Language-Aware Vision Transformer for Referring Segmentation

Zhao Yang*, Jiaqi Wang*, Xubing Ye*, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H.S. Torr

Abstract—Referring segmentation is a fundamental vision-language task that aims to segment out an object from an image or video in accordance with a natural language description. One of the key challenges behind this task is leveraging the referring expression for highlighting relevant positions in the image or video frames. A paradigm for tackling this problem in both the image and the video domains is to leverage a powerful vision-language ("cross-modal") decoder to fuse features independently extracted from a vision encoder and a language encoder. Recent methods have made remarkable advances in this paradigm by exploiting Transformers as cross-modal decoders, concurrent to the Transformer's overwhelming success in many other vision-language tasks. Adopting a different approach in this work, we show that significantly better cross-modal alignments can be achieved through the early fusion of linguistic and visual features in intermediate layers of a vision Transformer encoder network. Based on the idea of conducting cross-modal feature fusion in the visual feature encoding stage, we propose a unified framework named Language-Aware Vision Transformer (**LAVT**), which leverages the well-proven correlation modeling power of a Transformer encoder for excavating helpful multi-modal context. This way, accurate segmentation results can be harvested with a light-weight mask predictor. One of the key components in the proposed system is a dense attention mechanism for collecting pixel-specific linguistic cues. When dealing with video inputs, we present the **video LAVT** framework and design a 3D version of this component by introducing multi-scale convolutional operators arranged in a parallel fashion, which can exploit spatio-temporal dependencies at different granularity levels. We further introduce **unified LAVT** as a unified framework that could handle both image and video inputs with enhanced segmentation capability on unified referring segmentation task. Our methods surpass previous state-of-the-art methods on seven benchmarks for referring image segmentation and referring video segmentation. The code to reproduce our experiments is available at LAVT-RS.

Index Terms—Referring Segmentation, Language-Aware Vision Transformer, Multi-Modal Understanding.

1 Introduction

Given an image or a sequence of images, and a text description of the target object, referring segmentation aims at predicting pixel-wise masks that delineate the object [1], [2], [3]. It yields great value for various applications such as language-interfaced human-robot interaction, image/video editing, and image/video generation. In contrast to conventional single-modality visual segmentation tasks, which are based on fixed category conditions [4], [5], [6], referring segmentation has to deal with the much richer vocabularies and syntactic varieties of human natural languages. In this task, the target object is identified based on a free-form expression, which uses words and phrases governed by syntactic rules to describe entities, actions, positions, and other aspects. Therefore, the key challenge of this task is to exploit visual features that are relevant to the given text conditions.

There has been a growing effort devoted to referring segmentation over the past few years. A widely adopted paradigm in both the image and the video domains is to first independently extract vision and language features from different encoder networks, and then fuse them together to make predictions with a cross-modal decoder. Concretely, the fusion strategies include recurrent interaction [7], [8], cross-modal attention [9], [10], [11], [12], [13], multi-modal graph reasoning [14], linguistic structure-guided context modeling [15], dynamic filtering [3], capsule routing [16], *etc*. Recent advances (*e.g.*, [17], [18], and [19]) bring performance improvements via employing a cross-modal Transformer [20] decoder (illustrated in Fig. 1 (a)) to learn more effective cross-modal alignments, which is in concurrence with Transformer's overwhelming success in many other vision-language tasks [21], [22], [23], [24].

Although great progress has been achieved, the potential of the Transformer for enhancing referring segmentation is still far from fully explored. Specifically, cross-modal interactions mainly happen after feature encoding, and a cross-modal decoder is mostly responsible for aligning the visual and linguistic features. As a result, previous methods fail to effectively leverage the rich Transformer layers in the encoder for excavating helpful multi-modal context. To address these issues, a potential solution is to exploit a visual encoder network for jointly embedding linguistic and visual features during visual encoding.

Accordingly, we propose a **L**anguage-**A**ware **V**ision **T**ransformer (**LAVT**) network, in which visual features are encoded together with linguistic features, being "aware" of their relevant linguistic context at each spatial location. As shown in Fig. 1 (b), LAVT makes full use of the multi-stage design in a modern vision Transformer backbone, leading to a hierarchical language-aware visual encoding scheme. Specifically, we densely integrate linguistic features with visual features via a **P**ixel-**W**ord **A**ttention **M**odule (**PWAM**).

- *Zhao Yang and Philip H.S. Torr are with the Department of Engineering Science, University of Oxford.*
- *Jiaqi Wang and Kai Chen are with Shanghai AI Lab.*
- *Xubing Ye and Yansong Tang are with Shenzhen International Graduate School, Tsinghua University.*
- *Hengshuang Zhao is with the Department of Computer Science, The University of Hong Kong.*
- ** denotes co-first authorship.*

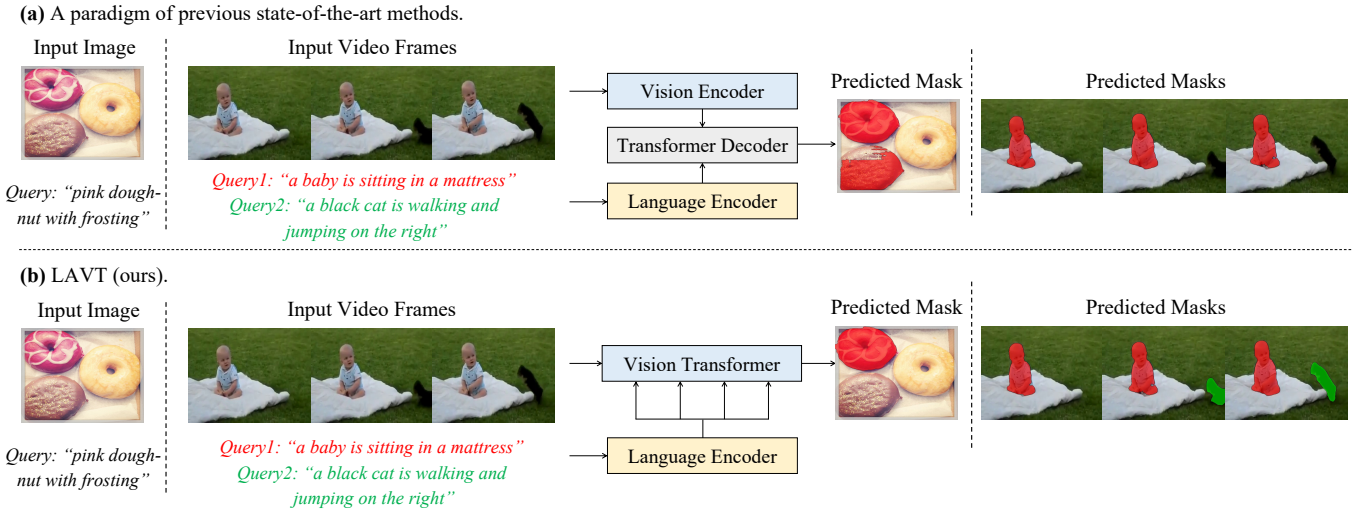

Fig. 1. The task of referring segmentation takes an image or a sequence of images (a video) and text descriptions as inputs, and predicts a mask delineating the object specified by the text in the image(s). (a) The previous state-of-the-art methods (*i.e.*, VLT [17] for images and ReferFormer [19] for videos) leverage a vision-language Transformer decoder for cross-modal feature fusion. (b) Conversely, we propose to directly integrate linguistic information with visual features in intermediate levels of a vision Transformer network, where beneficial vision and language cues are jointly exploited. A light-weight mask predictor can thus readily replace the complicated cross-modal decoder in previous counterparts.

The beneficial vision-language cues are then exploited by the following Transformer blocks, *e.g.*, [25] and [26], in the next encoder stage. This approach enables us to forgo a complicated cross-modal decoder, as accurate segmentation results can be harvested from the language-aware visual features via a lightweight mask predictor.

Moreover, we extend our proposed LAVT to the video domain. We first show that LAVT can readily serve as a simple yet effective baseline for the task of referring video segmentation, by simply switching its image-based vision Transformer layers [25] to the corresponding video-based version [26]. In order to better integrate spatio-temporal and linguistic representations, we further present an analogous framework named **video LAVT**, in which the PWAM is extended to the 3D domain, leading to what we refer to as the 3D PWAM. The 3D PWAM leverages multi-scale 3D convolutions at several places within the module to extract spatio-temporal information from the respective inputs. Such convolutions are placed on top of visual features, linguistic features, and integrated multi-modal features, the choices of which we carefully validate through experiments. At each place, we consistently apply the pairwise construct of two multi-scale convolutions placed side-by-side, leading to an overall design that presents clarity and conformity with the original PWAM. The dynamic nature of videos presents a variety of challenges to the segmentation of objects from language expressions, and in broad terms, effective temporal information modeling can be a generic cure to such challenges. Later in the experiment section, we illustrate several common types of challenges that we observe and demonstrate the efficacy of the video LAVT in addressing them via qualitative analyses.

Building upon LAVT and video LAVT, we further propose a framework named unified LAVT that is capable of concurrently processing both image and video inputs. For the multi-level outputs from the visual encoding backbone of video LAVT, which are temporal, we enhance the local modeling for the image input to allow unified LAVT to handle temporal information flexibly while strengthening its modeling of local information. This approach enables the model to handle both video and image inputs simultaneously, establishing a unified framework.

To evaluate the effectiveness of the proposed methods, we conduct a series of experiments on seven widely adopted benchmark datasets, including RefCOCO [27], RefCOCO+ [27], G-Ref (UMD partition) [28], and G-Ref (Google partition) [29] for referring image segmentation, and Refer-YouTube-VOS [12], A2D-Sentences [3], Ref-DAVIS-17 [30] and JHMDB-Sentences [3] for referring video segmentation. Extensive experimental results demonstrate the promising competitiveness of our methods in relation to state-of-the-art approaches.

We summarize our contributions as follows.

- We propose LAVT, a Transformer-based referring image segmentation framework that performs language-aware visual encoding in place of cross-modal fusion post feature extraction.
- We propose video LAVT, the extended version of LAVT for referring video segmentation, which exploits spatio-temporal information for more effective multi-modal information fusion for object segmentation in videos. Based on this, we further propose unified LAVT, a framework with capability of handling both image and video inputs at the same time, with enhanced segmentation capability.
- We show that the proposed "jointly-encode-and-align" methodology is not only beneficial to the image task, but works surprisingly well for videos despite its simplicity. Besides, we demonstrate the "scale-up" capability of this methodology on unified LAVT with unified tasks and scaling data.
- We obtain promisingly competitive results on three benchmark datasets for referring image segmenta-

tion and four benchmark datasets for referring video segmentation, which demonstrates the effectiveness and generality of the proposed methods. Source code is available at LAVT-RS.

It is to be noted that a preliminary conference version of this work was initially presented in [31]. As an extension, we propose the video LAVT framework for referring video segmentation, which enhances the original PWAM for dealing with videos by introducing multi-scale 3D convolutional operators. We further propose the unified LAVT framework for unified segmentation task of RIS and RVOS, which enhances the video backbone for augmenting the local information modeling. The proposed Transformer-based "jointly-encode-and-align" approach is not only a first in the image domain, but a first in the video domain, which we show works very well despite its simplicity. Moreover, we have conducted experiments on two other large-scale datasets for referring video segmentation and demonstrated the effectiveness of video LAVT. In this journal version, we more thoroughly validate our proposed approaches and provide comprehensive analyses on the experimental results, including detailed descriptions for the construction of the 3D PWAM, more comprehensive ablation studies, and additional visualization results.

2 Related Work

2.1 Referring Image Segmentation

Over the past years, referring image segmentation has attracted growing attention in the research community and there are two main processes in conventional pipelines: (1) extracting features from the text and image inputs respectively, and (2) fusing the multi-modal features to predict the segmentation mask. In the first process, previous methods adopt pre-trained language models (*e.g.*, based on recurrent neural networks [2], [7], [8], [32], [33] or Transformers [34], [35]) to encode text inputs, and powerful vision network architectures (*e.g.*, plain fully convolutional networks [2], [7], [36], DeeplabV3 [8], [34], [37], and DarkNet [33], [38], [39]) to encode visual inputs.

The multi-modal feature fusion module which joins the two types of features is the key component that prior arts focus on. For example, Hu *et al.* [2] propose the first baseline based on the concatenation operation, which is improved by Liu *et al.* [7] with a recurrent strategy. Shi *et al.* [9], Chen *et al.* [10], Ye *et al.* [40], and Hu *et al.* [11] model cross-modal relations between language and vision features via various attention mechanisms. Yu *et al.* [41] and Huang *et al.* [14] explore the modular decomposition of language for better capturing different concepts (*e.g.*, categories, attributes, relations, *etc.*) in multi-modal features, while Hui *et al.* [15] exploit syntactic structures for guiding multi-modal context aggregation.

The methods most related to ours are VLT [17] and EFN [42], where the former designs a Transformer decoder for fusing linguistic and visual features, and the latter adopts a convolutional vision backbone network for encoding language information. Different from [17], we propose an early fusion scheme which effectively exploits the Transformer encoder for modeling multi-modal context. Compared to [42], we do not rely on a complicated cross-modal decoder, leading to a clearer and more effective framework. Under fair comparisons, our method outperforms these two previous counterparts by large margins. The extended version of VLT [43] proposes a Spatial-Dynamic Fusion (SDF) module and a masked contrastive learning objective. The former (SDF) is very similar to our proposed PWAM, with the main difference being that after the pixel-word attention step, PWAM combines visual and linguistic features via element-wise multiplication while SDF does it via concatenation; the masked contrastive learning objective is in principle a generic method applicable to many referring segmentation networks: It operates based on the idea of pulling close multi-modal features obtained from different expressions for the same object, while pushing afar multi-modal features obtained from different objects.

