

# Neighbourhood Watch: Referring Expression Comprehension via Language-guided Graph Attention Networks\*

Peng Wang<sup>1</sup> Qi Wu<sup>1</sup> Jiewei Cao<sup>1</sup> Chunhua Shen<sup>1</sup> Lianli Gao<sup>2</sup> Anton van den Hengel<sup>1</sup>

<sup>1</sup>Australian Institute for Machine Learning, The University of Adelaide

<sup>2</sup>The University of Electronic Science and Technology of China

## Abstract

The task in referring expression comprehension is to localise the object instance in an image described by a referring expression phrased in natural language. As a language-to-vision matching task, the key to this problem is to learn a discriminative object feature that can adapt to the expression used. To avoid ambiguity, the expression normally tends to describe not only the properties of the referent itself, but also its relationships to its neighbourhood. To capture and exploit this important information we propose a graph-based, language-guided attention mechanism. Being composed of node attention component and edge attention component, the proposed graph attention mechanism explicitly represents inter-object relationships, and properties with a flexibility and power impossible with competing approaches. Furthermore, the proposed graph attention mechanism enables the comprehension decision to be visualisable and explainable. Experiments on three referring expression comprehension datasets show the advantage of the proposed approach.

## 1. Introduction

A referring expression is a natural language phrase that refers to a particular object visible in an image. Referring expression comprehension thus requires to identify the unique object of interest, referred to by the language expression [29]. The critical challenge is thus the joint understanding of the textual and visual domains.

Referring expression comprehension can be formulated as a language-to-region matching problem, where the region with highest matching score is selected as the prediction. Learning a discriminative region representation that can adapt to the language expression is thus critical. The predominant approaches [7, 15, 16] tend to represent the region by stacking various types of features, such as

\*Q. Wu's participation was supported in part by the National Science of China (No. 61876152).

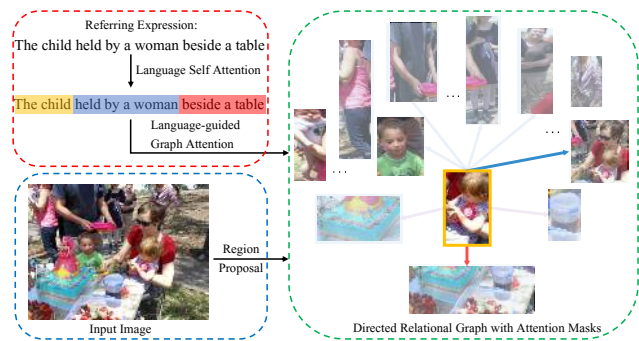


Figure 1. A directed graph is built over the object instances of the image, where nodes correspond to object regions and edges (partially visualised) represent relationships between objects (blue and red edges denote intra- and inter-class relationships respectively). Graph attention predicts the attention distribution over the nodes as well as the edges, based on the decomposed information present in the expression. Summarising the attended object and its highlighted neighbours enables more discriminative feature. Higher transparency here denotes a lower attention value.

CNN features, spatial features or heuristic contextual features, and employ a LSTM to process the expression simply as a series of words. However, these approaches are limited by the monolithic vector representations that ignore the complex structures in the compound language expression as well as in the image. Another potential problem for these approaches, and for more advanced modular schemes [5, 27], is that the language and region features are learned or designed independently without being informed by each other, which makes the features of the two modalities difficult to adapt to each other, especially when the expression is complex. Co-attention mechanisms are employed in [3, 32] to extract more informative features from both the language and the image to achieve better matching performance. These approaches, however, treat the objects in the image in isolation and thus fail to model the relationships between them. These relationships are naturally important in identifying the referent, especially when the expression is compound. For example, in Fig. 1, the expression “the child held by a woman beside a table” describes

not only the child but her relationships with another person and the table. In cases like this, focusing on the properties of the object only is not enough to localise the correct referent but we need to watch the neighbourhood to identify more useful clues.

To address the aforementioned problems, we propose to build a **directed graph** over the object regions of an image to **model the relationships between objects**. In this graph the nodes correspond to the objects and the edges represent the relationships between objects. On top of the graph, we propose a **language-guided graph attention network (LGRAN)** to highlight the relevant content referred to by the expression. The graph attention is composed of two main components: a *node attention* component to highlight relevant objects and an *edge attention* component to identify the object relationships present in the expression. Furthermore, the edge attention is divided into intra-class edge attention and inter-class edge attention to distinguish relationships between objects of the same category and those crossing categories. Normally, these two types of relationships are different visually and semantically. The three types of attention are guided by three corresponding language parts which are identified within the expression through a self-attention mechanism [5, 27]. By summarising the attended sub-graph centred on a potential object of interest, we can dynamically enrich the representation of this object in order that it can better adapt to the expression, as illustrated in Fig. 1.

Another benefit of the proposed graph attention mechanism is that it renders the referring expression decision both **visualisable** and **explainable**, because it is capable of grounding the referent and other supporting clues (*i.e.* its relationships with other objects) onto the graph. We conduct experiments on three referring expression datasets (RefCOCO, RefCOCO+ and RefCOCOg). The experimental results show the advantage of the proposed language-guided graph attention network. We outperform the previous best results on almost all splits, under different settings.

## 2. Related Work

**Referring Expression Comprehension** Conventional referring expression comprehension is approached using a CNN/LSTM framework [7, 15, 16, 28]. The LSTM takes as input a region-level CNN feature and a word vector at each time step, and aims to maximize the likelihood of the expression given the referred region. These models incorporate contextual information visually, and how they achieve this is one of the major differentiators of the various approaches. For example, the work in [7] uses a whole-image CNN feature as the region context, the work in [16] learns context regions via multiple-instance learning, and in [28], the authors use visual differences between objects to represent the visual context.

