Bridging Vision and Language Encoders: Parameter-Efficient Tuning for Referring Image Segmentation

Zunnan Xu[1*] Zhihong Chen[2,3*] Yong Zhang[4] Yibing Song[5] Xiang Wan[3] Guanbin Li[1†]
[1]Sun Yat-sen University [2]The Chinese University of Hong Kong, Shenzhen
[3]Shenzhen Research Institute of Big Data [4]Tencent AI Lab [5]AI[3] Institute, Fudan University
xuzn3@mail2.sysu.edu.cn zhihongchen@link.cuhk.edu.cn
{zhangyong201303, yibingsong.cv}@gmail.com wanxiang@sribd.com liguanbin@mail.sysu.edu.cn

Abstract

Parameter Efficient Tuning (PET) has gained attention for reducing the number of parameters while maintaining performance and providing better hardware resource savings, but few studies investigate dense prediction tasks and interaction between modalities. In this paper, we do an investigation of efficient tuning problems on referring image segmentation. We propose a novel adapter called Bridger to facilitate cross-modal information exchange and inject task-specific information into the pre-trained model. We also design a lightweight decoder for image segmentation. Our approach achieves comparable or superior performance with only 1.61% to 3.38% backbone parameter updates, evaluated on challenging benchmarks. The code is available at https://github.com/kkakkkka/ETRIS.

1. Introduction

Referring image segmentation (RIS) aims to predict a mask for the target object described by a given natural language sentence based on the input image and text. This task is distinct from semantic segmentation, which assigns each pixel in an image with a label from a fixed word set. Instead, RIS needs to recognize the objects indicated by the language expression, which is of greater complexity due to its arbitrary context length and involving an open-world vocabulary such as object names, attributes, positions, etc.

Recent studies [43, 12, 11] have shown the effectiveness of fine-tuning general-purpose pre-trained models for visual grounding. However, these approaches have a separate copy of fine-tuned model parameters for each dataset, making it expensive to deploy models across multiple scenarios. This issue is particularly significant for large-scale pre-trained models, which now consist of hundreds of millions to trillions of parameters [31, 50, 5].

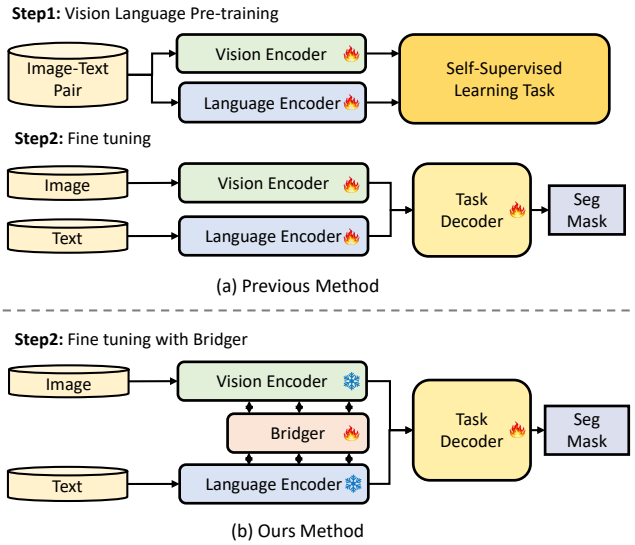

Figure 1: Previous method vs. our method. (a) The conventional method pre-trains visual language models on datasets with image-text pairs using self-supervised learning and fine-tunes them on downstream tasks. (b) We propose Bridger, an Adapter-like module that incorporates inductive biases and task-specific information into the pre-trained model.

Therefore, we ask an essential question: *can the model maintain a competitive performance with pre-trained backbone network parameters fixed?*. Various parameter-efficient training methods [16, 20, 15, 4, 9, 50] have been proposed to achieve a balance between parameter efficiency and performance. However, most of the existing methods are limited to either single-modal tasks [16, 20, 9] or simple classification tasks [15, 4, 50] with few studies focusing on dense prediction tasks and the interaction between different modalities,

*Equal contribution
†Corresponding author

which limits their generality.

We aim to adapt pre-trained vision-language models for referring image segmentation with comparable performance to full fine-tuning, but in a more parameter-efficient way, as demonstrated in Figure 1. This approach improves adaptability and eliminates the parameter inefficiencies and prohibitive expenses associated with previous methods that require creating separate copies of fine-tuned backbone model parameters for each dataset. In detail, firstly, we introduce an additional network named Bridger that does not require pre-training and can be seamlessly integrated into the original architecture of the pre-trained model, where we introduce vision-specific inductive biases and facilitate interaction between the dual encoder. There are two tailored modules for Bridger: (i) a spatial prior module for capturing the local semantics (spatial prior) from feature maps of the intermediate layer and (ii) a cross-modal attention module that enables information exchange between the two modalities. Secondly, we designed a *lightweight* task-specific decoder for referring image segmentation to make further alignment on visual and linguistic features. Under this framework, the backbone network can be any general-propose (dual-encoder) model that is pre-trained on vision-language datasets, and we adopt CLIP [40], a pre-trained image-text alignment model, as our vision and language encoders. As a result, utilizing ViT [13] and ResNet [19] as the visual backbone and updating only 1.61% to 3.38% parameters, our framework achieves comparable or even better performance than previous methods which employ the same backbone for full-finetuning. Our main contributions are as follows:

- We propose to do an in-depth investigation of the parameter-efficient tuning problem on the dense prediction tasks. To the best of our knowledge, this is the first empirical study to date that considers this problem.
- We design a novel Bridger that can be seamlessly integrated into any pre-trained dual-encoder vision-language models to enhance and interact with their intermediate features. It incorporates vision-specific inductive biases and task-specific information and can be integrated with prompts, adapter, and their variants.
- We also propose a lightweight decoder for referring image segmentation to further align visual and linguistic features.
- Extensive experiments and analyses demonstrate the effectiveness of the proposed approach, where it achieves comparable performance compared to existing full fine-tuning methods while updating only 1.61% to 3.38% parameters.[1]

2. Related work

This work aims to design an *efficient tuning* approach to *referring image segmentation* built upon pre-trained *vision-language models*. In this section, we summarize previous literature and discuss the relations and differences.

