OnlineRefer: A Simple Online Baseline for Referring Video Object Segmentation

Dongming Wu[1‡], Tiancai Wang[2], Yuang Zhang[3], Xiangyu Zhang[2,4], Jianbing Shen[5†]
[1] Beijing Institute of Technology, [2] MEGVII Technology, [3] Shanghai Jiao Tong University,
[4] Beijing Academy of Artificial Intelligence, [5] SKL-IOTSC, CIS, University of Macau
wudongming97@gmail.com, wangtiancai@megvii.com, shenjiangbingcg@gmail.com

Abstract

Referring video object segmentation (RVOS) aims at segmenting an object in a video following human instruction. Current state-of-the-art methods fall into an offline pattern, in which each clip independently interacts with text embedding for cross-modal understanding. They usually present that the offline pattern is necessary for RVOS, yet model limited temporal association within each clip. In this work, we break up the previous offline belief and propose a simple yet effective online model using explicit query propagation, named OnlineRefer. Specifically, our approach leverages target cues that gather semantic information and position prior to improve the accuracy and ease of referring predictions for the current frame. Furthermore, we generalize our online model into a semi-online framework to be compatible with video-based backbones. To show the effectiveness of our method, we evaluate it on four benchmarks, i.e., *Refer-Youtube-VOS, Refer-DAVIS$_{17}$, A2D-Sentences, and JHMDB-Sentences. Without bells and whistles, our OnlineRefer with a Swin-L backbone achieves **63.5 J&F** and **64.8 J&F** on Refer-Youtube-VOS and Refer-DAVIS$_{17}$, outperforming all other offline methods. Our code is available at https://github.com/wudongming97/OnlineRefer.*

1. Introduction

Given a natural language expression, the purpose of referring video object segmentation (RVOS) is to segment the described object in a streaming video. The emerging task has attracted great attention in the computer vision community as it provides potential benefits for many applications, *e.g.*, video editing and human-computer interaction. Its core challenge is associating all frames with constructing an efficient video representation, further promoting cross-modal understanding of two modalities, *i.e.*, video and language.

†Corresponding author: *Jianbing Shen*. This work was supported in part by the FDCT grants 0154/2022/A3 and SKL-IOTSC(UM)-2021-2023, the MYRG-CRG2022-00013-IOTSC-ICI grant and the SRG2022-00023-IOTSC grant. ‡The work is done during the internship at MEGVII.

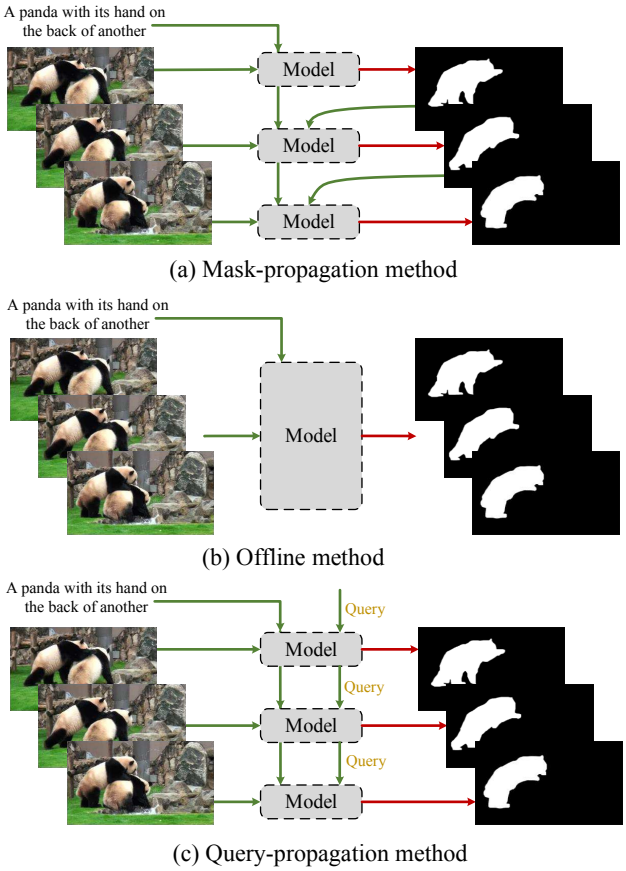
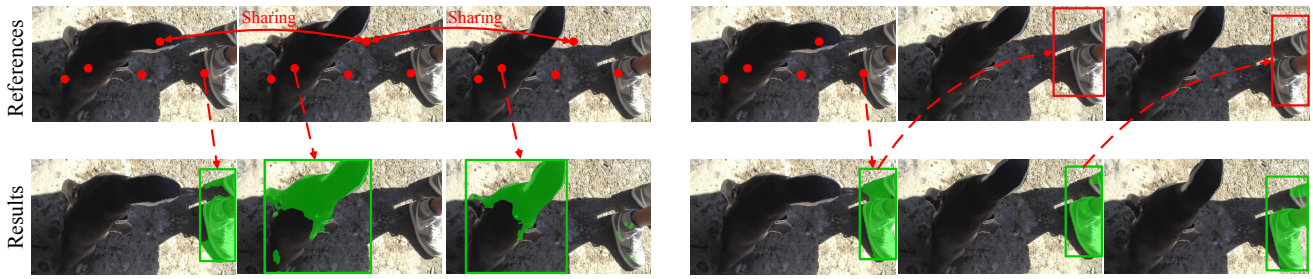


Figure 1: **Conceptual comparison on current methods**: (a) the mask-propagation method [12, 28], (b) the offline method [10, 1, 35], and (c) our query-propagation method.

Pioneer methods [12, 28] integrate mask propagation into the referring image segmentation in an online manner, as shown in Fig. 1 (a). However, the complexity and performance of their model remain far from satisfactory.

