Multi-Level Representation Learning with Semantic Alignment for Referring Video Object Segmentation

Dongming Wu\textsuperscript{1,2*}, Xingping Dong\textsuperscript{2*}, Ling Shao\textsuperscript{3}, Jianbing Shen\textsuperscript{4†}

\textsuperscript{1} Beijing Institute of Technology, \textsuperscript{2} Inception Institute of Artificial Intelligence, \textsuperscript{3} Terminus Group, China, \textsuperscript{4} SKL-IOTSC, University of Macau

wudongming@bit.edu.cn, \{xingping.dong, shenjiangbingcg\}@gmail.com, ling.shao@ieee.org

Abstract

Referring video object segmentation (RVOS) is a challenging language-guided video grounding task, which requires comprehensively understanding the semantic information of both video content and language queries for object prediction. However, existing methods adopt multi-modal fusion at a frame-based spatial granularity. The limitation of visual representation is prone to causing vision-language mismatching and producing poor segmentation results. To address this, we propose a novel multi-level representation learning approach, which explores the inherent structure of the video content to provide a set of discriminative visual embedding, enabling more effective vision-language semantic alignment. Specifically, we embed different visual cues in terms of visual granularity, including multi-frame long-temporal information at video level, intra-frame spatial semantics at frame level, and enhanced object-aware feature prior at object level. With the powerful multi-level visual embedding and carefully-designed dynamic alignment, our model can generate a robust representation for accurate video object segmentation. Extensive experiments on Refer-DAVIS\textsubscript{17} and Refer-YouTube-VOS demonstrate that our model achieves superior performance both in segmentation accuracy and inference speed.

1. Introduction

Given a natural language expression, referring video object segmentation (RVOS) aims to predict the most relevant visual target from a video. It has wide applications, including video editing, virtual reality and human-robot interaction [49]. Different from the regular unsupervised or semi-supervised video object segmentation (VOS) [12, 21, 33, 53, 54], which localizes objects with salience or annotations of key frames, RVOS requires cross-modal understanding between the language query and video content.

As a human recognizes a referent object with the guidance of language, it is natural to rely on three steps: 1) observe its appearance (i.e., frame-based), 2) check its movement based on multiple frames (i.e., video-based), 3) shift more attention to the occluded or small objects (i.e., object-based). Most current approaches [1, 25, 43] simply leverage successful referring image comprehension methods to the cross-model understanding. They either use referring image grounding [24, 31, 58, 60] to generate target object bounding boxes as proposals, or utilize referring image segmentation directly [6, 10, 18, 22, 27, 56]. However, these solutions build on the simple frame-level visual representation to perform frame-sentence interaction. These frame-level modeling methods suffer from two limitations compared to the human recognition system: ignoring long-temporal information and lacking attention to salient spatial objects.

The limitation of visual representations causes the mis-
alignment between two modalities, further producing inaccurate segmentation results. For example, as illustrated in Fig. 1, given an input video and its corresponding description, “a lion is walking towards right side”, RVOS aims to segment the moving lion from the video. However, as there are multiple lions in the video, the frame-level modeling cannot recognize the correct one by employing only spatial appearance information as shown in Fig. 1(b). Since the referent object has a temporally moving status, it requires incorporating long-temporal information from multiple frames to identify the action. In addition, another expression, “a lion lying on a high rock” refers to an occluded and small-size lion. However, the frame-level modeling focuses only on the global semantics concerning each frame, and ignores these important and representative visual regions. It will lead to the referent object being missing, as shown in Fig. 1(c). To ease this difficulty, it is also necessary to capture the salient spatial objects from each frame as candidates to facilitate cross-modal understanding.

In this paper, we propose a novel multi-level learning framework for addressing RVOS. The model first presents a fine-grained analysis of video content for multi-granularity visual embedding:

- At the video granularity, we propose to model long-temporal dependencies of the entire video using a cross-frame pixel-wise calculator, which makes the feature representations to capture the object movement and dynamic scenes information.
- At the frame granularity, we encourage the frame representation to describe global content within a whole image, by learning to aggregate intra-frame information following the self-attention mechanism.
- At the object granularity, we leverage object-aware information generated from an object detector to enhance the foreground and background discriminability, benefiting from addressing the cases of occlusion and small object.