Vision-Language Models (VLMs) target exploring a unified representation for vision and language modalities to tackle vision-and-language tasks. They can be generally divided into two types of workflow: single-stream and dual-stream. The former includes [6, 33, 38, 8, 10, 30], which use a fusion module to interact the visual and textual embeddings; The latter consists of studies, e.g., [40, 25, 31], which use contrastive learning to align the vision and language embeddings. Our work focus on the efficient tuning of dual-stream model due to their expansibility and the necessity of aligning features when transferring the models to downstream tasks.

Parameter-efficient Tuning (PET) aims to reduce the number of trainable parameters of a pre-trained model when transferring it to the downstream tasks. Compared with fine-tuning that retrains the whole model on a specific task, PET can make tuning a large model feasible when deploying it to each user considering the recent proliferation in model sizes. Recent PET methods can be divided into three types: (i) updating newly added parameters to the model or input like Adapter [20], Prefix-tuning [34] and Prompt tuning [50]; (ii) sparsely updating a small number of parameters of the model like Bit-Fit [49] and Diff Pruning [16]; (iii) low-rank factorization for the weights to be updated like LoRA [21], Compacter [27] and Consolidator [17]. Adapters balance performance and extensibility in computer vision and natural language processing. Nonetheless, most current work focuses on classification and generation tasks, neglecting dense prediction tasks like segmentation and special design for multi-modal tasks. Our method addresses this gap by designing a multi-modal adapter-like module that enhances feature interaction between the two encoders of the pre-trained vision language model, which facilitates efficient transfer to downstream tasks.

Referring Image Segmentation (RIS) aims to segment the target objects referred by natural language descriptions by understanding the given images and expressions. Early works can be tracked back to those CNN-LSTM-based approaches, e.g., RRN [32] and RMI [36]. They used CNN and LSTM to extract visual and linguistic features, respectively. These features are concatenated to obtain cross-modal features, which are then fed into an FCN to perform the segmentation. With the rapid development of Transformer, many works have begun to explore the powerful representation of the attention mechanism. Simply concatenating features from different modalities, MDETR [26] achieves great performance on different VL-tasks by feeding the fusion features into the Transformer encoder and decoder. VLT [12] has designed a query generation module to augment global context information, thereby enriching linguistic expressions and enhancing robustness. Taking advantage of the strong image-text alignment ability of CLIP [40], CRIS [43] fo-

[1] Note that we do not count the parameters of the task-specific decoder since it is required by all the baselines for the segmentation prediction.

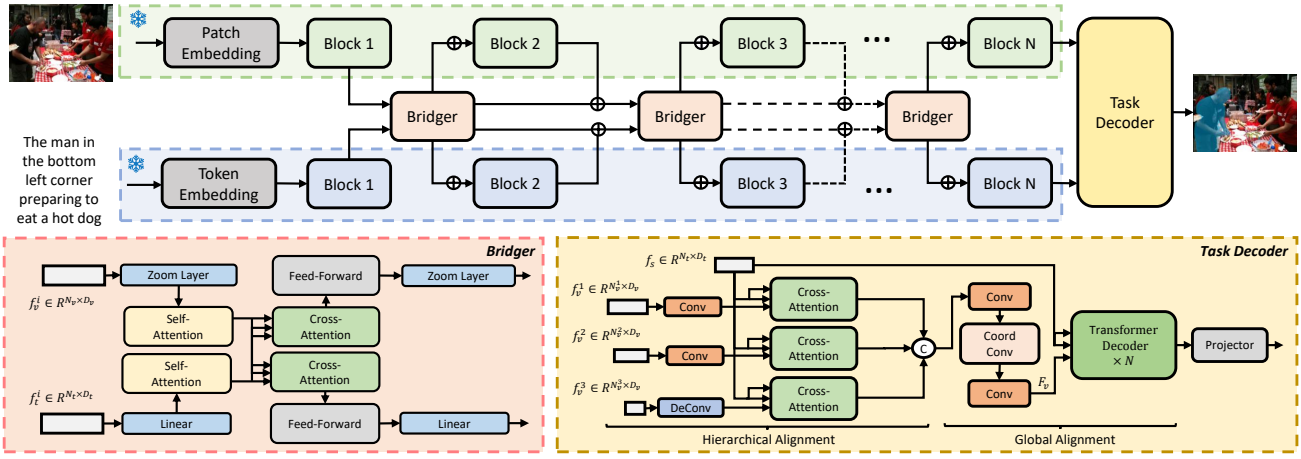

Figure 2: Given an image and a language sentence, our model extracts multiple image features $f_v^1, ..., f_v^N$ from different stages of an image encoder, and word-level features f_t and sentence-level features f_s from a language encoder. The Bridger enables cross-modal interactions at each encoder stage. The Hierarchical Alignment Module fuses hierarchical features with global textual representations to obtain fusion features F_v. The Global Alignment Module combines sentence-level information with fine-grained visual features to produce the representation F_c. Finally, the Projector generates the mask prediction using F_c.

cus on sentence-pixel alignment to leverage multi-modal corresponding information. To better make use of language-related object location information for visual-linguistic interaction, PCAN [2] focuses on position-aware contrastive alignment to enhance the alignment of multi-modal features. Our method focuses on fusing and aligning features from different modalities using a parameter-efficient approach that leverages pre-trained vision-language models. It achieves competitive performance and scalability compared to methods using the same backbone network, while avoiding modification of the backbone network's weight. This reduces the number of parameters to be updated and provides better hardware resource savings.

