SLVP: Self-supervised Language-Video Pre-training for Referring Video Object Segmentation

Jie Mei\textsuperscript{1}, AJ Piergiovanni\textsuperscript{2}, Jenq-Neng Hwang\textsuperscript{1}, Wei Li\textsuperscript{2}
\textsuperscript{1}University of Washington, Seattle jiemei@uw.edu, hwang@uw.edu
\textsuperscript{2}Google Research, Brain Team ajpiergi@google.com, mweili@google.com

Abstract

The referring video object segmentation (R-VOS) task requires a model to understand both referring expression and video input. Most recent works are mainly based on an encoder-decoder type of architecture. Although their text and visual encoders can benefit from separately pre-trained backbones, their decoder is trained from scratch on a combination of image/video segmentation datasets. However, pixel-wise annotation with referring expressions is extremely expensive which makes it challenging to further improve the performance. Due to the same reason, current vision-language pre-training works mainly focus on learning general feature representations for image-level or object-level tasks, which may be not optimal for the downstream pixel-level segmentation task. To bridge this gap, we present a general self-supervised language-video pre-training (SLVP) architecture. With the relatively cheap video caption dataset, SLVP can learn pixel-level features by introducing optical flow as the intermediate target during pre-training. Correspondingly, we propose simple transfer learning models that can reuse pre-trained modules for the downstream R-VOS task. Furthermore, the proposed general SLVP architecture can support either ‘language as query’ fusion or ‘vision as query’ fusion. Experiments show the superiority of the under-studied ‘vision as query’ method which can achieve better performance than the state-of-the-art methods on Ref-Davis\textsuperscript{17} and Ref-Youtube-VOS benchmarks even with fewer model parameters. We further adopt the challenging VISOR benchmark to the R-VOS task and our SLVP serves as the first strong baseline for R-VOS task on it.

1. Introduction

Referring video object segmentation (R-VOS) is an emerging multi-modal task, requiring the model to segment the specific object referred by a language description in all input frames. This task is gathering great attention in the research community because of the potential benefits to many applications in an interactive way, e.g., video editing and video surveillance. Compared with the traditional video-object segmentation (Semi-VOS) task \cite{31, 43} which assumes the availability of ground-truth mask annotation in the first frame during inference, the R-VOS task is more challenging because it requires the model to have a comprehensive understanding of the raw input videos and language description without any available mask during inference. Therefore, the model should know what the target object is described by the referring expression and then accurately segment it from the raw video.

Existing approaches for the R-VOS task can be categorized into three groups: (1) Bottom-up approaches. These approaches directly decode the target object masks using fully convolution networks (FCNs) \cite{21} based on vision-language fused features. (2) Top-down approaches. These approaches first segment all potential objects in each frame using an instance segmentation model then associate each object using a tracking algorithm. Finally, the target object masks are selected based on the language description. (3) Language as queries approaches. These methods are an encoder-decoder type of architecture, which takes advantage of the query mechanism in Transformer \cite{38}, treating referring expressions as queries and still using some convolution heads to decode the object mask.

These three streams of approaches have shown promising results but share an intrinsic limitation, i.e., only some parts of the model can benefit from pre-training such as backbones while the remaining parts of the model can only be trained from scratch on a combination of image/video referring segmentation datasets. This makes it challenging to further improve the model performance since pixel-level annotations are extremely expensive. Besides, current pre-training strategies are mainly designed for image-level or object-level tasks. For example, existing vision-language pre-training strategies can utilize a large amount of relatively cheap image-text pairs \cite{33} or object bounding-box-text pairs \cite{17} and inherently benefit down-
stream image-level or object-level tasks. On the other hand, self-supervised pre-training strategies may light the way to help pixel-level tasks since they show vision transformers such as [2] contain explicit information about the semantic segmentation of an image even when there are no labels during pre-training. However, most existing works focus on single-modality self-supervised pre-training such as DINO [2] and MAE [10].

Thus a natural question is when there are no pixel-level annotation datasets available, how to design a multi-modal i.e., language and video, self-supervised pre-training strategy to learn pixel-level semantic information. Furthermore, the second question is how to design the model architecture so that the pre-training strategy can benefit the whole model and further bring improvement to the downstream pixel-level and temporal-based R-VOS task. The above questions motivate us to design a synchronous pre-training and transfer-learning architecture to tackle the R-VOS task elegantly. In contrast to existing approaches, our decoder served for the fusion purpose can also benefit from the pre-training. Thanks to its simplicity, our general architecture supports not only the ‘language as query’ fusion method but also the under-studied ‘vision as query’ fusion method to be explained in Sec. 3.

