CoupAlign: Coupling Word-Pixel with Sentence-Mask Alignments for Referring Image Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Referring image segmentation aims at localizing all pixels of the visual objects described by a natural language sentence. Previous works learn to straightforwardly align the sentence embedding and pixel-level embedding for highlighting the referred objects, but ignore the semantic consistency of pixels within the same object, leading to incomplete masks and localization errors in predictions. To tackle this problem, we propose **CoupAlign**, a simple yet effective multi-level visual-semantic alignment method, to couple **sentence-mask alignment** with **word-pixel alignment** to enforce object mask constraint for achieving more accurate localization and segmentation. Specifically, the Word-Pixel Alignment (WPA) module performs early fusion of linguistic and pixel-level features in intermediate layers of the vision and language encoders. Based on the word-pixel aligned embedding, a set of mask proposals are generated to hypothesize possible objects. Then in the Sentence-Mask Alignment (SMA) module, the masks are weighted by the sentence embedding to localize the referred object, and finally projected back to aggregate the pixels for the target. To further enhance the learning of the two alignment modules, an auxiliary loss is designed to contrast the foreground and background pixels. By hierarchically aligning pixels and masks with linguistic features, our CoupAlign captures the pixel coherence at both visual and semantic levels, thus generating more accurate predictions. Extensive experiments on popular datasets (e.g., RefCOCO and G-Ref) show that our method achieves consistent improvements over state-of-the-art methods, e.g., about 2% oIoU increase on the validation and testing set of RefCOCO. Especially, CoupAlign has remarkable ability in distinguishing the target from multiple objects of the same class.

1 Introduction

Taking an image and a natural language sentence as input, a referring image segmentation (RIS) model is required to predict a mask for the object described by the sentence. Beyond semantic segmentation which assigns each pixel in an image with a label from a fixed word set, referring image segmentation needs to localize the object pixels indicated by the language expression, which is of arbitrary length and involves an open-world vocabulary, such as object names, attributes, positions, etc. Therefore, this task is the foundation of many applications in the real world, such as image editing [25, 34], autonomous driving [3, 38], and language-based human-robot interaction [35, 1].

The main challenge of RIS is how to align the given language expression with visual pixels for highlighting the target. Early works [15, 6] typically follow the paradigm that first extracts visual and language features from the corresponding encoder and then fuses them for language-pixel alignment. Based on the aligned feature, a decoder is employed to predict a mask. Recently, Transformer-based vision-language models [19, 23, 21] have shown great potential in aligning visual and language

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

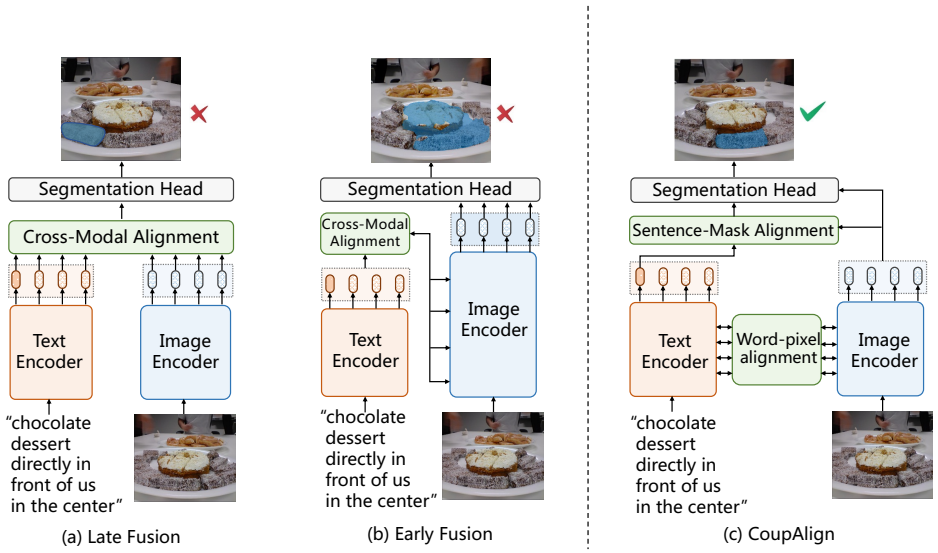

Figure 1: Illustration of the cross-modal alignment (Late/Early Fusion) in existing RIS methods and our CoupAlign that enables hierarchical alignment at word-pixel and sentence-mask levels.

features from fine to coarse levels, e.g., pixel, concept, region and image from visual view, and e.g., word, phrase, sentence, paragraph from language perspective. Inspired by this, extensive works [16, 6, 41, 10, 45] have been investigated to take the advantage of the remarkable cross-modal alignment ability of Transformer architectures for aligning language with pixels more effectively. Some works [6, 41] seek to improve the cross-modal alignment of the encoded vision and language features, as shown in Fig. 1(a). Their cross-modal alignment mainly occurs at the sentence-pixel level. Furthermore, other works [10, 45] fuse the language information into the pixel embeddings at early stages of the image encoder, as shown in Fig. 1(b). The alignment happens more earlier than before, starting from the word-pixel level which is more fine-grained.