Recently, the state-of-the-art performance on RVOS has been dominated by offline methods [31, 10, 1, 4, 13, 35, 46]. They typically follow a clip-level paradigm, dividing the entire video into multiple non-overlapped clips and generating referring object masks for each clip, as illustrated in Fig. 1



(a) Query-sharing method (b) OnlineRefer

Figure 2: **Visualization of query references and corresponding results** of (a) query-sharing method and (b) our OnlineRefer. The reference points/boxes are marked red, while the final predictions of mask and box are marked green.

(b). In terms of different inter-frame interaction ways within an individual clip, existing offline methods [31, 10, 1, 4, 13, 35, 46] can be categorized into two groups: feature association methods and query-sharing methods. The former feature association methods [31, 10, 13, 33, 46] integrate multi-frame features into a holistic clip-level visual representation, which is further fused with text embedding for referring prediction. However, their temporal feature modeling is commonly complicated and heavy-weighted.

In contrast to the feature association methods, the query-sharing methods [1, 35] provide a simplified pipeline as they build on the query-based Transformer method [2, 47]. They first construct clip-level cross-modal features and then use a set of repeated queries to retrieve the same referent object from different frames. In other words, the cross-frame object correspondence relies heavily on sharing input queries. The interaction between frames is typically limited, hindering the association potential of the learned queries. Fig. 2 (a) shows a typical example: *all video frames share the same reference points (or queries)*, which misses the occluded object. In addition, due to resource limitations, the referring prediction has to be performed separately on each clip, which lacks inter-clip association.

In this paper, we propose a new and insightful online referring video object segmentation framework, OnlineRefer. It goes beyond the intuition of the online model not working well in RVOS. Its core idea is to take advantage of the query-based set prediction in Deformable DETR [47] and link all video frames via continuous query propagation. Specifically, we first provide a powerful query-based referring segmentation pipeline, which outputs the embedding representations of the referent object, further generating mask, box, and category. As these outputs gather rich target information, we propose a cross-frame query propagation module to transform them as new query inputs of the next frame. The propagation process has three significant advantages. **First**, the referring target is automatically associated with its precursors on all previous frames. **Second**, the box information of the last frame provides a very good spatial regional prior, benefiting the model for accurately inferring the same object in the current frame (see an example in Fig. 2 (b)). **Third**, our architecture avoids complicated temporal modeling or limited cross-frame association so that the overall training and inference progress is smooth and effective. Thanks to the remarkable performance, we expect to contribute the elegant and effective online model as a new baseline to the community.

To summarize, our main contributions are three-fold:

- We are the first to challenge the widespread belief that only offline models can deal well with RVOS and make online RVOS great again.
- We propose a simple yet solid online baseline based on query propagation. The explicit association across video frames facilities temporal target matching and improves referring prediction accuracy.
- Our method is evaluated on four benchmarks: Refer-Youtube-VOS, Refer-DAVIS$_{17}$, A2D-Sentences, and JHMDB-Sentences, outperforming all previous offline methods and achieving state-of-the-art performance.

2. Related Work

2.1. Referring Video Object Segmentation

Referring video object segmentation is to localize a text-referred object using a mask. Earlier works [12, 28] used the spatial-temporal memory mechanism [25], which stored the mask of the previous frames to promote referring image segmentation [44, 42, 17]. Currently, however, most existing methods [31, 30, 41, 10, 4, 13, 46, 33, 40, 3, 29, 39] concentrated on designing offline frameworks, *i.e.*, clip-in and clip-out. For example, Hui *et al*. [10] proposed a two-stream network, one branch being a temporal encoder to recognize the object motion and another branch being a spatial encoder to generate accurate referring segmentation. Wu *et al*. [33] additionally considered an object-level branch with salient object regions to enhance the foreground and background discriminability. These methods inevitably need a complicated spatial-temporal modeling module, which is not trivial. Several parallel works [1, 35] employed simple query-based Transformer models, which share the same

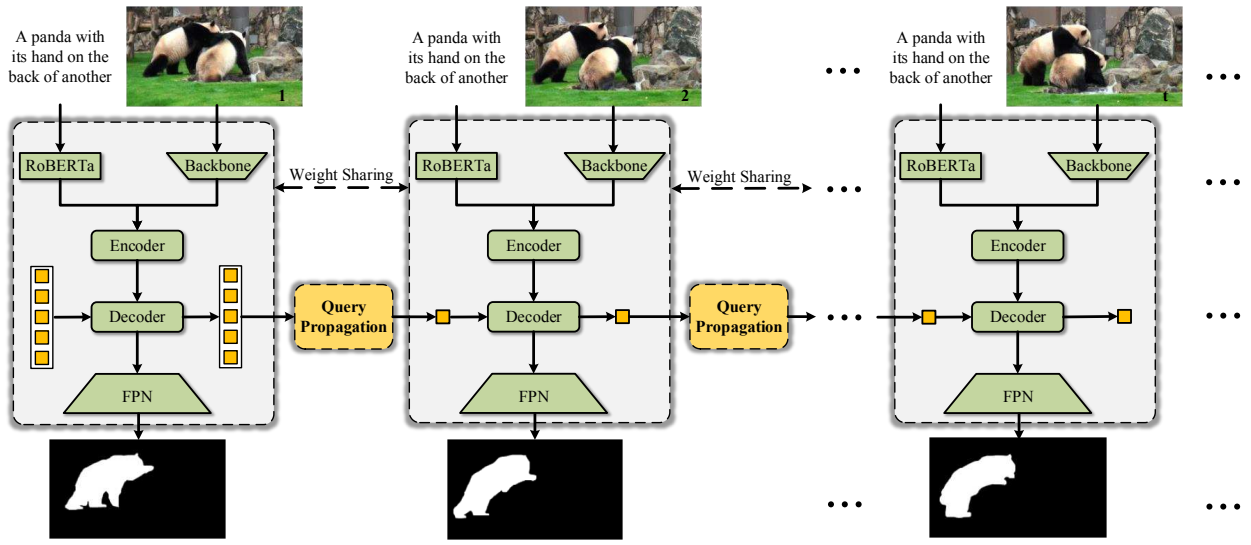


Figure 3: **Overall architecture of our OnlineRefer.** It consists of two main components: query-based referring segmentation and cross-frame query propagation. The query-based referring segmentation (§ 3.1) employs a set of queries to predict the referring object. The outputs contain rich target-aware information, so the cross-frame query propagation (§ 3.2) transforms them into a new query of the next frame. By repeating the two parts, we complete the RVOS task online frame-by-frame. OnlineRefer can also be generalized into a semi-online baseline using a clip-by-clip manner (§ 3.3).

