	68.29	76.00	61.75	67.57	67.58
VisionLLM-H [66]	-	-	-	86.70	-	-	-	61.75	-	-
Shikra (7B) [14]	224	256	87.01	90.61	80.24	81.60	87.36	72.12	82.27	82.19
Shikra (13B) [14]	224	256	87.83	91.11	81.81	**82.89**	87.79	74.41	82.64	83.16
MiniGPT-v2 (7B)* [13]	448	256	88.69	91.65	**85.33**	79.97	85.12	74.45	**84.44**	84.66
Ferret (7B) [75]	336	608	87.49	91.35	82.45	80.78	87.38	73.14	83.93	**84.76**
GroundingGPT (7B) [37]	336	576	88.02	91.55	82.47	81.61	87.18	73.18	81.67	81.99
Elysium (*ours*, 7B)	336	108	**89.07**	**92.12**	84.95	82.86	**88.93**	**75.64**	82.92	83.62

Table 3: Results on VideoQA. Our Elysium achieves state-of-the-art performance on the public datasets.

Model	MSVD-QA		MSRVTT-QA		TGIF-QA		ActivityNet-QA	
	Acc.	Score	Acc.	Score	Acc.	Score	Acc.	Score
FrozenBiLM [74]	32.2	-	16.8	-	41.0	-	24.7	-
VideoLLaMA [78]	51.6	2.5	29.6	1.8	-	-	12.4	1.1
LLaMA-Adapter [79]	54.9	3.1	43.8	2.7	-	-	34.2	2.7
VideoChat [36]	56.3	2.8	45.0	2.5	34.4	2.3	26.5	2.2
VideoChatGPT [44]	64.9	3.3	49.3	2.8	51.4	3.0	35.2	2.7
Valley-v3 [43]	60.5	3.3	51.1	2.9	-	-	45.1	3.2
MovieChat [59]	75.2	**3.8**	52.7	2.6	-	-	**45.7**	**3.4**
Elysium (*ours*)	**75.8**	3.7	**67.5**	**3.2**	**66.6**	**3.6**	43.4	2.9

tasks. Secondly, we gauge its proficiency in handling video-level tasks by testing its performance on VideoQA datasets. Lastly, we evaluate Elysium's object-level perception ability in videos through tasks such as Single Object Tracking. By conducting evaluations across these different task types, we aim to comprehensively assess the effectiveness of Elysium in various domains and levels of perception.

Image Grounding. Image Grounding is a task that necessitates the model to produce the corresponding coordinates based on a given language expression. We evaluate the performance of our model, Elysium, on commonly used datasets such as RefCOCO [28], RefCOCO+ [28], and RefCOCOg [45] datasets. The performance results of Elysium are presented in Table 2. Despite employing a visual token compression method and representing images with fewer tokens, Elysium achieves state-of-the-art performance. This outcome validates the effectiveness of our token compression network in maintaining high performance while reducing token use.

Table 4: Results on Single Object Tracking. Our Elysium conducted a **zero-shot** experiment on the tracking datasets without any finetuning. All other methods are finetuned on corresponding datasets.

Model	LaSOT			UAV123	
	AUC	P	P_{Norm}	AUC	P
SiamFC [7]	33.6	42.0	33.9	48.5	69.3
SiamRPN [34]	-	-	-	52.7	74.8
MDNet [27]	39.7	46.0	37.3	52.8	-
SiamRPN++ [33]	49.6	56.9	49.1	61.0	80.3
DiMP [9]	**56.9**	**65.0**	**56.7**	**65.4**	-
MAML [63]	52.3	-	-	-	-
SiamFC++ [73]	54.4	62.3	54.7	-	-
CGACD [17]	51.8	62.6	-	63.3	83.3
SiamGAT [22]	53.9	63.3	53.0	64.6	**84.3**
Elysium (*ours, zero-shot*)	56.1	61.0	50.1	56.6	79.2

Table 5: The results of the RSOT and SOT tasks on the evaluation split of ElysiumTrack-1M dataset are presented. As a baseline method, we employ MiniGPT-v2 to perform REC tasks on each frame of the videos.

Model	Task	AUC	P	P_{Norm}
MiniGPT-v2 [13]	RSOT	65.4	70.1	67.4
Elysium (*ours*)	RSOT	87.5	94.5	93.7
Elysium (*ours*)	SOT	**88.7**	**94.6**	**93.8**

Zero-Shot VideoQA Evaluation. Following previous protocols [36, 43, 44], we evaluate Elysium's video question-answering ability on multiple publicly available datasets, *i.e.,* MSVD-QA [12], MSRVTT-QA [70], TGIF-QA [26], and ActivityNet-QA [77]. To ensure a fair evaluation, we employ a zero-shot approach and utilize GPT-assisted evaluation techniques [44]. This evaluation process involves assessing the accuracy of the model's generated predictions and assigning a relative score on a scale of 1 to 5. We aim to comprehensively evaluate and compare the performance of Elysium across different video question-answering datasets. As depicted in Table 3, our Elysium achieves state-of-the-art performance.

