Rethinking Diversified and Discriminative Proposal Generation for Visual Grounding

Zhou Yu¹, Jun Yu¹,∗, Chenchao Xiang¹, Zhou Zhao², Qi Tian³, Dacheng Tao⁴
¹ School of Computer Science and Technology, Hangzhou Dianzi University, P. R. China
² College of Computer Science, Zhejiang University, P. R. China
³ Department of Computer Science, University of Texas at San Antonio, USA
⁴ UBTECH Sydney AI Centre, SIT, FEIT, University of Sydney, Australia
{yuz, yujun, ccxiang}@hdu.edu.cn, zhaozhou@zju.edu.cn,
qi.tian@utsa.edu, dacheng.tao@sydney.edu.au

Abstract

Visual grounding aims to localize an object in an image referred to by a textual query phrase. Various visual grounding approaches have been proposed, and the problem can be modularized into a general framework: proposal generation, multi-modal feature representation, and proposal ranking. Of these three modules, most existing approaches focus on the latter two, with the importance of proposal generation generally neglected. In this paper, we rethink the problem of what properties make a good proposal generator. We introduce the diversity and discrimination simultaneously when generating proposals, and in doing so propose Diversified and Discriminative Proposal Networks model (DDPN). Based on the proposals generated by DDPN, we propose a high performance baseline model for visual grounding and evaluate it on four benchmark datasets. Experimental results demonstrate that our model delivers significant improvements on all the tested data-sets (e.g., 18.8% improvement on ReferItGame and 8.2% improvement on Flickr30k Entities over the existing state-of-the-arts respectively).

1 Introduction

Recent advances in deep neural networks have helped to solve many challenges in computer vision and natural language processing. These advances have also stimulated high-level research into the connections that exist between vision and language such as visual captioning [Donahue et al., 2015; Xu et al., 2015], visual question answering [Fukui et al., 2016; Yu et al., 2017b; 2018] and visual grounding [Rohrbach et al., 2016; Chen et al., 2017].

∗Jun Yu is the corresponding author

Figure 1: The general framework of the visual grounding task. Given an arbitrary image and an open-ended query phrase as the inputs, the visual grounding model outputs the predicted bounding box of the referred object.

Visual grounding (a.k.a., referring expressions) aims to localize an object in an image referred to by a textual query phrase. It is a challenge task that requires a fine-grained understanding of the semantics of the image and the query phrase and the ability to predict the location of the object in the image. The visual grounding task is a natural extension of object detection [Ren et al., 2015]. While object detection aims to localize all possible pre-defined objects, visual grounding introduces a textual query phrase to localize only one best matching object from an open vocabulary.

Most existing visual grounding approaches [Rohrbach et al., 2016; Fukui et al., 2016; Hu et al., 2016] can be modularized into the general framework shown in Figure 1. Given an input image, a fixed number of object proposals are first generated using a proposal generation model and visual features for each proposal are then extracted. Meanwhile, the input query is encoded by a language model (e.g., LSTM [Hochreiter and Schmidhuber, 1997]) that outputs a textual feature vector. The textual and visual features are then integrated by a multi-modal fusion model and fed into a proposal ranking model to output the location of the proposal with the highest ranking score.

Of the three modules shown in Figure 1, most existing visual grounding approaches focus on designing robust multi-modal feature fusion models. SCRC [Hu et al., 2016] and phrase-based CCA [Plummer et al., 2015] learn the multi-
modal common space via canonical correlation analysis (C-CA) and a recurrent neural network (RNN), respectively. MCB learns a compact bilinear pooling model to fuse the multi-modal features, to obtain a more discriminative fused feature [Fukui et al., 2016]. Further, some approaches have explored designing different loss functions to help improve the accuracy of the final object localization. Rohrbach et al. use an auxiliary reconstruction loss function to regularize the model training [Rohrbach et al., 2016]. Wu et al. propose a model that gradually refine the predicted bounding box via reinforcement learning [Wu et al., 2017].

Compared with the aforementioned two modules, proposal generation has been less thoroughly investigated. Most visual grounding approaches usually use class-agnostic models (e.g., selective search [Uijlings et al., 2013] or data-driven region proposal networks (RPNs) [Ren et al., 2015] trained on specific datasets) to generate object proposals and extract visual features for each proposal [Fukui et al., 2016; Hu et al., 2016; Rohrbach et al., 2016; Chen et al., 2017; Li et al., 2017]. Although many visual grounding approaches have been proposed, what makes a good proposal generator for visual grounding remains uncertain.

