WiCo: Win-win Cooperation of Bottom-up and Top-down Referring Image Segmentation

Zesen Cheng[1,2], **Peng Jin**[1,2], **Hao Li**[1,2], **Kehan Li**[1,2], **Siheng Li**[4]
Xiangyang Ji[4], **Chang Liu**[4 †] and **Jie Chen**[1,2,3 †]
[1] School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2] AI for Science (AI4S)-Preferred Program, Peking University Shenzhen Graduate School, China
[3] Peng Cheng Laboratory, Shenzhen, China
[4] Tsinghua University, Beijing, China
{cyanlaser, jp21, kehanli}@stu.pku.edu.cn, lisiheng21@mails.tsinghua.edu.cn
{xyji, liuchang2022}@tsinghua.edu.cn, {lihao1984, jiechen2019}@pku.edu.cn

Abstract

The top-down and bottom-up methods are two mainstreams of referring segmentation, while both methods have their own intrinsic weaknesses. Top-down methods are chiefly disturbed by **Polar Negative** (PN) errors owing to the lack of fine-grained cross-modal alignment. Bottom-up methods are mainly perturbed by **Inferior Positive** (IP) errors due to the lack of prior object information. Nevertheless, we discover that two types of methods are highly complementary for restraining respective weaknesses but the direct average combination leads to harmful interference. In this context, we build **Wi**n-win **Co**operation (WiCo) to exploit complementary nature of two types of methods on both interaction and integration aspects for achieving a win-win improvement. For the interaction aspect, **Complementary Feature Interaction** (CFI) provides fine-grained information to top-down branch and introduces prior object information to bottom-up branch for complementary feature enhancement. For the integration aspect, **Gaussian Scoring Integration** (GSI) models the gaussian performance distributions of two branches and weighted integrates results by sampling confident scores from the distributions. With our WiCo, several prominent top-down and bottom-up combinations achieve remarkable improvements on three common datasets with reasonable extra costs, which justifies effectiveness and generality of our method.

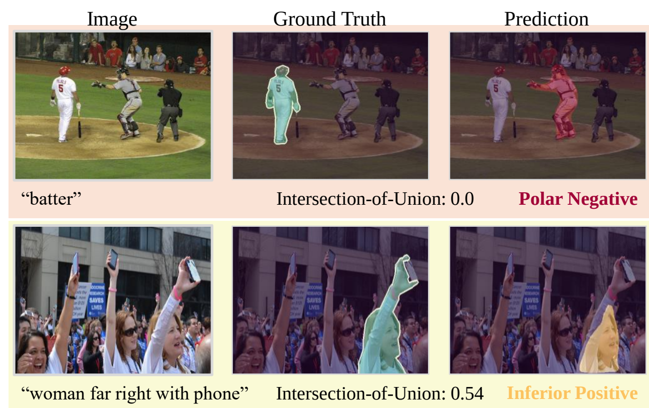


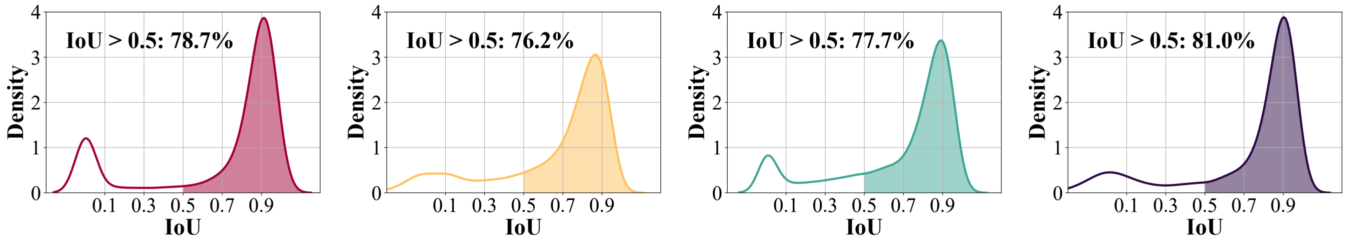
Figure 1: **Visualization of some failure cases**. The sample of first row is defined as **Polar Negative** samples which have no overlap with the **Ground Truth** and the Intersection-of-Union (IoU) closes to 0. The sample of second row is defined as **Inferior Positive** samples which ignore some object parts and IoU ranges from 0.5 to 0.8. Existing methods still fail to process these two types of errors.

1 Introduction

Referring image segmentation (RIS) is a new type of segmentation task aiming to segment the object referred by a natural query expression. The current approaches for referring image segmentation can be broadly classified into two categories [Hui *et al.*, 2020], i.e., top-down and bottom-up methods. Top-down methods calculate the object-centric cross-modal alignment between each region proposal from pretrained detector and query for getting cross-modal instance embeddings and then decode cross-modal instance embeddings to alignment score for retrieving the most confident region proposal as segmentation result [Yu *et al.*, 2018; Liu *et al.*, 2019]. Bottom-up methods calculate the fine-grained cross-modal alignment between each pixel and query for acquiring cross-modal pixel embeddings and then decode the embeddings to retrieve those pixels of referred object [Wang *et al.*, 2022; Yang *et al.*, 2022]. However, according to our observations in Figure 1, existing top-down and bottom-up methods are still perturbed by two types of errors: **Polar Negative** (PN) and **Inferior Positive** (IP). These two errors can be identified by the Intersection-over-Union (IoU) between predictions and ground truths. PN samples are those predictions that have nearly no overlap with the ground truth (IoU $\rightarrow$ 0). IP samples are those predictions that ignore some components of the referred object (IoU $\in [0.5, 0.8]$).