Though effective, in Fig. 1(a) and (b), we can also observe many undesirable pixel fragments and localization errors in their failure cases. The reason lies behind these issues may be that the pixel-level alignment (e.g., word-pixel and sentence-pixel level) connects each pixel embedding to language independently, and ignores the visual and semantic consistency among pixels in an object. Therefore, the predicted masks may miss some target pixels or spill over to other objects. To deal with this problem, we propose to regard the pixels of an object as a whole and explicitly introduce object mask constraints for the cross-modal alignment, at both the fine-grained and the coarse-grained levels.

Specifically, in this paper, we propose **CoupAlign** that couples sentence-mask alignment with word-pixel alignment to enforce the object mask constraint for improving the localization and segmentation performance of RIS. The **Word-Pixel Alignment (WPA)** module enables cross-modal interaction at the intermediate layers of the vision and language encoders. Then a set of mask proposals are calculated based on the word-pixel aligned features to indicate the possible objects in an image. The **Sentence-Mask Alignment (SMA)** module learns to weight the masks using the sentence embedding to localize the referred object, and the weighted masks are projected back to provide mask signals for effectively aggregating the pixels of the target. To further enhance the word-pixel and sentence-mask alignments, we design an auxiliary loss that forces the pixels of the target to get close and the pixels not belonging to the target to get away. Our CoupAlign captures the pixel coherence at both visual and semantic levels, and hierarchically aligns the pixels and masks with language information. By doing this, CoupAlign generates more accurate mask predictions for RIS.

To validate the effectiveness of our method, we conduct extensive experiments on popular datasets (RefCOCO [17] and G-ref [31]). It is shown that CoupAlign outperforms state-of-the-art RIS methods by significant margins, e.g, about 2% oIoU increase on the validation and testing sets of RefCOCO. Besides, CoupAlign shows remarkable ability in distinguishing the target from multiple (even greater than twenty) objects of the same class, for example, localizing one person in a team photo using a language query like "middle row second kid from right".

The main contributions of this paper are summarized as follows:

- We tackle the challenging problem of RIS via integrating the mask-aware constraint into the pixel-level semantic alignment for capturing the pixel coherence within objects.
- We propose CoupAlign, a simple yet effective alignment framework, to couple sentence-mask alignment with word-pixel alignment to hierarchically align pixels and masks with linguistic information.
- We conduct comprehensive experiments to demonstrate that our CoupAlign predicts more accurate object locations and masks and significantly outperforms state-of-the-art RIS methods on popular benchmarks.

2 Related Work

Referring Image Segmentation aims to segment the objects referred by natural language expressions. RIS Methods can be summarized into two categories. One focuses on cross-model alignment and representation learning [15, 24, 10]. Another views this problem as a language-conditioned visual reasoning [44, 13], which utilizes a dependency parsing tree to iteratively select the proposal regions.

Due to the breakthrough of Transformer [7] in the computer vision field and its remarkable fusion power for multi-modality, many works [16, 6, 45] begin to explore the cross-model alignment based on Transformer. MDETR [16] shows that simply concatenating vision and language features and then feeding them into the Transformer encoder and decoder can achieve surprising results in many vision-language downstream tasks. VLT [6] is the first work to introduce the Transformer architecture to referring segmentation. It designs a query generation module to utilize the contextual information of images to enrich language expressions and improve robustness. EFN [10] and LAVT [45] insert language-aware attention modules into the image encoding process to achieve early fusion of cross-modal features. MaiL [24] abandons the dual-stream encoders and adopts an unified cross-model Transformer to achieve word-pixel alignment. CRIS [41] focuses on how to transfer CLIP's [36] strong image-text alignment ability to benefit sentence-pixel alignment required by RIS. CRIS uses a Transformer decoder with word queries and a sentence-pixel contrastive loss to achieve this goal.

The above mentioned RIS methods mainly focus on aligning each pixel to word or sentence independently while ignoring the pixel coherence of the target object, thus resulting in inaccurate masks and localization. In contrast, our method couples sentence-mask alignment with word-pixel alignment to hierarchically align pixels and masks with linguistic information, and improving the segmentation and localization accuracy.

Vision Language Modeling targets at seeking an unified representation for the two modalities. From the aspect of architecture, vision language models can be divided into single-stream and dual-stream architecture structures. Single-stream models such as VisualBERT [21], Uniter [2], Uniencoder-vl [20] and Oscar [23], utilizes an unified encoder based on self-attention to fuse the vision and language embeddings. For dual-stream models such as CLIP [36], ALIGN [14], FILIP [46] and GLIP [22], they adopt two unimodality encoders to encode vision and language embeddings and utilizes the dot product to align the outputs. Others like ViLBERT [28] and LXMERT [37] uses two self-attention-based encoders to model intra-modality interactions and employ cross-attention to model cross-modality interactions. Inspired by GLIP [22], we adopt a hybrid encoder design, where dual encoders are used to extract vision and language features and the word-pixel alignment is realized at the intermediate stages of the encoders.