Once we obtain the multi-level visual embedding, we propose Dynamic Semantic Alignment (DSA) to interact them with the linguistic features. In particular, to effectively capture the granularity-specific information, we first separately incorporate global linguistic semantics according to the different visual cues. The generated vision-conditioned linguistic features are combined with the corresponding visual embedding to provide a granularity-specific representation for the referent object. Finally, we integrate the multi-level target-aware features and boundary information to guide the mask prediction of all frames using a Boundary-Aware Segmentation (BAS).

Overall, our contributions are summarized as three-fold: First, we propose a new framework for RVOS based on multi-level representation learning. It precludes the limitation of single frame-level visual modeling by a more structural video representation, promoting accurate vision-language semantic alignment. Second, we introduce a Dynamic Semantic Alignment (DSA), which dynamically learns and matches linguistic semantics with the different-granularity visual representation more compactly and effectively. Third, our approach achieves compelling performance on two challenging benchmarks, including Refer-DAVIS17 [25] and Refer-YouTube-VOS [43]. Notably, we obtain a significant improvement of 6.6% than the best frame-grained method in terms of J’ on Refer-DAVIS17. Meanwhile, it achieves a high inference speed at 53.2 FPS.

2. Related Work

2.1. Referring Video Object Segmentation

The goal of referring video object segmentation (RVOS) is to localize the entities in a video that are matched with the description of a natural language expression. Khoreva et al. [25] introduce a two-stage method, the first stage to generate bounding boxes in image [58, 60] and the second one to segment the referent object from video [20, 40]. Recently, RefVOS [1] employs the fine-grained categorization of expressions to overcome the overfitting. However, their frame-sentence interaction mechanism lacks the long-temporal and fine-grained visual representations, further resulting in the cross-modal misalignment as discussed before. Although a large number of works on actor and action video segmentation [11, 19, 34, 45, 46, 57] also study the problem of language-queried video segmentation, their descriptions are limited into the format of ‘actors’ performing a salient ‘action’. The newly appearing RVOS shows improved difficulties in both visual and linguistic modalities. Thus, our method can be regarded as a more generalized work to handle real-life situations.

2.2. Multi-Level Representation Learning

Multi-level representation learning is a common concept in feature embedding, including natural language processing [9, 14, 32] and computer vision [2, 7, 8, 13, 17, 52, 61]. The language processing usually cooperates with the word-phrase-sentence composition semantics to enrich word embedding, while the visual tasks focus on exploiting spatial or temporal granularity to learn a robust and powerful visual feature representation [29, 30, 47, 48]. For the video understanding task, the most popular granularity analysis is built on the temporal order [16, 17, 28, 49]. For instance, Hu et al. [17] associate different sub-networks to leverage the inherent temporal continuity of previous frames for fast video semantic segmentation. Lu et al. [30] summarizes the frame-term, short-term, long-term and global features of
each video for robust unsupervised video object segmentation. However, these methods are limited in visual modeling and cannot deal with the crucial cross-modal understanding. Recently, several language-queried tasks [23, 36, 50, 59] address these drawbacks and achieve promising object or moment localization via global-local video-language alignment, but they are not suitable for fine-grained object segmentation. This paper proposes a new perspective of exploring multi-level video information and cross-modal semantic alignment for precise mask prediction.

3. Methodology

Given a video clip and a natural language query, the goal of our approach is to automatically generate a set of referent object masks. We illustrate the overall pipeline in Fig. 2. The multi-level visual representation first separately embeds the CNN-encoded features at video, frame, and object level, which provides three enhanced visual representations (§3.1). The specific visual representation and the linguistic embedding are then fed into our dynamic semantic alignment to jointly highlight the visual features of interest (§3.2). Finally, the boundary-aware segmentation integrates the target-aware features and boundary information to guide the referent object prediction (§3.3). In the following, we will introduce them carefully.

3.1. Multi-Level Visual Representation

Before learning multi-level visual representations, we first extract frame-wise video features for a given video clip. The $T \times 3 \times H \times W$ frame video $I \in \mathbb{R}^{T \times 3 \times H \times W}$ is fed into the ResNet-50 [15] to obtain the res5 features $F \in \mathbb{R}^{T \times c \times h \times w}$, where $c$, $h$, $w$ represent the channel, height, weight number of the 3D tensors, respectively. Furthermore, a $1 \times 1$ convolution is used to reduce the channel dimension from $c$ to a smaller $d$ ($d << c$), as well as keep the same dimension with linguistic features in §3.2. The transformed video features are then fed into our multi-level visual representation module to embed different types of visual cues.