3. Methodology

In this section, we present the proposed parameter-efficient approach for referring image segmentation. As illustrated in Figure 2, our framework contains four components, i.e., a frozen vision-language backbone (§3.1), a tunable Bridger (§3.2), a task-specific decoder (§3.3), and the learning objective (§3.4). We aim to utilize the powerful pre-trained backbone as the image and text encoders for the downstream task while refraining from modifying its substantial quantity of original parameters.

3.1. Image & Text Feature Extraction

Given an image and a text, we extract their features through the image encoder and text encoder, respectively:
Image Encoder. For an input image $I \in R^{H\times W\times 3}$, we extract visual features from layers of the image encoder. In detail, for CNN (e.g., ResNet [19]), we exploit the visual features from the last $N-1$ stages defined as $F_v^i, i \in \{2, ..., N\}$; For vision Transformer [13] (ViT), we evenly split the transformer encoders of ViT into N blocks, each containing L/N encoder layers. We employ the outputs of the last $N-1$ blocks to make feature interaction. The multi-level visual features from different blocks in ViT or different stages in ResNet will be utilized in our framework as the input of the Bridger and decoder for multi-modal feature alignments.
Text Encoder. For an input referring expression T, a Transformer [42] modified by [40] is used to extract text features. Similar to Image Encoder, we evenly split the transformer encoders into N blocks and extract $F_t^i \in R^{L\times C}, i \in \{2, ..., N\}$ from different blocks of Transformer, where C is the feature dimension, and L is the length of the expression. The Transformer is applied to a lower-cased byte pair encoding representation of the text, and the text sequence is bracketed with the `[SOS]` and `[EOS]` tokens. The activation of the last layer of the Transformer at the `[EOS]` token is further processed to generate the global textual representation $F_s \in R^{C'}$, where C' is the feature dimension.

Since the backbone image encoder and text encoder normally take the majority of the parameters, we freeze them during the tuning procedure in our approach.

3.2. Image & Text Feature Interaction

While the features from the image and text encoders do not "see" each other during the pre-training process, referring image segmentation requires intensive multi-modal interaction to understand the image and text features jointly.

Therefore, we propose a novel vision-and-language interaction module (i.e., Bridger) to process the intermediate features from the image and text encoders. By doing so, the model can learn to fully use the extracted image and text features to enhance the multi-modal interaction.

Briefly, given multiple visual features $F_v^i, i \in \{2, ..., N\}$ and linguistic features $F_t^i \in R^{L \times C}, i \in \{2, ..., N\}$, we firstly adjust the shape of visual and linguistic features via Zoom Layer (ZL). This process can be formalized:

$$\begin{aligned} \hat{F}_v^i &= \text{ZL}_{in}(F_v^i) \\ \hat{F}_t^i &= \text{Linear}(F_t^i) \end{aligned} \tag{1}$$

where the ZL_{in} means the zoom-in operation of the Zoom layer. Afterward, we fuse the features through Interactor (ITA). This process can be mathematically expressed as

$$\begin{aligned} \hat{f}_v^i &= \text{ITA}(\hat{f}_v^{i-1} + \hat{F}_v^i, \hat{f}_t^{i-1} + \hat{F}_t^i) \\ \hat{f}_t^i &= \text{ITA}(\hat{f}_t^{i-1} + \hat{F}_t^i, \hat{f}_v^{i-1} + \hat{F}_v^i) \end{aligned} \tag{2}$$

Finally, we recover the dimension through the zoom layer and linear projection and make a residual connection to the original feature of the next stage (blocks) of the backbone. This process can be expressed mathematically as

$$\begin{aligned} f_v^i &= \text{ZL}_{out}(\hat{f}_v^i) \\ f_t^i &= \text{Linear}(\hat{f}_t^i) \\ F_v^{i+1} &= F_v^{i+1} + f_v^i \\ F_t^{i+1} &= F_t^{i+1} + f_t^i \end{aligned} \tag{3}$$

where the ZL_{out} means the zoom-out operation of the Zoom layer. Next, we describe their architecture in detail:

Zoom Layer (ZL). With the features extracted from the image and text encoders, we design a module to make dimensional changes on visual and linguistic features with consideration to the time complexity and spatial priority. For ViT, the plain design of the architecture suffers inferior performance as a result of lacking vision-specific inductive biases. Recent studies [44, 18] show that convolutions can help transformer to capture the local spatial contexts of images. Therefore, we reshape the feature from middle layers from ViT from $R^{D \times C}$ to $R^{H \times W \times C}$ and use convolution to compose the Zoom layer. Moreover, for ResNet, the feature map from the first two stages can be large when the resolution of the input increases, which will make the length too long to process when using it as the input of the attention algorithm. Therefore, we adopt stride-2 2x2 convolution to reduce the size of feature maps. To unify dimensions and enlarge smaller feature maps, we use stride-2 2x2 deconvolution to enrich information. In short, when extracting the feature from the middle layers of the backbone, we use the Zoom layer to resize the feature map from the visual encoder. The process can be formalized as

$$\hat{F}_v^i = \begin{cases} \text{Conv}\left(F_v^i\right), & h_i >= h', w_i >= w' \\ \text{DeConv}\left(F_v^i\right), & h_i < h', w_i < w' \end{cases} \tag{4}$$

where the h', w' are one of the feature map's height and width from the visual backbone. With $\hat{F}_v^i$, we make interactive operations between feature maps of different modalities. After that, before adding the features back to the backbone, we utilize the Zoom layer to make zoom-out operations, which is the reverse process of zoom-in.

