LAVT: Language-Aware Vision Transformer for Referring Image Segmentation

Zhao Yang[1], Jiaqi Wang[2], Yansong Tang[1], Kai Chen[2,3], Hengshuang Zhao[1,4], Philip H.S. Torr[1]
[1]University of Oxford, [2]Shanghai AI Laboratory,
[3]SenseTime Research, [4]The University of Hong Kong

Abstract

Referring image segmentation is a fundamental vision-language task that aims to segment out an object referred to by a natural language expression from an image. One of the key challenges behind this task is leveraging the referring expression for highlighting relevant positions in the image. A paradigm for tackling this problem 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 advancements 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. By conducting cross-modal feature fusion in the visual feature encoding stage, we can leverage the well-proven correlation modeling power of a Transformer encoder for excavating helpful multi-modal context. This way, accurate segmentation results are readily harvested with a light-weight mask predictor. Without bells and whistles, our method surpasses the previous state-of-the-art methods on RefCOCO, RefCOCO+, and G-Ref by large margins.

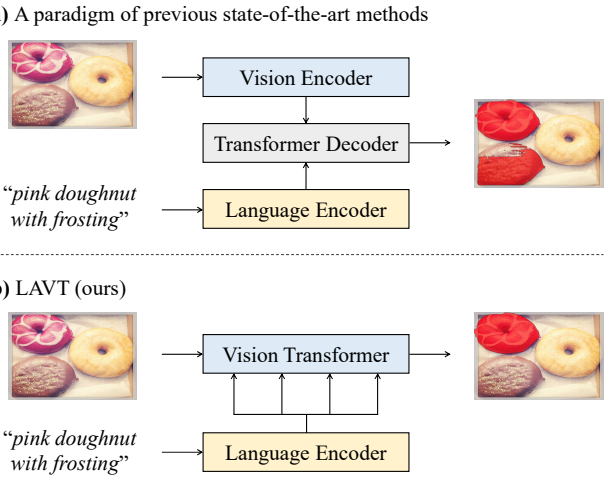

Figure 1. The task of referring image segmentation takes one image and one text description as inputs, and predicts a mask delineating the object specified in the description. (a) The previous state-of-the-art method (*i.e.*, VLT [12]) leverages a vision-language Transformer decoder for cross-modal feature fusion. (b) Conversely, we propose to directly integrate linguistic information into visual features at intermediate levels of a vision Transformer network, where beneficial vision-language cues are jointly exploited. A light-weight mask predictor can thus readily replace the complicated cross-modal decoder in previous counterparts.

1. Introduction

Given an image and a text description of the target object, referring image segmentation aims at predicting a pixel-wise mask that delineates that object [8, 18]. It yields great value for various applications such as language-based human-robot interaction [53] and image editing [5]. Differently from conventional single-modality visual segmentation tasks based on fixed category conditions [29, 63], referring image segmentation has to deal with the much richer vocabularies and syntactic varieties of human natural languages. In this task, the target object is inferred from a free-form expression, which includes words and phrases presenting the concepts of entities, actions, attributes, positions, *etc*., organized by syntactic rules. Therefore, the key challenge of this task is to exploit visual features that are relevant to the given text conditions.

There have been growing efforts devoted to referring image segmentation over the past few years. A widely adopted paradigm 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. In concrete terms, the fusion strategies include concatenation [18], recurrent interaction [27, 30], cross-modal attention [4, 20, 46, 58], multi-modal graph

reasoning [21], linguistic structure-guided context modeling [22], *etc*. Recent advances (*e.g*., [12]) bring performance improvements via employing a cross-modal Transformer [52] 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 [19, 26, 36, 44].

Although great progress has been achieved, the potentiality of the Transformer for enhancing the quality of referring image segmentation is still far from being sufficiently explored in the conventional paradigm. Specifically, cross-modal interactions occur only after feature encoding. And a cross-modal decoder is solely 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 network, leading to a hierarchical language-aware visual encoding scheme. Specifically, we densely integrate linguistic features into visual features via a pixel-word attention mechanism, which occurs at each stage of the network. The beneficial vision-language cues are then exploited by the following Transformer blocks, *e.g*., [32], in the next encoder stage. This approach enables us to forgo a complicated cross-modal decoder, since the extracted language-aware visual features can be readily adopted to harvest accurate segmentation masks with a lightweight mask predictor.

To evaluate the effectiveness of the proposed method, we conduct extensive experiments on various mainstream referring image segmentation datasets. Our LAVT achieves 72.73%, 62.14%, 61.24%, and 60.50% overall IoU on the validation sets of RefCOCO [61], RefCOCO+ [61], G-Ref (UMD partition) [39], and G-Ref (Google partition) [41], improving the state of the art for these datasets by absolute margins of 7.08%, 6.64%, 6.84%, and 8.57%, respectively.