To analyze how top-down and bottom-up methods are dis-

† Corresponding Author



(a) Top-down (b) Bottom-up (c) Direct combination (d) WiCo

Figure 2: **The IoU distribution** of (a) top-down method (62.62 **IoU**), (b) bottom-up method (65.65 **IoU**), (c) direct combination between top-down and bottom-up methods (68.55 **IoU**) and (d) our WiCo (71.74 **IoU**). We discover that the valid scope of the two methods is highly complementary. However, the direct combination leads to adverse cooperation. Our proposed WiCo mechanism adequately absorbs the advantages for better coping with those failure cases. These distributions are calculated on the RefCOCO val split.

turbed by PN errors and IP errors, we visualize the IoU distribution of top-down and bottom-up methods in Figure 2. We split the distribution curve into two parts: positive set (samples with IoU > 0.5) and negative set (samples with IoU < 0.5). As shown in Figure 2 (a), the positive set of top-down method achieves higher location precision than bottom-up method because the prior object information suppresses low-quality IP samples. But top-down method easily generates PN samples on negative samples due to the lack of fine-grained cross-modal alignment. In Figure 2 (b), bottom-up method outputs a large number of IP samples on positive set owing to the lack of prior object information. Nevertheless, the negative set of bottom-up methods has smoother distribution than top-down methods and the PN samples of bottom-up methods are far fewer than top-down methods because fine-grained information provides robust cross-modal alignment. According to the analysis above, we can conclude that top-down and bottom-up methods are highly complementary.

Intuitively, we can fuse top-down and bottom-up methods by direct combination, i.e., straightly averaging their results. However, as shown in Figure 2 (c), this scheme leads to harmful cooperation between top-down and bottom-up methods which can be attributed to the lack of feature interaction and the inappropriate integration of results. In this context, we build **Wi**n-win **Co**opEration (WiCo) to exploit the complementary nature of top-down and bottom-up branches by interacting with each other and integrating results of two branches in an adaptive manner, which follows "Interaction then Integration" paradigm for compensating the defect of direct combination. WiCo contains two modules: **Complementary Feature Interaction** (CFI) and **Gaussian Scoring Integration** (GSI). CFI is designed to perform interaction between features of two branches for compensating the lack of fine-grained information in top-down branch and prior object information in bottom-up branch. GSI is designed to model the gaussian performance distributions of top-down and bottom-up branches and adaptively integrate results of two branches by sampling confidence scores from the distributions. Figure 2 (d) shows that our framework largely reduces IP errors and PN errors and generates fine IoU distribution with the merits of top-down and bottom-up methods, which demonstrates our method is more effective than direct combination scheme for incorporating top-down and bottom-up methods.

In summary, the main contributions are as follows:

- We analyze the behavior of several top-down methods and bottom-up methods when facing PN and IP errors. According to the analysis, we discover that existing top-down and bottom-up methods are highly complementary in how to cope with PN errors and IP errors.
- We propose WiCo to adequately exploit the characteristics of top-down and bottom-up methods methods and let them effectively complement each other on both interaction and integration aspects, which can better process PN errors and IP errors than intuitive direct combination.
- Extensive experiments show that our WiCo can boost the performance of top-down and bottom-up methods methods by 2.25%~6.66% under three common datasets: *RefCOCO*, *RefCOCO+* and *G-Ref* with reasonable cost.

2 Related Work

Top-down Method. Previous efforts on top-down style referring image segmentation are about how to calculate better object-centric cross-modal alignment between region proposals of instances and referring expression query. For example, MAttNet [Yu *et al.*, 2018] decomposes referring expressions into three components to match instances. NMTree [Liu *et al.*, 2019] regularizes the cross-modal alignment along the dependency parsing tree of the sentence. CAC [Chen *et al.*, 2019b] introduces cycle consistency between referring expression and its reconstructed caption into the reasoning part of network for boosting cross-modal alignment.

Bottom-up Method. Previous efforts on bottom-up style referring image segmentation mainly focus on densely aligning and fusing visual and linguistic features for better cross-modal pixel features. For example, early works [Hu *et al.*, 2016a; Li *et al.*, 2018] propose to use simple concatenation to align and fuse visual feature maps and linguistic feature vectors, respectively. For replacing simple concatenation, some prior works use cross-modal attention to focus on important pixel regions and informative keywords for long-range cross-modal context [Shi *et al.*, 2018; Ye *et al.*, 2019; Chen *et al.*, 2019a]. Besides, some other works use complex visual reasoning to capture more explainable cross-modal context [Huang *et al.*, 2020; Hui *et al.*, 2020; Yang