Interactor (ITA). With the features processed by the zoom layer, we design a module to make the interaction of modal information between the visual encoder and text encoder, which enhances the features in the middle of the pre-trained backbone while fixing the original parameters. Specifically, the Interactor is based on an attention mechanism and feed-forward network. For each feature from different modalities, we employ the original modality feature as a query and obtain the keys and values from the other modality. The interaction can be formalized as

$$\begin{aligned} \hat{f}_v^i &= \mathcal{F}_{\text{MHSA}}(\hat{f}_v^{i-1} + \hat{F}_v^i) \\ \hat{f}_t^i &= \mathcal{F}_{\text{MHSA}}(\hat{f}_t^{i-1} + \hat{F}_t^i) \\ \hat{f}_v^i, \hat{f}_t^i &= \mathcal{F}_{\text{MHCA}}(\hat{f}_v^i, \hat{f}_t^i), \mathcal{F}_{\text{MHCA}}(\hat{f}_t^i, \hat{f}_v^i) \\ \hat{f}_v^i, \hat{f}_t^i &= \text{FFN}(\hat{f}_v^i), \text{FFN}(\hat{f}_t^i) \end{aligned} \tag{5}$$

3.3. Task-specific Decoder

Hierarchical Alignment Module. Given multiple visual features $F_v^i, i \in \{2, ..., N\}$ from different stages and the global textual representation F_s, we obtain the fusion of multi-modal feature by convolution and cross-attention mechanism. For hierarchical fusion features, we simply concatenate and use a 1 × 1 convolution layer to aggregate them:

$$\begin{aligned} f_m^i &= \text{Conv}(F_v^i) \\ f_m^i &= \mathcal{F}_{\text{MHCA}}(f_m^i, F_s) \\ F_m &= \text{Conv}\left(\left[f_m^2, ..., f_m^N\right]\right) \end{aligned} \tag{6}$$

where $[,]$ is the concatenation operation, and the convolution is adopted to unify the dimension of features from different stages. Finally, we concatenate a 2D spatial coordinate feature $F_{coord} \in R^{\frac{H}{16} \times \frac{W}{16} \times C}$ with F_m and use a 3 × 3 convolution to fuse them. The visual feature $F_v \in R^{\frac{H}{16} \times \frac{W}{16} \times C}$ is then calculated:

$$F_v = \text{Conv}\left([F_m, F_{coord}]\right) \tag{7}$$

The 2D spatial domain of F_v is flattened into a sequence, forming the visual feature F_v, which is then used in the subsequent process.

Global Alignment Module. With multi-modal features gained from hierarchical alignment, we combine ample textual information corresponding to visual features by using

the attention model of the Transformer. Taking the multi-modal features F_v and the sentence-level feature F_s as input, we firstly add the fixed sine spatial positional encoding to F_v and F_s respectively. Subsequently, a sequence of evolved multi-modal features F_c is generated by self and cross-attention module to capture global contextual information:

$$\begin{aligned} f_c &= \mathcal{F}_{\text{MHSA}}(F_v) \\ f_c &= \mathcal{F}_{\text{MHCA}}(f_c, F_s) \\ F_c &= \text{FFN}(f_c) \end{aligned} \tag{8}$$

where the evolved multi-modal features F_c are finally utilized for the segmentation task.

Projector. To obtain mask prediction on each pixel according to the corresponding semantic information, we use a Projector to make transformation on cross-modal feature F_c and sentence-level feature F_s as Equation 9.

$$\begin{aligned} F_c' &= \text{UpSample}\left(F_c\right) \\ Z_c &= \text{Conv}\left(F_c'\right) \\ Z_t &= \text{Linear}\left(F_s\right) \end{aligned} \tag{9}$$

where UpSample denotes $4\times$ upsampling, and convolution and linear projection are used to transform F_c and F_s into $Z_c \in R^{N\times D}, N = \frac{H}{4} \times \frac{W}{4}$ and $Z_t \in R^C, C = K \times K \times D + 1$. We split and reshape Z_t into weights $\in R^{D\times K\times K}$ and bias $\in R^D$, where K denotes the kernel size of the convolution layer. This enables it to function as a Conv2D layer, which is utilized to convert the cross-modal representation Z_c into the ultimate mask prediction.

3.4. Training Objective

Considering the suboptimality of CLIP [40]'s pre-training strategy for referring image segmentation due to its reliance on aligning the textual representation with the image-level representation, we utilize a text-to-pixel contrastive loss [43] as our training objective, which is employed to optimize the relationship between two modalities. The objective of this contrastive loss is to ensure that Z_t is similar to its corresponding Z_c, while being dissimilar to other irrelevant Z_c. The text-to-pixel contrastive loss can be formulated as follows:

$$L_{\text{con}}\left(Z_t, Z_c\right) = \frac{1}{|\mathcal{P} \cup \mathcal{N}|} \sum_{i \in \mathcal{P} \cup \mathcal{N}} L_{\text{con}}^i\left(Z_t, Z_c^i\right) \tag{10}$$

where $\mathcal{P}$ and $\mathcal{N}$ denote the class of 1 and 0 in the ground truth and L_{con}^i is defined as:

$$L_{\text{con}}^i\left(Z_t, Z_c^i\right) = \begin{cases} -\log\left(\sigma\left(Z_t \cdot Z_c^i\right)\right), & i \in \mathcal{P} \\ -\log\left(1 - \sigma\left(Z_t \cdot Z_c^i\right)\right), & i \in \mathcal{N} \end{cases} \tag{11}$$

where σ is the sigmoid function. The segmentation result is obtained by reshaping $\sigma\left(Z_t \dot{Z}_c\right)$ into $\frac{H}{4} \times \frac{W}{4}$ and then upsampling it back to the original image size.

