Neurocomputing 413 (2020) 51–60

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

vtGraphNet: Learning weakly-supervised scene graph for complex visual grounding

Fan Lyu [a], Wei Feng [a,*], Song Wang [a,b]

[a] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[b] Computer Science and Engineering, University of South Carolina, SC 29208, USA

ARTICLE INFO

Article history:
Received 11 November 2019
Revised 29 March 2020
Accepted 24 June 2020
Available online 7 July 2020
Communicated by Zhu Jianke

Keywords:
Visual grounding
Referring expression comprehension

ABSTRACT

As a challenging cross-modal task, current visual grounding is usually addressed by directly analyzing the unstructured scene and matching the query text with all region proposals, which is prone to errors, especially when the scene and/or query text are complex. In this paper, we study such complex visual grounding problem and propose to build a query dependent visual-textual (VT) scene graph to jointly understand the image and query text. To avoid the difficulty of obtaining ground-truth scene graphs, we propose vtGraphNet to effectively learn the bi-modal scene graph in a weakly-supervised way, where the only supervision is the manually annotated grounding region. Specifically, we first use an ARU Tagging model to sequentially tag every query word as either an attribute, a relationship or an auxiliary. If a word is tagged as attribute, we develop an attribute-assigning model to associate it to a region proposal. If a word is tagged as relationship, we develop a relationship-referring model to associate it to a pair of region proposals. A simple yet effective graph consistency loss function is constructed to constrain the above associations to form a feasible compact VT scene graph, from which discriminative region features can be extracted and used to locate the grounding object by classification. Extensive experiments on benchmark datasets validate the superiority of our approach in handling both simple and complex visual grounding tasks.

1. Introduction

Using text description to automatically localize an object of interest from an image, visual grounding (VG) is a challenging cross-modal task that has attracted lots of interests in both computer vision and natural language processing areas. It plays an important role in human–machine interaction systems [1] and intelligent robotic manipulation [2].

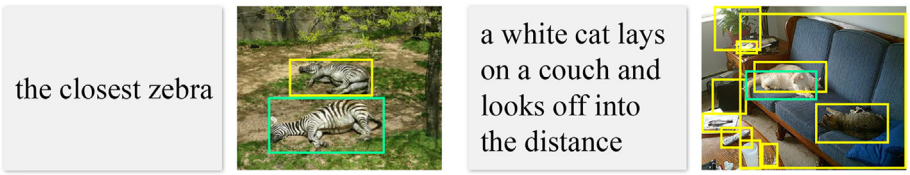
In recent years, the research on visual grounding has achieved significant progress using Deep Neural Networks (DNNs) [3–10]. Some recent works [11–13] seek to fuse features from both visual and textual modalities, and treat visual grounding as a classification problem. Another type of effective methods [14,15] aims to reconstruct a caption for each region proposal, and uses textual similarity between the reconstructed caption and the query text to identify the most relevant region. Recently, visual cues and syntax are jointly considered decomposing the query text into key items w.r.t. the region proposals [16–18]. Note, despite the detailed variances, most state-of-the-art visual grounding methods follow a similar *direct matching strategy*, which analyzes the image and selects the region proposal best matching the query as the prediction object.

This direct matching strategy is skilled in simple VG, however, is prone to failure for complex VG (Fig. 1a). This is mainly because complex VG may be set in complex scene containing visually similar objects and cluttered backgrounds. Also, highly structured and ambiguous expressions may involve a lot of reference objects, which may undermine a VG model. For instance, Fig. 1a shows a complex VG example where the couch and two cats are easily misrecognized due to the complex indoor environment and the long expressions with reference objects.

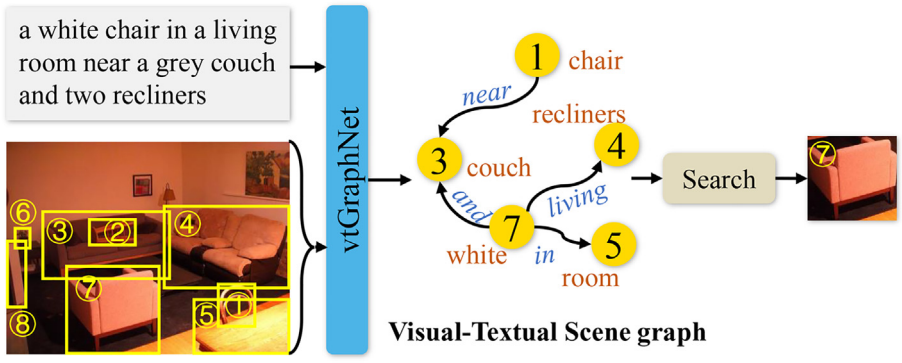
In this paper, we propose to mitigate the complex VG challenge by modeling visual-textual attributes and relationships (Fig. 1b). Specifically, we propose a query dependent visual-textual (VT) scene graph to jointly understand the image and query text, through a new DNN model vtGraphNet to effectively learn the bi-modal VT scene graph. In practice, the first part of vtGraphNet is an ARU Tagging model, sequentially classifying every query word to three tags, i.e., Attribute, Relationship or aUxiliary. We then design an attribute-assigning model and a relationship-

* Corresponding author.
E-mail addresses: fanlyu@tju.edu.cn (F. Lyu), wfeng@tju.edu.cn (W. Feng), songwang@cec.sc.edu (S. Wang).

https://doi.org/10.1016/j.neucom.2020.06.091
0925-2312/



(a) Simple VG (left) vs. complex VG (right)



(b) The proposed VT scene graph for VG task

Fig. 1. (a) Unlike simple VG, complex VG deals with more complex scenes, longer and complicated descriptions. That is, many reference objects and their relationships need to be considered (the green rectangles mean the correct objects). (b) The proposed vtGraphNet can produce a VT scene graph, which will help to localize the correct object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

referring model to respectively associate region proposals to corresponding words tagged as attribute and relationship. To guarantee an effective and compact VT scene graph, a graph consistency loss function is constructed to constrain the association of attribute and relationship region proposals. Finally, from the VT scene graph, region features are extracted via cross graph representation, which is used to locate an object through classification. To avoid the expensive cost of obtaining ground-truth scene graphs, the proposed vtGraphNet is trained in a weakly-supervised manner, with the only supervision being ground-truth grounding regions. Since our vtGraphNet captures both the attribute and relationship information from visual and textural modalities, as validated by the experimental results on benchmark datasets, our approach outperforms state-of-the-art methods in both simple and complex visual grounding tasks.

2. Related work

2.1. Visual grounding

Visual grounding (VG), also known as referring expression comprehension, seeks to locate the most related object by a given language query, which draws much concern in recent years for its potential applications on man–machine interactive. Some earlier studies [19,20] are inspired by real world requirements that learn to map natural language statements to their referents. Recently, some researchers formulated VG as selecting the best regions from a set of given proposals[11,21,22,13,12,23] by fusing features from multiple modalities. Reconstruction strategy is also popular in VG task, such as [14] that propose to localize phrases in images by reconstructing the phrase with none, little and full supervision. In addition, some researchers [24,25,15] locate object in the weakly-supervised way, which is able to find the target object with a heat map or other forms. Besides, some works [18,17,16,25] seek to ground referring expression by parsing it with some specific rules. Specifically, [22,17] consider relationship only in a small local area for each object, but never aggregate semantic relationships among objects. However, most previous studies focus on matching query with every object but rarely take global relationship into account, which makes them prone to fail in complex visual grounding.

