LoSh: Long-Short Text Joint Prediction Network for Referring Video Object Segmentation

Linfeng Yuan, Miaojing Shi*, Zijie Yue, Qijun Chen
College of Electronic and Information Engineering, Tongji University
linfengyuan1997@gmail.com, {mshi,zijie,qjchen}@tongji.edu.cn

Abstract

Referring video object segmentation (RVOS) aims to segment the target instance referred by a given text expression in a video clip. The text expression normally contains sophisticated description of the instance's appearance, action, and relation with others. It is therefore rather difficult for a RVOS model to capture all these attributes correspondingly in the video; in fact, the model often favours more on the action- and relation-related visual attributes of the instance. This can end up with partial or even incorrect mask prediction of the target instance. We tackle this problem by taking a subject-centric short text expression from the original long text expression. The short one retains only the appearance-related information of the target instance so that we can use it to focus the model's attention on the instance's appearance. We let the model make joint predictions using both long and short text expressions; and insert a long-short cross-attention module to interact the joint features and a long-short predictions intersection loss to regulate the joint predictions. Besides the improvement on the linguistic part, we also introduce a forward-backward visual consistency loss, which utilizes optical flows to warp visual features between the annotated frames and their temporal neighbors for consistency. We build our method on top of two state of the art pipelines. Extensive experiments on A2D-Sentences, Refer-YouTube-VOS, JHMDB-Sentences and Refer-DAVIS17 show impressive improvements of our method. Code is available here.

1. Introduction

Referring video object segmentation (RVOS) [10] aims to segment the target instance in a video given a text expression, which can potentially benefit many video applications such as video editing and surveillance. For the research community, RVOS is a challenging and interesting multi-modal task: videos provide dense visual information while text expressions provide symbolic and structured linguistic information. This makes the alignment of the two modalities very challenging, especially for the segmentation task. A similar task to RVOS is the referring image segmentation (RIS) [25, 47, 51], which aims to segment the target instance in an image given a text expression. Compared to RIS, RVOS is significantly harder due to the difficulties to tackle the motion blur and occlusion in video frames. It is essential to build the data association across multiple frames and track the target instance in the video. Besides the visual difference between RIS and RVOS, the text expressions in them are also different: those in RIS mainly describe the appearance of the target instances in static images while those in RVOS describe both the appearance and action of the target instances over several frames in videos. Therefore, a robust RVOS model is expected to capture complex textual information from the input text expression and correspondingly align it with visual features in video frames.

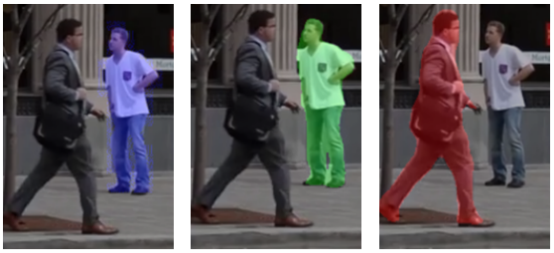

Figure 1. Qualitative comparison between our LoSh-M and MTTR [3]. The text expression for the target instance is 'a man in white t-shirt is walking'. LoSh-M generates an accurate prediction while MTTR predicts a wrong mask compared to ground truth.

Recently, the development of transformer in video feature extraction [23] and instance segmentation [41] benefits RVOS a lot. Especially, MTTR [3] and ReferFormer [44] are pioneering works to adapt query-based transformer models [5, 41, 52] to RVOS and have achieved significant improvements compared with previous methods. The performance gains in most query-based RVOS frameworks

*Corresponding author.

mainly come from the strong capabilities of transformer on feature extraction and multi-modal feature fusion while the exploitation of the linguistic part has not been emphasized.

The text expression for the target instance normally contains a sophisticated description of the instance (subject)'s appearance, action, and relation with others. Capturing all these attributes in a video presents a significant challenge for an RVOS model. We have analyzed the failure predictions by a state of the art model, MTTR [3], on the A2D-Sentences [10] dataset, whose IoUs with GTs are less than 0.5: we randomly sample 400 such cases and observe that over 70% of them either mis-align with appearance-related phrases or overly concentrate on the discriminative regions corresponding to actions or relations. This suggests that the model's mask prediction tends to favour more on the instance's action or relation with others rather than its appearance. This can lead to partial segmentation focusing on the action-related part of the target instance, or incorrect segmentation if the prediction mistakenly focuses on another instance that behaves similarly to the target instance. Fig. 1 illustrates an example of this: the video frame contains two men on the street. Given the text expression, 'a man in white t-shirt is walking', it refers to the man in a white t-shirt by the right of the frame. However, we run MTTR [3] on this video and observe that its prediction instead covers the walking man in a gray suit by the left of the frame. Apparently, MTTR has favoured more on the word 'walking' rather than 'white t-shirt' in the text expression.

To tackle this problem, our essential idea is to reduce the excessive impact of action/relation-related expression on the final mask prediction. We can first generate a subject-centric short text expression (*e.g.*, 'a man in white t-shirt') by removing the predicate- and object-related textual information from the original long text expression (*e.g.*, 'a man in white t-shirt is walking'). The short text expression should contain only the subject and the adjective/phrase that describes the subject. It is a more general expression for the referred instance compared to the long one. Given the input video, the mask prediction for the short text expression pays attention to the instance's appearance; while the mask prediction for the long text expression pays attention to both the instance's appearance and action, though the latter is often more favored. Ideally, the latter mask prediction should be included in the former. To take advantage of the relations between long and short expressions, in the feature level, we introduce a long-short cross-attention module to strengthen the feature corresponding to the long text expression via that from the short one; in the mask level, we introduce a long-short predictions intersection loss to regulate the model predictions for the long and short text expressions.

Apart from improving the RVOS model on the linguistic part, we also target to improve the model on the visual part by exploiting temporal consistency over visual features. Previous methods [45, 48] in RVOS assume the availability of optical flows between video frames and utilize them to generate auxiliary visual features to enhance video features. Different from them, we introduce a forward-backward visual consistency loss to directly compute optical flows between video frames and use them to warp the features of every annotated frame's temporal neighbors to the annotated frame for consistency optimization.

