URVOS: Unified Referring Video Object Segmentation Network with a Large-Scale Benchmark

Seonguk Seo[1,†], Joon-Young Lee[2], and Bohyung Han[1]

[1] Seoul National University
[2] Adobe Research

Abstract. We propose a unified referring video object segmentation network (URVOS). URVOS takes a video and a referring expression as inputs, and estimates the object masks referred by the given language expression in the whole video frames. Our algorithm addresses the challenging problem by performing language-based object segmentation and mask propagation jointly using a single deep neural network with a proper combination of two attention models. In addition, we construct the first large-scale referring video object segmentation dataset called Refer-Youtube-VOS. We evaluate our model on two benchmark datasets including ours and demonstrate the effectiveness of the proposed approach. The dataset is released at https://github.com/skynbe/Refer-Youtube-VOS.

Keywords: video object segmentation, referring object segmentation

1 Introduction

Video object segmentation, which separates foreground objects from background in a video sequence, has attracted wide attention due to its applicability to many practical problems including video analysis and video editing. Typically, this task has been addressed in unsupervised or semi-supervised ways. Unsupervised techniques [31, 7] perform segmentation without the guidance for foreground objects, and aim to estimate the object masks using salient features, independent motions, or known class labels automatically. Due to the ambiguity and the lack of flexibility in defining foreground objects, such approaches may be suitable for video analysis but not for video editing that requires to segment arbitrary objects or their parts flexibly. In the semi-supervised scenario, where the ground-truth mask is available at least in a single frame, existing methods [2, 24, 35, 32, 29, 22] propagate the ground-truth object mask to the rest of frames in a video. They fit well for interactive video editing but require tedious and time-consuming step to obtain ground-truth masks. To overcome such limitations, interactive approaches [1, 3, 4, 21] have recently been investigated to allow user interventions during inference.

[†] This work was done during an internship at Adobe Research.

Despite great progress in semi-supervised and interactive video object segmentation, pixel-level interactions are still challenging especially in mobile video editing and augmented reality use-cases. To address the challenge, we consider a different type of interaction, language expressions, and introduce a new task that segments an object referred by the given language expression in a video. We call this task as *referring video object segmentation*.

A naïve baseline for the task is applying referring image segmentation techniques [16, 12, 37, 36] to each input frame independently. However, it does not leverage temporal coherency of video frames and, consequently, may result in inconsistent object mask predictions across frames. Another option is a sequential integration of referring image segmentation and semi-supervised video object segmentation. In this case, a referring image segmentation method initializes an object mask at a certain frame and then a video object segmentation method propagates the mask to the rest of the frames. This would work well if the initialization is successful. However, it often overfits to the particular characteristics in the anchor frame, which may not be robust in practice in the presence of occlusions or background clutter. Recently, Khoreva *et al.* [10] tackle this task by generating a set of mask proposals and choosing the most temporally-consistent set of candidates, but such a post-selection approach has inevitable limitation in maintaining temporal coherence.

We propose URVOS, a unified referring video object segmentation network. URVOS is an end-to-end framework for referring video object segmentation, which performs referring image segmentation and semi-supervised video object segmentation jointly in a single model. In this unified network, we incorporate two attention modules, cross-modal attention and memory attention modules, where memory attention encourages temporal consistency while cross-modal attention prevents drift. In addition, we introduce a new large-scale benchmark dataset for referring video object segmentation task, called *Refer-Youtube-VOS*. Our dataset is one order of magnitude larger than the previous benchmark [10], which enables researchers to develop new models and validate their performance. We evaluate the proposed method extensively and observe that our approach achieves outstanding performance gain on the new large-scale dataset.

Our contributions are summarized below.

- We construct a large-scale referring video object segmentation dataset, which contains $27,000+$ referring expressions for $3,900+$ videos.
- We propose a unified end-to-end deep neural network that performs both language-based object segmentation and mask propagation in a single model.
- Our method achieves significant performance gains over previous methods in the referring video object segmentation task.

2 Related Work

Referring Image Segmentation This task aims to produce a segmentation mask of an object in an input image given a natural language expression. Hu *et*

Full : *"A person on the right dressed in blue black walking while holding a white bottle."*
First : *"A woman in a blue shirt and a black bag."*

Full : *"A person showing his skateboard skills on the road."*
First : *"A man wearing a white cap."*

Fig. 1: Annotation examples of Refer-Youtube-VOS dataset. "Full" denotes that annotators watch the entire video for annotation while "First" means that they are given only the first frame of each video.

al. [8] first propose the task with a baseline algorithm that relies on multi-modal visual-and-linguistic features extracted from LSTM and CNN. RRN [12] utilizes the feature pyramid structures to take advantage of multi-scale semantics for referring image segmentation. MAttNet [37] introduces a modular attention network, which decomposes a multi-modal reasoning model into a subject, object and relationship modules, and exploits attention to focus on relevant modules. CMSA [36] employs cross-modal self-attentive features to bridge the attentions in language and vision domains and capture long-range correlations between visual and linguistic modalities effectively. Our model employs a variant of CMSA to obtain the cross-modal attentive features effectively.

