General Object Foundation Mode l for Images and Videos at Scale
Supplemental Material

In this supplementary material, we first provide more detailed information on data usage and model training in Sec 1. Subsequently, in Sec 2, we supplement additional zero-shot and fine-tuning results on classic object-level video tasks, such as VOS and RVOS. In Sec 3, we provide ablation studies on model architecture design, data utilization, and data scaling. In Sec 4, detailed zero-shot experimental results on the ODinW [11] benchmark are provided to validate the transferability of GLEE to various real-world tasks. Finally, in Sec 5, we showcase the results in interactive segmentation and tracking for images and videos.

1. Datasets and Implementation Details

Data Preparation. To ensure the generalization of GLEE as an object-level foundation model, we conduct joint training using a substantial amount of data with region-level annotations from both images and videos. Existing datasets exhibit variations in annotation granularity: detection datasets such as Objects365 [28] and OpenImages [10] provide bounding boxes and category names; COCO [17] and LVIS [7] offer more detailed mask annotations; RefCOCO [22, 40] and Visual Genome [9] include comprehensive object descriptions. Furthermore, video datasets [25, 27, 31, 34, 35, 37] contribute to the temporal consistency of models, and open-world data [8, 31] enrich the annotations with class-agnostic object information. We extracted subsets of 500,000 and 2,000,000 images from the SA1B [8] dataset for joint training in stage 2 and scale-up training respectively. To ensure that objects from SA1B are at the object-level rather than the part-level, we apply mask IoU based NMS and use area as NMS score to eliminate part-level object annotations. For SA1B and UVO data, we set the category name for each object to be 'object' and train in instance segmentation paradigm. For GRIT [24] data, we extract 5,000,000 samples for scale-up training to enhance the richness of object descriptions. A comprehensive list of the datasets we utilized, along with their respective sizes and annotation granularities, is presented in Table 1.

	Sizes		Annotations			
dataset	images	objects	semantic	box	mask	track id
Detection Data						
Objects365 [28]	1817287	26563198	category	✓	-	-
OpenImages [10]	1743042	14610091	category	✓	-	-
LVIS [7]	100170	1270141	category	✓	✓	-
COCO [17]	118287	860001	category	✓	✓	-
BDD [39]	69863	1274792	category	✓	✓	-
Grounding Data						
RefCOCO [40]	16994	42404	description	✓	✓	-
RefCOCOg [22]	21899	42226	description	✓	✓	-
RefCOCO+ [40]	16992	42278	description	✓	✓	-
VisualGenome [9]	77396	3596689	description	✓	-	-
GRIT [24]	5117307	9090607	description	✓	-	-
OpenWorld Data						
UVO [31]	16923	157624	-	✓	✓	-
SA1B [8]	2147712	99427126	-	✓	✓	-
Video Data						
YTVIS19 [37]	61845	97110	category	✓	✓	✓
YTVIS21 [34]	90160	175384	category	✓	✓	✓
OVIS [25]	42149	206092	category	✓	✓	✓
UVO-dense [31]	45270	657990	-	✓	✓	✓
VOS [35]	94588	156310	-	✓	✓	✓
RefVOS [27]	93857	159961	description	✓	✓	✓

Table 1. The tasks GLEE learns to complete and the datasets used in training.

Proportions of datasets used. We balance the overall dataset proportion by annotation types (category: description: class-agnostic=7:5:3) due to extreme size imbalances among training datasets, and we reduce the inclusion of video frames to ensure they only provide supplementary categories. As some studies [20] have found, adding certain datasets to the mix can result in negative downstream performance; we too observed that the balance of data ratios affects final outcomes. For COCO, GLEE's performance remains stable regardless of ratio adjustments, but for LVIS and RefCOCO, increasing their proportions can boost performance on those datasets. However, joint-training may also lead to lower performance on RefCOCO but improve LVIS compared to task-specific training. Despite this, our unified training objective significantly reduces this effect and aims for global optimality. Resource constraints prevent us from confirming the absolute optimality of our mix ratios, but we ensure that versatility and performance stay competitive with state-of-the-art.

Model and Training Details. Following the image backbone, text encoder, and visual prompter, we incorporate a 6-layer deformable transformer encoder and a 9-layer decoder to serve as our Object Decoder following MaskDINO [14]. We adopt 300 object queries, query de-

Datasets	OpenImages	Objects365	LVIS	VisualGenome	COCO	RefCOCO-mixed	SA1B	UVO-frame	BDD	YTVIS19	YTVIS21	OVIS	Ref-YTBVOS
Ratio	1.5	1.5	1.5	2	1.5	2.5	2.5	0.2	0.15	0.3	0.3	0.3	0.3

Table 2. The data sampling ratios during the joint-training of stage 2. RefCOCO-mixed refers to the mixed dataset of RefCOCO [40], RefCOCO+ [40], RefCOCOg [22], and the last four video datasets are treated as independent image data for training.

	backbone			GLEE		
	name	Pre-trained	Top1 Acc	Params(M)	FLOPs(T)	FPS(A100)
Lite	ResNet-50	ImageNet-1K	76.0	127.27	1.17	7.6
Plus	Swin-Large	ImageNet-22K	87.3	295.53	2.47	4.2
Pro	EVA02-Large	Merged-38M	90.0	399.60	7.25	1.5

Table 3. Comparison of backbones and GLEE models. ‘Top1 Acc’ refers to the accuracy of the backbone on ImageNet-1K classification task, the FLOPs and FPS are test on COCO val 2017 with an input resolution of 1024^2 for Lite/Plus and 1536^2 for Pro.