The multi-level visual representation consists of three independent embedding modules, 1) a video-level embedding to describe the global and long-temporal statistics of the entire video, 2) a frame-level embedding to learn the intra-frame long-distance semantic context, 3) an object-level embedding to highlight the object-aware features.

**Video-level Embedding.** Inspired by the recent success of visual transformer [3, 51], we take advantage of the core long-distance modeling ability of self-attention to formulate our video-level embedding module. It handles all video frames in a unified manner and models the pixel-level pairwise relation directly. Specifically, we flatten the entire video features into a 2D pixel-wise sequence $P \in \mathbb{R}^{T \times hw \times d}$. Three different fully-connected layers $W_Q, W_K, W_V$ are used to transform the sequence as $Q^\text{video} = W_Q P \in \mathbb{R}^{T \times hw \times d}$, $K^\text{video} = W_K P \in \mathbb{R}^{T \times hw \times d}$, $V^\text{video} = W_V P \in \mathbb{R}^{T \times hw \times d}$. Then we calculate a similarity matrix $A^\text{video}$ with pairwise dot product and normalize it with softmax,

$$A^\text{video} = \text{Softmax}(\frac{Q^\text{video} K^\text{video}^\top}{\sqrt{d}}) \in \mathbb{R}^{T \times Tw \times Tw},$$

where $A^\text{video}$ measures the relevance between each pixel in the video. These video sequence vectors are weighted according to the relevance and added with the original $P$:

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$

$$P^\text{video} = A^\text{video} V^\text{video} + P \in \mathbb{R}^{T \times Tw \times d},$$
where $\hat{P}^{\text{video}}$ is the video-level feature embedding. It models multi-frame information and represents holistic understanding of the video.

**Frame-level Embedding.** To learn frame-level feature embedding, following self-attention mechanism, we build the spatial pixel-wise relationship for each frame. Unlike prior work [43] that processes each pixel, our method is used on each frame independently. It maps each frame feature into 2D tensor $P^t \in \mathbb{R}^{hw \times d}$ ($t = 1, ..., T$), applies linear transformation to generate $Q^t_{\text{frame}}, K^t_{\text{frame}}, V^t_{\text{frame}}$, and performs weighting operation using learnable attention:

$$
\hat{P}^t_{\text{frame}} = \text{Softmax}\left(\frac{Q^t_{\text{frame}} K^t_{\text{frame}}^T}{\sqrt{d}}\right) V^t_{\text{frame}} + P^t,
$$

where $\hat{P}^t_{\text{frame}} \in \mathbb{R}^{hw \times d}$ represents the feature embedding of the $t$th frame.

**Object-level Embedding.** In addition to learning the global semantics for video and image, we also conduct object-level feature embedding to capture salient spatial information. This can be viewed an object detection process, which includes two sequential parts, an object encoder for object-aware feature extraction, and a segmentation decoder for salient object generation.

Let $F_{\text{enc}}$ denote the object encoder, which accepts the original video features $F$ as input, and directly outputs the object-level embedding $\hat{P}^{\text{object}}$.

$$
\hat{P}^{\text{object}} = F_{\text{enc}}(F).
$$

After that, we implement the segmentation decoder $F_{\text{dec}}$ to generate all salient objects:

$$
Y^{\text{object}} = F_{\text{dec}}(\hat{P}^{\text{object}}) \in \mathbb{R}^{T \times 1 \times H \times W},
$$

where $Y^{\text{object}}$ are one-channel feature maps for all frames, which are activated using a sigmoid function, and supervised by the object-level ground-truth $Y^{\text{object}}$.

$$
L^{\text{object}} = L_{\text{mask}}(Y^{\text{object}}, \hat{Y}^{\text{object}}).
$$

Here, with the encouragement of object-level loss $L^{\text{object}}$, the object encoder can highlight the object-sensitive features to serve as the object-level embedding $\hat{P}^{\text{object}}$. The mask loss $L_{\text{mask}}$ is the summation of Dice loss $L_{\text{dice}}$ [35] and focal loss $L_{\text{focal}}$ [26], i.e., $L_{\text{mask}} = L_{\text{dice}} + L_{\text{focal}}$.