The main contributions of this work are as follows. (1) We propose a self-supervised language-video pre-training strategy that can leverage relatively cheap video-caption datasets to make the decoder learn temporal semantic information based on video and text input. Experiments show the self-supervised pre-trained decoder can bring non-negligible improvement to the downstream R-VOS task. (2) We present a synchronous transfer learning architecture for the R-VOS task that can maximally benefit from the pre-trained model. It shares modules as much as possible with the pre-training architecture and employs a simple shared linear mask head on each token. (3) Experiments show the superior of the under-studied ‘vision as query’ method and that even when there are fewer segmentation training data or fewer model parameters, our proposed method can achieve on-par or even better performance than the state-of-the-art methods.

2. Related Work

Semi-supervised Video Object Segmentation. This related task assumes the ground-truth masks of target objects are available in the first frame during inference. Thus the model only needs to propagate these masks to other frames. Tracking the object based on feature matching is one mainstream approach in most recent works [4, 29, 40, 44]. STM [29] stores a memory of objects’ features in the past frames and utilizes the attention mechanism to perform feature matching to predict the masks in the current frame. This single-modality-based task does not require the model to understand any language description.

Referring Video Object Segmentation. Referring video object segmentation (R-VOS) is a multi-modality task. It provides the language description instead of the first frame’s mask ground truth for the target object during inference. Thus, it is a more challenging task. As mentioned previously, the current methods for R-VOS mainly follow three groups: (1) Bottom-up methods, which directly apply the image-based methods to each video frame independently [9, 14, 22, 49] without learning any temporal information to predict consistent masks. (2) Top-down methods, which first find many potential object tracklets using a tracking algorithm, and the target object is filtered out using a language grounding model [19] without considering the model complexity and heavy computation. (3) Language as query methods. The typical language as query methods, Referformer [41] and R²-VOS [18], propose a transformer-based [38] encoder-decoder architecture to fuse language and vision features and apply dynamic convolution operation to decode masks for each target object. Although their text encoder and vision encoders are pre-trained on non-segmentation datasets, their decoders do not benefit from any pre-training. This makes it challenging to improve the performance since pixel-level annotation with referring expressions is extremely expensive.

In contrast to the above approaches, we propose a synchronous pre-training and transfer-learning architecture to tackle the R-VOS task elegantly. The proposed self-supervised pre-training model shares a similar architecture with the transfer-learning pipeline. Thus, our decoder can benefit from the pre-training on relatively cheap video-caption datasets. Thanks to its simplicity, i.e., applying a shared linear mask head on each token, our architecture supports two fusion methods, i.e., ‘language as query’ or ‘vision as query’. For both methods, we explore the gains from the proposed self-supervised language-video pre-training strategy.

Self-Supervised Learning. Supervised training demonstrates outstanding performance in many tasks [13, 23–26, 26, 48, 50]. But with Transformers [38] successfully becoming a general building block in both language and vision, the computer vision community starts to bring in self-supervised representation learning methods such as MAE [10, 11] by referring to denoising/masked autoencoding methodology [39] introduced in BERT [8]. The features from the pre-trained model can achieve outstanding performance in image-level tasks such as zero-shot image classification. Recently, contrastive learning [3, 12, 30, 42] which models image similarity and dissimilarity between augmented views is getting popular. Besides single-modality training, existing vision-language pre-training strategies can utilize a large amount of relatively cheap image-text pairs [33] or object bounding-box-text pairs [17] and inher-
ently benefit downstream image-level or object-level tasks. However, these approaches mainly focus on learning a good encoder. Although MAE [11] uses a decoder during the pre-training to reconstruct the original frame, the decoder is discarded for the downstream tasks. Our proposed self-supervised language-video pre-training strategy aims to also make the decoder learn temporal semantic information based on video and text input by leveraging relatively cheap video-caption datasets so that it can bring improvement to the downstream pixel-level R-VOS task.

3. Methodology

Given a video clip $V = \{v_t\}_{t=1}^T$ with $T$ frames and one corresponding referring expression $R = \{r_t\}_{t=1}^L$ with $L$ words, the R-VOS model is expected to produce $T$-frame binary segmentation masks for the referred object, i.e., $M = \{m_t\}_{t=1}^T$, $m_t \in \mathbb{R}^{H \times W}$, where $H$ and $W$ are the frame height and width. Our proposed general architecture, called self-supervised language-video pre-training (SLVP), is an encoder-decoder architecture based on pure transformer modules. By applying a shared linear head on each token to get the prediction, the proposed general architecture can support not only ‘language as query’ fusion but also ‘vision as query’ fusion.