Video Object Segmentation Video object segmentation is categorized into two types. Unsupervised approaches do not allow user interactions during test time, and aim to segment the most salient spatio-temporal object tubes. They typically employ two-stream networks to fuse motion and appearance cues [27, 13, 38] for learning spatio-temporal representations.

Semi-supervised video object segmentation tracks an object mask in a whole video, assuming that the ground-truth object mask is provided for the first frame. With the introduction of DAVIS [25] and Youtube-VOS [34] datasets, there has been great progress in this task. There are two main categories, online learning and offline learning. Most approaches rely on online learning, which fine-tunes networks using the first-frame ground-truth at test-time [2, 14, 24]. While the online learning achieves outstanding results, its computational complexity at test-time limits its practical use. Offline methods alleviate this issue and reduce runtime [35, 32, 29, 22]. STM [22] presents a space-time memory network by non-local matching between previous and current frames, which achieves state-of-the-art performance, even beating online learning methods. Our model also

Table 1: Datasets for referring video object segmentation. J-HMDB and A2D Sentences [6] focus on 'action' recognition along with 'actor' segmentation, which have different purposes than ours. Although Refer-DAVIS$_{16/17}$ are well-suited for our task, they are small datasets with limited diversity. Our dataset, Refer-Youtube-VOS, is the largest dataset containing objects in diverse categories.

Dataset	Target	Videos	Objects	Expressions
J-HMDB Sentences [6]	Actor	928	928	928
A2D Sentences [6]	Actor	3782	4825	6656
Refer-DAVIS$_{16}$ [10]	Object	50	50	100
Refer-DAVIS$_{17}$ [10]	Object	90	205	1544
Refer-Youtube-VOS (Ours)	Object	**3975**	**7451**	**27899**

belongs to offline learning, which modifies the non-local module of STM for its integration into our memory attention network and exploits temporal coherence of segmentation results.

Multi-modal Video Understanding The intersection of language and video understanding has been investigated in various areas including visual tracking [15], action segmentation [6, 30], video captioning [19] and video question answering [5]. Gavrilyuk *et al.* [6] adopt a fully-convolutional model to segment an actor and its action in each frame of a video as specified by a language query. However, their method has been validated in the datasets with limited class diversities, A2D [33] and J-HMDB [9], which only have 8 and 21 predefined action classes, respectively. Khoreva *et al.* [10] have augmented the DAVIS dataset with language referring expressions and have proposed a way to transfer image-level grounding models to video domain. Although [10] is closely related to our work, it fails to exploit valuable temporal information in videos during training.

3 Refer-Youtube-VOS Dataset

There exist previous works [6, 10] that constructed referring segmentation datasets for videos. Gavrilyuk *et al.* [6] extended the A2D [33] and J-HMDB [9] datasets with natural sentences; the datasets focus on describing the 'actors' and 'actions' appearing in videos, therefore the instance annotations are limited to only a few object categories corresponding to the dominant 'actors' performing a salient 'action'. Khoreva *et al.* [10] built a dataset based on DAVIS [25], but the scales are barely sufficient to learn an end-to-end model from scratch.

To facilitate referring video object segmentation, we have constructed a large-scale video object segmentation dataset, Youtube-VOS [34], with referring expressions. Youtube-VOS has 4,519 high-resolution videos with 94 common object categories. Each video has pixel-level instance segmentation annotation at every 5 frames in 30-fps videos, and their durations are around 3 to 6 seconds.

We employed Amazon Mechanical Turk to annotate referring expressions. To ensure the quality of the annotations, we selected around 50 turkers after a validation test. Each turker was given a pair of videos, the original video and the mask-overlaid one with the target object highlighted, and was asked to provide a discriminative sentence within 20 words that describes the target object accurately. We collected two kinds of annotations, which describe the highlighted object (1) based on a whole video (Full-video expression) and (2) using only the first frame of the video (First-frame expression). After the initial annotation, we conducted verification and cleaning jobs for all annotations, and dropped objects if an object cannot be localized using language expressions only. The followings are the statistics and analysis of the two annotation types of the dataset after the verification.

Full-video expression Youtube-VOS has 6,459 and 1,063 unique objects in train and validation split, respectively. Among them, we cover 6,388 unique objects in 3,471 videos ($6,388/6,459 = 98.9\%$) with 12,913 expressions in train split and 1,063 unique objects in 507 videos ($1,063/1,063 = 100\%$) with 2,096 expressions in validation split. On average, each video has 3.8 language expressions and each expression has 10.0 words.

First-frame expression There are 6,006 unique objects in 3,412 videos ($6,006/6,459 = 93.0\%$) with 10,897 expressions in train split and 1,030 unique objects in 507 videos ($1,030/1,063 = 96.9\%$) with 1,993 expressions in validation split. The number of annotated objects is lower than that of the full-video expressions because using only the first frame makes annotation more ambiguous and inconsistent and we dropped more annotations during the verification. On average, each video has 3.2 language expressions and each expression has 7.5 words.