4. Experiments Setting

4.1. Datasets

In order to assess the efficacy of each component of our method, we have conducted comprehensive experiments on three benchmarks datasets:

- **RefCOCO [28]** is a widely employed benchmark dataset for referring image segmentation. It comprises 19,994 images annotated with 142,210 referring expressions for 50,000 objects, which have been sourced from the MSCOCO [35] dataset through a two-player game. The dataset is divided into four subsets, consisting of 120,624 train, 10,834 validation, 5,657 test A, and 5,095 test B samples, respectively. The average length of the expressions is 3.6 words, and each image contains a minimum of two objects.
- **RefCOCO+ [28]** dataset consists of 141,564 referring expressions associated with 49,856 objects in 19,992 images. The dataset is divided into four subsets: 120,624 train, 10,758 validation, 5,726 test A, and 4,889 test B samples. Notably, the RefCOCO+ dataset has been constructed to be more challenging than the RefCOCO dataset by excluding certain types of absolute-location words.
- **G-Ref [48]** comprises 104,560 referring expressions associated with 54,822 objects in 26,711 images. In contrast to the two datasets described above, the expressions in G-Ref were collected from Amazon Mechanical Turk and had an average length of 8.4 words, which includes more words about locations and appearances. We present results for both the Google and UMD partitioning for G-Ref.

4.2. Implementation Details

We initiate the text and image encoder with CLIP [40], and respectively adopt ResNet-50 [19], ResNet-101 [19], ViT-B [42] as the image encoder for all ablation studies. We opted for CLIP because our work aims to better transfer model state from the source dataset/scenario to the target dataset/scenario via model tuning. For CLIP model, the discrepancy between model states is higher than those of GLIP and MDETR in dense prediction scenarios, which is more challenging for our Bridger design. We resize input images to 416×416, following the setting of CRIS [43]. To accommodate the extra `[SOS]` and `[EOS]` tokens, RefCOCO and RefCOCO+ input sentences are limited to 17 words, while G-Ref supports up to 22 words. Our Transformer Decoder has three layers, each with 8 heads and a feed-forward hidden dimension of 512. The Projector uses a kernel size of 3 for the last convolution layer composed of Z_t. We train the network for 50 epochs using the Adam optimizer with a learning rate of $\lambda = 0.0001$. The learning rate of Bridger is set to $\lambda = 0.001$ for ViT and $\lambda = 0.0001$ for ResNet. We decrease the learning rate by 0.1 at the 35th epoch and train the model with a batch size of 32 on 2 NVIDIA A100

Method	Backbone	Param.	RefCOCO			RefCOCO+			G-Ref		
			val	test A	test B	val	testA	testB	val (u)	test (u)	val (g)
PCAN [2]	ResNet-50	25.56 M	69.51	71.64	64.18	58.25	63.68	48.89	**59.98**	**60.80**	**57.49**
CRIS [43]	ResNet-50	40.42 M	69.52	72.72	64.70	**61.39**	67.10	**52.48**	59.87	60.36	–
ETRIS (Ours)	ResNet-50	1.68 M	**70.39**	**73.11**	**66.38**	60.47	**67.11**	50.73	59.71	59.95	57.22
RMI [36]	DeepLab ResNet-101	61.00 M	45.18	45.69	45.57	29.86	30.48	29.50	–	–	–
RRN [32]	DeepLab ResNet-101	61.00 M	55.33	57.26	53.95	39.75	42.15	36.11	–	–	36.45
MAttNet [47]	MaskRCNN ResNet-101	27.57 M	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	–
CMSA [46]	DeepLab ResNet-101	61.00 M	58.32	60.61	55.09	43.76	47.60	37.89	–	–	39.98
CAC [7]	DeepLab ResNet-101	61.00 M	58.90	61.77	53.81	–	–	–	46.37	46.95	44.32
STEP[3]	DeepLab ResNet-101	61.00 M	60.04	63.46	57.97	48.19	52.33	40.41	–	–	46.40
BRINet [22]	DeepLab ResNet-101	61.00 M	61.35	63.37	59.57	48.57	52.87	42.13	–	–	48.04
CMPC [23]	DeepLab ResNet-101	61.00 M	61.36	64.53	59.64	49.56	53.44	43.23	–	–	–
LSCM [24]	DeepLab ResNet-101	61.00 M	61.47	64.99	59.55	49.34	53.12	43.50	–	–	–
CMPC+ [37]	DeepLab ResNet-101	61.00 M	62.47	65.08	60.82	50.25	54.04	43.47	–	–	49.89
EFN [14]	Wide ResNet-101	126.89 M	62.76	65.69	59.67	51.50	55.24	43.01	–	–	–
BUSNet [45]	DeepLab ResNet-101	61.00 M	63.27	66.41	61.39	51.76	56.87	44.13	–	–	–
CGAN [39]	DeepLab ResNet-101	61.00 M	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	–
CRIS [43]	CLIP ResNet-101	57.31 M	70.47	73.18	66.10	**62.27**	68.08	**53.68**	59.87	60.36	–
ETRIS (Ours)	CLIP ResNet-101	1.94 M	**71.06**	**74.11**	**66.66**	62.23	**68.51**	52.79	**60.28**	**60.42**	**57.86**
ReSTR [29]	ViT-B-16	86.19 M	67.22	69.30	64.45	55.78	60.44	48.27	54.48	-	54.48
ETRIS (Ours)	ViT-B-16	1.39 M	**70.51**	**73.51**	**66.63**	**60.10**	**66.89**	**50.17**	**59.82**	**59.91**	**57.88**

Table 1: Comparison with SOTA method using ResNet and ViT as backbone on the oIoU metric on RIS datasets. Param.: The trainable parameters of the backbone model. u: The UMD partition. g: The Google partition. The best results are in bold.

with 40 GPU VRAM. At inference, the predicted results are upsampled to the original image size and binarized at a threshold of 0.35, producing the final result without any additional post-processing.