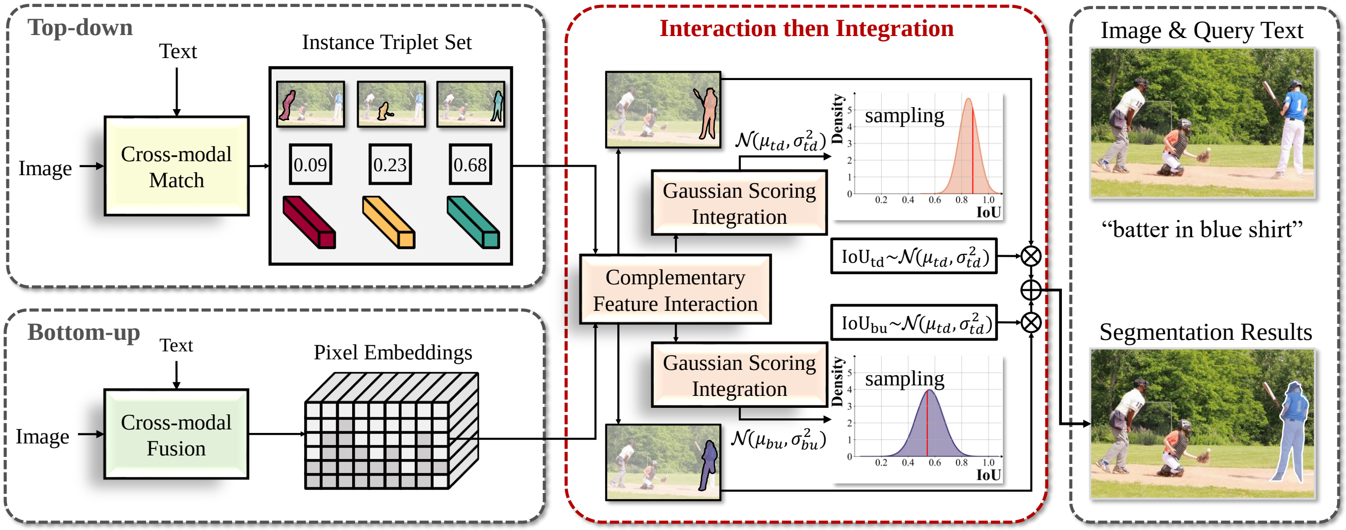


Figure 3: **The overall pipeline of our WiCo.** Firstly, top-down and bottom-up branches acquire the respective features and results. Then, these features and results are input into CFI (Complementary Feature Interaction) for knowledge interaction. Finally, we use GSI (Gaussian Scoring Integration) to predict the performance distributions of two branches and weighted integrate the results of two branches according to the confidence score sampled from the performance distributions. The modules inside the **red dashed box** are our main contribution.

et al., 2021]. Recently, Vision transformer (ViT) [Dosovitskiy *et al.*, 2020] has been proposed as a new visual network paradigm. Due to its compatibility with multi-modal data, some works use it to jointly encode visual and linguistic features for intensive cross-modal alignment [Ding *et al.*, 2021; Li and Sigal, 2021; Wang *et al.*, 2022; Yang *et al.*, 2022].

3 Methods

3.1 Overall Pipeline

To ensure the generality of our framework, the WiCo is designed to be compatible with arbitrary top-down and bottom-up methods. As shown in Figure 3, WiCo has three parts: top-down branch, bottom-up branch and "Interaction then Integration". Top-down branch is used to deploy top-down methods. Bottom-up branch is used to equip bottom-up methods. "Interaction then Integration" is the key component of WiCo which is used to build cooperation between top-down and bottom-up branches for achieving a win-win improvement.

Top-down style methods are essentially a cross-modal match network [Yu *et al.*, 2018]. It uses the pretrained detector and cross-modal match network to obtain instance masks $\mathcal{M} = \{m^1 \in \{0,1\}^{H\times W}, m^2, ..., m^n\}$, cross-modal instance embeddings $\mathcal{E} = \{E_i^1 \in \mathbb{R}^C, E_i^2, ..., E_i^n\}$ and cross-modal alignment scores $\mathcal{S} = \{s^1, s^2, ..., s^n\}$. In general, top-down branch outputs a instance triplet set $\{\mathcal{M}, \mathcal{E}, \mathcal{S}\} = \{(m^1, E_i^1, s^1), (m^2, E_i^2, s^2), ..., (m^n, E_i^n, s^n)\}$. Extracting segmentation results P_{td} from triplet set is fomulated as:

$$P_{td} = m^{\texttt{argmax}(\mathcal{S})} * s^{\texttt{argmax}(\mathcal{S})}, \tag{1}$$

where P_{td} is the segmentation logits results. The binary segmentation results are $m^{\texttt{argmax}(\mathcal{S})}$.

Bottom-up methods are essentially a cross-modal fusion network [Hu *et al.*, 2016b]. It uses a cross-modal fuse network to jointly encode images and texts to cross-modal pixel embeddings $E_p \in \mathbb{R}^{C\times H\times W}$ and decode cross-modal pixel embeddings to segmentation results $P_{bu} \in \mathbb{R}^{H\times W}$. Decoding cross-modal pixel embeddings into segmentation results is formulated as:

$$P_{bu} = \sigma(\texttt{Linear}(E_p)), \tag{2}$$

where $\texttt{Linear}(\cdot)$ denotes 1x1 convolution for logit regression and $\sigma(\cdot)$ is sigmoid function. P_{bu} is the probability map and the binary segmentation results are extracted from it by threshold τ ($P_{bu} > \tau$). In general, bottom-up branch outputs cross-modal pixel embeddings and segmentation results.

"Interaction then Integration" is designed to exploit the complementary nature of top-down and bottom-up methods. To complement on interaction aspect, the outputs of bottom-up branch and top-down branch are input into CFI (Section 3.2) for updating features and results. To complement on integration aspect, the updated results are input into GSI (Section 3.3) to integrate results.

3.2 Complementary Feature Interaction

The detailed calculation flow is illustrated in Figure 4. Suppose that we already acquire pixel embeddings E_p from bottom-up branch and instance triplet set $\{\mathcal{M}, \mathcal{E}, \mathcal{S}\}$ from top-down branch, we hope that CFI can let the fine-grained information of pixel embeddings and object-centric information of instance triplet set enhance each other.

