GLUS: Global-Local Reasoning Unified into A Single Large Language Model for Video Segmentation

Supplementary Material

A. Demo Video

In Demo, we provide six qualitative comparisons between the previous state-of-the-art (DsHmp [13]) and our GLUS with the videos in MeViS [9]. Notably, these examples illustrate three challenging aspects of RefVOS: (1) **Motion Understanding**: RefVOS models have to distinguish similar objects with their motions; (2) **Global Reasoning**: RefVOS models should be capable of realizing global reasoning to segment the objects presented only in a short video clip; (3) **Vision-Language Reasoning**: RefVOS models should perform vision-language unified reasoning in complex scenarios. The six examples demonstrate that our GLUS effectively tackles RefVOS in challenging language-guided segmentation cases.

B. Implementation Details

This section provides a detailed explanation of the specific model architectures and workflow of GLUS.

B.1. Model Architectures

Multimodal LLM. The input embeddings for the MLLM are generated by processing each context and query frame individually through the vision backbone, `VB`. Subsequently, a vision-to-language projection layer, $\phi_{V\to L}$, is applied to the outputs:

$$F_t^C = \phi_{V\to L}(\texttt{VB}(I_t^C)), F_t^Q = \phi_{V\to L}(\texttt{VB}(I_t^Q)), \quad \text{(A)}$$

where F_t^C and F_t^Q are the features for the context and query frames. Then MLLM generates the t-th segmentation token as:

$$\begin{aligned}\langle\texttt{SEG}\rangle_t = \texttt{LLM}([R, F_{1:N_c}^C,\\ F_1^Q, \langle\texttt{SEG}\rangle_1, F_2^Q, \langle\texttt{SEG}\rangle_2, ..., F_t^Q]).\end{aligned} \quad \text{(B)}$$

This process follows our global-local unified design, and we adopt LISA-7B-v1 [19] for the initialization of `LLM`, projector $\phi_{V\to L}$, and backbone `VB`.

Mask Decoder. Our utilization of the mask decoder follows the style of LISA [19] and SAM-2 [32]. After obtaining $\langle\texttt{SEG}\rangle_t$, GLUS first extracts the hidden embedding $\hat{h}_t$ from the penultimate layer of the MLLM. A language-to-vision projection layer, $\phi_{L\to V}$, is then applied to $\hat{h}_t$ to generate a prompt for the mask decoder, h_t. Next, a vision encoder, `Enc`, processes the query frames to produce encoded features. Using the prompt and the encoded features, the mask decoder, `Dec`, is applied to the query image I_t^Q, generating its corresponding mask M_t:

$$h_t = \phi_{L\to V}(\hat{h}_t), M_t = \texttt{Dec}(\texttt{Enc}(I_t^Q), h_t) \quad \text{(C)}$$

In our experiments, we initialize the weights of $\phi_{L\to V}$ projection layer with LISA-7B-v1 and utilize SAM-2 to initialize image encoder `Enc` and mask decoder `Dec`..

Memory Bank. Each time a mask M_t is generated, GLUS is able to encode it using a memory encoder, $\texttt{Enc}_M$, and stores the resulting feature F_t^M in `MemBank`. For memory attention, we adopt the design of SAM-2 [32], selecting features from up to m masks in `MemBank`. Attention is then applied to these features along with the decoded image to produce the input for the mask decoder:

$$\begin{aligned}F_t^M &= \texttt{Enc}_M(M_t),\ \texttt{MemBank.Push}(F_t^M)\\ \hat{F}_{t+1}^M &= \texttt{Concat}(F_{i_1}^M, F_{i_2}^M, ..., F_{i_m}^M)\\ \hat{F}_{t+1}^Q &= \texttt{MemAttn}(\texttt{Enc}(I_{t+1}^Q), \hat{F}_{t+1}^M)\\ M_{t+1} &= \texttt{Dec}(\hat{F}_{t+1}^Q, h_{t+1})\end{aligned} \quad \text{(D)}$$

where $\{i_p\}_{p=1}^m$ is the selected masks from memory bank following SAM-2. We adopt SAM-2's memory attention module and memory encoder in our experiments.