	Method	YT-VOS 2018 val [35]					MOSE val [4]		
		$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Memory	STM [23]	79.4	79.7	84.2	72.8	80.9	-	-	-
	SWEM [18]	82.8	82.4	86.9	77.1	85.0	50.9	46.8	64.9
	STCN [3]	83.0	81.9	86.5	77.9	85.7	50.8	46.6	55.0
	XMem [2]	86.1	85.1	89.8	80.3	89.2	57.6	53.3	62.0
Non-Memory	SiamMask [30]	52.8	60.2	58.2	45.1	47.7	-	-	-
	Siam R-CNN [29]	73.2	73.5	-	66.2	-	-	-	-
	TVOS [41]	67.8	67.1	69.4	63.0	71.6	-	-	-
	FRTM [26]	72.1	72.3	76.2	65.9	74.1	-	-	-
	UNINEXT-R50 [36]	77.0	76.8	81.0	70.8	79.4	-	-	-
	UNINEXT-L [36]	78.1	79.1	83.5	71.0	78.9	-	-	-
	UNINEXT-H [36]	78.6	79.9	84.9	70.6	79.2	-	-	-
	GLEE-Lite	80.4	80.2	85.5	74.3	81.4	56.1	51.8	60.4

Table 4. Performance comparison of our GLEE on video object segmentation tasks.

noising, and hybrid matching to accelerate convergence and improve performance. During the pretraining phase of stage 1, we sample data from Objects365 and OpenImages in a 1:1 ratio, with the batch size of 128 for 500,000 training iterations. Moving to stage 2, we train GLEE for 500,000 iterations on all image-level data jointly according to the ratios outlined in Table 2. For the scale-up training, we set the sampling ratios for SA1B and GRIT to 5.0 in Table 2, and train for an extra 500,000 iterations. We used AdamW [19] optimizer with base learning rate of 1×10^{-4}, and weight decay of 0.05, learning rate is decayed at the 400,000 iterations by a factor of 0.1. Learning rates of the image backbone and text encoder are multiplied by a factor of 0.1. For the ResNet-50 backbone and Swin backbone, we use scale augmentation [33], resizing the input images such that the shortest side is at least 480 and at most 800 pixels while the longest at most 1333. For EVA02-L backbone, we use the large-scale jittering (LSJ) [6] augmentation with a random scale sampled from range 0.1 to 2.0 followed by a fixed size crop to 1536×1536. For the Lite, Plus, and Pro models, each was trained for 3, 5, and 7 days respectively for each pretraining, joint-training, and scale-up stages. We select three backbones based on capacity to construct GLEE with expected performance of Lite $<$ Plus $<$ Pro. Inference cost, sizes, FLOPs are listed in Table 3.

2. Transfer to Video Tasks

To substantiate the effectiveness of GLEE across diverse object-level video tasks, we present the performance on VOS and RVOS tasks in Table 4 and Table 6 respectively.

VOS. Video object segmentation (VOS) aims at segmenting a particular object throughout the entire video clip sequence. We evaluate GLEE on semi-supervised VOS [1] that gives the first-frame mask of the target object on YouTube-VOS 2018 [35] and MOSE [4]. Given the first-frame mask of the target object, we first crop the prompt square area from RGB image and send it to the image backbone to obtain the visual prompt feature of the corresponding area, and send it to the early fusion module before the Transformer encoder. Then we sample fine-grained visual embeddings from the pixel embedding map M_p inside the given mask area and make them interacted with object queries through self-attention module in the Transformer decoder layer. We conduct fine-tuning of GLEE-Lite jointly on YouTube-VOS [35], YTVIS2019 [37], YTVIS2021 [34], OVIS [25], and UVO-video [31] for 40,000 iterations. The evaluation is performed on YouTube-VOS and MOSE, as shown in the Table 4. It is noteworthy that semi-supervised VOS is almost dominated by space-time memory networks [2, 3, 18, 23] which construct a memory bank for each object in the video. GLEE achieves the best results among all non-memory-based methods on YouTube-VOS and even demonstrating competitive results compared to memory-based methods on the more challenging MOSE dataset.

GLEE-Lite	COCO	RefCOCO	ODinW13	TAO	BURST	YTVIS	OVIS
	AP_{box}	P@0.5	Avg.AP	TETA	HOTA	AP	AP
CLIP-frozen	51.6	81.8	41.4	36.2	22.6	51.0	24.2
CLIP-unfrozen	52.6	84.1	37.2	35.8	21.9	51.8	24.1
CLIP-distillation (ours)	52.8	84.5	41.5	36.4	24.9	52.2	24.7
w/o video frames	53.0	84.9	40.0	35.0	22.9	39.0	17.0

Table 5. Ablation studies on text encoder and video frame data, all results were obtained directly after joint training without any fine-tuning. Higher metrics indicate better performance.

Method	Backbone	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
CMSA [38]	ResNet-50	36.4	34.8	38.1
YOFO [12]		48.6	47.5	49.7
ReferFormer [32]		58.7	57.4	60.1
UNINEXT [36]		61.2	59.3	63.0
PMINet + CFBI [5]	Ensemble	54.2	53.0	55.5
CITD [16]		61.4	60.0	62.7
ReferFormer [32]	Video-Swin-B	64.9	62.8	67.0
SOC [21]		67.3	65.3	69.3
UNINEXT [36]	ConvNext-L	66.2	64.0	68.4
UNINEXT [36]	ViT-H	70.1	67.6	72.7
GLEE-Plus	Swin-L	67.7	65.6	69.7
GLEE-Pro	EVA02-L	70.6	68.2	72.9

Table 6. Performance comparison of our GLEE on Ref-YouTube-VOS task.