query set across different frames. Precisely, MTTR [1] followed an instance-level segmentation pipeline to predict sequences of all instances and select the sequence that best fits the referent object. ReferFormer [35] transformed the input expression as the decoder queries for directly attending to the most relevant regions in the video frames.

Although our approach employs the basic query-based prediction architecture, there are two key differences. *First*, our model is entirely online and has an inherent advantage in handling long or ongoing videos, while offline methods fail due to the limitation of computation sources. *Second*, the target information of previous frames is explicitly and effectively used to strengthen cross-frame tracking and frame-wise referring segmentation, achieving better results.

2.2. Query-based Online Models

Employing query to associate cross-frame objects has been recently explored in several online models, such as TrackFormer [23], MOTR [45], IDOL [36], and InsPro [7]. They show the effectiveness and potential of query-based object association. However, they are different from our OnlineRefer from two perspectives. *On the one hand*, OnlineRefer does not need to detect multiple objects due to the guidance of language expression. TrackFormer [23] and MOTR [45, 34, 32] adopted an extra query subset to detect new-born objects. They require additional heuristic rules to combine two types of queries *i.e.*, *track query* and *detect query*. IDOL [36] designed a re-identification module as post-processing to link instances between frames. InsPro [7] kept the fixed query number and employed *empty query* to detect the new-born objects. These query propagation methods are also evaluated in § 4.5, while our method performs better. *On the other hand*, IDOL [36] and InsPro [7] designed additional training strategies with contrast learning to avoid identification switches or suppress duplicates. In contrast, our framework minimizes the gap between training and inference of long videos because it avoids the heuristic rules during the training stage.

3. Methodology

Given an input video and a natural language expression, our method aims to output binary masks of the referred object in a streaming way. The overall architecture of OnlineRefer is illustrated in Fig. 3. It comprises two essential parts: query-based referring segmentation and cross-frame query propagation. The query-based referring segmentation in § 3.1 is an advanced referring segmentation pipeline conditioned on the query set. The cross-frame query propagation in § 3.2 is to generate the input query set of the current frame by updating the outputs from the last frame. In addition, for training and inference on video-based backbones, OnlineRefer is extended into a semi-online pattern, which propagates the query across video clips in § 3.3.

3.1. Query-based Referring Sementation

Similar to ReferFormer [35], our query-based referring segmentation mainly follows the Deformable DETR detector [47], and we make several modifications on it for referring object prediction. It accepts a video frame, a language

expression, and a set of learnable queries as input. Its outputs are the target box, mask, category, and a set of output embeddings corresponding to the expression.

In specific, given the t^{th} frame $I_t \in \mathbb{R}^{3\times H\times W}$ and its corresponding expression S, we separately utilize visual and linguistic backbone to extract their features. The two features are mapped into the same dimension and fed into an encoder to perform cross-modal fusion using a cross-attention module before encoder layers. The generated cross-modal features contain critical target awareness, represented by $\boldsymbol{M}_t$. In the decoder, we define two types of queries: content query $\boldsymbol{q}_t^c \in \mathbb{R}^{N_t\times d}$ and position query (*i.e.*, position embedding) $\boldsymbol{q}_t^p \in \mathbb{R}^{N_t\times d}$, where N_t is the number of queries. Here, the content query follows the common usage of DETR [2], while the position query is transformed into base values of output boxes, denoted as $\boldsymbol{b}_t^{base} \in \mathbb{R}^{N_t\times 4}$, which *decreases prediction difficulty and benefits model convergence*. After that, the decoder transforms the queries and cross-modal features into output embedding $\boldsymbol{E}_t \in \mathbb{R}^{N_t\times d}$ (see Fig. 4 for more details).

On top of the output embedding, a 3-layer feed-forward network (FFN) is used to predict box offset $\boldsymbol{b}_t^{offset} \in \mathbb{R}^{N\times 4}$, which add on the base box coordinate to formulate the final box predictions, *i.e.*, $\boldsymbol{b}_t = \boldsymbol{b}_t^{base} + \boldsymbol{b}_t^{offset}$. Another 3-layer FFN generates class probabilities $\boldsymbol{c}_t \in \mathbb{R}^{N_t\times c}$, where c is the category number. For per-frame mask generation, we first employ a cross-modal FPN [35] to perform multi-scale interactions between linguistic features and visual feature maps. A new FFN then encodes the output embedding into parameters of the mask head, which performs three-layer 1×1 convolution on the generated FPN feature map, producing mask $\boldsymbol{m}_t \in \mathbb{R}^{N_t\times \frac{H}{4}\times \frac{W}{4}}$.

Since there is only one referent object in the video, we can find the best prediction as positive sample by minimizing the matching cost between predictions and ground truth:

$$\mathcal{L}_{match} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{box}\mathcal{L}_{box} + \lambda_{mask}\mathcal{L}_{mask}, \quad (1)$$

where $\mathcal{L}_{cls}$ is the class-related loss using the focal loss [16]. $\mathcal{L}_{box}$ represents the box-related loss that combines $\mathcal{L}_1$ loss and GIoU loss [27]. $\mathcal{L}_{mask}$ is the mask-related loss that sums up DICE loss [24] and binary mask focal loss. λ_{cls}, λ_{box} and λ_{box} are the corresponding loss coefficients. After completing the matching, we optimize the network using the loss function $\mathcal{L}_{match}$ for positive samples while letting negative samples predict the $\varnothing$ class.

