RISCLIP: Referring Image Segmentation Framework using CLIP

Seoyeon Kim **Minguk Kang** **Jaesik Park**

Pohang University of Science and Technology (POSTECH), South Korea
{syeonkim07,mgkang,jaesik.park}@postech.ac.kr

Abstract

Recent advances in computer vision and natural language processing have naturally led to active research in multi-modal tasks, including Referring Image Segmentation (RIS). Recent approaches have advanced the frontier of RIS by impressive margins, but they require an additional pretraining stage on external visual grounding datasets to achieve the state-of-the-art performances. We attempt to break free from this requirement by effectively adapting Contrastive Language-Image Pretraining (CLIP) to RIS. We propose a novel framework that residually adapts frozen CLIP features to RIS with Fusion Adapters and Backbone Adapters. Freezing CLIP preserves the backbone's rich, general image-text alignment knowledge, whilst Fusion Adapters introduce multi-modal communication and Backbone Adapters inject new knowledge useful in solving RIS. Our method reaches a new state of the art on three major RIS benchmarks. We attain such performance without additional pretraining and thereby absolve the necessity of extra training and data preparation. Source code and model weights will be available upon publication.

1 Introduction

Recent progress in computer vision and natural language processing has prompted vigorous exploration in multi-modal tasks, including text-to-image generation [1, 2, 3, 4, 5, 6, 7, 8], text-to-video generation [9, 10, 11, 12], and visual-question-answering [13, 14, 15]. Among them is Referring Image Segmentation (RIS): a multi-modal task that aims to produce a pixel-wise mask of an instance referred to by a natural language expression. The task holds great potential with various applications, such as language-based image editing [16, 17, 18] and human-robot interaction [19].

RIS is a challenging task requiring deep knowledge of visual and linguistic modalities. Thus, conventional methods [20, 21, 22, 23, 24] take benefit of the profound knowledge learned by large-scale pretrained models. They adopt pretrained image and text encoders as backbones, such as ViT [25] trained on ImageNet-21K [26] and BERT [27] trained on Wikipedia and Google's BooksCorpus. Furthermore, since RIS requires joint reasoning of the two modalities, various fusion techniques have been invented [28, 29, 30, 31, 32]. Extracting strong features from pretrained models and employing cross-modal fusion, methods have advanced the frontier of RIS by impressive margins.

However, recent cutting-edge methods [20, 22, 24] require additional pretraining on large-scale image-text data to attain such performances: they adopt bounding box prediction pretraining on external data, demanding extra instance-text alignment supervision. On the other hand, CLIP [33] already holds instance-text alignment knowledge. MaskCLIP [34] recently revealed that CLIP carries primitive but general instance-text correlation with notable zero-shot open vocabulary segmentation results. Thus, we attempt to break free from the need for extra visual grounding pretraining by leveraging the instance-text alignment expertise of CLIP.

We first experiment with MaskCLIP [34] and observe that directly applying CLIP to RIS is not enough to reach state-of-the-art performances: computing pixel-wise similarity maps between CLIP

image and text features results in a mere 23.86 mIoU as shown in the first row of Table 3. We hypothesize the underlying reason as the absence of joint reasoning of image and text, which results in only a rough alignment between the target instance and referring text. Consider Fig. 1. Given the input image of two giraffes and the text referring to "A giraffe looking up while another giraffe next to it looks down", the image features corresponding to the target giraffe are unlikely to perfectly align with the text feature, as the giraffe can be described with numerous other texts like "giraffe on the left", "the taller giraffe", and "giraffe sticking its chin up". Thus, for the target image features to better align with the text feature, they must evolve to be like the text feature, or vice versa, through multi-modal interaction. Therefore, we introduce Fusion Adapters between the CLIP image and text encoders, which communicate and combine the two modalities through cross-attention.

We also attempt to fully benefit from CLIP's original, general knowledge by freezing CLIP and residually adapting its features. CLIP's comprehensive knowledge is particularly beneficial for RIS, which requires the model to locate any instance described by any natural language expression. However, directly finetuning CLIP on RIS may lose general information, and using the frozen features only misses the opportunity to learn new knowledge specific to RIS. Thus, we adopt a compromise of adapting frozen CLIP features with our newly attached Fusion Adapters in a residual manner. In addition, we also introduce Backbone Adapters into the CLIP encoders to residually inject new RIS-specific knowledge throughout the entire feature extraction process.

In summary, we propose a Referring Image Segmentation framework using CLIP—*RISCLIP*—which effectively adapts CLIP features to RIS with cross-modal communication while maintaining the original, rich knowledge with residual adapters. With such an approach, RISCLIP reaches new state-of-the-art results on three major RIS benchmarks. We attain such performance without the additional pretraining required in previous state-of-the-arts and thereby absolve the necessity of extra training and data preparation. Source code and model weights will be available upon publication.

2 Related Work

Referring Image Segmentation. Referring Image Segmentation (RIS) is a multi-modal task of predicting a pixel-wise mask of an object described by a natural language text. The pioneering work [28] extracts image and text features with recurrent LSTMs and a CNN and concatenates them along the channel dimension into multi-modal features. Follow-up works expand on this framework by incorporating recurrent multi-modal interactions [35] along with more fine-grained segmentation with hierarchical visual features [36, 37, 38]. Another line of research focuses on attending to more important words in the referring expression [39, 40, 41], and another proposes effective cross-modal attention modules [29, 30, 31, 32]. Recent methods adopt pretrained transformer encoders to extract image and text features [42, 21], and [43, 23, 44] leverage the encoder transformer layers for multi-modal feature extraction by feeding in multi-modal features. Our work is similar to the last line of approaches but differs in that we do not finetune our CLIP image and text encoders. Such distinction allows our framework to preserve the comprehensive knowledge of CLIP.

Visual Grounding Pretraining. Recent state-of-the-art performances in RIS are achieved by multi-task learning methods that predict both pixel-wise masks and bounding boxes [45, 20, 22, 24]. Nevertheless, [20, 22, 24] require an additional visual grounding pretraining with bounding box annotations to achieve such results. For example, [20] pretrains on 100K images and 4M texts from Visual Genome (VG) [46], and [22] on a combination of datasets which amounts to 174K images and 6.1M expressions. Without additional visual grounding pretraining on these datasets, [20] and [22] experience an IoU drop of 8.88 and 10.1 on the RefCOCOg [47] (UMD [48]) test set, respectively. This suggests that these methods demand extra instance-text alignment supervision for desirable performance. Nonetheless, such visual grounding pretraining is not favorable as it requires extra training and effort to collect, annotate, and store data. Hence, we attempt to eliminate this pretraining stage by adapting CLIP features which already hold instance-text alignment knowledge.

Contrastive Language-Image Pre-training (CLIP). CLIP [33] is well-known for its general image-text alignment capacity. Thanks to extensive contrastive pretraining on large-scale image-text pairs, CLIP carries not only expertise knowledge in both visual and linguistic modalities but also general image-text alignment knowledge. Various multi-modal tasks, including text-to-image generation [3, 4, 7] and visual captioning [49, 50], benefit from CLIP's rich multi-modal alignment. Also, several works attempt to adapt CLIP to dense prediction tasks, such as open vocabulary object