The object encoder $F_{\text{enc}}$ can use various feature embedding models, such as fully-convolutional network (FCN), video-level encoder and frame-level encoder as aforementioned. Empirically, we choose the video-level encoder with a $3 \times 3$ convolution to be our object encoder according to the experiments in §4.4. The segmentation decoder $F_{\text{dec}}$ is built on fully-convolutional network that is similar to the pyramid segmentation head in §3.3. To sum up, the joint multi-grained learning provides an enhanced and informative visual representation, which will facilitate the following vision-language semantic alignment.

![Figure 3. Two solutions of semantic alignment: (a) video-level alignment for global and long-temporal alignment, and (b) frame-level alignment for spatial alignment.](image)

### 3.2. Dynamic Semantic Alignment

Given the different-level visual embedding representations $P^{\text{video}}, \{P^t_{\text{frame}}\}_{t=1}^T, P^{\text{object}}$ as well as its corresponding language description, the goal of DSA is to enable two modalities interact with each other for characterizing the referent object representation. To dynamically learn the global linguistic semantics that have the most relationship with each visual granularity, we individually embed three linguistic representations $S^{\text{video}}, S^{\text{frame}}, S^{\text{object}}$. Taking the video branch as an example, for the input language query with $N$ words, we follow the work [56] to encode each word into a feature vector. A transformer encoder [44] is trained to extract the specific linguistic features, which are denoted as $S^{\text{video}} \in \mathbb{R}^{N \times d}$ where $d$ is the feature dimension. The same operation is also applied on frame and object branch to obtain $S^{\text{frame}}$ and $S^{\text{object}}$.

DSA includes two kinds of solutions in terms of the interaction level, i.e., video-level alignment and frame-level alignment, as depicted in Fig. 3. The video-level semantic alignment $F^{\text{video}}$ (used at video granularity) takes the temporal information across two modalities to be aligned, while the frame-level alignment $F^{\text{frame}}$ (used at frame and object granularity) is responsible for spatial alignment:

$$
M^{\text{video}} = F^{\text{video}}(\hat{P}^{\text{video}}, S^{\text{video}}),
$$

$$
M^t_{\text{frame}} = F^{\text{frame}}(\hat{P}^t_{\text{frame}}, S^{\text{frame}}),
$$

$$
M^t_{\text{object}} = F^{\text{frame}}(\hat{P}^t_{\text{object}}, S^{\text{object}}),
$$

where $M^{\text{video}}, M^t_{\text{frame}},$ and $M^t_{\text{object}}$ are cross-modal features. Both $F^{\text{video}}$ and $F^{\text{frame}}$ have a standard semantic alignment architecture as presented in the following.

**Semantic Alignment.** For the convenience of description, we omit the granularity superscript and frame subscript of visual features and reshape them as $P \in \mathbb{R}^{T \times hw \times d}$. We add the position embedding, as proposed in [44, 51], into the vi-
visual and linguistic features to keep the coordinate alignment and employ linear layers to transform them:
\[
\hat{P'} = \text{Linear}(\hat{P} + \text{POS}^V) \in \mathbb{R}^{(T)hw \times d},
\]
\[
S' = \text{Linear}(S + \text{POS}^L) \in \mathbb{R}^{N \times d},
\]
where POS$^V$ and POS$^L$ are the visual and linguistic position, respectively. The transformed visual and linguistic features are further calculated through matrix product and softmax normalization:
\[
A_{lang}^\text{lang} = \text{Softmax}(\frac{\hat{P}'S'T}{\sqrt{d}}) \in \mathbb{R}^{(T)hw \times N}.\]

Here attention map $A_{lang}^\text{lang}$ represents the similarity between each word and each location of the visual representation. Next, the granularity-specific linguistic features are summarized as $S = A_{lang}^\text{lang}S \in \mathbb{R}^{(T)hw \times d}$, and are added into original $\hat{P}$ to automatically align two features:
\[
M = \text{Linear}(A_{lang}^\text{lang}S + \hat{P}) \in \mathbb{R}^{(T)hw \times d},
\]
where $M$ represents the activated target-aware features after semantic alignment. We recover their level superscript and the format of video, i.e., $M_{\text{video}}, M_{\text{frame}}, M_{\text{object}} \in \mathbb{R}^{T \times hw \times d}$. Their size is reshaped as $T \times d \times h \times w$, and we concatenate them along channel dimension, i.e., $M = [M_{\text{video}}, M_{\text{frame}}, M_{\text{object}}] \in \mathbb{R}^{T \times d \times hw}$ for following mask estimation.