Dataset analysis Fig. 1 illustrates examples of our dataset and shows the differences between two annotation types. The full-video expressions can use both static and dynamic information of a video while the first-frame expressions focus mostly on appearance information. We also provide the quantitative comparison of our dataset against the existing ones in Table 1, which presents that our dataset contains much more videos and language expressions.

4 Unified Referring VOS Network

Given a video with N frames and a language query Q, the goal of referring video object segmentation is to predict binary segmentation masks for the object(s) corresponding to the query Q in the N frames. As mentioned earlier, a naïve approach is to estimate the mask for each frame independently. However, a direct application of image-based solutions to referring object segmentation [12, 18, 36, 37] would fail to exploit valuable information, temporal coherence across the frames. Therefore, we cast the referring video object segmentation task as a

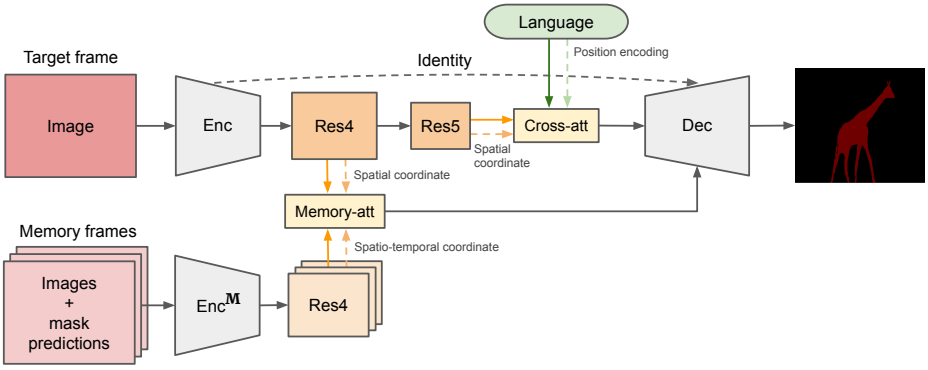


Fig. 2: The overall architecture of our framework. We employ ResNet-50 as our encoder and use the 4^{th} and the 5^{th} stage features (Res4 and Res5) to estimate memory and cross-modal attention, respectively. Both memory-attentive and cross-modal attentive features are progressively combined in the decoder.

joint problem of referring object segmentation in an image [12, 18, 36, 37] and mask propagation in a video [32, 22].

4.1 Our Framework

We propose a unified framework that performs referring image segmentation and video object segmentation jointly. Given a video and a referring expression, our network estimates an object mask in an input frame using the linguistic referring expression and the mask predictions in the previous frames. We iteratively process video frames until the mask predictions in all frames converge. Fig. 2 illustrates the overall architecture of our network.

Visual Encoder We employ ResNet-50 as our backbone network to extract visual features from an input frame. To include spatial information of the visual feature, we augment 8-dimensional spatial coordinates following [8]. Formally, let $\mathbf{F} \in \mathbb{R}^{H \times W \times C_{\mathbf{f}}}$ and $\mathbf{f}_p \in \mathbb{R}^{C_{\mathbf{f}}}$ denote a visual feature map[3] and a sliced visual feature at a certain spatial location p on $\mathbf{F}$, where $p \in \{1, 2, ..., H \times W\}$. We concatenate the spatial coordinates to the visual features $\mathbf{f}_p$ to obtain location-aware visual features $\bar{\mathbf{f}}_p$ as follows.

$$\bar{\mathbf{f}}_p = [\mathbf{f}_p; \mathbf{s}_p] \in \mathbb{R}^{C_{\mathbf{f}}+8}, \tag{1}$$

where $\mathbf{s}_p$ is a 8-dimensional spatial coordinate features[4].

[3] We use Res5 and Res4 feature maps in our model.

[4] For each spatial grid (h, w), $\mathbf{s}_p = [h_{\text{min}}, h_{\text{avg}}, h_{\text{max}}, w_{\text{min}}, w_{\text{avg}}, w_{\text{max}}, \frac{1}{H}, \frac{1}{W}]$, where $h_*, w_* \in [-1, 1]$ are relative coordinates of the grid. H and W denotes the height and width of the whole spatial feature map.

Language Encoder Given a referral expression, a set of words in the expression is encoded as a multi-hot vector and projected onto an embedding space in C_e dimensions using a linear layer. To model the sequential nature of language expressions while maintaining the semantics in the expression, we add positional encoding [28] at each word position. Let $\mathbf{w}_l \in \mathbb{R}^{C_\mathbf{e}}$ and $\mathbf{p}_l \in \mathbb{R}^{C_\mathbf{e}}$ denote the embeddings for the l-th word and the position of the expression, respectively. Our lingual feature is obtained by the sum of the two embedding vectors, *i.e.*, $\mathbf{e}_l = \mathbf{w}_l + \mathbf{p}_l \in \mathbb{R}^{C_\mathbf{e}}$.

