Segment Every Reference Object in Spatial and Temporal Spaces
(Supplementary Material)

Jiannan Wu, Yi Jiang, Bin Yan, Huchuan Lu, Zehuan Yuan, Ping Luo

1The University of Hong Kong  2ByteDance
3Dalian University of Technology  4Shanghai AI Laboratory

Appendix A. Architecture

Reference Encoding. Figure 1 illustrates the process of reference encoding. (i) For mask references, we employ the same visual encoder \( \text{Enc}_V \) for both the current and reference frames to generate multi-scale features (i.e., C3, C4, C5). We denote the encoded features of the reference frame as \( F^V_f \), where the \( \ell \)-th feature (\( \ell = 2, 3, 4 \)) has a size of \( H_\ell \times W_\ell \times C \), with a spatial stride of \( 2^{\ell+1} \) relative to the original size. Next, we use a lightweight mask encoder (ResNet-18 in all our experiments) that takes the reference frame and annotated mask as inputs. We concatenate the last three layer features with the corresponding level features in \( F^V_f \) and further process them with two ResBlocks [2] and a CBAM block [10] to obtain the final outputs, denoted as \( F^m_V \). Finally, we flatten each level feature in \( F^V_f \) and \( F^m_V \) into 1-dimensional vectors. (ii) For language references, we directly use off-the-shelf text encoder RoBERTa [6] to extract the 1-d linguistic features.

![Visual Encoder](image1)
![Mask Encoder](image2)
![Text Encoder](image3)

Figure 1: The process of reference encoding for (a) mask references and (b) language references.

Appendix B. Implementation Details

Training Details. Our training process consists of three sequential stages: VG pretraining, image-level training and video-level training. We train models on NVIDIA A100 GPUs and it takes 2-3 days (depends on the visual backbone) to complete the whole training. The text encoder is unfrozen during the first two stages and then frozen for the final stage. The detailed configurations are summarized in Table 1. We follow the implementation of Detic [13] for the multi-dataset training. The learning rate is reduced by the factor of 10 when the iteration reaches the specified step in the table. Data augmentation includes random horizontal flip and scale jitter for resizing the input images. In the table, short side means the range of values for the shortest side and long side represents the maximum value for the longest side. During video-level training, for COCO [5] and RefCOCO+/g [12, 7], we apply two different augmentations on the same image to generate the pseudo videos for training. And for OVIS [8], we convert the dataset into a class-agnostic format to make it suitable for VOS training.

Inference Details. For both RVOS and VOS tasks, all the videos are rescaled to 480p for inference. And the score thresholds are set as 0.4 for VOS datasets and 0.3 for RVOS datasets, respectively. For these two tasks, both the masks in the first frame and previous frame are adopted as references.

Appendix C. More Results

How to Use Both Language and Mask Reference for RVOS? As shown in Figure 2, we design three strategies to utilize both language and mask references for RVOS: sequential M>L, sequential L>M and parallel. The ablation results are presented in Table 2, where it is evident that the sequential strategies yield poor performance, while the parallel strategy emerges as the most effective way to integrate both mask and language references. This finding is reasonable since the parallel strategy is a post-fusion process so it does not impact the visual features of the current frame.

Reference Frames for Mask Propagation. In this study, we investigate the effect of reference frames for mask prop-
Table 1: The detailed configurations for the three training stages.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Task</th>
<th>Dataset</th>
<th>Sampling Weight</th>
<th>Batch Size</th>
<th>Short Side</th>
<th>Long Side</th>
<th>GPU Number</th>
<th>Learning Rate</th>
<th>Weight Decay</th>
<th>Max Iteration Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>RIS</td>
<td>Visual Genome [4]</td>
<td>1</td>
<td>2</td>
<td>480 ∼ 800</td>
<td>1333</td>
<td>32</td>
<td>0.0002</td>
<td>0.0001</td>
<td>90000</td>
</tr>
<tr>
<td>II</td>
<td>RIS</td>
<td>RefCOCO+/g [12, 7]</td>
<td>1</td>
<td>2</td>
<td>480 ∼ 800</td>
<td>1333</td>
<td>16</td>
<td>0.0001</td>
<td>0.0001</td>
<td>90000</td>
</tr>
<tr>
<td>III</td>
<td>VOS</td>
<td>COCO [5]</td>
<td>0.40</td>
<td>2</td>
<td>480 ∼ 800</td>
<td>1333</td>
<td></td>
<td></td>
<td></td>
<td>90000</td>
</tr>
<tr>
<td></td>
<td>VOS</td>
<td>Youtube-VOS2019 [11]</td>
<td>0.30</td>
<td>2</td>
<td>320 ∼ 640</td>
<td>768</td>
<td></td>
<td></td>
<td></td>
<td>90000</td>
</tr>
<tr>
<td></td>
<td>VOS</td>
<td>LVOS [3]</td>
<td>0.20</td>
<td>2</td>
<td>320 ∼ 640</td>
<td>768</td>
<td></td>
<td></td>
<td></td>
<td>90000</td>
</tr>
<tr>
<td></td>
<td>VOS</td>
<td>OVIS [8]</td>
<td>0.10</td>
<td>2</td>
<td>320 ∼ 640</td>
<td>768</td>
<td></td>
<td></td>
<td></td>
<td>90000</td>
</tr>
<tr>
<td></td>
<td>RVOS</td>
<td>RefCOCO/g/+ [12, 7]</td>
<td>0.45</td>
<td>2</td>
<td>480 ∼ 800</td>
<td>1333</td>
<td></td>
<td></td>
<td></td>
<td>90000</td>
</tr>
<tr>
<td></td>
<td>RVOS</td>
<td>Ref-Youtube-VOS</td>
<td>0.55</td>
<td>2</td>
<td>320 ∼ 640</td>
<td>768</td>
<td></td>
<td></td>
<td></td>
<td>90000</td>
</tr>
</tbody>
</table>