3 Methodology

3.1 Overview

The overall architecture of our CoupAlign is shown in Fig. 2. It is designed following the encoder-decoder paradigm. The CoupAlign encoder consists of an image encoder f_v and a language encoder f_l for extracting vision and language embeddings, a Word-Pixel Alignment (WPA) module that enables the cross-modal interaction at the intermediate layers of the visual and language encoders, and a cross attention module for cross-modal fusion after encoders. The CoupAlign decoder includes a mask generator module that produces N segment proposals, a segmentation head (SegHead) that

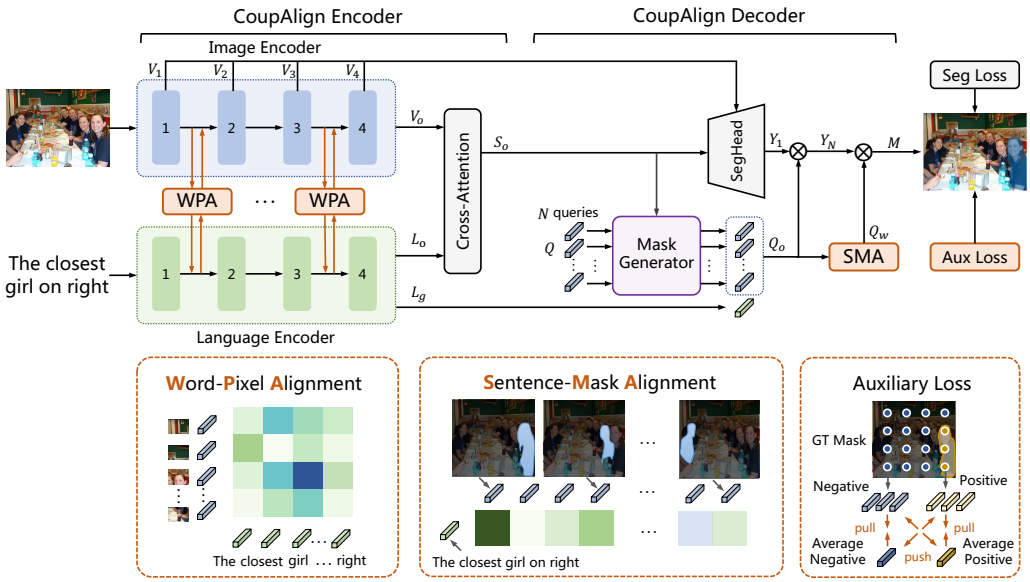

Figure 2: Architecture of our CoupAlign. Given an image and a language sentence, the image features V_o can be extracted from the image encoder, and the word-level features L_o and sentence feature L_g are extracted from the language encoder. The Word-Pixel Alignment (WPA) module enables cross-modal interactions at each encoder stage. Based on the word-pixel aligned features S_o, the mask generator produces N mask embeddings Q_o. The Sentence-Mask Alignment (SMA) weights Q_o using L_g and projects the mask signals back to Y_N, and finally generates the prediction M.

upsamples the pixel-level embeddings, and a Sentence-Mask Alignment module (SMA) to weight the masks via the sentence and then project back to the pixel-level embeddings to enforce object mask constraint. Together with the segmentation loss, an auxiliary loss that compares foreground pixels with background ones to enhance the hierarchical aligning at word-pixel and sentence-mask levels.

3.2 CoupAlign Encoder

Image Encoder. Given an image $I \in \mathbb{R}^{H \times W \times 3}$, we adopt Swin Transformer [26] as the image encoder backbone to extract multi-level visual features. We utilize the features from the 1st–4th stages of Swin Transformer, denoted as $\{V_i \in \mathbb{R}^{H_i \times W_i \times C_i}\}_{i=1}^{4}$, where C_i, H_i, W_i are the number of channels, the height, and the width of the feature map at the i-th stage, respectively.

Language Encoder. We use BERT [5] with 12 layers as our language encoder backbone. Given a referring expression $\{w_i\}_{i=1}^{T}$, where T is the length of the expression, we first use the BERT tokenizer based on WordPiece embeddings [43] to get the word embeddings $E \in \mathbb{R}^{T \times D}$, where D is the hidden feature dimensionality of BERT. Then the word embeddings E are fed into the hidden layers of BERT to extract language features. In order to make word-pixel alignment with vision features from Swin Transformer, we aggregate every 3 hidden layers into a stage, so that we can have the same stage number of language features as the vision features, which are denoted as $\{L_i \in \mathbb{R}^{T \times D}\}_{i=1}^{4}$. We also utilize the output [CLS] token as the global semantic representation of the referring expression, which is denoted as $L_g \in \mathbb{R}^{1 \times D}$.

Word-Pixel Alignment (WPA). For segmentation task, we need to obtain a fine-grained segmentation mask, which means that low-level features like words in sentences and edges and shapes of objects in images are crucial for finding correct segments. The cross-modal alignment at early encoder stage is proved to be effective for referring image segmentation [6, 45]. Their early alignments are in single-direction of language to vision. In contrast, our WPA module is implemented with bidirectional cross attention to achieve both language-to-vision and vision-to-language alignment, as:

$$V_i', L_i' = \text{BiAttn}(V_i, L_i), \quad i \in \{1, \ldots, 4\}, \tag{1}$$

$$V_i' \leftarrow \text{Gate}(V_i'), \quad L_i' \leftarrow \text{Gate}(L_i'), \tag{2}$$

$$V_{i+1} = \text{SwinStage}_i(V_i + L_i'), \quad L_{i+1} = \text{BERTStage}_i(L_i + V_i'), \tag{3}$$

where $\text{SwinStage}_i(\cdot)$ represents the Stage i defined in Swin Transformer. $\text{BERTStage}_i(\cdot)$ represents the Stage i of BERT. The cross-modality interaction is achieved by the bidirectional cross attention (BiAttn), which computes the context vectors of one modality by attending to the other modality:

$$\hat{V}_i = V_i W_i^v, \quad \hat{L}_i = L_i W_i^l, \quad Attn_i = \hat{V}_i \hat{L}_i^T / \sqrt{d}, \tag{4}$$

$$L_i' = \text{softmax}(Attn_i) L_i \hat{W}_i^l, \quad V_i' = \text{softmax}(Attn_i^T) V_i \hat{W}_i^v, \tag{5}$$

where d is the dimension of the joint embedding space, and $W_i^v \in \mathbb{R}^{C_i \times d}$, $W_i^l \in \mathbb{R}^{D \times d}$, $\hat{W}_i^l \in \mathbb{R}^{d \times C_i}$, $\hat{W}_i^v \in \mathbb{R}^{d \times D}$ are all projection matrices.