Cross-modal Attention Module Using both visual and lingual features, we produce a joint cross-modal feature representation by concatenating the features in both the domains. Unlike [36], we first apply self-attention to each feature independently before producing a joint feature to capture complex alignments between both modalities effectively. Each self-attention module maps each feature to a C_a-dimensional space for both modalities as follows:

$$\widehat{\mathbf{f}}_p = \mathrm{SA}^{\mathrm{vis}}(\mathbf{f}_p) \in \mathbb{R}^{C_\mathrm{a}}, \quad \widehat{\mathbf{e}}_l = \mathrm{SA}^{\mathrm{lang}}(\mathbf{e}_l) \in \mathbb{R}^{C_\mathrm{a}} \tag{2}$$

where $\mathrm{SA}^*(\cdot)$ $(* \in \{\mathrm{vis}, \mathrm{lang}\})$ denotes a self-attention module for each domain. Then a joint cross-modal feature at each spatial position p and each word position l is given by

$$\mathbf{c}_{pl} = [\widehat{\mathbf{f}}_p; \widehat{\mathbf{e}}_l] \in \mathbb{R}^{C_a + C_a}. \tag{3}$$

We collect all cross-model features $\mathbf{c}_{pl}$ and form a cross-modal feature map as $\mathbf{C} = \{\mathbf{c}_{pl} \mid \forall p, \forall l\} \in \mathbb{R}^{H \times W \times L \times (C_a + C_a)}$.

The next step is to apply self-attention to this cross-modal feature map $\mathbf{C}$. Fig. 3(a) iluustrates our cross-modal attention module. We generate a set of (key, query, value) triplets, denoted by $(\mathbf{k}, \mathbf{q}, \mathbf{v})$, using 2D convolutions as follows:

$$\mathbf{k} = \mathrm{Conv}_{\mathrm{key}}(\mathbf{C}) \in \mathbb{R}^{L \times H \times W \times C_\mathrm{a}} \tag{4}$$

$$\mathbf{q} = \mathrm{Conv}_{\mathrm{query}}(\mathbf{C}) \in \mathbb{R}^{L \times H \times W \times C_\mathrm{a}} \tag{5}$$

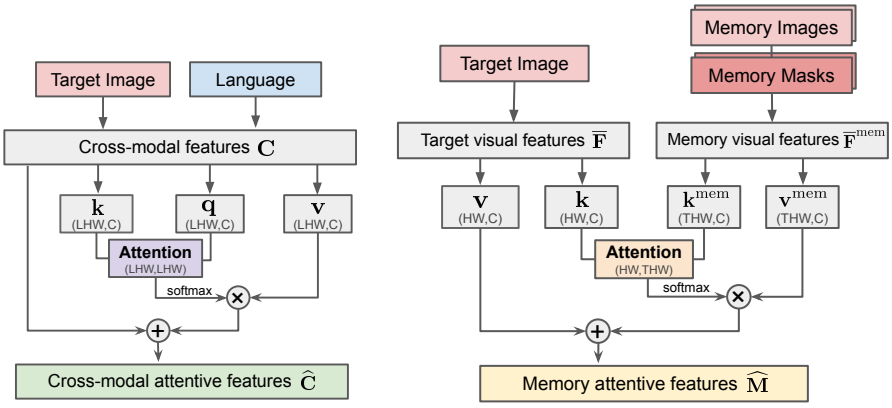
$$\mathbf{v} = \mathrm{Conv}_{\mathrm{value}}(\mathbf{C}) \in \mathbb{R}^{L \times H \times W \times C_\mathrm{a}} \tag{6}$$

and we compute cross-modal attentive features by estimating the correlation between all combinations of pixels and words as

$$\widehat{\mathbf{c}}_{pl} = \mathbf{c}_{pl} + \sum_{\forall p', \forall l'} \mathrm{Softmax}(\mathbf{q}_{pl} \cdot \mathbf{k}_{p'l'}) \mathbf{v}_{p'l'}, \tag{7}$$

where $\cdot$ denotes the dot-product operator. We average the self-attentive features over words and derive the final cross-modal feature as $\widehat{\mathbf{c}}_p = \frac{1}{L} \sum_l \mathbf{c}_{pl}$ and $\widehat{\mathbf{C}} = \{\widehat{\mathbf{c}}_p \mid \forall p\} \in \mathbb{R}^{H \times W \times C_b}$.

Memory Attention Module To leverage information in the mask predictions at the frames processed earlier, we extend the idea introduced in [22] and design



(a) Cross-modal attention module (b) Memory attention module

Fig. 3: Detailed illustrations of cross-modal and memory attention modules. Each module retrieves relevant information from language and memory frames for the target image to obtain self-attentive features.

a memory attention module. This module retrieves the relevant information from the previous frame by computing the correlation between the visual feature map of the current frame and the mask-encoded visual feature map of the previous frame. Note that the mask-encoded visual features is obtained from another feature extractor that takes 4-channel inputs given by stacking an RGB image and its segmentation mask in the channel direction. We will call the current and previous frames as target and memory frames, respectively, hereafter.