Top-down for Bottom-up. For enhancing pixel embeddings E_p, we assign object-centric information of each enhanced instance embeddings $\hat{\mathcal{E}}$ to corresponding pixels according to the instance masks $\mathcal{M}$ and concatenate these instance embeddings with raw pixel embeddings to generate enhanced pixel embeddings $\hat{E}_p$:

$$\hat{E}_p^{\{x,y\}} = \texttt{concat}(E_p^{\{x,y\}}; \sum_{j=1}^{n} \mathbb{1}_{\{m^j[x,y]=1\}} \hat{E}_i^j), \tag{3}$$

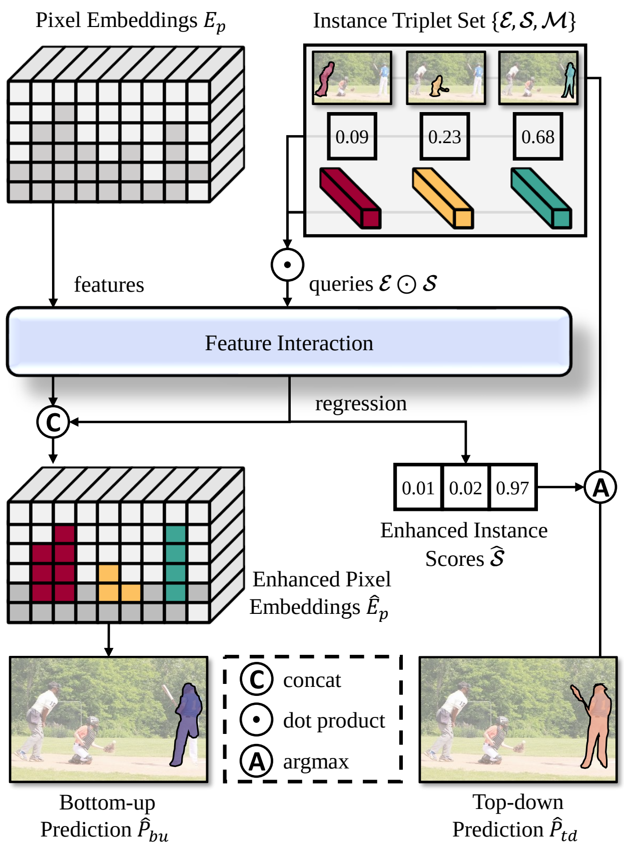


Figure 4: **Complementary Feature Interaction.** The modulated instance embeddings $\mathcal{E} \odot \mathcal{S}$ and pixel embeddings E_p are input into the "Feature Interaction" for generating enhanced instance embeddings $\hat{\mathcal{E}}$. Enhanced instance embeddings are used to predict enhanced alignment scores $\hat{\mathcal{S}}$ for generating new top-down segmentation results $\hat{P}_{td}$. The enhance instance embeddings are also assigned to corresponding pixels of pixel embeddings for enhancing pixel embeddings $\hat{E}_p$ and generating new bottom-up segmentation $\hat{P}_{bu}$.

where $E_p^{\{x,y\}}$ denotes enhanced pixel embeddings at (x, y) pixel location and E_i^j is the enhanced instance embeddings of i-th instance. $\mathbb{1}_{\{m^j[x,y]=1\}}$ is the indicator function, where it is equal to 1 when the (x, y) pixel location of j-th mask is 1, and 0 otherwise. The enhanced pixel embeddings are then decoded to new bottom-up results:

$$\hat{P}_{bu} = \texttt{sigmoid}(\texttt{Linear}(\hat{E}_p)), \tag{4}$$

where the $\texttt{Linear}(\cdot)$ shares same weights with Eq. 2.

Bottom-up for Top-down. For enhancing instance embeddings $\mathcal{E}$, we use vision transformer decoder [Cheng *et al.*, 2021] as "Feature Interaction" module to refine the instance embeddings by fine-grained information of pixel embeddings E_p. Before inputting, the instance embeddings are modulated by cross-modal alignment scores $\mathcal{S}$ for preserving cross-modal information:

$$\mathcal{E} \odot \mathcal{S} = \{E_p^1 * s^1, E_p^2 * s^2, ..., E_p^n * s^n\}. \tag{5}$$

Then, transformer decoder sets these modulated instance embeddings $\mathcal{E} \odot \mathcal{S}$ as queries to generate enhanced instance embeddings $\hat{\mathcal{E}}$ and predict enhanced alignment scores $\hat{\mathcal{S}}$. With new alignment scores, we can update the segmentation results of top-down branch:

$$\hat{P}_{td} = m^{\texttt{argmax}(\hat{\mathcal{S}})} * \hat{s}^{\texttt{argmax}(\hat{\mathcal{S}})}. \tag{6}$$

3.3 Gaussian Scoring Integration

After obtaining top-down results $\hat{P}_{td}$ and bottom-up results $\hat{P}_{bu}$, we use GSI to integrate them for generating more robust and higher-performance results. GSI has three steps: Distribution Prediction, Score Sampling and Results Blend. The details of three steps are introduced below:

Distribution Prediction. Because of the uncertainty, we set the performance score as a latent variable following a specific distribution. Due to the excellent computability, we use gaussian distribution to model the performance distribution [Kingma and Welling, 2013]. For representing gaussian distribution, we predict the mean μ and standard deviation σ according to the results and features of two branches:

$$\mu_{td}, \sigma_{td} = \texttt{split}(\texttt{MLP}(\hat{E}_i^{\texttt{argmax}(\hat{\mathcal{S}})}), \tag{7}$$

$$\mu_{bu}, \sigma_{bu} = \texttt{split}(\texttt{MLP}(\texttt{GAP}(E_p \odot \hat{P}_{bu}))), \tag{8}$$

where $\texttt{MLP}(\cdot)$ denotes 3 fully connected layers, $\texttt{GAP}(\cdot)$ denotes global average pooling operation and $\texttt{split}(\cdot)$ denotes channel split operation. With predicted mean and standard deviation, we obtain the performance distribution of bottom-up and top-down branches, i.e., $\mathcal{N}(\mu_{bu}, \sigma_{bu})$ and $\mathcal{N}(\mu_{td}, \sigma_{td})$.