RVOS. Referring Video Object Segmentation (R-VOS) aims at finding objects matched with the given language expressions in a given video and segment them. Ref-YouTube-VOS [27] is a popular R-VOS benchmarks, which are constructed by introducing language expressions for the objects in the original YouTube-VOS [35] dataset. As same as semi-supervised VOS, region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and the averaged score $\mathcal{J}\&\mathcal{F}$ are adopted as the metrics. Given an object expression and a video, we send the description into the text encoder, select the object query with the highest confidence score and compute its mask. Additionally, we introduce temporal consistency by adding the similarity between the 300 object queries of the current frame and the object query selected in the previous frame to the current confidence score. We directly evaluate the GLEE trained from stage 2 on Ref-YouTube-VOS. As shown in Table 6, GLEE outperforms all previous R-VOS approaches and unified method.

3. Abaltion Study

In this section, we first conduct ablation experiments and discussions on certain model design structures: distilling knowledge from CLIP, using contrastive loss, and extracting visual prompt embeddings twice. Then, we explore the use of data and the effects of data scaling.

3.1. Model Designs

We conducted ablation studies by training GLEE-Lite for 340,000 steps on half the joint data to assess the effects of the CLIP text encoder and video-as-image data usage. There are three approaches to utilizing the CLIP text encoder: using a frozen pretrained CLIP text encoder, denoted as CLIP-frozen; using a pretrained CLIP text encoder with updatable weights during training, denoted as CLIP-unfrozen; and employing two pretrained CLIP text encoders, one frozen as a teacher model and the other unfrozen as a student model, denoted as CLIP-distillation. The text embeddings from student model are employed to query objects in images, while the teacher model generates a standard text embedding. The student's predictions are then aligned with the teacher's through an L1 loss to ensure the student's text embeddings remain within the CLIP space. In this setting, we also compared the effect of removing image data from the VIS dataset in our joint-dataset and reported the performance of these four settings on COCO, RefCOCO, ODinW13, TAO, BURST, YTVIS19, and OVIS tasks in Table 5. It can be seen that unfreezing the text encoder reduces zero-shot performance on ODinW, indicating a decrease in downstream task generalization, while a frozen encoder has difficulty with region-level description discernment, resulting in lower performance on REC tasks such as RefCOCO. Distillation balances both, retaining their advantages. In addition, not using video frames as images has little impact on zero-shot video tasks (TAO, BURST) but reduces performance on YTVIS and OVIS, thus we add video frames to improve data diversity.

Contrastive loss is only employed during VIS task (OVIS) fine-tuning to improve tracking. Its omission results a decrease in AP from **32.3** → **26.7** on OVIS, demonstrating its significant impact on tracking in complex scenes. Extracting visual prompt twice is designed to provide finer-grained guidance for video object segmentation, removing the first extraction (early-fusion) leads to a drop in YT-VOS 2018 performance from **80.4** → **60.1**.

3.2. Data Scaling

We conducted experiments to investigate the impact of training data scale on zero-shot performance across various tasks. To this end, we trained GLEE-Pro with 10%,

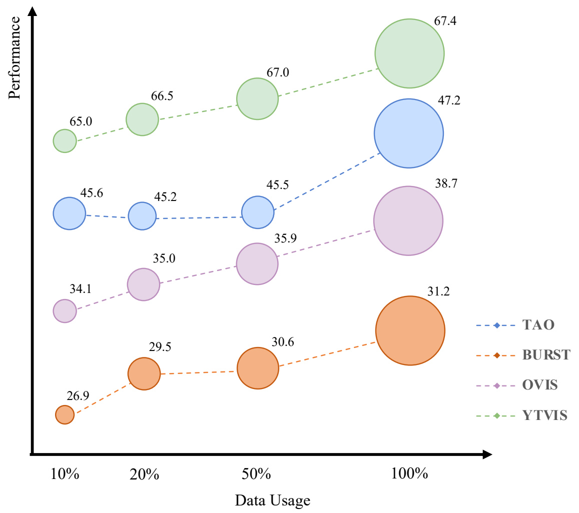

Figure 1. **Data scaling.** The performance of GLEE-Pro after training on 10%, 20%, 50%, 100% of the total data on TAO, BURST, OVIS, YTVIS19. Increased scale of training data result in enhanced zero-shot performance across diverse downstream tasks.

20%, 50%, 100% of the training data to evaluate the performance on zero-shot transfer tasks, including TAO, BURST, OVIS, and YTVIS as illustrated in the Figure 1. Our data scaling experiments reveal that increased sizes of training datasets result in enhanced zero-shot performance across diverse downstream tasks. This outcome implies that larger pre-training datasets are a valuable investment, offering a more effective and adaptable basis for a broad spectrum of downstream tasks. Thanks to the unified training approach of GLEE, we can efficiently incorporate any manually or automatically annotated data into our training process to achieve enhanced generalization capabilities.

4. Object Detection in the Wild

To evaluate generalization of GLEE on diverse real-world tasks, we conducted additional experiments on the "Object Detection in the Wild" (ODinW) benchmark [11], which is a suite of datasets covering a wide range of domains. We report the average mAP on the subset of 13 ODinW detection datasets introduced in [15], and report the per-dataset performance in a zero-shot manner, as shown in Table 8. GLEE performs better than GLIP [15] on the average of 13 public datasets, showcasing its robust generalization capability. Furthermore, it is evident that by introducing automatically labeled data at a low cost for scaling up the training data, the zero-shot capabilities can be further enhanced, this reveals that GLEE has greater potential through scale-up.