To summarize, our contributions are twofold:

- We propose LAVT, a Transformer-based referring image segmentation framework that performs language-aware visual encoding in place of cross-modal fusion in a post-feature extraction step.

- We achieve new state-of-the-art results on three datasets for referring image segmentation, demonstrating the effectiveness and generality of the proposed method. Source code will be available at LAVT-RIS.

2. Related work

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 recurrent neural networks [17, 18, 25, 27, 30] and language Transformers [2, 11] to encode language inputs. To encode visual inputs, vanilla fully convolutional networks [18, 30, 34], DeeplabV3 [2, 6, 27], and DarkNet [25, 38, 45] have been successively employed in previous methods with the purpose of learning discriminative representations.

The multi-modal feature fusion module is the key component that prior arts focus on. For example, Hu *et al*. [18] propose the first baseline based on the concatenation operation, which is improved by Liu *et al*. [30] with a recurrent strategy. Shi *et al*. [46], Chen *et al*. [4], Ye *et al*. [58], and Hu *et al*. [20] model cross-modal relations between language and vision features via various attention mechanisms. Yu *et al*. [60] and Huang *et al*. [21] leverage knowledge about sentence structures to capture different concepts (*e.g*., categories, attributes, relations, *etc*.) in multi-modal features, while Hui *et al*. [22] exploit syntactic structures among words for guiding multi-modal context aggregation.

The methods most related to ours are VLT [12] and EFN [14], 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. Differently from [12], we propose an early fusion scheme which effectively exploits the Transformer encoder for modeling multi-modal context. Compared to [14], 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.

Transformer is first introduced as a sequence-to-sequence deep attention-based language model [52], and has dominated the natural language processing (NLP) field [9, 11, 56] due to its strong capability on global context modeling. More recently, it has achieved great success on various computer vision tasks, *e.g*., image classification [13, 32, 50], action recognition [1, 33], object detection [3, 32, 64], and semantic segmentation [32, 49, 62].

There has also been a rich line of work on Transformers in the intersection area of computer vision and NLP. For example, Radford *et al*. devise a large-scale pretraining model, named CLIP [44], which applies contrastive learning [15, 16, 48] on features learned by a vision Transformer and a language Transformer. Hu *et al*. [19] propose a Unified Transformer (UniT) model that jointly learns multiple vision-language tasks across different domains. Besides, growing efforts have been devoted to other tasks

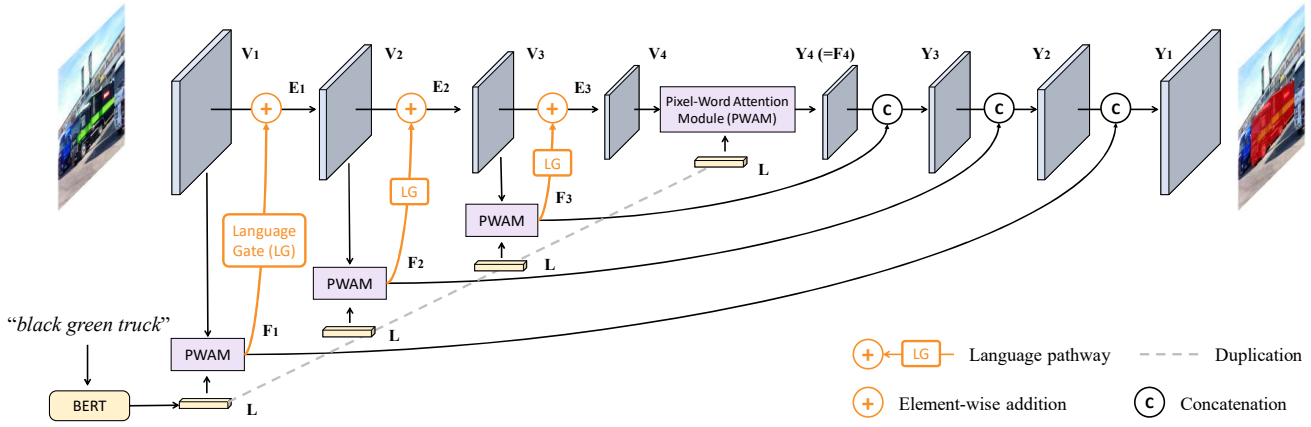

Figure 2. Overall pipeline of the proposed LAVT. We leverage a hierarchical vision Transformer [32] to perform language-aware visual encoding. At each stage, visual feature maps V_i, $i \in \{1, 2, 3, 4\}$ are encoded from the corresponding stage of Transformer layers (which are described in Sec. 3.1 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. 3.2). Next, we adaptively fuse F_i with the original V_i via a language pathway (Sec. 3.3). 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 decoder head (Sec. 3.4) produces the final segmentation output.

such as visual question answering [36] and text-to-video retrieval [26]. However, to the best of our knowledge, there have been very few attempts on designing a unified Transformer model for the task of referring image segmentation.