Overall, our main contribution is we develop a long-short text joint prediction network (LoSh) to segment the referred target instance under properly designed guidance from both long and short text expressions. Two components are emphasized, the long-short cross-attention module and long-short predictions intersection loss. Besides, we inject a forward-backward visual consistency loss into LoSh to exploit the feature-level temporal consistency between temporally adjacent visual features.

We evaluate our method on standard RVOS benchmarks, *i.e.*, A2D-Sentences [10], Refer-YouTube-VOS [31], JHMDB-Sentences [10] and Refer-DAVIS17 [17] datasets. We show that LoSh significantly outperforms state of the art across all metrics.

2. Related work

Referring video object segmentation. Previous RVOS methods are realized either in a single-stage or two-stage manner. Single-stage methods [1, 13, 31, 46] directly fuse the visual and linguistic features extracted from the input video and text expression; and generate the final mask prediction on top of the fused features with a pixel decoder. In contrast, two-stage methods[19, 20, 45] first generate a number of instance candidates in the video based on the visual features; then select the one that has the highest matching score with the input text expression as the final prediction. In general, two-stage methods perform better than single-stage ones. However, they also suffer from heavy workloads compared to single-stage ones.

Recently, owing to the success of query-based transformer models in computer vision [5, 41, 49, 52], a simple and unified framework, MTTR [3], is firstly introduced in RVOS. It is adapted from the query-based detection work, DETR [5], where a set of trainable object queries are utilized in the transformer decoder to generate predictions. Furthermore, ReferFormer [44] retains the transformer architecture but utilizes a much smaller set of object queries conditioned on text expressions. SgMg [27] uses segmentation optimizer to replace the spatial decoder in ReferFormer [44] and leverages spectrum information to guide the fusion of visual and textural features. TempCD [35] introduces a global referent token and uses collection and distribution mechanisms to interact information between the referent token and object queries. OnlineRefer [43] breaks up the offline belief in previous query-based RVOS frame-

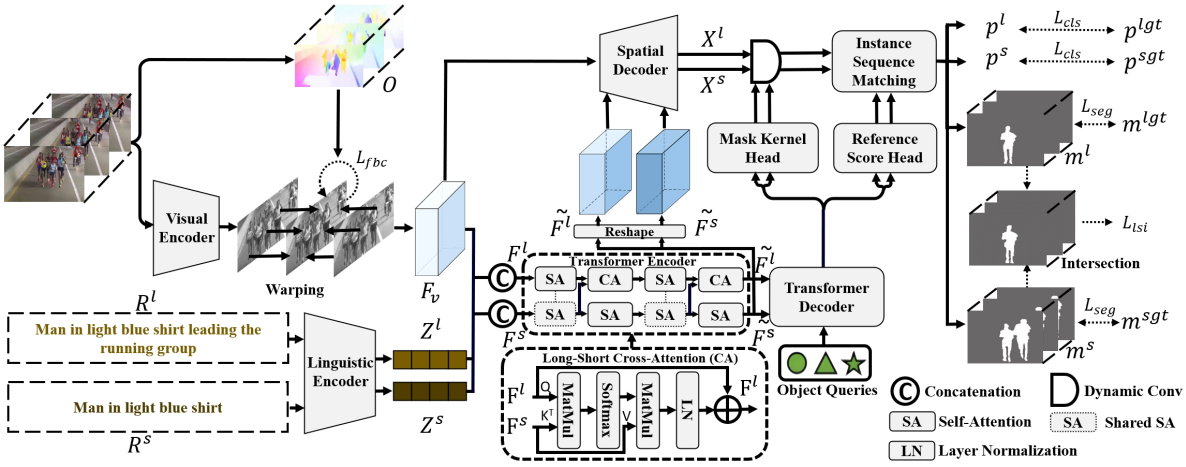

Figure 2. The overall pipeline of LoSh built upon the query-based model [3]. Our model takes long and short text expressions as text inputs and uses them to guide the target instance's segmentation in the given video. A long-short cross-attention module, a long-short predictions intersection loss ($\mathcal{L}_{lsi}$) and a forward-backward visual consistency loss ($\mathcal{L}_{fbc}$) are specifically introduced. Note that feed-forward networks in transformer encoder are omitted for simplicity.

works [3, 44] and proposes an online RVOS framework using explicit query propagation. There are other works focusing on temporal modeling based on the query-based transformer framework, *e.g.*, HTML [12] and SOC [24].

Without loss of generality, we can build our work on top of the recent query-based transformer frameworks given their efficiency and superiority. Prevailing methods typically utilize complete referring expressions and focus on exploiting the cross-modal feature interactions. Our idea instead utilizes a subject-centric short text expression from the original long expression for joint predictions. It shares some similarities to [21, 45], but having a closer look, they are fundamentally different. In [45], the idea of predicting the actor- and action-related scores can only work for the two-stage pipeline in which the instance proposals are given at the first stage. It can not be adapted to the single-stage pipeline by directly predicting the masks, as neither the actor- nor action-related words in text expressions are sufficient for the segmentation of the referred target instance. In [21], given the input text expression, it extracts features of individual words and classifies them as *entity*, *relation*, *etc*. The entity and relation features are obtained via the weighted combination of word features by using word classification probabilities as weights. However, the word classification in [21] is not supervised by ground truth word types but is implicitly supervised by the segmentation loss, which can not be fully reliable.

Transformer. Transformer [39] has a strong ability to draw long-term global dependencies and has achieved huge success in NLP tasks [4, 6]. After the introduction of vision transformer [8], transformer-based models have shown promising results in object detection [5, 52], semantic segmentation [11, 33], and multi-modal tasks [30, 50]. DETR [5] first introduces the query-based mechanism in object detection. It utilizes a set of object queries as box candidates and predicts the final box embeddings in the transformer decoder. In video instance segmentation, VisTR [41] employs the idea of DETR and models the task as a sequence prediction problem to perform natural instance tracking. Later on, MTTR [3], ReferFormer [44], SgMg [27], *etc*. extend the DETR [5] and VisTR [41] into the RVOS task and gain significant improvements.