Another line of work treats referring expression comprehension as a metric learning problem [14, 15, 18, 25], whereby the expression feature and the region feature are embedded into a common feature space to measure the compatibility. The focus of these approaches lies in how to define the matching loss function, such as softmax loss [14, 18], max-margin loss [25], or Maximum Mutual Information (MMI) loss [15]. These approaches tend to use a single feature vector to represent the expression and the image region. These monolithic features ignore the complex structures in the language as well in the image, however. To overcome this limitation of monolithic features, self-attention mechanisms have been used to decompose the expression into sub-components and learn separate features for each of the resulting parts [6, 27, 30]. Another potential problem for the aforementioned methods is that the language and region features are learned independently without being informed by each other. To learn expression features and region features that can better adapt to each other, co-attention mechanisms have been used [3, 32]. These methods process the objects in isolation, however, and thus fail to model the object dependencies, which are critical in identifying the referent. In our model, we build a directed graph over the object regions of an image to model the relationships between objects. On top of that, a language-guided graph attention mechanism is proposed to highlight the relevant content referred to by the expression.

**Graph Attention** In [24] graph attention is applied to other graph-structured data, including document citation networks, and protein-protein interactions. The differences between their graph attention scheme and ours are three-fold. First, their graph edges reflect the connections between nodes only, while ours additionally encode the relationships between objects (that have properties of their own). Second, their attention is obtained via self-attention or the interaction between nodes, but our attention is guided by the referring statement. Third, they update the node information as a weighted sum of the neighbouring representations, but we maintain different types of features to represent the node properties and node relationships. In terms of building a graph to capture the structure in the structural data, our work is also related to graph neural networks [4, 8, 11, 12, 22]. Our focus in this paper is on identifying the expression-relevant information for an object for better language-to-region matching.

## 3. Language-guided Graph Attention Networks (LGRANs)

Here we elaborate on the proposed language-guided graph attention networks (LGRANs) for referring expression comprehension. Given the expression  $r$  and an image  $I$ , the aim of referring expression comprehension is to localise the object  $o^*$  referred to by  $r$  from the object

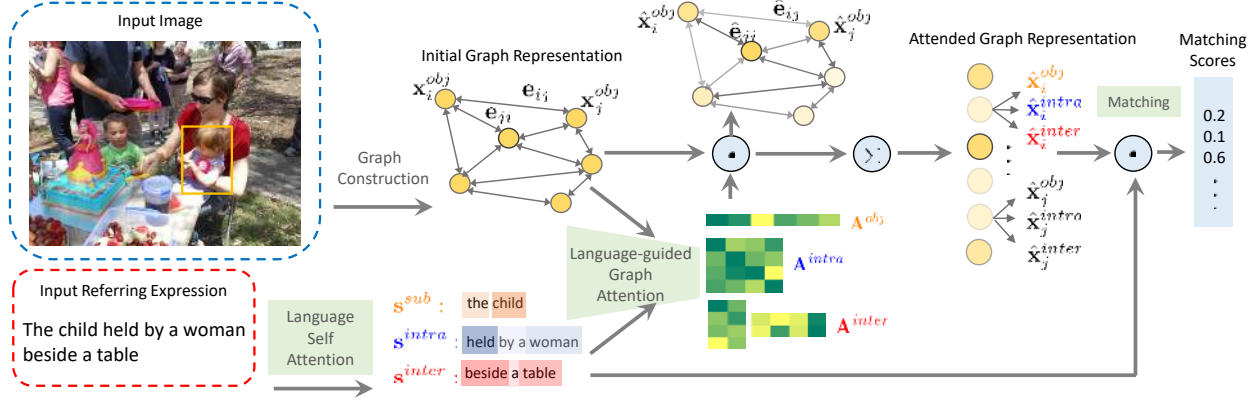


Figure 2. Overview of the proposed language-guided graph attention networks for referring expression comprehension. The network is composed of three modules: language-self attention module, language-guided graph attention module, and matching module.

set  $\mathcal{O} = \{o_i\}_{i=1}^N$  of  $I$ . The candidate object set is given as ground truth or obtained by an object proposal generation method, such as region proposal network [17], depending on the experimental setting. We evaluate both cases in Sec. 4.

As illustrated in Fig. 2, LGRANs is composed of three modules: (1) the language self-attention module, which adopts a self-attention scheme to decompose the expression  $r$  into three parts that describe the *subject*, *intra-class relationships* and *inter-class relationships*, and learn the corresponding representations  $s^{sub}$ ,  $s^{intra}$  and  $s^{inter}$ ; (2) the language-guided graph attention module, which builds a directed graph over the candidate objects  $\mathcal{O}$ , highlights the nodes (objects), intra-class edges (relationships between objects of the same category) and inter-class edges (relationships between objects from different categories) that are relevant to  $r$  under the guidance of  $s^{sub}$ ,  $s^{intra}$  and  $s^{inter}$ , and finally obtains three types of expression-relevant representations for each object; (3) the matching module, which computes the expression-to-object matching score. We now describe these modules in detail.

### 3.1. Language Self-Attention Module

Languages are compound and monolithic vector representations (such as the output of a LSTM at the final state) ignore the rich structure in the language. Inspired by the idea of decomposing compound language into substructures in various vision-to-language tasks [2, 5, 6, 27], we decompose the expression into sub-components as well. To fulfill their purpose referring expressions tend to describe not only the properties of the referent, but also its relationships with nearby objects. We thus decompose the expression  $r$  into three parts: subject  $r^{sub}$ , intra-class relationship  $r^{intra}$ , and inter-class relationship  $r^{inter}$ .

There are mainly two language parsing approaches: off-the-shelf language parsers [2] or self-attention [5, 6, 27].

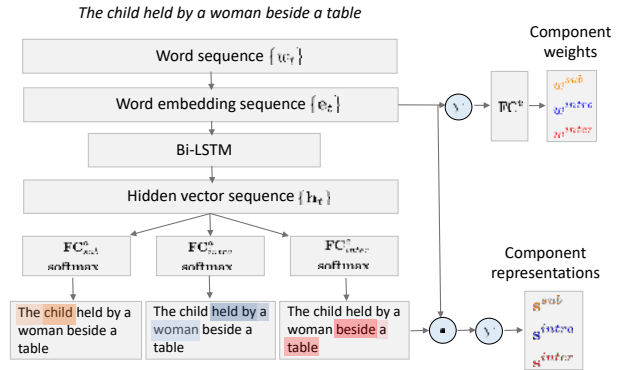


Figure 3. Illustration of the language self-attention module.