Different from the previous method [22], we introduce a 12-dimensional spatio-temporal coordinate feature, $\widetilde{\mathbf{s}}_{tp}$[5], where the first 3 dimensions encode normalized temporal positions, the next 6 dimensions represent normalized vertical and horizontal positions, and the last 3 dimensions contain the information about duration and frame size of the whole video.

Let T be the number of memory frames. For a target frame and T memory frames, we first compute key $(\mathbf{k}, \mathbf{k}^{\text{mem}})$ and value $(\mathbf{v}, \mathbf{v}^{\text{mem}})$ embeddings as follows:

$$\overline{\mathbf{F}} = \{[\mathbf{f}_p; \mathbf{s}_p] | \forall p\} \in \mathbb{R}^{H \times W \times (C_{\mathbf{f}}+8)} \tag{8}$$

$$\mathbf{k} = \text{Conv}_{\text{key}}(\overline{\mathbf{F}}) \in \mathbb{R}^{H \times W \times C_{\text{b}}} \tag{9}$$

$$\mathbf{v} = \text{Conv}_{\text{value}}(\overline{\mathbf{F}}) \in \mathbb{R}^{H \times W \times C_{\text{b}}} \tag{10}$$

$$\overline{\mathbf{F}}^{\text{mem}} = \{[\mathbf{f}^{\text{mem}}_{tp}; \widetilde{\mathbf{s}}_{tp}] | \forall t, \forall p\} \in \mathbb{R}^{T \times H \times W \times (C_{\mathbf{f}}+12)} \tag{11}$$

$$\mathbf{k}^{\text{mem}} = \text{Conv}^{\text{mem}}_{\text{key}}(\overline{\mathbf{F}}^{\text{mem}}) \in \mathbb{R}^{T \times H \times W \times C_{\text{b}}} \tag{12}$$

$$\mathbf{v}^{\text{mem}} = \text{Conv}^{\text{mem}}_{\text{value}}(\overline{\mathbf{F}}^{\text{mem}}) \in \mathbb{R}^{T \times H \times W \times C_{\text{b}}} \tag{13}$$

[5] $\widetilde{\mathbf{s}}_{tp} = [t_{\min}, t_{\text{avg}}, t_{\max}, h_{\min}, h_{\text{avg}}, h_{\max}, w_{\min}, w_{\text{avg}}, w_{\max}, \frac{1}{T}, \frac{1}{H}, \frac{1}{W}]$.

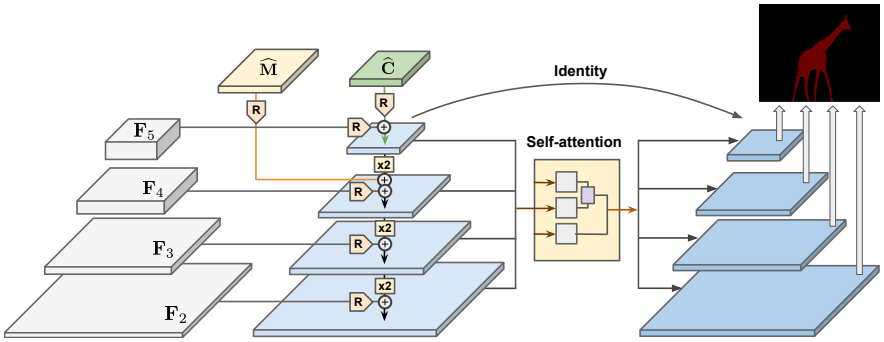


Fig. 4: Detailed illustration of our decoder. It first combines the cross-modal attentive feature map $\widehat{\mathbf{C}}$, the memory attentive feature map $\widehat{\mathbf{M}}$, and the original visual feature map $\mathbf{F}_l$ in multiple levels $l \in \{2, 3, 4, 5\}$ in a progressive manner. 'R' denotes ResBlocks and '×2' denotes upsampling layers in this figure. The multi-scale outputs are strengthened through a self-attention module, and then employed to estimate the final segmentation masks.

where $\mathbf{f}$ and $\mathbf{f}^{\text{mem}}$ denotes target and memory visual features, and $\mathbf{s}$ and $\widetilde{\mathbf{s}}$ denotes spatial and spatio-temporal coordinate features, respectively. Then, the memory-attentive feature $\widehat{\mathbf{m}}_p$ at the spatial location p is given by

$$\widehat{\mathbf{m}}_p = \mathbf{v}_p + \sum_{\forall t', \forall p'} f(\mathbf{k}_p, \mathbf{k}^{\text{mem}}_{t'p'}) \mathbf{v}^{\text{mem}}_{t'p'} \tag{14}$$

and $\widehat{\mathbf{M}} = \{\widehat{\mathbf{m}}_p \mid \forall p\} \in \mathbb{R}^{H \times W \times C_b}$. Fig. 3(b) presents the detailed illustration of the memory attention module, which shows how it computes the relevance between target frame and memory frames using key-value structure. Since it attends the regions in the target frame that are relevant to previous predictions, our algorithm produces temporally coherent segmentation results. Note that we employ the 4^{th} stage features (Res4) for both target and memory frames in this module because it requires more descriptive features to compute the correlation between local regions of the frames, while cross-modal attention module employs the 5^{th} stage features (Res5) to exploit more semantic information.