3. Method

3.1. Problem setting

The input of RVOS framework consists of a video clip $\mathcal{V} = \{v_t\}_{t=1}^{T}$, $v_t \in \mathbb{R}^{3\times H\times W}$ with T frames and a text expression $\mathcal{R} = \{r_l\}_{l=1}^{L}$ with L words. The aim of this task is to generate pixel-wise mask predictions $\mathcal{M} = \{m_t\}_{t=1}^{T}$, $m_t \in \mathbb{R}^{H'\times W'}$ for the target instance referred by $\mathcal{R}$ in several frames of $\mathcal{V}$.

3.2. Preliminary

As shown in Fig. 2, we build the proposed LoSh upon state of the art query-based transformer model, MTTR [3]. Without loss of generality, LoSh can also be plugged into many other RVOS baselines. Below, we briefly review this query-based transformer framework for RVOS.

Visual encoder. Given the input frames in $\mathcal{V}$, visual features are extracted via a spatial-temporal visual encoder, namely the Video Swin transformer [23], such that $\mathcal{F}_v = \{f_t\}_{t=1}^{T}$, $f_t \in \mathbb{R}^{c_1\times h\times w}$ corresponding to the t-th frame in $\mathcal{V}$. This visual encoder is pre-trained on Kinetics-400 [15], which can simultaneously aggregate spatial and temporal information in a video.

Linguistic encoder. Given the input words in $\mathcal{R}$, linguistic features are extracted using the linguistic encoder,

RoBERTa [22], such that $\mathcal{Z} = \{z_l\}_{l=1}^{L}$, $z_l \in \mathbb{R}^{c_2}$ corresponding to the l-th word in $\mathcal{R}$. This linguistic encoder is pre-trained on several English-language corpora [37, 53].

Mask generation. Having $\mathcal{F}_v$ and $\mathcal{Z}$, their channel dimensions are scaled to $\mathcal{D}$ (*e.g.*, 256) by a 1×1 convolutional layer and a fully connected layer, respectively. Then, they are fused in the transformer encoder, which is devised on top of visual and linguistic encoders, to generate the multi-modal features, $\widetilde{\mathcal{F}}$.

In the transformer decoder, a set of trainable object queries are utilized to predict entity-related information. Each object query corresponds to a potential instance. The same query across multiple frames is trained to represent the same instance in the video. This design allows the natural tracking of each instance in the video. Supposing that the number of object queries is N, we can get N instance sequences, where the length of each sequence is T. On top of the decoded object queries, several lightweight heads are utilized to predict the reference scores and segmentation kernels. The reference score indicates the probability that the predicted instance is the referred target instance. The predicted segmentation kernels, inspired by other works in instance segmentation [36, 40], are used to convolve the text-related video features, $\mathcal{X}$. $\mathcal{X}$ is generated via a cross-modal FPN-like spatial decoder which takes the input of $\mathcal{F}_v$ and $\widetilde{\mathcal{F}}$, as illustrated in Fig. 2. For more details of this generation process, we refer the readers to [3].

3.3. Long-short text expressions

The long text expression ($\mathcal{R}^{\text{l}}$) for each video clip is given in the training set (*e.g.*, 'a man in a white t-shirt is walking'), which normally consists of a subject, predicate, object, some adjectives and descriptive phrases describing the subject or the object. We can generate the subject-centric short text expression ($\mathcal{R}^{\text{s}}$) by removing the predicate- and object-related contents in the long text expression. This can be either manually done or automatically achieved via a part-of-speech tagging method provided by [2]. For the latter, we can let the part-of-speech tagger identify the position of the first verb in a text expression and keep the words before it to create a subject-centric short text expression.

3.4. Long-short mask predictions

$\mathcal{R}^{\text{l}}$ and $\mathcal{R}^{\text{s}}$ are fed into the linguistic encoder simultaneously, resulting in text embeddings $\mathcal{Z}^{\text{l}}$ and $\mathcal{Z}^{\text{s}}$. As illustrated in Fig. 2, they are respectively concatenated with the same visual features $\mathcal{F}_v$ to generate $\mathcal{F}^{\text{l}}$ and $\mathcal{F}^{\text{s}}$. After the transformer encoder, we obtain multi-modal features $\widetilde{\mathcal{F}^{\text{l}}}$ and $\widetilde{\mathcal{F}^{\text{s}}}$, we interact each object query with them in the transformer decoder to obtain the soft mask predictions for the referred instance by $\mathcal{R}^{\text{l}}$ and $\mathcal{R}^{\text{s}}$, respectively. Sharing queries between $\widetilde{\mathcal{F}^{\text{l}}}$ and $\widetilde{\mathcal{F}^{\text{s}}}$ would facilitate the model training. The generated soft masks are denoted by $\mathcal{M}^{\text{l}} = \{m_t^{\text{l}}\}_{t=1}^{T}$ and $\mathcal{M}^{\text{s}} = \{m_t^{\text{s}}\}_{t=1}^{T}$, where m signifies a probability map after the sigmoid operation, each of its pixel value indicating the probability that this pixel belongs to the referred instance. Since $\mathcal{R}^{\text{l}}$ is a sophisticated description of the target instance, we observe $\mathcal{M}^{\text{l}}$ tends to favour more on the instance's action rather than appearance (see Fig. 1 and Fig. 3). To strengthen the appearance-related information in $\mathcal{F}^{\text{l}}$, we introduce a long-short cross-attention module within the transformer encoder to take advantage of the subject-centric appearance information encoded in $\mathcal{F}^{\text{s}}$.