To evaluate the effectiveness of our model, we use two metrics commonly used in previous works: Intersection over Union (IoU) and Precision@X. IoU calculates the overlap between the predicted segmentation mask and the ground truth, while Precision@X measures the percentage of test images with an IoU score above a threshold $X \in 0.5, 0.6, 0.7, 0.8, 0.9$. This metric assesses the model's ability to accurately localize objects.

5. Experiments Results

5.1. Main Results

We compare our proposed method with existing RIS methods on the same datasets, reporting the oIoU results in Table 1. To ensure a fair comparison, we categorize these methods based on their visual backbone and report their tunable parameters. Our approach achieves competitive performance on all tasks compared to existing methods using the same backbone, validating the effectiveness of our parameter-efficient approach. Our approach's effectiveness is further amplified with increasing model scale, as observed in our experiments. This parameter-efficient approach is beneficial not only because of the strong representation abilities of pre-trained models but also due to their ability to reduce the risk of overfitting by constraining the number of parameters that require fine-tuning for downstream tasks. The Bridger plays a crucial role in early feature fusion between modalities. Additionally, our method's ability to inject vision-specific inductive biases into the pre-trained backbone reduces the performance gap between ViT-based and ResNet-based approaches. This finding also suggests the low intrinsic dimension of pre-trained models for fine-tuning [1].

Table 2 compares our method with other parameter-efficient methods using oIoU metrics on RefCOCO's val-test split. To ensure a rigorous and equitable comparison, we standardized the reduction factor to 4 to minimize any potential confounding effects arising from differences in this parameter. For CoOp, we set the number of learnable tokens to 8 following [50]. For Conv Adapter, we applied the original method of inserting the adapter into the visual encoder. For Adapter and Compactor, we inserted adapters into both encoders. For LoRA, we incorporated LoRA into the encoders following the primary approach. Our approach achieves a 3.33% improvement in oIoU compared to the method of freezing the weights of the backbone. Furthermore, our method shows an oIoU improvement of $1.60\% \sim 3.19\%$ over other parameter-efficient methods while using a comparable amount of fine-tuned parameters.

To highlight the differences between the previous and new task decoders, and their respective number of tuned parameters, we present the parameter counts in Table 3. These counts were calculated using their open-source code. For "+", we only calculated the parts of the model with known structures, as some modules are not open source. Our method has fewer total tunable parameters, ranging from only 10.75% to 20.15% compared to other methods.