Most recently, more methods are developed in the direction of scaling up vision-language segmentation models, such as SADLR [44], GRES [45], CGFormer [46], UNINEXT [47] and PolyFormer [48]. SADLR [44] leverage a continuously updated query as the representation of the target object for progressively learning discriminative multi-modal features in RIS task. GRES [45] propose a new benchmark that extends Referring Expression Segmentation to support expressions referring to any number of target objects, with a novel region-based approach, ReLA. CGFormer [46] propose Group Transformer to achieve object-aware cross-modal reasoning by explicitly grouping visual features into different regions and modeling their dependencies conditioning on linguistic features. The UNINEXT [47] model unifies a wide variety of (*i.e.*, 10) object-centric visual understanding tasks under a prompt-conditional object discovery paradigm, while the PolyFormer [48] model formulates the problems of referring image segmentation and referring expression comprehension as sequence-to-sequence modeling. Their contributions consist in proposing large, unified frameworks or systems that can handle multiple tasks simultaneously, fueled by large-scale paired multi-modal training data, whereas we focus on demonstrating the efficacy of the early fusion of language and vision features in a segmentation-oriented framework, as well as proposing a simple but effective mechanism towards achieving this goal.

2.2 Referring Video Segmentation

Unlike referring image segmentation, referring video segmentation requires segmenting out the target object in each frame of a video where the object is present, and therefore, poses the additional challenge of temporal information modeling to methods.

Inspired by the problem's link to actor-action segmentation in videos [49] (the difference being that the actors and actions are described by language rather than fixed categories), many methods exploit 3D convolutional neural networks (*e.g.*, C3D [50], I3D [51], and P3D [52]) as visual encoders for addressing temporal information modeling [3], [13], [53], [54]. Different from 2D convolutions, 3D convolutions can detect features that span a period of time, which is conducive to action understanding (*e.g.*, running, jumping, kicking, *etc.*) in videos, but may at the same time introduce spatial misalignment into the features, which affects the spatial accuracy of segmentation [18], [55].

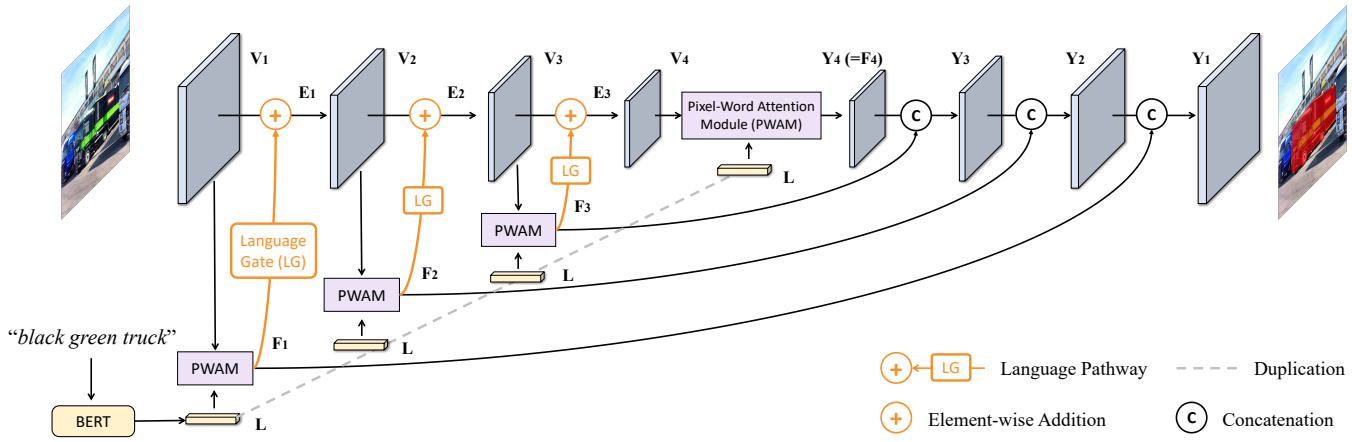

Fig. 2. The overall pipeline of the proposed LAVT. We leverage a hierarchical vision Transformer [25] to perform language-aware visual encoding. At each stage, visual feature maps V_i, $i \in \{1, 2, 3, 4\}$ are encoded from the corresponding stage of Transformer layers (which are described in Sec. 3.1.1 and for diagrammatic clarity, are not illustrated in this figure). Then V_i are used as queries for generating a set of position-specific language feature maps F_i, $i \in \{1, 2, 3, 4\}$ in the pixel-word attention module (Sec. 3.1.2). Next, we adaptively fuse F_i with the original V_i via a language pathway (Sec. 3.1.3). The new visual feature maps E_i, $i \in \{1, 2, 3\}$ are then passed into the next stage of Transformer layers for further processing. A standard segmentation head (Sec. 3.1.4) produces the final segmentation output.

With the purpose of improving the temporal consistency of predictions, another line of work instead approaches temporal modeling from the perspective of correspondence learning, as is frequently done for video object segmentation and tracking. Among methods that follow this approach, Seo *et al.* [12] leverage the space-time attention network [56] to fuse features from history frames with those from the current frame, while Khoreva *et al.* [30] establish temporal tracks of predictions based on certain measures of temporal consistency among those predictions. These methods can typically benefit from multiple rounds of refinement at inference time.

Observing that sometimes modeling temporal information may be at odds with preserving fine spatial details, as is the case with 3D convolutional networks, some methods explore complementary ways for utilizing these two types of signals. For instance, Hui *et al.* [55] encode both 3D and 2D convolutional features in parallel branches, which are joined after having been fused with language features respectively. Ding *et al.* [57] adopt a similar approach, but instead encode 2D convolutional features of the current frame and its difference with a previous frame.

Aside from temporal modeling, another focus in the recent literature is to develop Transformer-based multi-modal feature fusion mechanisms for fusing linguistic and visual features [18], [19], [58], [59], which achieve strong performance. After visual and linguistic feature extraction, the idea is to employ the Transformer encoder-decoder architecture [20] equipped with learnable queries [60] to join these two types of features. One of the key differences among these methods lies in the choices for the query, key, and value in the decoder. Due to space constraints, we refer readers to their papers for details. One potential drawback of this type of approaches lies in the overall complexity of the systems, which may involve exceedingly many components and the complicated process of instance sequence matching.

Recently, there has been a series of work focusing on the implementation of comprehensive feature interaction and alignment in the field of computer vision. SgMg [58] explains how existing R-VOS methods suffer from the feature drift problem and proposes Spectrum-guided Cross-modal Fusion to encourage intra-frame global interactions in the spectral domain. While TempCD [59] proposes a novel collection-distribution mechanism for interacting between the referent token and object queries. DMFormer [61] explicitly couples visual features with different syntactic parts of the text, which would ultimately encourage more explicit and comprehensive feature interactions. HTML [62] can effectively align lingual and visual features to discover core object semantics in the video, by learning multimodal interaction hierarchically from different temporal scales.

Different from the state-of-the-art video-specific/video-capable approaches [18], [19], [43], [48] which mainly rely on additional multi-modal feature fusion components for learning vision-language alignments, our proposed video LAVT essentially converts the vision Transformer backbone network into a multi-modal feature encoder, which jointly processes linguistic and visual features.

2.3 Transformers

The Transformer architecture is first introduced for the task of neural machine translation [20] and has since dominated the natural language processing (NLP) field [35], [63], [64] due to its strong capability of global context modeling. More recently, it has achieved great success on various computer vision tasks, *e.g.*, image classification [25], [65], [66], action recognition [26], [67], object detection [25], [60], [68], and semantic segmentation [25], [69], [70].

There has also been a rich line of work on Transformers in the intersection area of computer vision and NLP [71], [72]. For example, Radford *et al.* devise a large-scale pre-training model, named CLIP [21], which applies contrastive learning [73], [74], [75] on features learned by a vision Transformer and a language Transformer. Hu *et al.* [22] propose a Unified Transformer (UniT) model that jointly learns multiple vision-language tasks across different domains. Besides, growing effort has been devoted to other

tasks such as visual question answering [23] and text-to-video retrieval [24]. However, to the best of our knowledge, there have been very few attempts on designing a unified Transformer model for the task of referring segmentation.

3 METHOD

In this section, we start with the introduction of our language-aware vision Transformer (LAVT) for image segmentation (illustrated in Fig. 2) in Sec. 3.1, and then discuss its 3D counterpart for video segmentation in Sec. 3.2, to which we refer as video LAVT (illustrated in Fig. 5 (a)). The fundamental workings of both models involve a process of language-aware visual encoding, which we introduce in Sec. 3.1.1. It is achieved via a pixel-word attention module and a language pathway, which we detail in Secs. 3.1.2 and 3.1.3, respectively. Then in Sec. 3.1.4, we describe the light-weight mask predictor used to obtain final results. In video LAVT, the vision Transformer layers and the pixel-word attention module are modified to enhance the model's ability to capture spatio-temporal information, and we clarify these changes in Sec. 3.2.

3.1 LAVT for Referring Image Segmentation

3.1.1 Language-aware visual encoding

Given an input pair of an image and a natural language expression that specifies an object from the image, our model outputs a pixel-wise mask that delineates the object. To extract language features, we employ a deep language representation model to embed the input expression into high-dimensional word vectors. We denote the language features as $L \in \mathbb{R}^{C_l \times N}$, where C_l and N denote the number of channels and the number of words, respectively.

After obtaining the language features, we perform joint visual feature encoding and vision-language (which is also called "cross-modal" or "multi-modal" in the following content) feature fusion through a hierarchy of vision Transformer layers organized into four stages. We index each stage using $i \in \{1, 2, 3, 4\}$ in the bottom-up direction. Each stage employs a stack of Transformer encoding layers (with the same output size) ϕ_i, a multi-modal feature fusion module θ_i, and a learnable gating unit ψ_i. Within each stage, language-aware visual features are generated and refined via three steps. First, the Transformer layers ϕ_i take the features from the previous stage as input, and output enriched visual features, denoted as $V_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. Then, the output visual features V_i are combined with language features L via the multi-modal feature fusion module θ_i to produce a set of multi-modal features, denoted as $F_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. Finally, each element in F_i is rescaled by the learnable gating unit ψ_i and then added element-wise to V_i to produce a set of enhanced visual features embedded with linguistic information, which we denote as $E_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. We refer to the computations in this final step as the language pathway. Here, C_i, H_i, and W_i denote the number of channels, the height, and the width of feature maps in the i-th stage, respectively.

The four stages of Transformer encoding layers correspond to the four stages in a Swin Transformer [25], which is an efficient hierarchical vision backbone designed for addressing dense prediction tasks. The multi-modal feature fusion module within each stage is our proposed pixel-word attention module (PWAM), which is designed with the aim to densely align linguistic meanings with visual cues. And the gating unit is what we refer to as the language gate (LG), a special unit that we devise for regulating the flow of linguistic information along the language pathway (LP).

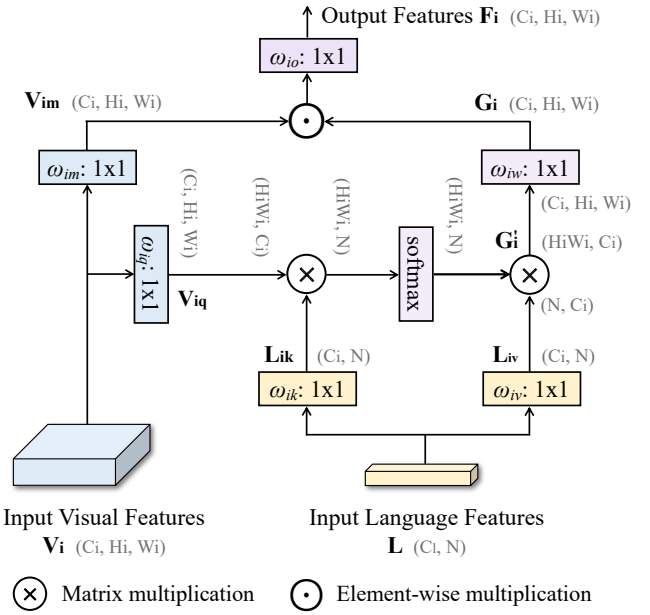

Fig. 3. Pipeline of the pixel-word attention module (PWAM). First, a single-head scaled dot-product attention [20] is performed using the input visual feature maps V_i as queries and the input linguistic feature maps L as keys and values. The result, G_i, is a set of linguistic feature maps of the same spatial size as V_i. G_i is then multiplied element-wise with a projection of the input visual feature maps V_{im}, followed by another projection before final output.