Long-short cross-attention. The original transformer encoder performs a series of self-attention (SA) within $\mathcal{F}^{\text{l}}$ or $\mathcal{F}^{\text{s}}$ separately. In order to facilitate the interaction between $\mathcal{F}^{\text{l}}$ and $\mathcal{F}^{\text{s}}$, we inject a long-short cross-attention (CA) module to replace the even-numbered self-attention module of the transformer encoder. Specifically, the proposed long-short cross-attention module treats $\mathcal{F}^{\text{s}}$ as *key* and *value* while $\mathcal{F}^{\text{l}}$ as *query*. In this way, $\mathcal{F}^{\text{l}}$ can be strengthened by aggregating appearance-related information in $\mathcal{F}^{\text{s}}$, therefore alleviating the RVOS model's excessive focus on the action-related information of the target instance. We design the cross-attention to be uni-directional between $\mathcal{F}^{\text{s}}$ and $\mathcal{F}^{\text{l}}$ because our ultimate target is to utilize the auxiliary information in $\mathcal{F}^{\text{s}}$ to help accurately segment the instance referred by $\mathcal{R}^{\text{l}}$ ($\mathcal{F}^{\text{l}}$). Empirically, we also did not observe benefits by using $\mathcal{F}^{\text{l}}$ to strengthen $\mathcal{F}^{\text{s}}$ (see Sec. 4.4). We suggest the reason is this can conversely down-weight the appearance-related information in $\mathcal{F}^{\text{s}}$, which is the key information we care about in $\mathcal{F}^{\text{s}}$.

Overall, the new transformer encoder now has two pathways (see Fig. 2), one for $\mathcal{F}^{\text{s}}$ containing only SA modules same to the original version, one for $\mathcal{F}^{\text{l}}$ alternating between SA and CA modules.

3.5. Long-short predictions intersection

Given the same input video frames, the short text expression is the subject-centric part of the long one, making it a more generic expression of a certain subject than the longer one. They focus on different levels of details and the mask prediction for the short text expression should be ideally included in that for the long one. To regulate our network predictions to conform to this observation, we introduce a new long-short predictions intersection loss, specified below.

Given m_t^{l} and m_t^{s}, we first use a threshold (*e.g.*, 0.5) to filter out those pixels whose values are below it. We consider they are more likely to be the background pixels. Next, we compute the intersection between m_t^{l} and m_t^{s} by pixel-wisely multiplying them (after the background removal) and summing their products to obtain the probability weighted intersection area between them, *i.e.*, $\left|m_t^{\text{l}} \cap m_t^{\text{s}}\right|$. We also calculate the sum of the probabilities of foreground pixels in m_t^{l}, *i.e.*, $\left|m_t^{\text{l}}\right|$. Theoretically, the mask prediction by the long text expression should be included in that by

the short text expression; in other words, $\left|m_t^{\mathrm{l}} \cap m_t^{\mathrm{s}}\right|$ should be equivalent or at least very close to $\left|m_t^{\mathrm{l}}\right|$. In practice, $\frac{\left|m_t^{\mathrm{l}} \cap m_t^{\mathrm{s}}\right|}{\left|m_t^{\mathrm{l}}\right|}$ belongs to $[0, 1]$. In network optimization, we maximize $\frac{\left|m_t^{\mathrm{l}} \cap m_t^{\mathrm{s}}\right|}{\left|m_t^{\mathrm{l}}\right|}$ to encourage the agreement between the long-short predictions on the referred instance; in other words, enforcing a partial alignment between them. The long-short predictions intersection loss is written as:

$$\mathcal{L}_{lsi} = \sum_{t=1}^{T} \left(1 - \frac{\left|m_t^{\mathrm{l}} \cap m_t^{\mathrm{s}}\right| + \epsilon}{\left|m_t^{\mathrm{l}}\right| + \epsilon}\right) \tag{1}$$

We add a constant ϵ (1.0) to Eqn. 1 in case $\left|m_t^{\mathrm{l}}\right|$ is zero. $\mathcal{L}_{lsi}$ is integrated into the matching cost (below) to find the best-matched object query for the ground truth. For cases when m_t^{l} and m_t^{s} are completely disjoint during training, meaning at least one of them is mis-predicted, they will not only be penalized by $\mathcal{L}_{lsi}$, the mis-predicted one will also be penalized by its own segmentation loss (see Eqn. 5).

3.6. Long-short matching cost

According to the mask prediction process in Sec. 3.2, the query-based RVOS framework generates $T \times N$ predictions where T is the number of input frames and N is the number of object queries. The predictions of each object query maintain the same relative positions across different frames. The predictions for the i-th object query can be denoted by:

$$y_i = \left\{\left(p_{i,t}^{\mathrm{l}}, m_{i,t}^{\mathrm{l}}\right), \left(p_{i,t}^{\mathrm{s}}, m_{i,t}^{\mathrm{s}}\right)\right\}_{t=1}^{T} \tag{2}$$

where $p_{i,t}^{\mathrm{l}}$ ($p_{i,t}^{\mathrm{s}}$) is the probability that the i-th object query corresponds to the instance referred by $\mathcal{R}^{\mathrm{l}}$ ($\mathcal{R}^{\mathrm{s}}$) in the t-th frame. $m_{i,t}^{\mathrm{l}}$ ($m_{i,t}^{\mathrm{s}}$) is the corresponding mask prediction by the i-th query in the t-th frame given $\mathcal{R}^{\mathrm{l}}$ ($\mathcal{R}^{\mathrm{s}}$).

In contrast, the ground truth is represented by:

$$y^{gt} = \left\{\left(p_t^{lgt}, m_t^{lgt}\right), \left(p_t^{sgt}, m_t^{sgt}\right)\right\}_{t=1}^{T} \tag{3}$$

where p_t^{lgt} (p_t^{sgt}) equals to 1 when the referred instance is visible in the t-th frame, otherwise 0. m_t^{lgt} (m_t^{sgt}) is the ground truth mask for $\mathcal{R}^{\mathrm{l}}$ ($\mathcal{R}^{\mathrm{s}}$) in the t-th frame. In most cases, m_t^{lgt} and m_t^{sgt} are the same corresponding to one specific instance (so as p_t^{lgt} and p_t^{sgt}), though there exist cases when m_t^{sgt} corresponds to multiple instances (see also Sec. 4.2). We can handle both situations.