Score Sampling. Performance distribution indicates the confidence score range of prediction. We sample a value from the performance distribution as the detailed confidence score of this prediction. For differentiable optimization, we utilize re-parameterization trick [Kingma and Welling, 2013] to modify the sampling process:

$$\begin{aligned} \mathrm{IoU}_{td} &= \mu_{td} + \sigma_{td} * \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}), \\ \mathrm{IoU}_{bu} &= \mu_{bu} + \sigma_{bu} * \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}), \end{aligned} \tag{9}$$

where IoU_{td} and IoU_{bu} denote confidence score of top-down and bottom-up branch results. For optimizing the distribution prediction model, we calculate smooth-l1 loss between predicted confidence score and ground truth IoU value.

Results Blend. Note that $\texttt{argmax}(\cdot)$ is essentially a non-differentiable operation during gradient backward, we adopt a differentiable implementation [van den Oord *et al.*, 2017] of $\texttt{argmax}(\cdot)$ function during training pharse:

$$\lambda = \texttt{one-hot}(\texttt{argmax}(\hat{\mathcal{S}})) + \hat{\mathcal{S}} - \texttt{sg}(\hat{\mathcal{S}}), \tag{10}$$

where $\lambda \in \{0, 1\}^n$ is a binary vector to indicate the index of the max value, $\texttt{one-hot}(\cdot)$ is the one-hot encoding function and $\texttt{sg}(\cdot)$ is the stop gradient operation. The λ is used to build differentiable segmentation results of top-down branch $\hat{P}'_{td}$:

$$\hat{P}'_{td} = \sum_{j=1}^{n} m^j * \lambda^j * s^j, \tag{11}$$

where n is the number of instances. For generating final segmentation results, we use confidence scores to calculate a weighted sum results of top-down and bottom-up branches:

$$\hat{P} = (\hat{P}'_{td} * \mathrm{IoU}_{td} + \hat{P}_{bu} * \mathrm{IoU}_{bu})/2. \tag{12}$$

The final results $\hat{P}$ are used to calculate segmentation loss with ground truth mask during training phrase and are decoded to binary mask by threshold τ during inference phrase.

Method	Type	RefCOCO			RefCOCO+			RefCOCOg
		val	test A	test B	val	test A	test B	val
MAttNet [Yu *et al.*, 2018]	TD	56.51	62.37	51.70	46.67	52.39	40.08	-
NMTree [Liu *et al.*, 2019]	TD	56.59	63.02	52.06	47.40	53.01	41.56	-
CAC [Chen *et al.*, 2019b]	TD	58.90	61.77	53.81	-	-	-	44.32
MCN [Luo *et al.*, 2020b]	BU	62.44	64.20	59.71	50.62	54.99	44.69	-
CMPC [Huang *et al.*, 2020]	BU	61.36	64.53	59.64	49.56	53.44	43.23	39.98
LSCM [Hui *et al.*, 2020]	BU	61.47	64.99	59.55	49.34	53.12	43.50	48.05
CGAN [Luo *et al.*, 2020a]	BU	64.86	68.04	62.07	51.03	55.51	44.06	46.54
BUSNet [Yang *et al.*, 2021]	BU	62.56	65.61	60.38	50.98	56.14	43.51	49.98
EFN [Feng *et al.*, 2021]	BU	62.76	65.69	59.67	51.50	55.24	43.01	-
LTS [Jing *et al.*, 2021]	BU	65.43	67.76	63.08	54.21	58.32	48.02	-
VLT [Ding *et al.*, 2021]	BU	65.65	68.29	62.73	55.50	59.20	49.36	49.76
ResTR [Kim *et al.*, 2022]	BU	67.22	69.30	64.45	55.78	60.44	48.27	-
CRIS [Wang *et al.*, 2022]	BU	69.52	72.72	64.70	61.39	67.10	52.48	55.77*
LAVT [Yang *et al.*, 2022]	BU	72.73	75.82	68.79	62.14	68.38	55.10	60.50
SeqTR [Zhu *et al.*, 2022]	BU	67.26	69.79	64.12	54.14	58.93	48.19	-
CoupleAlign [Zhang *et al.*, 2022]	BU	74.70	77.76	70.58	62.92	68.34	56.69	-
WiCo VLT + MAttNet	TD+BU	71.74 ↑ 6.09	74.07 ↑ 5.78	67.23 ↑ 4.50	60.17 ↑ 4.67	65.15 ↑ 5.95	53.55 ↑ 4.19	53.37 ↑ 3.61
WiCo CRIS + MAttNet	TD+BU	73.46 ↑ 3.94	76.95 ↑ 4.23	68.08 ↑ 3.38	63.42 ↑ 2.03	69.17 ↑ 2.07	55.76 ↑ 3.28	60.17 ↑ 4.4
WiCo LAVT + MAttNet	TD+BU	**75.50** ↑ 6.66	**78.07** ↑ 2.25	**71.30** ↑ 2.51	**65.75** ↑ 3.61	**70.52** ↑ 2.14	**57.14** ↑ 2.04	**61.27** ↑ 0.77

Table 1: **Main results** on three classical datasets (RefCOCO, RefCOCO+ and RefCOCOg). "TD" denotes top-down methods. "BU" denotes bottom-up methods. The **improvement** is calculated based on bottom-up method. * denotes the results are re-implemented by us.