3.2. Cross-frame Query Propagation

As described above, the decoder of query-based referring segmentation progressively refines the base coordinates from the position query into the final prediction along decoder layers. Inspired by this, we additionally consider the refinement domain in *temporal axis*, because the target box predicted from the last frame can be a better reference coordinate. Therefore, we propose the cross-frame query propagation, whose pipeline is illustrated in Fig. 4.

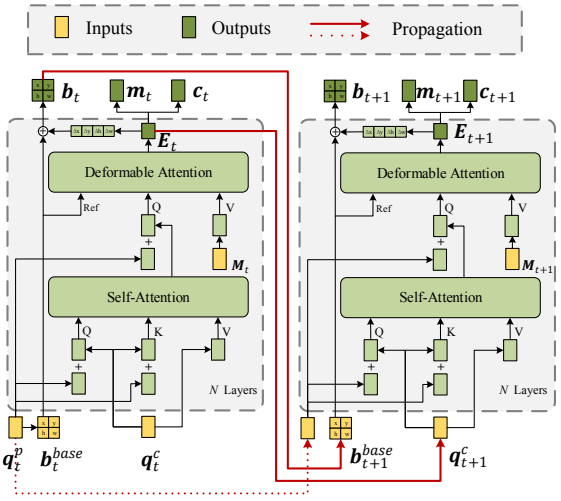


Figure 4: **Illustration of cross-frame query propagation**. The propagated representations consist of three cues: the output boxes, output embeddings, and position embeddings.

Given the outputs of the last frame, we first filter out informative representations that contain rich target awareness. In specific, we choose the query with the highest class score, and its index is denoted as $\hat{n}$:

$$\hat{n} = \arg\max_{n\in N_t}(\boldsymbol{c}_n). \quad (2)$$

Here, as the first frame follows the original setting [47] and uses $N_1 = 5$ learned queries, we can determine one query from multiple ones. This also leads to the subsequent frames containing only $N_t = 1$ query (*i.e.*, $t > 1$), which is retained across the entire video.

Once the index is obtained, we propagate three kinds of corresponding target cues from t^{th} frame to $(t+1)^{th}$ frame, including the prediction box, output embedding, and position embedding. The box and position embedding represent the explicit position information, which can be transformed as the base coordinate and position query of $(t+1)^{th}$ frame, which is seamlessly inserted query-based referring segmentation. The output embedding gathers the semantic information of the target, which is transformed as the content query of $(t+1)^{th}$ frame. Formally, the propagation process is:

$$\begin{aligned} \boldsymbol{b}_{t+1}^{base} &= \boldsymbol{b}_{t,\hat{n}} && \in \mathbb{R}^{1\times 4}, \\ \boldsymbol{q}_{t+1}^{p} &= \boldsymbol{q}_{t,\hat{n}}^{p} && \in \mathbb{R}^{1\times d}, \\ \boldsymbol{q}_{t+1}^{c} &= \mathcal{F}^{FFN}(\boldsymbol{E}_{t,\hat{n}}) && \in \mathbb{R}^{1\times d}, \end{aligned} \quad (3)$$

where $\mathcal{F}^{FFN}$ refers to one 3-layer FFN. Using the query propagation in multiple training frames, the matching cost is independently computed from each frame and the final loss is averaged by the frame number.

Discussion. In a streaming video, it is common for the referent object to enter our view in the middle frames. In other

Method	Backbone	Online/Offline	Refer-Youtube-VOS			Refer-DAVIS$_{17}$		
			$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
CMSA [42]	ResNet-50	online	34.9	33.3	36.5	34.7	32.1	37.2
CMSA + RNN [42]	ResNet-50	online	36.4	34.7	38.0	40.2	36.9	43.4
URVOS [28]	ResNet-50	online	47.2	45.2	49.1	51.6	47.2	55.9
MLSA [33]	ResNet-50	offline	49.7	48.4	50.9	57.9	53.8	62.0
ReferFormer [35]	ResNet-50	offline	55.6	54.8	56.5	58.5	**55.8**	61.3
OnlineRefer	ResNet-50	online	**57.3**	**55.6**	**58.9**	**59.3**	55.7	**62.9**
PMINet + CFBI [5]	Ensemble	offline	54.2	53.0	55.5	-	-	-
CITD [15]	Ensemble	offline	61.4	60.0	62.7	-	-	-
ReferFormer [35]	Swin-L	offline	62.4	60.8	64.0	60.5	57.6	63.4
OnlineRefer	Swin-L	online	**63.5**	**61.6**	**65.5**	**64.8**	**61.6**	**67.7**
MTTR (ω=12) [1]	Video-Swin-T	offline	55.3	54.0	56.6	-	-	-
ReferFormer (ω=5) [35]	Video-Swin-T	offline	59.4	58.0	60.9	-	-	-
ReferFormer (ω=5) [35]	Video-Swin-B	offline	62.9	**61.3**	64.6	61.1	58.1	64.1
OnlineRefer (ω=2)	Video-Swin-B	semi-online	**62.9**	61.0	**64.7**	**62.4**	**59.1**	**65.6**

Table 1: **The quantitative evaluation on Refer-Youtube-VOS and Refer-DAVIS$_{17}$**, with region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and average of $\mathcal{J}\&\mathcal{F}$. The best results are in bold and the second ones are underlined.

words, if an object is invisible in the first frame, our model will generate an *empty query* that contains no object for propagation. Despite this, OnlineRefer still handles well the entrance objects, as shown in the qualitative results of Fig. 6. This indicates that our cross-frame query propagation aims to provide a good prior while the cross-modal understanding in the query-based referring segmentation still plays an important role in referring prediction.