3. Method

Fig. 2 illustrates the pipeline of our Language-Aware Vision Transformer (LAVT), which leverages a hierarchical vision Transformer to jointly embed language and vision information to facilitate cross-modal alignments. In this section, we start by introducing our language-aware visual encoding strategy in Sec. 3.1, which is achieved with a pixel-word attention module detailed in Sec. 3.2 and a language pathway detailed in Sec. 3.3. Then in Sec. 3.4 we describe the light-weight mask predictor used to obtain final results.

3.1. 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 $L \in \mathbb{R}^{C_t \times T}$, where C_t and T 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 contents) 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, V_i are combined with language features L via the multi-modal feature fusion module ϕ_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 weighted 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 original four stages in a Swin Transformer [32]. 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 in the language features with visual clues from the visual features. And the gating unit is what we refer to as the language gate (LG), which is a special unit that we devise for regulating the flow of linguistic information through the language pathway (LP) to the visual features.

3.2. Pixel-Word Attention Module

In order to separate the "referent" object from its background, it is important to align the visual and linguistic

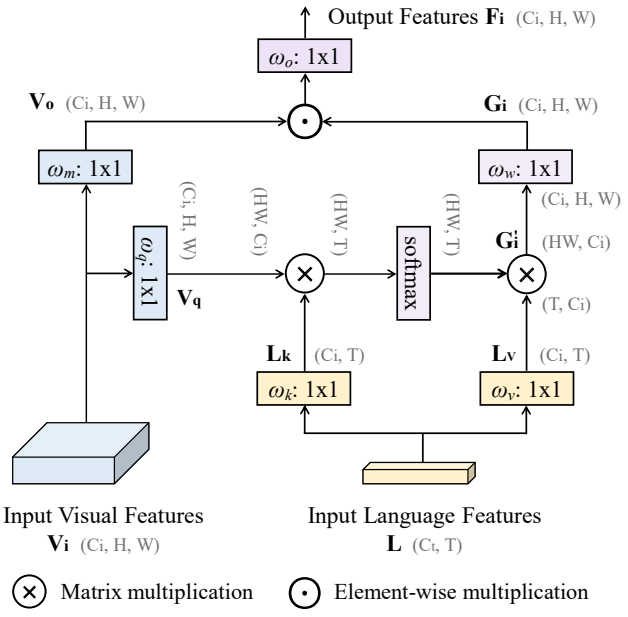

Figure 3. Pipeline of the pixel-word attention module (PWAM). First, a single-head scaled dot-product attention [52] is performed using the input visual feature maps V_i as queries and the input language feature maps L as keys and values. The results, G_i, are a set of language feature maps of same spatial sizes as V_i. G_i are then multiplied element-wise with a projection of the input visual feature maps V_o, followed by a projection before the final output. A detail which we found important empirically is the adoption of an instance normalization [51] layer in the projection functions ω_q and ω_w (see the text below and Table 3).

representations of an 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 [40], concatenations [18, 27, 40], cross-modal attentions [14, 20, 37, 47, 58], graph neural networks [31], *etc*. Compared to most of the previous cross-modal attention mechanisms [14, 20, 37, 47, 58], our pixel-word attention module (PWAM) produces a much smaller memory footprint as we avoid 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 language features $L \in \mathbb{R}^{C_t \times T}$, PWAM performs multi-modal fusion in two steps, as introduced in the following. First, at each spatial location, PWAM aggregates the language 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

$$V_q = \text{flatten}(\omega_q(V_i)), \tag{1}$$
$$L_k = \omega_k(L), \tag{2}$$
$$L_v = \omega_v(L), \tag{3}$$
$$G_i' = \text{softmax}(\frac{V_q^T L_k}{\sqrt{C_i}}) L_v^T, \tag{4}$$
$$G_i = \omega_w(\text{unflatten}(G_i'^T)), \tag{5}$$

where ω_q, ω_k, ω_v, and ω_w are projection functions. Each of the language projections ω_k and ω_v is implemented as 1×1 convolution with C_i number of output channels, and each of the visual projection ω_q and the final projection ω_w is implemented as 1×1 convolution followed by an instance normalization layer, 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 exactly 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 [52] using visual features V_i as the query and language features L as the key and the value, with the addition of instance normalization in the key projection and the output projection.

Second, after obtaining the language 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, our step is described as follows

$$V_o = \omega_m(V_i), \tag{6}$$
$$F_i = \omega_o(V_o \odot G_i), \tag{7}$$

where $\odot$ denotes element-wise multiplication and ω_m and ω_o 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 [42] nonlinearity.