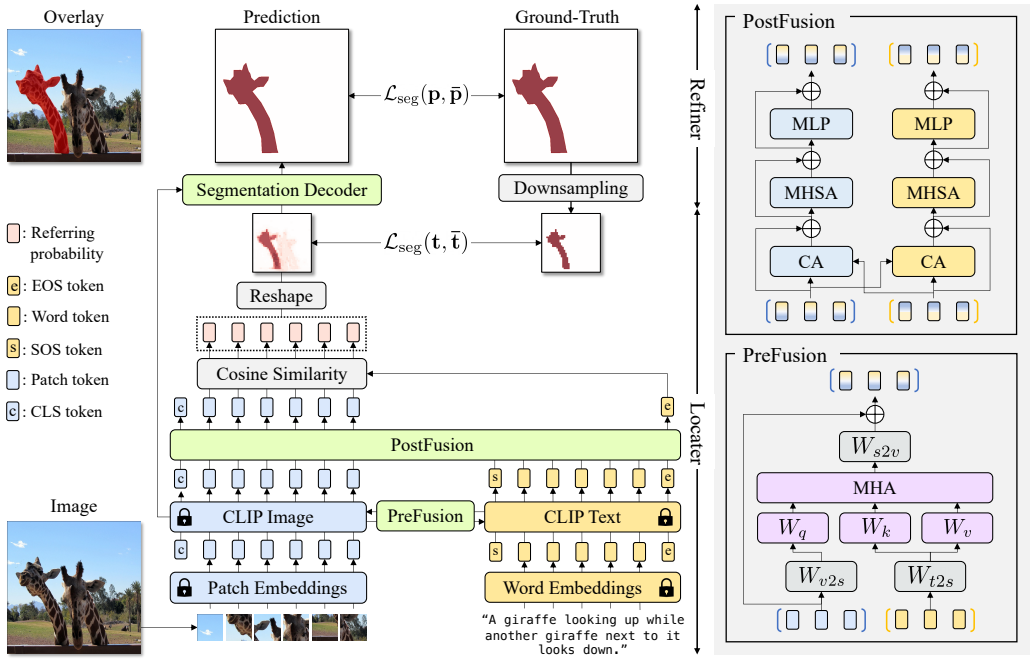

Figure 1: We illustrate the feed-forward process of our framework, RISCLIP. The architecture consists of two parts: the Locator and the Refiner. The Locator takes in an image-text pair and extracts CLIP multi-modal image and text features, which communicate via Pre- and PostFusion Adapters. The patch tokens from the image features and [EOS] token from the text features are computed into a cosine similarity map, sigmoided, and reshaped into a probability map ($\mathbf{t}$) that locates the referred instance. A token-level segmentation loss is applied between $\mathbf{t}$ and the downsampled ground-truth mask, $\bar{\mathbf{t}}$. Then, the Refiner takes in $\mathbf{t}$ and refines the token-level map into a pixel-level map ($\mathbf{p}$) with a segmentation decoder. A pixel-level loss is applied between $\mathbf{p}$ and the ground-truth mask, $\bar{\mathbf{p}}$. The CLIP image and text backbones are frozen.

detection [51, 52, 53] and semantic segmentation [54, 55, 56]. In particular, MaskCLIP [34] unveils CLIP's instance-text alignment capacity by producing zero-shot open vocabulary segmentation maps: a simple cosine similarity map between the patch and [EOS] tokens produces viable results. Such image-text and instance-text alignment is appropriate for solving RIS which locates an instance in an image referred to by a natural language expression. Also, we hypothesize that such knowledge will be able to substitute the instance-text alignment learned by additional pretraining in recent RIS methods. Thus, we adopt CLIP as the backbone of our framework, RISCLIP, and succeed in achieving a new state of the art without additional pretraining.

3 Referring Image Segmentation Framework using CLIP

3.1 Overview

Fig. 1 illustrates the overall pipeline of our method, Referring Image Segmentation framework using CLIP—*RISCLIP*. RISCLIP consists of two components: the Locator and the Refiner. Given an image-text pair, the Locator extracts image and text features and produces a token-level probability map ($\mathbf{t}$) which locates the referred object. Then, the Refiner refines $\mathbf{t}$ with the aid of intermediate visual features ($\mathbf{v}$) from the Locator and produces a final pixel-level probability map ($\mathbf{p}$).

The Locator adopts CLIP image and text encoders as backbones to utilize the instance-text aligned features. Nevertheless, the features alone are not enough to solve RIS since precise alignment between the target instance and text requires feature updates conditioned on the other modality. Hence, we attempt to communicate the features with each other by introducing cross-attention based Fusion Adapters. Also, to learn new knowledge specific to RIS whilst preserving CLIP's general features, we freeze CLIP and adapt the frozen features in a residual manner with Backbone Adapters. In summary, both the Fusion and Backbone Adapters residually adapt frozen CLIP features to RIS, and

we train our newly introduced Adapters only whilst the CLIP backbone remains intact. Meanwhile, the Refiner employs a simple convolutional network to upsample the token-level probability map output from the Locator into a final pixel-wise prediction. We introduce the Locator and Refiner in Section 3.2 and 3.3, respectively.

3.2 Locator

The Locator takes an image and a text and outputs a probability map that locates the referred instance. It can be divided into three parts: the CLIP backbone that provides general image-text aligned features, Backbone Adapters that enrich the CLIP features with specific knowledge required for RIS, and Fusion Adapters that perform cross-attention between the image and text features for cross-modal conditioning. Each component is explained successively below.

3.2.1 CLIP Backbone

We adopt CLIP image and text encoders as backbones to extract rich, general features carrying instance-text alignment knowledge. We explain the feature extraction process below.

Both the CLIP image and text encoders consist of repeated transformer layers, a final layer normalization [57], and a linear projection layer to a shared image-text embedding space. Each transformer layer consists of two submodules: the multi-head self-attention (MHSA) and the multilayer perceptron (MLP), with each submodule preceded by layer normalization (LN). The feedforward process of the i-th transformer layer can be written as below where $\mathbf{f}_i \in \mathbb{R}^{N \times C}$ denotes the i-th transformer layer's output, N the number of tokens in either the image or text feature, and C the channel dimension:

$$\bar{\mathbf{f}}_i = \mathrm{MHSA}(\mathrm{LN}(\mathbf{f}_{i-1})) + \mathbf{f}_{i-1}, \qquad i = 1, ..., L \tag{1}$$

$$\mathbf{f}_i = \mathrm{MLP}(\mathrm{LN}(\bar{\mathbf{f}}_i)) + \bar{\mathbf{f}}_i. \tag{2}$$

Image Encoder. The image encoder extracts image features, $\mathbf{v}$, from an image. First, the image is divided into a sequence of patches, which are flattened and transformed into embeddings through a linear projection layer. Then, a learnable [CLS] embedding is concatenated at the front of the patch embeddings, resulting in N_{visual} visual tokens. Afterwards, positional embeddings are added, and a layer normalization is applied. Finally, the sequence of tokens is passed through the transformer, final layer normalization, and linear projection to a shared image-text embedding space with dimension d as explained above. The final image features are a sequence of the [CLS] and patch tokens, $\mathbf{v} = \mathrm{Proj}(\mathrm{LN}(\mathbf{f}_L^v)) = [\mathbf{v}_{\text{cls}}, \mathbf{v}_{\text{patch}}]$, $\mathbf{v}_{\text{cls}} \in \mathbb{R}^{1 \times d}$ and $\mathbf{v}_{\text{patch}} \in \mathbb{R}^{(N_{\text{visual}}-1) \times d}$. We use a superscript v to indicate that the feature $\mathbf{f}_L$ is from the image encoder.

Text Encoder. The text encoder computes text features, $\mathbf{t}$, from a referring expression. First, the text is transformed into a sequence of word embeddings using lower-cased byte pair encoding (BPE) representation [58]. The word embeddings are encased with a [SOS] and [EOS] token, producing a sequence of length N_{text}. These tokens are summed with positional embeddings and passed through the transformer, final layer normalization, and shared image-text embedding space projection as in the image encoder. The final text features are a sequence of [SOS], words, and [EOS] tokens, $\mathbf{t} = \mathrm{Proj}(\mathrm{LN}(\mathbf{f}_L^t)) = [\mathbf{t}_{\text{sos}}, \mathbf{t}_{\text{words}}, \mathbf{t}_{\text{eos}}]$, where $\mathbf{t}_{\text{sos}}, \mathbf{t}_{\text{eos}} \in \mathbb{R}^{1 \times d}$ and $\mathbf{t}_{\text{words}} \in \mathbb{R}^{(N_{\text{text}}-2) \times d}$. The [EOS] token, $\mathbf{t}_{\text{eos}}$, is treated as the global representation of the text.