Decoder with Feature Pyramid Network We employ a coarse-to-fine hierarchical structure in our decoder to combine three kinds of semantic features; the cross-modal attentive feature map $\widehat{\mathbf{C}}$, the memory attentive feature map $\widehat{\mathbf{M}}$, and the original visual feature map $\mathbf{F}_l$ in different levels $l \in \{2, 3, 4, 5\}$. Fig. 4 illustrates how our decoder combines those three features using a feature pyramid network in a progressive manner. Each layer in the feature pyramid network takes the output of the previous layer and the ResBlock-encoded visual feature in the same level $\mathbf{F}_l$. Additionally, its first and the second layers incorporate cross-attentive features $\widehat{\mathbf{C}}$ and memory-attentive features $\widehat{\mathbf{M}}$, respectively, to capture

multi-modal and temporal information effectively. Note that each layer in the feature pyramid network is upsampled by the factor of 2 to match the feature map size to that of the subsequent level.

Instead of using the outputs from individual layers in the feature pyramid for mask generation, we employ an additional self-attention module following BFPN [23] to strengthen feature semantics of all levels. To this end, we first average the output features in all levels after normalizing their sizes and apply a self-attention to the combined feature map. The resulting map is rescaled to the original sizes, and the rescaled maps are aggregated to the original output feature maps forwarded through identity connections. Finally, these multi-scale outputs are employed to estimate segmentation masks in 1/4 scale of the input image, following the same pipeline in [11].

Inference Our network takes three kinds of inputs: a target image, memory images and their mask predictions, and a language expression. Since there is no predicted mask at the first frame, we introduce a novel two-stage procedure for its inference to fully exploit our two attention modules.

In the first stage, we run our network with no memory frame, which results in independent mask prediction at each frame based only on the language expression. After the initial per-frame mask estimation, we select an anchor frame, which has the most confident mask prediction for the language expression. To this end, we calculate the confidence score of each frame by averaging the pixelwsie final segmentation scores and select the frame with the highest one.

In the second stage, we update our initial segmentation results starting from the anchor to both ends using our full network. We first set the anchor frame as a memory frame, and re-estimate the object mask by sequentially propagating the mask prediction from anchor frame. After updating mask prediction at each frame, we add the image and its mask to the memory. In practice, however, cumulating all previous frames to the memory may cause memory overflow issues and slow down inference speed. To alleviate this problem, we set the maximum number of memory frames to T. If the number of memory frames reaches T, then we replace the least confident frame in the memory with the new prediction. Note that we leverage the previous mask predictions in the memory frames and estimate the mask of the target frame. At the same time, we use a language expression for guidance during the second stage as well, which allows us to handle challenging scenarios like drift and occlusions. We iteratively refine segmentation by repeating the second stage based on the new anchor identified at each iteration.

5 Experiments

We first evaluate the proposed method on our Refer-Youtube-VOS dataset, and perform comparison to the existing work on the Refer-DAVIS$_{17}$ dataset [10]. We also provide diverse ablation studies to validate the effectiveness of our dataset and framework.

Table 2: The quantitative evaluation of referring video object segmentation on the Refer-Youtube-VOS validation set.

Method	prec@0.5	prec@0.6	prec@0.7	prec@0.8	prec@0.9	$\mathcal{J}$	$\mathcal{F}$
Baseline (Image-based)	31.98	27.66	21.54	14.56	4.33	33.34	36.54
Baseline + RNN	40.24	35.90	30.34	22.26	9.35	34.79	38.08
Ours w/o cross-modal attention	41.58	36.12	28.50	20.13	8.37	38.88	42.82
Ours w/o memory attention	46.26	40.98	34.81	25.42	10.86	39.38	41.78
Ours	**52.19**	**46.77**	**40.16**	**27.68**	**14.11**	**45.27**	**49.19**

Table 3: The quantitative evaluation of referring video object segmentation on Refer-DAVIS$_{17}$ validation set.

Method	Pretrained	$\mathcal{J}$	$\mathcal{F}$
Khoreva *et al.* [10]	RefCOCO [20]	37.3	41.3
Ours	RefCOCO [20]	41.23	47.01
Baseline (frame-based)	Refer-YV (ours)	32.19	37.23
Basline + RNN	Refer-YV (ours)	36.94	43.45
Ours w/o cross-modal attention	Refer-YV (ours)	38.25	43.20
Ours w/o memory attention	Refer-YV (ours)	39.43	45.87
Ours (pretraining only)	Refer-YV (ours)	44.29	49.41
Ours	Refer-YV (ours)	**47.29**	**55.96**

5.1 Implementation Details

We employ a pretrained ResNet-50 on the ImageNet dataset as our backbone network. Every frame of an input video is resized to 320×320. The maximum length of an expression, L, is 20 and the dimensionality of the word embedding space, $C_{\mathbf{e}}$, is 1,000. We train our model using the Adam optimizer with a batch size 16. Our model is trained end-to-end for 120 epochs. The learning rate is initialized to 2×10^{-5} and decayed by the factor of 10 at every 80 epochs. We set the maximum number of memory frames, T, to 4.