3.3. 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 re-scale each element in F_i. The language pathway is schematically illustrated in Fig. 4 and mathematically described as follows

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

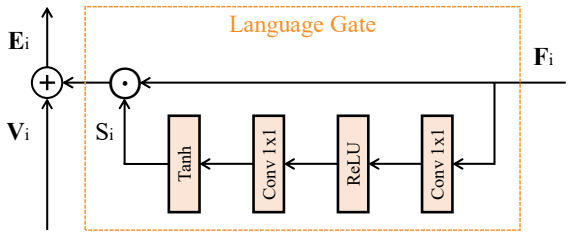

Figure 4. The schema of the language pathway, which leverages a language gate (LG) for controlling multi-modal information flow. LG is implemented as a two-layer perceptron.

where $\odot$ indicates element-wise multiplication and γ is a two-layer perceptron, with the first layer being a 1×1 convolution followed by ReLU [42] nonlinearity and the second layer being a 1×1 convolution followed by a hyperbolic tangent function. As detailed in the ablation studies in Table 3, 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.

3.4. 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 [24] and ReLU [42] nonlinearity. The final feature maps, Y_1, are projected into two class score maps via a 1×1 convolution.

3.5. Implementation

We implement our method in PyTorch [43] and use the BERT implementation from HuggingFace's Transformer library [54]. The Transformer layers in LAVT are initialized with classification weights pre-trained on ImageNet-22K [10] from the Swin Transformer [32]. Our language encoder is the base BERT model with 12 layers and hidden size 768 from [52] and is initialized using the official pre-trained weights. The rest of weights in our model are randomly initialized. The model is optimized with cross-entropy loss. Following [32], we adopt the AdamW [35] optimizer with weight decay 0.01 and initial learning rate 0.00005 with polynomial learning rate decay. We train our model for 40 epochs with batch size 32. 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.

4. Experiments

4.1. Datasets and Metrics

We evaluate our method on three standard benchmark datasets, RefCOCO [61], RefCOCO+ [61], and G-Ref [39, 41]. Images in the three datasets are collected from the MS COCO dataset [29] 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 annotated in two-player games and are thus very succinct (containing 3.5 words on average). In contrast, expressions in G-Ref are more complex (containing 8.4 words on average) and more descriptive, which makes the dataset particularly challenging. On the other hand, 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 better evaluate an algorithm's ability to comprehend instance-level details. A special characteristic of RefCOCO+ is that location words are banned in its expressions, which also makes it a more challenging dataset. Finally, there are two different partitions of the G-Ref dataset, one by UMD [41] and the other by Google [39]. We report performance on both partitions. When evaluating on each dataset, we train our model on the training set of that dataset and do not employ additional training data.

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.

4.2. Comparison with Others

In Table 1, we evaluate LAVT against the state-of-the-art referring image segmentation methods on the RefCOCO [61], RefCOCO+ [61], and G-Ref [39, 41] datasets using the oIoU metric. LAVT outperforms all previous methods on all evaluation subsets of all three datasets. Compared with the second-best method, VLT [12], LAVT achieves higher performance with absolute margins of

	RefCOCO			RefCOCO+			G-Ref		
	val	test A	test B	val	test A	test B	val (U)	test (U)	val(G)
DMN [40]	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [28]	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [60]	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CMSA [59]	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
CAC [7]	58.90	61.77	53.81	-	-	-	46.37	46.95	44.32
STEP [4]	60.04	63.46	57.97	48.19	52.33	40.41	-	-	46.40
BRINet [20]	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [21]	61.36	64.53	59.64	49.56	53.44	43.23	-	-	49.05
LSCM [23]	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
CMPC+ [31]	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
MCN [38]	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
EFN [14]	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [55]	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [37]	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [25]	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [12]	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	49.76
LAVT (Ours)	**72.73**	**75.82**	**68.79**	**62.14**	**68.38**	**55.10**	**61.24**	**62.09**	**60.50**

Table 1. Comparison with state-of-the-art methods on three standard benchmark datasets. U: The UMD partition. G: The Google partition.

7.08%, 7.53%, and 6.06% on the validation, testA, and testB subsets of RefCOCO, respectively. Similarly, LAVT attains noticeable improvements over the previous state of the art on RefCOCO+ with wide margins of 6.64%, 9.18%, and 5.74% on the validation, testA, and testB subsets, respectively. On the most challenging G-Ref dataset (which contains significantly longer expressions), LAVT surpasses the respective second-best method on the validation and test subsets from the UMD partition by absolute margins of 6.84% and 5.44%, respectively. Similarly on the validation set from the Google partition, LAVT outperforms the second-best method EFN [14] by an absolute margin of 8.57%. This performance is achieved without using RefCOCO as additional training data in contrast to EFN.