Furthermore, due to the consistency between the proposed pre-training and transfer-learning architectures, our decoder served for the fusion purpose can benefit from pre-training and contribute a non-negligible improvement to the downstream R-VOS task. Details of using SLVP in pre-training and transfer-learning for ‘vision as query’ and ‘language as query’ fusion methods are in Fig. 1.

3.1. Vision as Query

Existing methods mainly use ‘language as query’ fusion. For example, Referformer [41] introduces some learnable queries conditioned on text features as input to the decoder, adopted from Deformable-DETR [1, 51] while our general SLVP architecture supports under-studied ‘vision as query’ fusion as shown in Fig. 1a and 1b.

Self-supervised Pre-training. We use a relatively cheap non-segmentation dataset for pre-training, i.e., a video-caption dataset. The intermediate pre-training target is to predict the optical flow, $o_{t,1} \in \mathbb{R}^{H \times W \times 2}$, between the first frame and any $i$th frame based on the caption and video input without any optical flow ground truth. Thus, self-supervised loss function is applied on the original $i$th frame and an RGB image, $I_i$, reconstructed by applying $o_{t,1}$ on the original first frame:

$$I_i = \text{Flow}(o_{t,1}, V_1),$$

$$L_{MSE} = |I_i - V_1|^2,$$

where the mean square error is used as the loss function, $\text{Flow}(\cdot)$ operation is denoted as the blue module in Fig. 1a and there are no learnable parameters because this operation just moves every pixel in the first frame $V_1$ to a new location based on $o_{t,1}$.

To predict the optical flow $o_{t,1} \in \mathbb{R}^{H \times W}$ for the $i$th frame from a sequence of text features $f_r \in \mathbb{R}^{L \times d}$ output from the text encoder and a sequence of frames’ features $f_v \in \mathbb{R}^{(T \times N) \times d}$ ($N$ is the number of tokens of one frame, i.e., the number of patches of one frame), we first use a Transformer that consists of self-, cross-attentions and feed-forward network as our decoder to fuse the text features and frames feature by using frames’ features as query and text features as key and value:

$$f_{\text{fused}} = T(f_v + E_{pos} + E_{tem}, f_r),$$

where $T$ is a Transformer, $E_{pos} \in \mathbb{R}^{N \times d}$ and $E_{tem} \in \mathbb{R}^{T \times d}$ are the learnable positional and temporal encodings respectively, the first input of $T$ is the query and the second is key and value. $E_{pos}$ and $E_{tem}$ will be repeated by $T$, $N$ times respectively when added to $f_v$. The output $f_{\text{fused}} \in \mathbb{R}^{(T \times N) \times d}$ is the fused sequence of frames’ features.
Finally, inspired by MAE [11], we use a shared linear layer on each token of $f_{\text{fused}} \in \mathbb{R}^{(T \times N) \times d}$ to transform the last dimension from $d$ to $p \times p \times 2$, where $p$ is the patch size and 2 represents two-dimensional optical flow values for each pixel:

$$\hat{f}_{\text{fused}} = \text{Linear}(f_{\text{fused}}),$$

where $\hat{f}_{\text{fused}} \in \mathbb{R}^{(T \times N) \times (p \times p \times 2)}$ is the generated optical flow for each image patch. Then we reshape $\hat{f}_{\text{fused}}$ into $\mathbb{R}^{T \times H \times W \times 2}$ to get optical flow, $o_{i,1} \in \mathbb{R}^{H \times W \times 2}$, for the $i$th frame. A predicted optical flow demo is shown in Fig. 2.

In detail, our vision encoder is applied on each frame independently while the decoder takes in the concatenated sequence of frames’ features. This is because it is necessary to allow the decoder to observe nearby frames before it can predict the meaningful optical flow. Thus, the proposed architecture is a temporal-based method. When only observing the input and output of the proposed pre-training architecture, we can see it predicts optical flow for each input image patch.

**Transfer Learning.** The transfer learning architecture shares all the modules with the pre-training architecture except for the last linear layers. Thus all encoders, decoders, and learnable positional/temporal encodings are initialized with the pre-trained weights.