3.1.2 The pixel-word attention module

In order to separate a target object from its background, it is important to align the visual and linguistic representations of the object across modalities. One general approach is to combine the representation of each pixel with the representation of the referring expression, and learn multi-modal representations that are discriminative of a "referent" class and a "background" class. Previous approaches have developed various mechanisms for addressing this challenge, including dynamic convolutions [76], concatenations [2], [8], [76], cross-modal attentions [9], [11], [40], [42], [77], graph neural networks [54], *etc*. Compared to most of the previous cross-modal attention mechanisms [9], [11], [40], [42], [77], our pixel-word attention module (PWAM) produces a much smaller memory footprint as it avoids computing attention weights between two image-sized spatial feature maps, and is also simpler due to fewer attention steps.

Fig. 3 illustrates PWAM schematically. Given the input visual features $V_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ and linguistic features $L \in \mathbb{R}^{C_l \times N}$, PWAM performs multi-modal fusion in two steps, as introduced in the following. First, at each spatial location, PWAM aggregates the linguistic features L across the word dimension to generate a position-specific, sentence-level feature vector, which collects linguistic information

most relevant to the current local neighborhood. This step generates a set of spatial feature maps, $G_i \in \mathbb{R}^{C_i \times H_i \times W_i}$. Concretely, we obtain G_i as follows

$$V_{iq} = \text{flatten}(\omega_{iq}(V_i)), \tag{1}$$
$$L_{ik} = \omega_{ik}(L), \tag{2}$$
$$L_{iv} = \omega_{iv}(L), \tag{3}$$
$$G'_i = \text{softmax}(\frac{V_{iq}^T L_{ik}}{\sqrt{C_i}}) L_{iv}^T, \tag{4}$$
$$G_i = \omega_{iw}(\text{unflatten}(G_i'^T)), \tag{5}$$

where ω_{iq}, ω_{ik}, ω_{iv}, and ω_{iw} are projection functions. Each of the language projections ω_{ik} and ω_{iv} is implemented as a 1×1 convolution with C_i number of output channels. And the query projection ω_{iq} and the final projection ω_{iw} each is implemented as a 1×1 convolution followed by instance normalization, with C_i number of output channels. Here, 'flatten' refers to the operation of unrolling the two spatial dimensions into one dimension in row-major, C-style order, and 'unflatten' refers to the opposite operation. These two operations and transposing are used to transform feature maps into proper shapes for calculation. Eqs. 1 to 5 implement the scaled dot-product attention [20] using visual features V_i as the query and linguistic features L as the key and the value, with instance normalization [78] after linear transformation in the query projection function ω_{iq} and the output projection function ω_{iw}.

Second, after obtaining the linguistic features G_i which have the same shape as V_i, we combine them to produce a set of multi-modal feature maps F_i via element-wise multiplication. Specifically, this step is described as follows

$$V_{im} = \omega_{im}(V_i), \tag{6}$$
$$F_i = \omega_{io}(V_{im} \odot G_i), \tag{7}$$

where $\odot$ denotes element-wise multiplication and ω_{im} and ω_{io} are a visual projection and a final multi-modal projection, respectively. Each of the two functions is implemented as a 1×1 convolution followed by ReLU [79] nonlinearity.

3.1.3 Language pathway

As described earlier, at each stage, we merge the output from PWAM, F_i, with the output from the Transformer layers, V_i. We refer to the computations in this merging operation as the language pathway. In order to prevent F_i from overwhelming the visual signals in V_i and to allow an adaptive amount of linguistic information flowing to the next stage of Transformer layers, we design a language gate which learns a set of element-wise weight maps based on F_i to rescale each element in F_i. The language pathway is schematically illustrated in Fig. 4 and mathematically described as follows

$$S_i = \gamma_i(F_i), \tag{8}$$
$$E_i = S_i \odot F_i + V_i, \tag{9}$$

where $\odot$ indicates element-wise multiplication and γ_i is a two-layer perceptron, with the first layer being a 1×1 convolution followed by ReLU [79] nonlinearity and the second layer being a 1×1 convolution followed by a hyperbolic tangent function. As detailed in the ablation studies in Table 6, we have experimented with and without using a language gate along the language pathway, as well as different final nonlinear activation functions in the language gate, and found that using the gate with $tanh$ final nonlinearity works the best for our model. The summation operation in Eq. 9 is an effective way of utilizing pre-trained vision Transformer layers for multi-modal embedding, as the treatment of multi-modal features as "supplements" (or "residuals") avoids disrupting the initialization weights obtained from pre-training on visual data. We have observed much worse results in the case of adopting replacement or concatenation.

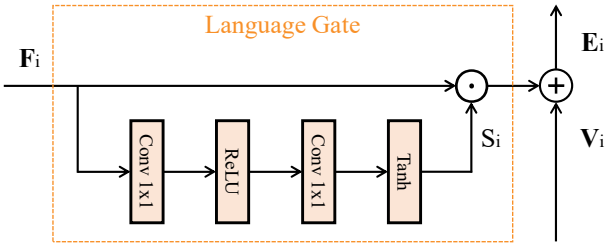

Fig. 4. The schema of the language pathway, which leverages a language gate (LG) for modulating multi-modal information flow. LG is implemented as a two-layer perceptron.

3.1.4 Segmentation

We combine the multi-modal feature maps, F_i, $i \in \{1, 2, 3, 4\}$, in a top-down manner to exploit multi-scale semantics for final segmentation. The decoding process can be described by the following recursive function

$$\begin{cases} Y_4 &= F_4, \\ Y_i &= \rho_i([\upsilon(Y_{i+1}); F_i]), \quad i = 3,\ 2,\ 1. \end{cases} \tag{10}$$

Here '[;]' denotes feature concatenation along the channel dimension, υ represents upsampling via bilinear interpolation, and ρ_i is a projection function implemented as two 3×3 convolutions connected by batch normalization [80] and ReLU [79] nonlinearity. The final feature maps, Y_1, are projected into two class score maps via a 1×1 convolution, representing the "background" class and the "object" class respectively. The average Dice loss [81] on these two classes is used for training the model. More details are in Sec. 3.3

3.2 LAVT for Referring Video Segmentation

3.2.1 Language-aware visual encoding for videos

The overall pipeline of video LAVT is illustrated schematically in Fig. 5 (a). Given a video clip of T frames and the referring expression, our task is to embed language information with video content and predict a mask that delineates the target object in each frame of the input clip. Our language-aware visual encoding scheme described in Sec. 3.1.1 for images is readily applicable for this purpose with some small modifications.

The simplest approach which constitutes a naïve version of video LAVT is to change the Swin Transformer layers [25], denoted as ϕ_i in Sec. 3.1.1, to the Video Swin Transformer layers in [26]. Concretely, the windows in each Transformer layer that introduce locality to attention are changed from 2D spatial windows to 3D spatio-temporal

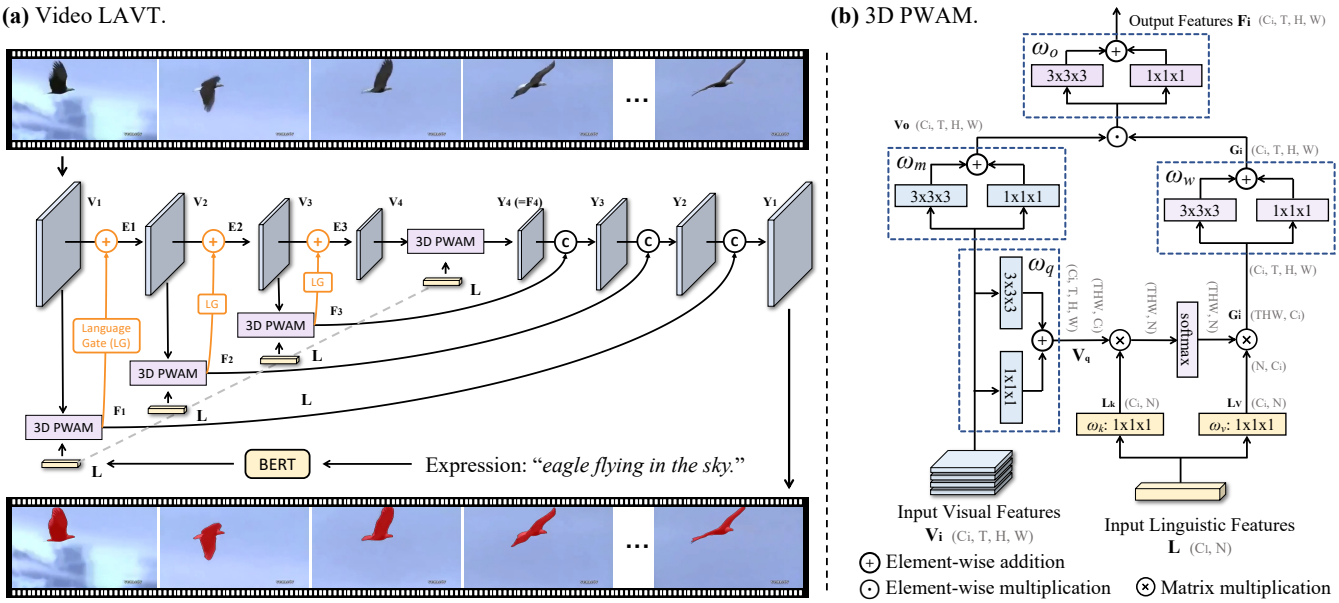

Fig. 5. (a) The pipeline of video LAVT is similar to that of LAVT. V_i, $i \in \{1, 2, 3, 4\}$ represents features encoded by Video Swin Transformer [26] layers, and 3D PWAMs instead of PWAMs are used for producing the multi-modal features, F_i, $i \in \{1, 2, 3, 4\}$, which are sent to the next stage of Transformer layers along with V_i. The language gate and segmentation head remain unchanged, with the input features' temporal dimension rolled into the batch dimension. (b) The pipeline of the 3D PWAM. For each of the spatial projection functions in PWAM, ω_m, ω_q, ω_w, and ω_o, we substitute its 1×1 convolution with a $3 \times 3 \times 3$ convolution and a $1 \times 1 \times 1$ convolution placed side by side, with their outputs joined by addition (illustrated in blue dashed boxes). This allows the modeling of spatio-temporal information when fusing linguistic and visual information. Note that we have omitted the stage index i from the notations of projection functions. More details are in the text below.

cubes. In these new Transformer layers, correspondences are established between nodes residing in not only spatial neighborhoods but also adjacent time steps, which enables temporal information modeling. With this approach, the pixel-word attention module (PWAM) and the language pathway (LP) described earlier in effect do not have to be modified in order to work within the new framework, as 1×1 convolutions are equivalent to linear layers and it would only require proper tensor reshaping operations to accommodate the new temporal dimension in the tensors. For instance, given V_i and F_i in $\mathbb{R}^{C_i \times T \times H_i \times W_i}$, we roll the temporal and spatial dimensions into one dimension and apply linear layers in place of 1×1 convolutions.

Despite that, it makes sense for us to consider extending PWAM with the ability to model spatio-temporal information as it learns to establish multi-modal correspondences and produce integrated features. In the following section, we will introduce our extension of PWAM, where effort is devoted to making the module more apt for processing video inputs and 3D convolutions are exploited to this end.

3.2.2 The 3D pixel-word attention module

Fig. 5 (b) illustrates the extended 3D pixel-word attention module (3D PWAM) schematically, where 3D convolutions with kernel size 3×3×3 are "inserted" next to 1×1×1 convolutions at several places, forming two parallel paths at each place with the outputs joined via element-wise addition. The places are chosen where there was a 1×1 convolution in the original PWAM that transformed *spatial* feature maps, *i.e.*, V_i, G'_i, and $V_{im} \odot G_i$, which correspond to the visual features output by the Transformer layers, the aggregated linguistic features from the attention step, and the multi-modal features after element-wise multiplication, respectively. Correspondingly, the new projection functions for these features, denoted as ω_m, ω_q, ω_w, and ω_o, with stage index omitted, are now each represented by side-by-side 3×3×3 and 1×1×1 convolutions followed by addition, in contrast to the 1×1 convolution in the original PWAM. In this way, multi-scale spatio-temporal information may be extracted and utilized. Nonlinearity and normalization in the functions stay unchanged from those in PWAM.

The 3×3×3 convolutions detect spatio-temporal features from the input, and we may wonder if reinforcing either temporal (*e.g.*, 3×1×1) or spatial (*e.g.*, 1×3×3) information in the input might suffice. In addition, it remains open questions that (1) whether it suffices to use only 3×3×3 convolutions without adding 1×1×1 convolutions to the side, and (2) whether it is actually helpful to model spatio-temporal information in the linguistic features G'_i and multi-modal features $V_{im} \odot G_i$ (while this appears natural for visual features V_i). In Table 1, we list such alternative structures of 3D PWAM that could potentially work; in Table 7, Sec. 4.3.2, we report our empirical findings for them. Across structures 1 to 5, we focus on comparing different types of convolutions, *i.e.*, "element-wise" (an element being a pixel feature vector) convolution, temporal convolution, spatial convolution, spatio-temporal convolution, and sequentially decomposed spatio-temporal convolutions, respectively. In structures 6 to 8, we explore the options of two differently sized convolutions placed side by side with outputs joined by addition or concatenation, the rationale being that the two convolutions may capture complementary information. And in structures 9 to 12, based on our previous discoveries, we verify that whether it is necessary to model spatio-

TABLE 1
Design choices for the 3D PWAM. '$a \times b \times c$' denotes kernel sizes in the time, height, and width directions, respectively. '$f \circ g$' denotes that function f accepts the output of function g as the input. Diagrammatically, this means that f is placed on top of g (see Fig. 6 (a)). We use '$f; g$' to denote the concatenation of the outputs of f and g followed by channel reduction, and '$f + g$' to denote the addition of the outputs of f and g. Asterisk denotes the final design of the 3D PWAM.