Choosing the optimal number of generated proposals for visual grounding is difficult. If the number is small, recall of the true objects is limited, which seriously influencing visual grounding performance. If the number is large, recall is satisfactory but it does not necessarily improve visual grounding performance, as it increases the difficulty for accurate prediction of the proposal ranking model. Here, we rethink this problem and suggest that the ideal generated proposal set should contain as many different objects as possible but be of a relatively small size. To achieve this goal, we introduce diversity and discrimination simultaneously when generating proposals, and in doing so propose Diversified and Discriminative Proposal Networks (DDPN).

Based on the proposals generated by DDPN, we propose a high performance baseline model to verify the effectiveness of the DDPN for visual grounding. Our model uses simple feature concatenation as the multi-modal fusion model, and trained with two novel loss functions: Kullback-Leibler Divergence (KLD) loss with soft labels to penalize the proposals while capturing contextual information of the generated proposals, and smoothed $L_1$ regression loss to refine the proposal bounding boxes.

The main contributions of this study are as follows: 1) we analyze the limitations of existing proposal generators for visual grounding and propose DDPN to generate high-quality proposals; 2) based on DDPN, we propose a high performance baseline model trained with two novel losses; and 3) by way of extensive ablation studies, we evaluate our models on four benchmark datasets. Our experimental results demonstrate that our model delivers a significant improvement on all the tested datasets.

## 2 The Diversified and Discriminative Proposal Networks (DDPN)

For visual grounding, we propose that the ideal generated proposals should be diversified and discriminative simulta-

![Figure 2: Examples of diversified proposals and discriminative proposals.](image)
model (e.g., VGG-16 [Simonyan and Zisserman, 2014] or ResNet-101 [He et al., 2016]) pre-trained on ImageNet. We then train Faster R-CNN on the processed Visual Genome datasets. Slightly different from the standard Faster R-CNN model with four losses, we add an additional loss to the last fully-connected layer to predict attribute classes in addition to object classes. Specifically, we concatenate the fully-connected feature with a learned embedding of the ground-truth object class and feed this into an additional output layer defining a softmax distribution over the attribute classes.

Note that although DDPN is similar to [Anderson et al., 2017], we have a different motivation. In their approach, Faster R-CNN is used to build bottom-up attention to provide high-level image understanding. In this paper, we use Faster R-CNN to learn diversified and discriminative proposals for visual grounding.

3 Baseline Model for Visual Grounding

We first introduce a high performance baseline model for visual grounding based on DDPN. Given an image-query pair, our model is trained to predict the bounding boxes of the referred object in the image. The flowchart of our model is shown in Figure 3.

\[ q = \text{LSTM}(\text{Embed}(p)) \]  

(1)

For an input image, we first use the DDPN model to extract the top-\(N\) proposals with diversity and discrimination. Each proposal is represented as a visual feature (the last fully-connected layer after RoI pooling) \(v_{vis} \in \mathbb{R}^{d_v}\) and a 5-D spatial feature \(v_{spat} = [x_d/W, y_d/H, x_b/W, y_b/H, w/h/WH]\) proposed in [Yu et al., 2016]. We concatenate \(v_{vis}\) (with \(L_2\) normalization) and \(v_{spat}\) to obtain the visual feature \(v \in \mathbb{R}^{d_v+5}\) for each region proposal. For \(N\) region proposals, the visual features are defined as \(V = [v_1, ..., v_N]\).

For an input textual query \(p = [w_1, w_2, ..., w_t]\) describing the object in the image, \(t\) is the query length, and each word \(w_j\) is represented as a one-hot feature vector referring to the index of the word in a large vocabulary. As the length varies for different queries, we use the LSTM network to encode variable-length queries [Hochreiter and Schmidhuber, 1997]. Specifically, we first learn a word embedding layer to embed each word into a \(d_w\)-dimensional latent space. The sequence of the embedded word features is then fed to a one-layer LSTM network with \(d_q\) hidden units. We extract the feature of the last word from the LSTM network as the output feature \(q \in \mathbb{R}^{d_q}\) for the query phrase \(p\). The encoding procedure for the query phrases is represented as follows.