3.3. Boundary-Aware Segmentation

The BAS aims to produce pixel-wise masks using rich target-aware and boundary-aware information. It first generates a one-channel boundary map $B$ by accepting the modulated target-aware features $M$ and original visual features $F$ as input:
\[
B = F_{\text{bdry}}(M, F_{[2,3,4,5]}) \in \mathbb{R}^{T \times 1 \times H \times W},
\]
where $F_{[2,3,4,5]}$ is a simplified feature denotation from different backbone layers (Res2, Res3, Res4, Res5). The boundary head $F_{\text{bdry}}$ and segmentation head $F_{\text{seg}}$ have the same pyramid architecture by inserting the different-scale original features, as like the pyramid decoder of [43]. The outputs from two heads are concatenated together to estimate finer object masks $E$:
\[
E = F_{\text{cnf}}(B, F_{\text{seg}}(M, F_{[2,3,4,5]})) \in \mathbb{R}^{T \times 1 \times H \times W},
\]
where $F_{\text{cnf}}$ includes a superficial 3x3 convolutional layer. The adopted instance-level loss combines the mask and boundary supervision:
\[
\mathcal{L}_{\text{instance}} = \mathcal{L}_{\text{mask}}(B, \hat{Y}) + \alpha \mathcal{L}_{\text{bdry}}(E, \hat{Y}_{\text{bdry}}),
\]
where $\hat{Y}, \hat{Y}_{\text{bdry}}$ represent the ground-truth of $B$ and $E$. $\mathcal{L}_{\text{mask}}$ is the mask loss as mentioned in §3.1. $\mathcal{L}_{\text{bdry}}$ is the boundary loss following [42] and $\alpha$ is a hyper-parameter. The overall objective is the summation of object-level loss (Eq. 7) and instance-level loss (Eq. 14):
\[
\mathcal{L} = \mathcal{L}_{\text{object}} + \mathcal{L}_{\text{instance}}.
\]

3.4. Implementation Details

Network. The backbone model adopted in our approach is ResNet-50 [15], which is pretrained on ImageNet [5]. We only use the feature maps from the last layer for visual embedding and semantic alignment, while BAS accepts the feature pyramids of the backbone model for FPN-like coarse-to-fine segmentation. In BAS, the mapping block between two levels consists of a $3 \times 3$ convolution, a group normalization (8 groups) and a bilinear upsampling layer. The final one-channel feature maps of $B$ and $E$ are activated using sigmoid for training and inference.

Training. The input video has $T = 12$ frames with the size of $432 \times 240$. The language length is $N = 20$, and the feature dimension is set to $d = 384$. The object annotations (Eq. 7) can be obtained by combining all instance-level labels. We calculate the boundary annotations (Eq. 14) following the work of [42]. The hyper-parameter $\alpha$ is 0.2. Our model is implemented on PyTorch [39] and trained on four NVIDIA Tesla V100 GPUs with 32GB memory per card. We optimize the overall model with AdaW optimizer using learning rate $1e^{-4}$ for backbone, $1e^{-5}$ for the remaining part. The batch size is set to 2. Note that we predict confidence scores $C = F_{\text{conf}}(M) \in \mathbb{R}^{T \times 1}$ for all frames with an extra confidence estimation head $F_{\text{conf}}$, as shown in Fig. 2. Therefore, we build a new overall objective:
\[
\mathcal{L} = \mathcal{L}_{\text{object}} + \mathcal{L}_{\text{instance}} + \beta \mathcal{L}_{\text{conf}}(C, \text{IoU}(Y, \hat{Y})),
\]
where IoU indicates the IoU calculation operation, and $\mathcal{L}_{\text{conf}}$ is $L_2$ loss. $\beta = 0.1$ serves as a hyperparameter. $F_{\text{conf}}$ contains a global average pooling and three fully-convolution layers with the final layer outputting $T$ scores.