Figure 2: Three strategies for utilizing both language and mask references for RVOS. (a) Sequential M->L: the visual features of current frame are fused with mask and language references sequentially. (b) Sequential L->M: the visual features of current frame are fused with language and mask references sequentially. (c) Parallel (used in the paper): the visual features of current frame are fused with mask reference and language reference, respectively. The two fused features are multiplied in the end.

Table 2: Ablation on the strategies for utilizing both language and mask references for RVOS. Experiments are conducted on Ref-Youtube-VOS. Our default settings are marked in gray.

<table>
<thead>
<tr>
<th>Variants</th>
<th>J &amp; F</th>
<th>J</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential M-&gt;L</td>
<td>13.8</td>
<td>14.6</td>
<td>13.0</td>
</tr>
<tr>
<td>Sequential L-&gt;M</td>
<td>9.5</td>
<td>9.6</td>
<td>9.5</td>
</tr>
<tr>
<td>Parallel</td>
<td>60.1</td>
<td>58.9</td>
<td>61.4</td>
</tr>
</tbody>
</table>

Table 3: Ablation on the reference frames used for mask propagation during inference. We use the final model with ResNet-50 visual backbone in this ablation. Our default settings are marked in gray.

Efficiency Comparison with Memory-based Methods.

We compare the efficiency of our UniRef and the representative memory-based method STCN [1] in Table 4. The results indicate that while our method is slightly slower than STCN on the Youtube-VOS dataset, it is much more efficient than STCN on the long-term video LVOS dataset. This is because the memory-based methods have linear memory complexity with respect to the video duration, while our method has constant memory cost. To better highlight the advantages of our method, we further plot the single-object FPS in Figure 3.

Appendix D. Visualization Results

We provide the visualization results of UniRef-L for RVOS tasks in Figure 4 and Figure 5. It can be seen that our model can segment the referred objects correctly and accurately in various challenging scenes, e.g., partial display, similar objects and fast moving, as illustrated in Figure 4.

Visualization results for the VOS tasks are presented in Figure 6 and Figure 7. Notably, our model reveals strong ability in handling long-term videos that typically last for...
Table 4: Efficiency comparison between our method and the representative memory-based method STCN. ‘YT-VOS18’ represents Youtube-VOS2018 dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mean Frames</th>
<th>Mean Objects</th>
<th>STCN FPS</th>
<th>UniRef FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>YT-VOS18</td>
<td>27</td>
<td>1.9</td>
<td>13.2</td>
<td>10.5</td>
</tr>
<tr>
<td>LVOS</td>
<td>574</td>
<td>1.3</td>
<td>4.8</td>
<td>19.3</td>
</tr>
</tbody>
</table>

Figure 3: FPS scaling of our method and the representative memory-based method STCN.

over a minute, such as those in LVOS [3]. As shown in Figure 7, our model can accurately segment the target objects throughout the whole video, despite the objects have significant pose variation. We further provide a video demo in the supplementary material.

References


a dog is waiting to catch the ball shown to him.
a hand is showing a ball to a dog. a lawn tennis ball in the hand of a person.
a whale swimming from the bottom to the top of the water.
a whale on the top right swimming underwater.
a skateboard being rolled through a road filled with cars and people.
a boy in black shorts and white tee shirt roller skating.

Figure 4: Visualization results on Ref-Youtube-VOS validation set.

a go-cart type car. a person driving the go cart.
person at the back of the go-cart without a helmet.
a man wearing a green helmet. a motor-bike.
a blonde haired girl dancing in a blue dress.

Figure 5: Visualization results on Ref-DAVIS17 validation set.
Figure 6: Visualization results on Youtube-VOS2018 validation set.

Figure 7: Visualization results on LVOS validation set.