Given y^{gt}, the best-matched prediction y^* over $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ is found by minimizing a long-short matching cost:

$$y^* = \underset{y_i \in \mathcal{Y}}{\arg\min}\, \mathcal{L}_{lsm}\left(y_i, y^{gt}\right) \tag{4}$$

Based on the matching cost in query-based RVOS frameworks [3, 27], we develop the long-short matching cost as:

$$\begin{aligned} \mathcal{L}_{lsm}\left(y_i, y^{gt}\right) = \sum_{t=1}^{T} \Big[& \lambda_{lsi}\mathcal{L}_{lsi}\left(m_{i,t}^{\mathrm{l}}, m_{i,t}^{\mathrm{s}}\right) \\ & + \lambda_{cls}\left(\mathcal{L}_{cls}\left(p_{i,t}^{\mathrm{l}}, p_t^{lgt}\right) + \mathcal{L}_{cls}\left(p_{i,t}^{\mathrm{s}}, p_t^{sgt}\right)\right) \\ & + \lambda_{seg}\left(\mathcal{L}_{seg}\left(m_{i,t}^{\mathrm{l}}, m_t^{lgt}\right) + \mathcal{L}_{seg}\left(m_{i,t}^{\mathrm{s}}, m_t^{sgt}\right)\right) \Big] \end{aligned} \tag{5}$$

where $\mathcal{L}_{lsi}$ is the long-short predictions intersection loss introduced in Eqn. 1. $\mathcal{L}_{cls}$ signifies the binary classification loss while $\mathcal{L}_{seg}$ the segmentation loss. Each segmentation loss consists of a DICE loss and a mask focal loss according to MTTR [3]. λ_{lsi}, λ_{seg}, and λ_{cls} are hyper-parameters.

3.7. Forward-backward visual consistency

Previously, we focus on the linguistic aspect of the RVOS by employing long-short text expressions to exploit the spatial correlations between pixels (segments) within each frame. In this section, we focus on the visual aspect of the RVOS by employing the optical flow to exploit the temporal correlations between pixels across frames. Optical flow represents the motion of pixels between consecutive video frames, caused by the movement of instances or the camera. It plays an important role in video segmentation [7, 28, 38] and action recognition [32, 34]. Particularly, [7] devises a learnable module to estimate optical flow between two video frames and utilize it for occlusion modeling and feature warping. Inspired by it, and since there are nonconsecutive annotated frames in each video clip, we employ forward and backward optical flows to warp every annotated frame's neighbors to it for visual consistency.

Specifically, we first calculate the optical flows from a certain annotated frame (*e.g.*, k-th frame) to their adjacent four frames, which are denoted by $\mathcal{O} = \{o_{k\to k+t}\}_{t=-2}^{2}$, $o_{k\to k+t} \in \mathbb{R}^{H\times W}$. Unlike [7], we use the Farneback method [9] to directly compute $\mathcal{O}$ without learning given the video frames. The forward optical flows, $\mathcal{O}_f = \{o_{k\to k+t}\}_{t=1}^{2}$, are utilized to warp the visual features of $(k+t)$-th frames to the k-th frame; the backward optical flows, $\mathcal{O}_b = \{o_{k\to k-t}\}_{t=1}^{2}$ are utilized to warp the visual features of $(k-t)$-th frames to the k-th frame. We denote the warped features of k-th frame by $\mathcal{F}^w = \left\{f_{k+t\to k}^w\right\}_{t=-2}^{2}$ ($f_{k\to k}^w$ is equivalent to f_k). The warped features and the original visual feature of the k-th frame should be similar, we minimize their distances and write out our forward-backward visual consistency loss as:

$$\mathcal{L}_{fbc}\left(\mathcal{F}^w, f_k\right) = \sum_{t=-2}^{2} \left\| f_{k+t\to k}^w - f_k \right\|_2 \tag{6}$$

This loss enhances the representation of semantic- and motion-related information in the visual features $\mathcal{F}_v$.

3.8. Network training and inference

Network training. After getting the best-matched prediction y^*, the final loss function is a combination of the

matching cost in Eqn. 5 and the forward-backward consistency loss in Eqn. 6 with a hyperparameter λ_{fbc}:

$$\mathcal{L}\left(y^{*}, y^{gt}\right) = \mathcal{L}_{lsm}\left(y^{*}, y^{gt}\right) + \lambda_{fbc}\mathcal{L}_{fbc}\left(\mathcal{F}^{w}, f_{k}\right) \quad (7)$$

Network inference. We deactivate the prediction head for the short text expression but use the long text expression for inference. Following Sec. 3.2, LoSh generates N instance sequences, each containing reference scores and mask predictions for the predicted instance in T frames. We average the reference scores of predicted instance across frames. The sequence with the highest average reference score is regarded as the final prediction. The final mask predictions are obtained from the selected sequence.

4. Experiments

4.1. Datasets and evaluation metrics

We conduct experiments on four popular RVOS benchmarks: A2D-Sentences [10], Refer-YouTube-VOS [31], JHMDB-Sentences [10] and Refer-DAVIS17 [17]. A2D-Sentences contains 3,754 videos, with 3,017 for training, and 737 for testing. Each video has three annotated frames with pixel-wise segmentation masks for different target instances. It has 6,655 text expressions and each text expression corresponds to only one instance in the annotated frames of one video. Refer-YouTube-VOS [31] is the largest RVOS dataset, containing 3,978 videos with 15,009 text expressions. Each video in this dataset is annotated with pixel-wise instance segmentation masks for every fifth frame. We train our model on the training partition and obtain the results on the validation partition of Refer-YouTube-VOS. Furthermore, following [27, 44], the models trained on A2D-Sentences and Refer-YouTube-VOS are respectively evaluated on JHMDB-Sentences and Refer-DAVIS17. JHMDB-Sentences and Refer-DAVIS17 are extensions of JHMDB [14] and DAVIS17 [29] with additional text expressions. JHMDB-Sentences has 928 videos with 928 corresponding text expressions while Refer-DAVIS17 contains 90 videos with 1,544 expression sentences in total.

