DTOS: Dynamic Time Object Sensing with Large Multimodal Model

Supplementary Material

A. Implementation Details

A.1. Training details

We report the training hyperparameters for TCS and TCD in Tab. A1. All models were trained on 4x40GB A100 GPUs, employing the ZeRO-2 optimization from DeepSpeed [13] to further reduce memory consumption. During training, we fully froze the embedding layer of the LLM and replaced the newly added tokens with learnable parameters. In the inference phase, these learnable parameters were directly mapped to the extended embedding layer. Training TCS took approximately 50 hours, and training TCD required around 62 hours. Both TCS and TCD have approximately 9 billion parameters, with around 0.7 billion being trainable about 8% of the total parameter count.

We adopt VILA1.5-LlaMA3-8B [11] as the base model, and it is equipped with an understanding of multi-image sequences after being pretrained on multiple images. We leverage the world knowledge encoded in MLLM's pretraining and the localization capabilities obtained through fine-tuning to achieve unified spatiotemporal object localization. Thanks to the model's zero-shot capability, even with limited overlap in scene and target words between the MR and RVOS datasets, we still achieve strong results, demonstrating the excellent generalization ability of MLLM.

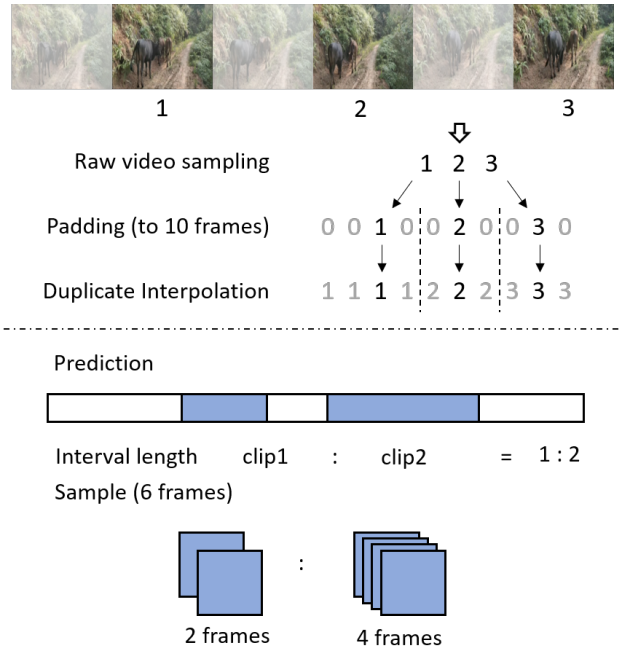

Figure A1. Insert frame schematics (top) and fragment sampling details (bottom).

Hyperparameter	TCS	TCD
Batch size	1	1
Epochs	8	8
Learning rate	5e-5	3e-5
Learning rate warmup steps	0.03	0.03
Optimizer	AdamW	AdamW
Weight decay	1e-4	1e-4
Gradient accumulation step	1	1
Input frames	20	7
LoRA rank	64	64
$\lambda^m_{gIoU}, \lambda^m_{L1}, \lambda^m_{label}$	1, 3, 1	3, 9, 1
$\lambda_{gIoU}, \lambda_{L1}, \lambda_{label}$	-	0.5, 0.5, 10
Special token num	10	7

Table A1. Training Hyperparameters for DTOS. λ^m_{gIoU}, λ^m_{L1} and λ^m_{label} are the weights for gIoU loss, L1 loss and label loss of TCS. λ_{gIoU}, λ_{L1} and λ_{label} are the weights for gIoU loss, L1 loss and label loss of TCD.