Since the last shared linear layer in the pre-training is trained to predict optical flow, we replace it with another two reinitialized linear heads, i.e., one for bounding box regression and the other for binary mask prediction for the downstream R-VOS task as shown in Fig. 1b. The box linear head is applied on the max-pooled fused features, $f_{\text{max}} \in \mathbb{R}^{T \times d}$, and the mask linear head is shared among all tokens:

$$f_{\text{max}} = \text{MaxPooling}(f_{\text{fused}}),$$

where $B \in \mathbb{R}^{T \times 4}$ is the predicted bounding box for $T$ frames, $\sigma(\cdot)$ is the sigmoid operation, $M \in \mathbb{R}^{(T \times N) \times (p \times p \times 1)}$ is the predicted binary mask for each image patch. $M$ will be reshaped into $\mathbb{R}^{T \times H \times W}$ and then we can get a binary mask, $m_i \in \mathbb{R}^{H \times W}$, for the $i$th frame.

When only observing the input and output of the proposed transfer learning architecture, we can see it predicts binary masks for each input image patch based on temporal information and referring expression. It also outputs the regressed bounding boxes for each frame.

For box loss, we use GIoU [35] and L1 loss; for mask loss, we use Dice loss [37] and binary cross-entropy.

**Box loss.** If we denote a predicted bounding box as $B_p(x_1, y_1, x_2, y_2)$ and the ground truth bounding box as $B_g(X_1, Y_1, X_2, Y_2)$, then GIoU [35] is defined as following:

$$\text{IoU} = \frac{|B_p \cap B_g|}{|B_p \cup B_g|},$$

$$\text{GIoU} = \text{IoU} - \frac{|C \setminus (B_p \cup B_g)|}{|C|},$$

where $C$ is the smallest enclosing bounding box for $B_p$ and $B_g$; the nominator in the second equation is the area occupied by $C$ excluding $B_p$ and $B_g$. IoU has a major weakness when used as a loss function: if $\text{IoU}(B_p, B_g) = 0$, IoU can not reflect if two bounding boxes are in the vicinity of each other or very far from each other. However, GIoU takes the smallest enclosing bounding box into consideration to overcome this issue. Finally, the GIoU loss is $1 - \text{GIoU}$.

L1 loss is a straightforward loss between four coordinates (top-left point and bottom-right point) of $B_p$ and $B_g$:

$$L_1 = |x_1 - X_1| + |y_1 - Y_1| + |x_2 - X_2| + |y_2 - Y_2|.$$  

**Mask loss.** Dice loss [37] and binary cross entropy (BCE) loss are as follows:
\[\text{Dice Loss} = 1 - \frac{2 \sum_{i=1}^{N} m_i g_i}{\sum_{i=1}^{N} m_i^2 + \sum_{i=1}^{N} g_i^2}\]

\[\text{BCE Loss} = \frac{1}{N} \sum_{i=1}^{N} - (g_i \log(m_i) + (1 - g_i) \log(1 - m_i)),\]

where \(N\) is the total number of pixels; \(m_i\) and \(g_i\) are the values in the predicted mask \(M\) and ground-truth binary mask \(G\) respectively. Finally, the total loss is the summation of Dice loss, BCE loss, GIoU loss, and L1 loss. The coefficients for losses are set as \(\lambda_{L1} = 5\), \(\lambda_{dice} = 5\), \(\lambda_{giou} = 1\), and \(\lambda_{bce} = 1\).

3.2. Language as Query

Our general SLVP architecture also supports the ‘language as query’ fusion method with a slight modification as shown in Fig. 1c and 1d.

Self-supervised Pre-training. Since we still hold the same spirit mentioned in the pre-training, i.e., predicting the optical flow for each image patch, we have to make sure the length of tokens output from the decoder is the same as the input sequence of frames’ features \(f_c \in \mathbb{R}^{(T \times N) \times d}\). Thus, we create a shared learnable query token, \(\hat{q} \in \mathbb{R}^d\), and repeat it by \(T \times N\) times to get \(\hat{q} \in \mathbb{R}^{(T \times N) \times d}\), denoted as gray cubes in Fig. 1c. Then we fuse the text features and frame features by using text’s features \(f_r \in \mathbb{R}^{L \times d}\) concatenated by \(\hat{q}\) as query and frame features as key and value:

\[f_{\text{fused}} = \mathbb{T}(\text{cat}(f_r, \hat{q} + E_{\text{pos}} + E_{\text{tem}}, f_v + E_{\text{pos}} + E_{\text{tem}}))\]

where output \(f_{\text{fused}} \in \mathbb{R}^{(L+(T \times N)) \times d}\) is the fused sequence of features. But we only use the last \(T \times N\) tokens as the input to the later shared linear layer to predict the optical flow for each image patch. The other parts including the loss function in the architecture are the same as those of the ‘vision as query’ architecture.