4 Experiments

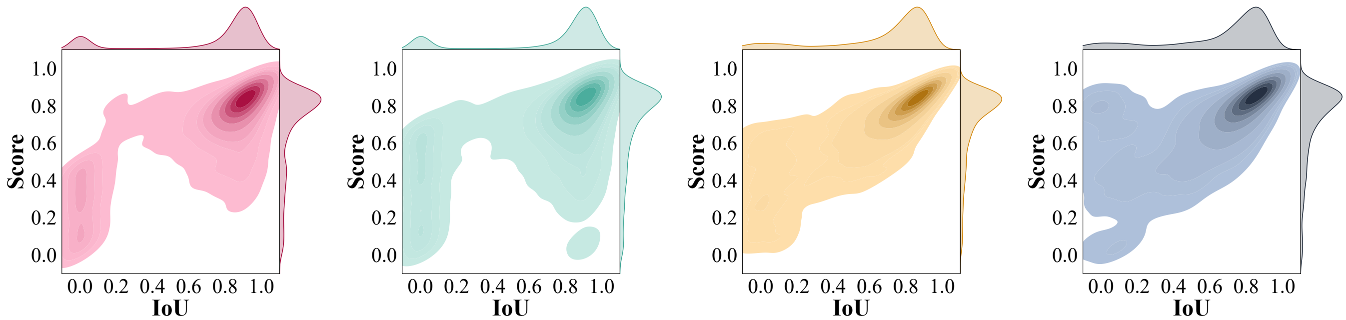
4.1 Experimental Setup

Our model is evaluated on three standard referring image segmentation datasets: RefCOCO [Yu *et al.*, 2016], RefCOCO+ [Yu *et al.*, 2016] and RefCOCOg [Mao *et al.*, 2016]. For top-down branch, MAttNet [Yu *et al.*, 2018] is selected as the main equipment due to its simple structure and effectiveness. As for the bottom-up branch, several advanced and representative methods are selected, e.g., VLT [Ding *et al.*, 2021], CRIS [Wang *et al.*, 2022] and LAVT [Yang *et al.*, 2022], to show the effectiveness and generality of our method. The data preprocessing operations are in line with the original implementation of those selected methods. Because MAttNet is an early method that has an obsolete instance extractor, Mask2Former [Cheng *et al.*, 2021] (ResNet-50) is adopted as an instance extractor to compensate for the top-down branch to avoid the cask effect, which improves the performance of MAttNet from 56.51 to 62.62 on RefCOCO val set. Based on previous works [Luo *et al.*, 2020b; Ding *et al.*, 2021], mask IoU is adopted to evaluate the performance of methods. To reduce the training cost, the selected models are initialized by pretrained weights and just finetune when inserting them into our framework. AdamW [Loshchilov and Hutter, 2017] is adopted as our optimizer, and the learning rate and weight decay are set to 1e-5 and 5e-2. We train our models for 5,000 iterations on an NVIDIA V100 with a batch size of 24. To binarize the probability map and get segmentation results, the threshold τ is set to 0.35 to calibrate previous works [Ding *et al.*, 2021].

4.2 Quantitative Analysis

Main Results. Table 1 reports the comparison results between our method and previous state-of-the-art methods in three common datasets, i.e., RefCOCO, RefCOCO+ and RefCOCOg. Some top-down and bottom-up methods that are easy to reproduce are selected for benchmark. Specifically, there are three combinations, i.e., VLT + MAttNet, CRIS + MAttNet and LAVT + MAttNet. Because bottom-up methods are mainstream methods, we mainly describe the performance improvement based on bottom-up methods in Table 1. Utilizing WiCo to incorporate these three model combinations, the fusion results improve the results of VLT, CRIS and LAVT by 6.09%, 3.94% and 6.66% on RefCOCO val split, 5.78%, 4.23%, 2.25% on RefCOCO testA split and 4.5%, 3.38% and 2.51% on RefCOCO testB split. Other datasets also consistently show the performance improvements of our method over the selected baseline models.

Different Results Integration Strategies. In Table 2, we attempt different results integration strategies and check if these integration strategies can boost the integration results of top-down and bottom-up branches. In terms of results integration strategies, GSI is compared to three straight strategies, i.e., "Intersection", "Union", and "Average". Although these strategies improve the performance of top-down and bottom-up methods, our proposed GSI still performs better than them, indicating that GSI provides a more perceptive and robust way to integrate results. Moreover, we build an abbreviated version of GSI to check the effectiveness of the gaussian distribution-based performance modeling, i.e., Scoring



(a) GSI (Top-down) (b) SI (Top-down) (c) GSI (Bottom-up) (d) SI (Bottom-up)

Figure 5: **Correlation between predicted confidence score and IoU.** The density map of samples from GSI and SI. Darker area indicates more samples are of the corresponding IoU (%) value and confidence score. "SI" is Scoring Integration, abbreviated from GSI by removing the gaussian distribution-based performance modeling. Marginal plots denote the distribution of confidence score and IoU.

Integration	Interaction	IoU_α	IoU_β	IoU_+
Intersection	-	65.65	62.62	63.85 $\downarrow$ 4.70
Union	-			67.79 $\downarrow$ 0.76
Average	-			68.55
SI	-	65.65	62.62	68.95 $\uparrow$ 0.40
GSI	-			69.63 $\uparrow$ 1.08
SI	CFI	68.07 $\uparrow$ 2.42	65.34 $\uparrow$ 2.72	70.51 $\uparrow$ 1.96
GSI	CFI			**71.74** $\uparrow$ 3.19

Table 2: **Diagnostic Experiments.** IoU_α, IoU_β and IoU_+ denotes the IoU of model α (VLT), model β (MAttNet) and integration results, respectively. "Intersection", "Union" and "Average" means taking the intersection, union and average of the top-down and bottom-up results as the fusion result. "SI" is Scoring Integration, abbreviated from GSI by removing the gaussian distribution-based performance modeling. "Average" scheme is set as the baseline for comparison.