Probability Map. Since CLIP image and text features are aligned, a simple cosine similarity map between the patch tokens, $\mathbf{v}_{\text{patch}}$, and the [EOS] token, $\mathbf{t}_{\text{eot}}$, can be interpreted as a detection map locating the referred instance. This detection map is transformed into a probability map through a sigmoid function. Following MaskCLIP [34], we extract value tokens from the image encoder's last transformer layer, pass them through the subsequent LN and MLP, and adopt them as patch tokens to compute probability maps. Such simple process with a decoder attached provides a decent mIoU of 23.86 on the RefCOCOg [47] (UMD split [48]) test set, proving that CLIP enjoys instance-text alignment. Nevertheless, this is far from the state of the arts, suggesting that the frozen CLIP tokens need to be adapted to RIS. To achieve this, we introduce Backbone Adapters and Fusion Adapters.

3.2.2 Backbone Adapters

Thanks to extensive pretraining on large-scale image-text data, CLIP carries rich, comprehensive knowledge. Such general knowledge is particularly useful for RIS, which requires the model to locate

any instance described by any natural language expression. However, finetuning the backbone on the downstream task can lose such general knowledge [59, 60], especially since RIS has small datasets (RefCOCOg [61] amounts to a mere 27K images). On the other hand, freezing the backbone loses the opportunity to learn new knowledge specific to RIS and the downstream dataset.

Inspired by [62, 63], we combat this dilemma by freezing CLIP and attaching BackBone Adapters. The Backbone Adapters are attached in a residual manner so that their newly learned features are summed to the original CLIP features. Freezing the backbone conserves CLIP's comprehensive knowledge, whilst employing Backbone Adapters further enriches the features with new information essential for RIS. We adopt the adapter architecture from [62, 63], which consists of a down-projection linear layer that reduces the channel dimension, a non-linear activation, and an up-projection linear layer that restores the channel dimension. We add these simple structures in a residual manner after the MHSA and MLP modules in the transformer layers:

$$\bar{\mathbf{f}}_i^{'} = \mathrm{AD}_{\mathrm{MHSA}}\Big(\underbrace{\mathrm{MHSA}(\mathrm{LN}(\mathbf{f}_{i-1})) + \mathbf{f}_{i-1}}_{\text{Eq (1)}}\Big) + \underbrace{\mathrm{MHSA}(\mathrm{LN}(\mathbf{f}_{i-1})) + \mathbf{f}_{i-1}}_{\text{Eq (1)}}, \tag{3}$$

$$\mathbf{f}_i = \mathrm{AD}_{\mathrm{MLP}}\Big(\underbrace{\mathrm{MLP}(\mathrm{LN}(\bar{\mathbf{f}}_i^{'})) + \bar{\mathbf{f}}_i^{'}}_{\text{Eq (2)}}\Big) + \underbrace{\mathrm{MLP}(\mathrm{LN}(\bar{\mathbf{f}}_i^{'})) + \bar{\mathbf{f}}_i^{'}}_{\text{Eq (2)}}, \tag{4}$$

where $\mathrm{AD}_{\mathrm{MHSA}}$ and $\mathrm{AD}_{\mathrm{MLP}}$ denote the Backbone Adapters attached after MHSA and MLP in the i-th transformer layer.

3.2.3 Fusion Adapters

Backbone Adapters alone are insufficient to solve RIS since cross-modal interaction between the image and text features is missing. Without cross-modal conditioning, the target patch tokens and [EOS] token can only align up to a certain degree since the instance can be described by various other texts. To better align with the [EOS] token, the patch tokens should "communicate" with the text features and evolve to be like the [EOS] token, and vice versa. To achieve this, we introduce Fusion Adapters that allow the image and text features to communicate through cross-attention.

Cross-modal Fusion can be performed during and after the backbone feature extraction: we can fuse the intermediate image and text features within CLIP or the output features after CLIP. We experimentally find that fusing both intermediate and output features yields the best performance. We name the fusion modules PreFusion and PostFusion Adapters, respectively. Whilst PostFusion Adapters consist of Cross-Attention (CA), MHSA, and MLP, PreFusion Adapters consist of CA only. CA alone is enough in PreFusion because it is placed within CLIP such that the output multi-modal features are re-fed into CLIP and processed by the backbone's successive MHSA and MLP modules. Also, LN is applied before every shared space projection, CA, MHSA, and MLP in both Pre- and PostFusion, although we do not include LN in the equations below for simplicity.

PreFusion Adapters. The PreFusion Adapters are attached between the CLIP image and text encoders to fuse intermediate image and text features. Starting from the deepest layers of the backbone, we pair an image and text encoder layer and attach a single PreFusion Adapter in between.

Consider an Adapter between the n-th image and m-th text encoder layer. First, the Adapter projects the input image and text features, $\mathbf{f}_{n-1}^{v}$ and $\mathbf{f}_{m-1}^{t}$, to a shared image-text embedding space with linear projections, W_{v2s} and W_{t2s}. Then, two separate cross-attention modules produce visual and text multi-modal features, $\mathbf{m}_{n-1}^{v}$ and $\mathbf{m}_{m-1}^{t}$, where each modality is set as query and the other key and value in the multi-head attention (MHA). Lastly, the multi-modal features are projected from the shared image-text embedding space back to each modalities' space with linear projections, W_{s2v} and W_{s2t} to produce the final multi-modal features $\mathbf{m}_{n-1}^{v'}$ and $\mathbf{m}_{m-1}^{t'}$. We elaborate the process to output $\mathbf{m}_{n-1}^{v'}$ below, where $\mathbf{m}_{m-1}^{t'}$ can be computed in vice versa:

$$\mathbf{s}_{n-1}^{v} = W_{v2s}\mathbf{f}_{n-1}^{v}, \quad \mathbf{s}_{m-1}^{t} = W_{t2s}\mathbf{f}_{m-1}^{t}, \tag{5}$$

$$\mathbf{q}^{v} = W_{q}\mathbf{s}_{n-1}^{v}, \quad \mathbf{k}^{t} = W_{k}\mathbf{s}_{m-1}^{t}, \quad \mathbf{v}^{t} = W_{v}\mathbf{s}_{m-1}^{t}, \tag{6}$$

$$\mathbf{m}_{n-1}^{v} = \mathrm{MHA}(\mathbf{q}^{v}, \mathbf{k}^{t}, \mathbf{v}^{t}), \tag{7}$$

$$\mathbf{m}_{n-1}^{v'} = W_{s2v}\mathbf{m}_{n-1}^{v}. \tag{8}$$

These multi-modal features, $\mathbf{m}_{n-1}^{v'}$ and $\mathbf{m}_{m-1}^{t'}$, are added back to the input features as $\mathbf{f}_{n-1}^{v} = \mathbf{f}_{n-1}^{v} + \mathbf{m}_{n-1}^{v'}$ and $\mathbf{f}_{m-1}^{t} = \mathbf{f}_{m-1}^{t} + \mathbf{m}_{m-1}^{t'}$, to inject multi-modal information into the backbone CLIP features. Then $\mathbf{f}_{n-1}^{v}$ and $\mathbf{f}_{m-1}^{t}$ are fed into the n-th image and m-th text encoder layers for further processing by the subsequent MHSA and MLP modules, as written in Eqs. (3) and (4).

PostFusion Adapters. PostFusion Adapters are attached behind CLIP to fuse the extracted image and text features, $\mathbf{v}$ and $\mathbf{t}$. These features already reside in the same space as they have been projected by the final shared image-text embedding projections (Proj) in the backbone feature extraction process. Thus, PostFusion does not require linear projections to a shared space. PostFusion applies CA as explained in PreFusion (without the shared space linear projections), MHSA, and MLP in order. The feed-forward process of the Pre- and PostFusion Adapters are illustrated in Fig. 1.