Inference. During inference, we also exploit the recent VOS method, STCN [4] to improve the cross-frame object consistency as well as refine the segmentation results as a post-processing strategy. STCN propagates the highest-confidence mask in a bi-directional way to obtain the final segmentation masks for evaluation. We regard the output feature maps whose sigmoid activation value is higher than 0.5 as binary results.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments on two popular VOS benchmarks, i.e., Refer-DAVIS$_{17}$ [25] and Refer-YouTube-VOS [43]. Refer-DAVIS$_{17}$ expands DAVIS$_{17}$ [41] by annotating the objects of video with more than 1,500 refer-
ring expressions. It includes 90 videos, which are further split into two subsets: training set (60 videos), val set (30 videos). Refer-YouTube-VOS is a large-scale dataset, which includes 3,975 videos from YouTube-VOS [54] and 27,899 corresponding language descriptions. Similar to Refer-DAVIS17, this dataset contains two subsets: training set and val set. Although both provide the full-video expression based on an entire video and the first-frame expression based on the first frame, we only use their full-video expression for training and validation.

**Evaluation Metrics.** Following the standard evaluation protocol [43], we adopt the region similarity \( J \) (%), contour accuracy \( F \) (%), and Precision@\( \lambda \) (%) as our evaluation metrics. The region similarity \( J \) calculates the mean IoU between the prediction and ground truth, while the contour accuracy \( F \) measures the similarity between the boundary of the prediction and the ground truth. Precision@\( \lambda \) (prec@\( \lambda \)) denotes the percentage of testing samples whose region similarity is higher than a predefined threshold \( \lambda \), where \( \lambda \) is sampled from the range \([0.5, 0.9]\).

### 4.2. Quantitative Results

We compare our approach with several previous models on the two aforementioned challenging benchmarks. Baseline is a frame-based method proposed in [43], which employs a cross-modal attention module [56] for vision-language understanding, and a feature pyramid decoder for mask prediction. Baseline+RNN [43] denotes a variant of baseline, which utilizes a GRU layer to visual features from multiple input frames for estimation of masks. URVOS [43] builds on frame-level interaction, which unifies a memory network to replay previous frames and masks for refining the mask prediction of the current frame. RefVOS [1] is a simple frame-based modeling method, which directly conducts element-wise multiplication between visual and linguistic features to obtain the cross-modal representation.

**Refer-DAVIS17 val set.** Before training on Refer-DAVIS17, we pre-train our model on the large-scale Refer-YouTube-VOS training set, and test its performance on the Refer-DAVIS17 val set. As reported in Table 1, our approach has a remarkable performance improvement compared to the most recent model URVOS under the same ‘pretraining only’ case (\( J \): +5.8\%, \( F \): +6.0\%). After fine-tuning the pretrained model on the Refer-DAVIS17 training set, our approach largely outperforms all the comparative methods across all metrics (\( J \): +6.6\%, \( F \): +6.1\%) compared with URVOS. Besides, we also provide the results of our model pre-trained on RefCOCO [37], a referring image segmentation benchmark, which achieves higher scores than these frame-based methods, like URVOS [43] and RefVOS [1].

**Refer-YouTube-VOS val set.** We further examine the performance of the proposed approach on the Refer-YouTube-VOS val. We directly test the model trained on the Refer-YouTube-VOS training set. As seen in Table 2, our model significantly outperforms all the state-of-the-art methods in all metrics. Compared to URVOS [43], we improve the region similarity \( J \) by +3.1\% and the contour accuracy \( F \) by +1.8\%. Our method obtains a much higher score on precision@\( \lambda \) (e.g., prec@0.8:+5.0\%, prec@0.9:+4.8\%). All the results indicate the superiority of our multi-level representation learning with semantic alignment.

<table>
<thead>
<tr>
<th>Method</th>
<th>Prec@0.5</th>
<th>Prec@0.6</th>
<th>Prec@0.7</th>
<th>Prec@0.8</th>
<th>Prec@0.9</th>
<th>( J )</th>
<th>( F )</th>
<th>( J &amp; F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (frame-based) [56]</td>
<td>31.98</td>
<td>27.66</td>
<td>21.54</td>
<td>14.56</td>
<td>4.33</td>
<td>33.34</td>
<td>36.54</td>
<td>34.94</td>
</tr>
<tr>
<td>Baseline + RNN [56]</td>
<td>40.24</td>
<td>35.90</td>
<td>30.34</td>
<td>22.26</td>
<td>9.35</td>
<td>34.79</td>
<td>38.08</td>
<td>36.44</td>
</tr>
<tr>
<td>URVOS [43]</td>
<td>51.19</td>
<td>46.77</td>
<td>40.16</td>
<td>27.68</td>
<td>14.11</td>
<td>45.27</td>
<td>49.19</td>
<td>47.23</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>54.18</strong></td>
<td><strong>48.99</strong></td>
<td><strong>42.20</strong></td>
<td><strong>33.62</strong></td>
<td><strong>18.94</strong></td>
<td><strong>48.43</strong></td>
<td><strong>50.96</strong></td>
<td><strong>49.70</strong></td>
</tr>
</tbody>
</table>

Table 1. The quantitative evaluation on Refer-DAVIS17 val set, with region similarity \( J \), boundary accuracy \( F \), and average of \( J \& F \). Success percentage (prec@\( X \)) is also reported.