To further validate transferability of GLEE on diverse real-world detection tasks, we assess its few-shot transfer ability on the ODinW [11]. We vary the amount of task-specific annotated data from X-shot, providing at least X examples per category, to using all the available data in the training set, following the procedure established by GLIP [15]. We fine-tune the models on the provided data using the same hyper-parameters across all models in a full-model tuning regime. For manually designed prompts, we revise the category names for the two datasets ("Cottontail-Rabbit" to "rabbit" and "Cow/Chanterelle" to "Cow/Chanterelle mushroom") to provide language guidance. Models train with a batch size of 4 and a learning rate of 1×10^{-4}, undergoing 200, 300, 400, 600, and 2000 iterations for the 1, 3, 5, 10, and ALL shot splits, respectively. The optimal model is selected based on the validation split for each train/val split. For each few-shot setting, we train the models three times using different random seeds for train/val splits, and the average score and standard deviation on the test split are reported, as shown in the Table 9.

Method	Point	Box	Point (ffn)	Box (ffn)
SAM-B	52.0	74.9	-	-
SAM-L	56.6	77.2	-	-
Semantic-SAM (T)	54.5	-	-	-
Semantic-SAM (L)	57.0	-	-	-
GLEE-Lite	62.2	72.8	62.6	72.8
GLEE-Pro	64.7	72.9	64.8	73.1

Table 7. Promptable segmentation results.

5. Interactive Segmentation and Tracking

As described in Sec 2, GLEE achieves interactive segmentation and tracking by introducing a visual prompt. Sending points, boxes, or scribbles along with the image to the model enables the segmentation of specified objects. Moreover, by feeding the mask from the previous frame and its corresponding prompt feature into early fusion and self-attention, GLEE performs segmentation in the current frame based on the segmentation results from the previous frame. The features of objects in the previous frame serve as referring features at this point. As illustrated in the Figure 2, we showcase the interactive segmentation results of different prompts on images and videos.

As shown in the Table 7, we compared the 1-click mIoU for interactive segmentation on COCO between SAM [8] and Semantic-SAM [13]. As introduced in Method, we empoly a FFN to predict confidence scores to identify the most confident object query in interactive segmentation. Compared to calculating scores based on the similarity with the 'object' text prompt, the FFN provides more stable results by avoiding the influence of text prompts.

Model	PascalVOC	AerialDrone	Aquarium	Rabbits	EgoHands	Mushrooms	Packages	Raccoon	Shellfish	Vehicles	Pistols	Pothole	Thermal	Avg
GLIP-T	56.2	12.5	18.4	70.2	50.0	73.8	72.3	57.8	26.3	56.0	49.6	17.7	44.1	46.5
GLIP-L	61.7	7.1	26.9	75.0	45.5	49.0	62.8	63.3	68.9	57.3	68.6	25.7	66.0	52.1
GLEE-Lite	61.7	7.9	23.2	72.6	41.9	51.6	32.9	51.1	35.0	59.4	45.6	21.8	56.9	43.2
GLEE-Lite-Scale	61.2	5.0	23.9	71.9	46.2	57.8	25.6	56.8	33.1	60.6	57.1	25.3	52.5	44.4
GLEE-Plus	67.8	10.8	38.3	76.1	47.4	19.2	29.4	63.8	66.7	63.8	62.6	15.3	66.5	48.3
GLEE-Plus-Scale	67.5	12.1	39.7	75.8	50.3	41.1	42.4	66.4	64.0	62.8	61.8	17.5	63.8	51.2
GLEE-Pro	68.9	16.5	37.6	77.2	23.3	40.1	44.7	68.2	66.2	66.1	63.2	18.1	65.8	50.5
GLEE-Pro-Scale	69.1	13.7	34.7	75.6	38.9	57.8	50.6	65.6	62.7	67.3	69.0	30.7	59.1	53.4

Table 8. Zero-shot performance on 13 ODinW datasets.