The final patch and [EOS] tokens output by PostFusion are computed into a cosine similarity map and then sigmoided into a token-level probability map ($\mathbf{t}$), which locates the referred instance.

3.3 Refiner

Since the probability map ($\mathbf{t}$) is computed between tokens, $\mathbf{t}$ is at token level and should be restored to pixel-level to produce a fine-grained prediction ($\mathbf{p}$). We introduce a Refiner to upsample the probability map to the input image resolution with the aid of intermediate visual features from the CLIP image backbone.

Since the role of our Refiner is to simply figure out the boundary of the referred instance given a probability map, we can adopt a light decoder from [23] as our Refiner. Experiments show that the simple decoder consisting of repeated 3x3 convolutions, ReLU [64], and batch normalization [65] is enough to do the job. Nevertheless, our Refiner is model-agnostic and can be replaced with any segmentation model, such as FPN [66] and UPerNet [67].

The Refiner takes as input the probability map concatenated with an intermediate visual feature map from the CLIP image encoder along the channel dimension. Then, the Refiner residually connects shallower intermediate visual image features to use as an aid in successively upsampling and refining the decoded feature maps. A final linear projection transforms the feature maps into background and foreground score maps, which are sigmoided into the final pixel-wise map ($\mathbf{p}$). The binary prediction mask is obtained via argmax during inference.

3.4 Loss Functions

The Locator and Refiner are trained separately in two stages with the same loss at different resolutions (token-level vs. pixel-level). The Locator's output probability map, $\mathbf{t}$ is trained to converge to the token-level downsampled ground truth mask, $\bar{\mathbf{t}}$, whilst the Refiner's output prediction mask, $\mathbf{p}$, is trained to conform to the pixel-level ground truth mask, $\bar{\mathbf{p}}$. Following [20], we adopt a linear combination of DICE/F-1 loss [68] and focal loss [69]:

$$\mathcal{L}_{\text{seg}}(\mathbf{t}, \bar{\mathbf{t}}) = \lambda_{\text{focal}}\mathcal{L}_{\text{focal}}(\mathbf{t}, \bar{\mathbf{t}}) + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}(\mathbf{t}, \bar{\mathbf{t}}), \tag{9}$$

$$\mathcal{L}_{\text{seg}}(\mathbf{p}, \bar{\mathbf{p}}) = \lambda_{\text{focal}}\mathcal{L}_{\text{focal}}(\mathbf{p}, \bar{\mathbf{p}}) + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}(\mathbf{p}, \bar{\mathbf{p}}), \tag{10}$$

where λ_{focal} and λ_{dice} are hyperparameters. In the first stage, the Locator only is trained, and in the second stage, the Refiner only is trained for a single epoch.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We evaluate RISCLIP on three major RIS datasets: RefCOCO [61], RefCOCO+ [61], and RefCOCOg [47], UMD split [48]. The RefCOCO family originates from the same MSCOCO [70] dataset and thus shares images but possesses different texts. RefCOCO [61] and RefCOCO+ [61] texts are relatively concise, consisting of 3.6 words and 1.6 nouns on average. RefCOCO+ [61] differs from RefCOCO [61] in that the texts do not include absolute positional information, such as first, second, left, and right, is thus more difficult. Lastly, RefCOCOg [47] comprises of longer, more complex texts (8.4 words and 2.8 nouns per text) and is thus the most challenging.

Table 1: We compare RISCLIP with previous methods on the RefCOCO family [61, 47, 48]. RISCLIP-B attains a new state of the art, and RISCLIP-L extends the frontier even further. RN101 is ResNet-101 [72], DN53 Darknet-53 [73], and WRN101 Wide ResNet-101 [74]. CLIP-B and CLIP-L denote the Transformer-based CLIP backbones which adopt ViT-B and -L [71] as the image encoder, respectively, and a 12-layer transformer as the text encoder. Differenly, CLIP-L* with the asterisk denotes the ResNet-based CLIP backbone which replaces the image encoder with ResNet-101 [72].

Method	Image Encoder	Text Encoder	RefCOCO			RefCOCO+			RefCOCOg	
			Val	Test A	Test B	Val	Test A	Test B	Val	Test
oIoU										
BRINet [31]	RN101	LSTM	60.98	62.99	59.21	48.17	52.32	42.11	-	-
CMPC [75]	RN101	LSTM	61.36	64.53	59.64	49.56	53.44	43.23	-	-
LSCM [32]	RN101	LSTM	61.47	64.99	59.55	49.34	53.12	43.50	-	-
CMPC+ [41]	RN101	LSTM	62.47	65.08	60.82	50.25	54.04	43.47	-	-
MCN [76]	DN53	Bi-GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40
BUSNet [77]	RN101	Self-Attn	63.27	66.41	61.39	51.76	56.87	44.13	-	-
CGAN [78]	DN53	Bi-GRU	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69
LTS [79]	DN53	Bi-GRU	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25
ReSTR [42]	ViT-B	TX	67.22	69.30	64.45	55.78	60.44	48.27	-	-
LAVT [23]	Swin-B	BERT	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09
RISCLIP-B	CLIP-B	CLIP-B	74.82	76.64	71.06	67.64	72.89	59.33	67.76	69.05
RISCLIP-L	CLIP-L	CLIP-L	**76.92**	**80.99**	**73.04**	**69.33**	**74.56**	**61.87**	**69.20**	**70.19**
mIoU										
VLT [80]	DN53	Bi-GRU	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65
CRIS [21]	CLIP-L*	CLIP-L*	70.47	73.18	66.10	62.27	68.06	53.68	59.87	60.36
SeqTR [22]	DN53	Bi-GRU	71.70	73.31	69.82	63.04	66.73	58.97	64.69	65.74
RefTR [20]	RN101	BERT	74.34	76.77	70.87	66.75	70.58	59.40	66.63	67.39
LAVT [23]	Swin-B	BERT	74.46	76.89	70.94	65.81	70.97	59.23	63.34	63.62
RISCLIP-B	CLIP-B	CLIP-B	75.68	78.01	72.46	69.16	73.53	60.68	67.62	67.97
RISCLIP-L	CLIP-L	CLIP-L	**78.87**	**81.46**	**75.41**	**74.38**	**78.77**	**66.84**	**71.82**	**71.65**

Table 2: Comparison between RISCLIP and PolyFormer [24] with training on a combined RefCOCO dataset [61, 47, 48]. RISCLIP attains comparable performance with CLIP-B as the backbone and outperforms PolyFormer [24] with CLIP-L.

Method	Image Encoder	Text Encoder	RefCOCO			RefCOCO+			RefCOCOg	
			Val	Test A	Test B	Val	Test A	Test B	Val	Test
PolyFormer-B [24]	Swin-B	BERT	**75.96**	77.09	**73.22**	70.65	**74.51**	**64.64**	69.36	**69.88**
RISCLIP-B	CLIP-B	CLIP-B	75.68	**78.01**	72.46	**72.46**	74.30	61.37	**69.49**	69.53
PolyFormer-L [24]	Swin-L	BERT	76.94	78.49	74.83	72.15	75.71	66.73	71.15	71.17
RISCLIP-L	CLIP-L	CLIP-L	**79.53**	**81.78**	**75.78**	**74.88**	**78.88**	**68.09**	**73.45**	**74.52**

Evaluation Metrics. We employ two metrics widely used in RIS: the overall intersection-over-union (oIoU) and the mean intersection-over-union (mIoU). The oIoU is the sum of all intersections over the sum of all unions, whilst the mIoU is the average of intersection over unions. The mIoU is a fairer metric than the oIoU, which is biased towards large objects [23]. Hence, we report both oIoU and mIoUs but adopt mIoUs when comparing with previous methods.