4.3. Ablation Study

We conduct several ablations to evaluate the effectiveness of the key components in our proposed network.

Language pathway (LP). Table 2 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 1.95 and 2.50 absolute points in overall IoU and mean IoU, respectively. In addition, precision drops by 3 to 4 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 obtained via dense pixel-word attention in PWAM, with a globally pooled sentence feature vector obtained by average pooling all words, the feature extraction method adopted in [57]. As shown in Table 2, this ablation leads to a performance drop of 1.70 and 2.15 absolute points in overall IoU and mean IoU, respectively, and a drop of 1 to 2 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.

LP	PWAM	P@0.5	P@0.7	P@0.9	oIoU	mIoU
✓	✓	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
	✓	81.46	70.80	30.95	70.78	71.96
✓		81.76	72.76	32.46	71.03	72.31
		77.87	66.93	27.95	68.82	68.87

Table 2. Main ablation results on the RefCOCO validation set.

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 3 (a), we compare the sigmoid function and the hyperbolic tangent function as the final activation function in LG. Using the sigmoid function leads to inferior results.

Normalization layer in PWAM. As described in Sec. 3.2, we adopt an instance normalization layer before the final output in the visual projection function ω_q and the final projection function ω_w in PWAM. As we illustrate in Table 3 (b), this particular choice of normalization function has a non-trivial effect on performance. In addition to in-

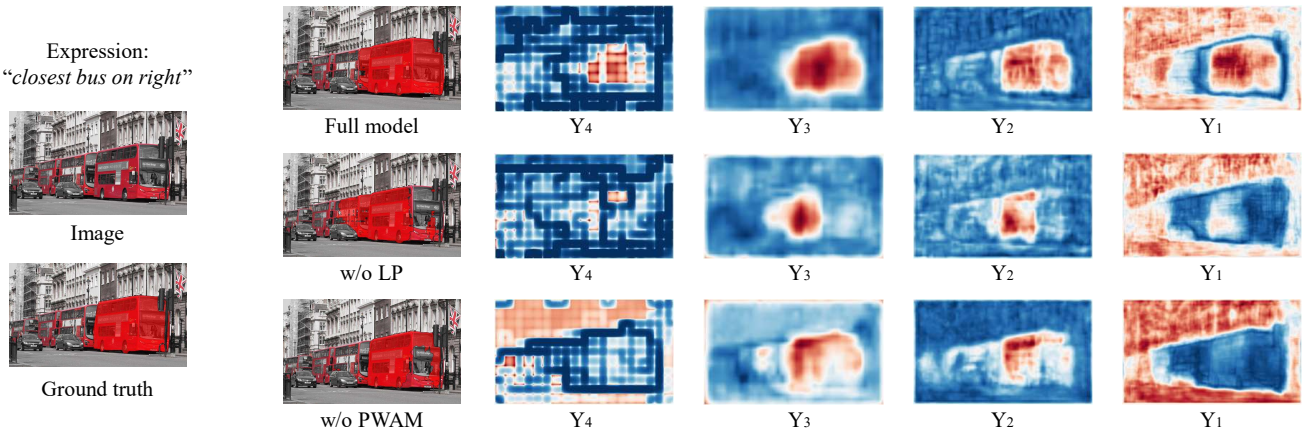

Figure 5. Visualized predictions and feature maps on an example from the RefCOCO validation set. From top to bottom, the left-most column illustrates the input expression, the input image, and the ground-truth mask overlaid on the input image. In each row, we visualize the predicted mask and the feature 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.

	P@0.5	P@0.7	P@0.9	oIoU	mIoU
(a) activation function in the language gate (LG)					
Tanh (*)	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
Sigmoid	81.89	72.71	33.35	70.49	72.47
(b) normalization layer in pixel-word attention module (PWAM)					
InstanceNorm (*)	**84.46**	**75.28**	**34.30**	**72.73**	**74.46**
LayerNorm	82.97	74.15	33.99	71.92	73.32
BatchNorm	82.89	73.82	33.53	71.59	73.09
None	81.91	72.73	33.11	70.66	72.34
(c) features used for final classification					
F_4, F_3, F_2, F_1 (G*)	**84.46**	**75.28**	34.30	**72.73**	**74.46**
F_4, F_3, F_2, F_1 (NG)	84.00	74.96	33.47	72.24	73.94
E_4, E_3, E_2, E_1 (G)	83.84	74.96	34.48	72.06	73.98
E_4, E_3, E_2, E_1 (NG)	84.33	74.94	**34.77**	72.27	74.12
V_4, V_3, V_2 (G)	83.36	74.47	32.61	71.38	73.29
V_4, V_3, V_2 (NG)	83.83	74.76	32.14	72.29	73.67

Table 3. Ablation studies 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.