In this paper, we apply the self-attention scheme due to its better performance. Fig. 3 shows the high-level idea of our language attention mechanism. Given an expression  $r$  with  $T$  words  $r = \{w_t\}_{t=1}^T$ , we first embed the words' one-hot representations into a continuous space  $\{e_t\}_{t=1}^T$  using a non-linear mapping function  $f_e$ . Then  $\{e_t\}$  are fed into a Bi-LSTM [20] to obtain a set of hidden state representations  $\{h_t\}_{t=1}^T$ . Next, three individual fully-connected layers followed by softmax layers are applied to  $\{h_t\}$  to obtain three types of attention values, being subject attention  $\{a_t^{sub}\}_{t=1}^T$ , intra-class relationship attention  $\{a_t^{intra}\}_{t=1}^T$  and inter-class relationship attention  $\{a_t^{inter}\}_{t=1}^T$ . As the attention values are obtained by the same way for all three components, for simplicity we only show the details for the calculation of the subject component  $r^{sub}$ . Let

$$a_t^{sub} = \frac{\exp(\mathbf{w}_{sub_a}^T \mathbf{h}_t)}{\sum_{i=1}^T \exp(\mathbf{w}_{sub_a}^T \mathbf{h}_i)}, \quad (1)$$

where  $\mathbf{w}_{sub_a}$  denotes  $\mathbf{FC}_{sub}^a$  in Fig. 3. Then, the attention values are applied to the embedding vectors  $\{e_t\}$  to derive three representations:  $s^{sub}$ ,  $s^{intra}$  and  $s^{inter}$ . Here

we choose  $\mathbf{s}^{sub}$  for illustration:

$$\mathbf{s}^{sub} = \sum_{t=1}^T \alpha_t^{sub} \cdot \mathbf{e}_t. \quad (2)$$

Inspired by [27], we apply another linear mapping  $\text{FC}^w$  to the pooled embedding vector,  $\mathbf{e} = \sum_{t=1}^T \mathbf{e}_t$ , to derive three weights  $[w^{sub}, w^{intra}, w^{inter}]$ . These serve as the weights for  $[r^{sub}, r^{intra}, r^{inter}]$  in expression-to-region matching, that will be introduced in Sec. 3.3. Again we present how to obtain  $w^{sub}$  only,

$$w^{sub} = \frac{\exp(\mathbf{w}_{sub_w}^\top \mathbf{e})}{\exp(\mathbf{w}_{sub_w}^\top \mathbf{e}) + \exp(\mathbf{w}_{intra_w}^\top \mathbf{e}) + \exp(\mathbf{w}_{inter_w}^\top \mathbf{e})}, \quad (3)$$

where  $\mathbf{w}_{sub_w}$ ,  $\mathbf{w}_{intra_w}$ ,  $\mathbf{w}_{inter_w}$  denote linear mappings.

## 3.2. Language-guided Graph Attention Module

The language-guided graph attention module is the key of the network. It builds a graph over the objects of an image to model object dependencies and identifies the nodes and edges relevant to the expression to dynamically learn object representations that adapt to the language expression.

### 3.2.1 Graph construction

Given the object or region set  $\mathcal{O} = \{o_i\}_{i=1}^N$  of an image  $I$ , we build a directed graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  over  $\mathcal{O}$ , where  $\mathcal{V} = \{v_i\}_{i=1}^N$  is the node set and  $\mathcal{E} = \{e_{ij}\}$  is the edge set. Each node  $v_i$  corresponds to an object  $o_i \in \{\mathcal{O}\}$  and an edge  $e_{ij}$  denotes the relationship between  $o_i$  and  $o_j$ . Based on whether the two nodes connected by an edge belong to the same category or not, we divide the edges into two sets: intra-class edges  $\mathcal{E}^{intra}$  and inter-class edges  $\mathcal{E}^{inter}$ . That is,  $\mathcal{E} = \mathcal{E}^{intra} \cup \mathcal{E}^{inter}$  and  $\mathcal{E}^{intra} \cap \mathcal{E}^{inter} = \emptyset$ . Assume  $c(v_i)$  denotes the category of  $v_i$ , the two types of edges can be represented as,  $\mathcal{E}^{intra} = \{e_{ij} : c(v_i) = c(v_j)\}$  and  $\mathcal{E}^{inter} = \{e_{ij} : c(v_i) \neq c(v_j)\}$ .

Considering that an object typically only interacts with objects nearby, we define edges between an object and its neighbourhood. Specifically, given a node  $v_i$ , we rank the remaining objects of the same category,  $\{v_j : c(v_j) = c(v_i)\}$ , based on their distances to  $v_i$  and define the intra-class neighbourhood  $\mathcal{N}_i^{intra}$  of  $v_i$  as the top  $k$  ranked intra-class objects. Similarly, we define the inter-class neighbourhood  $\mathcal{N}_i^{inter}$  of  $v_i$  to be the top  $k$  ranked objects that belong to other categories. For a node  $v_i$ , we define an edge between  $v_i$  and  $v_j$  if and only if  $v_j \in \mathcal{N}_i^{intra}$  or  $v_j \in \mathcal{N}_i^{inter}$ . A bigger  $k$  leads to a denser graph, and to balance the efficiency and representation capacity, we set  $k = 5$ .