Transfer Learning. Same as the ‘vision as query’ architecture, we also replace the last shared linear layer in the pre-training with another two reinitialized linear heads for bounding box regression and binary mask prediction respectively as shown in Fig. 1d. In both the ‘vision as query’ and ‘language as query’ methods, our decoder served for the fusion purpose can benefit from the self-supervised pre-training.

4. Experiments

4.1. Implementation Details

Model Settings. We use T5-pretrained text encoder [34], and CoCa-pretrained visual encoder [46], denoted as ‘Pre-trained Es’ in all experiment tables. Each of the encoders has 12 transformer self-attention layers. We use an 8-layer transformer, that consists of self-, cross-attention, and feedforward networks, as our decoder. For both pre-training and transfer learning, we use 18 as patch size, \(360 \times 648\) as the frame resolution, 64 as the maximum sentence length, and 4 as video clip length.

Pre-training Details. During our pre-training, we freeze the text and vision encoders. This is because we want to see the improvement contributed only by the self-supervised pre-trained decoder on the downstream pixel-level R-VOS task. Besides, we use the sliding windows to obtain the short clips from videos and each clip consists of 4 randomly sampled frames with 6 as the sampling rate to cover enough object movement. There is no augmentations used during pre-training.

Transfer Learning Details. In both ‘vision as query’ and ‘language as query’ methods, we concatenate the bounding box prediction with each of the fused tokens before applying the mask linear head so that the mask prediction can consider the object location. We also use random-flip, random-crop augmentation, and color-jittering during transfer learning, denoted as ‘Aug’, in all experiment tables. During the inference on R-VOS benchmarks, we directly output the predicted segmentation masks without any post-processing such as mask propagation [44] used in some previous works so that we can see the authentic segmentation improvement contributed by the pre-trained decoder.

4.2. Datasets and Metrics

Pre-training Dataset. We use the large-scale Spoken Moments in Time (S-MiT) dataset [28] as the pre-training dataset. It consists of 500K pairs of video clips and corresponding captions depicting a broad range of different dynamic events. The captions are semantically rich compared to simple action labels. S-MiT covers a subset of the videos in the Moments in Time dataset [27]. The clips are 3 seconds long. On average, the captions have a length of 18 words and contain 1.58 verbs. Thus these attributes make it well-suited for our self-supervised pre-training target, i.e., predicting the optical flow for frames.

R-VOS Benchmarks. After the pre-training, we fine-tune and evaluate the models on Ref-Davis17 [16] and Ref-Youtube-VOS [36]. Ref-Youtube-VOS [36] is a large-scale benchmark that covers 3,978 videos with about 15K language descriptions. Among them, 3,471 videos are for training and 202 videos are for validation. For a fair comparison, we follow ReferFormer’s [41] training setup, i.e., before finetuning on Ref-Youtube-VOS, we also first fine-tune our pre-trained model on RefCOCO+/g [15,47]. Ref-Davis17 [16] is a traditional R-VOS benchmark built upon DAVIS17 [32] by providing the language description for a specific object in each video and contains 90 videos with 1,544 expression sentences describing 205 objects in total.
Table 1. Comparison with the state-of-the-art methods on Ref-Davis17 [16] and Ref-Youtube-VOS [36]. Decoder∗ represents if the decoder can benefit from self-supervised pre-training.

<table>
<thead>
<tr>
<th>Method</th>
<th>Vision Encoder</th>
<th>Query type</th>
<th>#params</th>
<th>Ref-Davis17</th>
<th>Ref-Youtube-VOS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>J</td>
<td>F</td>
</tr>
<tr>
<td>CMSA [45]</td>
<td>ResNet-50</td>
<td>-</td>
<td>-</td>
<td>32.2</td>
<td>37.2</td>
</tr>
<tr>
<td>CMSA+RNN [45]</td>
<td>ResNet-50</td>
<td>-</td>
<td>-</td>
<td>36.9</td>
<td>43.5</td>
</tr>
<tr>
<td>URVOS [36]</td>
<td>ResNet-50</td>
<td>-</td>
<td>-</td>
<td>47.3</td>
<td>56.0</td>
</tr>
<tr>
<td>ReferFormer [41]</td>
<td>ResNet-50</td>
<td>language</td>
<td>186M</td>
<td>55.8</td>
<td>61.3</td>
</tr>
<tr>
<td>R²-VOS [18]</td>
<td>ResNet-50</td>
<td>language</td>
<td>186M</td>
<td>57.2</td>
<td>62.4</td>
</tr>
<tr>
<td>ReferFormer [41]</td>
<td>Swin-L</td>
<td>language</td>
<td>360M</td>
<td>57.6</td>
<td>63.4</td>
</tr>
<tr>
<td>SLVP</td>
<td>ViT-B</td>
<td>vision</td>
<td>258M</td>
<td>✓</td>
<td>57.6</td>
</tr>
</tbody>
</table>