Once we have obtained the query feature \(q \in \mathbb{R}^{d_q}\) and visual features \(V = [v_1, ..., v_N] \in \mathbb{R}^{d_v \times N}\), we can fuse the multi-modal features with a feature fusion model and output the integrated features \(F = [f_1, ..., f_N]\). To better demonstrate the capacity of the visual features, we do not introduce a complex feature fusion model (e.g., bilinear pooling) and only use simple feature concatenation followed by a fully-connected layer for fusion. For each pair \([q, v]\), the output feature \(f\) is obtained as follows.

\[ f = \varphi(W_f^T (q||v) + b_f) \]  

(2)

where \(W_f \in \mathbb{R}^{(d_v+d_q) \times d_f}\) and \(b_f \in \mathbb{R}^{d_f}\) are the weights and bias respectively, \(||\) denotes concatenation of vectors, \(f \in \mathbb{R}^{d_f}\) is the fused feature. \(\varphi(\cdot)\) is the ReLU activation function.

3.1 Learning Objective

Given the fused feature \(f \in F\), we use an additional fully-connected layer to predict the score \(s \in \mathbb{R}\) for feature \(f\).

\[ s = W_s^T f + b_s \]  

(3)

where \(W_s \in \mathbb{R}^o\) and \(b_s \in \mathbb{R}\) are the weights and bias, respectively. For \(N\) proposals, we can obtain their scores \(S = [s_1, ..., s_N] \in \mathbb{R}^N\) using Eq.(3). We apply softmax normalization to \(S\) to make it satisfy a score distribution.

The problem now becomes how to define the loss function to make the predicted ranking scores \(S\) consistent with their ground-truth ranking scores \(S^*\). In most existing visual grounding approaches [Fukui et al., 2016; Hu et al., 2016; Chen et al., 2017], the ground-truth ranking scores \(S^* = [s_1^*, ..., s_N^*] \in \{0, 1\}^N\) are defined as a one-hot vector in which the only element is set to 1 when the corresponding proposal most overlaps with the ground-truth bounding box (i.e., the largest IoU score) and 0 otherwise. Based on the one-hot single label, softmax cross-entropy loss is used to learning the ranking model.

Slightly different to their strategy, we use the soft label distribution as the ground-truth. We calculate the IoU scores of all proposals w.r.t. the ground-truth bounding box and assign the IoU score of the \(i\)-th proposal to \(s_i^*\) if the IoU score is larger than a threshold \(\gamma\) and 0 otherwise. In this way, we obtain the real-value label vector \(S^* = [s_1^*, ..., s_N^*] \in \mathbb{R}^N\). To make \(S^*\) satisfy a probability distribution, \(S^*\) is \(L_1\)-normalized to guarantee \(\sum_i^N s_i^* = 1\). Accordingly, we use Kullback-Leibler Divergence (KLD) as our ranking function for model optimization and to make the predicted score distribution \(S\) and ground-truth label distribution \(S^*\) as close as possible.

\[ L_{\text{rank}} = \frac{1}{N} \sum_i^N s_i^* \log \left( \frac{s_i^*}{s_i} \right) \]  

(4)

The benefits of using soft labels are three-fold: 1) except for the max-overlapping proposal, other proposals may also contain useful information, that provides contextual knowledge of the ground-truth; 2) the soft label can be seen as a model regularization strategy, as introduced in [Szegedy et al., 2016] as label smoothing; and 3) during testing, the
The predicted bounding box is considered to be correct when its IoU score with the ground-truth is larger than a threshold \( \eta \). Therefore, optimizing the model with soft labels applies consistency to training and testing and may improve visual grounding performance.

Although using the KLD loss in Eq.(4) can capture the proposals’ contextual information, the accuracies of the region proposals themselves could be a performance bottleneck for visual grounding. In the case that all the generated region proposals do not overlap with the ground-truth bounding box, the grounding accuracy will be zero. Inspired by the strategy used in Faster R-CNN [Ren et al., 2015], we append an additional fully-connected layer on top of fused feature \( f \) and add a regression layer to refine the proposal coordinates.

\[
t = W^T_t f + b_t
\]

where \( t \in \mathbb{R}^4 \) corresponds to the coordinates of the refined bounding box, and \( W_t \in \mathbb{R}^{d_f \times 4} \) and \( b_t \in \mathbb{R}^4 \) are the weights and bias respectively. Accordingly, the smoothed \( L_1 \) regression loss is defined as follows to penalize the difference between \( t \) and its ground-truth bounding box coordinates \( t^* \) for all the proposals.

\[
L_{reg} = \frac{1}{N} \sum_{i} \text{smooth}_L_1(t^*_i, t_i) \tag{6}
\]

where \( \text{smooth}_L_1(x, y) \) is the smoothed \( L_1 \) loss function for input features \( x \) and \( y \) [Ren et al., 2015]. It is worth noting that although the proposals generated by DDPPN has been refined by a regression layer in the Faster R-CNN model, the regression layer here has a different meaning: it acts as a domain adaptation function to align the semantics of the source domain (i.e., the Visual Genome dataset) and the target domain (i.e., the specific visual grounding dataset).