We extract two types of node features for each node  $v_i$ : appearance feature  $\mathbf{v}_i$  and spatial feature  $\mathbf{l}_i$ . To obtain the appearance feature, we first resize the corresponding region  $o_i$  to  $224 \times 224$  and feed it to VGG16 net

[21]. The *Conv5\_3* features  $\mathbf{V} \in \mathbb{R}^{7 \times 7 \times 512}$  are pooled over the *height* and *width* dimensions to obtain the representation  $\mathbf{v}_i \in \mathbb{R}^{512}$ . The spatial feature  $\mathbf{l}_i$  is obtained as in [29], which is a 5-dimensional vector, encoding the top-left, bottom-right coordinates and the size of the bounding box with respect to the whole image, *i.e.*,  $\mathbf{l}_i = [\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$ . The node representation is a concatenation of the appearance feature and spatial feature, *i.e.*  $\mathbf{x}_i^{obj} = [\mathbf{v}_i, \mathbf{l}_i]$ . It has been shown that the relative spatial feature between two objects is a strong representation to encode their relationship [31]. Similarly, we model the edge between  $v_i$  and  $v_j$  based on their relative spatial information. Suppose the centre coordinate, width and height of  $v_i$  are represented as  $[x_{c_i}, y_{c_i}, w_i, h_i]$ , and the top-left coordinate, bottom-right coordinate, width and height of  $v_j$  are represented as  $[x_{tl_j}, y_{tl_j}, x_{br_j}, y_{br_j}, w_j, h_j]$ , then the edge representation is represented as,  $\mathbf{e}_{ij} = [\frac{x_{tl_j} - x_{c_i}}{w_i}, \frac{y_{tl_j} - y_{c_i}}{h_i}, \frac{x_{br_j} - x_{c_i}}{w_i}, \frac{y_{br_j} - y_{c_i}}{h_i}, \frac{w_j \cdot h_j}{w_i \cdot h_i}]$ .

### 3.2.2 Language-guided Graph attention

The aim of graph attention is to highlight the nodes and edges that are relevant to the expression  $r$  and consequently obtain object features that adapt to  $r$ . The graph attention is composed of two parts: *node attention* and *edge attention*. Furthermore, the edge attention can be divided into *intra-class edge attention* and *inter-class edge attention*. Mathematically, this process can be expressed as,

$$\{\mathbf{A}^{obj}, \mathbf{A}^{intra}, \mathbf{A}^{inter}\} = f(\{\mathbf{x}_i^{obj}\}, \{\mathbf{e}_{i,j}\}, \mathbf{s}^{sub}, \mathbf{s}^{intra}, \mathbf{s}^{inter}), \quad (4)$$

where  $\mathbf{A}^{obj}$ ,  $\mathbf{A}^{intra}$ , and  $\mathbf{A}^{inter}$  denote node attention values, intra-class edge attention values, and inter-class edge attention values respectively. The series of  $\mathbf{s}$  are the attended features from the language part. The function  $f$  is a graph attention mechanism that is guided by the language, which will be introduced as following three parts.

**The node attention** The node attention mechanism is inspired by the bottom-up attention [1], which enables attention to be calculated at the level of objects and other salient image regions [23, 33]. Given the node features  $\{\mathbf{x}_i^{obj}\}_{i=1}^N$ , where  $\mathbf{x}_i^{obj} = [\mathbf{v}_i, \mathbf{l}_i]$ , and the subject feature  $\mathbf{s}^{sub}$  of  $r$  in Sec. 3.1, the node attention is computed as,

$$\begin{aligned} \mathbf{v}_i^e &= f_{emb}^v(\mathbf{v}_i) \\ \mathbf{l}_i^e &= f_{emb}^l(\mathbf{l}_i) \\ \mathbf{x}_i^{e,obj} &= [\mathbf{v}_i^e, \mathbf{l}_i^e] \\ \mathbf{x}_i^{a,obj} &= \tanh(\mathbf{W}_{s,sub}^a \mathbf{s}^{sub} + \mathbf{W}_{g,obj}^a \mathbf{x}_i^{e,obj}) \\ A_i^{obj'} &= \mathbf{w}_{a,obj}^\top \mathbf{x}_i^{a,obj} \\ A_i^{obj} &= \frac{\exp(A_i^{obj'})}{\sum_j^N \exp(A_j^{obj'})}, \end{aligned} \quad (5)$$

where  $f_{emb}^v$  and  $f_{emb}^l$  are MLPs used to encode appearance and local features of  $v_i$  separately,  $\mathbf{W}_{g,obj}^a$  and  $\mathbf{W}_{s,sub}^a$  map the encoded node feature  $\mathbf{x}_i^{e,obj}$  and subject feature  $\mathbf{s}^{sub}$  of  $r$  into vectors of the same dimensionality,  $\mathbf{w}_{a,obj}$  calculates the attention values  $\{A_i^{obj'}\}$  for  $\{v_i\}$ , and all these attention values  $\{A_i^{obj'}\}_{i=1}^N$  are fed into a softmax layer to obtain the final attention values,  $\mathbf{A}^{obj} = \{A_i^{obj}\}$ .

**The intra-class edge attention** We obtain the attention values for intra-class edges  $\mathcal{E}^{intra}$  and inter-class edges  $\mathcal{E}^{inter}$  in similar ways. Given an intra-class edge  $e_{i,j} \in \mathcal{E}^{intra}$  and the intra-class relationship feature  $\mathbf{s}^{intra}$  of the expression  $r$ , the attention value for  $e_{i,j}$  is calculated as,

$$\begin{aligned} \mathbf{e}_{ij}^{intra} &= f_{emb}^{intra}(\mathbf{e}_{ij}) \\ \mathbf{e}_{ij}^{a,intra} &= \tanh(\mathbf{W}_{s,intra}^a \mathbf{s}^{intra} + \mathbf{W}_{g,intra}^a \mathbf{e}_{ij}^{intra}) \\ A_{ij}^{intra'} &= \mathbf{w}_{a,intra}^\top \mathbf{e}_{ij}^{a,intra} \\ A_{ij}^{intra} &= \frac{\exp(A_{ij}^{intra'})}{\sum_{k \in \mathcal{N}_i^{intra}} \exp(A_{ik}^{intra'})}, \end{aligned} \quad (6)$$

where  $f_{emb}^{intra}$  is a MLP to encode the edge feature,  $\mathbf{W}_{g,intra}^a$  and  $\mathbf{W}_{s,intra}^a$  map the encoded edge feature and intra-class relationship feature  $\mathbf{s}^{intra}$  of expression  $r$  into vectors of the same dimensionality,  $\mathbf{w}_{a,intra}$  calculates the intra-class attention values for  $e_{ij}$ , and these attention values are normalised among the intra-class neighbourhood  $\mathcal{N}_i^{intra}$  of  $v_i$  via a softmax.