Model	Shot	Tune	PascalVOC	AerialDrone	Aquarium	Rabbits	EgoHands	Mushrooms	Packages	Raccoon	Shellfish	Vehicles	Pistols	Pothole	Thermal	Avg
DyHead coco	1	Full	31.7$_{\pm 3.1}$	14.3$_{\pm 2.4}$	13.1$_{\pm 2.0}$	63.6$_{\pm 1.4}$	40.9$_{\pm 7.0}$	67.0$_{\pm 3.6}$	34.6$_{\pm 12.1}$	45.9$_{\pm 3.8}$	10.8$_{\pm 5.0}$	34.0$_{\pm 3.3}$	12.0$_{\pm 10.4}$	6.1$_{\pm 1.3}$	40.9$_{\pm 7.4}$	31.9$_{\pm 3.3}$
DyHead coco	3	Full	44.1$_{\pm 0.7}$	19.2$_{\pm 3.0}$	22.6$_{\pm 1.3}$	64.8$_{\pm 1.7}$	54.4$_{\pm 2.5}$	78.9$_{\pm 1.3}$	61.6$_{\pm 10.3}$	50.0$_{\pm 2.1}$	20.8$_{\pm 3.5}$	44.9$_{\pm 1.9}$	34.4$_{\pm 11.1}$	20.6$_{\pm 2.4}$	57.9$_{\pm 2.3}$	44.2$_{\pm 0.3}$
DyHead coco	5	Full	44.9$_{\pm 1.5}$	22.2$_{\pm 3.0}$	31.7$_{\pm 1.0}$	65.2$_{\pm 1.5}$	55.6$_{\pm 3.7}$	78.7$_{\pm 3.9}$	50.1$_{\pm 13.7}$	48.7$_{\pm 4.8}$	22.8$_{\pm 3.3}$	52.0$_{\pm 1.2}$	39.8$_{\pm 6.7}$	20.9$_{\pm 1.5}$	48.0$_{\pm 2.8}$	44.7$_{\pm 1.7}$
DyHead coco	10	Full	48.4$_{\pm 1.2}$	27.5$_{\pm 1.4}$	39.3$_{\pm 2.7}$	62.1$_{\pm 5.9}$	61.6$_{\pm 1.4}$	81.7$_{\pm 3.4}$	58.8$_{\pm 9.0}$	52.9$_{\pm 3.2}$	30.1$_{\pm 3.2}$	54.1$_{\pm 3.3}$	44.8$_{\pm 4.9}$	26.7$_{\pm 2.4}$	63.4$_{\pm 2.8}$	50.1$_{\pm 1.6}$
DyHead coco	All	Full	60.1	27.6	53.1	76.5	79.4	86.1	69.3	55.2	44.0	61.5	70.6	56.6	81.0	63.2
DyHead o365	1	Full	25.8$_{\pm 3.0}$	16.5$_{\pm 1.8}$	15.9$_{\pm 2.7}$	55.7$_{\pm 6.0}$	44.0$_{\pm 3.6}$	66.9$_{\pm 3.9}$	54.2$_{\pm 5.7}$	50.7$_{\pm 7.7}$	14.1$_{\pm 3.6}$	33.0$_{\pm 11.0}$	11.0$_{\pm 6.5}$	8.2$_{\pm 4.1}$	43.2$_{\pm 10.0}$	33.8$_{\pm 3.5}$
DyHead o365	3	Full	40.4$_{\pm 1.0}$	20.5$_{\pm 4.0}$	26.5$_{\pm 1.3}$	57.9$_{\pm 2.0}$	53.9$_{\pm 2.5}$	76.5$_{\pm 2.3}$	62.6$_{\pm 13.3}$	52.5$_{\pm 5.0}$	22.4$_{\pm 1.7}$	47.4$_{\pm 2.0}$	30.1$_{\pm 6.9}$	19.7$_{\pm 1.5}$	57.0$_{\pm 2.3}$	43.6$_{\pm 1.0}$
DyHead o365	5	Full	43.5$_{\pm 1.0}$	25.3$_{\pm 1.8}$	35.8$_{\pm 0.5}$	63.0$_{\pm 1.0}$	56.2$_{\pm 3.9}$	76.8$_{\pm 5.9}$	62.5$_{\pm 8.7}$	46.6$_{\pm 3.1}$	28.8$_{\pm 2.2}$	51.2$_{\pm 2.2}$	38.7$_{\pm 4.1}$	21.0$_{\pm 1.4}$	53.4$_{\pm 5.2}$	46.4$_{\pm 1.1}$
DyHead o365	10	Full	46.6$_{\pm 0.3}$	29.0$_{\pm 2.8}$	41.7$_{\pm 1.0}$	65.2$_{\pm 2.5}$	62.5$_{\pm 0.8}$	85.4$_{\pm 2.2}$	67.9$_{\pm 4.5}$	47.9$_{\pm 2.2}$	28.6$_{\pm 5.0}$	53.8$_{\pm 1.0}$	39.2$_{\pm 4.9}$	27.9$_{\pm 2.3}$	64.1$_{\pm 2.6}$	50.8$_{\pm 1.3}$
DyHead o365	All	Full	53.3	28.4	49.5	73.5	77.9	84.0	69.2	56.2	43.6	59.2	68.9	53.7	73.7	60.8
GLIP-T	1	Full	54.8$_{\pm 2.0}$	18.4$_{\pm 1.0}$	33.8$_{\pm 1.1}$	70.1$_{\pm 2.9}$	64.2$_{\pm 1.8}$	83.7$_{\pm 3.0}$	70.8$_{\pm 2.1}$	56.2$_{\pm 1.8}$	22.9$_{\pm 0.2}$	56.6$_{\pm 0.5}$	59.9$_{\pm 0.4}$	18.9$_{\pm 1.3}$	54.5$_{\pm 2.7}$	51.1$_{\pm 0.1}$