Figure 3. Demos of ‘Vision as Query’ model on Ref-Davis17 (left) and VISOR (right). Each row has two frames randomly sampled from the same video. The referring expression input to the model is displayed on the top or bottom of each row.

The dataset is split into 60 videos for training and 30 videos for validation. Since there are two annotators and each of them gives the first frame and full-video language description for each referred object, we report the results by averaging the evaluation scores. For a fair comparison, following [41], we also fine-tune the pre-trained model on RefCOCO+/g [15, 47] and Ref-Youtube-VOS [36] and then directly test it on Ref-Davis17 without finetuning.

Adopted R-VOS Benchmark EPIC-KITCHENS VI-
SOR [6] is a new dataset of pixel annotations and a benchmark suitable for segmenting hands and active objects in egocentric videos. It annotates videos from EPIC-KITCHENS [5] and consists of 174.4K masks from 32.8K frames of 33 kitchens covering 242 entity classes for training and 41.5K masks from 7.7K frames of 24 kitchens covering 182 entity classes for validation. There are 5 unseen kitchens and 9 zero-shot entity classes in the validation. Thus it comes with a new set of challenges not encountered in existing R-VOS benchmarks. It is proposed for the single-modality Semi-VOS task. We adopt it into the R-VOS task by treating the object names as the referring expressions. Our proposed method serves as the first strong baseline for the R-VOS task on this benchmark.

Evaluation Metrics. Following the protocol used by [32, 41, 43], we use the following evaluation metrics: region similarity defined by Jaccard Index/Intersection over Union (J), contour accuracy defined by Boundary F-Measure (F) and their average value (J&F).

Reference Performance On the adopted VISOR benchmark, our proposed method serves as the first strong baseline for the R-VOS task. Thus, we also report STM [29] method trained with VISOR and additional COCO [20] data under the relatively easier Semi-VOS task as the reference performance. COCO [20] is used for temporal-based training by synthesizing a video clip of 3 images from random affine transforms.

We also demonstrate when there are fewer pixel-level annotated datasets, our proposed SLVP can still bring non-negligible improvement to the downstream R-VOS task.

4.3. Vision as Query Results

Ref-Davis17 and Ref-Youtube-VOS Benchmarks. The results and demos are in Table 1, Fig. 3 Fig. 4, and Fig. 5. Even with less number of parameters, our model’s performance can surpass Referformer [41] by +0.8 in terms of J&F on Ref-Davis17, and +2.0 on Ref-Youtube-VOS.

Ablation Study. Table 2 shows the ablation study on description-rich Ref-Davis17. It shows that the self-
**Ablation Study** of ‘Vision as Query’ Model on Ref-Davis17 Benchmark. *Decoder* represents if the decoder uses pre-trained weights from the proposed self-supervised pre-training stage. ‘Pretrained-Es’ represents pretrained encoders. ‘Frozen T-E’ represents the frozen text encoder. ‘Augs’ represents augmentations.

<table>
<thead>
<tr>
<th>Segmentation Training Datasets</th>
<th>Pretrained-Es</th>
<th>Augs</th>
<th>Frozen T-E</th>
<th>Decoder*</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref-Davis17 [16]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>10.4</td>
<td>21.7</td>
<td>16.1</td>
</tr>
<tr>
<td>RefCOCO/g/+ [15,47], Ref-Davis17 [16]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>37.3</td>
<td>41.0</td>
<td>39.2</td>
</tr>
<tr>
<td>RefCOCO/g/+ [15,47], Ref-Youtube-VOS [36]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>40.8</td>
<td>49.2</td>
<td>45.0</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>41.7</td>
<td>50.5</td>
<td>46.1</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>46.2</td>
<td>55.7</td>
<td>50.5</td>
</tr>
</tbody>
</table>