2.2. Scene graph generation

Graph-structured representations [26,27] have attained widespread use in computer graphics to efficiently represent compositional scene [28,29], which often makes use of Markov Random Field [30–32]. Scene graph, proposed by Ref. [33], leverages a graph to represent the relationships (edges) among the regions (nodes) in a given image. Graph structure can be applied to many visual applications [34,35,33], especially tasks crossing computer vision and natural language processing [33,36,35]. Scene graph generation, a primal task of computer vision, builds up such graph, and gets a lot of research [37–39] in the past few years. However, most of these methods need a fully supervised training process, which requires numerous professional labeling and time cost, which significantly hinder the promotion of scene graph. The primary obstacle is the difficulty of obtaining semantic relationships among objects. In this paper, we construct visual-textual (VT) scene graph from the given query in a weakly-supervised way, which is verified helpful to visual grounding.

3. Methodology

Given a query, *i.e.*, a sequence of words $\mathcal{Q} = \{q_1, \ldots, q_T\}$ and an image I, visual grounding aims to locate a query related object from I. Region proposals extracted from I denote as $\mathcal{B} = \{b_1, \ldots, b_C\}$, where C is the number of region proposals. Then, visual grounding problem becomes how to select the query related region b^* from $\mathcal{B}$. In contrast to existing methods that ignore the relationship among region proposals, we propose a scene graph based visual grounding method (see Fig. 2)where the relationships among region proposals are effectively presented by a visual-textual graph network

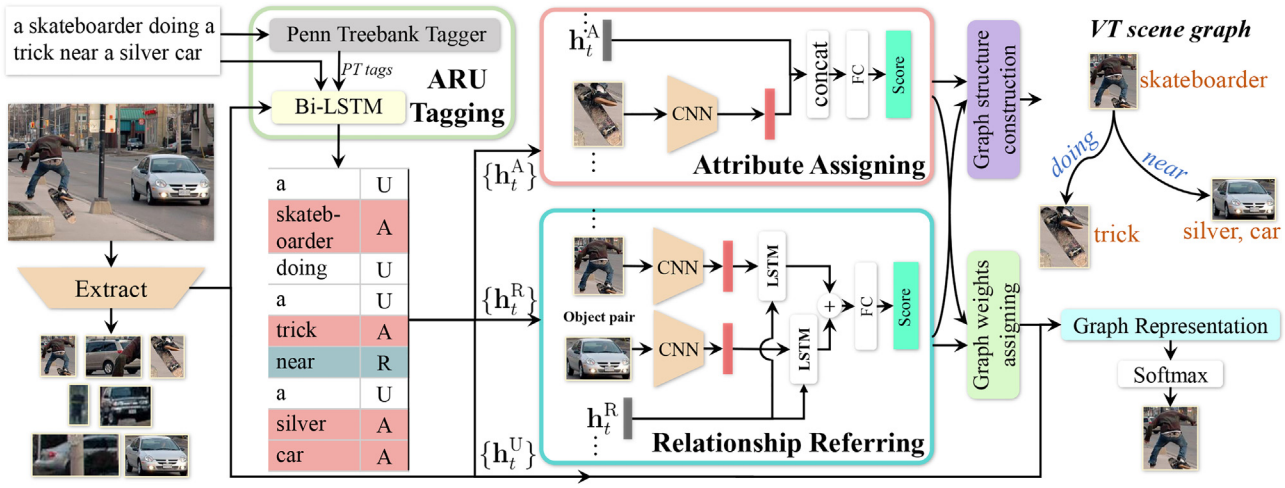


Fig. 2. Pipeline of the proposed vtGraphNet for complex visual grounding. We first use a proper CNN model to extract image features and region proposals. Then, an ARU Tagging model is used to sequentially identify the tag of each word. If a word is regarded as (A)ttribute (*e.g.*, "trick"), it is assigned to corresponding object proposals by an attribute-assigning model. If a word is recognized as (R)elationship (*e.g.*, "near"), it will be associated to a pair of objects by a relationship-referring model. Grounding objects is obtained by classification using graph cross representation as features. (Best view in color.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(vtGraphNet) jointly considering textual attributes and relationships from query.

3.1. VT scene graph formulation

Visual-textual scene graph. A VT scene graph is denoted as $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{V}$ is the set of nodes representing region proposals, and $\mathcal{E}$ is the set of directed edges between those region proposals. $\mathcal{A}$ is the set of textual attributes for nodes, and indicates which attribute words in $\mathcal{Q}$ should be included into a node of $\mathcal{V}$. $\mathcal{R}$ is the set of textual relationships for edges and presents which relationship words in $\mathcal{Q}$ should be assigned to an edge of $\mathcal{E}$. Note, a region proposal could have several attributes, and an edge could have several relationships too. Our goal is to learn how to transform a complicated image I and its proposals $\mathcal{B}$ to such a VT scene graph with a query $\mathcal{Q}$, and locate the correct region corresponding to $\mathcal{Q}$ by this graph.

Region proposal generation. Region proposals in $\mathcal{B}$ are the bounding boxes in image I and can be obtained by object detectors [40,41], or proposal generating models [42,43]. In our implementation, we use SSD [41] as our proposal extractor. Also, following some state-of-the-arts [22,44,45], we evaluate the object candidates provided by datasets (Ground-truth bounding boxes). We denote features of all region proposals as a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_C]$, each column of which is the feature vector of a region proposal extracted from a pre-trained CNN. We also generate the high-level feature $\mathbf{f}_I$ of the image I via the same CNN. Note, we never use the label of bbox from detector.

3.2. ARU tagging

We construct an ARU tagging module to classify all words into three different tags: **Attribute** (inherent description for an object, such as "woman" or "young"), **Relationship** (correlation between a pair of objects, such as "right" or "holding") and **aUxiliary** (structural auxiliary word, such as "the" or "of"), which are denoted as A, R and U respectively and make up a tagset, *i.e.* $\mathcal{T} = \{\mathrm{A}, \mathrm{R}, \mathrm{U}\}$. This is something like Part-Of-Speech (POS) taggers [46–48] that syntactically assigns each word in a sentence an appropriate part of speech tag of a tagset, such as the Penn Treebank [49] (the tagset is shown in Table 4). Even so, it is inappropriate to hard label each word only accords to the tag from POS Tagger. For example, in most situations, a noun should be the subject because it is always the carrier of an action or an attribute, thus be tagged as A (Attribute). But a noun can also be the relationship between two subjects such as "the friend (A/R) in red of the girl". Another example is the words of position such as "right" and "top" that they can be labeled as Relationship to represent spatial relationship, but can also be labeled as Attribute of some subjects ("the guy in green shirt to the right (A/R) of the little girl"). Consider these situations, we propose to use a Bi-LSTM [50] to automatically assign A, R and U to each word guided by the PT labels.

Thus, we design an ARU tagger with the input query $\mathcal{Q}$, image feature $\mathbf{f}_I$ and Penn Treebank (PT) tag of each word. Specifically, we first initialize Bi-LSTM with $\mathbf{f}_I$. Then, given the t-th word q_t and its PT tag, we represent this word by the concatenation of hidden states of forward and backward LSTMs, *i.e.*, $\mathbf{h}_t = \left[\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t}\right]$. We assign q_t a tag $\tau_t \in \mathcal{T}$ by putting $\mathbf{h}_t$ into an fc layer followed with softmax function. All words labeled by tag A, R and U can be grouped to three subsets of $\mathcal{Q}$ denoted as $\mathcal{Q}^{\mathrm{A}}, \mathcal{Q}^{\mathrm{R}}$ and $\mathcal{Q}^{\mathrm{U}}$ respectively.