	ω_m	ω_q	ω_w	ω_o
1	1×1×1	1×1×1	1×1×1	1×1×1
2	3×1×1	3×1×1	1×1×1	1×1×1
3	1×3×3	1×3×3	1×1×1	1×1×1
4	3×3×3	3×3×3	1×1×1	1×1×1
5	3×1×1 ∘ 1×3×3	3×1×1 ∘ 1×3×3	1×1×1	1×1×1
6	3×1×1 + 1×3×3	3×1×1 + 1×3×3	1×1×1	1×1×1
7	3×3×3 + 1×1×1	3×3×3 + 1×1×1	1×1×1	1×1×1
8	3×3×3 ; 1×1×1	3×3×3 ; 1×1×1	1×1×1	1×1×1
9	3×3×3	3×3×3	3×3×3	3×3×3
10	3×3×3 + 1×1×1	3×3×3 + 1×1×1	3×3×3 + 1×1×1	1×1×1
11	3×3×3 + 1×1×1	3×3×3 + 1×1×1	1×1×1	3×3×3 + 1×1×1
12 (*)	3×3×3 + 1×1×1	3×3×3 + 1×1×1	3×3×3 + 1×1×1	3×3×3 + 1×1×1

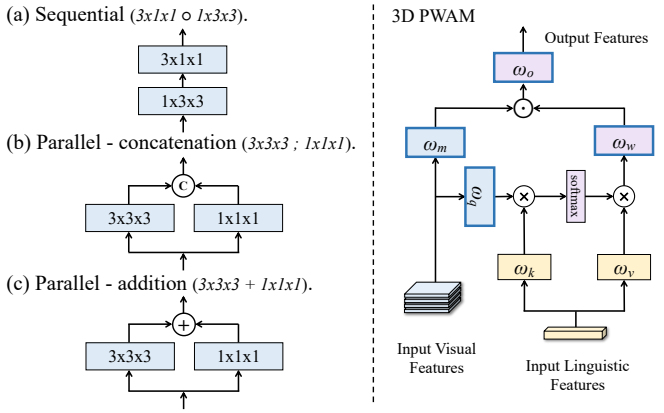

Fig. 6. Schematic illustration for several different constructions (left) of the projection functions in 3D PWAM (right). The constructs (a), (b), and (c) are used in structure 5, structure 8, and structures 7, 10, 11, and 12 in Table 1, respectively. The stage index i is omitted from the notations of projection functions. More discussions about the reasons for their consideration are in the text below.

temporal information in the linguistic features (G_i', using ω_w) and/or multi-modal features ($V_{im} \odot G_i$, using ω_o).

3.2.3 *Unified LAVT for referring segmentation*

The pipeline of unified LAVT is illustrated schematically in Fig. 7. $E_i \in \mathbb{R}^{C_i \times T \times H_i \times W_i}$ represents the output feature of each encode-and-fusion stage, corresponding to E_1, E_2 and E_3 of video LAVT demonstrated in Fig. 5 (a). The Video Swin Transformer Module represents the temporal encoding backbone in our proposed video LAVT. To simultaneously enhance local information modeling, we extract the central frame features $E_{ci} \in \mathbb{R}^{C_i \times 1 \times H_i \times W_i}$ from the visual features E_i for local information enhancement encoding, which runs parallel to the temporal vision encoding. We employ the 2D swin transformer block as the enhanced local modeling module. The encoding stage forms two parallel paths for spatio-temporal information encoding and local information encoding, with the outputs V_{ci+1} and V_{ti+1} joined via element-wise addition. It's worth noting that when conducting inference with the unified LAVT, we employ a single-stream network architecture. Specifically, for the RIS task, we exclusively utilize the 2D Swin Transformer Module as the visual encoding module. Likewise, for the RVOS task, we solely rely on the Video Swin Transformer module. For subsequent cross-modal information fusion, we retain the design of 3D-PWAM and Language Gate, as cross-modal information fusion and modeling are equally important in the encoding process of unified LAVT.

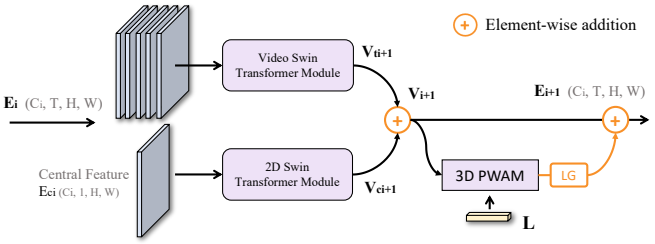

Fig. 7. The illustration of our new proposed unified LAVT's encoding layer. E_i, $i \in \{1, 2, 3, 4\}$, represents the output feature of each encode-and-fusion stage, corresponding to E_1, E_2 and E_3 of video LAVT demonstrated in Fig. 5 (a). It's worth noting that when conducting inference with the unified LAVT, we employ a single-stream network architecture. Specifically, for the RIS task, we exclusively utilize the 2D Swin Transformer Module as the visual encoding module. Likewise, for the RVOS task, we solely rely on the Video Swin Transformer module.

3.3 Implementation

We implement our method in PyTorch [82] and use the BERT implementation from HuggingFace's Transformer library [83]. The Transformer layers in LAVT are initialized with weights pre-trained on ImageNet-22K [84] from the Swin Transformer [25], and those in video LAVT are initialized with weights pre-trained on Kinetics 400 [85] and ImageNet-1K [84] from the Video Swin Transformer [26]. Our language encoder is the base BERT model [86] with 12 layers and hidden size 768 (hence C_t in Sec. 3 is 768) and is initialized using pre-trained weights from HuggingFace. The rest of weights in our model are randomly initialized. The number of channels C_i in Sec. 3 is 512. The main loss used to optimize LAVT, video LAVT, and all ablation/reference models is a multi-class Dice loss. The original Dice loss [81] is a binary segmentation loss computed over a single class score map. As our model outputs two class score maps, one for the "object" class and the other for the "background" class, we compute the average Dice loss for the two classes. To enhance the boundary accuracy of the predictions, when training some of the video LAVT models, we additionally adopt a boundary loss [87], which penalizes misalignment of boundaries.

For training LAVT, we adopt the AdamW [88] optimizer with weight decay 0.01 and initial learning rate 5e-5 with polynomial learning rate decay. We train our model on each dataset for 40 epochs with batch size 32. We iterate through each object (while randomly sampling one referring expression for it) exactly once in an epoch. Images are resized to 480×480 and no data augmentation techniques are applied. During inference, $argmax$ along the channel dimension of the score maps are used as predictions. Besides, for fair comparison with techniques (*i.e.* UNINEXT [47] and PolyFormer [48]) training with the scaled data, we further trained LAVT and these methods with augmented data,

please see more in the supplementary materials.

For training video LAVT, we adopt the same optimizer with weight decay 0.01 and initial learning rate 4e-5 and 6e-5 with polynomial learning rate decay for Refer-YouTube-VOS and A2D-Sentences, respectively. We conduct experiments of video LAVT on both "train-from-scratch" setting and "pretrain-then-finetune" setting following ReferFormer [19]. For training unified LAVT, we also adopt AdamW optimizer with weight decay 0.01 and initial learning rate 4e-5 with polynomial learning rate decay for Refer-YouTube-VOS. In addition to "pretrain-then-finetune" setting, unified LAVT is further trained on "joint-train" setting by training the mixed data from Ref-Youtube-VOS and Ref-COCO for unified task. Please see more in the supplementary materials.

4 Experiments

4.1 Datasets and Metrics

Referring image segmentation. We evaluate our method on three standard benchmark datasets, RefCOCO [27], RefCOCO+ [27], and G-Ref [28], [29]. Images in the three datasets are collected from the MS COCO dataset [4] and annotated with natural language expressions. Each of RefCOCO, RefCOCO+, and G-Ref contains 19,994, 19,992, and 26,711 images, with 50,000, 49,856, and 54,822 annotated objects and 142,209, 141,564, and 104,560 annotated expressions, respectively. Expressions in RefCOCO and RefCOCO+ are very succinct (containing 3.5 words on average). In contrast, expressions in G-Ref are more complex (containing 8.4 words on average), which makes the dataset more challenging. Conversely, RefCOCO and RefCOCO+ tend to have more objects of the same category per image (3.9 on average) compared to G-Ref (1.6 on average), therefore, they can better evaluate an algorithm's ability to comprehend instance-level details. RefCOCO+ bans the use of location words in its expressions, therefore, the model can only make predictions based on appearance information. Additionally, there are two different partitions of the G-Ref dataset, one by UMD [28] and the other by Google [29]. We report results on both. Ambiguities and foul language can occasionally be found in expressions of these datasets, and we hope that there will be collective effort from the community to address these issues in the future.

We adopt the common metrics of overall intersection-over-union (oIoU), mean intersection-over-union (mIoU), and precision at the 0.5, 0.7, and 0.9 threshold values. The overall IoU is measured as the ratio between the total intersection area and the total union area of all test samples, each of which is a language expression and an image. This metric favors large objects. The mean IoU is the IoU between the prediction and ground truth averaged across all test samples. This metric treats large and small objects equally. The precision metric measures the percentage of test samples that pass an IoU threshold.

Referring video segmentation. We conduct experiments on four challenging referring video segmentation benchmarks, Refer-YouTube-VOS [12], Ref-DAVIS17 [30], A2D-Sentences [3] and JHMDB-Sentences [3]. Refer-YouTube-VOS is based on the YouTube-VOS dataset [6], and contains roughly 4K videos, 7K unique objects, and 15K language expressions. There are 3,471 videos for training and 202 videos for validation. We train our model on the training set and evaluate it on the validation set using the official evaluation server. JHMDB-Sentences [3] are created by providing the additional textual annotations on the original A2D [49] and JHMDB [90] datasets. The A2D-Sentences dataset is extended from the Actor-Action Dataset (A2D) [49] with language expressions annotated for the 43 actor-action categories in A2D. It contains 6,656 sentences describing 6,656 unique objects. There are 3,036 videos for training and 746 videos for testing. We train our model on the training set and evaluate it on the test set. Ref-DAVIS17 [30] extends DAVIS17 [91] with language descriptions for specific objects across 90 videos. For A2D-Sentences, we report the aforementioned metrics of precision at five IoU thresholds (from 0.5 to 0.9 with 0.1 increments), overall IoU, and mean IoU. For JHMDB-Sentences, we report overall IoU, and mean IoU. For Refer-YouTube-VOS and Ref-DAVIS17, we report the mean region similarity $\mathcal{J}$, which is the same as the mean IoU defined above, and the mean contour accuracy $\mathcal{F}$, which is the mean F-measure defined over contour points from the predictions and the ground truths.

We want to note that another common metric on the A2D-Sentences dataset, which is the mean Average Precision (mAP) following COCO definition [4], is not particularly suitable for evaluating "semantic segmentation-style" models, which employ a convolutional classifier to produce a single mask for the entire image/frame. The convolutional approach is valid because currently any frame in A2D-Sentences that doesn't have a ground-truth mask is ignored during evaluation (a standard practice established by state-of-the-art methods such as CMPC+ [54], MTTR [18], and ReferFormer [19]), thus, every frame encountered during evaluation contains one and only one target object given the referring expression. For methods which produce multiple mask candidates for an object, mAP is meaningful as it ranks the confidence scores associated with these masks, which are required for the models to produce the final prediction. For semantic segmentation-style methods, however, as they do not predict multiple candidates nor do they have confidence scores for these candidates, it can be unclear or ambiguous to try to adopt the mAP metric. For this reason, we leave this metric out.

4.2 Comparison with Others

Referring image segmentation. In Table 2, we evaluate LAVT against the state-of-the-art referring image segmentation methods on the RefCOCO [27], RefCOCO+ [27], and G-Ref [28], [29] datasets using the overall IoU metric. LAVT outperforms its counterparts on all evaluation subsets of all three datasets. Compared with the respective second-best methods on the validation, testA, and testB subsets of RefCOCO, LAVT obtains higher overall IoU with absolute margins of 7.85%, 7.68%, and 6.25%, respectively. Similarly, LAVT attains noticeable improvements compared with the previous state of the art on RefCOCO+, with large margins of 7.97%, 9.89%, and 6.56% on the validation, testA, and testB subsets, respectively. On the most challenging G-Ref dataset (which contains longer expressions), LAVT

TABLE 2
Comparison with state-of-the-art methods in terms of overall IoU on three benchmark datasets. U: The UMD partition. G: The Google partition. We refer to the language model as neural networks that transform word embeddings before multi-modal feature fusion. Readers can refer to the respective papers for details. * means employing the same augmented data for training. † indicates model is jointly trained with Ref-Youtube-VOS and Ref-COCO dataset. We exclusively utilize Swin-B as the backbone during inference process with the unified LAVT.

