Visual-Semantic Graph Matching for Visual Grounding

Chenchen Jing
chenchen.jing@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Yuwei Wu*
wuyuwei@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Mingtao Pei
peimt@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Yao Hu
yaoohu@alibaba-inc.com
Alibaba Youku Cognitive and Intelligent Lab
Beijing, China

Yunde Jia
jiayunde@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Qi Wu
qi.wu01@adelaide.edu.au
University of Adelaide
Adelaide, Australia

ABSTRACT

Visual Grounding is the task of associating entities in a natural language sentence with objects in an image. In this paper, we formulate visual grounding as a graph matching problem to find node correspondences between a visual scene graph and a language scene graph. These two graphs are heterogeneous, representing structure layouts of the sentence and image, respectively. We learn unified contextual node representations of the two graphs by using a cross-modal graph convolutional network to reduce their discrepancy. The graph matching is thus relaxed as a linear assignment problem because the learned node representations characterize both node information and structure information. A permutation loss and a semantic cycle-consistency loss are further introduced to solve the linear assignment problem with or without ground-truth correspondences. Experimental results on two visual grounding tasks, *i.e.*, referring expression comprehension and phrase localization, demonstrate the effectiveness of our method.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Natural language processing**.

KEYWORDS

Visual Grounding, Graph Matching, Visual Scene Graph, Language Scene Graph

ACM Reference Format:
Chenchen Jing, Yuwei Wu, Mingtao Pei, Yao Hu, Yunde Jia, and Qi Wu. 2020. Visual-Semantic Graph Matching for Visual Grounding. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413902

*corresponding author

MM '20, October 12–16, 2020, Seattle, WA, USA.

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413902

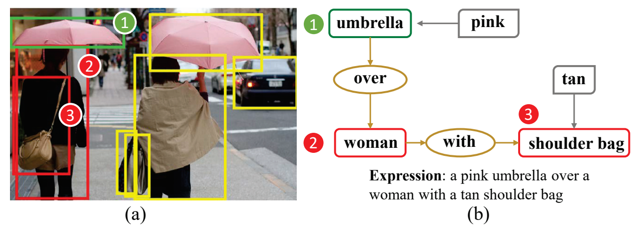

Figure 1: Illustration of graph matching between a visual scene graph (a) and a language scene graph (b) for referring expression comprehension. Green boxes represent a referent object in the visual graph and a referent entity in the language graph. Red boxes represent context objects and context entities. Yellow boxes represent objects irrelevant to the language expression.

1 INTRODUCTION

Visual grounding is to associate entities in a natural language sentence with objects in an image. It is a fundamental building block for vision-language tasks such as visual captioning [8, 38], visual question answering [3, 14, 44], and vision-language navigation [2, 43]. Recently, visual grounding tasks such as referring expression comprehension and phrase localization have gained considerable attention. Referring expression comprehension is to ground a referring expression to an object described by the expression, and phrase localization is to ground all noun phrases in an image description to objects in the corresponding image. Both tasks are challenging because establishing such fine-grained correspondences requires comprehensively understanding textual semantics and visual concepts, modeling similarities between the semantics and concepts, and finding their correspondences, *i.e.*, one-to-one mapping.

Most existing methods [6, 10, 24, 32, 34, 41] for visual grounding focus more on modeling object-phrase similarities than finding their correspondences in a global manner, which may result in matching ambiguity. In this paper, we formulate visual grounding as a graph matching problem to find node correspondences between a visual scene graph and a language scene graph. We present an end-to-end visual-semantic graph matching method that jointly models similarities between objects and phrases and finds their correspondences to achieve accurate visual grounding.

The scene graph is a prevailing structure to represent the contextual layouts of both images and sentences, and has been proven to be effective in various vision-language tasks [13, 24, 47]. We observe that referring expression comprehension and phrase localization can be naturally cast as a graph matching problem between the visual scene graph and language scene graph. Graph matching is to find node correspondences between two graphs to maximize the corresponding node and edge's affinity [42, 53]. Figure 1 shows the graph matching by taking referring expression comprehension as an example. By solving the graph matching of the language scene graph and the visual scene graph, textual semantics in the sentence and visual concepts in the image can be fully aligned for accurate visual grounding. To this end, two challenges must be considered: (1) The nodes and edges of the two graphs lie in heterogeneous spaces and thus are not ready for matching due to the gap between the language domain and the vision domain. (2) The graph matching is generally regarded as a quadratic assignment programming problem, an NP-hard combinatorial optimization problem.