**The inter-class edge attention** The attention value for inter-class edge  $e_{ij} \in \mathcal{E}^{inter}$  is calculated under the guidance of the inter-class relationship feature  $\mathbf{s}^{inter}$  of expression  $r$ ,

$$\begin{aligned} \mathbf{e}_{ij}^{inter} &= f_{emb}^{inter}([\mathbf{e}_{ij}, \mathbf{x}_j^{obj}]) \\ \mathbf{e}_{ij}^{a,inter} &= \tanh(\mathbf{W}_{s,inter}^a \mathbf{s}^{inter} + \mathbf{W}_{g,inter}^a \mathbf{e}_{ij}^{inter}) \\ A_{ij}^{inter'} &= \mathbf{w}_{a,inter}^\top \mathbf{e}_{ij}^{a,inter} \\ A_{ij}^{inter} &= \frac{\exp(A_{ij}^{inter'})}{\sum_{k \in \mathcal{N}_i^{inter}} \exp(A_{ik}^{inter'})}, \end{aligned} \quad (7)$$

where  $f_{emb}^{inter}$  is a MLP. Comparing Eq. 6 and Eq. 7, the features used to represent the intra-class relationship and inter-class relationship are different. When the subject  $v_i$  and object  $v_j$  are from the same category, we only use their relative spatial feature  $\mathbf{e}_{ij}$  to represent the relationship between them. However, when  $v_i$  and  $v_j$  are from different classes (e.g. man riding horse) we need to explicitly model the object  $v_j$  and thus we design the relationship representation to be the concatenation of the edge feature  $\mathbf{e}_{ij}$  and the node feature  $\mathbf{x}_j^{obj}$ .

### 3.2.3 The Attended Graph Representation

With the node and edge attention determined under the guidance of the expression  $r$ , the next step is to obtain the final representation for the object by aggregating the attended content. Corresponding to the decomposition of the expression, we obtain three types of features for each node: object features, intra-class relationship features, and inter-class relationship features.

The node representation for  $v_i$  will be updated to  $\hat{\mathbf{x}}_i^{obj}$ ,

$$\hat{\mathbf{x}}_i^{obj} = A_i^{obj} \mathbf{x}_i^{e,obj}, \quad (8)$$

where  $A_i^{obj}$  denotes the node attention value for  $v_i$  and  $\mathbf{x}_i^{e,obj}$  is the encoded node feature in Eq. 5.

The intra-class relationship representation  $\hat{\mathbf{x}}_i^{intra}$  will be the weighted sum of the intra-class edge representations,

$$\hat{\mathbf{x}}_i^{intra} = \sum_{j \in \mathcal{N}_i^{intra}} A_{ij}^{intra} \mathbf{e}_{ij}^{intra}, \quad (9)$$

where  $\mathcal{N}_i^{intra}$  denotes the intra-class neighbourhood of  $v_i$ ,  $A_{ij}^{intra}$  denotes the intra-class edge attention value and  $\mathbf{e}_{ij}^{intra}$  is the encoded intra-class edge feature in Eq. 6.

The inter-class relationship representation  $\hat{\mathbf{x}}_i^{inter}$  is obtained as the weighted sum of the inter-class edge representations,

$$\hat{\mathbf{x}}_i^{inter} = \sum_{j \in \mathcal{N}_i^{inter}} A_{ij}^{inter} \mathbf{e}_{ij}^{inter}, \quad (10)$$

where  $\mathcal{N}_i^{inter}$  denotes the inter-class neighbourhood of  $v_i$ ,  $A_{ij}^{inter}$  denotes the inter-class edge attention value and  $\mathbf{e}_{ij}^{inter}$  is the encoded inter-class edge feature in Eq. 7.

### 3.3. Matching Module and Loss Function

The matching score between the expression  $r$  and an object  $v_i$  is calculated as the weighted sum of three parts: subject, intra-class relationship, and inter-class relationship,

$$\begin{aligned} p_i^{obj} &= \tanh(\mathbf{W}_{s,subj}^m \mathbf{s}^{obj})^\top \tanh(\mathbf{W}_{g,obj}^m \hat{\mathbf{x}}_i^{obj}) \\ p_i^{intra} &= \tanh(\mathbf{W}_{s,intra}^m \mathbf{s}^{intra})^\top \tanh(\mathbf{W}_{g,intra}^m \hat{\mathbf{x}}_i^{intra}) \\ p_i^{inter} &= \tanh(\mathbf{W}_{s,inter}^m \mathbf{s}^{inter})^\top \tanh(\mathbf{W}_{g,inter}^m \hat{\mathbf{x}}_i^{inter}) \\ p_i &= w^{subj} p_i^{obj} + w^{intra} p_i^{intra} + w^{inter} p_i^{inter}, \end{aligned} \quad (11)$$

where each expression component feature and object component feature are encoded by a MLP (linear mapping + non-linear function  $\tanh(\cdot)$ ) respectively before a dot product. The weights of the three parts are obtained from  $r$  as introduced in Sec. 3.1. The probability for  $v_i$  being the referent is  $prob_i = \text{softmax}(p_i)$ , where the softmax is applied over all of the objects in the image. We choose CrossEntropy as the loss function. That is, if the ground truth label of  $r$  is  $l(r) \in [0, \dots, N-1]$ , then the loss function will be,

$$\mathcal{L} = - \sum_r \log(prob_{l(r)}). \quad (12)$$

Table 1. Structures of MLPs. The number after linear and DP (dropout) denotes the dim of the hidden layer and the dropout ratio.