In order to prevent the fused features V_i' and L_i' from overwhelming the original signals V_i and L_i respectively. We design a Gate network [45] to control the fused information flow:

$$\text{Gate}(F_i) = \text{MLP}(F_i) \odot F_i, \tag{6}$$

where F_i is the fused feature from BiAttn, $\odot$ indicates element-wise multiplication, and MLP is a two-layer perceptron with the first layer being a linear layer followed by the ReLU activation [32] and the second layer being a linear layer followed by a hyperbolic tangent activation.

Cross Attention. To better capture high-level semantics and generate the pixel-level fused features, we use a plain multi-head attention layer [40] to fuse the high-level vision and language features from the image and language encoders. As illustrated in Fig. 2, first, we obtain the high-level vision features $V_o \in \mathbb{R}^{H_o \times W_o \times C_o}$ from the output of the image encoder and the high-level language features $L_o \in \mathbb{R}^{T \times D}$ from the language encoder. Then we map the vision and language features to $\mathbb{R}^D$. After that we concatenate them to produce a $F_o \in \mathbb{R}^{(H_o W_o + T) \times D}$, which is fed into the cross attention layer. We add learnable positional emdeddings $e_p \in \mathbb{R}^{H_o \times W_o \times D}$ to the vision features before the concatenation. Finally the output of the cross attention layer is added with the original input vision features to produce a low-resolution feature map $S_o \in \mathbb{R}^{H_o \times W_o \times C_o}$, as:

$$V_o' = V_o W_o^v + e_p, \quad L_o' = L_o W_o^l, \tag{7}$$

$$S_o = \text{CrossAttn}(F_o) + V_o, \quad F_o = [V_o'; L_o'], \tag{8}$$

where $W_o^v \in \mathbb{R}^{C_o \times D}$ and $W_o^l \in \mathbb{R}^{D \times D}$ are projection matrices and [;] denotes concatenation.

3.3 CoupAlign Decoder

Mask Generator. To provide object mask constraints for the word-pixel alignment, we first design a mask generator that produces N segment proposals based on the output feature of the CoupAlign encoder. The generator is built upon a 6-layer transformer decoder. It takes $S_o \in \mathbb{R}^{H_o \times W_o \times C_o}$ and N learnable queries $Q \in \mathbb{R}^{N \times d_q}$ as input, where d_q is the hidden dimensionality of the transformer decoder, and outputs the hidden feature $Q_o \in \mathbb{R}^{N \times d_q}$ of the N segment tokens, as:

$$Q_o = \text{MaskGenerator}(Q, S_o). \tag{9}$$

Segmetation Head (SegHead). To upsample the pixel-level features for generating final segmentation maps, we construct a segmentation head that takes the CoupAlign encoder output feature S_o and the multi-level visual features $\{V_i\}_{i=1}^4$ as input, and obtains output features as:

$$\begin{cases} Y_5 = S_o, \\ Y_i = \text{Up}(\rho(Y_{i+1})) + \gamma(V_i), \quad i = 4, 3, 2, 1, \end{cases} \tag{10}$$

where ρ is a two-layer convolution network where each layer performs a 3×3 convolution followed by ReLU and batch normalization, Up denotes $2\times$ bilinear upsampling, γ is a 1×1 convolution that maps the vision features to $\mathbb{R}^{d_s}$. We take $Y_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times d_s}$ as the output of SegHead.

Sentence-Mask Alignment (SMA). After generating mask proposals and obtaining upsampled pixel-level embeddings, the sentence-mask alignment assigns each mask a score according to the language query and then project the weighted masks back to the pixel-level feature maps to introduce object mask constraint for word-pixel alignment. At first, We project Q_o and Y_1 to $\hat{Q}_o \in \mathbb{R}^{N \times D}$ and $\hat{Y}_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times D}$ via learnable weights $W^Q \in \mathbb{R}^{d_q \times D}$ and $W^Y \in \mathbb{R}^{d_s \times D}$. We use the mask embeddings $\hat{Q}_o$ and the sentence representation L_g generated from the language encoder to compute attention weight $Q_w \in \mathbb{R}^{1 \times N}$. Higher weight in Q_w indicate that the corresponding mask is most

likely to contain the object referred by the language. Then, $\hat{Q}_o$ and $\hat{Y}_1$ are multiplied to get N mask predictions $Y_N \in \mathbb{R}^{N \times \frac{H}{4} \times \frac{W}{4}}$. Finally, Y_N is multiplied by Q_w to generate the final mask prediction $M \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4}}$, which is further upsampled to the input image size. This procedure is represented as:

$$Q_w = \text{softmax}(sim(L_g, \hat{Q}_o)), \tag{11}$$

$$M = Y_N \otimes Q_w, \quad Y_N = \hat{Y}_1 \otimes \hat{Q}_o, \tag{12}$$

where $sim(\cdot)$ is cosine similarity and $\otimes$ denotes the broadcast tensor multiplication.