<table>
<thead>
<tr>
<th>Method</th>
<th>Prec@0.5</th>
<th>Prec@0.6</th>
<th>Prec@0.7</th>
<th>Prec@0.8</th>
<th>Prec@0.9</th>
<th>( J )</th>
<th>( F )</th>
<th>( J &amp; F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (frame-based) [56]</td>
<td>31.98</td>
<td>27.66</td>
<td>21.54</td>
<td>14.56</td>
<td>4.33</td>
<td>33.34</td>
<td>36.54</td>
<td>34.94</td>
</tr>
<tr>
<td>Baseline + RNN [56]</td>
<td>40.24</td>
<td>35.90</td>
<td>30.34</td>
<td>22.26</td>
<td>9.35</td>
<td>34.79</td>
<td>38.08</td>
<td>36.44</td>
</tr>
<tr>
<td>URVOS [43]</td>
<td>51.19</td>
<td>46.77</td>
<td>40.16</td>
<td>27.68</td>
<td>14.11</td>
<td>45.27</td>
<td>49.19</td>
<td>47.23</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>54.18</strong></td>
<td><strong>48.99</strong></td>
<td><strong>42.20</strong></td>
<td><strong>33.62</strong></td>
<td><strong>18.94</strong></td>
<td><strong>48.43</strong></td>
<td><strong>50.96</strong></td>
<td><strong>49.70</strong></td>
</tr>
</tbody>
</table>

Table 2. The quantitative evaluation on Refer-YouTube-VOS val set, with region similarity \( J \), boundary accuracy \( F \), average of \( J \& F \).
Figure 4. Qualitative results on Refer-DAVIS $\text{val}$ and Refer-YouTube-VOS $\text{val}$ set. The first four sequences represent the referring video object segmentation results. The last two sequences are object-level results with respect to the salient object prediction (Eq. 6).

4.3. Qualitative Results

Fig. 4 shows some typical visual results of our approach. In the first sequence (i.e., lab-coat), camera movement brings size deformation for the girl. In the second sequence (i.e., soapbox), the blue wooden car moves forward, which has difficulty in boundary estimation due to the considerable appearance variation. The third and fourth sequences come from the same video (i.e., 6031809500) but are more challenging due to local occlusion and visually similar objects in the background. Otherwise, our model succeeds in segmenting all the referent objects. Overall, benefiting from taking the multi-level embedding into account during the vision-language understanding, our model yields remarkable referring video object segmentation results.

In addition to the referring video object segmentation results, we also provide some object prediction results from object-level embedding in Fig. 4. It is well seen that all the objects are predicted with sharp boundaries, including the occluded and small ones, indicating that the object-aware feature maps can guide the generation of the salient object and provide object prior.

4.4. Ablation Studies

To analyze the effect of each component in our model, we conduct ablative studies on two benchmarks. Table 3 and Table 4 tabulate the results.

Multi-Level Analysis. To investigate our multi-level representation, we separately analyze video, frame, and object embedding in Table 3. As seen, by dropping the video embedding, the model encounters a performance drop ($J$: -2.2%, $F$: -1.9%). A similar trend is observed after discarding another two modules, thereby demonstrating the effectiveness of the multi-level representations. Moreover, we test two different variants of the object encoder, i.e., FCN or frame-level encoder. But both two have lower scores than video-level encoder (i.e., the full model).

Fig. 5 shows the ablative qualitative results by adding frame, video, object embedding one by one. The simple frame-level modeling cannot identify the moving and occluded objects accurately. Using video-level and object-level embedding can promote performance by learning the long-temporal information and shifting more attention.

Importance of Semantic Alignment. DSA is a key module in our method to achieve cross-modal understanding.
A qualitative result: robe grabbing a deer with identification number 13.