A.2. Pipline details

Negative Response We considered scenarios where the target may be absent from the frame due to occlusion or camera movement, as well as cases of target loss caused by sampling errors in our TCS. In such situations, it becomes essential for the system to generate a Negative Response to handle the absence of the target. Unlike SAM2, which trains a separate discriminator to detect target presence, we incorporate negative sample response templates directly into the training process in the Fig.3 (bottom), leveraging the MLLM's inherent visual-text understanding to make these determinations. For instance, In Fig. C3 (h) , the input prompt "*people standing*" fails to match the content of the sampled clip due to TCS error, where "*people*" are absent. Our TCD detects this and responds "*No match for the target*", refraining from segmentation. This method also helps to reduce hallucinations in the model's responses.

Duplicate Interpolation During video sampling, errors in keyframe extraction or insufficient video length often result in fewer sampled frames than required. To address this, we use duplicate interpolation to fill in the missing frames, as shown in Fig. A1 (top). For example, if 3 frames (1, 2, and 3) are sampled from the video and need to be expanded to 10 frames, we distribute the sampled frames evenly as anchor frames, using "*0*" for padding. We then compute the

TCS Prompt

You are a video location assistant. Your task name is **Moment Retrieval**. Your need to find the snippets of video that are most relevant to the query semantic information and return them in the format of the example below.

(1) You will receive a 90 frame video with rough information **which arranges in a grid view**, and 10 frame details at 0/10/20/30/40/50/60/70/80/90, and you need to correlate the two information.
(2) Then I will give you a description sentence about this video, which is relevant to particular segments.
(3) Please also note that each query may appear at any position in the video, and each query may correspond to multiple non-overlapping fragments. You need to locate the corresponding fragments of each query and generate one or more results.
(4) Please do not copy the content and answers in the examples.

=== EXAMPLES ===
[user]: (image message) (video message) (image message) (video message)... **Can you locate these descriptions in the video?** the dog ...
[system]: Considered, I would pinpoint ten video moments as: **<reg>**...**<reg>** My analysis yields detailed answers above.
=== END ===

If you understand, Please begin to answer the following questions.

TCD Prompt

You are a video segmentation assistant. Your task name is **Refferring Video Object Segmentation**. Your need to find the objects that are most relevant to the query semantic information and return them in the format of the example below.

(1) You will receive key frames of the most relevant video clips, but you still need to **find the best match among them**.
(2) Then I will give you a description sentence about this video, which is relevant to particular objects.
(3) Please also note that each query may appear at any position in these images, and each query may correspond to multiple objects. You need to locate the corresponding objects of each query and generate one or more results.
(4) Please do not copy the content and answers in the examples.

=== EXAMPLES ===
[user]: (image message) (image message)... **Could you identify the locations depicted in <tgt_i >image?** the men ...
[system]: Sure! The image shows the following locations: **<seg>**...**<seg>** These are the places depicted.
=== END ===

If you understand, Please begin to answer the following questions.

Figure A2. TCS and TCD prompts. The bolded parts are the introduction to the task, the objectives to be achieved, and the questions about the specific functionality to be implemented. The text in blue represents different special tokens that the MLLM needs to respond to.

midpoints between anchor frames to define intervals, and duplicate frames within each interval to match the anchor frames.

Sampling of Clip Length Proportions As shown in Fig. A1 (bottom), we apply this sampling method when the sampled clip consists of multiple frames. In this case, we first calculate the proportion of each clip relative to the total frame count of the TCD and then allocate the number of sampled frames based on this proportion. For example, if 6 frames are sampled in total and the length ratio of the two clips is 1:2, we would allocate 2 frames from the first clip

Data Augmentation	ActivityNet Captions [9]				QVHighlights [10] val			
	R1@0.3	R1@0.5	R1@0.7	mIoU	R1@0.3	R1@0.5	R1@0.7	mIoU
None	55.56	34.32	17.91	37.63	69.78	43.88	24.16	46.05
Aug	**60.96**	**43.27**	**25.16**	**42.77**	**82.45**	**68**	**44.84**	**60.14**

Table B2. Comparison of TCS with and without data augmentation on ActivityNet Captions [9] and QVHighlights [10] val. Aug denotes using data augmentation by referring MixGen [4].