Method	Visual Encoder	Language Model	RefCOCO			RefCOCO+			G-Ref		
			val	test A	test B	val	test A	test B	val (U)	test (U)	val (G)
"train with separate data"											
CMPC [14]	ResNet101	LSTM	61.36	64.53	59.64	49.56	53.44	43.23	-	-	49.05
LSCM [15]	ResNet101	LSTM	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
CMPC+ [54]	ResNet101	LSTM	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
MCN [39]	Darknet-53	Bi-GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
EFN [42]	ResNet101	Bi-GRU	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [89]	ResNet101	Self-Att	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [77]	Darknet-53	Bi-GRU	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [33]	Darknet-53	Bi-GRU	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [17]	Darknet-56	Bi-GRU	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	49.76
LTS [33]	Swin-B	BERT	69.51	72.39	65.98	59.17	65.30	52.92	60.03	59.71	57.89
EFN [42]	Swin-B	BERT	71.32	72.47	66.17	59.62	64.87	52.90	60.31	60.76	58.22
VLT [17]	Swin-B	BERT	70.94	73.07	66.01	60.42	64.28	52.47	60.23	61.52	57.52
LAVT [31] + VLT [17]	Swin-B	BERT	71.55	73.82	67.03	61.28	65.87	53.62	61.11	62.37	58.16
LAVT [31] + SADLR [44]	Swin-B	BERT	74.24	76.25	70.06	64.28	69.09	55.19	63.60	63.56	61.16
LAVT [31]	Swin-B	BERT	73.50	75.97	69.33	63.79	69.79	56.49	64.02	64.49	61.31
"train with augmented data"											
PolyFormer [48]	Swin-B	BERT	74.82	76.64	71.06	67.64	72.89	59.33	67.76	69.05	-
PolyFormer* [48]	Swin-B	BERT	72.10	74.48	69.20	65.31	70.58	58.24	65.17	66.82	64.11
UNINEXT* [47]	ConvNeXt-L	BERT	76.35	78.21	73.86	67.24	71.97	59.82	70.25	71.09	67.64
LAVT*	Swin-B	BERT	79.18	80.68	75.35	71.71	75.64	64.25	72.11	74.57	69.76
unified LAVT†	Swin-B	BERT	79.32	81.27	75.98	71.82	75.16	65.34	72.82	73.93	68.43

TABLE 3
Comparison between the proposed LAVT, LTS, VLT, and EFN on the RefCOCO validation set, where all models use the same backbone, language model, and training recipes (*e.g.*, the multi-class Dice loss).

Method	P@0.5	P@0.7	P@0.9	oIoU	mIoU
LTS (Swin-B+BERT) [33]	81.14	69.84	26.25	69.51	70.99
EFN (Swin-B+BERT) [42]	84.78	74.54	28.39	71.32	73.71
VLT (Swin-B+BERT) [17]	84.34	73.85	25.04	70.94	72.94
LAVT + VLT [17]	84.89	74.28	23.67	71.55	73.08
LAVT	**85.87**	**76.64**	**35.30**	**73.50**	**75.41**

surpasses the respective second-best methods on the validation and test subsets from the UMD partition by absolute margins of 9.62% and 7.84%, respectively. In a similar way, on the validation set from the Google partition, LAVT outperforms the second-best EFN [42] by an absolute 9.38%.

We want to note that the multi-class Dice loss adopted in this work is more effective for training LAVT than the cross-entropy loss adopted in the conference version of this work [31]. While on most datasets, the new loss brings relatively small improvements, on the G-Ref UMD partition, it improves the overall IoU by more than 2 absolute points. Conversely, the effects of the new loss is less obvious on the reference methods presented in Table 3. This highlights the efficacy of our loss as well as the potential of our model.

The reference methods in Table 2 adopt different visual backbones, language encoders, and training recipes. To make fair comparisons and verify the effectiveness of our approach, in Table 3, we report results of the proposed LAVT and three other state-of-the-art methods, LTS [33], VLT [17], and EFN [42], obtained by adopting $\text{BERT}_{\text{BASE}}$ as the language encoder and Swin-B as the vision backbone network and following the same training setting (described in Sec. 3.3) for all models. While LTS employs a "locate-then-segment" pipeline, VLT is representative of methods that employ a cross-modal Transformer decoder. Conversely, EFN is representative of methods that fuse cross-modal information via an encoder network and additionally rely on a complicated decoder for obtaining the best results. As shown in Table 2 and Table 3 , our method outperforms LTS, VLT, and EFN on RefCOCO/g/+ datasets in all metrics. To further verify that our proposed LAVT encoding scheme is more effective than its counterpart cross-modal decoder approach, we combine our approach with VLT by substituting our original light-weight mask predictor with the cross-modal Transformer decoder from VLT. As shown in this experiment (indicated by "LAVT + VLT" in Table 2 and Table 3), employing a Transformer decoder to perform additional cross-modal feature fusion after language-aware visual encoding by LAVT does not bring gains.

For fair comparison with PolyFormer and UNINEXT, we report results of LAVT and these two methods using identical data and training setting, while PolyFormer and UNINEXT respectively apply stronger visual backbones (Swin-L and ConvNeXt-L) than LAVT (Swin-B) due to their architectural design. Results reported in the "train with augmented data" of Table 2 demonstrate the potential of "scaling up" LAVT by adding more data. Furthermore, the unified LAVT trained under the "jointly-train" setting with mixing data continue to demonstrate robust performance in there dataset.

Referring video segmentation. In Table 4, we evaluate video LAVT against the state-of-the-art referring video segmentation methods on the Refer-YouTube-VOS and A2D-Sentences datasets. In these experiments, we adopt different types of Transformer layers in our model, namely, those from the tiny, small, and base versions of Video Swin. There are two training settings for this task. The first one is what we call "train-from-scratch," which means that the models are trained on the training set of each evaluation

TABLE 4
Comparison with state-of-the-art methods on the Refer-YouTube-VOS and A2D-Sentences datasets under the "train-from-scratch" and "pretrain-then-finetune" training settings with different encoder networks employed. We refer to the language model as neural networks that transform word embeddings before multi-modal feature fusion. Readers can refer to the respective papers for details (*e.g.*, the word embeddings). Experiments for LAVT + SgMg / TempCD adopt the encoding strategy of LAVT, maintaining BERT as the Language Model.

Method	Visual Encoder	Language Model	Refer-YouTube-VOS			A2D-Sentences						
			$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	oIoU	mIoU
"train-from-scratch"												
Gavrilyuk *et al*. [3]	I3D	1D Conv.	-	-	-	47.5	34.7	21.1	8.0	0.2	53.6	42.1
Wang *et al*. [13]	I3D	1D Conv.	-	-	-	55.7	45.9	31.9	16.0	2.0	60.1	49.0
CMSA+CFSA [92]	DeepLab-ResNet-101	None	-	-	-	48.7	43.1	35.8	23.1	5.2	61.8	43.2
URVOS [12]	ResNet-50	Linear	47.23	45.27	49.19	-	-	-	-	-	-	-
CMPC+ [54]	I3D	ConvLSTM	47.48	45.64	49.32	65.5	59.2	50.6	34.2	9.8	65.3	57.3
CSTM [55]	Inception-v3 + I3D	GRU $\times$ 2	-	-	-	65.4	58.9	49.7	33.3	9.1	66.2	56.1
LBDT-4 [57]	ResNet-50 $\times$ 2	LSTM	49.38	48.18	50.57	73.0	67.4	59.0	42.1	13.2	70.4	62.1
MTTR [18]	Video Swin-T	GPT-2	55.32	54.00	56.64	75.4	71.2	63.8	48.5	16.9	72.0	64.0
ReferFormer [19]	Video Swin-T	RoBERTa	56.00	54.80	57.30	76.0	72.2	65.4	49.8	17.9	72.3	64.1
"naïve" vid. LAVT	Video Swin-T	BERT	55.93	54.31	57.55	72.6	67.0	58.9	43.6	15.1	72.2	63.1
video LAVT	Video Swin-T	BERT	57.04	55.39	58.69	77.3	73.2	65.0	49.0	17.3	74.4	65.9
video LAVT	Video Swin-S	BERT	58.79	57.10	60.49	78.6	75.4	67.6	52.2	20.3	75.5	67.7
video LAVT	Video Swin-B	BERT	60.45	58.49	62.42	80.0	76.2	69.1	53.9	21.0	77.0	68.7
"pretrain-then-finetune"												
ReferFormer [19]	Video Swin-T	RoBERTa	59.40	58.00	60.90	82.8	79.2	72.3	55.3	19.3	77.6	69.6
video LAVT	Video Swin-T	BERT	60.91	59.37	62.45	82.8	79.3	71.5	54.6	19.5	77.9	70.0
ReferFormer [19]	Video Swin-S	RoBERTa	60.10	58.60	61.60	82.6	79.4	73.1	57.4	21.1	77.7	69.8
video LAVT	Video Swin-S	BERT	62.96	60.35	65.56	82.9	79.6	73.1	57.2	21.2	79.1	70.4
ReferFormer [19]	Video Swin-B	RoBERTa	62.90	61.30	64.60	83.1	80.4	74.1	57.9	21.2	78.6	70.3
VLT [43]	Video Swin-B	BERT	63.80	61.90	65.60	-	-	-	-	-	-	-
SgMg [58]	Video Swin-B	RoBERTa	65.70	63.90	67.40	-	-	-	-	-	79.9	72.0
TempCD [59]	Video Swin-B	RoBERTa	65.80	63.60	68.00	-	-	-	-	-	-	-
video LAVT	Video Swin-B	BERT	64.90	62.22	67.58	84.6	81.1	74.7	58.1	22.3	80.7	71.9
unified LAVT	Video Swin-B	BERT	65.77	63.61	67.93	85.7	82.9	75.8	59.2	23.5	81.4	72.4
LAVT + SgMg [58]	Video Swin-B	BERT	66.73	64.82	68.63	85.9	82.7	76.1	60.6	24.9	81.8	72.6
LAVT (bi-3D-PWAM)+ SgMg [58]	Video Swin-B	BERT	67.14	65.17	69.09	86.0	82.8	76.2	60.8	25.2	82.0	73.0
LAVT + TempCD [59]	Video Swin-B	BERT	66.80	64.73	68.87	86.1	82.6	76.3	60.9	25.0	82.0	72.5
LAVT (bi-3D-PWAM)+ TempCD [59]	Video Swin-B	BERT	67.19	65.14	69.23	86.3	82.8	76.7	61.3	25.4	82.3	72.9

dataset with initialization weights from the Video Swin Transformer [26] without employing additional training data. This is the main setting adopted in virtually all papers prior to the ReferFormer paper [19]. The second is what we call the "pretrain-then-finetune" setting. This is a more compute-intensive setting that ReferFormer mainly refers to. In this setting, the model is first pretrained on the concatenated training sets of RefCOCO, RefCOCO+, and G-Ref (the UMD partition), by setting $T = 1$, and then respectively finetuned on the Refer-YouTube-VOS training set and the A2D-Sentences training set. This setting requires several weeks of training with a minimum of eight V100 cards for each model.

Under the "train-from-scratch" setting, when all models employ Video Swin-T, on Refer-YouTube-VOS, the proposed video LAVT outperforms ReferFormer by margins (absolute) of 1.04%, 0.59%, and 1.39%, and MTTR by margins (absolute) of 1.72%, 1.39%, and 2.05%, in terms of $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$, respectively; similarly, on A2D-Sentences, video LAVT leads ReferFormer by absolute 2.1% and 1.8%, and MTTR by absolute 2.4% and 1.9%, in overall IoU and mean IoU, respectively. In terms of the P@K metrics (where K is a threshold), our method also compares favorably against ReferFormer and MTTR. Compared to these two models, video LAVT enjoys a simpler architecture with many fewer components and a simpler training protocol without the need of instance sequence matching, which could all make it a more adaptation-friendly model. Adopting more powerful Video Swin layers leads to better results, with video LAVT based on Video Swin-B achieving the best results.

The '"naïve" vid. LAVT' entry in Table 4 refers to the implementation where we only switch from 2D to 3D Transformer layers but maintain the original 2D PWAMs. This is structure 1 in Tables 1 and 7. On Refer-YouTube-VOS, with 55.93 $\mathcal{J}\&\mathcal{F}$, 54.31 $\mathcal{J}$, and 57.55 $\mathcal{F}$, "naïve" video LAVT works surprisingly well—it outperforms MTTR and performs on a par with ReferFormer. On A2D-Sentences, where action information is key to segmentation, "naïve" video LAVT (PWAMs inside which do not model spatio-temporal information) does not do as well by comparison. But as 3D PWAMs replace 2D PWAMs, the efficacy of video LAVT is clearly demonstrated again. Overall, these results show that our proposed language-aware visual encoding scheme is reasonably general for both images and videos, and that the proposed 3D PWAM is an effective way to gather helpful spatio-temporal context for multi-modal feature fusion.

Under the "pretrain-then-finetune" setting, the proposed video LAVT and unified LAVT outperform ReferFormer on all backbones in all metrics for Refer-YouTube-VOS, while also compares favorably with respect to ReferFormer on A2D-Sentences. Specifically, on Refer-YouTube-VOS, video LAVT surpasses ReferFormer by absolute points of 1.51, 2.86, and 2.00 in terms of $\mathcal{J}\&\mathcal{F}$ when using Video Swin-T, Video Swin-S, and Video Swin-B as the visual backbone, respectively. Unified LAVT outperforms ReferFormer by absolute points of 2.87, 2.31, and 3.33 in terms of $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$ and $\mathcal{F}$ when using Video Swin-B as the visual backbone. On A2D-Sentences, when comparing to ReferFormer, video LAVT gains an edge most of the time: Looking at the adopted backbone networks, from Video Swin-T, to Video Swin-S, and then to Video Swin-B, the corresponding video LAVT

model produces better results according to an increasing amount of metrics, from 4 metrics, to 5 metrics, and then to 7 (all) metrics. This pattern also suggests that our method yields good learning capacity—as model complexity grows, its performance keeps getting better. Compared to video LAVT, unified LAVT achieves further improvements in all evaluated metrics on the A2D-Sentences dataset. It may be worth mentioning that in contrast to ReferFormer, which adopts a full-fledged semi-supervised video object segmentation algorithm (*i.e.*, CFBI [93]) for mask refinement in a post-processing step, our results are directly obtained from our model without any post-processing.