3.3. Weakly-supervised VT scene graph generation

Most scene graph generating works [37–39] are under full supervision. That is, we must provide the ground truth scene graph for each image to learn an effective generation model, which is unavailable in most real-world situations. In contrast, we explore to build scene graph in a labor-saving and weakly-supervised way where only grounding annotations are available. We will define the query-related nodes $\mathcal{V}$ and edges $\mathcal{E}$, their corresponding attributes $\mathcal{A}$ and relationships $\mathcal{R}$ from image, query and region proposals. We regard this requirement as a two-step process named attributes assigning and relationship referring.

Attributes assigning. We consider the t-th word q_t in $\mathcal{Q}$ is an attribute in the image I if we have $\tau_t = \mathrm{A}$ according to the ARU tagging. An attribute word can be a noun, *e.g.* "girl" and "dog", an adjective, *e.g.* "red" and "little", an adverb, *e.g.* "carefully" and "happily", or other related words. In graph $\mathcal{G}, \mathcal{A}$ indicates which attribute words selected from the given query $\mathcal{Q}$ should be included in a node in $\mathcal{V}$, which is equivalent to assigning attributes

words to a region proposal. To this end, by concatenating the t-th hidden state $\mathbf{h}_t^{\mathrm{a}}$ and the feature of a proposal b_c, and passing them into an fc layer with ReLU, we can obtain the representation $\mathbf{f}_{c,t}^{\mathrm{a}}$ of the pair of region proposal and attribute word $\langle b_c, q_t\rangle$ as well as their matching score by softmax. The proposal with the highest score will be structurally assigned attribute q_t. As a result, for each attribute word, we obtain the probability vector $\mathbf{w}_t^{\mathrm{a}} \in \mathcal{R}^{1\times C}$ and the indicated one-hot vector $\mathbf{m}_t^{\mathrm{a}} \in \mathcal{R}^{1\times C}$. Furthermore, we get

$$\mathbf{M}^{\mathrm{a}} = \mathrm{diag}(\mathbf{m}^{\mathrm{a}}), \tag{1}$$

where $\mathbf{m}^{\mathrm{a}}$ is computed by applying max-pooling to $\{\mathbf{m}_t^{\mathrm{a}}\}$ along all attribute words. Easy to know, the diagonal binary matrix $\mathbf{M}^{\mathrm{a}} \in \mathbb{R}^{C\times C}$ represents the most possible assigned proposals for all attribute words, while $\mathbf{W}^{\mathrm{a}} \in \mathbb{R}^{C\times T}$ represents the weights across all pairs attribute word and proposal. By attribute assigning, the region proposals that are assigned an attribute word constitute a subset denoted as $\mathcal{V}_{\mathrm{A}}$ (from $\mathbf{m}^{\mathrm{a}}$), and assigned attributes make up the set $\mathcal{A}$. For a proposal b_c, we calculate its attribute representation $\mathbf{f}_c^{\mathrm{a}}$ by

$$\mathbf{f}_c^{\mathrm{a}} = \phi_t\left(\left\{\mathbf{W}_{(c,t)}^{\mathrm{a}}\mathbf{f}_{c,t}^{\mathrm{a}}\right\}\right), \quad \mathbf{W}^{\mathrm{a}} = \left[\mathbf{w}_1^{\mathrm{a}}, \ldots, \mathbf{w}_T^{\mathrm{a}}\right], \tag{2}$$

where ϕ_t means the max-pooling along the dimension t. $\mathbf{f}_c^{\mathrm{a}}$ is used to represent graph $\mathcal{G}$, so as to embed the textual attribute information effectively, as introduced later.

Relationship referring. For the t-th word q_t in ARU tagging, if we have $\tau_t = \mathrm{R}$, it will be considered as a type of relationship *w.r.t.* an edge. In a graph $\mathcal{G}, \mathcal{R}$ indicates which relationship words should be assigned to an edge in $\mathcal{E}$, which is equivalent to referring relationship among all region proposals according to the relationship words, *i.e.* $\mathcal{Q}^{\mathrm{R}}$. This directed edge can be denoted as a triplet $\langle b_c, q_t, b_{c'}\rangle$ where b_c and $b_{c'}$ are two region proposals in $\mathcal{B}$ and re-denoted as "subject" and "object" respectively to represent the direction of the edge. The problem becomes to assign each $q_t \in \mathcal{Q}^{\mathrm{R}}$ to a proper edge to accurately represent relationship between region proposals. A simple solution is to regard such problem as a classification task for relationship words, which, however is difficult due to the extremely large size of edge set. Hence, we take a trade-off by decomposing the problem to two subproblems by first matching $\langle b_c, q_t\rangle$ and $b_{c'}$ and then matching b_c and $\langle q_t, b_{c'}\rangle$. To this end, we first construct an LSTM with the inputs b_c and $\mathbf{h}_t^{\mathrm{r}}$ to get the probability of $b_{c'} \in \mathcal{B}$ being related to b_i according to q_t, and denoted as $P(b_{c'}|b_c, \mathbf{h}_t^{\mathrm{r}})$. Then, by considering the negative direction, the probability of b_c being related to $b_{c'}$ according to q_t can be also calculated via another LSTM, *i.e.*, $P(b_c|\mathbf{h}_t^{\mathrm{r}}, b_{c'})$. For each relationship word in $\mathcal{Q}^{\mathrm{R}}$, we then obtain two one-hot matrices $\overrightarrow{\mathbf{M}}_t^{\mathrm{r}} \in \mathbb{R}^{C\times C}$ and $\overleftarrow{\mathbf{M}}_t^{\mathrm{r}} \in \mathbb{R}^{C\times C}$ and two probability matrices $\overrightarrow{\mathbf{W}}_t^{\mathrm{r}} \in \mathbb{R}^{C\times C}$ and $\overleftarrow{\mathbf{W}}_t^{\mathrm{r}} \in \mathbb{R}^{C\times C}$ that defines the relationship structures and matching scores between arbitrary two region proposals in $\mathcal{B}$, where $\rightarrow$ and $\leftarrow$represents the direction of relationship is from b_c to $b_{c'}$, or the opposite. Consequently, the ensemble matrix representing relationship structure of each triplet can be computed by

$$\mathbf{M}_t^{\mathrm{r}} = \overrightarrow{\mathbf{M}}_t^{\mathrm{r}} + \left(\overleftarrow{\mathbf{M}}_t^{\mathrm{r}}\right)^{\top}. \tag{3}$$

Note the diagonal of $\mathbf{M}_t$ are zeros, due to no object is related to itself, and if a word is not in $\mathcal{Q}^{\mathrm{R}}$, the matrices are a zero matrix. Then, we generate the relationship structure matrix $\mathbf{M}^{\mathrm{r}}$ by max-pooling all $\{\mathbf{M}_t^{\mathrm{r}}\}$ along the dimension t,

$$\mathbf{M}^{\mathrm{r}} = \phi_t(\{\mathbf{M}_t^{\mathrm{r}}\}). \tag{4}$$

We use $\mathcal{V}_R$ to represent the node set whose nodes are all included into the edges assigned relationship words. With the two direction's probability matrices $\overrightarrow{\mathbf{W}}_t^{\mathrm{r}}$ and $\overleftarrow{\mathbf{W}}_t^{\mathrm{r}}$, we can calculate the relationship representation of b_c by

$$\mathbf{f}_c^{\mathrm{r}} = \phi_t\left(\left\{\overrightarrow{\mathbf{f}}_{c,t}^{\mathrm{r}}\right\}\right) + \phi_t\left(\left\{\overleftarrow{\mathbf{f}}_{c,t}^{\mathrm{r}}\right\}\right), \tag{5}$$

where

$$\overrightarrow{\mathbf{f}}_{c,t}^{\mathrm{r}} = \sum_{c'}\left(\left\{\overrightarrow{\mathbf{W}}_{t(c,c')}^{\mathrm{r}}\mathbf{f}_{(c,t,c')}\right\}\right), \quad c' \in \{1, \ldots, C\}, \tag{6}$$

$$\overleftarrow{\mathbf{f}}_{c,t}^{\mathrm{r}} = \sum_{c'}\left(\left\{\overleftarrow{\mathbf{W}}_{t(c',c)}^{\mathrm{r}}\mathbf{f}_{(c',t,c)}\right\}\right), \quad c' \in \{1, \ldots, C\}. \tag{7}$$