stance normalization (our default choice), we experiment with batch normalization, layer normalization, and without having a normalization layer in the functions ω_q and ω_w. All three other choices lead to 1 to 2 absolute points drop in the overall IoU and mean IoU metrics. Among these three choices, using batch normalization or layer normalization produces better results 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 the final segmentation. 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. As a result, we use E_i, $i \in \{1, 2, 3, 4\}$ to predict the segmentation masks in this ablation study. Also, multi-modal information has been progressively integrated into V_2, V_3, and V_4 along the bottom-up computation pathway, whereas V_1 contains pure visual information. Therefore we do not use V_1 for prediction in comparison. In Table 3 (c), we report segmentation results when using each type of features with and without our proposed LG (indicated by "G" and "NG", respectively). Table 3 (c) shows that using our default choice of F_i with LG produces the best overall results among all choices. Also we observe that while LG has a positive effect when using F_i for segmentation, it slightly degrades the performance when E_i (72.06% vs. 72.27% in oIoU) or V_i (71.38% vs. 72.29% in oIoU) are used for segmentation.

Visualized predictions. In Fig. 5, we visualize the predictions and feature maps of our full model and two ablated models (without the language pathway ("w/o LP") and without the pixel-word attention module ("w/o PWAM"), respectively). From the first row, we can observe that the higher-level feature maps (*i.e.*, Y_4, Y_3, Y_2) in our full model can accurately locate the semantic concept given in text, while the low-level feature maps (*i.e.*, Y_1) contain rich boundary information important to binary segmentation. Comparing the predicted masks between three models, we can observe that the removal of LP and the removal of PWAM both lead to false negative predictions on the front

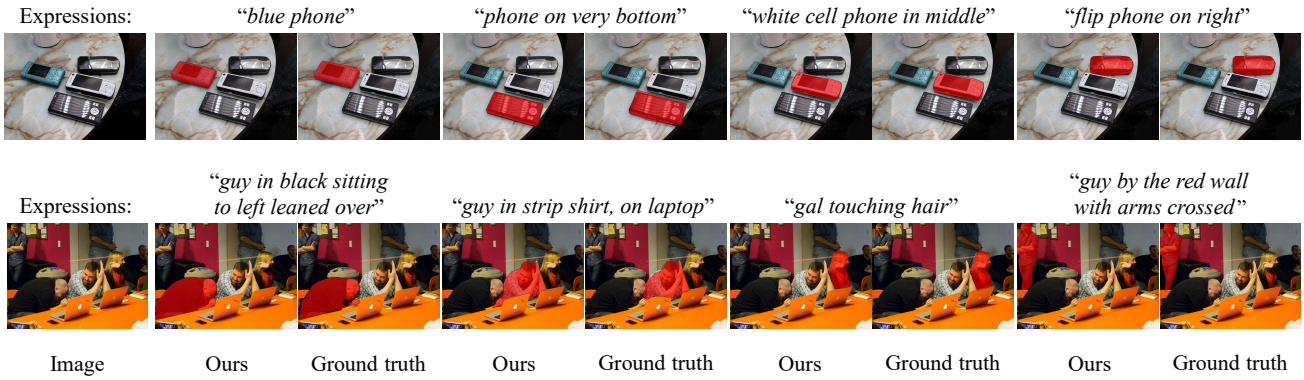

Figure 6. Visualizations of our predicted masks and the ground-truth masks on two examples from the RefCOCO validation set.

window 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 example visualizations are shown in Fig. 6.

Comparisons with VLT [12] and EFN [14]. To further validate the effectiveness of our proposed method of fusing cross-modal information via a vision Transformer encoder network, in Table 4, we unify the different training conditions and the backbone networks adopted in two previous state-of-the-art methods, VLT and EFN, and compare our performance with theirs on a fair ground. VLT is representative of methods that employ a cross-modal Transformer decoder. Conversely, EFN is representative of methods which fuse cross-modal information via an encoder network and additionally rely on a complicated decoder for achieving the best performance. We adopt Swin-B [32] as the visual backbone network and train both methods under exactly the same settings (described in Sec. 3.5) as ours. As shown in Table 4, under the same settings, our method outperforms VLT and EFN on the validation set of RefCOCO across all metrics. Specifically, on the overall IoU and mean IoU metrics, we achieve 1.84% and 2.48% absolute improvements over VLT, and 1.97% and 1.51% absolute improvements over EFN, respectively. On the precision metric, our method achieves an absolute 9.66% gain on the challenging IoU threshold of 0.9 compared with VLT, and surpasses EFN by absolute 1.91%, 2.01%, and 2.62% on the 0.5, 0.7, and 0.9 thresholds, respectively. These precision gains highlight the segmentation accuracy of our method.