Besides, under the "pretrain-then-finetune" setting, we apply our "jointly-encode-and-align" strategy in recent innovative "encode-then-align" works to further demonstrate the effectiveness of video LAVT. We substituted the multi-modal information encoding phase (*i.e.*, the visual encoder and text encoder) of the aforementioned methods with the cross-modal information fusion strategy inherent to LAVT. Specifically, as "LAVT + SgMg / TempCD" delineated in Table 4, we observed an improvement of 1.00, 1.03, and 0.87 absolute points on $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$ respectively for TempCD. Additionally, we saw an enhancement of 1.03, 0.92, and 1.23 absolute points on $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$ respectively for SgMg on Ref-Youtube-VOS dataset. On A2D-Sentences, we also observed a similar trend of performance enhancement. Upon integrating the "jointly-encode-and-align" strategy from LAVT, we noted that both TempCD and SgMg exhibited enhanced performance.

Furthermore, we find that akin to the approach updating visual information in LAVT, the iterative update of textual information at each stage of hierarchical visual information encoding is of equivalent significance. We have engineered a bi-3D-PWAM for this language-as-query approach, which employs a bidirectional 3D-PWAM to concurrently update textual information. This involves the incorporation of an attention module analogous to the 3D-PWAM, with visual and textual features alternating as inputs. The experimental outcomes presented in Table 4 of "bi-3D-PWAM" demonstrate that the strategy of updating text can contribute to further boosting the model's efficacy on both Ref-Youtube-VOS and A2D-Sentences dataset. Due to space constraints, we have further discussed the details of bi-3D-PWAM, the results of Ref-DAVIS17 and JHMDB-Sentences, the results of unified LAVT under the "jointly-train" setting and the comparison with recent methods in the supplementary section. Moreover, we have done discussion, direct comparison and conduct experiments applying LAVT's "jointly-encode-and-align" strategy in recent innovative works [46], [58], [59], [62] to demonstrate the effectiveness of (video) LAVT and make our work up-to-date. Please see more in the supplementary materials.

4.3 Ablation Study

4.3.1 Components of LAVT

We conduct several ablations to evaluate the effectiveness of the key components and design choices in LAVT. All experiments use the same visual backbone network, language model, and training recipes as described in Sec. 3.3.

TABLE 5
Main ablation results for LAVT on the RefCOCO validation set.

LP	PWAM	P@0.5	P@0.7	P@0.9	oIoU	mIoU
✓	✓	**85.87**	**76.64**	**35.30**	**73.50**	**75.41**
	✓	83.38	73.08	33.42	72.01	73.76
✓		82.91	73.80	33.96	71.75	73.44
		81.49	71.02	32.79	70.73	71.94

TABLE 6
Ablation studies for LAVT on the RefCOCO validation set. (G) indicates that LG is adopted in the language pathway and (NG) indicates the opposite. Rows with (*) indicate default choices.

	P@0.5	P@0.7	P@0.9	oIoU	mIoU
(a) activation function in the language gate (LG)					
Tanh (*)	**85.87**	**76.64**	35.30	**73.50**	**75.41**
Sigmoid	85.20	75.80	**35.35**	72.96	75.24
(b) normalization layer in the pixel-word attention module (PWAM)					
InstanceNorm (*)	**85.87**	**76.64**	**35.30**	**73.50**	**75.41**
LayerNorm	84.93	75.48	35.25	72.91	74.73
BatchNorm	83.84	74.76	33.86	72.35	74.10
None	83.85	74.16	34.01	72.07	73.82
(c) features used to produce the final segmentation					
F_4, F_3, F_2, F_1 (G*)	**85.87**	**76.64**	35.30	**73.50**	**75.41**
F_4, F_3, F_2, F_1 (NG)	85.40	76.53	**35.61**	72.75	75.22
E_4, E_3, E_2, E_1 (G)	85.07	76.08	34.97	72.88	75.09
E_4, E_3, E_2, E_1 (NG)	84.87	76.05	34.72	72.92	74.99
V_4, V_3, V_2 (G)	84.25	75.28	33.15	71.88	74.26
V_4, V_3, V_2 (NG)	84.58	76.12	33.13	72.45	74.49
(d) multi-modal attention module					
PWAM (*)	**85.87**	**76.64**	**35.30**	**73.50**	**75.41**
BCAM [11]	82.29	73.03	33.22	69.75	72.55
GA (GARAN) [39], [77]	85.20	75.80	34.29	72.36	74.75

Language pathway (LP). Table 5 shows that removing LP (which corresponds to, mathematically, the removal of Eqs. 8 and 9, or schematically, the removal of the orange stream in Fig. 2) leads to a drop of 1.49 and 1.65 absolute points in overall IoU and mean IoU, respectively. In addition, precision drops by 2 to 3 points across all three thresholds. These results demonstrate the benefit of exploiting our vision Transformer encoder network for jointly embedding linguistic and visual features.

Pixel-word attention module (PWAM). In this ablation study, we replace the spatial language feature maps G_i in PWAM with a sentence feature vector globally pooled from all words [94]. As shown in Table 5, this ablation leads to a drop of 1.75 and 1.97 absolute points in overall IoU and mean IoU, respectively, and a drop of 2 to 3 absolute points in precision across the three thresholds. These results illustrate the effectiveness of densely aggregating linguistic context via our proposed attention mechanism for enhancing cross-modal alignments.

Activation function in the language gate (LG). Our proposed LG learns a set of spatial weight maps, which give our network the flexibility to control the flow of language information in the language pathway. In Table 6 (a), we compare the sigmoid function and the hyperbolic tangent function as the final activation function in LG. Using the hyperbolic tangent function generally leads to better results.

Normalization layer in PWAM. As described in Sec. 3.1.2,

TABLE 7
Investigation of potential structures for 3D PWAM as detailed in Table 1 and Sec. 3.2.2, using Refer-YouTube-VOS validation set. Red indicates the overall best results and blue indicates the best results among counterpart structures.

structure	$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)
1	55.93	54.31	57.55
2	43.52	44.47	42.57
3	54.69	53.00	56.39
4	56.49	54.95	58.04
5	41.97	42.54	41.41
6	51.92	50.93	52.90
7	56.34	54.77	57.90
8	55.31	53.61	57.02
9	56.56	55.01	58.10
10	56.52	54.91	58.14
11	56.54	54.93	58.16
12 (*)	57.04	55.39	58.69

we adopt a final instance normalization layer in the projection functions ω_{iq} and ω_{iw} in PWAM. As we illustrate in Table 6 (b), this particular choice of normalization function has a non-trivial effect. In addition to instance normalization (our default choice), we experiment with layer normalization, batch normalization, and not having a normalization layer in the functions ω_{iq} and ω_{iw}. All three other choices produce inferior results. Among these three choices, layer normalization works better than batch normalization, which works better than not using a normalization layer.

Features used for prediction. As shown in Fig. 4, the language-aware visual encoding process of LAVT produces three kinds of spatial feature maps which encapsulate visual and linguistic information, *i.e.*, the outputs from PWAMs ($F_i, i \in \{1, 2, 3, 4\}$), the outputs from the Transformer layers (V_i, $i \in \{2, 3, 4\}$), and the inputs to the following Transformer layers ($E_i, i \in \{1, 2, 3\}$). While our default choice is to use F_i for predicting the object mask, we also consider the other two types of feature maps natural candidates for this purpose. As shown in Fig. 2, E_4 is not generated in the standard architecture of LAVT. To have a convincing ablation study, we compute E_4 with an additional language pathway as defined in Eqs. 8 and 9. Conversely, since V_1 contains pure visual information, while V_2, V_3, and V_4 contain multi-modal information, we exclude V_1 from the experiments. In Table 6 (c), we report segmentation results obtained using F_i, E_i, and V_i with and without language gate (indicated by 'G' and 'NG,' respectively). Table 6 (c) shows that using our default choice of F_i with language gate produces the best overall results among all choices. Also, we observe that while the language gate tends to be useful for F_i and E_i, it slightly degrades results for V_i.

Multi-modal attention module. In Table 6 (d), we compare PWAM with two state-of-the-art attention modules by directly replacing PWAM with them in our framework. Compared to both the grouped attention (GA or GARAN) [39], [77] and the bi-directional cross-modal attention module (BCAM) [11], PWAM achieves higher scores on all metrics. BCAM is representative of the computationally-heavy attention modules, while GA is a recent, top-performing module.

TABLE 8
Comparison between "PWAM" and "3D PWAM" on the Refer-YouTube-VOS dataset under the "train-from-scratch" setting and the *image-based* visual backbone, Swin-B.

Attention Module	$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)	$\triangle_{\mathcal{J}\&\mathcal{F}}$ (%)
PWAM	58.95	56.98	60.91	+0.00
3D PWAM	**60.10**	**57.97**	**62.23**	+1.15

4.3.2 The design of the 3D PWAM

Design choices. As described in Section 3.2.2, we evaluate many different ways to construct the 3D PWAM. Please refer to the text below, Table 1, and Fig. 6 for the details. We report the empirical results in Table 7. First, in structures 1 to 8, we consider variants of the ω_m and ω_q projection functions, because extracting spatio-temporal information from visual features directly is the most intuitive option. Then in structures 9 to 12, we consider the best variants found previously and apply them to the projection functions ω_w and ω_o, which extract information from the linguistic features and multi-modal features, respectively.

In structures 1 to 5, "element-wise" (an element being a pixel feature vector) convolution, temporal convolution, spatial convolution, spatio-temporal convolution, and sequentially decomposed spatio-temporal convolutions are explored, respectively. Results show that structure 4 (with a 3×3×3 convolution) works the best among the five structures. In structures 6 to 8, we investigate whether two parallel convolutions complement each other and work better than a single spatio-temporal convolution. Specifically, structures 6 to 8 correspond to the '$3 \times 1 \times 1 + 1 \times 3 \times 3$', '$3 \times 3 \times 3 + 1 \times 1 \times 1$', and '$3 \times 3 \times 3$; $1 \times 1 \times 1$' variants in Table 1, respectively, and results show that structure 7 with parallel 3×3×3 and 1×1×1 convolutions followed by element-wise addition is the best option.

So far we have found that a single 3×3×3 convolution (structure 4) and parallel convolutions joined by addition (structure 7) are among the best implementations for projection functions ω_m and ω_q. Based on these results, we then let the projection functions ω_w and ω_o adopt these constructions, leading to structures 9 and 12, respectively, and results show that structure 12 is the better option. Finally, in structures 10 and 11, we implement '$3 \times 3 \times 3 + 1 \times 1 \times 1$' for either ω_m or ω_q. Neither structure performs better than structure 12, which verifies that it is helpful to implement '$3 \times 3 \times 3 + 1 \times 1 \times 1$' for all four projection functions and this leads to our final design of the 3D PWAM.

Effectiveness of the 3D PWAM. In Table 8, to better understand the temporal modeling capabilities of the 3D PWAM, we conduct a comparison with the 2D PWAM using a robust image-based visual backbone, Swin-B. In this framework, the 3D PWAM is the sole unit capable of temporal modeling. We see that 3D PWAM leads to 1.15, 0.99, and 1.32 absolute points of improvement over the static model in terms of $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$, respectively. This result demonstrates that the proposed 3D PWAM is effective at capturing helpful temporal cues that can lead to more accurate segmentation in videos.

Comparison with the CM-FPN. The ReferFormer's CM-

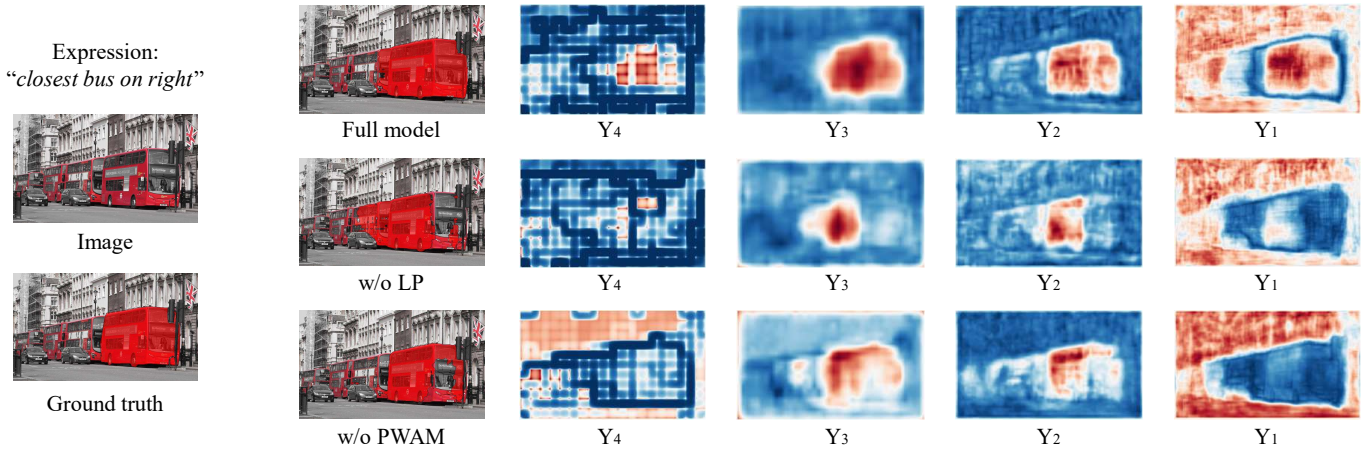

Fig. 8. Visualizations of predictions and feature maps on an example from the RefCOCO validation set. The left-most column illustrates the input expression, the input image, and the ground-truth mask overlaid on the input image, from top to bottom. In each row on the right, we visualize the predicted mask and the feature maps used for final classification (*i.e.*, Y_4, Y_3, Y_2, and Y_1) from left to right. LP represents the language pathway and PWAM represents the pixel-word attention module.