B.2. GLUS Training Details

This section provides detailed training configurations for GLUS (Sec. 4), as summarized in Table A. During training, only the MLLM (fine-tuned with LoRA [15]), mask decoder, and projection layers are trainable. DeepSpeed [31] is employed to improve training efficiency. The sampling frequency in the memory bank is set to 1 during training to maximize its utilization. The training process takes approximately 25 hours on 4 NVIDIA A100 GPUs (40 GB each), with 3000 steps, 10 gradient accumulation steps and a batch size of 2 per device.

The training objective incorporates cross entropy (CE) loss, mask loss (comprising mask DICE loss and mask BCE loss), and contrastive loss, as described in Sec. 4.3. The corresponding weights, λ_{ce}, λ_{dice}, λ_{bce}, and λ_{ct}, are used to compute their respective averages.

B.3. GLUS Inference Details

During inference, GLUS employs a sliding window approach with a size of 4 and a stride of 1 for the query frames. The mask of the last query frame is used as the context of the next group of query frames. The sampling frequency for the memory bank is set to sample once per 3 frames, and a maximum of 7 masks are used in mask attention. Addi-

Config	Value
context frame num	4
question frame num	4
input resolution	224
features downsampling rate	4
optimizer	Adam
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
optimizer weight decay	0.0
learning rate	3e-4
LoRA rank	8
λ_{ce}	1.0
λ_{dice}	0.5
λ_{bce}	2.0
λ_{ct}	0.1
batch size	80
gradient accumulation steps	10
warmup steps	100

Table A. Implementation details of GLUS training process.

tional ablation studies on sampling frequency are provided in Sec. C.

B.4. Selector training and inference

Data Annotation To generate the pseudo-labels for fine-tuning the selector model, we use GLUS to generate the masks on the training set and compute the IoU of the masks. To mitigate the risk of overfitting, we adopt an early-stop model (trained for 500 steps) rather than the final model (trained for 3000 steps). For faster training of the selector, we label only half of the training set as the training data for selector fine-tuning.

Implementation Details We use Chat-Univi [17] as the base Video-QA model. Similar to the design of recent grounding LLMs [3, 19, 45, 54], we introduce a special token, $\langle\texttt{SCORE}\rangle$, into the LLM vocabulary and employ an MLP to project the corresponding embeddings. During training, we randomly sample 8 frames to represent video context and produce the score for each query frame. The hidden embedding of the score token, $\hat{h}_s$, is generated as:

$$\hat{h}_s = \texttt{Selector}([P, F^C_{1:8}, F^Q, \langle\texttt{SCORE}\rangle]) \tag{E}$$

where P represents the language prompt. The hidden embedding of $\langle\texttt{SCORE}\rangle$ is then projected to score s through an MLP layer. The selector fine-tuning objective consists of two components: $\mathcal{L}_s$, an L_1 loss that supervises the frame score s using the IoU pseudo-labels y of the query frame, and $\mathcal{L}_{\text{txt}}$, a cross-entropy loss that supervises the text outputs of the LLM:

$$\begin{aligned} s &= \phi_{\text{proj}}(\hat{h}_s), \\ \mathcal{L}_s &= |y - s|, \\ \mathcal{L}_{\text{all}} &= \mathcal{L}_{\text{txt}} + \lambda_s \cdot \mathcal{L}_s \end{aligned} \tag{F}$$

For efficient training, the selector LLM is fine-tuned with LoRA [15], while the MLP layer is fully trainable. Further details on selector training are provided in Table B.

Config	Value
context frame num	8
query frame num	1
optimizer	Adam
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
optimizer weight decay	0.0
learning rate	3e-4
LoRA rank	8
λ_s	1.0
batch size	80
gradient accumulation steps	10
MLP layer num	3

Table B. Implementation details of selector training process.

Inference and Propagation The selector is trained to predict a confidence score for each frame in a test-time video, reflecting the importance of a frame with respect to the given expression. During inference, we first select the frame with the highest score as the key frame for each video-expression pair. We then use GLUS to initiate tracking from the selected frame in both forward and backward propagation directions and iteratively generate the predictions for the entire video.

C. Additional Studies

Sampling Ratio of Training Datasets As noticed in previous works [41], balancing the training data is critical for vision language models. We observe the same when training GLUS with Ref-Youtube-VOS and MeViS. For this ablation, we use the GLUS with memory bank and global-local unified reasoning enabled, and train it across different sampling ratios of the two datasets. The performance and optimization steps needed for convergence are in Table C. For balanced performance and training efficiency, we select 1:1 as the standard sampling rate for our models.