**Ablation studies on Refer-YouTube-VOS**

- **Method**: Propagation
- **Variants**: J, F, J&F
- **FPS**:
  - URVOS
    - STM [38] ICCV19: 39.43, 45.87, 42.65
    - Our: 47.29, 55.96, 51.45
  - Ours
    - STM [38] ICCV19: 49.96, 56.53, 53.25
    - CFBi+ [55] PAMIR1: 51.02, 58.65, 54.84

**Table 3. Ablation studies on Refer-YouTube-VOS val set, with region similarity J, boundary accuracy F.**

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Variants</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Model</td>
<td></td>
<td>48.43</td>
<td>50.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-Level Visual Representation</td>
<td>w/o video level</td>
<td>46.25</td>
<td>49.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>w/o frame level</td>
<td>47.09</td>
<td>49.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>w/o object level</td>
<td>46.95</td>
<td>49.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Encoder</td>
<td>FCN</td>
<td>47.24</td>
<td>49.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>frame encoder</td>
<td>47.56</td>
<td>50.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Alignment</td>
<td>w/o alignment</td>
<td>36.23</td>
<td>40.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numbers of Frames</td>
<td>1</td>
<td>46.12</td>
<td>48.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>46.80</td>
<td>49.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>47.47</td>
<td>49.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>48.11</td>
<td>50.41</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From Table 3, we can see that removing semantic alignment from our full model brings a considerable performance drop across all metrics (J: -12.2%, F: -10.6%). Fig. 5 clearly shows that semantic alignment plays an important role in identifying different objects.

**Number of Frames.** We also study the influence of different numbers of video frames on the final performance in Table 3. Better performance can be obtained with more input frames (e.g., 1 → 8). This observation indicates that the long-temporal modeling can mine cross-frame relationships to facilitate referring video segmentation. Due to the computation and memory limitation, we set the maximum number to 12 in our full model.

**Mask Propagation Method.** Next, we experiment several state-of-the-art mask propagation methods in Table 4, such as STM [38], CFBi+ [55], STCN [4], where STCN brings more refinement improvements. Further, we can observe that the performance gain is 3.9% and 3.5% in J and F, respectively. It is worth noticing that our model without mask propagation still achieves better performance in comparison to the state-of-art URVOS (J: +2.7%, F: +0.6%).

**Inference Speed.** Finally, we calculate the inference speed on a NVIDIA Tesla V100 GPU using the entire Refer-

**5. Conclusion**

In this paper, we proposed a novel multi-level representation learning framework to address RVOS task. We started with the observation that most RVOS methods rely heavily on frame-level modeling and omit the structural information of video content, leading to poor vision-language matching. Based on this motivation, we proposed to embed video-, frame-, and object-level semantics to provide a robust and informative visual representation. Afterward, to distinguish the referent object, we introduced dynamic semantic alignment for adaptively fusing two modalities. The boundary-aware segmentation integrated the generated target-aware feature and boundary information to predict the final results. Experiments show that our method outperforms previous methods by large margins on both Refer-DAVIS17 and Refer-YouTube-VOS.

**Table 4. Ablation studies about mask propagation on Refer-DAVIS17 val set, with region similarity J, boundary accuracy F, average of J&F. Inference speed (FPS) is also reported.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Propagation</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>URVOS</td>
<td>–</td>
<td>39.43</td>
<td>45.87</td>
<td>42.65</td>
<td>–</td>
</tr>
<tr>
<td>STM [38] ICCV19</td>
<td>47.29</td>
<td>55.96</td>
<td>51.45</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>–</td>
<td>49.96</td>
<td>56.53</td>
<td>53.25</td>
<td>53.2</td>
</tr>
<tr>
<td>CFBi+ [55] PAMIR1</td>
<td>51.02</td>
<td>58.65</td>
<td>54.84</td>
<td>5.59</td>
<td></td>
</tr>
<tr>
<td>STCN [4] SodFPS21</td>
<td>53.85</td>
<td>62.02</td>
<td>57.94</td>
<td>17.2</td>
<td></td>
</tr>
</tbody>
</table>
References


---

5004
[37] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 6
[42] Tao Ruan, Ting Liu, Zilong Huang, Yunchao Wei, Shikui Wei, and Yao Zhao. Devil in the details: Towards accurate single and multiple human parsing. In *AAAI*, 2019. 5
[54] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 1, 2, 6