ϕ_t and $\sum_{c'}$ are the max-pooling function and summation along the dimension t and c', respectively. Note for the proposal b_c and $b_{c'}$, we can compute the relationship representations of them in the left and right direction by

$$\mathbf{f}_{(c,t,c')}^{\mathrm{r}} = \overrightarrow{g}(\mathbf{b}_{c'}, \mathbf{h}_t), \quad \mathbf{f}_{(c',t,c)}^{\mathrm{r}} = \overleftarrow{g}(\mathbf{b}_{c'}, \mathbf{h}_t), \tag{8}$$

where $\overrightarrow{g}$ and $\overleftarrow{g}$ are two different MLPs. In Eq. (5), we obtain the relationship representation of region proposal b_c, which will help construct an effective graph representation in scene graph generation. Because of the proposed ARU tagger, the relationships in our scene graph are different from the previous relationship detection model [51,52] and scene graph model [37,39,38]. They take a "subject-predicate-object" template, and the relationships are always a predicate verb. In contrast, the relationship words in our model are up to the ARU tagging.

VT Scene graph generation. Given a query $\mathcal{Q}$, by assigning objects and referring relationships, a VT scene graph $\mathcal{G} = \{\mathcal{V}_A \cup \mathcal{V}_R, \mathcal{E}, \mathcal{A}, \mathcal{R}\}$ is generated. Meanwhile, through Eqs. (1) and (4), a graph structure matrix $\mathbf{M}^{\mathrm{ar}} = \mathbf{M}^{\mathrm{a}} + \mathbf{M}^{\mathrm{r}}$ is constructed to represent all nodes and edges recognized by the proposed model, and we call this operation *Graph Structure Construction*. However, the graph will be unnatural if $\mathcal{V}_A$ and $\mathcal{V}_R$ are with a big gap, in other words, $\mathcal{V}_A$ and $\mathcal{V}_R$ may contain totally different region proposals, which would lead to the recognized attributes and relationships are thoroughly irrelevant. To mitigate this, a graph consistency loss is proposed

$$L_{\mathcal{G}}(\mathbf{m}^{\mathrm{a}}, \mathbf{m}^{\mathrm{r}}) = -\frac{1}{C}\sum_{c=1}^{C} z_c, \tag{9}$$

$$z_c = \begin{cases} 1, & \mathbf{m}_c^{\mathrm{a}} = \mathbf{m}_c^{\mathrm{r}} = 1, \\ 0, & \mathbf{m}_c^{\mathrm{a}} \neq \mathbf{m}_c^{\mathrm{r}}, \\ 0.5, & \mathbf{m}_c^{\mathrm{a}} = \mathbf{m}_c^{\mathrm{r}} = 0, \end{cases} \tag{10}$$

where $\mathbf{m}^{\mathrm{a}}$ consists of the diagonal elements of $\mathbf{M}^{\mathrm{a}}$, and $\mathbf{m}^{\mathrm{r}}$ is a joint vector of $\mathbf{M}^{\mathrm{r}}$ computed by

$$\mathbf{m}^{\mathrm{r}} = \left[\max\left(\mathbf{m}_1^{\mathrm{r(v)}}, \mathbf{m}_1^{\mathrm{r(h)}}\right), \ldots, \max\left(\mathbf{m}_C^{\mathrm{r(v)}}, \mathbf{m}_C^{\mathrm{r(h)}}\right)\right], \tag{11}$$

where $\mathbf{m}^{\mathrm{r(v)}} = \max\left(\mathbf{M}_{(:,j)}^{\mathrm{r}}\right)$ and $\mathbf{m}^{\mathrm{r(h)}} = \max\left(\mathbf{M}_{(i,:)}^{\mathrm{r}}\right)$. Note, in our implementation, we use a Gaussian function $(f(x_i) = \exp\left(10000(x_i - p)^2\right)$, where p is the index indicating the maximum value of input $x = [x_1, \cdots, x_n]$) to approximate the original one-hot operation to ensure the consistency loss is differentiable for the model and the gradient can be backproped. To effectively represent an VT scene graph, we first use *Graph Weights Assigning* (Eq. (2) and (5))) to obtain the attribute feature and relationship feature of a region proposal b_c. Then, we take a cross strategy to represent visual-textual scene graph $\mathcal{G}$ named *Cross Graph Representation*

$$\mathbf{G} = [\mathbf{g}_1^\top, \ldots, \mathbf{g}_C^\top], \tag{12}$$

where

$$\mathbf{g}_c = g_G\left(\mathbf{f}_c^{\mathrm{a}}, \mathbf{f}_c^{\mathrm{r}}, \phi_t\left(\left\{\mathbf{h}_t^{\mathrm{U}}\right\}\right)\right), c \in \{1, \ldots, C\} \tag{13}$$

where g_G is an MLP network (see Fig. 3).

3.4. Visual grounding

We have obtained a visual-textual scene graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}\}$ in the previous section, where $\mathcal{V} = \mathcal{V}_A \cup \mathcal{V}_R$. To locate the most related region in a given image, we take the graph representation $\mathbf{G}$ and all query hidden state $\{\mathbf{h}_t\}$ in ARU tagging as input. A co-attention model [53] is used to model the cross modal attention between them:

$$\tilde{\mathbf{q}} = \sum_{t=1}^{T} \alpha_t^q \mathbf{h}_t, \quad \tilde{\mathbf{g}} = \sum_{i=c}^{C} \alpha_c^g \mathbf{g}_c, \tag{14}$$

$$\alpha_t^q = \mathrm{softmax}(F_q(\mathbf{h}_t)), \quad \alpha_c^g = \mathrm{softmax}(F_g(\mathbf{g}_c, \tilde{\mathbf{q}})). \tag{15}$$

F_q and F_g are two MLPs to embed query and graph information. The grounding result b^{pred} can be obtained by the maximum index of the probability vector

$$\mathbf{p} = \mathrm{softmax}(\tilde{\mathbf{g}}\mathbf{B}^\top). \tag{16}$$

Consequently, we define our loss function as:

$$L = -\frac{1}{N}\sum_{n}^{N}\left(\log P\left(b_n^{\mathrm{pred}} = b_n^* | \mathcal{Q}_n, I_n, \mathcal{B}_n\right) + \alpha L_{\mathcal{G}} + \beta L_{\mathcal{Q}}\right), \tag{17}$$

where $L_{\mathcal{G}}$ and $L_{\mathcal{Q}}$ are the graph consistency loss and textual consistency loss, respectively. α and β are the coefficients of two losses. b_n^* is the target correct region proposal. Inspired by the reconstruction structure [22,24,15], and to ensure the reconstructed sentence is the same as the input query, we build a textual consistency loss by an extra LSTM

$$L_{\mathcal{Q}} = -\frac{1}{T-1}\sum_{t=1}^{T-1}\log(P(q_{t+1}|(q_t, \ldots, q_1), \mathbf{f}_I, \mathbf{G})), \tag{18}$$

Unlike previous methods reconstructing from predicted region, we consider to reconstruct text description from the graph representation $\mathbf{G}$.