Table 3. Performance of ‘Vision as Query’ Model on VISOR Benchmark. We adopt VISOR benchmark into the more challenging R-VOS task by treating object names as referring expressions. Our proposed ‘Vision as Query’ method serves as the first strong baseline for the R-VOS task. Thus, we also report STM [29] method trained with VISOR and additional COCO [20] data under the relatively easier Semi-VOS task as the reference performance. ‘Frozen T-E’ represents the frozen text encoder. ‘Augs’ represents augmentations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrained-Es</th>
<th>Augs</th>
<th>Frozen T-E</th>
<th>Decoder*</th>
<th>Segmentation Training Datasets</th>
<th>Task</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLVP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>VISOR [7]</td>
<td>RVOS</td>
<td>49.7</td>
<td>53.6</td>
<td>51.7</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>VISOR [7]</td>
<td>RVOS</td>
<td>56.8</td>
<td>60.2</td>
<td>58.5</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>VISOR [7]</td>
<td>RVOS</td>
<td>66.6</td>
<td>74.7</td>
<td>70.7</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>VISOR [7]</td>
<td>RVOS</td>
<td>66.4</td>
<td>74.2</td>
<td>70.3</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>VISOR [7]</td>
<td>RVOS</td>
<td>70.8</td>
<td>78.8</td>
<td>74.8</td>
</tr>
<tr>
<td>STM [29] as reference performance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>VISOR [7]</td>
<td>VOS</td>
<td>60.6</td>
<td>64.9</td>
<td>62.8</td>
</tr>
<tr>
<td>STM [29] as reference performance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>MS-COCO [20] + VISOR [7]</td>
<td>VOS</td>
<td>73.6</td>
<td>78.0</td>
<td>75.8</td>
</tr>
</tbody>
</table>

Table 4. Comparison of ‘Language as Query’ and ‘Vision as Query’ of SLVP architecture on Ref-Davis17 [16].

<table>
<thead>
<tr>
<th>Method</th>
<th>#params</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language as Query</td>
<td>258M</td>
<td>54.1</td>
<td>61.3</td>
<td>57.7</td>
</tr>
<tr>
<td>Vision as Query</td>
<td>258M</td>
<td>57.6</td>
<td>64.9</td>
<td>61.3</td>
</tr>
</tbody>
</table>

The self-supervised pre-trained decoder brings non-negligible (+4.4 in terms of $J&F$) improvement. Besides, the performance of the ‘vision as query’ model also gains with pre-trained encoders and augmentations. Interestingly, freezing the text encoder during transfer learning brings +1.1 improvement in terms of $J&F$ on Ref-Davis17. This is because Ref-Davis17 is a relatively small benchmark with longer referring descriptions. Thus finetuning the text encoder may make the model overfit on the training data of Ref-Davis17.

**Adopted VISOR Benchmark.** The results and demos are in Table. 3 and Fig. 3. Our proposed method serves as the first strong baseline for the R-VOS task on this benchmark. Thus, we also report STM [29] method trained with VISOR and additional COCO [20] data under the relatively easier Semi-VOS task as the reference performance. We can see without the proposed pre-training strategy, our ‘vision as query’ architecture can already surpass STM [29] trained on VISOR-only which is under the relatively easier Semi-VOS setting. After initializing our decoder with the self-supervised pre-trained weights, the performance is boosted by +4.1 in terms of $J&F$, which is only 1.0 lower than the performance of STM [29] with COCO [20] as an additional pixel-level training dataset.

Besides, the performance of the ‘vision as query’ model also gains with pre-trained encoders and augmentations. Interestingly, freezing the text encoder during transfer learning hurts −0.4 in terms of $J&F$ on the VISOR. This is because VISOR is a relatively large benchmark but with short entity names as referring expressions thus the text encoder won’t overfit on the training data during finetuning.

**4.4. Language as Query Results**

**Ref-Davis17 Benchmark.** In Table. 4, with the same number of parameters, our ‘language as query’ model achieves worse performance than our ‘vision as query’ model, indicating the superiority of the ‘vision as query’ fusion method under our proposed SLVP architecture.

**Adopted VISOR Benchmark.** In Table. 5, after initializing our decoder with the self-supervised pre-trained weights, the performance is boosted by +2.0 in terms of $J&F$. This ‘language as query’ model also can surpass STM [29] trained on VISOR-only which is under the relatively easier Semi-VOS setting.