Following [3, 10], we adopt Overall IoU, Mean IoU and mAP to evaluate our model on A2D-Sentences and JHMDB-Sentences. Overall IoU calculates the ratio of the intersection and union of all predicted and annotated segment pixels in the test set. Mean IoU calculates the average of IOU results for each instance over all frames in the validation set. mAP computes the average of mAPs calculated under 10 IoU thresholds between 0.5 and 0.95 over testing samples. For Refer-YouTube-VOS and Refer-DAVIS17, we follow [44] to use region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$) and the average of them ($\mathcal{J}\&\mathcal{F}$). The calculation of $\mathcal{J}$ is the same as IoU between prediction and annotation. $\mathcal{F}$ is the F1-score calculated by the precision and recall of the contour of the predicted mask for the target instance.

4.2. Implementation details

We build the proposed LoSh upon two state of the art pipelines, MTTR [3] and SgMg [27] to obtain LoSh-M and LoSh-S, respectively.

LoSh-M. During training, we follow [3] to feed $w = 8$ frames around the annotated frame into the model. The input frames are resized such that the height is at least 360 and the width is at most 576. They are horizontally flipped with a probability of 0.5. In the transformer part, we utilize 4 encoder blocks and 3 decoder blocks with hidden dimension $D = 256$. The transformer encoder is modified according to Sec. 3.4. The number of object queries is set to 50. The losses weights in Eqn. 7 are set to 2, 5, 5, 0.1, for λ_{cls}, λ_{seg}, λ_{lsi}, and λ_{fbc}, respectively. Note that the focal loss in the segmentation loss term $\mathcal{L}_{seg}$ is with a factor 0.4. We set the learning rate as 10^{-4} and a batch size of 3. Except for λ_{lsi} and λ_{fbc}, other parameters are set the same as in [3].

LoSh-S. For LoSh-S, we only emphasize its main differences from LoSh-M. The number of input frames is 5 and the width of each frame is at most 640. We feed the fused multi-modal features into the deformable transformer, which has 4 deformable encoder and decoder blocks. The number of object queries is set to 5 and the linguistic features of the input text expression are added to each object query as in [27]. Following [27], we pre-train the model on RIS datasets [16, 26] and fine-tune it on RVOS datasets. To be consistent with [27], we add the same detection loss functions and weighting factors in LoSh-S. We refer the readers to [27] for more details.

Ground truth masks for short text expressions. In RVOS datasets, each video clip contains one or multiple target instances, each corresponding with a (long) text expression and mask annotation. For our extracted short text expressions, there exist a few cases (around 10% in RVOS datasets) when they correspond to multiple instances. In these cases, we simply merge the mask annotations of all the referred instances as the ground truth mask for certain short text expression.

4.3. Comparison with state of the art

A2D-Sentences. We first compare our model with state of the art methods on A2D-Sentences. As shown in Tab. 1, our LoSh-M built upon MTTR outperforms the baseline model by a large margin across all metrics. Specifically, LoSh-M shows +3.1 mAP, +2.7% Overall IoU and +3.1% Mean IoU gains compared with MTTR. We also build our LoSh on the previous best-performing method, SgMg. Following [27], we first pre-train our LoSh-S on RIS datasets [16, 26] and then fine-tune on A2D-Sentences. Our LoSh-S with Video-Swin-T shows clear gains, *e.g.*, +1.5 on mAP, +1.3% on Overall IoU and +1.2% on Mean IoU, over SgMg with the same backbone. Given a more powerful backbone (*i.e.*, Video-Swin-B), our proposed LoSh-S consistently outper-

Method	Backbone	A2D-Sentences			Refer-YouTube-VOS			Refer-DAVIS17		
		O-IoU	M-IoU	mAP	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Gavrilyuk *et al.* [10]	I3D	53.6	42.1	19.8	-	-	-	-	-	-
CMPC-V [21]	I3D	65.3	57.3	40.4	47.5	45.6	49.3	-	-	-
YOFO [18]	ResNet-50	-	-	-	48.6	47.5	49.7	53.3	48.8	57.8
MLRL [42]	ResNet-50	-	-	-	49.7	48.4	51.0	52.7	50.1	55.4
ReferFormer [44]	Video-Swin-B	78.6	70.3	55.0	62.9	61.3	64.6	61.1	58.1	64.1
MTTR ($w = 8$) [3]	Video-Swin-T	70.2	61.8	44.7	-	-	-	-	-	-
LoSh-M ($w = 8$)	Video-Swin-T	**72.9**	**64.9**	**47.8**	-	-	-	-	-	-
SgMg ($w = 5$) [27]	Video-Swin-T	78.0	70.4	56.1	62.0	60.4	63.5	61.9	59.0	64.8
LoSh-S ($w = 5$)	Video-Swin-T	**79.3**	**71.6**	**57.6**	**63.7**	**62.0**	**65.4**	**62.9**	**60.1**	**65.7**
SgMg ($w = 5$) [27]	Video-Swin-B	79.9	72.0	58.5	65.7	63.9	67.4	63.3	60.6	66.0
LoSh-S ($w = 5$)	Video-Swin-B	**81.2**	**73.1**	**59.9**	**67.2**	**65.4**	**69.0**	**64.3**	**61.8**	**66.8**

Table 1. Quantitative comparison with state of the art on A2D-Sentences [10], Refer-YouTube-VOS[31] and Refer-DAVIS17 [17]. O-IoU and M-IoU represent Overall IoU and Mean IoU. The number of input frames w follows the implementation details of [3, 27].