GLIP-T	3	Full	58.1$_{\pm 0.5}$	22.9$_{\pm 1.3}$	40.8$_{\pm 0.9}$	65.7$_{\pm 1.6}$	66.0$_{\pm 0.2}$	84.7$_{\pm 0.5}$	65.7$_{\pm 2.8}$	62.6$_{\pm 1.4}$	27.2$_{\pm 2.7}$	61.9$_{\pm 1.8}$	60.7$_{\pm 0.2}$	27.1$_{\pm 1.2}$	70.4$_{\pm 2.5}$	54.9$_{\pm 0.2}$
GLIP-T	5	Full	59.5$_{\pm 0.4}$	23.8$_{\pm 0.9}$	43.6$_{\pm 1.4}$	68.7$_{\pm 1.3}$	66.1$_{\pm 0.6}$	85.4$_{\pm 0.4}$	72.3$_{\pm 0.0}$	62.1$_{\pm 2.0}$	27.3$_{\pm 1.2}$	61.0$_{\pm 1.8}$	62.7$_{\pm 1.6}$	34.5$_{\pm 0.5}$	66.6$_{\pm 2.3}$	56.4$_{\pm 0.4}$
GLIP-T	10	Full	59.1$_{\pm 1.3}$	26.3$_{\pm 1.1}$	46.3$_{\pm 1.6}$	67.3$_{\pm 1.5}$	67.1$_{\pm 0.7}$	87.8$_{\pm 0.5}$	72.3$_{\pm 0.0}$	57.7$_{\pm 1.7}$	34.6$_{\pm 1.7}$	65.4$_{\pm 1.4}$	61.6$_{\pm 1.0}$	39.3$_{\pm 1.0}$	74.7$_{\pm 2.3}$	58.4$_{\pm 0.2}$
GLIP-T	All	Full	62.3	31.2	52.5	70.8	78.7	88.1	75.6	61.4	51.4	65.3	71.2	58.7	76.7	64.9
GLIP-L	1	Full	64.8$_{\pm 0.6}$	18.7$_{\pm 0.6}$	39.5$_{\pm 1.2}$	70.0$_{\pm 1.5}$	70.5$_{\pm 0.2}$	69.8$_{\pm 18.0}$	70.6$_{\pm 4.0}$	68.4$_{\pm 1.2}$	71.0$_{\pm 1.3}$	65.4$_{\pm 1.1}$	68.1$_{\pm 0.2}$	28.9$_{\pm 2.9}$	72.9$_{\pm 4.7}$	59.9$_{\pm 1.4}$
GLIP-L	3	Full	65.6$_{\pm 0.6}$	22.3$_{\pm 1.1}$	45.2$_{\pm 0.4}$	72.3$_{\pm 1.4}$	70.4$_{\pm 0.4}$	81.6$_{\pm 13.3}$	71.8$_{\pm 0.3}$	65.3$_{\pm 1.6}$	67.6$_{\pm 1.0}$	66.7$_{\pm 0.9}$	68.1$_{\pm 0.3}$	37.0$_{\pm 1.9}$	73.1$_{\pm 3.3}$	62.1$_{\pm 0.7}$
GLIP-L	5	Full	66.6$_{\pm 0.4}$	26.4$_{\pm 2.5}$	49.5$_{\pm 1.1}$	70.7$_{\pm 0.2}$	71.9$_{\pm 0.2}$	88.1$_{\pm 0.0}$	71.1$_{\pm 0.6}$	68.8$_{\pm 1.2}$	68.5$_{\pm 1.7}$	70.0$_{\pm 0.9}$	68.3$_{\pm 0.5}$	39.9$_{\pm 1.4}$	75.2$_{\pm 2.7}$	64.2$_{\pm 0.3}$
GLIP-L	10	Full	66.4$_{\pm 0.7}$	32.0$_{\pm 1.4}$	52.3$_{\pm 1.1}$	70.6$_{\pm 0.7}$	72.4$_{\pm 0.3}$	88.1$_{\pm 0.0}$	67.1$_{\pm 3.6}$	64.7$_{\pm 3.1}$	69.4$_{\pm 1.4}$	71.5$_{\pm 0.8}$	68.4$_{\pm 0.7}$	44.3$_{\pm 0.6}$	76.3$_{\pm 1.1}$	64.9$_{\pm 0.7}$
GLIP-L	All	Full	69.6	32.6	56.6	76.4	79.4	88.1	67.1	69.4	65.8	71.6	75.7	60.3	83.1	68.9
GLEE-Lite	1	Full	61.3$_{\pm 0.5}$	19.2$_{\pm 3.1}$	27.2$_{\pm 3.4}$	70.8$_{\pm 3.3}$	52.8$_{\pm 15.1}$	70.7$_{\pm 7.5}$	49.2$_{\pm 22.0}$	58.1$_{\pm 5.4}$	28.8$_{\pm 11.0}$	57.9$_{\pm 10.0}$	57.7$_{\pm 0.6}$	22.2$_{\pm 7.9}$	57.0$_{\pm 4.5}$	48.7$_{\pm 0.9}$
GLEE-Lite	3	Full	62.6$_{\pm 0.1}$	25.5$_{\pm 3.8}$	29.1$_{\pm 1.5}$	72.9$_{\pm 4.1}$	65.8$_{\pm 1.7}$	83.0$_{\pm 4.4}$	66.8$_{\pm 3.4}$	61.7$_{\pm 10.4}$	40.0$_{\pm 3.0}$	61.2$_{\pm 3.5}$	44.9$_{\pm 12.9}$	26.7$_{\pm 3.5}$	64.5$_{\pm 6.8}$	54.2$_{\pm 2.3}$
GLEE-Lite	5	Full	62.8$_{\pm 0.4}$	28.0$_{\pm 3.1}$	33.8$_{\pm 2.2}$	71.7$_{\pm 2.7}$	64.0$_{\pm 4.4}$	81.6$_{\pm 4.1}$	64.9$_{\pm 5.2}$	60.1$_{\pm 12.4}$	39.1$_{\pm 1.0}$	59.7$_{\pm 3.0}$	49.2$_{\pm 14.5}$	30.8$_{\pm 1.3}$	69.2$_{\pm 7.8}$	55.0$_{\pm 3.7}$