To address these challenges, we propose to jointly learn node representations of the two graphs and find their correspondences for visual grounding. We build a novel *cross-modal graph convolutional network* to learn unified node representations, which characterize both node information and implicit structure information, for reducing the discrepancy of the two heterogeneous graphs. Considering that the node representations are enriched with structure information, the graph matching is relaxed as a linear assignment problem. For phrase localization, we introduce a *permutation loss* to solve the linear assignment problem. For referring expression comprehension, the ground-truth node correspondences for graph matching are unavailable, because the context objects mentioned in sentences to determine the referred object are usually unlabeled. Apart from a standard referent object matching loss, we further introduce a *semantic cycle-consistency loss*, which encourages one-to-one mapping between the two graphs in a self-supervised manner, to solve the linear assignment problem without ground-truth correspondences for referring expression comprehension.

We evaluate the proposed method on both phrase localization and referring expression comprehension. Experimental results show the effectiveness of our method. The contributions of this paper are summarized as follows:

(1) We formulate visual grounding as a graph matching problem and present a visual-semantic graph matching method to fully align visual concepts and textually semantics for accurate visual grounding.
(2) We propose a novel cross-modal graph convolutional network to learn unified context-aware node representations to facilitate graph matching, and a semantic cycle-consistency loss to solve the graph matching without ground-truth correspondences.

2 RELATED WORK

2.1 Referring expression comprehension

Referring expression comprehension is to localize the object described by a language expression in an image. Typically, this task is formulated as an object retrieval task, where the object with the highest similarity with the language expression from a set of object proposals is identified as the referent object. Early methods [12, 27, 28, 50] adopt a CNN/LSTM framework to find the region that maximizes the likelihood of the language expression. The major difference among these methods is how they model the visual context. For example, Hu *et al.* [12] used the whole image as context. Yu *et al.* [50] adopted the visual difference between objects as context. Another line of works [11, 26, 34, 39, 49] project the objects and the language expressions into a common feature space to measure the similarity. Luo *et al.* [26] used the softmax loss as the matching loss function, while Mao *et al.* [27] exploited the max-margin loss. Specifically, Yu *et al.* [49] propose to exploit different types of information in expressions including subject, location, and relationship, and comprehensively measure the similarity between each object and the expression.

Recently, various methods that resort to graphs to represent the structure information of images or expressions to achieve relational reasoning have been proposed. Visual-graph-based methods [10, 41, 45, 46] represent images as graphs, learn context-aware node representations via graph networks, and measure the similarity between the nodes and the language expressions to determine referent objects. Language-graph-based methods [9, 21, 22] parse expressions to a graph structure to capture the semantics in expressions and perform reasoning over the structure. Our work differs from them in that we perform joint reasoning over both the language graph and visual graph for more comprehensive context modeling to fully align visual concepts and textually semantics.

2.2 Phrase localization

Phrase localization [40] is to ground phrases in an image description to corresponding objects in the image. Pioneering works [31, 32, 34, 52] for visual grounding usually independently ground each phrase in the description and ignore visual and textually contextual information. Rohrbach *et al.* [34] presented an attention-based method to attend to relevant object proposals for a given phrase and designed a loss to reconstruct the phrase. Yu *et al.* [52] focused on the proposal generation which aims to generate diverse and discriminative object proposals for phrase localization.

Recent methods take into account the contextual information to achieve accurate visual grounding. Dongan *et al.* [6] adopted chain-structured Long Short-Term Memory networks (LSTMs) [35] to encode the contextual information in the language and image domains, respectively. Liu and Hockenmaier [23] used chain-structured conditional random fields to model dependencies among regions for adjacent phrases. Bajaj *et al.* [4] exploited graphs to characterize the contextual information and fused the two graphs to capture cross-modal relationships. The aforementioned methods focus on modeling similarities between objects and phrases but ignore finding the assignment of objects and phrases and thus may lead to grounding ambiguity. Only a few methods [15, 24, 40] take into account the one-to-one mapping constraint, that is, while the contextual information in both the textual and visual domain are fully modeled each object corresponds to one entity and vice versa. However, they either are unable to model multi-order relationships [15, 15, 40], which may lead to the assumption of one-to-one mapping constraint invalid, or only find the assignment via post-processing [24]. Thus their solutions are sub-optimal. In this paper, we formulate the