4.2 Model Settings

To explore the effect of the CLIP backbone size, we experiment with two backbones trained with ViT-B and ViT-L [71] and dub our framework RISCLIP-B and -L, respectively. In RISCLIP-B, we use ViT-B [71] with patch size 16×16 as the image encoder and a 12-layer transformer as the text encoder. In RISCLIP-L, we use ViT-L [71] with patch size 14×14 and the same 12-layer transformer as in RISCLIP-B. For both RISCLIP-B and -L, we attach Backbone Adapters in all layers of both encoders, six PreFusion Adapters, and six PostFusion Adapters.

4.3 Comparison with State of the Arts

We compare RISCLIP with previous methods on the three aforementioned datasets. We include both oIoU and mIoU for LAVT [23] which reports both metrics and include either metric for other methods that report only one. As summarized in Table 1, RISCLIP-B outperforms all state-of-the-art methods, and RISCLIP-L further extends the margins. We first compare RISCLIP-B with second-place models that have similar backbone sizes to RISCLIP-B. On RefCOCO [61], RISCLIP surpasses LAVT [23] by 1.22, 1.12, and 1.52 mIoU points on the Val, TestA, and TestB splits, respectively. On the more challenging RefCOCO+ [61], RISCLIP outperforms RefTR [20] by 2.41, 2.95, and 1.28, respectively. Lastly, on the most demanding RefCOCOg [47], RISCLIP exceeds RefTR [20] by 0.99 and 0.58 on the val and test splits. Such performance improvement across all three datasets demonstrates the competency of RISCLIP.

In addition, we observe a significant boost in performance with RISCLIP-L which adopts a larger image encoder. RISCLIP-L further extends the margins set by RISCLIP-B by an average of 3.20, 5.54, and 3.94 on RefCOCO [61], RefCOCO+ [61], and RefCOCOg [47] (UMD [48]), respectively. The performance increase is significant on the harder RefCOCO+ [61] and RefCOCOg [47] (UMD [48]) datasets, indicating that RISCLIP-L effectively leverages the additional computational power and knowledge provided by the larger backbone to solve the more challenging problems. In overall, RISCLIP-L advances the frontier of RIS: On RefCOCO [61], our model surpasses LAVT [23] by 4.41, 4.57, and 4.47. Moreover, we outperform RefTR [20] by 7.63, 8.19, and 7.44 on RefCOCO+ [61] and 5.19 and 4.26 points on RefCOCOg [47] (UMD [48]) on the corresponding test splits, respectively.

We compare RISCLIP-L to CRIS [21] which also adopts CLIP as backbone. Different from RISCLIP-L which uses ViT-L [71] as the image encoder, CRIS [21] uses ResNet-101 [72] instead. RISCLIP surpasses CRIS [21] by an average mIoU gain of 8.66, 11.99, and 11.62 on the three datasets, respectively. Such performance difference shows that RISCLIP utilizes CLIP effectively.

Also, we compare RISCLIP with PolyFormer [24] in a separate Table 2, since PolyFormer [24] adopts a different training scheme from conventional methods. PolyFormer [24] trains on the combined RefCOCO family [61, 47], while the conventional way is to train on each dataset separately. We also train RISCLIP on the combined dataset following PolyFormer [24] for fair comparison. RISCLIP-B attains comparable performance to PolyFormer-B [24], but, when using bigger backbones, RISCLIP-L outperforms PolyFormer-L [24] by an average of 2.28, 2.42, and 2.83 mIoU points on the three datasets. In summary, RISCLIP achieves a new state of the art.

4.4 Ablation Studies

We conduct ablation studies on the test set of RefCOCOg [47] (UMD [48]) to prove the effectiveness of our framework and verify architectural hyperparameter settings.

Table 3: We conduct an ablation experiment on the RefCOCOg [47] (UMD [48]) test set to verify our design choice of residually adapting frozen CLIP features with Fusion and Backbone Adapters. We freeze CLIP and successively add Fusion and Backbone Adapters, where each module increases performance. This suggests that our newly introduced modules effectively adapts CLIP to RIS. However, finetuning CLIP with the new modules shows worse performance, indicating that finetuning CLIP loses useful features.

RISCLIP-B (VIT-B/16)	Fusion Adaptors	Backbone Adaptors	mIoU	oIoU
Frozen	✗	✗	23.86	33.13
Frozen	✓	✗	57.85	58.09
Frozen	✓	✓	62.64	62.02
Fine-tuned	✓	✓	57.88	55.75

Table 4: We conduct ablation experiments on the RefCOCOg [47] (UMD [48]) test set, where the asterisk denotes the same baseline model with 12 Backbone Adapters in each encoder and six Pre- and PostFusion Adapters.

	Prec@0.5	Prec@0.7	Prec@0.9	mIoU	oIoU
a) Backbone Adapters attached to N last CLIP encoder layers					
3	71.81	55.3	11.29	61.40	60.84
6	72.73	56.53	11.87	62.15	61.33
9	72.50	57.44	14.03	62.31	60.68
12*	73.19	57.68	14.21	62.64	62.02
b) PreFusion Adapters attached to N last CLIP encoder layers					
2	72.56	57.05	14.16	62.33	61.34
4	72.33	57.31	13.89	62.41	61.47
6*	73.19	57.68	14.21	62.64	62.02
c) PostFusion Adapters of N layers attached behind CLIP encoders					
2	72.17	56.72	14.23	62.30	61.56
4	72.73	57.68	14.57	62.79	61.95
6*	73.19	57.68	14.21	62.64	62.02

Residually Adapting Frozen CLIP Features. We validate our framework design choice of adapting frozen CLIP features by comparing four scenarios in Table 3. These are 1) using the frozen CLIP

Figure 2: We visualize RISCLIP predictions on RefCOCOg [47] (UMD [48]) test set samples. Row a) demonstrates RISCLIP's comprehensive understanding of various instances, row b) RISCLIP's ability to detect partial, blurry instances and differentiate similar objects, row c) RISCLIP's comprehensive multi-modal understanding that discerns the target instance among resembling instances described with lengthy texts.

backbone only, 2) attaching Fusion Adapters, 3) further attaching Backbone Adapters, and 4) unfreezing CLIP and training it along with Adapters. Introducing Fusion Adapters into the frozen CLIP backbone boosts performance by an mIoU/oIoU average of 29.5, proving that inducing multi-modal interaction into CLIP is an appropriate approach to RIS. Moreover, attaching Backbone Adapters further improves performance by an average of 4.36, indicating that Backbone Adapters learn additional knowledge useful in RIS. Lastly, finetuning CLIP along with the Adapters performs worse than its frozen CLIP twin, with an average IoU drop of 5.52. This suggests that finetuning CLIP can lose general information helpful in RIS. Thus, our design choice of residually adapting frozen CLIP features with Backbone and Fusion Adapters is a viable approach.

Increasingly Attaching Adapters. We investigate the effect of Adapters by varying their numbers in a baseline model: RISCLIP-B with 12 Backbone Adapters in each CLIP image and text encoder, six PreFusion Adapters, and six PostFusion Adapters. The results are summarized in Table 4. Section a) shows that performance improves with the number of Backbone Adapters attached to the latter CLIP encoder layers. Such a trend suggests that Backbone Adapters can inject useful information at all layers, and thus adopting Backbone Adapters throughout the entire feature extraction process is most beneficial. In section b), performance increases with the number of PreFusion Adapters, indicating that using more cross-modal interaction is advantageous. Nevertheless, in section c), the performance plateaus from 4 to 6 PostFusion Adapters, suggesting that there is a limit to the benefits that PostFusion Adapters can bring.

4.5 Visualizations

We visualize the predictions of RISCLIP-B on the RefCOCOg [47] (UMD split [48]) test set. Fig. 2 shows our model's ability to capture a wide variety of instances, detect partially visible or blurry targets, and differentiate the groundtruth from resemblances, even with complicated expressions.