The overall loss for our model is defined as:

\[
L = L_{rank} + \gamma L_{reg} \tag{7}
\]

where \( \gamma \) is a hyper-parameter to balance the two terms.

### 4 Experiments

We evaluate our approach on four benchmark datasets: Flickr30K Entities [Plummer et al., 2015], ReferItGame [Kazemzadeh et al., 2014], RefCOCO and RefCOCO+ [Yu et al., 2016]. These are all commonly used benchmark datasets for visual grounding.

#### 4.1 Implementation Details

We use the same hyper-parameters as in [Anderson et al., 2017] to train the DDPPN model. We train DDPPN with different CNN backbones (VGG-16 and ResNet-101). Both models are trained for up to 30 epochs, which takes two GPUs 2~3 weeks to finish.

Based on the optimized DDPPN model, we train our visual grounding model with the hyper-parameters listed as follows. The loss weight \( \gamma \) is set to 1 for all experiments. The dimensionality of the visual feature \( d_v \) is 2048 (for ResNet-101) or 4096 (for VGG-16), the dimensionality of the word embedding feature \( d_w \) is 300, the dimensionality of the output feature of the LSTM network \( d_o \) is 1024, the dimensionality of the fused feature \( d_f \) is 512, the threshold of IoU scores \( \eta \) is 0.5, and the number of proposals \( N \) is 100. During training, all the weight parameters, including the word embedding layer, the LSTM network and the multi-modal fusion model are initialized using the xavier method. We use the Adam solver to train the model with \( \beta_1 = 0.9, \beta_2 = 0.99 \). The base learning rate is set to 0.001 with an exponential decay rate of 0.1. The mini-batch size is set to 64. All the models are trained up to 10k iterations. During testing, we feed forward the network and output the scores for all the proposals before picking the proposal with the highest score and using its refined bounding box as the final output.

#### 4.2 Datasets

**Flickr30K Entities**

Flickr30K Entities contains 32k images, and 275k bounding boxes and 360k query phrases. Some phrases in the dataset may correspond to multiple boxes. In such cases, we merge the boxes and use their union as the ground-truth [Rohrbach et al., 2016]. We use the standard split in our setting, i.e., 1k images for validation, 1k for testing, and 30k for training.

**ReferItGame**

ReferItGame contains over 99k bounding boxes and 130k query phrases from 20k images. Each bounding box is associated with 1-3 query phrases. We use the same data split as in [Rohrbach et al., 2016], namely 10k images for testing, 9k for training and 1k for validation.

**RefCOCO and RefCOCO+**

Both the RefCOCO and RefCOCO+ datasets utilize images from the COCO dataset [Lin et al., 2014]. Both of them consist of 142k query phrases and 50k bounding boxes from 20k images. The difference between the two datasets is that the query phrases in RefCOCO+ does not contain any location word, which is more difficult to understand the query intent. The datasets are split into four sets: train, validation, testA and testB. The images in testA contain multiple people, while images in testB contain objects in other categories.

#### 4.3 Experimental Setup

**Evaluation Metric**

For all datasets, we adopt accuracy as the evaluation metric, which is defined as the percentage in which the predicted bounding box overlaps with the ground-truth of IoU > 0.5.

**Compared Methods**

We compare our approach with state-of-the-art visual grounding methods: SCRC [Hu et al., 2016], GroundeR [Rohrbach et al., 2016], MCB [Fukui et al., 2016], QRC [Chen et al., 2017], and the approaches of Li et al. [Li et al., 2017], Wu et al. [Wu et al., 2017] and Yu et al. [Yu et al., 2017a].