Method	R1@.3	R1@.5	R1@.7	mIoU
VLG-Net [18]	-	33.35	25.57	-
CONQUER [7]	-	38.17	29.9	-
EventFormer [5]	-	39.02	30.91	-
PREM [6]	-	40.79	33.77	-
T-CKCN [2]	-	41.83	34.5	-
DTOS-TCS	**72.5**	**62.01**	**48.36**	**57.31**

Table B3. Comparison on the DiDeMo [1] Dataset.

Frames	R1@.3	R1@.5	R1@.7	mIoU
15	78.71	67.47	42.04	56.11
20	**79.62**	**67.58**	**42.12**	**56.45**
25	73.39	56.56	30.73	49.35

Table B4. Ablation study of visual token length on Charades-STA [17]. The numbers in the table are expressed in terms of the token length of one image as the unit of measurement.

and 4 frames from the second clip, ensuring the sampling distribution reflects the relative length of each clip.

Optimal Frame Sampling Strategy We devised a sampling strategy to select optimal frames from all the predicted clip results in the TCD. In each frame's target detection results, we use the NMS method [12] to eliminate duplicate bounding boxes and filter out outliers based on an iou threshold (set at 0.6). We then count the remaining bounding boxes across all sampled frames, selecting the frame with the most common box count as the final target frame. After NMS and outlier detection, numerous candidate frames emerge. Our selection strategy computes the IoU between bounding boxes in each frame, sums IoU values. Low IoU means more dispersed targets. Since dispersion worsens false-tracking in video, so more overlap is better. The All Mode Frames strategy is the only method that can generate multiple optimal frames. It will not be filtered out but will all be sent to SAM2 for independent propagation.

Prompt details In Sec. A.2, we designed distinct prompts for both TCS and TCD. In these prompts, we begin by clearly specifying the task requirements, followed by a detailed description of the inputs. We also provide explicit guidance to the MLLM by highlighting key steps or cognitive pro-

Sampling Strategy	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
Top1 Frame	62.81	70.6	66.71
Random Frame	66.72	74.75	70.74
Middle Frame	69.38	78.11	73.74
All Mode Frames	65.43	73.64	69.54
Optimal Frame	**70.76**	**79**	**74.88**

Table B5. Ablation study of TCD sampling strategy for optimal frames on Ref- DAVIS-17 [8]. Top1 Frame means the frame with the highest prediction score among all frames and is selected as the optimal frame. Random Frame means selecting a frame at random to be the optimal frame. Middle Frame means choosing the frame located at the center of the segment as the optimal frame. All Mode Frames involves selecting all frames that have the most frequent prediction score as the optimal frames, which is the only frame selection strategy that can result in multiple optimal frames. Optimal Frame is the our method, which is detailed in the paper.

n_frame	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
5	63.01	68.06	65.53
7	**66.29**	**70.75**	**68.52**
9	55.61	60.53	58.07

Table B6. Ablation study of the total number of input frames on the Ref-YouTube-VOS [16]. n_frame denotes the total number of frames in the clip provided to TCD. The 9-frame is tested using 4x80G A100 GPUs.

Methods	Task	Dataset	Train	Infer	TFLOPs	Score
VISA	TFS		2d	5.17s	17.61	49.3
HawkEye	MR	2108k	7d			
DTOS-TCS		90k	2d	4.96s	78.59	58
VISA-7B	RVOS	1343k	3.5d	0.61s	26.06	43.5
DTOS-9B		36k	2.6d	8.55s	332.82	48.86

Table B7. VISA [60] was trained on 171k video and 1172k image samples with 8 A100 80G GPUs. HawkEye [52] was trained with 8 V100 32G GPUs. Both VISA and HawkEye datasets have much instruction-tuning data. Scores are mIoU on Charades-STA [49] and $\mathcal{J}\&\mathcal{F}$ on MeViS [8].