model α+β	IoU_α+IoU_β	IoU_+	Speed
VLT$^\clubsuit$+CRIS$^\clubsuit$	65.65+69.52	70.15 $\uparrow$ 2.57	6.63 2.17+3.72+0.74
VLT$^\clubsuit$+LAVT$^\clubsuit$	65.65+72.73	73.05 $\uparrow$ 3.86	10.2 2.17+7.27+0.74
CRIS$^\clubsuit$+LAVT$^\clubsuit$	69.52+72.73	73.87 $\uparrow$ 2.75	11.7 3.72+7.27+0.74
CAC$^\heartsuit$+MAttNet$^\heartsuit$	63.43+62.62	62.72 $\downarrow$ 0.31	4.43 2.11+1.88+0.44
VLT$^\clubsuit$+MAttNet$^\heartsuit$	65.65+62.62	69.63 $\uparrow$ 5.49	4.64 2.17+1.88+0.59
CRIS$^\clubsuit$+MAttNet$^\heartsuit$	69.52+62.62	73.01 $\uparrow$ 6.94	6.19 3.72+1.88+0.59
LAVT$^\clubsuit$+MAttNet$^\heartsuit$	72.73+62.62	**74.33** $\uparrow$ 6.66	9.74 7.27+1.88+0.59

Table 3: **The performance of different model combinations.** For checking the complementary effect between different models, model α and model β are integrated by only GSI. $^\clubsuit$ and $^\heartsuit$ denote bottom-up style methods and top-down style methods, respectively. Inference speed is acquired by counting the inference seconds of 100 samples. The **increase** value and decrease value are calculated by subtracting $(\text{IoU}_\alpha + \text{IoU}_\beta)/2$ from IoU_+, i.e., $\text{IoU}_+ - (\text{IoU}_\alpha + \text{IoU}_\beta)/2$.

Integration (SI). Based on the experiment results that the GSI performs 0.68% better than SI, it is concluded that the gaussian distribution-based performance modeling makes sense.

Effect of Feature Interaction. Feature interaction boosts results by improving the respective results of top-down and bottom-up branches. For diagnosing if feature interaction is beneficial for final results, we conduct comparison experiments of WiCo with CFI and without CFI. As shown in Table 2, WiCo with CFI improves baseline by 3.19% and performs 2.11% better than WiCo without CFI. The experiment results show that CFI effectively improves top-down and bottom-up branches by 2.42% and 2.72%. Moreover, it also shows that feature interaction (CFI) boosts final performance on a different aspect than results integration (GSI).

Complementary Effect of Different Model Combinations. Three kinds of combinations are constructed (bottom-up + bottom-up, top-down + top-down and bottom-up + top-down) to check the complementary effect of different model combinations in Table 3. The model combination with two same kinds of models is defined as "**Homogeneous**" combination. On the contrary, the model combination with two different kinds of models is defined as "**Heterogeneous**" combination. As shown in Table 3, the experimental results can be split into three parts: bottom-up homogeneous combinations (VLT+CRIS, VLT+LAVT and CRIS+LAVT), top-down homogeneous combinations (MAttNet+CAC), and heterogeneous combinations between bottom-up and top-down methods (VLT+MAttNet, CRIS+MAttNet and LAVT+MAttNet). Three bottom-up homogeneous combinations only improve original models by 2.57%, 3.86%, 2.75% and the top-down homogeneous combination even degrade origin models by 0.31%. However, three heterogeneous combinations consistently improve original models by a clear margin (5.49%, 6.94%, 6.66%). These results indicate that heterogeneous combinations have a stronger complementary effect than homogeneous combinations for boosting performance. In order to quantify the complementary effect, "**Mutually Exclusive Rate**" (MER) is defined as a metric for analyzing. MER denotes the rate of samples in which only one of the top-down and bottom-up branches outputs positive prediction (IoU > 0.5). In Figure 7, the MER of heterogeneous combinations is significantly higher than homogeneous combinations. These statistics results explain why the performance improvement of heterogeneous combinations is also remarkably higher than homogeneous combinations.

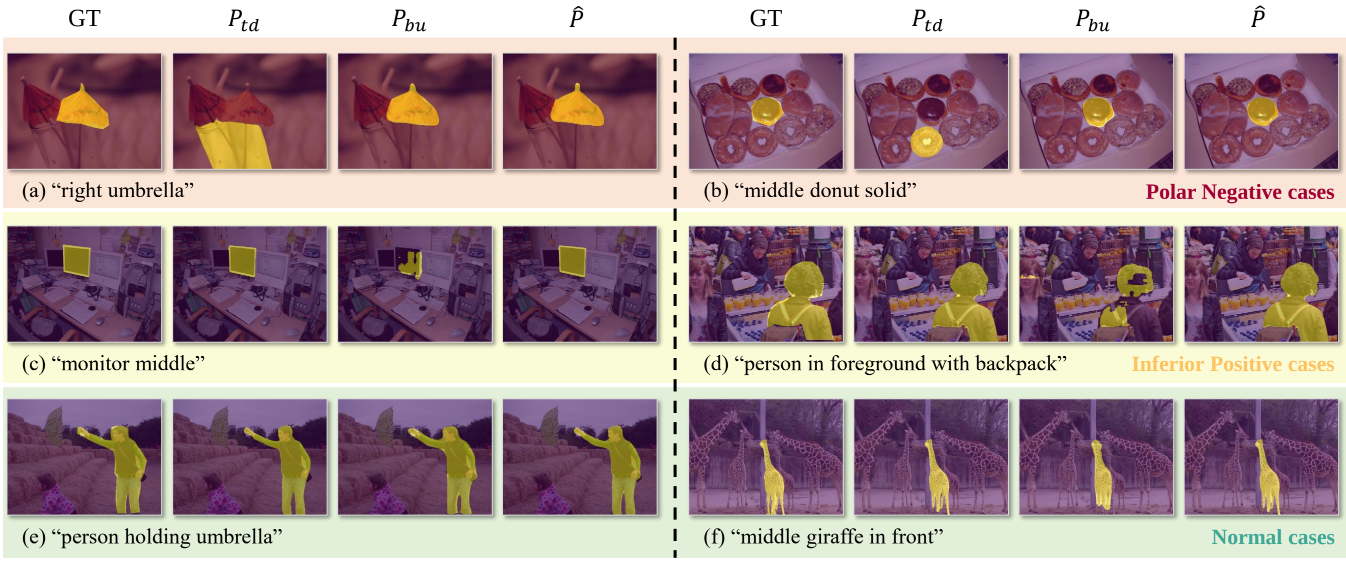


Figure 6: **Qualitative segmentation reults of different cases.** "GT", P_{td}, P_{bu} and $\hat{P}$ denotes ground truth, original results of bottom-up branch, original results of top-down branch and the integration results of two branches. There are totally three types of cases selected for showing the effectiveness of our WiCo The first, second and third rows are polar negative cases, inferior positive cases and normal cases.