Method	IoU		mAP
	Overall	Mean	
Baseline	70.2	61.8	44.7
LoSh-M w/o Sh	71.1	62.6	45.4
LoSh-M w/o $\mathcal{L}_{lsi}$	72.3	63.8	46.1
LoSh-M w/o CA	72.5	64.5	47.3
LoSh-M w/ Inv-CA	72.5	64.3	47.2
LoSh-M w/ Bi-CA	72.8	64.7	47.6
LoSh-M w/ MSh	**72.9**	64.8	**47.8**
Losh-M (Ours)	**72.9**	**64.9**	**47.8**

Table 2. Ablation study for long-short text joint prediction.

forms SgMg and other state of the art on all metrics. We provide the results of these models evaluated on JHMDB-Sentences in the supplementary material.

Refer-YouTube-VOS and Refer-DAVIS17. As shown in Tab. 1, LoSh-S outperforms the baseline SgMg with Video-Swin-T: +1.7 on $\mathcal{J}\&\mathcal{F}$, +1.6 on $\mathcal{J}$ and +1.9 on $\mathcal{F}$. With a stronger backbone (*i.e.*, Video-Swin-B), LoSh-S also exceeds all the methods with a large margin.

To demonstrate the generalizability of LoSh-S, we follow [27, 44] to evaluate the trained LoSh-S from Refer-YouTube-VOS on Refer-DAVIS17 without fine-tuning. As shown in Tab. 1, LoSh-S also gains massive improvements on all metrics compared to its counterpart baseline.

4.4. Ablation studies

In this section, we conduct ablation studies on A2D-Sentences dataset using our model, LoSh-M.

4.4.1 Long-short text joint prediction

We first study the performance of LoSh-M using only the long text expression without the short text expression (LoSh-M w/o Sh), which is equivalent to our baseline MTTR plus the forward-backward visual consistency. The result is reported in Tab. 2: compared to LoSh-M, LoSh-M w/o Sh has a clear drop on the mAP and IoU, *e.g.*, 2.4 on mAP, 1.8% on Overall IoU and 2.3% on Mean IoU. This indicates the significant benefit of adding short text expression into RVOS.

Long-short cross-attention. Next, we present the result of LoSh without long-short cross-attention modules, *i.e.* LoSh-M w/o CA in Tab. 2. We observe a 0.5 decrease on mAP and 0.4% decreases on both Overall IoU and Mean IoU. Recalling that the proposed long-short cross-attention is a unidirectional attention mechanism from $\mathcal{F}^{\text{s}}$ to $\mathcal{F}^{\text{l}}$, we also offer variants of inverse and bidirectional cross-attentions, denoted by LoSh-M w/ Inv-CA and LoSh-M w/ Bi-CA. The former uses $\mathcal{F}^{\text{l}}$ to strengthen $\mathcal{F}^{\text{s}}$ while the latter is a combination of both the proposed CA and its inverse version. According to Tab. 2, both variants obtain inferior results than LoSh-M with the proposed CA (Ours). The result of LoSh-M w/ Inv-CA is even worse than that of LoSh-M without any CA (*i.e.*, LoSh-M w/o CA). These results prove the effectiveness of our proposed unidirectional cross-attention module (see discussion in Sec. 3.4).

Long-short predictions intersection loss. Next, we study the proposed long-short predictions intersection loss $\mathcal{L}_{lsi}$ by presenting a variant of LoSh-M without using $\mathcal{L}_{lsi}$, *i.e.*, LoSh-M w/o $\mathcal{L}_{lsi}$. The long and short text expressions are still used and long-short cross-attention still aggregates useful information from $\mathcal{F}^{\text{s}}$ to $\mathcal{F}^{\text{l}}$. Tab. 2 shows the result. We observe a 1.7 decrease on mAP from LoSh-M to LoSh-M w/o $\mathcal{L}_{lsi}$. Without using $\mathcal{L}_{lsi}$, the model can not regulate the predicted masks from the long and short text expressions.

Short text expressions generation. As mentioned in Sec. 3.3, the short text expressions can be manually or automatically extracted from long text expressions. For the latter, we feed the long text expressions into a part-of-speech tagger provided by the NLTK library [2]: we can extract the words before the first verb (including 'is') predicted by the tagger as short text expressions. We denote this variant of using machine-generated short text expression as LoSh-M w/ MSh and report its result in Tab. 2. It produces al-

Method	IoU		mAP
	Overall	Mean	
LoSh-M w/o $\mathcal{L}_{fbc}$	72.5	64.4	47.3
LoSh-M w/ $\mathcal{L}_{ofbc}$	72.6	64.7	47.5
LoSh-M w/ $\mathcal{L}_{mfbc}$	72.8	**64.9**	47.7
LoSh-M (Ours)	**72.9**	**64.9**	**47.8**

Table 3. Ablation study for forward-backward visual consistency.

Method	FLOPs	$\mathcal{T}_{\text{train}}$	$\mathcal{T}_{\text{infer}}$
MTTR	238.9G	141ms	79ms
LoSh-M	253.2G	153ms	80ms

Table 4. Computation cost, training and inference time for a sample. The resolution of the input frames is 320×576.

most the same mAP and relatively closed IoU as the original LoSh-M. Statistically, we verify that 98.7% of machine-generated short expressions are identical to our manually extracted ones. It demonstrates the flexibility of our LoSh and its potential to transfer to larger datasets.

4.4.2 Forward-backward visual consistency

We first present the result of LoSh without using the forward-backward visual consistency loss, *i.e.*, LoSh-M w/o $\mathcal{L}_{fbc}$ in Tab. 3. We observe a 0.5 decrease on mAP and 0.4% as well as 0.5% decreases on Overall IoU and Mean IoU, compared to LoSh-M. Merely integrating $\mathcal{L}_{fbc}$ to baseline MTTR (LoSh-M w/o Sh in Tab. 2) also leads to clear improvement, illustrating the effectiveness of $\mathcal{L}_{fbc}$.