3.3. Extension to Semi-Online Model

In addition to the frame-by-frame pattern, we extend OnlineRefer into a more generalized framework that follows a clip-by-clip paradigm. Its primary motivation is to be capable of large video-based backbones. Different from the existing offline methods that independently process each clip, our approach provides query propagation between clips to achieve cross-clip object association. In this work, we define the new framework as *semi-online* method.

Formally, given the i^{th} video clip $V_i \in \mathbb{R}^{I\times 3\times H\times W}$, where I represents the clip length, we feed it into the query-based referring segmentation. Our semi-online model first extracts multi-frame visual features using a video-based backbone, *i.e.*, Video Swin Transformer [21], and perform cross-modal interaction between visual and linguistic embedding in the encoder. As the input query of the semi-online model is the same as our online model in the decoder, we then repeat the input query by I times to adapt to the multi-frame referring prediction. Thus, the semi-online model outputs multi-frame boxes $\boldsymbol{b}_i \in \mathbb{R}^{I\times N_i\times 4}$, masks $\boldsymbol{m}_i \in \mathbb{R}^{I\times N_i\times H\times W}$, and classes $\boldsymbol{c}_i \in \mathbb{R}^{I\times N_i\times c}$. The outputs can be regarded as N_i trajectory predictions on I frames. Finally, we find the positive sequence from N_i predictions by calculating the matching cost Eq. 1 and optimize the network. During the query propagation, we only transfer the high-score query of the last frame into the next clip.

4. Experiment

4.1. Dataset and Metric

Dataset. We evaluate our approach on four popular benchmarks: **Refer-Youtube-VOS** [28], **Refer-DAVIS$_{17}$** [12], **A2D-Sentences** [6], and **JHMDB-Sentences** [6]. Refer-Youtube-VOS expands the large-scale video object segmentation benchmark Youtube-VOS [38] using texutal desciptions. It consists of 3,975 videos and 27,899 expressions. Refer-DAVIS$_{17}$ extends another video object segmentation benchmark DAVIS$_{17}$ [26], which has 90 videos (60 for training and 30 for testing) and more than 1,500 expressions. A2D-Sentences and JHMDB-Sentences are developed by adding additional textual descriptions on the original action and actor datasets A2D [37] and JHMDB [11]. A2D-Sentences includes 3,782 videos and 6,655 expressions, where each video has 3-5 frames annotated with pixel-level segmentation mask. JHMDB-Sentences contains 928 videos, each being described by a corresponding expression (a total of 928 sentences).

Evaluation Metric. On Refer-Youtube-VOS and Refer-DAVIS$_{17}$, we employ region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and their average value $\mathcal{J}\&\mathcal{F}$ as our metrics. Since ground-truth annotations of Refer-Youtube-VOS validation are currently inaccessible, our predictions are uploaded to the official server for evaluation. On A2D-Sentences and JHMDB-Sentences, we adopt Overall IoU, Mean IoU, and precision@K to evaluate our method. Overall IoU is the ratio between the total intersection and the total union area over all the test samples. Mean IoU computes the averaged IoU over all the test samples. Precision@K measures the percentage of test samples whose IoU scores are higher than

Method	Backbone	Online/Offline	P0.5	P0.6	P0.7	P0.8	P0.9	Overall IoU	Mean IoU
Hu *et al*. [9]	VGG-16	offline	34.8	23.6	13.3	3.3	0.1	47.4	35.0
Gavrilyuk *et al*. [6]	I3D	offline	47.5	34.7	21.1	8.0	0.2	53.6	42.1
CMSA + CFSA [43]	ResNet-101	offline	48.7	43.1	35.8	23.1	5.2	61.8	43.2
ACAN [31]	I3D	offline	55.7	45.9	31.9	16.0	2.0	60.1	49.0
CMPC-V [18]	I3D	offline	65.5	59.2	50.6	34.2	9.8	65.3	57.3
ClawCraneNet [14]	ResNet-50/101	offline	70.4	67.7	61.7	48.9	17.1	63.1	59.9
MTTR (ω=10) [1]	Video-Swin-T	offline	75.4	71.2	63.8	48.5	16.9	72.0	64.0
ReferFormer (ω=5) [35]	Video-Swin-T	offline	82.8	79.2	72.3	55.3	19.3	77.6	69.6
ReferFormer (ω=5) [35]	Video-Swin-B	offline	83.1	80.4	74.1	57.9	21.2	78.6	70.3
OnlineRefer (ω=5)	Video-Swin-B	semi-online	**83.1**	80.2	73.4	56.8	**21.7**	**79.6**	**70.5**

Table 2: **The quantitative evaluation on A2D-Sentences**, with Precision@K, overall IoU and Mean IoU.

Method	Backbone	Online/Offline	P0.5	P0.6	P0.7	P0.8	P0.9	Overall IoU	Mean IoU
Hu *et al*. [9]	VGG-16	offline	63.3	35.0	8.5	0.2	0.0	54.6	52.8
Gavrilyuk *et al*. [6]	I3D	offline	69.9	46.0	17.3	1.4	0.0	54.1	54.2
CMSA + CFSA [43]	ResNet-101	offline	76.4	62.5	38.9	9.0	0.1	62.8	58.1
ACAN [31]	I3D	offline	75.6	56.4	28.7	3.4	0.0	57.6	58.4
CMPC-V [18]	I3D	offline	81.3	65.7	37.1	7.0	0.0	61.6	61.7
ClawCraneNet [14]	ResNet-50/101	offline	88.0	79.6	56.6	14.7	0.2	64.4	65.6
MTTR (ω=10) [1]	Video-Swin-T	offline	93.9	85.2	61.6	16.6	0.1	70.1	69.8
ReferFormer (ω=5) [35]	Video-Swin-T	offline	95.8	89.3	66.8	18.9	0.2	71.9	71.0
ReferFormer (ω=5) [35]	Video-Swin-B	offline	96.2	90.2	70.2	21.0	0.3	73.0	71.8
OnlineRefer (ω=5)	Video-Swin-B	semi-online	96.1	**90.4**	**71.0**	**21.9**	0.2	**73.5**	**71.9**

Table 3: **The quantitative evaluation on JHMDB-Sentences**, with Precision@K, overall IoU and Mean IoU.

a threshold K, where K$\in$[0.5, 0.6, 0.7, 0.8, 0.9].