Zero-Shot SOT Evaluation. We perform zero-shot evaluations on various SOT datasets using our Elysium model. As illustrated in Figure 4, Elysium demonstrates comparable performance to baseline methods, even in a zero-shot setting. However, we observe that Elysium's performance is relatively less satisfactory when dealing with datasets containing small objects, such as UAV123. This might be attributed to the limited resolution of the visual encoder we use.

Evaluation on ElysiumTrack-1M. As shown in Table 5 and Table 6, we evaluate the performance of RVOT, VOT, and VideoREG on the proposed dataset.

Table 6: The results of the Video-REG tasks on the evaluation split of ElysiumTrack-1M dataset. As a baseline, we employ MiniGPT-v2 to perform REG tasks on each given frame.

Model	**Meteor**	**CIDEr**
MiniGPT-v2 [13]	16.9	76.6
Elysium (*ours*)	**46.7**	**115.3**

Table 7: Ablation studies on the architecture of visual token compression network. *Concat.* and *C.A.* stands for concatenation and cross-attention operation, respectively. *None* represents not using any compression model.

Architectures			**RefCOCO**			**RefCOCO+**			**RefCOCOg**		**Avg.**
Model	Res.	αN	val	test-A	test-B	val	test-A	test-B	val	test	
None	224	256	80.67	88.47	73.11	74.17	83.41	63.55	74.84	76.26	76.81
None	336	576	**86.19**	**91.27**	**79.92**	**78.17**	**85.46**	**69.28**	**80.02**	**81.28**	**81.45**
Concat.	336	144	59.29	68.29	50.28	50.85	60.30	42.28	52.76	53.18	54.65
C.A.	336	144	52.24	55.52	50.40	37.23	40.85	32.81	40.52	40.60	49.23
T-Selector	336	1	51.57	54.01	50.35	37.31	40.28	35.25	39.77	41.05	43.70
T-Selector	336	36	81.39	85.47	76.30	70.84	76.12	61.91	71.59	72.20	74.48
T-Selector	336	72	82.93	85.82	78.85	72.86	77.06	65.26	73.65	73.91	76.29
T-Selector	336	108	84.98	88.24	80.15	75.77	79.27	67.41	76.29	76.21	78.54
T-Selector	336	144	84.77	88.42	79.56	74.80	80.61	66.98	76.08	76.36	78.45
T-Selector	336	256	**86.25**	**90.06**	**81.04**	77.03	82.43	**68.09**	77.61	78.21	**80.09**
T-Selector	336	288	86.18	89.29	80.37	**77.10**	**82.62**	67.55	**77.70**	**78.36**	79.90

In the RSOT task, the baseline method involves directly using MiniGPT-v2 to predict the coordinates of every frame based on the given expression, and the coordinates of each frame in the whole video are used for calculating metrics. In the Video-REG task, we utilize MiniGPT-v2 to generate the expression based on the given coordinates in the frame. The superior performance of Elysium, compared to MiniGPT-v2, can be attributed to its ability to capture temporal awareness and coherence, which are learned from the training data in ElysiumTrack-1M.

5.2 Ablation Studies

We conducted ablation studies to explore the impact of adapter architecture and the visual token count between the LLM and the visual encoder on object perception performance in images. The models are first pretrained on the LLaVA-558K [41] and then finetuned on the RefCOCO [28], RefCOCO+ [28], and RefCOCOg [45] datasets.

Table 7 presents the results of our experiments. We keep the visual token count per image constant, and observe that fusion operations such as cross-attention [4] and concatenation [13] do not achieve promising performances com-

pared to our T-Selector structure. Furthermore, we find that when the visual token count is the same, using a ViT@224p combined with an MLP achieves worse performance compared to using a ViT@336p along with our T-Selector architecture.

Additionally, we investigate the influence of the compression ratio on the final performance. We observe a trend where the performance degrades as the compression ratio α decreases. Specifically, we notice a significant performance drop when the visual token count per image αN decreases from 108 to 72. Therefore, to achieve a trade-off between a larger context window and performance, we select the value of $\alpha N = 108$ as default setting for subsequent experiments.