3.4 Loss Functions

Segmentation Loss. In the context of referring image segmentation, each pixel in the image I is discriminated into background or foreground. Thus this task can be viewed as a pixel-wise binary classification problem. Specifically, we use bilinear interpolation to upsample M to the original image size, obtaining M'. Let m'_i and $\hat{m}_i$ be the values of pixel i from M' and $\hat{M}$ (ground truth mask) respectively. Then the segmentation loss are implemented via a binary cross entropy loss as:

$$L_j^{\text{Seg}} = -\frac{1}{H \times W} \sum_{i=1}^{H \times W} \hat{m'}_i \log(\sigma(m_i)) + (1 - \hat{m}_i) \log(1 - \sigma(m'_i)), \tag{13}$$

where σ denotes sigmoid function, and j is the image index in a training batch.

Auxiliary Loss. The word-pixel and sentence-mask alignment collaborate in a hierarchically learning manner to enforce object mask constraint for improving referring image segmentation. We adopt a pixel-level contrastive loss as auxiliary to further enhance the learning of the two alignments and enable the aligning consistency at both the fine-grained and coarse-grained level. This loss forces the pixels within the ground truth masks to get closer, and the rest pixels to get away. Therefore the foreground pixels are aggregated together to generate more accurate predictions. Specifically, we resize the ground truth mask $\hat{M}$ to the same shape as the segmentation map Y_1. Let y_i be the vector of Y_1 in pixel i. Then we use the resized mask to divide positive and negative samples in each image. For pixel i, if $\hat{m}_i$ is 0, then y_i belongs to the negative set $\mathcal{N}$ and is denoted as y_i^-; otherwise y_i belongs to the positive set $\mathcal{P}$ and is denoted as y_i^+. Then we average the negative vectors and the positive vectors to get the negative prototype $\hat{y}^-$ and the positive prototype $\hat{y}^+$. Afterward we adopt InfoNCE [39] to compute the contrastive loss, as:

$$L_{P2N}^{\text{Aux}} = -\frac{1}{|\mathcal{P}|} \sum_{y_i^+ \in \mathcal{P}} \frac{\exp(y_i^+ \cdot \hat{y}^+/\tau)}{\exp(y_i^+ \cdot \hat{y}^+/\tau) + \sum_{y_k^- \in \mathcal{N}} \exp(y_i^+ \cdot y_k^-/\tau)}, \tag{14}$$

$$L_{N2P}^{\text{Aux}} = -\frac{1}{|\mathcal{N}|} \sum_{y_i^- \in \mathcal{N}} \frac{\exp(y_i^- \cdot \hat{y}^-/\tau)}{\exp(y_i^- \cdot \hat{y}^-/\tau) + \sum_{y_k^+ \in \mathcal{P}} \exp(y_i^- \cdot y_k^+/\tau)}, \tag{15}$$

$$L_j^{\text{Aux}} = L_{P2N}^{\text{Aux}} + L_{N2P}^{\text{Aux}}, \tag{16}$$

where τ is the temperature parameter. Finally, the total loss for the training is that:

$$L = \frac{1}{B} \sum_{j=1}^{B} (L_j^{\text{Seg}} + \lambda L_j^{\text{Aux}}), \tag{17}$$

where B is the batch size, λ is a hyper-parameter.

4 Experiments

4.1 Settings

Dataset. We evaluate our method on four widely-used datasets: RefCOCO [47], RefCOCO+[47], G-Ref [31] and ReferIt [17]. RefCOCO consists of 142,209 referring expressions for 50,000 objects in 19,994 images and RefCOCO+ has 141,564 referring expressions for 49,856 objects in 19,992 images. G-Ref was collected on Amazon Mechanical Turk in a non-interactive setting which consists

	Backbone	RefCOCO			RefCOCO+			G-Ref		ReferIt
		val	test A	test B	val	testA	testB	val (U)	test (U)	test
MCN [30]	Darknet-53	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
BRINet [11]	ResNet-101	60.98	62.99	59.21	48.17	52.32	42.11	-	-	63.46
CMPC [12]	ResNet-101	61.36	64.53	59.64	49.56	53.44	43.23	-	-	65.53
LSCM [13]	ResNet-101	61.47	64.99	59.55	49.34	53.12	43.50	-	-	66.57
CGAN [29]	ResNet-101	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	-
BUSNet [44]	ResNet-101	63.27	66.41	61.39	51.76	56.87	44.13	-	-	-
EFN [10]	ResNet-101	62.76	65.69	59.67	51.50	55.24	43.01	-	-	66.70
LTS [15]	DarkNet-53	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [6]	DarkNet-53	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	-
ReSTR [18]	ViT-B-16	67.22	69.30	64.45	55.78	60.44	48.27	54.48	-	70.18
CRIS [41]	CLIP-ResNet-101	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	-
LAVT [45]	Swin-B	72.73	75.82	68.79	62.14	**68.38**	55.10	61.24	62.09	-
CoupAlign (ours)	Swin-B	**74.70**	**77.76**	**70.58**	**62.92**	68.34	**56.69**	**62.84**	**62.22**	**73.28**

Table 1: Comparison with SOTA methods on the oIoU metric on RIS datasets.

of 85,474 referring expressions for 54,822 objects in 26,711 images. ReferIt has 130,525 referring expressions in 19,894 images which are collected from IAPR TC-12 [8]. RefCOCO+ is disallowed from using location words in referring expressions. The languages used in RefCOCO and RefCOCO+ tend to be more concise and less flowery than the languages used in the G-Ref. The expressions in G-Ref tend to be longer with an average of 8.43 words, which makes it more challenging compared with RefCOCO and RefCOCO+. Expressions in ReferIt are usually shorter than the other datasets.