Directed consistency. In Sec. 3.7, we warp the features from multiple adjacent frames to the annotated frame. On the contrary, we can also warp the feature of the annotated frame to its temporal neighbors and compute the consistency loss. We denote this variant of using opposite directions as $\mathcal{L}_{ofbc}$ and show the result in Tab. 3. The result can also be improved this way yet it is slightly inferior to $\mathcal{L}_{fbc}$, as we care more about the annotated frame. Besides, we also try to combine $\mathcal{L}_{ofbc}$ and $\mathcal{L}_{fbc}$ in the consistency, and denote this result of using mutual warping as $\mathcal{L}_{mfbc}$, the performance is basically the same to using the original $\mathcal{L}_{fbc}$, suggesting the mutual warping is unnecessary.

4.5. Analysis of computation cost

In Tab. 4, we provide the computation cost (*i.e.*, FLOPs) and training time of LoSh. Although LoSh introduces an additional branch to generate a mask for the short text expression, it actually incurs only a little additional computation cost (5% of that of the baseline) and training time (8% of that of the baseline). The reasons are that 1) short text expressions are on average one-third of the length of long ones; 2) around 90% of computation budgets are consumed by the visual encoder which is also only performed once in LoSh. Last, we also measure the time cost of optical flow calculation which is only performed during training. The calculation of optical flow consumes CPU time (74ms). It can happen in parallel when the GPU is processing the previous sample (153ms) or even offline, hence not reducing the overall training efficiency. Finally, the inference time is also provided, our LoSh-M basically consumes the same time to its baseline.

Figure 3. Qualitative comparison between LoSh-M and MTTR on A2D-Sentences. LoSh-M generates reasonable predictions while MTTR predicts incorrect or partial ones compared to ground truth.

4.6. Qualitative results

We also provide qualitative comparison between LoSh-M and baseline MTTR. Fig. 3 displays the cases where LoSh-M generates reasonable predictions while MTTR predicts incorrect or partial ones. This improvement is attributed to our model's enhanced attention on the appearance information. It can mitigate excessive impact of actions (*e.g.*, 'eating') or relations (*e.g.*, 'in the middle'). We provide more qualitative results in the supplementary material.

5. Conclusion

In this work, we propose LoSh, the long-short text joint prediction network for referring video object segmentation. We generate short text expressions from original long text expressions in RVOS and propose a long-short cross-attention module and a long-short predictions intersection loss to regulate the segmentation on the referred instance. Besides, a forward-backward visual consistency loss is also injected into LoSh to warp between the features of adjacent frames for visual consistency. Our proposed method can be easily plugged into many RVOS frameworks and brings no significant extra time during inference. Specifically, we build our method on top of two state of the art RVOS pipelines [3, 27], and achieve significant improvements over the previous best-performing methods.

References

[1] Miriam Bellver, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres, and Xavier Giro-i Nieto. A closer look at referring expressions for video object segmentation. *Multimedia Tools and Applications*, 82(3):4419–4438, 2023. 2

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 4, 7

[3] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 3

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[7] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *AAAI*, 2020. 5

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[9] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003. 5

[10] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018. 1, 2, 6, 7

[11] Ruohao Guo, Dantong Niu, Liao Qu, and Zhenbo Li. Sotr: Segmenting objects with transformers. In *ICCV*, 2021. 3

[12] Mingfei Han, Yali Wang, Zhihui Li, Lina Yao, Xiaojun Chang, and Yu Qiao. Html: Hybrid temporal-scale multimodal learning framework for referring video object segmentation. In *ICCV*, 2023. 3

[13] 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. 2

[14] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013. 6

[15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[16] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 6

[17] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, 2018. 2, 6, 7

[18] 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. 7

[19] 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. 2

[20] 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. 2

[21] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4761–4775, 2021. 3, 7

[22] 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. 4

[23] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022. 1, 3

[24] 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. *NeurIPS*, 2024. 3

[25] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 1

[26] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 6

[27] Bo Miao, Mohammed Bennamoun, Yongsheng Gao, and Ajmal Mian. Spectrum-guided multi-granularity referring video object segmentation. In *ICCV*, 2023. 2, 3, 5, 6, 7, 8

[28] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *CVPR*, 2018. 5

[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] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[31] 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

[32] Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J Black. On the integration of optical flow and action recognition. In *GCPR*, 2019. 5

[33] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 3

[34] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, 2018. 5

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

[36] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 4

[37] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*, 2018. 4

[38] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, 2016. 5

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

[40] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 4

[41] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 1, 2, 3

[42] Dongming Wu, Xingping Dong, Ling Shao, and Jianbing Shen. Multi-level representation learning with semantic alignment for referring video object segmentation. In *CVPR*, 2022. 7

[43] 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

[44] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022. 1, 2, 3, 6, 7

[45] Jianhua Yang, Yan Huang, Kai Niu, Linjiang Huang, Zhanyu Ma, and Liang Wang. Actor and action modular network for text-based video segmentation. *IEEE Transactions on Image Processing*, 31:4474–4489, 2022. 2, 3

[46] Linwei Ye, Mrigank Rochan, Zhi Liu, Xiaoqin Zhang, and Yang Wang. Referring segmentation in images and videos with cross-modal self-attention network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3719–3732, 2021. 2

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

[48] 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. 2

[49] Zijian Zhou, Miaojing Shi, and Holger Caesar. Hilo: Exploiting high low frequency relations for unbiased panoptic scene graph generation. In *ICCV*, 2023. 2

[50] Zijian Zhou, Miaojing Shi, and Holger Caesar. Vlprompt: Vision-language prompting for panoptic scene graph generation. *arXiv preprint arXiv:2311.16492*, 2023. 3

[51] Zijian Zhou, Oluwatosin Alabi, Meng Wei, Tom Vercauteren, and Miaojing Shi. Text promptable surgical instrument segmentation with vision-language models. *NeurIPS*, 2024. 1

[52] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 2, 3

[53] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015. 4