Special Tokens	Ref-DAVIS17			Ref-YT-VOS		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
1	60.21	68.34	64.28	55.34	58.87	57.1
4	69.86	77.51	73.68	62.89	67.42	65.15
7	69.22	77.36	73.29	66.86	71.21	69.03

Table B8. Ablation Study of One-to-Many Design on Ref-DAVIS17 [19] val and Ref-YT-VOS [48].

cesses to consider. Furthermore, we impose constraints on the response format to effectively leverage the multimodal knowledge encoded within the MLLM. Lastly, we include an example to illustrate the process that the model should follow.

B. More Experiments

B.1. More Comparison Experiments

The DiDeMo [1] dataset is primarily used for Video Retrieval tasks, with limited reports on its performance in the sub-task of Single Video Moment Retrieval (SVMR). We have gathered relevant scores from previous studies as comprehensively as possible. In Tab. B3, our TCS has achieved state-of-the-art results, improving the R1@0.7 scores by +13.86 compared to prior methods.

B.2. More Ablation Experiments

Visual Token Length In Tab. B4, we first estimate the optimal length for image and video tokens as inputs to the model, with the best performance achieved at around 20 frames. We hypothesize that this result correlates with the input length used during pretraining of the MLLM. Fewer visual tokens fail to provide adequate information, while an excessive number of visual tokens often diverges from the pretrained visual token length, potentially degrading performance.

Data Augmentation Tab. B2 shows a comparison of performance before and after data augmentation. We observe that data augmentation not only enables one-to-many detection capabilities but also enhances textual complexity in the augmented data. This yields notable improvements in datasets with longer texts (*e.g.* ActivityNet Captions [9]) or multiple bounding boxes (*e.g.* QVHighlights [10]). DTOS performs better with less task-specific data. We compared HawkEye and VISA on different tasks, which fine-tuned the joint dataset. Comparison results are in Tab. B7. To deal with the multi-referential data bias, we used MixGen for data augmentation. We recorded all queries in one video, then concatenated, split or repeated their labels. For instance, we made 'dog, cat' from 'dog' and 'cat' with a comma, split long text queries, or repeated a query's label to create multi-referential labels. The effects are discussed in Tab. B2. We used DeepSpeed FlopsProfiler to test inference time and FLOPs on part of MeViS. In almost same time, DTOS gives precise segment info. When segmenting full-video masks, DTOS is slower as it queries multiple frames. Its 1.22s/it per-frame time is close to VISA, but gives better results.

Sampling Strategy of the Optimal Frame In Tab. B5, our sampling strategy for optimal frame outperforms methods that rely on middle frames, random frames, or frames with the highest bounding box scores. Additionally, we compared the method of selecting multiple optimal frames with the single-frame optimal approach. The results show that the single-frame method outperforms the multi-frame approach, potentially because the TCD's performance has not yet reached an ideal level. Introducing multiple frames may introduce noise, negatively affecting the model's performance.

Input Frames of TCD As shown in Tab. B6, the model achieves optimal performance when input 7 frames. Fewer frames provide limited video information, while longer clips tend to enhance robustness. However, Excessively long clips introduce temporal redundancy, making it more difficult for the model to distinguish between frames. This redundancy can also dilute the model's focus and negatively impact its performance. Additionally, longer clips consume more memory and prolong training time. We anticipate that advancements in technology will gradually address limitations related to context length and memory consumption.

Multiple Special Tokens In autoregressive model research, their usage and properties are underexplored. Our special tokens have task-assigned and text-inherent semantic meanings, avoiding numerical-value issues. [15, 19] built data-driven benchmarks for multi-reference targets, but data biases limited them. Our approach enables the model to proactively generate multiple tokens instead of self-determining the number of targets. Our method improves MLLMs' instruction-following, eases answer construction, and boosts performance Tab. B8. We added one-to-many ablation studies on other datasets. As Tab. B8 shows, this design boosts performance. Finding multiple spatiotemporal targets is much harder than querying one. Multiple query responses make the model recall past info when generating each special token. This creates stronger supervision than single, greatly improving the capabilities.