GLEE-Lite	10	Full	62.1$_{\pm 0.9}$	32.0$_{\pm 1.6}$	39.3$_{\pm 2.0}$	71.2$_{\pm 1.5}$	64.4$_{\pm 2.7}$	88.0$_{\pm 2.7}$	64.3$_{\pm 9.8}$	65.5$_{\pm 1.5}$	36.4$_{\pm 4.2}$	62.1$_{\pm 3.4}$	54.8$_{\pm 10.9}$	38.8$_{\pm 1.2}$	70.6$_{\pm 4.0}$	57.7$_{\pm 0.6}$
GLEE-Lite	All	Full	62.8	37.9	52.9	73.6	76.5	88.9	69.7	65.0	51.1	58.9	67.4	57.2	82.3	64.9
GLEE-Plus	1	Full	68.2$_{\pm 2.2}$	20.4$_{\pm 0.2}$	43.9$_{\pm 4.1}$	75.5$_{\pm 1.6}$	68.4$_{\pm 2.7}$	50.6$_{\pm 29.0}$	47.3$_{\pm 0.8}$	70.4$_{\pm 4.0}$	64.6$_{\pm 0.5}$	67.7$_{\pm 1.5}$	62.3$_{\pm 1.0}$	30.0$_{\pm 9.2}$	71.6$_{\pm 7.7}$	57.0$_{\pm 0.8}$
GLEE-Plus	3	Full	70.6$_{\pm 0.9}$	24.8$_{\pm 2.1}$	47.6$_{\pm 0.8}$	79.5$_{\pm 0.7}$	69.0$_{\pm 2.0}$	83.1$_{\pm 5.9}$	66.2$_{\pm 1.3}$	75.6$_{\pm 3.5}$	65.3$_{\pm 1.1}$	69.0$_{\pm 0.8}$	65.7$_{\pm 4.2}$	38.1$_{\pm 3.1}$	76.3$_{\pm 4.6}$	63.9$_{\pm 1.2}$
GLEE-Plus	5	Full	69.9$_{\pm 0.9}$	29.6$_{\pm 2.9}$	48.8$_{\pm 1.2}$	75.0$_{\pm 1.7}$	67.7$_{\pm 5.1}$	83.6$_{\pm 9.9}$	68.5$_{\pm 3.2}$	71.6$_{\pm 5.9}$	61.6$_{\pm 4.0}$	67.7$_{\pm 0.8}$	66.8$_{\pm 4.5}$	38.8$_{\pm 1.9}$	78.9$_{\pm 1.0}$	63.7$_{\pm 1.0}$
GLEE-Plus	10	Full	69.3$_{\pm 1.2}$	32.5$_{\pm 1.9}$	50.8$_{\pm 0.9}$	76.4$_{\pm 0.6}$	70.7$_{\pm 0.9}$	88.2$_{\pm 1.2}$	68.9$_{\pm 3.3}$	68.2$_{\pm 3.0}$	60.0$_{\pm 1.9}$	69.3$_{\pm 1.5}$	62.6$_{\pm 10.3}$	41.7$_{\pm 3.1}$	81.7$_{\pm 1.7}$	64.6$_{\pm 1.7}$
GLEE-Plus	All	Full	70.4	34.8	54.1	76.4	74.5	89.7	68.6	67.6	57.8	69.2	71.4	57.1	82.9	67.3
GLEE-Pro	1	Full	70.9$_{\pm 1.7}$	24.5$_{\pm 2.3}$	46.7$_{\pm 0.4}$	76.4$_{\pm 0.8}$	68.2$_{\pm 3.8}$	60.4$_{\pm 7.8}$	58.9$_{\pm 2.7}$	68.2$_{\pm 4.5}$	58.5$_{\pm 8.8}$	67.6$_{\pm 0.8}$	69.2$_{\pm 0.2}$	31.8$_{\pm 2.6}$	70.8$_{\pm 7.6}$	59.4$_{\pm 1.5}$
GLEE-Pro	3	Full	72.3$_{\pm 0.4}$	28.4$_{\pm 0.5}$	49.6$_{\pm 2.2}$	76.1$_{\pm 1.3}$	69.3$_{\pm 3.9}$	79.4$_{\pm 9.5}$	67.4$_{\pm 3.5}$	74.1$_{\pm 4.9}$	63.7$_{\pm 2.0}$	68.4$_{\pm 0.6}$	68.3$_{\pm 2.1}$	42.1$_{\pm 5.3}$	76.9$_{\pm 1.6}$	64.3$_{\pm 1.3}$
GLEE-Pro	5	Full	71.4$_{\pm 0.9}$	33.4$_{\pm 1.5}$	50.6$_{\pm 4.3}$	73.8$_{\pm 3.9}$	71.9$_{\pm 0.3}$	83.6$_{\pm 6.8}$	66.6$_{\pm 1.8}$	72.5$_{\pm 4.3}$	59.1$_{\pm 4.8}$	68.7$_{\pm 1.4}$	69.7$_{\pm 1.3}$	39.5$_{\pm 4.8}$	77.4$_{\pm 3.2}$	64.5$_{\pm 0.9}$
GLEE-Pro	10	Full	71.1$_{\pm 1.9}$	37.8$_{\pm 2.1}$	54.2$_{\pm 1.2}$	73.9$_{\pm 7.2}$	70.7$_{\pm 1.3}$	90.9$_{\pm 1.4}$	66.0$_{\pm 9.4}$	73.9$_{\pm 6.8}$	57.8$_{\pm 3.9}$	69.4$_{\pm 0.9}$	62.9$_{\pm 6.3}$	44.3$_{\pm 3.8}$	79.8$_{\pm 0.6}$	65.6$_{\pm 0.4}$
GLEE-Pro	All	Full	72.6	36.5	58.1	80.5	74.1	92.0	67.0	76.5	66.4	-	-	55.7	-	