4.2. Experiment Details

Model. We implement different visual backbones for feature extraction, such as ResNet [8], Swin Transformer [20], Video Swin Transformer [21]. RoBERTa [19] is adopted as the text encoder, while its parameters are frozen during the entire training stage. The feature maps of the last three stages are used in the encoder and FPN. We utilize 4 encoder layers and 4 decoder layers with dimension $d = 256$. The query number of the first frame is set to $N_1 = 5$. The coefficients for losses are $\lambda_{cls} = 2$, λ_{box} = 5, $\lambda_{mask} = 2$.

Training. AdamW optimizer [22] is used to optimize our model with an initial learning rate of $1e-5$, except for the visual backbone with a learning rate of $5e-6$. The training procedure runs for 6 epochs with the learning rate decays divided by 10 at the 3^{th} and 5^{th} epoch. The data augmentation techniques include random horizontal flip, random resize, random crop, and photometric distortion. Each frame is resized such that the shorter side at least has a size of 320 and the longer side at most has a size of 576.

For Refer-Youtube-VOS, we randomly sample 3 frames during training online models. The inputs of semi-online models are 3 clips, each one containing a window size of 2 (denoted as $\omega = 2$). In order to improve training stability, we feed 2 frames/clips into the online/semi-online model before the 4^{th} epoch. For Refer-DAVIS$_{17}$, we directly report the results using the model trained on Refer-Youtube-VOS

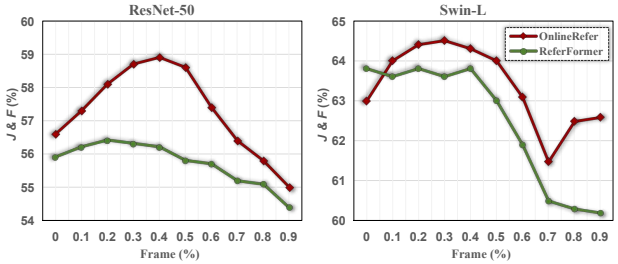


Figure 5: **Frame-wise J&F score** on Refer-Youtube-VOS. OnlineRefer has similar J&F accuracy with ReferFormer at the first frame but obtains better performance in subsequent frames benefiting from a good prior.

without fine-tuning. For A2D-Sentences, we use 2 training clips and increase into 3 clips at the 4^{th} epoch, where each clip has 5 frames (*i.e.*, $\omega = 5$) with the annotated target frame in the middle. For JHMDB-Sentences, the model trained on A2D-Sentences is directly employed for evaluation without fine-tuning. For a fair comparison, our model is pre-trained on Ref-COCO [44].

Inference. Since there is no gap between training and inference in our method, we directly output the predicted segmentation masks using the well-trained model without post-process. In the semi-online paradigm, the frame numbers of each clip remain the same with the training setup. We test inference speed on one Tesla V100 GPU over Refer-Youtube-VOS val set with an input size of 640$\times$320. More experiment details are included in supplementary materials.

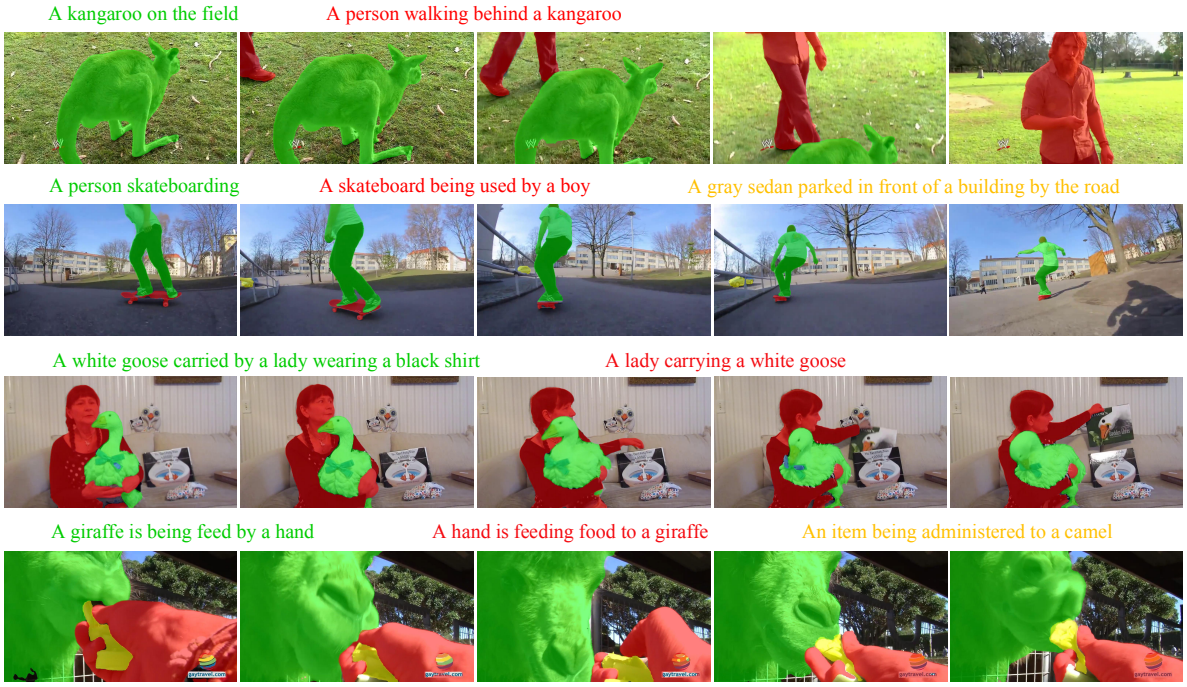


Figure 6: **Qualitative results from Refer-Youtube-VOS**. OnlineRefer accurately segments the referent object under various challenging scenes, *e.g.*, object occlusion or exit, appearance and size variation, and visually-similar objects.