Data Scarcity of MLLM in Video Segmentation

Fine-tuning LLMs requires large amounts of data, especially for video MLLMs [20, 40, 41]. However, video data is scarce, especially when requiring fine-grained annotations like RefVOS. With the default training steps 3000, the training of GLUS without extended datasets averagely spans ∼11.6 epochs over the whole frames set, which contrasts the common 1 or 2 epochs SFT schedule for vision-language models fine-tuned with sufficient data [17, 20, 22, 24, 40, 41].

This led to noticeable overfitting with more training steps, according to the change of validation set performance (MeViS valid_u) in Fig. A. Although the object contrastive loss alleviates the overfitting issue, they all suffer from a significant drop at the final steps. We hypothesize that such a data scarcity problem constrains the performance of video MLLMs, especially when they don't have tailored designs

MeViS : RefYTB	MeViS (valid_u)	MeViS (valid)	RefYTB (valid)	Best Step
2:1	**60.8**	49.0	64.1	1500
1:1	59.7	49.5	65.2	1500
1:2	59.6	49.3	**65.6**	2500
4:15	59.6	**49.9**	65.5	3000

Table C. Ablation studies on sampling ratio of MeViS:Ref-Youtube-VOS for training. We report the performance ($\mathcal{J}\&\mathcal{F}$) and the training steps needed for convergence. underline denotes the second best. We select 1:1 as the standard ratio for GLUS to balance performance across datasets and training efficiency. (The 4:15 ratio is adopted from [45].)

such as hierarchical perception [13]. We hope our observation can encourage more explorations on scaling up the video segmentation data.

Memory Bank Sampling Frequency The VOS memory bank is integrated into our framework and optimized end-to-end to enhance global-local reasoning capabilities in complex scenarios (Sec. 4.2). We evaluate the impact of memory stride in Table D, where a longer stride prioritizes global reasoning, while a shorter stride emphasizes local consistency. We show that GLUS performs stably with varied memory bank strides, because of its design unifying both global and local reasoning.

Sampling Frequency	MeViS (valid_u)
w/o MB	58.3
1	59.3
3	59.7
5	59.7
7	59.7
9	59.7

Table D. Ablation studies on the sampling frequency of memory bank. We select 3 as the default stride of the sampling frequency, following SAM2. "MB": Memory Bank.

D. Limitations and Future Works

Our work mainly focuses on the *fine-tuning* phase of a multimodal large language model for referring video object segmentation. Therefore, the visual backbone and LLM are limited in understanding the video. From this perspective, meaningful future work would start from an MLLM *pre-trained* for video understanding to further enhance the motion understanding.

In addition, our computational resources heavily constrain our *context lengths* for an MLLM and limit the capability for video understanding. Concretely, we have to downsample the visual features and can only sample 4 context frames to summarize the video content, which might not cover the critical contexts if motions are happening fast. We hope combining our GLUS design with longer context windows can further unleash its potential.

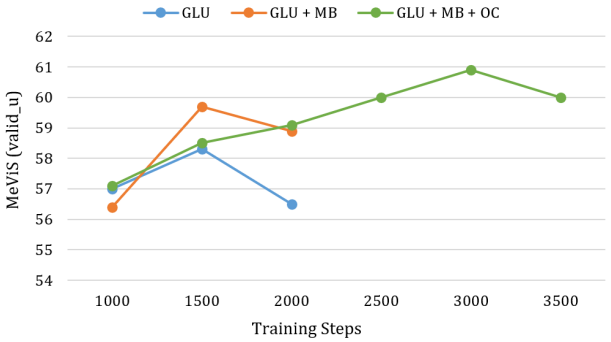

Figure A. Curves of MeViS valid_u performance ($\mathcal{J}\&\mathcal{F}$) with distinct training steps. The figure clearly demonstrates noticeable overfitting in the model. "GLU": Global-local unification, "MB": End-to-end memory bank, "OC": Object contrastive loss.

Finally, we notice that the amount of data has become a bottleneck for video reasoning (Fig. A). Therefore, future work can focus on improving the data scale and quality, where we hope the benefit of pseudo-labeling from GLUS can also be of use.