C. More Visualizations of DTOS

In this section, we present additional DTOS results for qualitative analysis, highlighting areas that can be improved in future research.

In Fig. C3, we present additional visual results that highlighting the powerful text understanding and localization capabilities of DTOS. In the successful cases, the model accurately identifies the objects referenced by the user's query and can even localize small targets that appear over a long duration (*e.g.* Fig. C3 (b)). It also makes precise predictions based on static information (*e.g.* "*yellow*" in Fig. C3 (c)) and dynamic information (*e.g.* "*descended*" in Fig. C3 (d)). In the failure cases, Fig. C3 (h) illustrates a failure where "*people*" is mistakenly detected, resulting in a failure

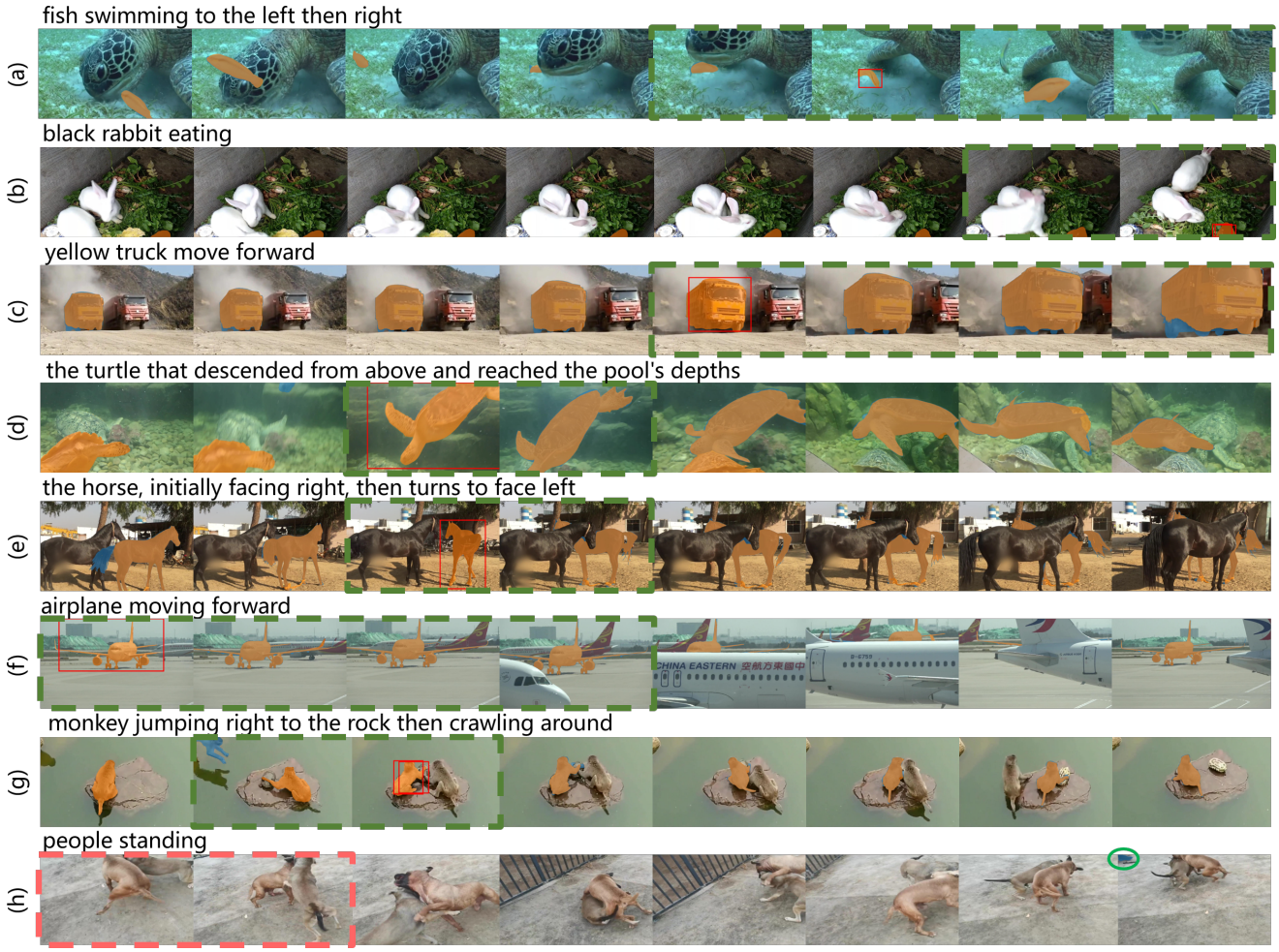

Figure C3. We provide more visual results on the MeViS [3] validation set. In the figures, the blue mask represents the ground truth, while the orange mask shows our predicted results. The green circles highlight the ground truth. The dashed boxes represent the predicted time segments from our TCS, with green indicating correct predictions and red indicating incorrect predictions where no target is detected.

in localization. However, our approach of using negative samples helps reduce hallucinations in incorrect segments and prevents the model from mislocalizing other objects, thus preventing the amplification of errors.

From these cases, several promising directions for future research arise:

- Maintaining independent information for each target as a basis for propagation and determination across frames.
- Enhancing localization accuracy to better capture the spatial-temporal characteristics of targets.
- Strengthening the model's understanding to handle more complex expressions and behaviors.

D. License

Our code and models will be publicly available under standard community licenses. Here are the links to the datasets, code, and models referenced in this paper:

MeViS[3]: MIT
Ref-YouTube-VOS[16]: CC BY 4.0
Ref-DAVIS17[8]: BSD 3-Clause
Charades-STA[17]: Non-Commercial
DiDeMo[1]: BSD 2-Clause
Activity-Captions[9]: MIT
QVHighlights[10]: CC BY-NC-SA 4.0
VILA[11]: Apache
SAM2[14]: Apache

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 3, 4, 5