5.3 Visualizations

As depicted in Figure 4, we gather videos from Sora [10] and a video webpage [1] to ensure that our Elysium model is not trained on such videos. In Figures 4(a)-(e), we demonstrate the capability of our Elysium model to effectively handle various expressions of different lengths in the context of RSOT. The model successfully focuses on and generates the coordinates of the corresponding object based on different expressions as cues. Furthermore, Figures 4(e) and 4(f) illustrate how Elysium can accurately track objects by utilizing both expressions and bounding boxes as cues. Lastly, Figure 4(g) shows that Elysium is capable of processing frame coordinates and generating appropriate expressions accordingly.

6 Conclusions and Limitations

Conclusions. In this paper, we present Elysium, an end-to-end trainable MLLM designed to leverage the full potential of MLLM in object-level perception tasks, encompassing both images and videos, paired with two tasks, RSOT and Video-REG, which bridge the gap between language and tracking in videos. To facilitate these object-level tasks, we construct a large-scale dataset called ElysiumTrack-1M, providing support for tasks such as SOT, RSOT, and Video-REG. Furthermore, we propose a visual token compression network, T-Selector, to strike a balance between a large context window and overall performance. Through extensive experiments, we demonstrate that MLLM exhibits remarkable object perception abilities in videos. The results validate the effectiveness and potential of our proposed approach in leveraging MLLM for object-level perception tasks.
Limitations. During our explorations, we have observed that Elysium's performance in tracking-related tasks is less satisfactory when dealing with tiny objects. However, we believe that this limitation can be addressed through further exploration, particularly by incorporating higher-resolution input images. Additionally, it is worth noting that our exploration thus far has primarily focused on tracking-related tasks within the realm of videos. As part of future work, we acknowledge the need to delve into other tasks such as VOS and RVOS, which present promising avenues for further research.

Fig. 4: Visualizations of the Elysium. We showcased the performance of Elysium on several videos. We demonstrated the RSOT capability of Elysium on cases (a) to (e) while showcasing the SOT on case (f) and Video-REG on case (g).

References

1. https://www.pexels.com 13
2. et al., M.K.: The visual object tracking vot2014 challenge results. In: ECCVW (2014) 5
3. et al., M.K.: The visual object tracking vot2017 challenge results. In: ICCVW (2017) 5
4. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966 (2023) 1, 8, 12
5. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: 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. pp. 1728–1738 (2021) 2, 6
6. Banerjee, S., Lavie, A.: 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. pp. 65–72 (2005) 6
7. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14. pp. 850–865. Springer (2016) 2, 3, 11
8. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016) 2, 6
9. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6182–6191 (2019) 11
10. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators 13
11. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. pp. 190–200 (2011) 2
12. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. pp. 190–200 (2011) 11
13. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023) 1, 3, 8, 10, 11, 12
14. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023) 1, 3, 8, 10
15. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), https://lmsys.org/blog/2023-03-30-vicuna/ 1, 7
16. Cui, Y., Jiang, C., Wang, L., Wu, G.: Mixformer: End-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13608–13618 (2022) 3, 6

17. Du, F., Liu, P., Zhao, W., Tang, X.: Correlation-guided attention for corner detection based visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6836–6845 (2020) 11
18. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5374–5383 (2019) 5
19. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19358–19369 (2023) 10
20. Feichtenhofer, C., Li, Y., He, K., et al.: Masked autoencoders as spatiotemporal learners. Advances in neural information processing systems **35**, 35946–35958 (2022) 7
21. Gavrilyuk, K., Ghodrati, A., Li, Z., Snoek, C.G.: Actor and action video segmentation from a sentence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5958–5966 (2018) 2
22. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9543–9552 (2021) 3, 11
23. Huang, B., Wang, X., Chen, H., Song, Z., Zhu, W.: Vtimellm: Empower llm to grasp video moments. arXiv preprint arXiv:2311.18445 (2023) 2
24. Huang, L., Zhao, X., Huang, K.: 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) 5
25. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6700–6709 (2019) 8
26. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2758–2766 (2017) 11
27. Jung, I., Son, J., Baek, M., Han, B.: Real-time mdnet. In: Proceedings of the European conference on computer vision (ECCV). pp. 83–98 (2018) 11
28. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 787–798 (2014) 1, 8, 10, 12
29. Khoreva, A., Rohrbach, A., Schiele, B.: Video object segmentation with language referring expressions. In: Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14. pp. 123–141. Springer (2019) 2, 3
30. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: Proceedings of the IEEE international conference on computer vision. pp. 706–715 (2017) 2
31. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**, 32–73 (2017) 8
32. Lei, J., Berg, T.L., Bansal, M.: Detecting moments and highlights in videos via natural language queries. Advances in Neural Information Processing Systems **34**, 11846–11858 (2021) 2

33. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4282–4291 (2019) 3, 11
34. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8971–8980 (2018) 2, 11
35. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023) 1, 3
36. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: Chat-centric video understanding. arXiv preprint arXiv:2305.06355 (2023) 2, 8, 10, 11
37. Li, Z., Xu, Q., Zhang, D., Song, H., Cai, Y., Qi, Q., Zhou, R., Pan, J., Li, Z., Vu, V.T., et al.: Lego: Language enhanced multi-modal grounding model. arXiv preprint arXiv:2401.06071 (2024) 10
38. Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122 (2023) 2
39. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014) 1
40. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023) 1, 3
41. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023) 1, 8, 12
42. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023) 6
43. Luo, R., Zhao, Z., Yang, M., Dong, J., Qiu, M., Lu, P., Wang, T., Wei, Z.: Valley: Video assistant with large language model enhanced ability. arXiv preprint arXiv:2306.07207 (2023) 2, 10, 11
44. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv preprint arXiv:2306.05424 (2023) 2, 10, 11
45. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 11–20 (2016) 1, 8, 10, 12
46. Matthew Honnibal, Ines Montani, S.V.L., Boyd., A.: spacy: Industrial-strength natural language processing in python (2020) 6
47. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 445–461. Springer (2016) 5
48. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European conference on computer vision (ECCV). pp. 300–317 (2018) 5

49. Munasinghe, S., Thushara, R., Maaz, M., Rasheed, H.A., Khan, S., Shah, M., Khan, F.: Pg-video-llava: Pixel grounding large video-language models. arXiv preprint arXiv:2311.13435 (2023) 3
50. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023) 1, 3, 6, 8
51. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016) 2
52. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 7
53. Rasley, J., Rajbhandari, S., Ruwase, O., He, Y.: Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3505–3506 (2020) 9
54. Ren, M., Kiros, R., Zemel, R.: Exploring models and data for image question answering. Advances in neural information processing systems **28** (2015) 8
55. Ren, S., Yao, L., Li, S., Sun, X., Hou, L.: Timechat: A time-sensitive multimodal large language model for long video understanding. arXiv preprint arXiv:2312.02051 (2023) 2
56. Roffo, G., Melzi, S., et al.: The visual object tracking vot2016 challenge results. In: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II. pp. 777–823. Springer International Publishing (2016) 5
57. Seo, S., Lee, J.Y., Han, B.: Urvos: Unified referring video object segmentation network with a large-scale benchmark. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 208–223. Springer (2020) 2
58. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8430–8439 (2019) 1
59. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Guo, X., Ye, T., Lu, Y., Hwang, J.N., et al.: Moviechat: From dense token to sparse memory for long video understanding. arXiv preprint arXiv:2307.16449 (2023) 2, 10
60. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022) 7
61. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 1, 8
62. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4566–4575 (2015) 6
63. Wang, G., Luo, C., Sun, X., Xiong, Z., Zeng, W.: Tracking by instance detection: A meta-learning approach. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6288–6297 (2020) 11

64. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: International Conference on Machine Learning. pp. 23318–23340. PMLR (2022) 10
65. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al.: Cogvlm: Visual expert for pretrained language models. arXiv preprint arXiv:2311.03079 (2023) 1, 3, 8
66. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. Advances in Neural Information Processing Systems **36** (2024) 10
67. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017) 2, 6
68. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2411–2418 (2013) 6
69. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis & Machine Intelligence **37**(09), 1834–1848 (2015) 5
70. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1645–1653 (2017) 2, 11
71. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5288–5296 (2016) 2
72. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018) 2
73. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 12549–12556 (2020) 3, 11
74. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Zero-shot video question answering via frozen bidirectional language models. Advances in Neural Information Processing Systems **35**, 124–141 (2022) 10
75. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. arXiv preprint arXiv:2310.07704 (2023) 10
76. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2**, 67–78 (2014) 8
77. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: A dataset for understanding complex web videos via question answering. In: AAAI. pp. 9127–9134 (2019) 2, 11
78. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023) 2, 10
79. Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., Qiao, Y.: Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv preprint arXiv:2303.16199 (2023) 10

80. Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Chen, K., Luo, P.: Gpt4roi: Instruction tuning large language model on region-of-interest. arXiv preprint arXiv:2307.03601 (2023) 1, 3
81. Zhou, J., Wang, P., Sun, H.: Discriminative and robust online learning for siamese visual tracking. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 13017–13024 (2020) 3
82. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023) 1
83. Zhu, Y., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7w: Grounded question answering in images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4995–5004 (2016) 8