References

[1] Ali Athar, Jonathon Luiten, Alexander Hermans, Deva Ramanan, and Bastian Leibe. HODOR: High-level object descriptors for object re-segmentation in video learned from static images. In *CVPR*, 2022. 3

[2] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. TarVis: A unified approach for target-based video segmentation. In *CVPR*, 2023. 3

[3] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *NeurIPS*, 2024. 2, 3, 4, 5, 6, 7

[4] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *CVPR*, 2022. 2, 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 2020. 5

[6] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 3

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

[8] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *CVPR*, 2024. 3, 8

[9] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, 2023. 2, 6, 7, 1

[10] Zihan Ding, Tianrui Hui, Junshi Huang, Xiaoming Wei, Jizhong Han, and Si Liu. Language-bridged spatial-temporal

interaction for referring video object segmentation. In *CVPR*, 2022. 7

[11] Mohamed El Banani, Karan Desai, and Justin Johnson. Learning visual representations via language-guided sampling. In *CVPR*, 2023. 5

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 5

[13] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. In *CVPR*, 2024. 2, 7, 1, 3

[14] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. In *CVPR*, 2024. 5

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ICLR*, 2021. 6, 1, 2

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 3

[17] 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. 6, 8, 2

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, 2023. 3

[19] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024. 2, 3, 4, 5, 6, 7, 1

[20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2

[21] Minghan Li, Shuai Li, Xindong Zhang, and Lei Zhang. Univs: Unified and universal video segmentation with prompts as queries. In *CVPR*, 2024. 3

[22] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 2

[23] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2025. 8

[24] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, 2024. 2

[25] Zhuoyan Luo, Yicheng Xiao, Yong Liu, Shuyan Li, Yitong Wang, Yansong Tang, Xiu Li, and Yujiu Yang. SOC: semantic-assisted object cluster for referring video object segmentation. In *NeurIPS*, 2023. 2, 7

[26] Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, Mubarak Shah, and Fahad Khan. Pg-video-llava: Pixel grounding large video-language models. *arXiv preprint arXiv:2311.13435*, 2023. 3

[27] Shehan Munasinghe, Hanan Gani, Wenqi Zhu, Jiale Cao, Eric Xing, Fahad Shahbaz Khan, and Salman Khan. Videoglamm: A large multimodal model for pixel-level visual grounding in videos. *arXiv preprint arXiv:2411.04923*, 2024. 2, 7

[28] Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, Lingpeng Kong, and Tong Zhang. Detgpt: Detect what you need via reasoning. In *ACL*, 2023. 3

[29] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[30] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *CVPR*, 2024. 3

[31] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*, 2020. 1

[32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 3, 5, 6, 8, 1

[33] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *CVPR*, 2024. 3

[34] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020. 2, 6, 7

[35] Jiajin Tang, Ge Zheng, and Sibei Yang. Temporal collection and distribution for referring video object segmentation. In *ICCV*, 2023. 2, 7

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[37] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FeelVOS: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 3

[38] Haochen Wang, Cilin Yan, Shuai Wang, Xiaolong Jiang, XU Tang, Yao Hu, Weidi Xie, and Efstratios Gavves. Towards open-vocabulary video instance segmentation. In *ICCV*, 2023. 6

[39] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 5

[40] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun

Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *ECCV*, 2022. 2

[41] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. In *ECCV*, 2024. 2

[42] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *ICCV*, 2023. 2, 7

[43] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022. 2, 7

[44] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *ECCV*, 2022. 5

[45] Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. Visa: Reasoning video object segmentation via large language models. *ECCV*, 2024. 2, 3, 4, 5, 6, 7

[46] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024. 3

[47] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 3

[48] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[49] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *TPAMI*, 44(9), 2021. 3

[50] Linfeng Yuan, Miaojing Shi, Zijie Yue, and Qijun Chen. Losh: Long-short text joint prediction network for referring video object segmentation. In *CVPR*, 2024. 2, 7

[51] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *CVPR*, 2024. 3

[52] 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. 2, 3, 6, 7

[53] Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. Rmem: Restricted memory banks improve video object segmentation. In *CVPR*, 2024. 3

[54] Jiawen Zhu, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Bin Luo, Huchuan Lu, Yifeng Geng, and Xuansong Xie. Tracking with human-intent reasoning. *arXiv preprint arXiv:2312.17448*, 2023. 2, 3, 7