MLPs	Illustration	Structure
$f_e$	encoding one-hot representations of words in 3.1	linear (512)+ReLU
$f_{emb}^v, f_{emb}^l$	encoding the visual and spatial features of nodes in Eq. 5	linear(512)+BN+ReLU+DP(0.4)+linear(512)+BN+ReLU
$f_{emb}^{intra}, f_{emb}^{inter}$	encoding the intra and inter-class edge features in Eq. 6, 7	linear(512)+BN+ReLU+DP(0.4)+linear(512)+BN+ReLU

## 4. Experiments

In this section, we introduce some key implementation details, followed by three experimental datasets. Then we present some quantitative comparisons between our method and existing works. Further, an ablation study shows the effectiveness of the key aspects of our method. Finally, visualisation for LGRANs are shown.

### 4.1. Implementation details

As mentioned in Sec. 3.2.1, we use VGG16 [21] pre-trained on ImageNet [19] to extract visual features for the objects in the image. In this paper, several MLPs are adopted to encode various feature representations. The details of these MLPs are illustrated in Tab. 1. The dimensionalities of the final representations of language representations  $\{s^m\}$  and object representations  $\{x_i^m\}$  are all 512, where  $\{m\}$  denote different components. The training batch size is 30, which means in each training iteration we feed 30 images and all the referring expressions associated with these images to the network. Adam [10] is used as the training optimizer, with initial learning rate to be 0.001, which decays by a factor of 10 every 6000 iterations. The network is implemented based on PyTorch.

### 4.2. Datasets

We conduct experiments on three referring expression comprehension datasets: RefCOCO [9], RefCOCO+ [9] and RefCOCOg [15], which are all built on MSCOCO [13]. The RefCOCO and RefCOCO+ are collected in an iterative game, where the referring expressions tend to be short phrases. The difference between these two datasets is that absolute location words are not allowed in the expressions in RefCOCO+. The expressions in RefCOCOg are longer declarative sentences. RefCOCO has 142,210 expressions for 50,000 objects in 19,994 images, RefCOCO+ has 141,565 expressions for 49,856 objects in 19,992 images, and RefCOCOg has 104,560 expressions for 54,822 objects in 26,711 images.

There are four splits for RefCOCO and RefCOCO, including “train”, “val”, “testA”, “testB”. “testA” and “testB” have different focus in evaluation. While “testA” has multiple persons, “testB” has multiple objects from other categories. For RefCOCOg, there are two data partition versions. One version is obtained by randomly splitting the objects into “train” and “test”. As the data is split by objects, the same image can appear in both “train” and “test”.

Another partition was generated in [16]. In this split, the images are split into “train”, “val” and “test”. We adopt this split for evaluation.

### 4.3. Experimental results

In this part, we show the experimental results on RefCOCO, RefCOCO+ and RefCOCOg. Accuracy is used as evaluation metric. Given an expression  $r$  and a test image  $I$  with a set of regions  $\{o_i\}$ , we use Eq. 11 to select the region with highest matching score with  $r$  as the prediction  $o_{pred}$ . Assume the referent of  $r$  is  $o^*$ , we compute the intersection-over-union (IOU) between  $o_{pred}$  and  $o^*$  and treat the prediction correct if  $IOU > 0.5$ . First, we show the comparison with state-of-the-art approaches on ground-truth MSCOCO regions. That is, for each image, the object regions  $\{o_i\}$  are given. Then, we conduct ablation study to evaluate the effectiveness of two attention components and their combination, *i.e.* node attention, edge attention and graph attention. Finally, the comparison with existing approaches on automatic detected regions are given.

**Overall Results** Tab. 2 shows the comparison between our method and state-of-the-art approaches on ground-truth regions. As can be seen, our method outperforms the other methods on almost all splits. CMN [6] and MattNet [27] are relevant to our method in the sense that they abandon the monolithic language representations and use self-attention mechanism to decompose the language into different parts. However, their approaches are limited by the static and heuristic object representations, which are formed as the stack of multiple features without being informed by the expression query. We use graph attention mechanism to dynamically identify the content relevant to the language and therefore producing more discriminative object representations. ParallelAttn [32] and AccumulateAttn [3] both focus on designing attention mechanisms to highlight the informative content of the language as well as the image to achieve better grounding performance. However, they treat the objects to be isolated and fail to model the relationships between them, which turn out to be important for identifying the object of interest.

**Ablation Study** Next, we conduct an ablation study to further investigate the key components of LGRANs. Specifically, we compare the following solutions:

- Node Representation (NodeRep): this baseline uses LSTM to encode the expression and uses the encod-

Table 2. Performance (Acc%) comparison with state-of-the-art approaches on ground-truth MSCOCO regions. “**Speaker**+listener+reinforcer” and “speaker+**listener**+reinforcer” mean using the speaker or listener module of a joint module [29] to do the comprehension task respectively. All comparing methods use VGG16 features.

Methods	RefCOCO			RefCOCO+			RefCOCOg		
	val	testA	testB	val	testA	testB	val*	val	test
MMI [15]	-	71.72	71.09	-	58.42	51.23	62.14	-	-
visdif [28]	-	67.57	71.19	-	52.44	47.51	59.25	-	-
visdif+MMI [28]	-	73.98	76.59	-	59.17	55.62	64.02	-	-
NegBag [16]	76.90	75.60	78.00	-	-	-	-	-	68.40
CMN [6]	-	75.94	79.57	-	59.29	59.34	69.3	-	-
listener [29]	77.48	76.58	78.94	60.5	61.39	58.11	71.12	69.93	69.03
<b>speaker</b> +listener+reinforcer [29]	78.14	76.91	80.1	61.34	63.34	58.42	72.63	71.65	71.92
speaker+ <b>listener</b> +reinforcer [29]	78.36	77.97	79.86	61.33	63.1	58.19	72.02	71.32	71.72
VariContxt [30]	-	78.98	82.39	-	62.56	62.90	73.98	-	-
ParallelAttn [32]	81.67	80.81	81.32	64.18	66.31	61.46	69.47	-	-
AccumulateAttn [3]	81.27	81.17	80.01	65.56	<b>68.76</b>	60.63	73.18	-	-
MattNet [27]	80.94	79.99	82.3	63.07	65.04	61.77	73.08	73.04	72.79
Ours-LGRANs	<b>82.0</b>	<b>81.2</b>	<b>84.0</b>	<b>66.6</b>	67.6	<b>65.5</b>	-	<b>75.4</b>	<b>74.7</b>

Table 3. Ablation study of key components of LGRANs.