Metrics. We adopt three common used metrics in referring image segmentation: the overall intersection-over-union (oIoU), the mean intersection-over-union (mIoU) and prec@X. The oIoU measures as the ratio between the total intersection area and the total union area of all test samples. By default, we use this metric to compare CoupAlign with other works. The mIoU is the average of the IoU over all test samples. prec@X measures the percentage of test images with an IoU score higher than a threshold ϵ, where $\epsilon \in \{0.5, 0.7, 0.9\}$.

Implementation Details. We implement our model in the PyTorch [33] framework. We choose $\text{BERT}_{\text{BASE}}$ [5] with 12 layers and hidden size 768 as our language encoder. We slightly modify the implementation of BERT in HuggingFace's Transformer library [42] and initialize it with official pre-trained weights. The image encoder layers are initialized with the classification weights pre-trained on ImageNet-22K [4]. The number of queries of the mask generator N is 100. The rest of the weights in our model are randomly initialized. We adopt AdamW [27] optimizer with weight decay 0.01. We adopt the polynomial learning rate decay schedule and set the initial learning rate to 3e-5, the end learning rate to 1.5e-5 and the max decay epoch to 25. Our model is trained for 50 epochs with batch size 16. The images are resized to 448×448 without data augmentation and the maximum sentence length is set to 30. The weight of the auxiliary loss λ is set to 0.1.

4.2 Comparison with State-of-the-Arts

Results on RefCOCO. In Tab. 1, we compare CoupAlign with state-of-the-art (SOTA) RIS methods by oIoU metric. The language expressions provided by RefCOCO contain many words of positional relationships, e.g., "The closet girl on the right", which requires the RIS models to not only capture the pixel coherence within objects, but also the spatial relationship among multiple objects of the same class. Compared with the SOTA method LAVT [45], CoupAlign achieves better performance with absolute oIoU increases of 1.97%, 1.94%, and 1.79% on the validation, testA, and testB sets of RefCOCO, respectively.

Results on RefCOCO+. As shown in the Tab. 1, the performance of CoupAlign on RefCOCO+ is comparable to or better than LAVT which is the most recent state-of-the-art method (published in CVPR2022). The improvements on RefCOCO+ are less significant than those on G-Ref and RefCOCO. This is because the language captions in RefCOCO+ hardly contain the descriptions of the relative or absolute spatial locations (e.g., "closest" or "right bottom") of the target object in images, and the effectiveness of our method mainly lies in the ability of localizing objects in such challenging scenarios, e.g., "middle row second kid from right" in Fig. 3.

Results on G-Ref. The language annotations of G-Ref have more complex language structure with more words, e.g., "chocolate dessert directly in front of us in the center", which requires more fine-detailed word-pixel alignment, more discriminative sentence-mask alignment, and a more

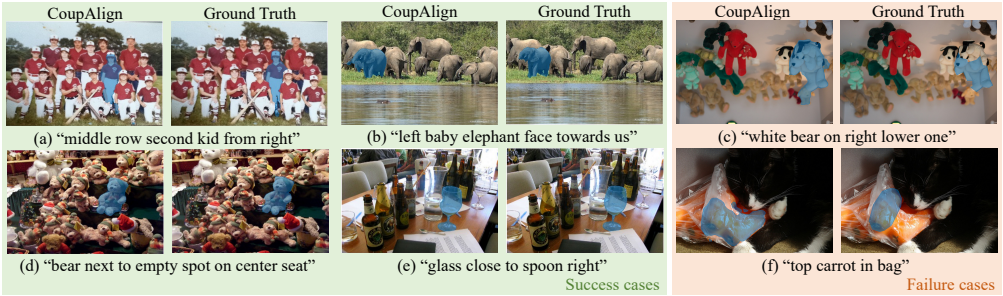

Figure 3: Visualization examples of the success cases and failure cases of CoupAlign.

Method \ IoU Range	[0,0.1)	[0.1,0.2)	[0.2,0.3)	[0.3,0.4)	[0.4,0.5)
LAVT	330	43	64	82	113
CoupAlign	271	47	44	62	94

Table 2: Analysis of localization ability.

Num of mask	mIoU	prec@0.5	prec@0.7	prec@0.9
20	74.77	85.46	76.22	30.91
50	74.98	85.67	76.20	31.88
100	75.49	86.40	77.59	32.40

Table 3: Ablation of different mask numbers.

comprehensive understanding of the sentence. As shown in Tab. 1 on G-Ref (UMD partition), our CoupAlign outperforms LAVT by 1.6% and 0.13% on the validation and test sets, respectively. The consistent improvements on all the evaluated datasets demonstrate the capability of our CoupAlign in modeling hierarchical cross-modal alignment at pixel and mask levels.