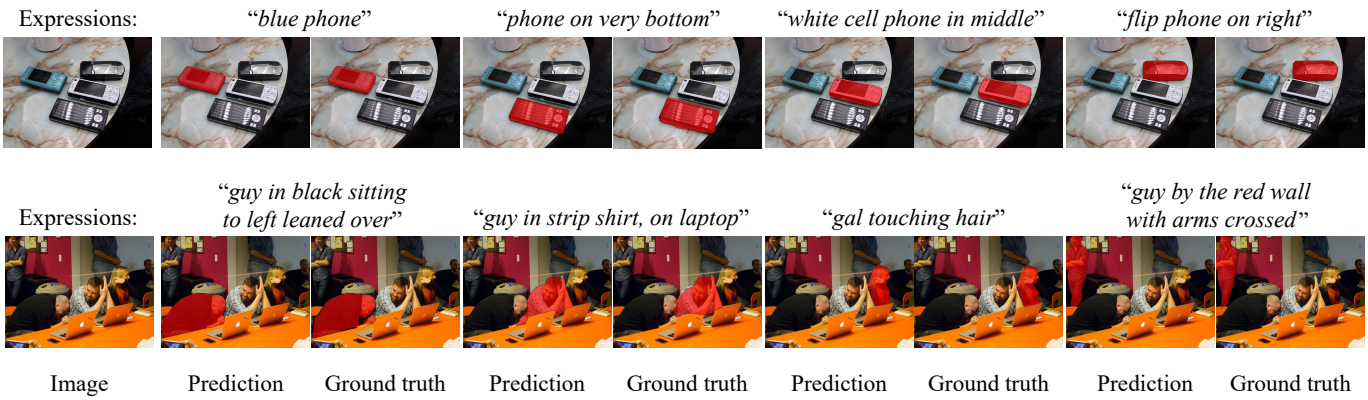

Fig. 9. Visualizations of predicted masks and the ground-truth masks on two examples from the RefCOCO validation set.

TABLE 9
Comparison between the vision-language fusion (VLF) module of CM-FPN [19] and our 3D PWAM in controlled experiments. Results are reported for the Refer-YouTube-VOS dataset under the "train-from-scratch" setting.

Method	Backbone	$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)
video LAVT w/ VLF [19]	Video Swin-T	54.27	52.63	55.91
video LAVT	Video Swin-T	**57.04**	**55.39**	**58.69**
video LAVT w/ VLF	Video Swin-S	54.95	52.97	56.92
video LAVT	Video Swin-S	**58.79**	**57.10**	**60.49**
video LAVT w/ VLF	Video Swin-B	58.07	55.90	60.25
video LAVT	Video Swin-B	**60.45**	**58.49**	**62.42**

FPN [19] is a multi-level cross-modal fusion strategy developed in the context of referring video object segmentation. At its core is a "vision-language fusion" module composed of a self-attention block followed by a cross-attention block, inserted into each level of an FPN [95]. The most sensible way to compare 3D PWAM to CM-FPN is by replacing our 3D PWAM with the "vision-language fusion" module in our framework, while at the same time maintaining the same backbones, language model, and training recipes.

Table 9 shows that the 3D PWAM outperforms the CM-FPN's vision-language fusion (VLF) module in terms of $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$ metrics across Video Swin-T, Video Swin-S, and Video Swin-B backbone networks. Particularly with Video Swin-S and Video Swin-B, the 3D PWAM's advantage is significant, showing a 3 to 4 point increase across metrics. This advantage likely stems from the 3D PWAM's unique feature processing, where language features with spatial dimensions from vision-to-language attention undergo element-wise multiplication with visual features, leading to multi-modal fusion. In contrast, the CM-FPN's VLF module lacks this multiplication step, outputting the results of vision-to-language attention without further fusion. The 3D PWAM's approach is integral to the LAVT framework, as segmentation maps are derived directly from its output, where convolutions assess the match between language and vision information at each spatial location. In ReferFormer's CM-FPN, there is no need for such direct fusion, as cross-modal information is supplemented at a later stage from the Transformer encoder-decoder output.

4.4 Visualizations

In Fig. 8, we visualize the predictions and feature maps of our full LAVT model and two ablated models (without

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3468640

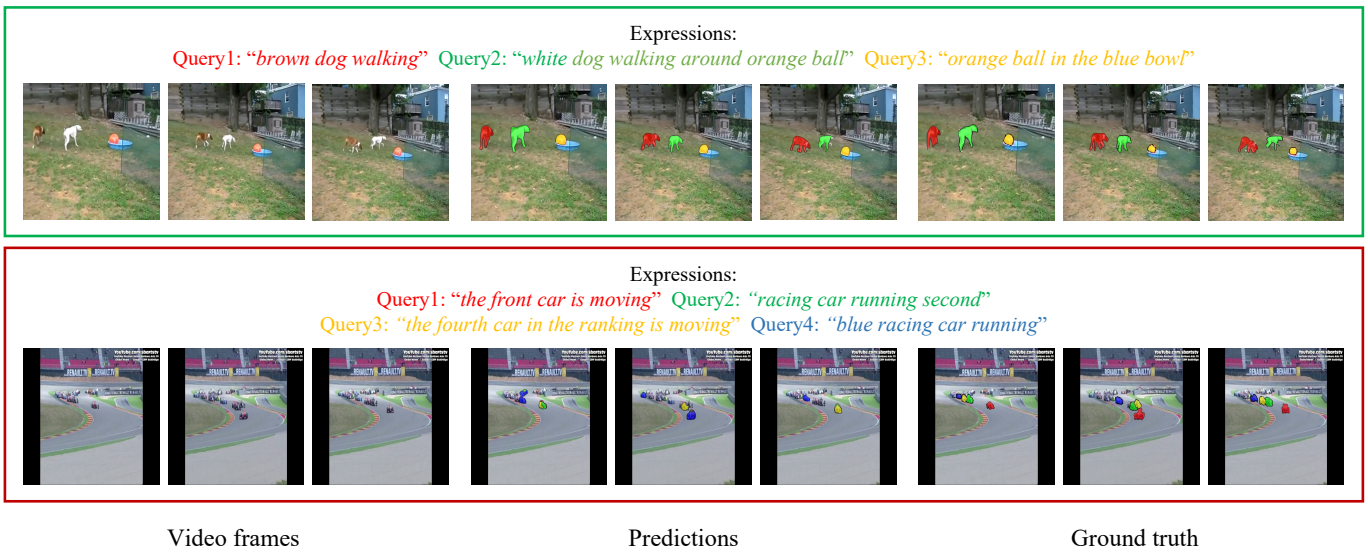

Fig. 10. Visualizations of predicted masks and the ground-truth masks on examples from the A2D-Sentences validation set. The example enclosed with green lines is a success case, and that enclosed with red lines is a failure case. The first example is quite challenging, which includes multiple small targets, and our model segments them out quite accurately. In the second example, the model seems confused by the many different race cars, which are small and close to each other, with indistinct colors. Zoom in to have a better view.

the language pathway ('w/o LP') and without the pixel-word attention module ('w/o PWAM')). From the first row, we can observe that the lower-resolution feature maps (*i.e.*, Y_4, Y_3, Y_2) in our full model can accurately locate the high-level concept described by the text, while the high-resolution feature maps (*i.e.*, Y_1) contain boundary cues that help with accurate segmentation. Comparing the predicted masks of the three models, we can observe that the removal of LP and the removal of PWAM both lead to false negative predictions on the windshield area of the target bus, while the removal of LP additionally results in the false positive identification of the middle bus. These qualitative results further validate the effectiveness of our proposed LP and PWAM mechanisms. More visualization examples of LAVT and video LAVT are shown in Fig. 9, Fig. 10, and the supplementary material.

5 Conclusion and Future Work

In this paper, we have proposed the Language-Aware Vision Transformer (LAVT), a general framework for addressing referring segmentation (*i.e.*, referring image/video segmentation). Unlike previous methods, LAVT leverages the intermediate layers of the visual Transformer encoder network to jointly embed linguistic and visual features, thereby relocating the key process of cross-modal feature fusion to the stage of image/video encoding. This is achieved by "injecting" linguistic cues into the vision Transformer network at every stage, where the cues are aligned with visual features by a pixel-word attention module and modulated by a gating function. In particular, we have designed a 3D version of the pixel-word attention module for processing video inputs, which leverages multi-scale 3D convolutions to effectively model spatio-temporal information. Extensive ablation studies validate our design choices, and experimental results on five benchmarks demonstrate the advantage of our method with respect to the state-of-the-art approaches.

Since LAVT is designed to be a simple baseline for the referring segmentation task, there can potentially be many interesting extensions from the current model. We briefly outline three in the following. First, beyond the currently proposed language-aware visual encoding scheme based on a vision Transformer, the encoding scheme that works in the opposite direction—vision-aware language encoding based on a language Transformer—may also produce strong features conducive to segmentation. This leads to a potential bi-directional fusion strategy that may yield better results. Second, from an architectural point of view, currently the (3D) PWAM at each stage amounts to a cross-attention layer (followed by an element-wise multiplication operation) inserted at the end of the corresponding stage of the (Video) Swin Transformer [25], [26], in effect converting the original vision Transformer into a multi-modal Transformer. This leads to the question that whether we can scale up our method by making *native* changes to the (Video) Swin Transformer, where cross-modal attention and element-wise multiplication can be implemented based on the original non-overlapping window and shifted window attention layers in the (Video) Swin Transformer. This fully integrated approach, if successful, has the potential to serve as a much more general architecture suitable for many vision-language tasks, including pre-training tasks. Third, the additional Transformer encoder-decoder component exploited for multi-modal feature fusion in MTTR [18] and ReferFormer [19] is a promising way of aligning multi-modal features, and it may be added to our existing framework to potentially produce stronger results.

Looking beyond the architectural perspective, there might be many other directions for improving the state-of-the-art referring segmentation models. As much of the interest in referring segmentation stems from a desire to develop an algorithm that can segment an image or video in an open world (*i.e.*, segment out *any* object), it makes

sense to explore the use of pre-trained large vision-language models (*e.g.*, CLIP [21] and Stable Diffusion [96]) to improve generalization. Some direct benefits of doing so can be derived from the following two aspects. First, the enlarged vocabulary of the language encoder may vastly expand the number of concepts understood by the model. Second, alignment information in the pre-trained model should be useful. For instance, the pre-trained weights can be used for initialization, or weak supervisory signals may be extracted for the supervision of the task-specific model. Of course, the successful exploitation of pre-trained representations requires our understanding of their nature, which differs for discriminative ones (*e.g.*, from CLIP) and generative ones (*e.g.*, from Stable Diffusion), and similar to many fields of artificial intelligence, building this understanding will likely require a sustained, empirically-driven effort from the community. Finally, a robust method for assessing the resilience of referring segmentation models—specifically, their ability to handle changes, ambiguities, and inaccuracies in expressions—would be beneficial. To this end, we hope that the advent of large language models capable of handling multi-modal inputs (*e.g.*, GPT-4 [97]) paves the way towards large-scale referring expression generation [29]. This advancement could provide data that enable comprehensive model evaluation against the vast variations of language and support the large-scale training of referring segmentation models.

We would like to conclude by suggesting that it may be desirable to extend our method to other related tasks such as referring expression comprehension, visual question answering, and spatio-temporal video grounding. And we leave these possibilities to future work.

ACKNOWLEDGEMENTS

This work is supported by the UKRI grant: Turing AI Fellowship EP/W002981/1, EPSRC/MURI grant: EP/N019474/1, Shanghai Committee of Science and Technology, China (Grant No. 20DZ1100800), National Natural Science Foundation of China (Grant No. 62206153, 62201484), Young Elite Scientists Sponsorship Program by CAST (No. 2022QNRC001), HKU Startup Fund, and HKU Seed Fund for Basic Research. We would also like to thank the Royal Academy of Engineering and FiveAI.