#### 4.4 Ablation Studies

We also perform the following ablation experiments on Flickr30k Entities to verify the efficacy of DDPPN, and the loss functions to train our baseline visual grounding model.
Effects of Diversity and Discrimination in Proposal Generation

We define the discrimination score ($S_{\text{DIS}}$) and the diversity score ($S_{\text{DIV}}$) in Eq. (8) to evaluate the proposal generators.

$$S_{\text{DIS}} = \frac{1}{M^2} \sum_{i, j} \max_{t^*_j} \sigma(t^*_j, t^*_i)$$

$$S_{\text{DIV}} = \frac{1}{M^2} \sum_{i, j} N \sum_{j} \sigma(t^*_j, t^*_i)$$

where $M$ is the number of query-image samples in the test set, $N$ is the number of proposals per sample, and $t^*_j$ indicates the $j$-th proposal in the $i$-th sample. For the $i$-th sample, the indicator function $\sigma(t^*_j, t^*_i) = 1$ if a proposal $t^*_j$ covers its corresponding ground-truth bounding box $t^*_i$ with IoU > 0.5 and 0 otherwise. $S_{\text{DIS}}$ is the discrimination score which is defined as the percentage of samples with at least one proposal covering its ground-truth object. A larger discrimination score leads to more accurate proposals. $S_{\text{DIV}}$ is the diversity score, which is defined as the reciprocal of the average number of covered proposals per ground-truth object. A larger diversity score leads to fewer redundant proposals and a more balanced proposal distribution. A good proposal generator should have high $S_{\text{DIV}}$ and $S_{\text{DIS}}$ simultaneously.

In Table 1, we report the $S_{\text{DIV}}$ and $S_{\text{DIS}}$, as well as the visual grounding accuracies of different proposal generators. The results show that: 1) SS and RPN (w/o ft) models have similar discriminative scores, while the diversity score of RPN (w/o ft) is smaller than that of SS. This reflects the fact that SS is more diversified compared to the data-driven RPN model pre-trained on PASCAL VOC 07 [Everingham et al., 2010]; 2) after fine-tuning on Flickr30k Entities, the RPN (w/ ft) model obtains a better discriminative score than RPN (w/o ft). However, the diversity score is reduced, indicating that RPN (w/ ft) fits Flickr30k Entities at the expense of model generalizability; 3) DDPN significantly outperforms other approaches for both discriminative and diversity scores. To further demonstrate the effect of DDPN, we illustrate proposals for an image using the aforementioned four models in Figure 4, and the visualized results show that DDPN generates proposals of higher quality than other approaches; 4) DDPN significantly improve the visual grounding performance compared to other proposal generation models, which proves our hypothesis that both diversity and discrimination are of great significance for visual grounding performance.

Effects of Different Losses on Visual Grounding

In Table 2, we report the performance of our model variants trained with different losses. It can be seen that: 1) training with the KLD loss leads to a 1.2~1.7-point improvement over the models with classical single-label softmax loss. This verifies our hypothesis that using soft labels captures contextual information and enhances the model’s capacity. Compared to the context modeling strategy proposed by [Chen et al., 2017], which exploits the ground-truth bounding boxes within an image from multiple queries, our strategy is more general and easier to implement; 2) training with an additional regression loss to refine the proposals leads to a 4.7~5.2-point improvement. We believe this will become a commonly-used strategy in future visual grounding studies.

4.5 Comparisons with State-of-the-Art

We next compare our models to current state-of-the-art approaches. We report the results of our approach with DDPN of two backbones, namely VGG-16 [Simonyan and Zisserman, 2014] and ResNet-101 [He et al., 2016].

Tables 3, 4 and 5 show the comparative results on RefCOCO, RefCOCO+, Flickr30k Entities and ReferItGame,
Table 3: The accuracies of different methods on RefCOCO and RefCOCO+ datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>RefCOCO TestA (%)</th>
<th>RefCOCO TestB (%)</th>
<th>Validation (%)</th>
<th>RefCOCO+ TestA (%)</th>
<th>RefCOCO+ TestB (%)</th>
<th>Validation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCRC [Hu et al., 2016]</td>
<td>18.5</td>
<td>20.2</td>
<td>19.0</td>
<td>14.4</td>
<td>13.3</td>
<td>13.7</td>
</tr>
<tr>
<td>Wu et al. [Wu et al., 2017]</td>
<td>54.8</td>
<td>41.6</td>
<td>48.2</td>
<td>40.4</td>
<td>22.8</td>
<td>31.9</td>
</tr>
<tr>
<td>Li et al. [Li et al., 2017]</td>
<td>55.2</td>
<td>46.2</td>
<td>52.4</td>
<td>45.2</td>
<td>32.2</td>
<td>40.2</td>
</tr>
<tr>
<td>Yu et al. [Yu et al., 2017a]</td>
<td>73.7</td>
<td>65.0</td>
<td>69.5</td>
<td>60.7</td>
<td>48.8</td>
<td>55.7</td>
</tr>
<tr>
<td>Ours (VGG-16)</td>
<td>76.9</td>
<td>67.5</td>
<td>73.4</td>
<td>67.0</td>
<td>50.2</td>
<td>60.1</td>
</tr>
<tr>
<td>Ours (ResNet-101)</td>
<td><strong>80.1</strong></td>
<td><strong>72.4</strong></td>
<td><strong>76.8</strong></td>
<td><strong>70.5</strong></td>
<td><strong>54.1</strong></td>
<td><strong>64.8</strong></td>
</tr>
</tbody>
</table>