**Ablation Study.** During transfer learning, we further freeze the vision encoder and find performance drops of −4.8 on Ref-Davis17 in Table. 6 and −7.5 in Table. 5 on
Table 5. Performance of ‘Language as Query’ Model on VISOR Benchmark. We adopt the VISOR benchmark into the more challenging R-VOS task by treating object names as referring expressions. Thus, we also report STM [29] method trained with VISOR and additional COCO [20] data under the relatively easier Semi-VOS task as the reference performance. ‘Frozen T-E’ and ‘Frozen V-E’ represent the frozen text encoder and vision encoder. ‘Augs’ represents augmentations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrained-Es</th>
<th>Augs</th>
<th>Frozen T-E</th>
<th>Frozen V-E</th>
<th>Decoder*</th>
<th>Segmentation Training Datasets</th>
<th>Task</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLVP</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td>VISOR [7]</td>
<td>RVOS</td>
<td>64.3</td>
<td>71.2</td>
<td>67.8</td>
</tr>
<tr>
<td>SLVP</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td>√</td>
<td>VISOR [7]</td>
<td>VISOR</td>
<td>65.4</td>
<td>73.0</td>
<td>69.2</td>
</tr>
<tr>
<td>SLVP</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td>VISOR [7]</td>
<td>VISOR</td>
<td>59.5</td>
<td>63.8</td>
<td>61.7</td>
</tr>
<tr>
<td>SLVP</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td>VISOR [7]</td>
<td>VISOR</td>
<td>67.2</td>
<td>75.1</td>
<td>71.2</td>
</tr>
<tr>
<td>STM [29] as reference performance</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td>VISOR [7]</td>
<td>VOS</td>
<td>60.6</td>
<td>64.9</td>
<td>62.8</td>
</tr>
<tr>
<td>STM [29] as reference performance</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td>MS-COCO [20] + VISOR [7]</td>
<td>73.6 73.6 73.6</td>
<td>78.0</td>
<td>78.0</td>
<td>75.8</td>
</tr>
</tbody>
</table>

Table 6. Ablation Study of ‘Language as Query’ Model on Ref-Davis17 Benchmark. ‘Pretrained-Es’ represents if the decoder uses pre-trained weights from the proposed self-supervised pre-training stage. ‘Pretrained-Es’ represents pretrained encoders. ‘Frozen T-E’ and ‘Frozen V-E’ represent the frozen text encoder and vision encoder. ‘Augs’ represents augmentations.

<table>
<thead>
<tr>
<th>Segmentation Training Datasets</th>
<th>Pretrained-Es</th>
<th>Augs</th>
<th>Frozen T-E</th>
<th>Frozen V-E</th>
<th>Decoder*</th>
<th>J</th>
<th>F</th>
<th>J&amp;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref-Davis 17 [16]</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.5</td>
<td>16.0</td>
<td>14.3</td>
</tr>
<tr>
<td>Ref-Davis 17 [16]</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>43.3</td>
<td>51.5</td>
<td>47.4</td>
</tr>
<tr>
<td>Ref-COCO/g/+ [15, 47], Ref-Youtube-VOS [36]</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>39.6</td>
<td>45.5</td>
<td>42.6</td>
</tr>
<tr>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>45.2</td>
<td>55.8</td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>54.1</td>
<td>61.2</td>
<td>57.7</td>
</tr>
</tbody>
</table>

VISOR. This indicates ‘language as query’ method also relies on strong visual features to perform R-VOS task, indirectly indicating the superiority of the ‘vision as query’ method. Besides, we still observe +3.1 improvement in terms of J&F brought by the self-supervised pre-trained decoder on Ref-Davis17 in Table 6 but only +1.0 improvement on VISOR in Table 5. This indicates that the ‘language as query’ method can benefit from the pre-trained decoder mainly when finetuning on description-rich datasets.

5. Conclusion

We proposed a general architecture, i.e., SLVP, for the R-VOS task which can support either the ‘vision as query’ or ‘language as query’ fusion method. Experiments showed the superiority of the under-studied ‘vision as query’ method on both description-rich and -poor datasets. Specifically, we presented an effective self-supervised language-vision pre-training strategy to benefit the decoder, enabling non-negligible improvement to the downstream R-VOS task. Existing works do not explore how to make the decoder benefit from the pre-training, leaving it challenging to further improve the performance. Our work is a step in trying to bridge this gap. Besides demonstrating our ‘vision as query’ model’s better performance on well-studied Ref-Davis17 and Ref-Youtube-VOS benchmarks even with fewer model parameters, we further adopt the challenging VISOR benchmark to the R-VOS task. Our ‘vision as query’ model serves as the first strong baseline. We sincerely acknowledge the inspiring discussions with Chen Sun and Anelia Angelova at Google.
References


[23] Jie Mei, Jenq-Neng Hwang, Suzanne Romain, Craig Rose, Braden Moore, and Kelsey Magrane. Absolute 3d pose es-