REFERENCES

[1] M.-M. Cheng, S. Zheng, W.-Y. Lin, V. Vineet, P. Sturgess, N. Crook, N. J. Mitra, and P. Torr, "Imagespirit: Verbal guided image parsing," in *TOG*, 2014.
[2] R. Hu, M. Rohrbach, and T. Darrell, "Segmentation from natural language expressions," in *ECCV*, 2016.
[3] K. Gavrilyuk, A. Ghodrati, Z. Li, and C. G. Snoek, "Actor and action video segmentation from a sentence," in *CVPR*, 2018.
[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.
[5] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," in *IJCV*, 2019.
[6] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *ICCV*, 2019.
[7] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. Yuille, "Recurrent multimodal interaction for referring image segmentation," in *ICCV*, 2017.
[8] R. Li, K. Li, Y. Kuo, M. Shu, X. Qi, X. Shen, and J. Jia, "Referring image segmentation via recurrent refinement networks," in *CVPR*, 2018.
[9] H. Shi, H. Li, F. Meng, and Q. Wu, "Key-word-aware network for referring expression image segmentation," in *ECCV*, 2018.
[10] D.-J. Chen, S. Jia, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, "See-through-text grouping for referring image segmentation," in *ICCV*, 2019.
[11] Z. Hu, G. Feng, J. Sun, L. Zhang, and H. Lu, "Bi-directional relationship inferring network for referring image segmentation," in *CVPR*, 2020.
[12] S. Seo, J.-Y. Lee, and B. H. Urvos, "Unified referring video object segmentation network with a large-scale benchmark," in *ECCV*, 2020.
[13] H. Wang, C. Deng, J. Yan, and D. Tao, "Asymmetric cross-guided attention network for actor and action video segmentation from natural language query," in *ICCV*, 2019.
[14] S. Huang, T. Hui, S. Liu, G. Li, Y. Wei, J. Han, L. Liu, and B. Li, "Referring image segmentation via cross-modal progressive comprehension," in *CVPR*, 2020.
[15] T. Hui, S. Liu, S. Huang, G. Li, S. Yu, F. Zhang, and J. Han, "Linguistic structure guided context modeling for referring image segmentation," in *ECCV*, 2020.
[16] B. McIntosh, K. Duarte, Y. S. Rawat, and M. Shah, "Visual-textual capsule routing for text-based video segmentation," in *CVPR*, 2020.
[17] H. Ding, C. Liu, S. Wang, and X. Jiang, "Vision-language transformer and query generation for referring segmentation," in *ICCV*, 2021.
[18] A. Botach, E. Zheltonozhskii, and C. Baskin, "End-to-end referring video object segmentation with multimodal transformers," in *CVPR*, 2022.
[19] J. Wu, Y. Jiang, P. Sun, Z. Yuan, and P. Luo, "Language as queries for referring video object segmentation," in *CVPR*, 2022.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.
[22] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *ICCV*, 2021.
[23] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, 2019.
[24] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, "Less is more: Clipbert for video-and-language learning via sparse sampling," in *CVPR*, 2021.
[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.
[26] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *CVPR*, 2022.
[27] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *ECCV*, 2016.
[28] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Modeling context between objects for referring expression understanding," in *ECCV*, 2016.
[29] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *CVPR*, 2016.
[30] A. Khoreva, A. Rohrbach, and B. Schiele, "Video object segmentation with language referring expressions," in *ACCV*, 2018.
[31] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr, "Lavt: Language-aware vision transformer for referring image segmentation," in *CVPR*, 2022, pp. 18 155–18 165.
[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural Computation*, 1997.
[33] Y. Jing, T. Kong, W. Wang, L. Wang, L. Li, and T. Tan, "Locate then segment: A strong pipeline for referring image segmentation," in *CVPR*, 2021.
[34] M. Bellver, C. Ventura, C. Silberer, I. Kazakos, J. Torres, and X. Giro-i Nieto, "Refvos: A closer look at referring expressions for video object segmentation," *arXiv:2010.00263*, 2020.
[35] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[37] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv:1706.05587*, 2017.

[38] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[39] G. Luo, Y. Zhou, X. Sun, L. Cao, C. Wu, C. Deng, and R. Ji, "Multi-task collaborative network for joint referring expression comprehension and segmentation," in *CVPR*, 2020.

[40] L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," in *CVPR*, 2019.

[41] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "Mattnet: Modular attention network for referring expression comprehension," in *CVPR*, 2018.

[42] G. Feng, Z. Hu, L. Zhang, and H. Lu, "Encoder fusion network with co-attention embedding for referring image segmentation," in *CVPR*, 2021.

[43] H. Ding, C. Liu, S. Wang, and X. Jiang, "VLT: Vision-language Transformer and query generation for referring segmentation," in *TPAMI*, 2022.

[44] Yang, Zhao and Wang, Jiaqi and Tang, Yansong and Chen, Kai and Zhao, Hengshuang and Torr, Philip H.S., "Semantics-aware dynamic localization and refinement for referring image segmentation," in *CVPR*, 2023.

[45] Liu, Chang and Ding, Henghui and Jiang, Xudong, "GRES: Generalized Referring Expression Segmentation," in *CVPR*, 2023.

[46] Tang, Jiajin and Zheng, Ge and Shi, Cheng and Yang, Sibei, "Contrastive Grouping With Transformer for Referring Image Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[47] Yan, Bin and Jiang, Yi and Wu, Jiannan and Wang, Dong and Yuan, Zehuan and Luo, Ping and Lu, Huchuan, "Universal Instance Perception as Object Discovery and Retrieval," in *CVPR*, 2023.

[48] Liu, Jiang and Ding, Hui and Cai, Zhaowei and Zhang, Yuting and Satzoda, Ravi Kumar and Mahadevan, Vijay and Manmatha, R., "PolyFormer: Referring Image Segmentation As Sequential Polygon Generation," in *CVPR*, 2023.

[49] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso, "Can humans fly? action understanding with multiple classes of actors," in *CVPR*, 2015.

[50] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[51] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017.

[52] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *ICCV*, 2017.

[53] H. Wang, C. Deng, F. Ma, and Y. Yang, "Context modulated dynamic networks for actor and action video segmentation with language queries," in *AAAI*, 2020.

[54] S. Liu, T. Hui, S. Huang, Y. Wei, B. Li, and G. Li, "Cross-modal progressive comprehension for referring segmentation," in *TPAMI*, 2021.

[55] T. Hui, S. Huang, S. Liu, Z. Ding, G. Li, W. Wang, J. Han, and F. Wang, "Collaborative spatial-temporal modeling for language-queried video actor segmentation," in *CVPR*, 2021.

[56] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *ICCV*, 2019.

[57] Z. Ding, T. Hui, J. Huang, X. Wei, J. Han, and S. Liu, "Language-bridged spatial-temporal interaction for referring video object segmentation," in *CVPR*, 2022.

[58] Miao, Bo and Bennamoun, Mohammed and Gao, Yongsheng and Mian, Ajmal, "Spectrum-guided Multi-granularity Referring Video Object Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[59] Tang, Jiajin and Zheng, Ge and Yang, Sibei, "Temporal Collection and Distribution for Referring Video Object Segmentation," in *ICCV*, 2023.

[60] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[61] Gao, Mingqi and Yang, Jinyu and Han, Jungong and Lu, Ke and Zheng, Feng and Montana, Giovanni, "Decoupling Multimodal Transformers for Referring Video Object Segmentation," in *TCSVT*, 2023.

[62] Han, Meifei and Wang, Yali and Li, Zhihui and Yao, Lina and Chang, Xiaojun and Qiao, Yu, "HTML: Hybrid Temporal-scale Multimodal Learning Framework for Referring Video Object Segmentation," in *ICCV*, 2023.

[63] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *ACL*, 2019.

[64] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *NeurIPS*, 2019.

[65] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[66] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021.

[67] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021.

[68] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: deformable transformers for end-to-end object detection," in *ICLR*, 2021.

[69] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *CVPR*, 2021.

[70] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *ICCV*, 2021.

[71] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu, "Denseclip: Language-guided dense prediction with context-aware prompting," in *CVPR*, 2022.

[72] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "Mdetr-modulated detection for end-to-end multi-modal understanding," in *ICCV*, 2021.

[73] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.

[74] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016.

[75] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[76] E. Margffoy-Tuay, J. C. Pérez, E. Botero, and P. Arbeláez, "Dynamic multimodal instance segmentation guided by natural language queries," in *ECCV*, 2018.

[77] G. Luo, Y. Zhou, R. Ji, X. Sun, J. Su, C.-W. Lin, and Q. Tian, "Cascade grouped attention network for referring expression segmentation," in *ACMMM*, 2020.

[78] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016.

[79] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[80] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[81] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3DV*, 2016.

[82] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[83] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer *et al.*, "Transformers: State-of-the-art natural language processing," in *EMNLP*, 2020.

[84] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[85] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[86] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[87] B. Alexey and B. Evgeny, "Boundary loss for remote sensing imagery semantic segmentation," in *ISNN*, 2019.

[88] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.
[89] S. Yang, M. Xia, G. Li, H. Zhou, and Y. Yu, "Bottom-up shift and reasoning for referring image segmentation," in *CVPR*, 2021.
[90] J. Hueihan, G. Juergen, Z. Silvia, S. Cordelia, and J. B. Michael, "Towards understanding action recognition," in *ICCV*, 2013.
[91] J. Pont-Tuset, F. Perazzi, S. Caelles, P. A. aez, A. Sorkine-Hornung, and L. V. Gool, "The 2017 davis challenge on video object segmentation," in *arXiv preprint arXiv:1704.00675*, 2017.
[92] L. Ye, M. Rochan, Z. Liu, X. Zhang, and Y. Wang, "Referring segmentation in images and videos with cross-modal self-attention network," in *TPAMI*, 2021.
[93] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *ECCV*, 2020.
[94] Z. Yang, Y. Tang, L. Bertinetto, H. Zhao, and P. H. Torr, "Hierarchical interaction network for video object segmentation from referring expressions," in *BMVC*, 2021.
[95] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.
[96] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.
[97] OpenAI, "Gpt-4 technical report," *arXiv:2303.08774*, 2023.

Zhao Yang received the PhD and MSc degrees from University of Oxford, and the BS degree from University of California, Los Angeles, summa cum laude. His work primarily focuses on developing methods for segmenting images and videos in an open-ended manner, based on temporal information and human language. He and his teammates won awards in academic challenges, including DAVIS and YouTube-VIS in 2019, and large hackathons, including LA Hacks and the AT&T Mobile App Hackathon in 2016.

Jiaqi Wang is a Research Scientist at Shanghai AI Laboratory. Before that, he received his Ph.D. at Multimedia Laboratory (MMLab) of The Chinese University of Hong Kong (CUHK), supervised by Prof. Dahua Lin. He also works closely with Prof. Chen Change Loy. His research interests focus on object recognition, scene understanding, and multi-modality in both 2D and 3D worlds. He and his teammates won COCO Detection Challenge in 2018 and 2019.

Xubing Ye received the BS degree from Tongji University. He is currently a master student at Shenzhen International Graduate School, Tsinghua University. His research interests include computer vision, video understanding and multimodal learning.

Yansong Tang received the BS and PhD degrees both from the Department of Automation, Tsinghua University, in 2015 and 2020, respectively. From 2020 to 2022, he served as a postdoctoral fellow with the Department of Engineering Science of the University of Oxford. He is currently a tenure-track Assistant Professor of Shenzhen International Graduate School, Tsinghua University. His research interests include computer vision, pattern recognition, and video processing. In recent years, he has authored more than 20 papers in top peer-reviewed journals and conferences such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, and CVPR. He is a member of the IEEE.

Kai Chen is a research scientist at Shanghai AI Laborotory. Prior to that, he was a director at SenseTime from 2019 to 2022. He received the PhD degree from the Chinese University of Hong Kong in 2019 and B.Eng. degree from Tsinghua University in 2015. His research interest covers computer vision and deep learning. He has published over 20 papers with 5000+ citations on top-tier conferences and journals including CVPR, ICCV, ECCV, NeurIPS, TPAMI, etc. He also leads the development of OpenMMLab open-source computer vision algorithm platform.

Hengshuang Zhao is currently an assistant professor in the Department of Computer Science at The University of Hong Kong. Previously, he was a postdoctoral researcher at Massachusetts Institute of Technology and University of Oxford. He received the Ph.D. degree in Computer Science and Engineering from The Chinese University of Hong Kong. He and his team won several champions in competitive academic challenges like ImageNet Scene Parsing, LSUN Semantic Segmentation, WAD Drivable Area Segmentation, Embodied AI iGibson Social Navigation, etc. His general research interests cover the broad area of computer vision, machine learning and artificial intelligence, with special emphasis on building intelligent visual systems. He was an area chair of CVPR'23, NeurIPS'23, WACV'23, and a senior program committee of AAAI'23. He is a member of the IEEE.

Philip H.S. Torr is currently a full professor at University of Oxford. He received the PhD degree from University of Oxford. After working for another three years at Oxford, he worked for six years for Microsoft Research, first in Redmond, then in Cambridge, founding the vision side of the Machine Learning and Perception Group. His papers have over 74,000+ citations to date. Most recently, his group has pioneered new approaches to combining deep learning with Bayesian structured graphical models which has applications in video understanding and camera localization. He has won the Marr Prize (highest award in computer vision) in 1998, and best papers at CVPR 2008 and ECCV 2010. In 2019, he was awarded the Royal Academy of Engineering Research Chair in Computer Vision in UK. In 2022 he was awarded the Turing AI World-Leading Researcher Fellowship and made a Fellow of the Royal Society in UK.