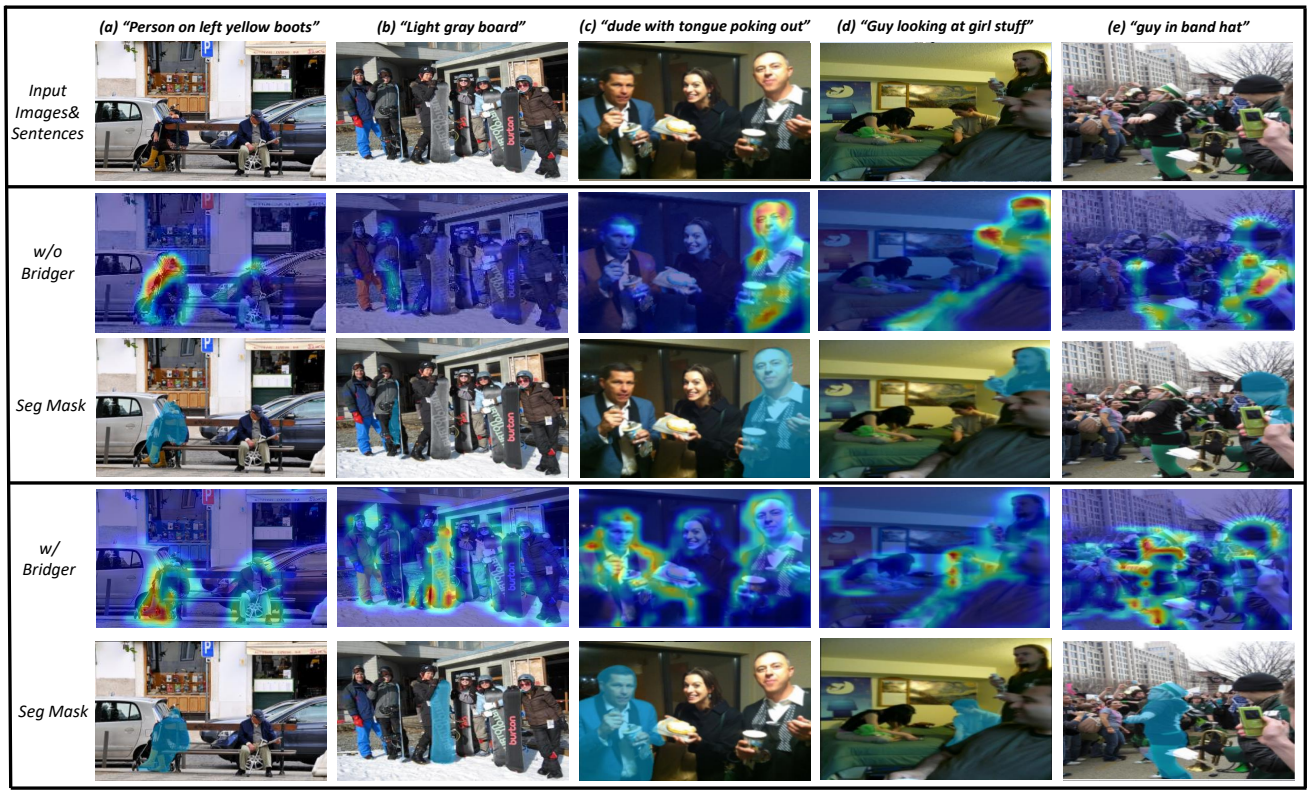

Figure 3: Visualization of the generated feature maps by the input images and the sentences describing objects in the images (first row). Feature maps produced by our framework w Bridger (third row) contain more fine-grained features with rich edges and textures than those generated by our framework w/o Bridger (second row), which is of great help for dense prediction. The fourth row demonstrates the final mask prediction of our framework w Bridger. Figure best viewed in color.

Method	Trainable Parameters			oIoU(%)
	Backbone	Prompt	Head	
Full-Tuning	120.74 M	0.00 K	23.98 M	70.47
Fix Backbone	0.00 M	0.00 K	23.98 M	67.73
Adapter [20]	2.39 M	0.00 K	23.98 M	69.46
Conv Adapter [41]	1.20 M	0.00 K	23.98 M	69.33
Compacter [27]	0.19 M	0.00 K	23.98 M	69.11
CoOp [50]	0.00 M	4.10 K	23.98 M	67.87
LoRA [21]	0.03 M	0.00 K	23.98 M	68.84
ETRIS (Ours)	1.94 M	0.00 K	23.98 M	**71.06**

Table 2: Comparison with previous parameter efficient tuning method using Resnet101 as backbone on the oIoU(%) metric on test-val-split of RefCOCO dataset.

Method	Tunable Param. (Backbone)	Tunable Param. (Decoder)	Tunable Param. (Total)
PCAN [2]	25.56 M (6.57%)	–	150.21+ M (17.08%)
CRIS [43]	40.42 M (4.16%)	42.88 M (50.92%)	146.85 M (17.47%)
ETRIS (Ours)	1.68 M	23.98 M	25.66 M
BRINet [22]	61.00 M (3.18%)	190.68 M (12.58%)	251.68 M (10.30%)
LSCM [24]	61.00 M (3.18%)	80.85 M (29.66%)	141.85 M (18.27%)
CMPC+ [37]	61.00 M (3.18%)	67.66 M (35.44%)	128.66 M (20.15%)
EFN [14]	126.89 M (1.53%)	96.36 M (24.89%)	232.78 M (11.13%)
CRIS [43]	57.31 M (3.39%)	40.66 M (58.98%)	161.25 M (16.07%)
ETRIS (Ours)	1.94 M	23.98 M	25.92 M
ReSTR [29]	86.19 M (1.61%)	–	200.63+ M (12.65%)
ETRIS (Ours)	1.39 M	23.98 M	25.37 M

Table 3: Comparison of parameters with existing methods, where the percentages denote the ratio of the number of tunable parameters of our method to other methods.

5.2. Qualitative Analysis

Figure 3 demonstrates that our method with Bridger generates more detailed features with distinct edges and textures, which is superior to the model without Bridger. Bridger assists the model in better understanding the semantic information from the sentence and making more accurate positional predictions. In case (a), the feature map generated without Bridger is inadequate in accurately pinpointing the location of the boot, causing the model to focus on the person's head. Similarly, in case (b), the model fails to comprehend the location description, resulting in inaccurate predictions. Our model with Bridger generates features that capture fine-grained details and better integrate visual and textual information, as demonstrated in cases (c), (d), and (e).

5.3. Ablation Study

We establish the efficacy of our proposed approach by performing ablation studies on crucial modules. Further information on Bridger's hidden dimension employed can be found in Appendix **??**.

Effect of Bridger's number and position. We studied the effect of the number and position of Bridger using ResNet101 as our backbone and set the scope of fusion as [2,n], [2, n/2], [n/2, n], where n is the layer number of the encoder. The scope of influence of fusion features is defined as the range in which such features have an effect. For example, when fusion is performed at the first stage, the fusion features that occur at the end of the first stage will have an influence that extends to the second to nth stages of the pre-trained backbone. Table 4 shows the results of different numbers of Bridgers under different scopes on RefCOCO's val-test split. The results indicate that Bridger scope expansion improves performance, while the number of Bridgers has little impact.

Scopes	Number	Params	oIoU(%)
$2 \rightarrow n$	1	0.43M	70.96
$n/2 \rightarrow n$	1	0.29M	70.18
$n \rightarrow n$	1	1.22M	69.65
$2 \rightarrow n$	2	0.72M	70.75
$n/2 \rightarrow n$	2	1.51M	70.53
$2 \rightarrow n$	3	1.94M	71.06

Table 4: Ablation study of the Bridger's number and scopes.

Effect of ZL's component. We conducted experiments with various components of the Zoom Layer to analyze the optimal way for the zoom operation. Table 5 shows that using convolutional and deconvolutional layers for zoom-in and out operations yielded the best balance between performance and parameters. These results demonstrate that by utilizing convolution-based operations, we can adjust the size of the feature map to facilitate upcoming attention operations and augment the local information of the feature map.