4. Experiments

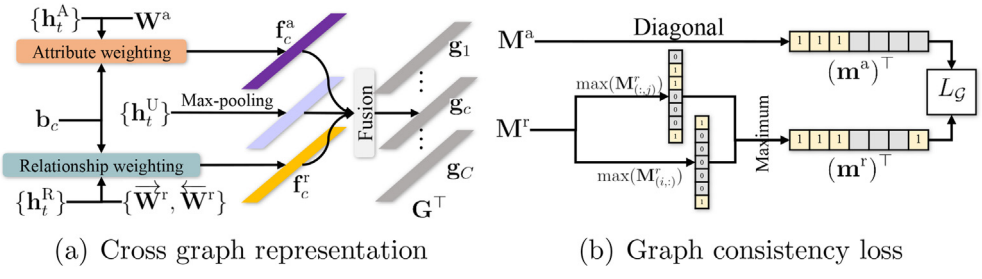
4.1. Datasets and implementation details

We evaluate the proposed framework on three datasets based on MS-COCO [56], *i.e.* ReferCOCO, ReferCOCO + and ReferCOCOg [57]. In the three datasets, the proposals and expressions labeled by labors are different. In ReferCOCO, the queries are short phrases, whilst in ReferCOCO+ and ReferCOCOg, their queries are normally declarative sentences, short and long. The data statistics are recorded in Table 3. We have the same data splits as [22,45,17] that we split ReferCOCO and ReferCOCO+ into 40,000 training, 5000 validation, and 5000 testing samples, where the testing set are further split into "TestA" and "TestB". More precisely, images containing multiple people are put into "TestA" while images containing other objects are in "TestB". ReferCOCOg is split into 44,822 training and 5000 validation samples.

In this paper, we use VGG-16 [4] as our backbone to extract image features from the FC-7 layer and the detected bounding boxes are from SSD [41]. The VGG-16 is pretrained on ImageNet [45] and fine-tuned on MS-COCO. For our language model, we rank the frequency of words in dataset, and the top 6000 words are selected while the others are thought to be word "⟨unk⟩". We obtain the Penn Treebank tags by the NLTK toolkit [58]. The Penn Treebank tagset has 36 different syntactic tags (omit punctuation and see Table 4), and we add two types of tag for position ("top", "bottom", "left", "right", "front" and "back") and "⟨unk⟩" respectively. we set the learning word embedding vector with the dimension of 512 and the hidden and cell state of all LSTMs have the dimension of 256. We take momentum stochastic gradient descent as the optimizer, where the momentum coefficient is set to 0.9. We train the model with the learning rate 0.1 and total epoch 30, and the learning rate will decrease to 1/10 after 10 epochs. The coefficients of graph consistency and textual consistency losses α and β are both set to 0.1. The evaluation metric is the accuracy. For detected bboxes, the correct grounding means the detected bbox has more than 50% IOU with the gt bbox. For gt bboxes, the accuracy means how accurate the query select the correct object.

4.2. Visual grounding results

In Table 1, we compare the results of our model with previous methods with ground-truth (gt) bboxes and detected ones. All the compared methods use VGG-16 to extract visual features, which is the same as our setting. First, with gt bboxes, the proposed method almost outperforms other state-of-the-arts from short phrase to full sentence. On ReferCOCO+, our model achieves accuracy 70.27% and 66.09% on val and testB set, while the previous best results are 65.56% and 62.90% (A-ATT-r4 and VC). On ReferCOCOg, the results on split val of the proposed method can reach to 77.45%, and the previous best is 73.18%. We think the reason is the sentences in ReferCOCO+ and ReferCOCOg always have longer descriptions of reference objects, which results in failure in previous methods. The vtGraphNet makes image into a text-related scene graph, which significantly strengthens the relationships among objects. On ReferCOCO, in which the queries are all short phrases, our model still outperforms on split val (83.10%) and testA (83.02%). We consider the reason is that few relationships are short phrases, which makes most words in ReferCOCO



(a) Cross graph representation (b) Graph consistency loss

Fig. 3. Cross graph representation is computed by first weighting attribute word features and relationship word features, then fusing them and the max-pooling of auxiliary word features. Graph consistency loss is computed by comparing the vertices corresponding attributes and relationships.

Table 1
Comparisons on ReferCOCO, ReferCOCO+ and ReferCOCOg. The bold number denotes the best across all compared methods.

Methods	Bboxes	ReferCOCO			ReferCOCO+			ReferCOCOg		
		Val	TestA	TestB	Val	TestA	TestB	Val*	Val	Test
Baseline [21]	gt	–	63.15	64.21	–	48.73	42.13	55.16	–	–
visdif [22]	gt	–	67.57	71.19	–	52.44	47.51	59.25	–	–
MMI [21]	gt	–	71.72	71.09	–	58.42	51.23	62.14	–	–
visdif+MMI [22]	gt	–	73.98	76.59	–	59.17	55.62	64.02	–	–
Luo et al. [23]	gt	–	74.14	71.46	–	59.87	54.35	63.39	–	–
Luo et al. (w2v)[23]	gt	–	74.04	73.43	–	60.26	55.03	65.36	–	–
Neg Bag [13]	gt	76.90	75.60	78.00	–	–	–	68.40	–	–
speaker+listener+MMI [44]	gt	79.22	77.78	79.90	61.72	64.41	58.62	71.77	–	–
speaker+reinforcer+MMI [44]	gt	78.38	77.13	79.53	61.32	63.99	58.25	67.06	–	–
speaker+listener+reinforcer+MMI [44]	gt	79.56	78.95	80.22	62.26	64.60	59.62	72.63	–	–
VC [54]	gt	–	78.98	**82.39**	–	62.56	62.90	73.98	–	–
A-ATT-r4 [12]	gt	81.27	81.17	80.01	65.56	68.76	60.63	73.18	–	–
MAttN [17]	gt	80.94	79.99	82.30	63.07	65.04	61.77	73.08	73.04	72.79
vtGraphNet	gt	**83.10**	**83.02**	82.17	**70.27**	**71.43**	**66.09**	**77.45**	**75.02**	**76.67**
MMI [21]	det	–	64.90	58.51	–	54.03	42.81	45.85	–	–
Neg Bag [13]	det	–	58.60	56.40	–	–	–	39.50	–	–
Luo et al. [23]	det	–	68.11	54.65	–	56.61	43.74	47.60	–	–
Luo et al. (w2v)[23]	det	–	67.94	55.18	–	57.05	43.33	49.07	–	–
Attr [55]	det	–	72.08	57.29	–	57.97	46.20	52.35	–	–
speaker+listener+MMI [44]	det	–	72.95	63.10	–	60.23	48.11	58.57	–	–
speaker+reinforcer+MMI [44]	det	–	72.34	63.24	–	59.36	48.72	58.70	–	–
speaker+listener+reinforcer+MMI [44]	det	–	72.88	63.43	–	60.43	48.74	59.51	–	–
VC [54]	det	–	**73.33**	**67.44**	–	58.40	**53.18**	62.30	–	–
vtGraphNet	det	–	72.47	63.26	–	**63.62**	51.18	**65.09**	–	–

are attributes, which is still helpful to visual grounding (analyze in Section 4.6). Then, with SSD detected bboxes, we have better performances on ReferCOCO+ and ReferCOCOg, and comparable results on ReferCOCO. Specifically, we obtain 63.62% on split testA of ReferCOCO+, and obtain 65.09% on ReferCOCOg, both of which are with more than 3% improvement.