5.2 Evaluation Metrics

We use two standard evaluation metrics, the region similarity ($\mathcal{J}$) and the contour accuracy ($\mathcal{F}$) following [26]. Additionally, we also measure prec@X, the percentage of correctly segmented frames in the whole dataset, given a predefined threshold X sampled from the range $[0.5, 0.9]$. Note that segmentation in a frame is regarded as successful if its $\mathcal{J}$ score is higher than a threshold.

5.3 Quantitative Results

Refer-Youtube-VOS We present the experimental results of our framework on the Refer-Youtube-VOS dataset in Table 2. We follow the original Youtube-

(a) *A person wearing a white shirt with white helmet riding a bike.*

(b) *A laying cat gets up and jumps towards the camera.*

(c) *A tiger to the right of another tiger.*

(d) *A man with an instrument standing in between two other people.*

Fig. 5: Qualitative results of our models on Refer-Youtube-VOS dataset.

VOS dataset [34] to split the data into training and validation sets. In Table 2, 'Baseline' denotes a variant of the frame-based model [36] with our feature pyramid decoder while 'Baseline + RNN' is an extension of the baseline model, which applies a GRU layer to the visual features extracted from multiple input frames for sequential estimation of masks. 'Ours w/o cross-modal attention' and 'Ours w/o memory attention' are the ablative models without the cross-modal attention module and the memory attention module, respectively, for both training and inference. As shown in Table 2, our full model achieves remarkable performance gain over all the compared models on the Refer-Youtube-VOS dataset. The huge performance boost in our full model with respect to the ablative ones implies crucial role of the integrated attention modules in this referring video object segmentation task.

Refer-DAVIS$_{17}$ DAVIS 2017 [25] is the most popular benchmark dataset for the video object segmentation task, which consists of 197 objects in 89 videos. Each video is composed of high-resolution frames with segmentation annotations, and involves various challenges including occlusions, multi-object interactions, camera motion, etc. Refer-DAVIS$_{17}$ [10] is the extension of DAVIS 2017 with natural language expressions. We evaluated all the models tested on the Refer-Youtube-VOS dataset. Because the number of videos in the DAVIS dataset is not sufficient to train the models for our task from scratch, we pretrain the models on Refer-Youtube-VOS and then fine-tune them on Refer-DAVIS$_{17}$. Table 3 shows the experimental results, where our model outperforms the existing

Table 4: The effects of dataset scale on our algorithm. We evaluate on the same validation set for each scale.

Dataset Scale	10%	20%	30%	50%	100%
$\mathcal{J}$	30.73	37.77	39.19	42.48	**45.27**
$\mathcal{F}$	32.92	41.02	43.15	46.05	**49.19**

method [10] and the ablative models. For the fair comparison with [10], we pretrained our model on a large-scale referring image segmentation benchmark [20]; our method turns out to be better than [10] under the same pretraining environment. Also, note that our model pretrained on Refer-Youtube-VOS with no fine-tuning on Refer-DAVIS$_{17}$ outperforms all other baselines while our full model further boosts accuracy significantly. This demonstrates the effectiveness of the new large-scale dataset and the proposed network.

5.4 Qualitative Results

Fig. 5 illustrates the qualitative results of our method on the Refer-Youtube-VOS dataset. The proposed model segments the target objects successfully with sharp boundaries on many videos and queries. We observe that our framework handles occlusion, deformation, and target identification effectively. See our supplementary documents for more qualitative results.

5.5 Analysis

Dataset Scale To investigate how the accuracy of a model changes depending on dataset sizes, we conduct experiments on four different subsets of the Refer-Youtube-VOS dataset, which contains 10%, 20%, 30%, and 50% of training examples, respectively. Table 4 presents the impact of dataset scale on model performance. As expected, the accuracy gradually improves upon the increase in the dataset size, which demonstrates the importance of a large-scale dataset on the referring video object segmentation task.

Inference procedure To validate the effectiveness of our inference scheme, we compare it with two other options for mask prediction. The baseline method, denoted by 'Forward', computes the mask at the first frame and propagates it in the forward direction until the end of video. We have also tested a variant ('Anchor + Previous') of the proposed two-stage inference method. 'Anchor + Previous' first estimates the masks in each frame independently and propagate an anchor frame in a sequential manner, where the previous T frames are used as memory frames during the second stage. Table 5 presents that our full inference technique gives the best performance, which implies that both use of anchor frames and memory frame selection by confidence contribute to improving segmentation results.

Table 5: Ablation study on the effects of inference procedures.

Inference scheme	$\mathcal{J}$	$\mathcal{F}$
Forward	43.13	49.07
Anchor + Previous	44.58	49.14
Ours	**45.27**	**49.19**

Table 6: Iteration of inference procedures in terms of region similarity ($\mathcal{J}$).