5 Conclusion

RISCLIP effectively adapts CLIP to RIS, resulting in new state-of-the-art results on three major RIS benchmarks. Residually adapting frozen CLIP features with Backbone and Fusion Adapters, we fully benefit from CLIP's rich, comprehensive instance-text alignment knowledge whilst leveraging multi-modal communication and new knowledge essential to RIS. With this approach, we avoid the need for additional visual grounding pretraining required in previous state-of-the-art methods.

Limitations. We can improve our work by adopting other image-text alignment backbones such as ALIGN [81] and Florence [82]. Such extension would allow us to investigate the effectiveness of residually adapting frozen image-text aligned features across various foundation models. Also, while RISCLIP achieves state-of-the-art results with impressive margins, there are complex cases where our framework struggles to accurately identify the target instance. We include these cases in Appendix.

Acknowledgments and Disclosure of Funding

This work was supported by the IITP grants (No.2019-0-01906: AI Graduate School Program - POSTECH, No.2021-0-00537: Visual Common Sense, No.2021-0-02068: AI Innovation Hub) funded by Ministry of Science and ICT, Korea.

References

[1] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021.

[2] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning (ICML)*, 2022.

[3] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 2022.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[6] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[7] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for Text-to-Image Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. StyleGAN-T: Unlocking the power of GANs for fast large-scale text-to-image synthesis. In *International Conference on Machine Learning (ICML)*, 2023.

[9] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[10] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

[11] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-Video Generation without Text-Video Data. In *International Conference on Learning Representations (ICLR)*, 2023.

[12] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. *arXiv preprint arXiv:2304.08818*, 2023.

[13] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. REVIVE: Regional Visual Representation Matters in Knowledge-Based Visual Question Answering. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[14] Matthew Walmer, Karan Sikka, Indranil Sur, Abhinav Shrivastava, and Susmit Jha. Dual-key multimodal backdoors for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] Yang Ding, Jing Yu, Bang Liu, Yue Hu, Mingxin Cui, and Qi Wu. MuKEA: Multimodal Knowledge Extraction and Accumulation for Knowledge-based Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[16] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-Based Image Editing with Recurrent Attentive Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023.

[18] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[19] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[21] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. CRIS: Clip-driven referring image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[22] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. SeqTR: A Simple Yet Universal Network for Visual Grounding. In *European Conference on Computer Vision (ECCV)*, 2022.

[23] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. PolyFormer: Referring Image Segmentation as Sequential Polygon Generation. *arXiv preprint arXiv:2302.07387*, 2023.

[25] Alexey Dosovitskiy et. al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[26] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, 2019.*, 2019.

[28] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision (ECCV)*, 2016.

[29] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[31] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] 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 (ECCV)*, 2020.

[33] 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 *International Conference on Machine Learning (ICML)*, 2021.

[34] Xiaoyi Dong, Yinglin Zheng, Jianmin Bao, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[35] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[36] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *European Conference on Computer Vision (ECCV)*, 2018.

[38] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[39] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[41] 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 (TPAMI)*, 2022.

[42] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. ReSTR: Convolution-free Referring Image Segmentation Using Transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[43] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder Fusion Network with Co-Attention Embedding for Referring Image Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[44] Zicheng Zhang, Yi Zhu, Jianzhuang Liu, Xiaodan Liang, and Wei Ke. Coupalign: Coupling word-pixel with sentence-mask alignments for referring image segmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[45] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision (IJCV)*, 2017.

[47] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[48] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling Context Between Objects for Referring Expression Understanding. In *European Conference on Computer Vision (ECCV)*, 2016.

[49] Ron Mokady, Amir Hertz, and Amit H Bermano. ClipCap: CLIP Prefix for Image Captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[50] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[51] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[52] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. PromptDet: Towards Open-vocabulary Detection using Uncurated Images. In *European Conference on Computer Vision (ECCV)*, 2022.

[53] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the Gap between Object and Image-level Representations for Open-Vocabulary Detection. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[54] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *arXiv preprint arXiv:2211.14813*, 2022.

[55] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A Simple Baseline for Open Vocabulary Semantic Segmentation with Pre-trained Vision-language Model. In *European Conference on Computer Vision (ECCV)*, 2022.

[56] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side Adapter Network for Open-Vocabulary Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[57] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[58] Philip Gage. A new algorithm for data compression. *C Users Journal*, 1994.

[59] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[60] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, 1989.

[61] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)*, 2016.

[62] 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 (ICML)*, 2019.

[63] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[64] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[65] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, 2015.

[66] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[67] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified Perceptual Parsing for Scene Understanding. In *European Conference on Computer Vision (ECCV)*, 2018.

[68] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 2016.

[69] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[70] 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 (ECCV)*, 2014.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[74] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Proc. British Machine Vision Conference (BMVC)*, 2016.

[75] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[76] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[77] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[78] 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*, 2020.

[79] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[80] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[81] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[82] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[83] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[84] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. *https://doi.org/10.5281/zenodo.5143773*, 2021.

[85] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 2021.

[86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[87] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[88] Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. Bfloat16 processing for neural networks. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*. IEEE, 2019.

Appendices

We provide supplementary information in the following order: training details in Appendix A, failure cases in Appendix B, visualizations in Appendix C, and broader impacts in Appendix D.

A Training Details

Refiner. We elaborate how the Refiner transforms the token-level probability map ($\mathbf{p}$) to the final pixel-level prediction mask ($\mathbf{m}$). The Refiner consists of four layers, where each layer comprises of 3×3 convolutions, ReLU [64], batch normalization [65], and a bilinear upsampling operation that doubles the resolution. Each layer takes as input an intermediate visual feature map from the CLIP image encoder to help restore fine-grained details. In particular, we use intermediate feature maps from layer 1 to 4, each denoted as $\mathbf{f}_1^v$, $\mathbf{f}_2^v$, $\mathbf{f}_3^v$, and $\mathbf{f}_4^v$. We explain the process in detail. Firstly, $\mathbf{p}$ is concatenated to $\mathbf{f}_4^v$ along the channel dimension and fed into the first Refiner layer, resulting in a feature map, $\mathbf{m}_4$ with double the resolution of $\mathbf{p}$. Then, $\mathbf{m}_4$ is concatenated to $\mathbf{f}_3^v$—also upsampled to match the resolution of $\mathbf{m}_4$—along the channel dimension and fed into the second Refiner layer, resulting in a feature map, $\mathbf{m}_3$ with quadruple the resolution of $\mathbf{p}$. This is repeated two more times with $\mathbf{f}_2^v$ and $\mathbf{f}_1^v$, resulting in a feature map $\mathbf{m}_1$, which has $2^4 = 16$ times the resolution of $\mathbf{p}$. Finally, $\mathbf{m}_1$ is passed through a final linear projection and sigmoid function, producing the final pixel-wise mask, $\mathbf{m}$.

As explained above, the resolution of $\mathbf{m}$ is 16 times compared to that of $\mathbf{p}$. This is appropriate for RISCLIP-B, which divides the input image into 16×16 patch tokens but an overshoot for RISCLIP-L which adopts 14×14 patch tokens. Hence, we adopt a final bicubic downsampling operation that resizes the upsampled pixel-wise mask to the original input image size for RISCLIP-L.

Although our Refiner is adopted from [23], it differs in that we use intermediate visual features instead multi-modal features. Since PreFusion Adapters are attached to the last six layers of the CLIP image backbone, feature maps from layer six to 11 ($\mathbf{f}_6^v$ to $\mathbf{f}_{11}^v$) are multi-modal, while those from layer zero to five ($\mathbf{f}_0^v$ to $\mathbf{f}_5^v$) are visual. We empirically find that using visual features produces the best results, although the performance differences are minor (within 0.52 IoU points). In overall, we use intermediate visual features from layer one to four ($\mathbf{f}_1^v$ to $\mathbf{f}_4^v$) of the CLIP image encoder.