[2] Tongbao Chen, Wenmin Wang, Zhe Jiang, Ruochen Li, and

Bingshu Wang. Cross-modality knowledge calibration network for video corpus moment retrieval. *IEEE Transactions on Multimedia*, 2023. 3

[3] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2694–2703, 2023. 5

[4] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multimodal data augmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 379–389, 2023. 3

[5] Danyang Hou, Liang Pang, Huawei Shen, and Xueqi Cheng. Event-aware video corpus moment retrieval. *arXiv preprint arXiv:2402.13566*, 2024. 3

[6] Danyang Hou, Liang Pang, Huawei Shen, and Xueqi Cheng. Improving video corpus moment retrieval with partial relevance enhancement. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pages 394–403, 2024. 3

[7] Zhijian Hou, Chong-Wah Ngo, and Wing Kwong Chan. Conquer: Contextual query-aware ranking for video corpus moment retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3900–3908, 2021. 3

[8] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. 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*, pages 123–141. Springer, 2019. 3, 5

[9] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 3, 4, 5

[10] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858, 2021. 3, 4, 5

[11] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024. 1, 5

[12] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, pages 850–855. IEEE, 2006. 3

[13] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 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*, pages 3505–3506, 2020. 1

[14] 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. 5

[15] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024. 4

[16] Seonguk Seo, Joon-Young Lee, and Bohyung Han. 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*, pages 208–223. Springer, 2020. 3, 5

[17] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016. 3, 5

[18] Mattia Soldan, Mengmeng Xu, Sisi Qu, Jesper Tegner, and Bernard Ghanem. Vlg-net: Video-language graph matching network for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3224–3234, 2021. 3

[19] Rongkun Zheng, Lu Qi, Xi Chen, Yi Wang, Kun Wang, Yu Qiao, and Hengshuang Zhao. Villa: Video reasoning segmentation with large language model. *arXiv preprint arXiv:2407.14500*, 2024. 4