	Stage 1	Stage 2					
		Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 10
$\mathcal{J}$	41.34	45.27	45.33	45.41	45.44	45.43	**45.46**

Iterative inference We study the benefit given by the multiple iterations of the second stage inference step. Table 6 illustrates that the iterative inference procedure gradually improves accuracy and tends to be saturated after 5 iterations.

6 Conclusion

We have proposed a unified referring video object segmentation network to exploit both language-based object segmentation and mask propagation in a single model. Our two attention modules, cross-modal attention and memory attention, collaborate to obtain accurate target object masks specified by language expressions and achieve temporally coherent segmentation results across frames. We also constructed the first large-scale referring video object segmentation dataset. Our framework accomplishes remarkable performance gain on our new dataset as well as the existing one. We believe the new dataset and our proposed method will foster the new direction in this line of research.

Acknowledgement

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) [2017-0-01779, 2017-0-01780].

References

1. Benard, A., Gygli, M.: Interactive video object segmentation in the wild. arXiv preprint arXiv:1801.00269 (2017) 1

2. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017) 1, 3
3. Caelles, S., Montes, A., Maninis, K.K., Chen, Y., Van Gool, L., Perazzi, F., Pont-Tuset, J.: The 2018 davis challenge on video object segmentation. arXiv preprint arXiv:1803.00557 (2018) 1
4. Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR (2018) 1
5. Fan, C., Zhang, X., Zhang, S., Wang, W., Zhang, C., Huang, H.: Heterogeneous memory enhanced multimodal attention model for video question answering. In: CVPR (2019) 4
6. Gavrilyuk, K., Ghodrati, A., Li, Z., Snoek, C.G.: Actor and action video segmentation from a sentence. In: CVPR (2018) 4
7. Goel, V., Weng, J., Poupart, P.: Unsupervised video object segmentation for deep reinforcement learning. In: NIPS (2018) 1
8. Hu, R., Rohrbach, M., Darrell, T.: Segmentation from natural language expressions. In: ECCV (2016) 3, 6
9. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: CVPR (2013) 4
10. Khoreva, A., Rohrbach, A., Schiele, B.: Video object segmentation with language referring expressions. In: ACCV (2018) 2, 4, 10, 11, 12, 13
11. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR (2019) 10
12. Li, R., Li, K., Kuo, Y.C., Shu, M., Qi, X., Shen, X., Jia, J.: Referring image segmentation via recurrent refinement networks. In: CVPR (2018) 2, 3, 5, 6
13. Li, S., Seybold, B., Vorobyov, A., Lei, X., Jay Kuo, C.C.: Unsupervised video object segmentation with motion-based bilateral networks. In: ECCV (2018) 3
14. Li, X., Change Loy, C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: ECCV (2018) 3
15. Li, Z., Tao, R., Gavves, E., Snoek, C.G., Smeulders, A.W.: Tracking by natural language specification. In: CVPR (2017) 4
16. Liu, C., Lin, Z., Shen, X., Yang, J., Lu, X., Yuille, A.: Recurrent multimodal interaction for referring image segmentation. In: ICCV (2017) 2
17. Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L.: Deep extreme cut: From extreme points to object segmentation. In: CVPR (2018)
18. Margffoy-Tuay, E., Pérez, J.C., Botero, E., Arbeláez, P.: Dynamic multimodal instance segmentation guided by natural language queries. In: ECCV (2018) 5, 6
19. Mun, J., Yang, L., Ren, Z., Xu, N., Han, B.: Streamlined dense video captioning. In: CVPR (2019) 4
20. Nagaraja, V.K., Morariu, V.I., Davis, L.S.: Modeling context between objects for referring expression understanding. In: ECCV (2016) 11, 13
21. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Fast user-guided video object segmentation by interaction-and-propagation networks. In: CVPR (2019) 1
22. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019) 1, 3, 6, 7, 8
23. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: CVPR (2019) 10
24. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR (2017) 1, 3
25. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016) 3, 4, 12

26. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016) 11
27. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: ICCV (2017) 3
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017) 7
29. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: CVPR (2019) 1, 3
30. Wang, H., Deng, C., Yan, J., Tao, D.: Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In: ICCV (2019) 4
31. Wang, W., Song, H., Zhao, S., Shen, J., Zhao, S., Hoi, S.C., Ling, H.: Learning unsupervised video object segmentation through visual attention. In: CVPR (2019) 1
32. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018) 1, 3, 6
33. Xu, C., Hsieh, S.H., Xiong, C., Corso, J.J.: Can humans fly? action understanding with multiple classes of actors. In: CVPR (2015) 4
34. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: ECCV (2018) 3, 4, 12
35. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: CVPR (2018) 1, 3
36. Ye, L., Rochan, M., Liu, Z., Wang, Y.: Cross-modal self-attention network for referring image segmentation. In: CVPR (2019) 2, 3, 5, 6, 7, 12
37. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: CVPR (2018) 2, 3, 5, 6
38. Zhou, T., Wang, S., Zhou, Y., Yao, Y., Li, J., Shao, L.: Motion-attentive transition for zero-shot video object segmentation. In: AAAI (2020) 3