4.3. Comparison to State-of-the-art

Refer-Youtube-VOS & Refer-DAVIS$_{17}$. We compare our method with existing approaches on Refer-Youtube-VOS and Refer-DAVIS$_{17}$, as shown in Table 1. Note that PMINet [5] and CITD [15] are top-2 solutions in 2021 Refer-Youtube-VOS Challenge. We can see that the earlier works [42, 28] usually employ an online manner, while the offline methods currently become mainstream due to better performance, such as MLSA [33], MTTR [1], and ReferFormer [35]. Surprisingly, our online model outperforms all offline methods on two datasets under all metrics. In specific, on Refer-Youtube-VOS, OnlineRefer with backbone ResNet-50 and Swin-L achieves $\mathcal{J}\&\mathcal{F}$ of 57.3 and 63.5. They are the highest accuracy among the models using the same backbone. Notably, our online model performance using Swin-L ($\mathcal{J}\&\mathcal{F}$: **63.5**) is higher than ReferFormer using Video Swin-B backbone ($\mathcal{J}\&\mathcal{F}$: 62.4). When the model is directly evaluated on Refer-DAVIS$_{17}$, it achieves the best scores ($\mathcal{J}\&\mathcal{F}$: **64.8**), which surpasses ReferFormer by a large margin. Overall, these impressive results sigificiantly demonstrate the effectiveness of the complete-online pipeline in referring video object segmentation.

Furthermore, attaching OnlineRefer on top of a Video Swin-B backbone formulates a semi-online model. The results are displayed in the last row of Table 1, which show that the new semi-online model leads to promising performance on two datasets, especially for the contour accuracy $\mathcal{F}$. This phenomenon also happens on other online models, which means that the regional prior of propagated boxes benefits producing high-quality segmentation masks. In addition, OnlineRefer with ResNet50 achieves 15.2 FPS, while SemiOnlineRefer with Video-Swin-B has 17.2 FPS.

Query Update	Position Update	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
×	×	49.8	49.5	50.1
✓	×	**57.3**	**55.8**	**58.8**
✓	✓	32.1	29.7	34.6

Table 4: **The update strategy of query and position** on Refer-Youtube-VOS, in terms of $\mathcal{J}$, $\mathcal{F}$.

In addition, we show the frame-wise $\mathcal{J}\&\mathcal{F}$ score curve on Refer-Youtube-VOS using ResNet-50 and Swin-L in Fig. 5. Both OnlineRefer and ReferFormer achieve similar accuracy in the first frame. But in subsequent frames, our OnlineRefer has a performance gain of around $1\sim2$ point. This clearly approves that the employment of explicit references can improve the referring segmentation performance.

A2D-Sentences & JHMDB-Sentences. We further present comparisons on the A2D-Sentences benchmark in Table 2. As the dataset is only annotated keyframes, existing methods generally follow an offline paradigm, which process clip-wise referring prediction without any cross-clip association. In contrast, our OnlineRefer is able to link all video clips by query propagation, *i.e.*, the semi-online framework. With the backbone Video Swin-B and window size of $\omega=5$, our semi-online model obviously exceeds all offline methods over IoU metrics and keeps enough competitiveness

Propagation Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
w/o Propagation	49.3	47.4	51.2
Concatenation	55.2	53.8	56.7
Fixed	55.0	53.6	56.5
Ours	**57.3**	**55.8**	**58.8**

(a) Comparison on propagation method.

Propagation Number	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Top-4	55.2	53.7	56.7
Top-3	55.9	54.4	57.4
Top-2	56.3	54.9	57.7
Top-1	**57.3**	**55.8**	**58.8**

(b) Comparison on propagation number.

Initial Queries	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
1	54.1	52.6	55.6
3	55.5	54.0	56.9
5	**57.3**	**55.8**	**58.8**
8	56.9	55.2	58.5

(c) Comparison on sampler frames.

Table 5: **Ablation studies of different propagation designs** on Refer-Youtube-VOS, in terms of region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and average of $\mathcal{J}\&\mathcal{F}$. The best results are in bold.

over precision metrics as well.

The semi-online model is directly evaluated on JHMDB-Sentences without finetuning to demonstrate the generality of our method. In Table 3, OnlineRefer achieves competitive performance compared to all other offline methods. In specific, OnlineRefer leads to higher IoU scores but comparable precision scores. Considering the clip gap on A2D-Sentences and JHMDB-Sentences, the above results demonstrate the potential of semi-online model.

4.4. Qualitative Results

In Fig. 6, we show several typical referring segmentation results of OnlineRefer from Refer-Youtube-VOS. The first video sequence is more challenging because the two referring objects become occluded and invisible in some frames. Taking the walking person as an example, it requires our online model to avoid the predicted empty box in the first frame causing the target missing in subsequent frames. Despite the difficulty, our OnlineRefer successfully segments out the target with sharp boundaries, showing strong correctness ability. In other scenes, the referring objects also face various challenges, such as appearance variation, pose deformation, and visually-similar objects. Otherwise, our OnlineRefer performs well in these difficult scenarios.

4.5. Ablation Study

To offer a deep insight into our OnlineRefer, we conduct ablation studies to analyze the effectiveness of each component. If not specialized, we report the online model performance on Refer-Youtube-VOS using ResNet-50.