Methods	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
NodeRep	77.6	77.7	77.8	61.5	62.8	58.0	67.1	68.4
GraphRep	80.2	79.4	81.5	63.3	64.4	61.9	70.5	72.1
NodeAttn	81.4	80.4	82.8	65.8	66.2	64.2	72.4	73.2
EdgeAttn	81.9	80.8	83.3	65.9	66.7	64.9	73.9	74.5
LGRANs	82.0	81.2	84.0	66.6	67.6	65.5	75.4	74.7

ings of node features to represent the objects, *i.e.*  $\mathbf{x}_i^{e,obj}$  in Eq. 5.

- **Graph Representation (GraphRep)**: apart from node representation, graph representation uses two other types of edge representations: pooling of the intra-class edge features  $\hat{\mathbf{x}}_i^{pool,intra} = \sum_{j \in \mathcal{N}_i^{intra}} \mathbf{e}_{ij}^{intra}$ , where  $\mathbf{e}_{ij}^{intra}$  is the intra-class edge feature encoding in Eq. 6, and pooling of inter-class edge feature  $\hat{\mathbf{x}}_i^{pool,inter} = \sum_{j \in \mathcal{N}_i^{inter}} \mathbf{e}_{ij}^{inter}$ , where  $\mathbf{e}_{ij}^{inter}$  is the inter-class edge feature encoding in Eq. 7.
- **NodeAttn**: on top of graph representation, NodeAttn applies node attention as introduced in Sec. 3.2.3.
- **EdgeAttn**: different from graph representation that directly aggregates the edge features, EdgeAttn applies edge attention to the edges, as introduced in Sec. 3.2.3.
- **LGRANs**: this is our full model, which applies both node attention and edge attention on the graph.

Tab. 3 shows the ablation study results. The limitation for the baseline “**Node Representation**” is that it treats the objects to be isolated and ignores the relationships between objects. The “**Graph Representation**” considers the relationships between objects by pooling the edge features

Table 4. Performance (Acc%) comparison with state-of-the-art approaches on automatically detected regions. All comparing methods use VGG16 features.

Methods	RefCOCO		RefCOCO+		RefCOCOg
	testA	testB	testA	testB	val*
MMI [15]	64.9	54.51	54.03	42.81	45.85
NegBag [16]	58.6	56.4	-	-	39.5
CMN [6]	71.03	65.77	54.32	47.76	57.47
listener [29]	71.63	61.47	57.33	47.21	56.18
<b>spe</b> +lis+Rl [29]	69.15	61.96	55.97	46.45	57.03
spe+ <b>lis</b> +RL [29]	72.65	62.69	58.68	48.23	58.32
VariContxt [30]	73.33	<b>67.44</b>	58.40	53.18	62.30
ParallelAttn [32]	75.31	65.52	61.34	50.86	58.03
LGRANs	<b>76.6</b>	66.4	<b>64.0</b>	<b>53.4</b>	<b>62.5</b>

directly. It observes some improvement comparing to the baseline. “**NodeAttn**” takes a step further by applying the region level attention to highlight the potential object described by the expression and this further improves the performance. Orthogonal to “NodeAttn”, “**EdgeAttn**” identifies the relationships relevant to the expression and this strategy results in a boost up to 3.4%. “**GraphAttn**” is our full model. As can be seen, it consistently outperforms the other incomplete solutions above.

**Automatically detected regions** Finally, we evaluate the performance of LGRANs using automatically detected object regions from Faster R-CNN [17]. Tab. 4 shows the results. Comparing to using ground-truth regions, the performance of all methods drops, which is due to the quality of the detected regions. In this setting, **LGRANs** still performs consistently better than other comparing methods. This shows the capacity of LGRANs in fully automatic referring expression comprehension.