4.3. Ablation study

We then evaluate the contributions of some key components in the proposed framework with gt bboxes, and the results are shown in Table 2. The baseline (vtGraphNet:co_att) is the model in which we only consider the co-attention of query and all objects, and no graph would be constructed. We first evaluate that using ARU tagging to generate a scene graph without extra loss (vtGraphNet:ARU +co_att). The results show that ARU tagging dramatically improves the performance because the constructed VT scene graph considers relationships among objects and aggregates them into a graph structure. Then, we evaluate the model with the proposed graph consistency loss (vtGraphNet:ARU+gph+co_att). On three datasets, most results increase in contrast to omitting this loss. The reason is that the graph consistency loss makes the nodes and relationships associate, and reduces noisy information. After that, we evaluate ARU tagging with the textual consistency loss. Obviously, the textual consistency loss can also improve the ARU tagging performance on all three datasets. When ARU tagging adds both graph consistency and textual consistency loss, the performance further increases on most splits of three datasets.

4.4. Visual-textual scene graph generation

We show some examples in Fig. 4 including VT scene graph and predictions. We can see the proposed vtGraphNet can transform an image to a VT scene graph. Regions are correlated via different relationship words, and some of them have their own attributes. Such as the top right example, "4" has the attributes "man, gun", and "1,2" have the relationship "holding". But the constructed graph is still not very "dense", which means that the VT scene graph does not describe the overall scene but only contain several important parts in image based on query. Another phenomenon is the attribute and relationship are not strictly and syntactically separate. For example, the word "front" is recognized as an attribute in the top left example. From the first column in Fig. 4, we show some examples that same image with different queries. Easy to know, even for the same image, different queries will generate different VT scene graphs, which illustrates the difference of our VT scene graph with common scene graph is VT scene graph needs to jointly consider visual and textual information.

4.5. ARU Tagging

Penn Treebank to ARU tags. In the proposed ARU tagging, we adopt the weakly-supervised training strategy to obtain three types of tag (Attribute, Relationship and aUxiliary) with the information from query, image and Penn Treebank (PT) tags. We also try to directly obtain ARU tags by mapping PT tags by the rule in Table 4. We can see the results in Table 5 that the grounding accu-

Table 2
Ablation studies for different key components of the proposed method. The bold number denotes the best across all compared methods.

Model	ReferCOCO			ReferCOCO+			ReferCOCOg
	Val	TestA	TestB	Val	TestA	TestB	Val*
vtGraphNet:co_att (baseline)	80.93	79.75	79.53	65.45	66.71	61.63	71.80
vtGraphNet:ARU+co_att	82.59	82.10	82.16	70.13	70.65	66.05	76.46
vtGraphNet:ARU+gph+co_att	82.85	82.12	**82.32**	**70.52**	71.19	**66.44**	76.60
vtGraphNet:ARU+rec+co_att	82.97	82.18	82.03	70.43	70.94	66.16	77.36
vtGraphNet:ARU+rec+gph+co_att	**83.10**	**83.02**	82.17	70.27	**71.43**	66.09	**77.45**

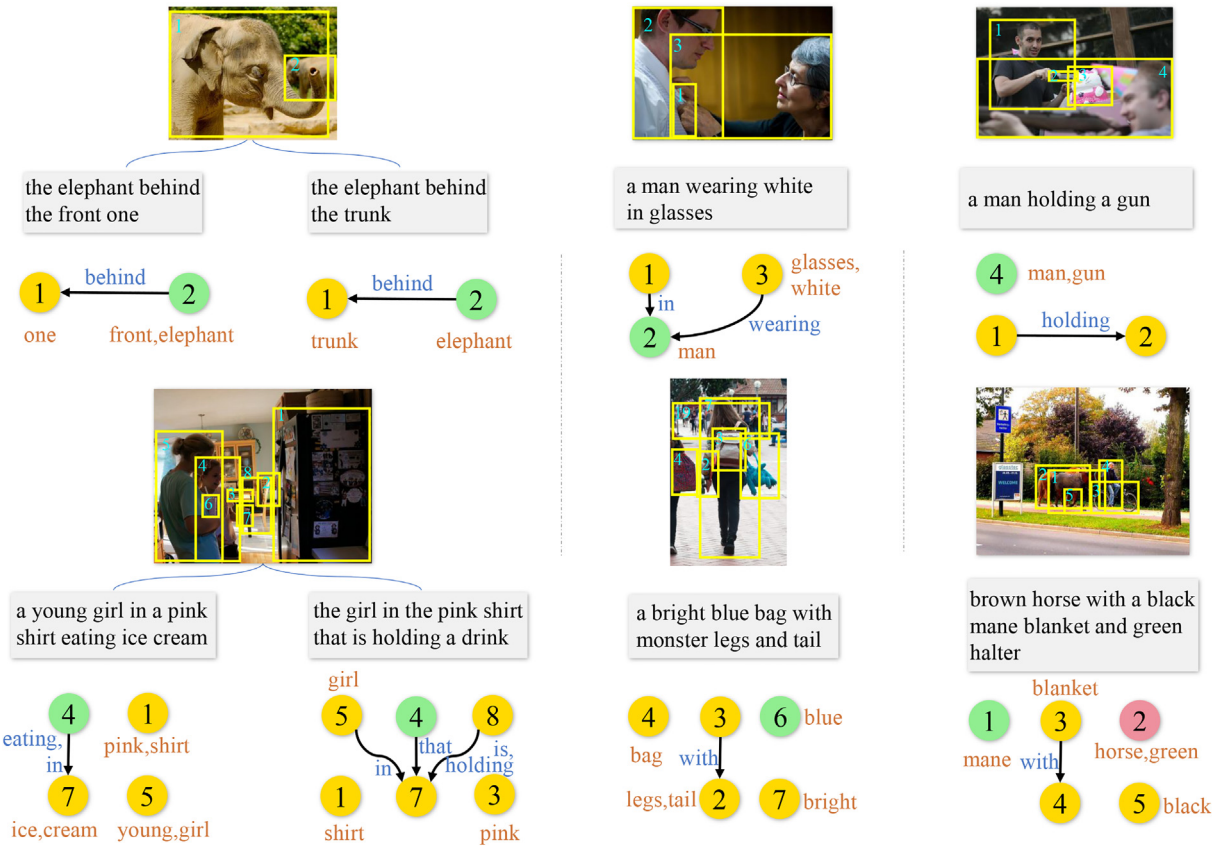


Fig. 4. Examples of VT scene graph for VG task. The green nodes mean the correct prediction, while the red means failure. The first column show when same image is with different queries. The last two columns are some correct predictions and a failure case. Results show the proposed vtGraphNet can obtain VT scene graph, with relationship and attribute words are assigned to nodes and edges. This process makes an image a structured topology, and different queries will obtain different VT graph, which is helpful to visual grounding. (Omit drawing some bboxes, and best view in color.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Dataset statistic.

Dataset	#images	#queries	#object	#min_obj	#max_obj
RefCOCO	19,994	142,209	50,000	2	75
RefCOCO+	19,992	141,564	49,856	2	75
RefCOCOg	26,711	85,474	54,822	2	69

Table 4
English Penn Treebank tagset to ARU tagset.

ARU tags	English Penn Treebank tags [49]
Attribute	JJ, JJR, JJS, CD, NN, NNS, NNP, NNPS, PRP, PRP$, RB, RBR, RBS
Relationship	VB, VBD, VBG, VBN, VBP, VBZ, IN, "top", "bottom", "left", "right", "front", "back"
a**U**xiliary	CC, DT, EX, FW, LS, MD, PDT, POS, RP, SYM, TO, UH, WDT, WP, WP$, WRB, "⟨unk⟩"

Table 5
Comparisons of different ARU. The bold number denotes the best across all compared methods.

Types	RefCOCOg(Acc.)
ARU (A+R+U)	**77.45**
ARU (Penn Treebank mapping)	77.32
Fine-grained ARU (A1+A2+R1+R2+U)	76.62

racy with this mapping even slightly decreases. This may be partial because the data-driven vtGraphNet trained directly on grounding tasks works satisfactorily, and the hard tagging that independent from image may not be proper.