Figure 5: The examples in Flickr30k Entities (the 1st row) and ReferItGame (the 2nd row) datasets. The ground-truth (green), the top-ranked predicted proposal (yellow) and the refined final prediction (red) are visualized respectively. The last three columns shows the examples with incorrect predictions (IoU \(\leq 0.5\)). Best viewed in color.

Table 4: The accuracies of different methods on Flickr30k Entities

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCRC [Hu et al., 2016]</td>
<td>27.8</td>
</tr>
<tr>
<td>GroundeR [Rohrbach et al., 2016]</td>
<td>47.8</td>
</tr>
<tr>
<td>MCB [Fukui et al., 2016]</td>
<td>48.7</td>
</tr>
<tr>
<td>QRC [Chen et al., 2017]</td>
<td>65.1</td>
</tr>
<tr>
<td>Ours (VGG-16)</td>
<td><strong>70.0</strong></td>
</tr>
<tr>
<td>Ours (ResNet-101)</td>
<td><strong>73.3</strong></td>
</tr>
</tbody>
</table>

Table 5: The accuracies of different methods on ReferItGame

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCRC [Hu et al., 2016]</td>
<td>17.9</td>
</tr>
<tr>
<td>GroundeR [Rohrbach et al., 2016]</td>
<td>28.5</td>
</tr>
<tr>
<td>MCB [Fukui et al., 2016]</td>
<td>28.9</td>
</tr>
<tr>
<td>Wu et al. [Wu et al., 2017]</td>
<td>36.8</td>
</tr>
<tr>
<td>QRC [Chen et al., 2017]</td>
<td>44.1</td>
</tr>
<tr>
<td>Li et al. [Li et al., 2017]</td>
<td>44.2</td>
</tr>
<tr>
<td>Ours (VGG-16)</td>
<td><strong>60.3</strong></td>
</tr>
<tr>
<td>Ours (ResNet-101)</td>
<td><strong>63.0</strong></td>
</tr>
</tbody>
</table>

5 Conclusions and Future Work

In this paper, we interrogate the proposal generation for visual grounding and in doing so propose Diversified and Discriminative Proposal Networks (DDPN) to produce high-quality proposals. Based on the proposals and visual features extracted from the DDPN model, we design a high performance baseline for visual grounding trained with two novel losses: KLD loss to capture the contextual information of the proposals and regression loss to refine the proposals. We conduct extensive experiments on four benchmark datasets and achieve significantly better results on all datasets.

Since the models studied here represent the baseline, there remains significant room for improvement, for example by introducing a more advanced backbone model for DDPN or introducing a more powerful multi-modal feature fusion model such as bilinear pooling.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant 61702143, Grant 61622205, Grant 61472110 and Grant 61602405, and in part

respectively. We note the following: 1) with the same CNN backbone (i.e., VGG-16), our model achieves an absolute improvement of 3.2 points on RefCOCO (testA), 6.3 points on RefCOCO+ (testA), 4.9 points on Flickr30k Entities and 16.1 points on ReferItGame, respectively. The improvement is primarily due to the high-quality proposals generated by DDPN, and the loss functions we use for our visual grounding model; 2) by replacing VGG-16 with ResNet-101 as the backbone for DDPN, all results improve by 3~4 points, illustrating that the representation capacity of DDPN significantly influences the visual grounding performance.

4.6 Qualitative Results

We visualize some visual grounding results of our model with ResNet-101 backbone model on Flickr30k Entities (the first row) and ReferItGame (the second row) in Figure 5. It can be seen that our approach achieves good visual grounding performance, and is able to handle the small and fine-grained objects. Moreover, the bounding box regression helps refine the results, rectifying some inaccurate proposals. Finally, our approach still has some limitations, especially when faced with complex queries or confused visual objects. These observations are useful to guide further improvements for visual grounding in the future.
by the Zhejiang Provincial Natural Science Foundation of China under Grant LR15F020002, in part by the Australian Research Council Projects under Grant FL-170100117 and Grant DP-180103424.

References