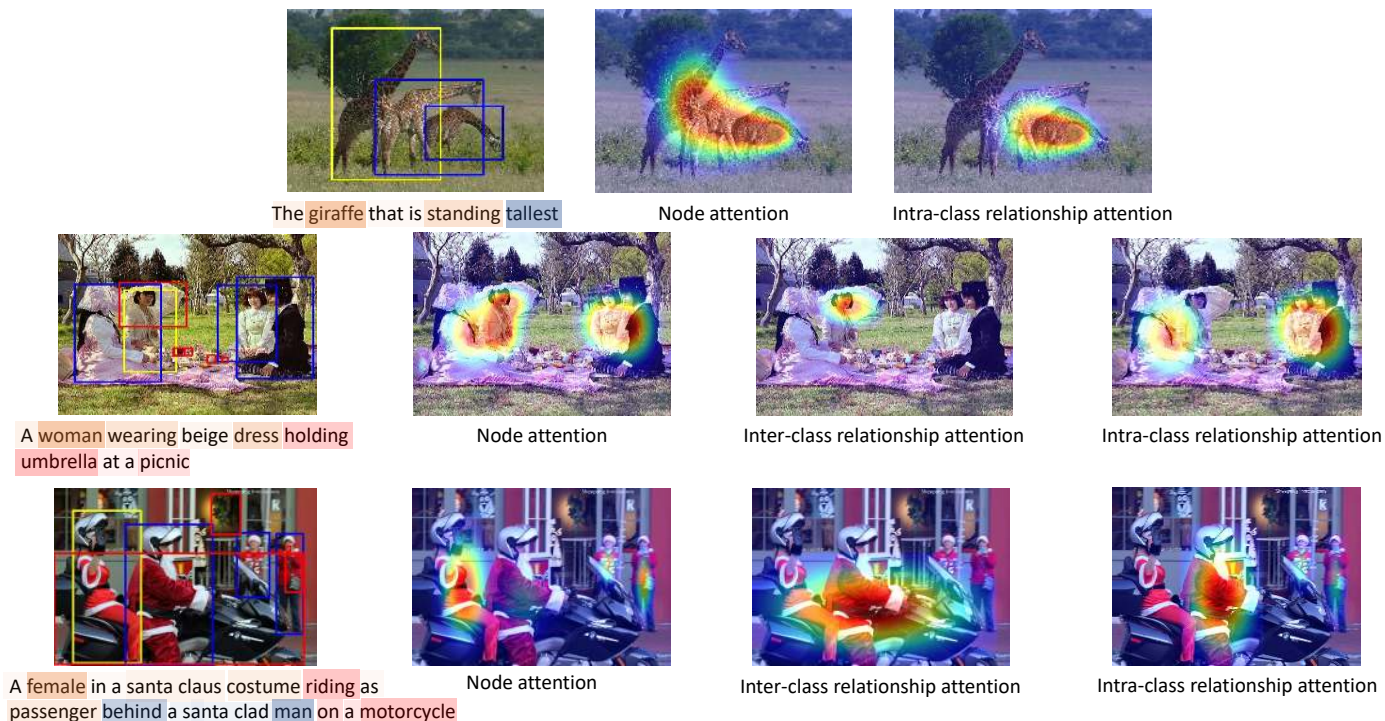


Figure 4. Visualisation for LGRANS. Three examples with variant difficulty levels are shown. For each example, the original image (the referent, its intra- and inter-class neighbourhood are marked by yellow, blue and red boxes respectively) with referring expression (subject, intra- and inter-class relationships are marked by yellow, blue and red), the node attention maps, the inter- and intra-class edge attention maps are given from left to right. We visualise a relationship between referent and other objects by highlighting the other objects that interact with the referent. For example, for relationship “woman holding umbrella”, we highlight “umbrella”. Since the giraffe example contains giraffe regions only, no inter-class relationships exist. Within each region, the attention value is smoothed by a 2D Gaussian kernel with the centre to be the region centre. Best viewed in colour.

#### 4.4. Visualisation

In contrast to conventional attention schemes that apply on isolated image regions, *e.g.* uniform grid of CNN feature maps [26] or object proposals [1], LGRANS simultaneously predict attention distributions over objects and inter-object relationships.

Fig. 4 shows three examples with variant difficulty levels. In the first example, *node attention* highlights all three giraffes and thus cannot distinguish the referent. To identify the tallest giraffe, it needs to compare one giraffe to the other two. As seen, the *intra-class relationship attention* highlights the relationships to the other two giraffes and provides useful clues to make correct localisation. In the second example, there are four women in the image. *Node attention* puts attention on people and excludes other objects, *e.g.* bag, bottle, umbrella. Then *inter-class relationship attention* identifies a relevant relationship between the referent and an umbrella. Since there are no *intra-class relationships* present in the expression, the intra-class edge attention values almost evenly distribute on other persons. In the last example, the *intra-class attention* and *inter-class*

*attention* identify man and motorcycle respectively, which correspond to “behind a Santa clad man” and “on a motorcycle”. In these examples, the useful information present in the expression is highlighted and this explains why an object is selected as referent.

#### 5. Conclusion

We proposed a graph-based, language-guided attention networks (LGRANS) to address the referring expression comprehension task. LGRANS is composed of two key components: a *node attention* component and an *edge attention* component, both guided by the *language attention*. The *node attention* highlights the referent candidates, narrowing down the search space for localising the referent, and the *edge attention* identifies the relevant relationships between the referent and its neighbourhood. Based on the attended graph, we can dynamically enrich the object representation that better adapts to the referring expression. Another benefit of LGRANS is that it renders the comprehension decision to be visualisable and explainable.



## References

- [1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.
- [2] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [3] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan. Visual grounding via accumulated attention. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.
- [4] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015.
- [5] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. *Proc. IEEE Int. Conf. Comp. Vis.*, 2017.
- [6] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.
- [7] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [8] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [9] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *Proc. Conf. Empirical Methods in Natural Language Processing*, 2014.
- [10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learn. Representations*, 2014.
- [11] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler. Situation recognition with graph neural networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017.
- [12] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel. Gated graph sequence neural networks. In *Proc. Int. Conf. Learn. Representations*, 2016.
- [13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, 2014.
- [14] R. Luo and G. Shakhnarovich. Comprehension-guided referring expressions. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.
- [15] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [16] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *Proc. Eur. Conf. Comp. Vis.*, 2016.
- [17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015.
- [18] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *Proc. Eur. Conf. Comp. Vis.*, 2016.
- [19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vision*, 2015.
- [20] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997.
- [21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [22] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.
- [23] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *Int. J. Comput. Vision*, 2013.
- [24] P. Velickovi, G. Cucurull, A. Casanova, A. Romero, P. Li, and Y. Bengio. Graph attention networks. In *Proc. Int. Conf. Learn. Representations*, 2018.
- [25] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [26] P. Wang, L. Liu, C. Shen, Z. Huang, A. van den Hengel, and H. Tao Shen. Multi-attention network for one shot learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.
- [27] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. MATTNET: Modular attention network for referring expression comprehension. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.
- [28] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *Proc. Eur. Conf. Comp. Vis.*, 2016.
- [29] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.
- [30] H. Zhang, Y. Niu, and S.-F. Chang. Grounding referring expressions in images by variational context. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.
- [31] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017.
- [32] B. Zhuang, Q. Wu, C. Shen, I. Reid, and A. van den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.
- [33] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proc. Eur. Conf. Comp. Vis.*, 2014.