To further verify that our proposed LAVT encoding scheme is more effective than its counterpart cross-modal decoder approach for addressing referring image segmentation, 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 "ours + VLT" in Table 4), employing a Transformer decoder to perform additional cross-modal feature fusion after language-aware visual encoding by LAVT generally does not bring extra performance gains (except a marginal 0.11% improvement in P@0.5).

Method	P@0.5	P@0.7	P@0.9	oIoU	mIoU
EFN (Swin-B)† [14]	82.55	73.27	31.68	70.76	72.95
VLT (Swin-B) [12]	83.24	72.81	24.64	70.89	71.98
Ours + VLT [12]	**84.57**	75.14	26.36	72.12	73.57
Ours	84.46	**75.28**	**34.30**	**72.73**	**74.46**

Table 4. Comparison between our method, VLT [12], and EFN [14] under exactly the same settings and using Swin-B [32] as the visual backbone on the RefCOCO validation set.

5. Conclusion

In this paper, we have proposed a Language-Aware Vision Transformer (LAVT) framework for referring image segmentation, which leverages the multi-stage design of a vision Transformer for jointly encoding multi-modal inputs. Experimental results on three benchmarks have demonstrated its advantage with respect to the state of the art.

Limitation. Beyond the proposed language-aware encoding method based on vision Transformer, there are other potential choices which have not been fully explored in this paper, *e.g.*, to leverage the language Transformer encoder for handling multi-modal inputs. In this design, the vision and language Transformers can mutually benefit each other, potentially leading to even stronger performance.

Future work. In the future, we will explore other cross-modal fusion strategies based on the Transformer architecture (*e.g.*, the method we describe above). It is also desirable to apply our approach to other related tasks such as referring video segmentation and visual question answering.

Acknowledgements. This work is supported by the EPSRC

†As public code is not available, these results are based on our implementation.