Analysis of Localization Ability. If the IoU between a mask prediction and the Ground Truth mask is smaller than 0.5, we consider the prediction as a failure case for localization. In Tab. 2, we can see that our CoupAlign has fewer failure cases than LAVT in most IoU ranges.

Mask Number N. From Tab. 3, we find that the mask number N in mask generator is correlated to the model performance. To balance the computational cost and performance, N is set to 100.

4.3 Visualization

Success cases and failure cases. In Fig. 3, we show some representative visualization examples of success predictions and failure cases of CoupAlign. The examples are from the RefCOCO val. set. We can see that CoupAlign works well in the scenes where crowded objects have similar color and context and stay close to each other. However, if the boundary of the object is blurred, or there exists occlusion, the segmentation prediction is roughly accurate, but not precise enough.

Word-Pixel and Sentence-Mask Alignment. To validate whether our WPA and SMA modules can perform accurate and consistent alignments, we visualize the intermediate attention maps of the alignment modules in Fig. 4. It can be observed that our WPA module accurately highlights the most relevant pixels corresponding to the word. Note that the vocabulary of RIS language annotations is very large compared to the small fixed label set of semantic segmentation. Surprisingly, our CoupAlign can not only capture the semantic meaning of words in various types, e.g., nouns, positions, and attributes, but also distinguish synonyms, e.g., child, man, and lady, which are roughly defined as “person” in many semantic segmentation datasets like PASCAL VOC [9]. As for the SMA module, we show three masks with their attention weights ranked in descending order. The mask with the highest weight is the most accurate one and the masks with lower weights are less accurate. Fig. 4(right) demonstrates that our SMA module provides the effective mask constraint for benefiting the hierarchical aligning at pixel and mask levels.

4.4 Ablation Study

Effect of WPA Types. Our WPA module enables cross-modal interaction at both low-level and high-level encoding stages. In Tab. 4, Bi-WPA indicates our WPA which is implemented via bidirectional cross attention, Uni-WPA represent unidirectional attention only from language to image features, as in LAVT [45]. The performance dramatically drops by 4.3% (74.70% vs. 70.43%) on oIoU when WPA is removed. And also when we replace our Bi-WPA with Uni-WPA, the oIoU result drops by 2% (74.70% vs. 72.70%). These results demonstrate the effectiveness of our WPA module.

Effect of WPA’s Number and Position. In our experiment, we used four WPA modules, two of which are in the early encoding stage and the other two are in the late encoding stage. To study the

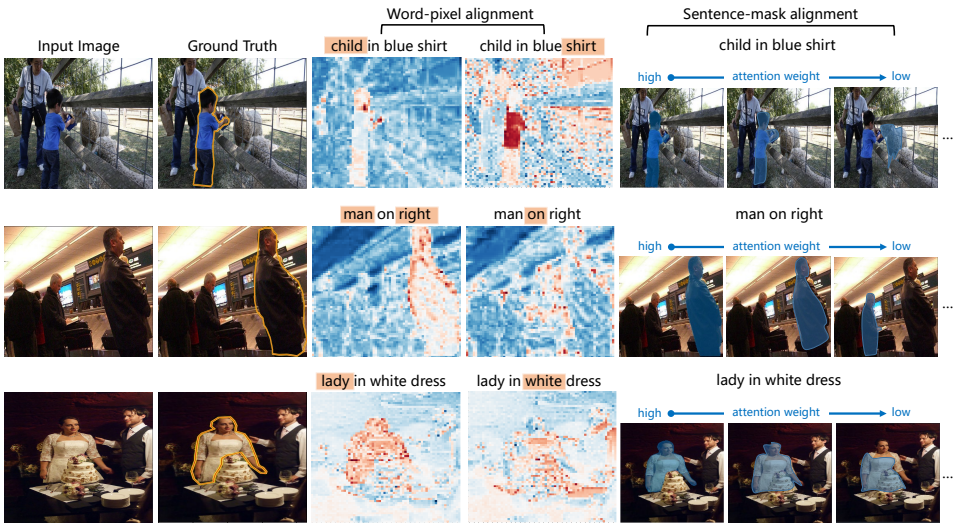

Figure 4: Visualization examples of our word-pixel alignment and sentence-mask alignment.

Bi-WPA	Uni-WPA	SMA	Aux Loss	oIoU	mIoU	prec@0.5	prec@0.7	prec@0.9
		✓	✓	70.43	69.61	80.03	68.91	27.47
	✓	✓	✓	72.70	73.44	84.07	72.60	29.02
✓			✓	73.02	73.85	84.86	74.62	29.25
✓		✓		73.70	74.21	85.32	75.31	30.14
✓		✓	✓	**74.70**	**75.49**	**86.40**	**77.59**	**32.40**

Table 4: Ablation studies of our WPA module, SMA module, and auxiliary loss.

WPA's position	stage 1-4	stage 1,2	stage 3,4	stage 4	stage 3	stage 2	stage 1
oIoU	**74.70**	72.74	73.93	73.61	72.53	72.63	72.59
mIoU	**75.49**	73.87	74.88	74.68	73.47	73.48	73.97

Table 5: Ablation study of WPA modules' number and position.

effects of the numbers of WPA modules, we first conduct two baseline models that alternatively remove two WPA at early or late encoding stages. As shown in the Tab. 5, when we remove the last two WPA modules the performance drops about 2%, and when we remove the first two WPA modules the performance drops about 0.8%. These results validate the effectiveness of WPA modules at both early and late stages and indicate that the latter WPA modules play a more important role in our modules. Then, we only use one WPA module, which is inserted at different encoding stages. The WPA module at the 4-th stage is more effective than those inserted at other stages.