Fine-grained ARU tags. The tagset of our method is in coarse level (Only Attribute, relationship and Auxiliary), which we think is enough in weakly-supervised situation. Nevertheless, we try that if the tagset of ARU tagging was fine-grained. In practice, we split attributes (A) into nouns (A1) and adjectives (A2) and split relationships into spatial relations (R1) and relationship predicates (R2), as well as auxiliary (U). As shown in Table 5, the fine-grained tags do not bring better performance, which shows that a coarse level tag set is enough for vtGraphNet.

4.6. Complexity of visual grounding

Simple visual grounding. In some cases, the queries are simple phrases without any relationship information, such as "a child". We show that, in these situations, the constructed VT scene graph may reduce to simpler forms but can still support accurate visual grounding (ReferCOCO result in Table 1). To validate the effectiveness of the vtGraphNet in simple tasks, we show some examples in these situations (Fig. 5(a)). In simple situation, the graph will

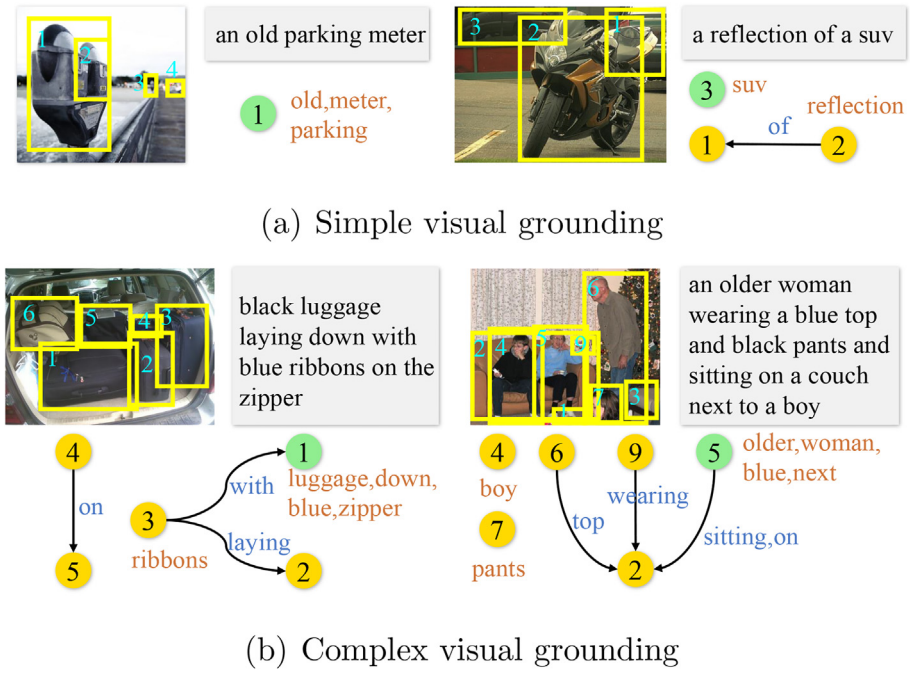


Fig. 5. VT scene graph examples of simple and complex VG. In simple case (a), the VT scene graph may degrade to simpler form that only contains one to two nodes. In contrast, complex VG may contain more complex structure. (Omit drawing some bboxes.).

Table 6
Comparison between a state-of-the-art method and vtGraphNet on complex ReferCOCOg dataset (query_len $> n$ and obj_num $> n$) where $n \in \{0, 5, 10, 15\}$. The bold number denotes the best across all compared methods.

Method	Complex level n	ReferCOCOg (val*)
A-ATT	0	73.18
vtGrapNet		**77.45**
A-ATT	5	70.80
vtGrapNet		**71.61**
A-ATT	10	40.12
vtGrapNet		**42.79**
A-ATT	15	20.31
vtGrapNet		**21.88**

reduce to a simpler node set without any relationship among each other. For example, in the first image, the query is "an old parking meter", which has no relationship word. The graph reduces to only a node "1" with the attribute "old, parking, meter". This also reduces the searching space, and makes the grounding model focus on the region proposal "1" by considering the attribute "old, parking, meter".

Complex visual grounding. We also evaluate complex visual grounding. The complexity appears as two factors, *i.e.*, the length of query and number of region proposals. Thus, we construct some complex subset (query_len $> n$ and obj_num $> n$, about half of original dataset) of the dataset ReferCOCOg and train from scratch where $n \in 0$(vanilla complexity), 5, 10, 15. We compare our results with A-ATT [12]'s in Table 6. Obviously, the accuracy values are both declined, but our result still outperforms [12]. The reason is that our vtGraphNet is able to build all visual-textual semantic relationships and aggregate them into a VT scene graph, which is helpful to visual-language tasks especially complex visual grounding. We also visualize the VT scene graph of some complex visual grounding scene in Fig. 5(b). Compared to simpler VG, complex situation could obtain more complex VT scene graph, even find some latent relationships in queries.

5. Conclusion

In this paper, we have proposed a novel framework, vtGraphNet, to produce visual-textual (VT) scene graph for complex visual grounding. Unlike previous works, our approach aims to construct an overall bi-modal understanding of image and query, through which the target object can be more accurately located. We first construct an ARU Tagging model to sequentially classify every query word to attribute, relationship or auxiliary. Then, we propose an attribute-assigning model and a relationship-referring model to associate related region proposals to the words tagged as attribute and relationship, respectively. To obtain compact and feasible VT scene graph, we propose a simple yet effective graph consistency loss function to constrain the association of attribute and relationship region proposals. Finally, discriminative region features are derived from the graph representation and help to locate the target object via classification. Extensive evaluations on benchmark datasets validate the effectiveness of the proposed approach in handling complex visual grounding problems. In the future, we want to explore integrating more sophisticated language model and semantics-driven similarity measure [59,60] in the proposed vtGraphNet. We are also interested in applying our model in real-world application scenarios, such as fast object localization and retrieval via in situ speech recognition.

CRediT authorship contribution statement

Fan Lyu: Conceptualization, Methodology, Software, Writing - original draft, Visualization, Investigation. **Wei Feng:** Supervision, Writing - review & editing. **Song Wang:** Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Nos. 61671325, 61672376 and U1803264). The authors would like to thank constructive and valuable suggestions for this paper from the experienced reviewers and AE.