Effect of Bridger, Hierarchical Alignment Module (HA) and Global Alignment Module (GA). We evaluated the necessity of the proposed modules by separately removing them and reporting the oIoU results on the val-test split of RefCOCO. From Table 6, it can be observed that the performance decreased by 3.33% in the absence of Bridger, 12.36% in the absence of HA and 8.85% in the absence of GA, which demonstrate the effectiveness of Bridger and verifies the ability of the proposed module to improve alignment.

Zoom Layer (x)	Params	oIoU(%)	Pr@0.5	Pr@0.7	Pr@0.9
(a) Linear	5.77 M	70.94	82.89	72.18	**17.89**
(b) Conv&Interpolate	1.45 M	70.08	81.17	68.07	15.95
(c) MLP	1.68 M	70.62	82.36	71.05	17.70
(d) Conv&Deconv	1.94 M	**71.06**	**83.43**	**72.68**	17.40

Table 5: Ablation study of the Component of Zoom Layer.

HA	GA	Bridger	oIoU(%)	Pr@0.5	Pr@0.7	Pr@0.9
	✓	✓	58.70	69.53	45.52	4.33
✓	✓		67.73	79.62	68.47	15.41
✓		✓	62.21	71.51	53.41	11.16
✓	✓	✓	71.06	83.43	72.68	17.40

Table 6: Ablation study of Hierarchical Alignment Module Global, Alignment Module, and Bridge.

6. Discussion

Our method can be beneficial to other tasks such as semantic segmentation or classification due to the model's ability to facilitate early modal fusion and multi-scale feature aggregation. To achieve this, we propose three transformations: (1) Semantic Segmentation by considering the category name as the text, (2) Object Detection by incorporating an FPN network, and (3) Classification by making minor modifications to the decoder. More details can be seen in Appendix **??**.

7. Conclusion

In this paper, we propose a parameter-efficient tuning framework for referring image segmentation. In detail, for injecting vision-specific inductive biases and task-specific information into the pre-trained model while keeping its original parameters fixed, we proposed Bridger to make an interaction between the vision and language encoders. Afterward, we design a lightweight decoder to make further hierarchical and global alignment on visual and linguistic features by combining convolution and attention mechanisms. Our model achieves competitive performance compared to full fine-tuning on three benchmark datasets with the same backbone. Larger pre-trained models improve performance, as shown by comparisons with different visual backbones.

Acknowledgements

This work was supported in part by the Chinese Key-Area Research and Development Program of Guangdong Province (2020B0101350001), in part by the Guangdong Basic and Applied Basic Research Foundation (NO. 2020B1515020048), in part by the National Natural Science Foundation of China (NO. 61976250), in part by the Shenzhen Science and Technology Program (NO. JCYJ20220530141211024, NO. JCYJ20220818103001002), in part by the Fundamental Research Funds for the Central Universities under Grant 22lgqb25 and in part by the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen. This work was also sponsored by Tencent CCF Open Fund (NO. RBFR2022009).

References

[1] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020. 6

[2] Bo Chen, Zhiwei Hu, Zhilong Ji, Jinfeng Bai, and Wangmeng Zuo. Position-aware contrastive alignment for referring image segmentation. *arXiv preprint arXiv:2212.13419*, 2022. 3, 6, 7

[3] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7454–7463, 2019. 6

[4] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022. 1

[5] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 1

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 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. *arXiv preprint arXiv:1910.04748*, 2019. 6

[8] Zhihong Chen, Yuhao Du, Jinpeng Hu, Yang Liu, Guanbin Li, Xiang Wan, and Tsung-Hui Chang. Multi-modal masked autoencoders for medical vision-and-language pre-training. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 679–689. Springer, 2022. 2

[9] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 1

[10] Zhihong Chen, Guanbin Li, and Xiang Wan. Align, reason and learn: Enhancing medical vision-and-language pre-training with knowledge. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5152–5161, 2022. 2

[11] Zhihong Chen, Ruifei Zhang, Yibing Song, Xiang Wan, and Guanbin Li. Advancing visual grounding with scene knowledge: Benchmark and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15039–15049, 2023. 1

[12] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vlt: Vision-language transformer and query generation for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[14] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15506–15515, 2021. 6, 7

[15] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 1

[16] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*, 2020. 1, 2

[17] Tianxiang Hao, Hui Chen, Yuchen Guo, and Guiguang Ding. Consolidator: Mergable adapter with group connections for vision transformer. In *International Conference on Learning Representations*, 2023. 2

[18] Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Camouflaged object detection with feature decomposition and edge reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22046–22055, 2023. 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 3, 5

[20] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 1, 2, 7

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 7

[22] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4424–4433, 2020. 6, 7

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10488–10497, 2020. 6

[24] 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 *European Conference on Computer Vision*, pages 59–75. Springer, 2020. 6, 7

[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

[26] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 2

[27] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021. 2, 7

[28] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, 2014. 5

[29] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. Restr: Convolution-free referring image segmentation using transformers. *arXiv preprint arXiv:2203.16768*, 2022. 6, 7

[30] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2

[31] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021. 1, 2

[32] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2018. 2, 6

[33] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 2

[34] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2

[35] 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 *European conference on Computer Vision*, pages 740–755. Springer, 2014. 5

[36] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1271–1280, 2017. 2, 6

[37] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4761–4775, 2021. 6, 7

[38] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[39] 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 *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1274–1282, 2020. 6

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3, 5

[41] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, 2022. 7

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 3, 5

[43] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. *arXiv preprint arXiv:2111.15174*, 2021. 1, 2, 5, 6, 7

[44] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross B. Girshick. Early convolutions help transformers see better. In *Neural Information Processing Systems*, 2021. 4

[45] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11266–11275, 2021. 6

[46] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10502–10511, 2019. 6

[47] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 6

[48] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 5

[49] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021. 2

[50] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 1, 2, 6, 7