Importance of query updating. To investigate the effect of updating strategy in Eq. 3, we perform experiments whether updating query and position embedding in Table 4. From the first row, discarding both query and position updating only achieves 49.8 over $\mathcal{J}\&\mathcal{F}$. After that, adding the query update achieves remarkable performance improvement (+6.7 on $\mathcal{J}\&\mathcal{F}$), and reaches the best score. However, updating position embedding largely hinders model performance (-25.2 on $\mathcal{J}\&\mathcal{F}$), as shown in the last row of Table 4. The problem demonstrates that fixed position embedding plays an important role in cross-frame object association.

Different propagation methods. We then analyze different query propagation methods in Table 5a. Removing propagation (*i.e.*, *w/o* propagation) leads to a frame-based referring segmentation pipeline, which results in a large performance drop (-6.0 on $\mathcal{J}\&\mathcal{F}$). 'Concatenation' represents concatenating the one propagated query and the initial query set, like MOTR [45] and TrackFormer [23]. 'Fixed' refers to giving up the query selection and keeping the initial number of queries, like InsPro [7]. Both two have a slight performance decrease. These results approve the effectiveness of our simple and heuristic-free query propagation.

Number of propagation query. It is also of interest to explore the impact of different propagation query numbers. As shown in Table 5b, we vary the query number from top-4 to top-1, where top-1 is our setting in OnlineRefer. Note that propagating top-5 queries equals the 'fixed' method in Table 5a. It is obvious that with the top query number decreasing, the performance on $\mathcal{J}\&\mathcal{F}$ is gradually improved. Overall, applying one query on cross-frame propagation is enough and significant for the inter-frame association.

Effect of initial queries. OnlineRefer starts with a set of initial queries in the first frame, which is further propagated across the entire video. To study its influence, we use a relatively small number of queries, as shown in Table 5c. We can see that fewer queries bring fewer proposals, further leading to lower $\mathcal{J}\&\mathcal{F}$ scores. However, the performance of more queries becomes flattened after $N=5$. Empirically, in this work, we set the initial query number as $N=5$.

5. Conclusion

In this paper, we proposed a simple, elegant, and strong baseline for online referring video object segmentation, named OnlineRefer. It broke up the widely accepted tradition that only offline models can handle well the challenging referring understanding task. OnlineRefer includes two crucial parts, query-based referring segmentation, and query propagation. The query-based referring segmentation outputs box, mask, and category based on input queries, while query generation part updates the output set as new queries. By iteratively using two parts, the objects in all video frames are automatically associated and predicted. To be compatible with video-based backbones, we developed a semi-online model that associates and predicts referent object clip by clip. The experiments are conducted on Refer-Youtube-VOS, Refer-DAVIS$_{17}$, A2D-Sentences and JHMDB-Sentences and our OnlineRefer shows the state-of-the-art performance on the four benchmarks.

References

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

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

[3] Weidong Chen, Dexiang Hong, Yuankai Qi, Zhenjun Han, Shuhui Wang, Laiyun Qing, Qingming Huang, and Guorong Li. Multi-attention network for compressed video referring object segmentation. In *ACM MM*, 2022.

[4] 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.

[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.

[6] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018.

[7] Fei He, Haoyang Zhang, Naiyu Gao, Jian Jia, Yanhu Shan, Xin Zhao, and Kaiqi Huang. Inspro: Propagating instance query and proposal for online video instance segmentation. In *NeurIPS*, 2020.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016.

[10] Tianrui Hui, Shaofei Huang, Si Liu, Zihan Ding, Guanbin Li, Wenguan Wang, Jizhong Han, and Fei Wang. Collaborative spatial-temporal modeling for language-queried video actor segmentation. In *CVPR*, 2021.

[11] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013.

[12] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, 2019.

[13] 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.

[14] Chen Liang, Yu Wu, Yawei Luo, and Yi Yang. Clawcranenet: Leveraging object-level relation for text-based video segmentation. *arXiv preprint arXiv:2103.10702*, 2021.

[15] 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.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[17] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, 2017.

[18] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *TPAMI*, 2021.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[21] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022.

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

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

[24] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[25] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[26] 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.

[27] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

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

[29] Mingjie Sun, Jimin Xiao, Eng GEE Lim, and Yao Zhao. Starting point selection and multiple-standard matching for video object segmentation with language annotation. *IEEE Transactions on Multimedia*, 2022.

[30] Hao Wang, Cheng Deng, Fan Ma, and Yi Yang. Context modulated dynamic networks for actor and action video segmentation with language queries. In *AAAI*, 2020.

[31] Hao Wang, Cheng Deng, Junchi Yan, and Dacheng Tao. Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In *ICCV*, 2019.

[32] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, 2023.

[33] Dongming Wu, Xingping Dong, Ling Shao, and Jianbing Shen. Multi-level representation learning with semantic alignment for referring video object segmentation. In *CVPR*, 2022.

[34] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-

object tracking. In *CVPR*, 2023.
[35] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022.
[36] 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.
[37] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *CVPR*, 2015.
[38] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.
[39] 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.
[40] Shilin Yan, Renrui Zhang, Ziyu Guo, Wenchao Chen, Wei Zhang, Hongyang Li, Yu Qiao, Zhongjiang He, and Peng Gao. Referred by multi-modality: A unified temporal transformer for video object segmentation. *arXiv preprint arXiv:2305.16318*, 2023.
[41] Zhao Yang, Yansong Tang, Luca Bertinetto, Hengshuang Zhao, and Philip HS Torr. Hierarchical interaction network for video object segmentation from referring expressions. In *BMVC*, 2021.
[42] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 2019.
[43] Linwei Ye, Mrigank Rochan, Zhi Liu, Xiaoqin Zhang, and Yang Wang. Referring segmentation in images and videos with cross-modal self-attention network. *TPAMI*, 2021.
[44] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016.
[45] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*, 2022.
[46] Wangbo Zhao, Kai Wang, Xiangxiang Chu, Fuzhao Xue, Xinchao Wang, and Yang You. Modeling motion with multi-modal features for text-based video segmentation. In *CVPR*, 2022.
[47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020.