Training Scheme. We train both RISCLIP-B and -L for 60 epochs with AdamW [83] optimizer, using weight decay of 5e-3 and an initial learning rate of 5e-5 with polynomial learning rate decay. Images are resized to 640×640 for RISCLIP-B and 560×560 for RISCLIP-L, such that the visual encoders are both fed 40×40 patch tokens. We apply random affine transformation and random intensity saturation data augmentations following RefTR [20]. The ratio between dice [68] and focal loss [69], λ_{dice} and λ_{focal}, is empirically set to 1.0 to 1.75, and alpha and gamma, α_{focal} and γ_{focal} in the focal loss are set to 0.65 and 2.0. We use batch size of 32 for the models trained on separate RefCOCO datasets [61, 47] (reported in Table 1), whilst we use bigger batch sizes of 96 for RISCLIP-B and 56 for RISCLIP-L trained on the combined RefCOCO family [61, 47] (reported in Table 2) to prevent prolonged training. Also, different from the recent state-of-the-art methods [20, 22, 24], we do not conduct additional visual grounding pretraining on external large-scale image-text datasets.

Initializations. The backbone encoders are initialized from different sources for RISCLIP-B and -L. In RISCLIP-B, the backbone encoders are initialized with the official weights of OpenCLIP [84] pretrained on LAION-400M [85]. On the other hand, RISCLIP-L's backbone encoders are initialized with the official weights of CLIP [33] pretrained on 400 million image-text pairs collected by OpenAI. We use different sources for the pretrained weights because each source provides a model pretrained with a bigger image size than the other source (*i.e.* OpenCLIP provides a ViT-B backbone pretrained with image size 240×240 pixels whilst OpenAI provides one with 224×224 pixels). We empirically find that using a backbone pretrained with a bigger image size provides better segmentation ability.

The Adapters adopt different initializations. For the Backbone Adapters, we follow [63] and initialize the down-projection linear layer with Kaiming Normal [86] and the up-projection layer with zeros. Initializing the up-projection with zeros makes the initial adapter output zero, which is required for stable training [63]. Inspired by this, we also initialize our Fusion Adapters such that the outputs are initially zero. In detail, for the PreFusion Adapters, we initialize the image-text shared embedding projections in the MHSA as zeros, and, for the PostFusion Adapters, the value projections in MHA

and MHSA as zeros. We experiment with other compositions and find that the adopted initialization provides the best performance, which is slightly better than the others (by about 0.6 IoU points).

Additional Techniques. Furthermore, we observe that incorporating learnable temperatures in the attention modules of the Adapters and introducing learnable channel-wise scalers before residual summation of the Adapter outputs lead to a slight enhancement in performance (up to 0.5 IoU points). All hyperparameters are listed in Table A5.

Table A5: We provide hyperparameters for training RISCLIP-B and -L on the separate RefCOCO datasets [61, 47, 48]. The only difference when training on the combined RefCOCO family [61, 47, 48] is the batch size, which is increased from 32 to 96 and 56 for RISCLIP-B and -L, respectively. We denote Adam with decoupled weight decay [87] as AdamW, rectified linear unit [64] as ReLU, Brain Floating Point [88] format as BF16, and single-precision floating-point format as FP32.

Hyperparameters	**RISCLIP-B**	**RISCLIP-L**
Backbone		
Pretrained Weight Source	OpenAI	OpenCLIP
Image Encoder Patch Size	16	14
Image Encoder Transformer Layers	12	24
Text Encoder Transformer Layers	12	12
Image Encoder MHA Head Number	14	16
Text Encoder MHA Head Number	10	12
$\mathbf{f}_L^v$ dimension	896	1024
$\mathbf{f}_L^t$ dimension	640	768
$\mathbf{v}$ dimension	640	768
$\mathbf{t}$ dimension	640	768
Backbone Adapters		
Image Backbone Adapter Bottleneck dimension	449	512
Text Backbone Adapter Bottleneck dimension	320	384
Non-linear Activation	ReLU	ReLU
Scaler Initial value	0.6	0.6
PreFusion Adapters		
Adapter Number	6	6
$\mathbf{s}_{m-1}^v$	640	768
$\mathbf{s}_{m-1}^t$	640	768
MHA Head Number	10	12
Scaler Initial value	0.5	0.5
PostFusion Adapters		
Adapter Number	6	6
MHA, MHSA Head Number	8	8
Scaler Initial value	0.5	0.5
Others		
Image Size	640	560
Batch Size	32	32
Epochs	60	60
Optimizer	AdamW	AdamW
β_1 for AdamW	0.9	0.9
β_2 for AdamW	0.999	0.999
Learning Rate Initial Value	5e-5	5e-5
Weight Decay Strength	5e-3	5e-3
λ_{dice}	1.0	1.0
λ_{focal}	1.75	1.75
α_{focal}	0.65	0.65
γ_{focal}	2.0	2.0
Locator Precision	BF16	BF16
Refiner Precision	FP32	FP32

In Appendix B and Appendix C, we analyse RISCLIP-B and RISCLIP-L trained on the RefCOCOg [47] (UMD split [48]) dataset. We choose RefCOCOg [47] among the three datasets since it possesses longer and more expressive texts, which offer greater insight about the types of texts that RISCLIP understands and struggles with.

B Failure Cases

Referring Image Segmentation is a challenging task that involves a various expressions and images. Thus, how to group and categorize the image-text pairs is ambiguous. Nevertheless, we attempt to identify common scenarios where RISCLIP often makes false predictions. Specifically, we analyse predictions made by RISCLIP-B on the RefCOCOg [47] (UMD split [48]) test set. We observe that RISCLIP tends to struggle in two situations: "Recognition of Characters" and "Comprehension of Absence". We illustrate each case with visualizations, where the ground-truth masks are displayed in blue and predictions made by RISCLIP in pink.

Recognition of Characters. The first case involves the recognition of characters. Figure B.3 shows that RISCLIP fails to detect numbers '13' and '48', the letter 'B', and the word 'STOP'.

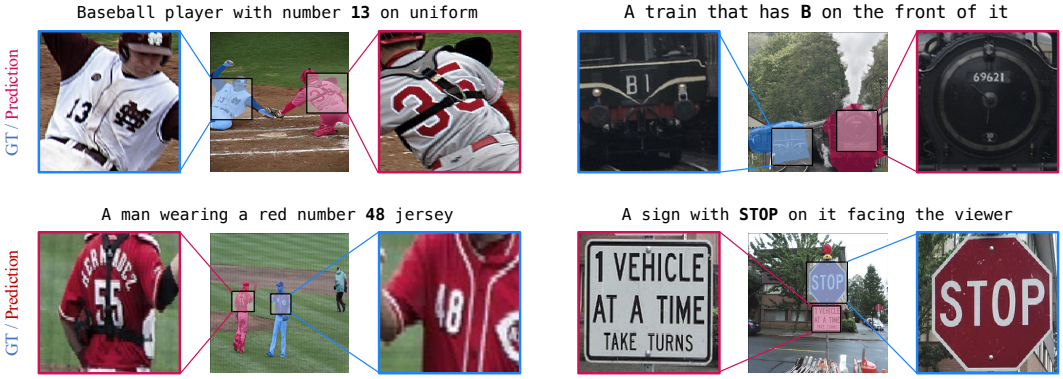

Figure B.3: We visualize RISCLIP-B predictions on RefCOCOg [47] (UMD [48]) test set samples. RISCLIP fails to recognize alphabetic and numeric characters.

Comprehension of Absence. The second case concerns texts that describe the target instance with the 'absence' of some attribute. Figure B.4 shows examples where RISCLIP struggles to comprehend instances described as "A squat vase with *no* flowers" and "The man with the bat wearing his shirt *un*tucked".