Effect of SMA. In Tab. 4, when SMA is removed, the oIoU result decreases by 1.7% (74.70% vs. 73.02%), since the cross-modal alignment becomes less accurate without mask constraints.

Effect of Auxiliary Loss. From the last two columns of Tab. 4, we observe a performance drop of 1% (74.70% vs. 73.70%) on oIoU when the auxiliary loss is disabled, which verifies the ability of the loss in enhancing the word-pixel and sentence-mask alignments.

5 Conclusion

In this work, we propose CoupAlign which is a simple yet effective cross-modal alignment framework that couples sentence-mask alignment with word-pixel alignment to enforce mask-aware constraint for referring segmentation. The word-pixel alignment module enables fine-grained cross-modal interactions at the early encoding stage. The sentence-mask alignment module uses the sentence to weight the mask proposals generated based on the word-pixel aligned features, and then projects the mask constraints back to guide the word-pixel alignment. The auxiliary contrastive loss that compares among pixel pairs to further enhance the hierarchical alignments at word-pixel and sentence-mask levels. CoupAlign captures both visual and semantic coherence of pixels within the referred object, and significantly outperforms state-of-the-art RIS methods.

References

[1] H. Ahn, S. Choi, N. Kim, G. Cha, and S. Oh. Interactive text2pickup networks for natural language-based human–robot collaboration. *IEEE Robotics and Automation Letters*, 3(4):3308–3315, 2018.

[2] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Learning universal image-text representations. 2019.

[3] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] H. Ding, C. Liu, S. Wang, and X. Jiang. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16321–16330, 2021.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] H. J. Escalante, C. A. Hernández, J. A. Gonzalez, A. López-López, M. Montes, E. F. Morales, L. E. Sucar, L. Villasenor, and M. Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer vision and image understanding*, 114(4):419–428, 2010.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[10] G. Feng, Z. Hu, L. Zhang, and H. 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.

[11] Z. Hu, G. Feng, J. Sun, L. Zhang, and H. 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.

[12] S. Huang, T. Hui, S. Liu, G. Li, Y. Wei, J. Han, L. Liu, and B. Li. Referring image segmentation via cross-modal progressive comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10488–10497, 2020.

[13] T. Hui, S. Liu, S. Huang, G. Li, S. Yu, F. Zhang, and J. Han. Linguistic structure guided context modeling for referring image segmentation. In *European Conference on Computer Vision*, pages 59–75. Springer, 2020.

[14] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[15] Y. Jing, T. Kong, W. Wang, L. Wang, L. Li, and T. Tan. Locate then segment: A strong pipeline for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9858–9867, 2021.

[16] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. 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.

[17] S. Kazemzadeh, V. Ordonez, M. Matten, and T. 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.

[18] N. Kim, D. Kim, C. Lan, W. Zeng, and S. Kwak. Restr: Convolution-free referring image segmentation using transformers. *arXiv preprint arXiv:2203.16768*, 2022.

[19] W. Kim, B. Son, and I. Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[20] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344, 2020.

[21] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[22] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021.

[23] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[24] Z. Li, M. Wang, J. Mei, and Y. Liu. Mail: A unified mask-image-language trimodal network for referring image segmentation. *arXiv preprint arXiv:2111.10747*, 2021.

[25] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34, 2021.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[27] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] G. Luo, Y. Zhou, R. Ji, X. Sun, J. Su, C.-W. Lin, and Q. Tian. Cascade grouped attention network for referring expression segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1274–1282, 2020.

[30] G. Luo, Y. Zhou, X. Sun, L. Cao, C. Wu, C. Deng, and R. Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10034–10043, 2020.

[31] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2016.

[32] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

[35] S. Pate, W. Xu, Z. Yang, M. Love, S. Ganguri, and L. L. Wong. Natural language for human-robot collaboration: Problems beyond language grounding. *arXiv preprint arXiv:2110.04441*, 2021.

[36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[37] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[38] M. Toromanoff, E. Wirbel, and F. Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020.

[39] A. Van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[41] Z. Wang, Y. Lu, Q. Li, X. Tao, Y. Guo, M. Gong, and T. Liu. Cris: Clip-driven referring image segmentation. *arXiv preprint arXiv:2111.15174*, 2021.

[42] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[43] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[44] S. Yang, M. Xia, G. Li, H.-Y. Zhou, and Y. 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.

[45] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr. Lavt: Language-aware vision transformer for referring image segmentation. *arXiv preprint arXiv:2112.02244*, 2021.

[46] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, and C. Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

[47] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please see the Abstract and Introduction sections.
   (b) Did you describe the limitations of your work? [Yes] Please see the Conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please see the Conclusion section.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code is enclosed in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section **??** for more experimental details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The experiments are conducted under a fixed seed, and have the same configuration as previous works.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] All experiments are conducted on an NVIDIA 3090 GPU.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Please see the Experiment and Reference sections.

(b) Did you mention the license of the assets? [No] The data and models used in our work are publicly released.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The data and models used in our work are publicly released.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data used in our work does not contain personally identifiable information or offensive content.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]