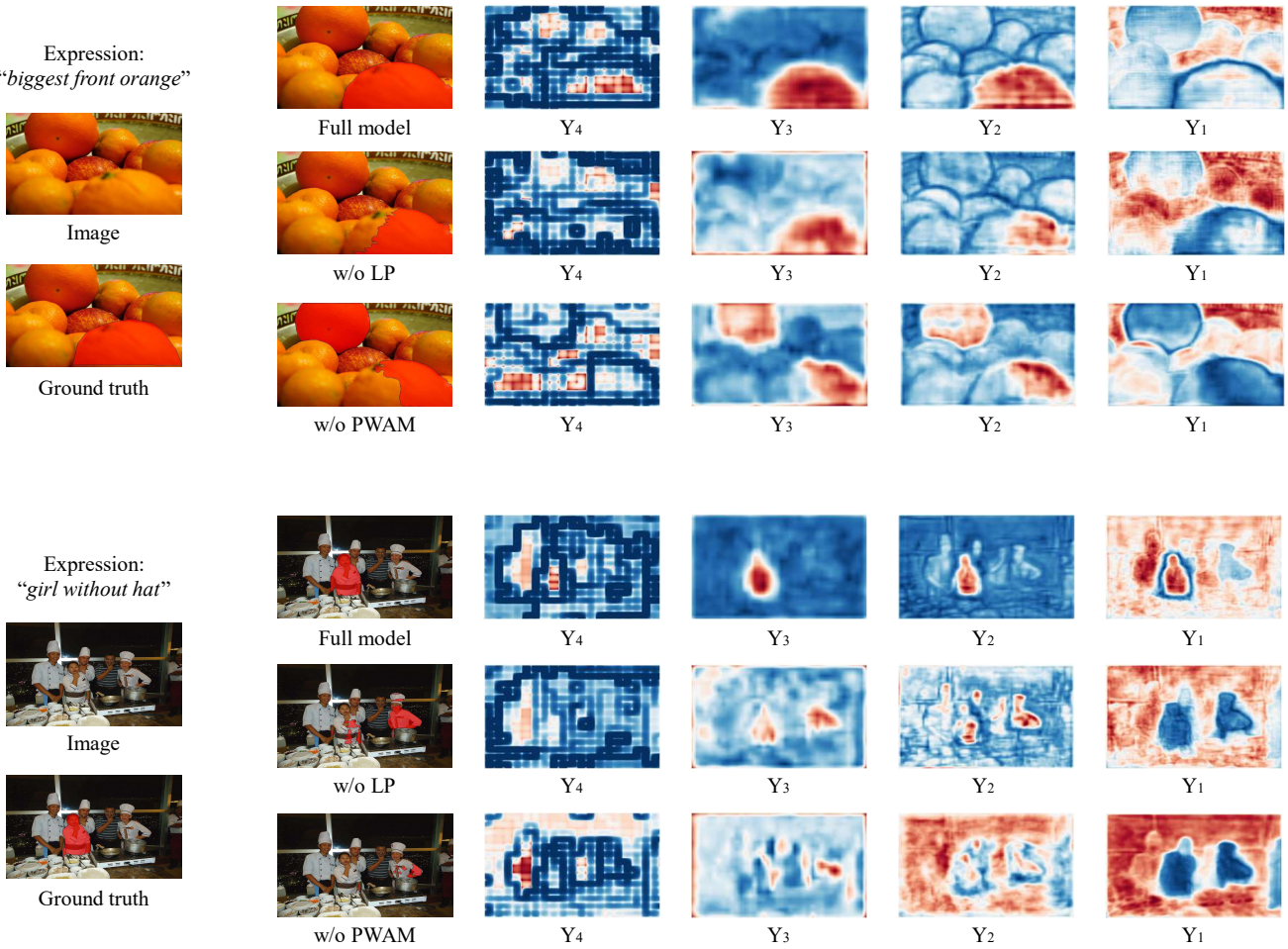

Figure 7. Additional visualizations of predictions and feature maps from the RefCOCO validation set. For each example, the left-most column illustrates the input expression, the input image, and the ground-truth mask overlaid on the input image. In each row, we visualize the predicted mask and the feature 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.

grant/Turing AI Fellowship EP/W002981/1, EPSRC/MURI grant EP/N019474/1, and the Shanghai Committee of Science and Technology, China (Grant No. 20DZ1100800). We would also like to thank the Royal Academy of Engineering, Tencent, and FiveAI.

Figure 8. Visualizations of our predicted masks and the ground-truth masks on examples from the RefCOCO validation set. Examples enclosed with green lines are successful cases, and those enclosed with red lines are failure cases. In the successful cases, our predictions are nearly identical to the ground truth and are sometimes more accurate than the ground truth (see the second example from the right column, where part of the body of the man behind the chair is missing in the annotation). Among the two demonstrated failure cases, the first one is caused by ambiguity in the given expression (there are two boys that are on a skateboard and our model segments out both) and the second one is caused by lack of knowledge of what a "pac man" is (obviously having not played the game Pac-Man, our model fails to associate the shape of the pizza to the shape of a Pac-Man).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 2

[2] Miriam Bellver, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres, and Xavier Giro-i Nieto. Refvos: A closer look at referring expressions for video object segmentation. *arXiv:2010.00263*, 2020. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[4] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *ICCV*, 2019. 1, 2, 6

[5] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018. 1

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 2

[7] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency. In *BMVC*, 2019. 6

[8] Ming-Ming Cheng, Shuai Zheng, Wen-Yan Lin, Vibhav Vineet, Paul Sturgess, Nigel Crook, Niloy J. Mitra, and Philip Torr. Imagespirit: Verbal guided image parsing. In *TOG*, 2014. 1

[9] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 5

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[12] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *ICCV*, 2021. 1, 2, 5, 6, 8

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[14] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *CVPR*, 2021. 2, 4, 6, 8

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 2

[18] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016. 1, 2, 4

[19] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *ICCV*, 2021. 2

[20] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *CVPR*, 2020. 1, 2, 4, 6

[21] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In *CVPR*, 2020. 2, 6

[22] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *ECCV*, 2020. 2

[23] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *ECCV*, 2020. 6

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[25] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *CVPR*, 2021. 2, 6

[26] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021. 2, 3

[27] Ruiyu Li, Kai-Can Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, 2018. 1, 2, 4

[28] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, 2018. 6

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, 2014. 1, 5

[30] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, 2017. 1, 2

[31] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. In *TPAMI*, 2021. 4, 6

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 3, 5, 8

[33] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv:2106.13230*, 2021. 2

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[36] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2, 3

[37] Gen Luo, Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jinsong Su, Chia-Wen Lin, and Qi Tian. Cascade grouped attention network for referring expression segmentation. In *ACMMM*, 2020. 4, 6

[38] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, 2020. 2, 6

[39] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 2, 5

[40] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, 2018. 4, 6

[41] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 2, 5

[42] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 4, 5

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[45] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018. 2

[46] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *ECCV*, 2018. 1, 2

[47] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *ECCV*, 2018. 4

[48] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016. 2

[49] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 2

[50] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016. 4

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 4, 5

[53] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*, 2019. 1

[54] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020. 5

[55] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *CVPR*, 2021. 6

[56] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019. 2

[57] Zhao Yang, Yansong Tang, Luca Bertinetto, Hengshuang Zhao, and Philip H.S. Torr. Hierarchical interaction network for video object segmentation from referring expressions. In *BMVC*, 2021. 6

[58] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 2019. 1, 2, 4

[59] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 2019. 6

[60] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 2, 6

[61] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 2, 5

[62] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 2

[63] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019. 1

[64] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2