References

[1] T.B. Sheridan, W.R. Ferrell, Man-Machine Systems; Information, Control, and Decision Models of Human Performance, The MIT press, 1974.
[2] E.A. Sisbot, R. Ros, R. Alami, Situation assessment for human-robot interactive object manipulation, in: Proceeding of the IEEE International Conference on Robot and Human Interactive Communication, IEEE, 2011, pp. 15–20.
[3] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105..
[4] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceeding of the Conference on International Conference on Learning Representations, 2014.
[5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[6] F. Lyu, Q. Wu, F. Hu, Q. Wu, M. Tan, Attend and imagine: Multi-label image classification with visual attention and recurrent neural networks, IEEE Transactions on Multimedia 21 (8) (2019) 1971–1981.
[7] F. Lyu, L. Li, S.S. Victor, Q. Fu, F. Hu, Multi-label image classification via coarse-to-fine attention, Chinese Journal of Electronics 28 (6) (2019) 1118–1126.
[8] F. Lyu, F. Hu, V.S. Sheng, Z. Wu, Q. Fu, B. Fu, Coarse to fine: Multi-label image classification with global/local attention, in: Proceeding of the IEEE conference on International Smart Cities Conference, IEEE, 2018, pp. 1–7.
[9] Z. Ye, F. Lyu, J. Ren, Y. Sun, Q. Fu, F. Hu, Dau-gan: Unsupervised object transfiguration via deep attention unit, in: Proceedings of the International Conference on Brain Inspired Cognitive Systems, Springer, Cham, 2018, pp. 120–129.
[10] Z. Ye, F. Lyu, L. Li, Q. Fu, J. Ren, F. Hu, Sr-gan: Semantic rectifying generative adversarial network for zero-shot learning, in: Proceedings of the IEEE International Conference on Multimedia and Expo, 2019, pp. 85–90.
[11] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, T. Darrell, Natural language object retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4555–4564..
[12] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, M. Tan, Visual grounding via accumulated attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7746–7755.
[13] V.K. Nagaraja, V.I. Morariu, L.S. Davis, Modeling context between objects for referring expression understanding, in: European Conference on Computer Vision, Springer, 2016, pp. 792–807.
[14] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, B. Schiele, Grounding of textual phrases in images by reconstruction, in: European Conference on Computer Vision, Springer, 2016, pp. 817–834.
[15] F. Zhao, J. Li, J. Zhao, J. Feng, Weakly supervised phrase localization with multi-scale anchored transformer network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5696–5705.
[16] V. Cirik, T. Berg-Kirkpatrick, L.-P. Morency, Using syntax to ground referring expressions in natural images, in: AAAI Conference on Artificial Intelligence, 2018.
[17] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, T.L. Berg, Mattnet: Modular attention network for referring expression comprehension, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1307–1315.
[18] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, K. Saenko, Modeling relationships in referential expressions with compositional modular networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1115–1124.
[19] J. Krishnamurthy, T. Kollar, Jointly learning to parse and perceive: Connecting natural language to the physical world, Transactions of the Association for Computational Linguistics 1 (2013) 193–206.
[20] C. Kong, D. Lin, M. Bansal, R. Urtasun, S. Fidler, What are you talking about? text-to-image coreference, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3558–3565.
[21] J. Mao, J. Huang, A. Toshev, O. Camburu, A.L. Yuille, K. Murphy, Generation and comprehension of unambiguous object descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 11–20.
[22] L. Yu, P. Poirson, S. Yang, A.C. Berg, T.L. Berg, Modeling context in referring expressions, in: European Conference on Computer Vision, vol. 9906, Springer, 2016, pp. 69–85..
[23] R. Luo, G. Shakhnarovich, Comprehension-guided referring expressions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7102–7111.
[24] K. Chen, J. Gao, R. Nevatia, Knowledge aided consistency for weakly supervised phrase grounding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4042–4050.
[25] F. Xiao, L. Sigal, Y. Jae Lee, Weakly-supervised visual grounding of phrases with linguistic structures, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5945–5954.
[26] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2008) 61–80.
[27] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral networks and locally connected networks on graphs, in: Proceeding of the Conference on International Conference on Learning Representations, 2014..
[28] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2) (2004) 167–181.
[29] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in Neural Information Processing Systems, 2007, pp. 545–552..
[30] W. Feng, J. Jia, Z.-Q. Liu, Self-validated labeling of markov random fields for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (10) (2010) 1871–1887.
[31] C. Rother, P. Kohli, W. Feng, J. Jia, Minimizing sparse higher order energy functions of discrete variables, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1382–1389.
[32] W. Feng, Z.-Q. Liu, Region-level image authentication using bayesian structural content abstraction, IEEE Transactions on Image Processing 17 (12) (2008) 2413–2424.
[33] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, L. Fei-Fei, Image retrieval using scene graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3668–3678.
[34] J. Johnson, A. Gupta, L. Fei-Fei, Image generation from scene graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1219–1228.
[35] D. Teney, L. Liu, A. van Den Hengel, Graph-structured representations for visual question answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1–9.
[36] D. Lin, S. Fidler, C. Kong, R. Urtasun, Visual semantic search: Retrieving videos via complex textual queries, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2657–2664.
[37] D. Xu, Y. Zhu, C.B. Choy, L. Fei-Fei, Scene graph generation by iterative message passing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5410–5419.
[38] Y. Li, W. Ouyang, B. Zhou, K. Wang, X. Wang, Scene graph generation from objects, phrases and region captions, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1261–1270.
[39] R. Zellers, M. Yatskar, S. Thomson, Y. Choi, Neural motifs: scene graph parsing with global context, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5831–5840.
[40] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99..
[41] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[42] C.L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: European Conference on Computer Vision, Springer, 2014, pp. 391–405.
[43] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11) (2012) 2189–2202.
[44] J. Wang, L. Specia, Phrase localization without paired training examples, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 4663–4672.
[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[46] T. Brants, Tnt: a statistical part-of-speech tagger, in: Proceeding of the Conference on Applied Natural Language Processing, Association for Computational Linguistics, 2000, pp. 224–231..
[47] A. Ratnaparkhi, A maximum entropy model for part-of-speech tagging, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1996.
[48] Y. Zhang, S. Clark, A fast decoder for joint word segmentation and pos-tagging using a single discriminative model, in: Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2010, pp. 843–852..
[49] M.P. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank, Computational Linguistics..
[50] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (11) (1997) 2673–2681.
[51] B. Dai, Y. Zhang, D. Lin, Detecting visual relationships with deep relational networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3076–3086.

[52] R. Krishna, I. Chami, M. Bernstein, L. Fei-Fei, Referring relationships, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6867–6876.
[53] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, in: Advances in Neural Information Processing Systems, 2016, pp. 289–297..
[54] H. Zhang, Y. Niu, S.-F. Chang, Grounding referring expressions in images by variational context, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4158–4166.
[55] J. Liu, L. Wang, M.-H. Yang, Referring expression generation and comprehension via attributes, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4856–4864.
[56] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[57] S. Kazemzadeh, V. Ordonez, M. Matten, T. Berg, Referitgame: Referring to objects in photographs of natural scenes, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014, pp. 787–798.
[58] E. Loper, S. Bird, Nltk: The Natural Language Toolkit, Association for Computational Linguistics, 2004.
[59] W. Feng, X. Nie, Y. Zhang, Z.-Q. Liu, J. Dang, Story co-segmentation of chinese broadcast news using weakly-supervised semantic similarity, Neurocomputing 355 (2019) 121–133.
[60] W. Feng, X. Nie, Y. Zhang, L. Xie, J. Dang, Unsupervised measure of chinese lexical semantic similarity using correlated graph model for news story segmentation, Neurocomputing 318 (2018) 236–247.

Fan Lyu received the BS and MS degree in Electronic & Information Engineering, Suzhou University of Science and Technology, China, in 2015 and 2018. He is working toward the PhD degree in the College of Intelligence and Computing, Tianjin University, China. His research interests include visual grounding, multi-label classification, and image captioning.

Wei Feng received the BS and MPhil degrees in computer science from Northwestern Polytechnical University, China, in 2000 and 2003, respectively, and the PhD degree in computer science from City University of Hong Kong in 2008. From 2008 to 2010, he was a research fellow at the Chinese University of Hong Kong and City University of Hong Kong. He is now a full professor in the School of Computer Science and Technology, Tianjin University, China. His major research interests are active robotic vision and visual intelligence, specifically including active camera relocalization and lighting recurrence, general Markov Random Fields modeling, energy minimization, active 3D scene perception, SLAM, and generic pattern recognition. Recently, he focuses on solving preventive conservation problems of cultural heritages via computer vision and machine learning. He is the Associate Editor of Neurocomputing and Journal of Ambient Intelligence and Humanized Computing.

Song Wang received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2002. He was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, image processing, and machine learning. Dr. Wang is a senior member of IEEE and a member of the IEEE Computer Society. He is currently serving as the Publicity/Web Portal Chair for the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society and as an Associate Editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Recognition Letters, and Electronics Letters.