Table 9. Per-dataset performance compared with DyHead, GLIP-T, and GLIP-L. For PascalVOC, we report the mAP (IoU=0.50:0.95) using the COCO evaluation script, to be consistent with other 12 datasets. "Full" denotes full-model tuning.

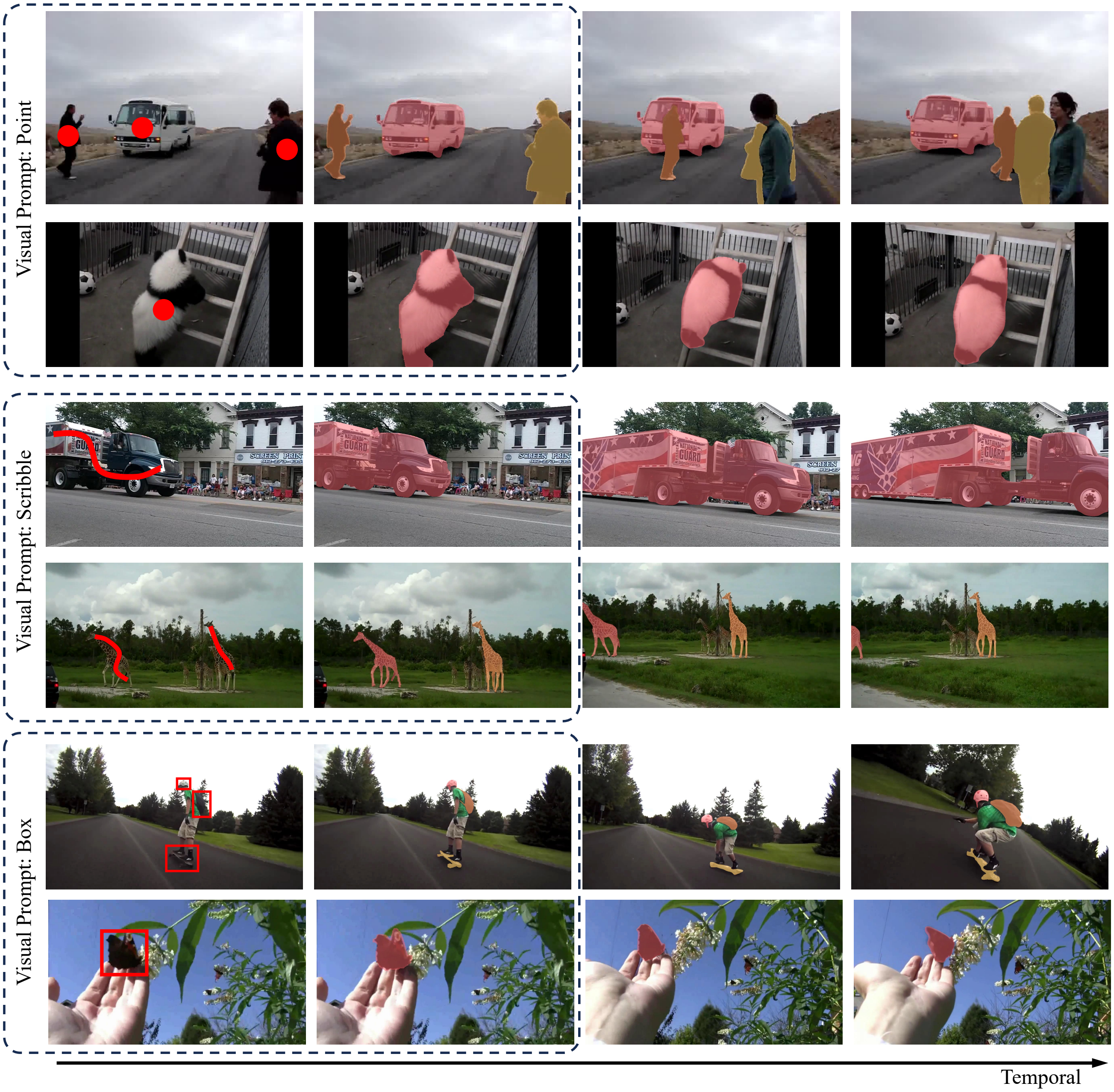

Figure 2. The visualization results of interactive segmentation and tracking. For image-level interactive segmentation, GLEE supports sending points, boxes, or scribbles as a visual prompts to the model, enabling direct segmentation of the specified object. In the case of video object segmentation, using the masked feature from the first frame as a prompt referring features allows segmentation of the corresponding object in subsequent frames of the video.

References

[1] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 2

[2] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 2

[3] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. 2

[4] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2302.01872*, 2023. 2

[5] Zihan Ding, Tianrui Hui, Shaofei Huang, Si Liu, Xuan Luo, Junshi Huang, and Xiaoming Wei. Progressive multimodal interaction network for referring video object segmentation. *The 3rd Large-scale Video Object Segmentation Challenge*, 2021. 3

[6] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928, 2021. 2

[7] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1, 4

[9] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 1

[10] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 1

[11] Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. In *NeurIPS*, 2022. 1, 4

[12] Dezhuang Li, Ruoqi Li, Lijun Wang, Yifan Wang, Jinqing Qi, Lu Zhang, Ting Liu, Qingquan Xu, and Huchuan Lu. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In *AAAI*, 2022. 3

[13] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 4

[14] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, 2023. 1

[15] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, pages 10965–10975, 2022. 4

[16] Chen Liang, Yu Wu, Tianfei Zhou, Wenguan Wang, Zongxin Yang, Yunchao Wei, and Yi Yang. Rethinking cross-modal interaction from a top-down perspective for referring video object segmentation. *arXiv preprint arXiv:2106.01061*, 2021. 3

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[18] Zhihui Lin, Tianyu Yang, Maomao Li, Ziyu Wang, Chun Yuan, Wenhao Jiang, and Wei Liu. Swem: Towards real-time video object segmentation with sequential weighted expectation-maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1362–1372, 2022. 2

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[20] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10437–10446, 2020. 1

[21] 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. 3

[22] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 1, 2

[23] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 2

[24] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 1

[25] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *International Journal of Computer Vision*, pages 1–18, 2022. 1, 2

[26] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020. 2

[27] 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. 1, 3

[28] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 1

[29] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by re-detection. In *CVPR*, 2020. 2

[30] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 2

[31] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 1, 2

[32] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4984, 2022. 3

[33] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2

[34] Ning Xu, Linjie Yang, Jianchao Yang, Dingcheng Yue, Yuchen Fan, Yuchen Liang, and Thomas S. Huang. Youtube-vis dataset 2021 version. https://youtube-vos.org/dataset/vis/. 1, 2

[35] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1, 2, 3

[36] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, 2023. 2, 3

[37] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2

[38] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 2019. 3

[39] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1

[40] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 1, 2

[41] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 2