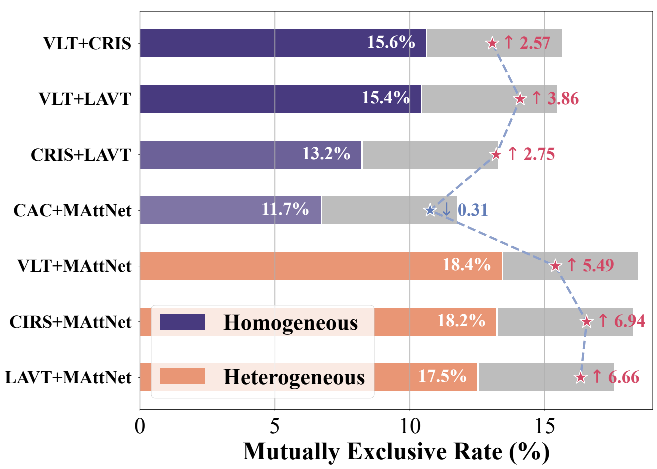


Figure 7: **The quantified analysis of complementary effect.** This figure is corresponding to Table 3. **Mutually Exclusive Rate** (MER) denotes the rate of samples in which only a single branch outputs positive prediction (IoU > 0.5). A higher MER denotes the outputs of top-down and bottom-up branches are more complementary.

4.3 Qualitative Analysis

The key reason why GSI and SI achieve better performance than straight schemes is that they can roughly estimate the confidence of outputs to adaptively integrate the results. For checking the precision of confidence regression, we plot the correlation between the predicted confidence score and real evaluation score (IoU) in Figure 5. Whether the regression target is the performance score of top-down or bottom-up results, the plot results show that the confidence score predicted by GSI presents a more linear correlation with the evaluation score than SI, which demonstrates the gaussian distribution-based performance modeling of GSI is significant.

Some representative samples of three cases (polar negative cases, inferior positive cases and normal cases) are selected to justify the refinement for PN and IP errors. In Figure 6, first and second rows clearly depict the integration results of WiCo fix the obvious errors of original top-down and bottom-up results, which demonstrates the effectiveness of our method. In Figure 6, third row also shows that our method can adaptively fetch better segmentation results from two branches.

5 Conclusion

Existing top-down and bottom-up methods fail to handle PN and IP errors. Nevertheless, top-down and bottom-up methods can complement each other's flaws for better processing PN and IP errors according to our analysis. To fully exploit the complementary nature, we follow a "Interaction then Integration" paradigm to build WiCo mechanism for achieving a win-win improvement. Specifically, CFI is proposed to let the prior object information of top-down branch and fine-grained information of bottom-up branch interact with each other for feature enhancement. GSI is designed to model the performance distributions of two branches for adaptively integrating results of two branches. We select some prominent top-down and bottom-up methods to equip our WiCo for experiments. The experiments consistently show that our WiCo can improve both top-down and bottom-up methods by a clear margin, which demonstrates the effectiveness of our methods.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No. 2022ZD0118201), the Natural Science Foundation of China (No. 61972217, 32071459, 62176249, 62006133, 62271465), the Natural Science Foundation of Guangdong Province in China (No. 2019B1515120049).

References

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

[Chen *et al.*, 2019b] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency. *arXiv preprint arXiv:1910.04748*, 2019.

[Cheng *et al.*, 2021] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021.

[Ding *et al.*, 2021] Henghui Ding, Chang Liu, Suchen Wang, and Xudong 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.

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

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

[Hu *et al.*, 2016a] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016.

[Hu *et al.*, 2016b] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4555–4564, 2016.

[Huang *et al.*, 2020] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10488–10497, 2020.

[Hui *et al.*, 2020] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *European Conference on Computer Vision*, pages 59–75. Springer, 2020.

[Jing *et al.*, 2021] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu 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.

[Kim *et al.*, 2022] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. Restr: Convolution-free referring image segmentation using transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18145–18154, 2022.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Li and Sigal, 2021] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. *Advances in Neural Information Processing Systems*, 34:19652–19664, 2021.

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

[Liu *et al.*, 2019] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4673–4682, 2019.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Luo *et al.*, 2020a] Gen Luo, Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jinsong Su, Chia-Wen Lin, and Qi Tian. Cascade grouped attention network for referring expression segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1274–1282, 2020.

[Luo *et al.*, 2020b] 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 *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 10034–10043, 2020.

[Mao *et al.*, 2016] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[Shi *et al.*, 2018] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–54, 2018.

[van den Oord *et al.*, 2017] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Wang *et al.*, 2022] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11686–11695, 2022.

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

[Yang *et al.*, 2022] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165, 2022.

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

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

[Yu *et al.*, 2018] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018.

[Zhang *et al.*, 2022] Zicheng Zhang, Yi Zhu, Jianzhuang Liu, Xiaodan Liang, and Wei Ke. Coupalign: Coupling word-pixel with sentence-mask alignments for referring image segmentation. *arXiv preprint arXiv:2212.01769*, 2022.

[Zhu *et al.*, 2022] 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. *arXiv preprint arXiv:2203.16265*, 2022.