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 Win-win Cooperation (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.

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→0). IP samples are those predictions that ignore some components of the referred object (IoU∈[0.5,0.8]).

To analyze how top-down and bottom-up methods are dis-
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 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 by $2.25\%$ to $6.66\%$ 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. NMTTree [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...
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 \( M = \{ m^1 \in \{0, 1\}^{H \times W}, m^2, ..., m^n \} \), cross-modal instance embeddings \( E = \{ E_1^1 \in \mathbb{R}^C, E_2^2, ..., E^n_n \} \) and cross-modal alignment scores \( S = \{ s^1, s^2, ..., s^n \} \). In general, top-down branch outputs a instance triplet set \( \{ M, E, S \} = \{ (m^1, E_1^1, s^1), (m^2, E_2^2, s^2), ..., (m^n, E^n_n, s^n) \} \). Extracting segmentation results \( P_{td} \) from triplet set is formulated as:

\[
P_{td} = m_{\text{argmax}}(S) \times s_{\text{argmax}}(S),
\]

where \( P_{td} \) is the segmentation logits results. The binary segmentation results are \( m_{\text{argmax}}(S) \).

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:

\[
P_{bu} = \sigma(\text{Linear}(E_p)),
\]

where \( \text{Linear()} \) 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 \( \tau (P_{bu} > \tau) \). 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 \( \{ M, E, S \} \) 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 \( \hat{E} \) to corresponding pixels according to the instance masks \( M \) and concatenate these instance embeddings with raw pixel embeddings to generate enhanced pixel embeddings \( \hat{E}_p \):

\[
\hat{E}_p^{(x,y)} = \text{concat}(E_p^{(x,y)}, \sum_{j=1}^{n} I_{m^{[1]}=1}(x_j, y_j)\hat{E}_j^{(x,y)}),
\]

where \( I \) is the indicator function and \( \sum_{j=1}^{n} I_{m^{[1]}=1}(x_j, y_j) \) is the sum of all the instances that are present in the mask.

**Bottom-up for Top-down.** For enhancing bottom-up features, we use Gaussian Scoring Integration 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.
new alignment scores, we can update the segmentation results of top-down branch:
\[
\hat{P}_{td} = m_{\text{argmax}}(\hat{S}) \ast \hat{S}_{\text{argmax}}(\hat{S}).
\] (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 \(\mu\) and standard deviation \(\sigma\) according to the results and features of two branches:
\[
\mu_{td}, \sigma_{td} = \text{split}(\text{MLP}(\hat{E}_{\text{argmax}}(\hat{S}))),
\] (7)
\[
\mu_{bu}, \sigma_{bu} = \text{split}(\text{MLP}(\text{GAP}(E_{\text{p}} \odot \hat{P}_{bu}))).
\] (8)

where \(\text{MLP}()\) denotes 3 fully connected layers, \(\text{GAP}()\) denotes global average pooling operation and \(\text{split}()\) 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:
\[
\text{IoU}_{td} = \mu_{td} + \sigma_{td} \ast \epsilon, \epsilon \sim \mathcal{N}(0, I),
\] (9)
\[
\text{IoU}_{bu} = \mu_{bu} + \sigma_{bu} \ast \epsilon, \epsilon \sim \mathcal{N}(0, I),
\] (10)
where \(\text{IoU}_{td}\) and \(\text{IoU}_{bu}\) denote confidence score of top-down and bottom-up branch results. For optimizing the distribution prediction model, we calculate smooth-1L loss between predicted confidence score and ground truth IoU value.

**Results Blend.** Note that \(\text{argmax}()\) is essentially a non-differentiable operation during gradient backward, we adopt a differentiable implementation [van den Oord et al., 2017] of \(\text{argmax}()\) function during training phase:
\[
\lambda = \text{one-hot}(\text{argmax}(\hat{S})) + \hat{S} - \text{sg}(\hat{S}),
\] (11)
where \(\lambda \in [0, 1]^n\) is a binary vector to indicate the index of the max value, \(\text{one-hot}()\) is the one-hot encoding function and \(\text{sg}()\) is the stop gradient operation. The \(\lambda\) is used to build differentiable segmentation results of top-down branch \(\hat{P}_{td}\):
\[
\hat{P}_{td} = \sum_{j=1}^{n} m_{ij} \ast \lambda_{ij} \ast s_{ij},
\] (12)
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:
\[
\hat{P} = (\hat{P}_{td} \ast \text{IoU}_{td} + \hat{P}_{bu} \ast \text{IoU}_{bu})/2.
\] (13)

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 \(\tau\) during inference phrase.
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 \( \tau \) 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% and 5.78% on RefCOCO+ val split and 5.95%, 4.23% and 5.78% on RefCOCOg. 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.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>RefCOCO</th>
<th>RefCOCO+</th>
<th>RefCOCOg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>val</td>
<td>test A</td>
<td>test B</td>
</tr>
<tr>
<td>MAttNet</td>
<td>TD</td>
<td>56.51</td>
<td>62.37</td>
<td>51.70</td>
</tr>
<tr>
<td>NMTree</td>
<td>TD</td>
<td>56.59</td>
<td>63.02</td>
<td>52.06</td>
</tr>
<tr>
<td>CAC</td>
<td>TD</td>
<td>58.90</td>
<td>61.77</td>
<td>53.81</td>
</tr>
<tr>
<td>MCN</td>
<td>BU</td>
<td>62.44</td>
<td>64.20</td>
<td>59.71</td>
</tr>
<tr>
<td>CMPC</td>
<td>BU</td>
<td>61.36</td>
<td>64.53</td>
<td>59.64</td>
</tr>
<tr>
<td>LSCM</td>
<td>BU</td>
<td>61.47</td>
<td>64.99</td>
<td>59.55</td>
</tr>
<tr>
<td>CGAN</td>
<td>BU</td>
<td>64.86</td>
<td>68.04</td>
<td>62.07</td>
</tr>
<tr>
<td>BUSNet</td>
<td>BU</td>
<td>62.56</td>
<td>65.61</td>
<td>60.38</td>
</tr>
<tr>
<td>EFN</td>
<td>BU</td>
<td>62.76</td>
<td>65.69</td>
<td>59.67</td>
</tr>
<tr>
<td>LTS</td>
<td>BU</td>
<td>65.43</td>
<td>67.76</td>
<td>63.08</td>
</tr>
<tr>
<td>VLT</td>
<td>BU</td>
<td>65.65</td>
<td>68.29</td>
<td>62.73</td>
</tr>
<tr>
<td>ResTR</td>
<td>BU</td>
<td>67.22</td>
<td>69.30</td>
<td>64.45</td>
</tr>
<tr>
<td>CRIS</td>
<td>BU</td>
<td>69.52</td>
<td>72.72</td>
<td>64.70</td>
</tr>
<tr>
<td>SeqTR</td>
<td>BU</td>
<td>72.73</td>
<td>75.82</td>
<td>68.79</td>
</tr>
<tr>
<td>CoupleAlign</td>
<td>BU</td>
<td>74.70</td>
<td>77.76</td>
<td>70.58</td>
</tr>
</tbody>
</table>

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.
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 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 experiments 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.

**Table 2:** Diagnostic Experiments. IoUα, IoUβ and IoUα denote 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.

**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 * and * 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 (IoUα + IoUβ)/2 from IoUα, i.e., IoUα = (IoUα + IoUβ)/2.

![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.](image)
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).