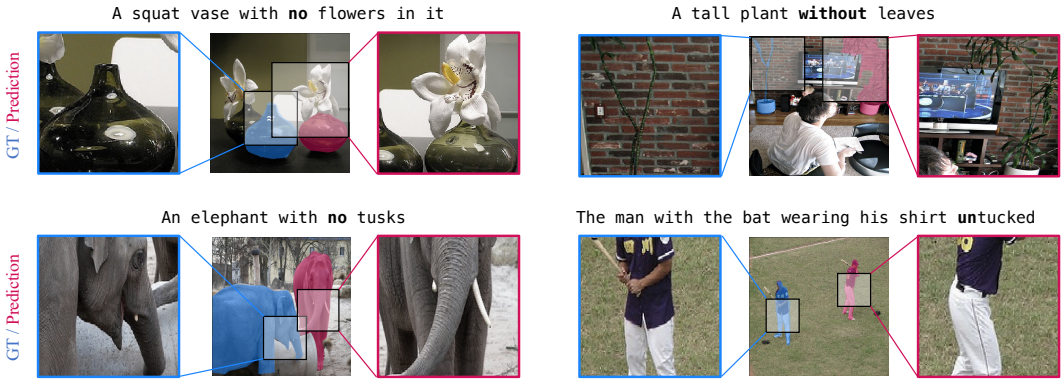

Figure B.4: We visualize RISCLIP-B predictions on RefCOCOg [47] (UMD [48]) test set samples. RISCLIP fails to comprehend texts that describe the target object with the 'absence' of some attribute.

We hypothesize that RISCLIP's relatively poor performance in the two scenarios arises from the limited number of such texts in the dataset. Improving RISCLIP to excel in these cases is another direction for future research.

C Visualizations

RISCLIP-B. We provide visualizations of cases where RISCLIP-B successfully segments the target instance on the RefCOCOg [47] (UMD split [48]) test set in Figure C.5. Even when the texts are lengthy and similar instances exist in the image, RISCLIP-B successfully discerns the referred instance.

Figure C.5: We visualize RISCLIP-B predictions on RefCOCOg [47] (UMD [48]) test set samples. 'L' denotes the text of the left subfigure whilst 'R' denotes that of the right. RISCLIP succeeds in locating different target instances within the same image, even when the texts are long and complex. We also present cases where there are similar instances to the target.

RISCLIP-L. As observed in Table 1, RISCLIP-L performs better than RISCLIP-B. Thus, we provide visual representations of examples where RISCLIP-L successfully identifies target instances that are overlooked by RISCLIP-B on the RefCOCOg [47] (UMD split [48]) test set in Figure C.6. The segments colored in pink on the left are the predictions made by RISCLIP-B, while the purple segments on the right are those made by RISCLIP-L.

The visualizations suggest that RISCLIP-L possesses an additional capability to detect targets that are only partially visible or require the recognition of subtle visual cues. Such ability can be attributed to the more fine-grained CLIP image encoder of RISCLIP-L: during CLIP [33] pretraining, the CLIP image encoder of RISCLIP-L is trained with image size 336×336 and patch size 14×14 which results in 24×24=576 tokens, whilst that of RISCLIP-B is pretrained with image size 240×240 and patch size 16×16 which amounts to 15×15=225 tokens. Thus, RISCLIP-L possesses are more fine-grained image feature extractor and thereby perceives subtle visual cues better.

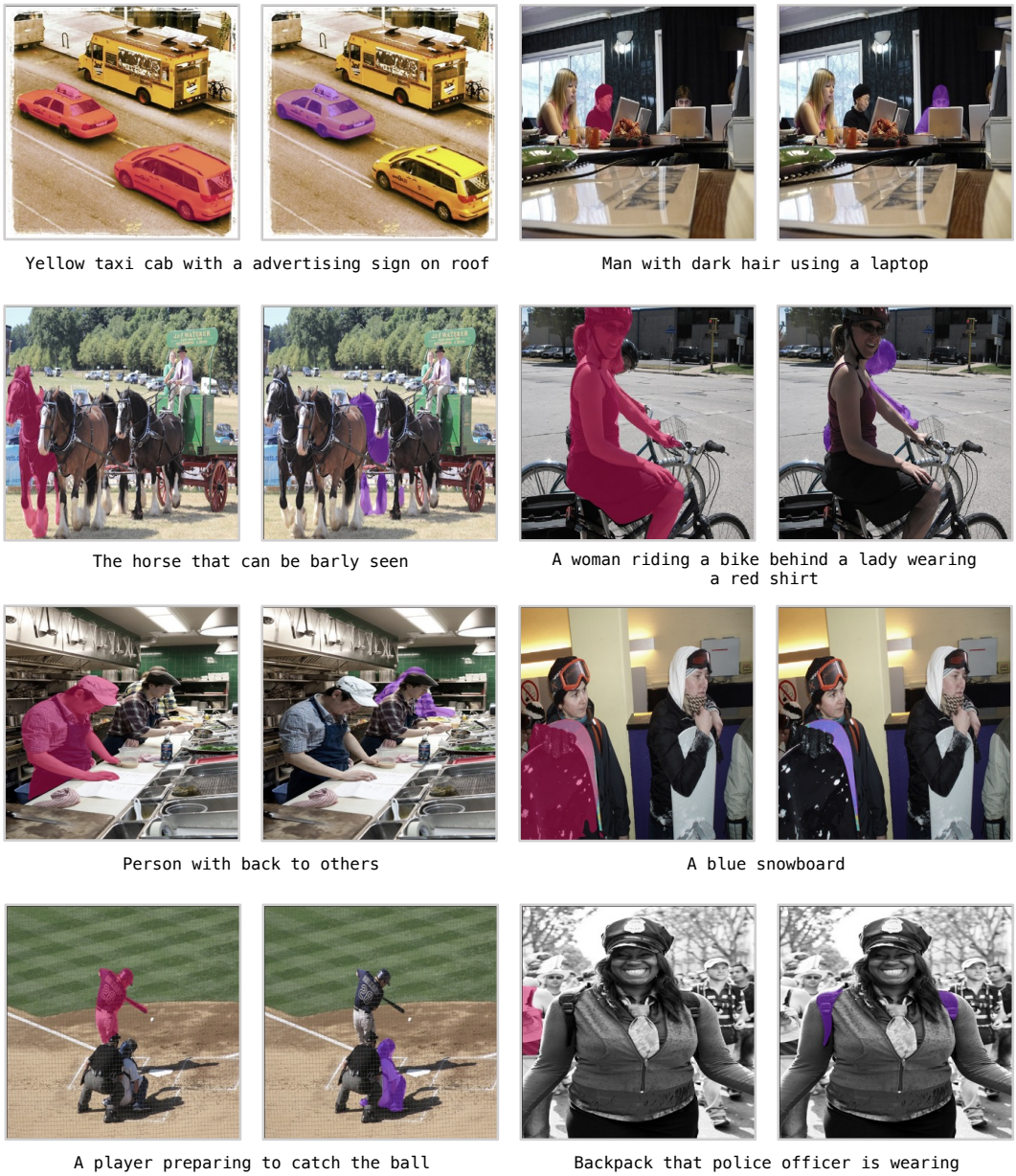

Figure C.6: We visualize RISCLIP-B (left subfigures in pink) and RISCLIP-L (right subfigures in blue) predictions on RefCOCOg [47] (UMD [48]) test set samples. RISCLIP-L detects instances that have small detecting cues or that are partially visible which are omit by RISCLIP-B.

D Broader Impacts

Referring Image Segmentation (RIS) holds the potential to impact numerous domains that use human-computer interaction, such as autonomous driving and assistant robots. For example, a user could instruct a domestic service robot to "fetch the blue cup, not the red one", and the RIS-built-in robot will be able to accurately detect the blue cup and serve his/her owner. Nevertheless, potential ethical concerns, including privacy, model bias, and data processing should be considered. Even the RefCOCO [61, 47] dataset includes offensive expressions and provocative images that require removal. In summary, RIS will impact diverse fields adopting human-computer interaction, but ethical issues should be addressed to ensure beneficial development and safe deployment.