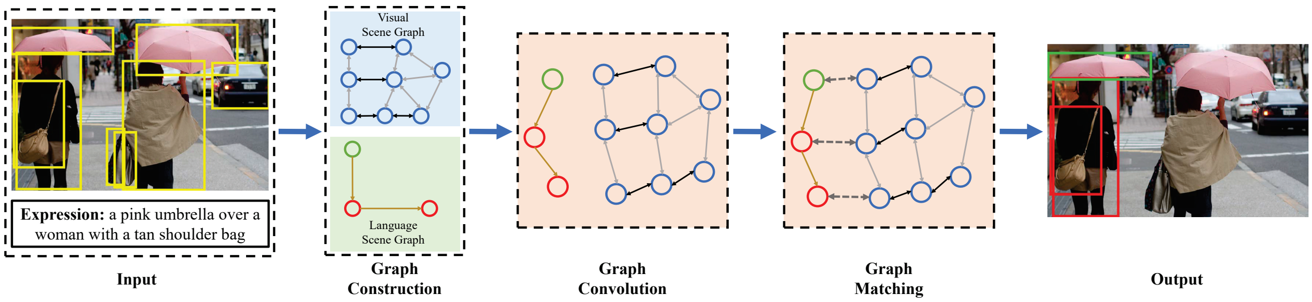

Figure 2: Diagram of our method. It constructs two graphs from an image and a sentence, respectively, uses a cross-modal graph convolutional network to learn unified contextual node representations, and solves the graph matching to find node correspondences. Golden arrows denote the edges in the language graph. Gray arrows and black arrows denote intra-class and inter-class edges in the visual graph, respectively. We do not show all the edges in the visual graph for convenience.

visual grounding as a graph matching problem, aiming to find node correspondences between two graphs to maximize the corresponding node and edge's affinity [42, 53], and solve graph matching in an end-to-end manner for better compatibility.

3 METHOD

In this section, we formally define visual grounding as a graph matching problem and describe our method shown in Figure 2. The proposed method constructs two graphs from an image and a sentence, respectively, uses a cross-modal graph convolutional network to learn unified contextual node representations, and solves the graph matching to find node correspondences.

3.1 Formulation

Referring expression comprehension and phrase localization are two visual grounding tasks. Referring expression comprehension aims to localize the object described by a referring expression L in the image I represented by a set of objects $O = \{o_i\}_{i=1}^{N}$. N is the number of objects. Phrase localization aims to localize all objects mentioned in an image description L in an image I. For convenience, here we use the same notations L and I to represent the input sentence (*i.e.* the description/expression) and the image, respectively.

We formulate each grounding task as graph matching to achieve the alignments between textual semantics and visual concepts. Specifically, we construct a visual scene graph $G^I = \{V^I, E^I\}$ and a language scene graph $G^L = \{V^L, E^L\}$ to represent the image I and the sentence L, respectively. In the visual scene graph, $V^I = \{v_i^I\}_{i=1}^{N}$ is a set of nodes corresponding to the objects in the image and $E^I = \{e_{ij}^I\}_{i,j=1}^{N}$ denotes the relationships among objects. Similarly, $V^L = \{v_i^L\}_{i=1}^{M}$ and $E^L = \{e_{ij}^L\}_{i,j=1}^{M}$ represent the objects and relationships mentioned in the sentence. Usually, we have $M \leq N$ as all entities mentioned in the sentence should appear in the image.

To accurately associate the objects and entities, both the unary similarity and the pairwise similarity of the two graphs should be taken into account. Thus the graph matching problem is naturally a quadratic assignment programming (QAP) problem [25],

$$J(\boldsymbol{A}) = vec(\boldsymbol{A})^\top \boldsymbol{K} vec(\boldsymbol{A}), \quad s.t.\ \boldsymbol{A}\mathbf{1} = \mathbf{1}, \boldsymbol{A}^\top \mathbf{1} \leq \mathbf{1} \tag{1}$$

where $\boldsymbol{A} \in \{0,1\}^{M\times N}$ is an assignment matrix indicating the node correspondences such that $\boldsymbol{A}_{ij} = 1$ if v_i^L and v_j^I are matched, and 0 otherwise. $\boldsymbol{K} \in \mathbb{R}^{MN\times MN}$ is the affinity matrix whose diagonal elements and off-diagonal ones encode the node-to-node and edge-to-edge affinity between two graphs, respectively. As illustrated in the constraint in Eq. (1), each node in the language scene graph should be assigned a corresponding node in the visual graph.

The QAP problem is a well-known NP-hard problem. Traditional graph matching methods [19, 55] usually relax the binary constraint and solve the QAP problem approximately with a fixed affinity matrix. These methods are inapplicable to our task because both the nodes and edges of the language graph and those of the visual graph lie in heterogeneous spaces. Thus we introduce a cross-modal graph convolutional network to learn unified contextual node representations, $\boldsymbol{U}^I = \{\boldsymbol{u}_i^I\}_{i=1}^{N}$ and $\boldsymbol{U}^L = \{\boldsymbol{u}_i^L\}_{i=1}^{M}$, that characterize both node information and structure information, for reducing the discrepancy of the two graphs. Considering that the node representations characterize both node information and structure information, the graph matching is thus relaxed as a linear assignment problem,

$$J(\boldsymbol{A}) = \boldsymbol{C}^T \boldsymbol{A}, \quad s.t.\ \boldsymbol{A}\mathbf{1} = \mathbf{1}, \boldsymbol{A}^\top \mathbf{1} \leq \mathbf{1} \tag{2}$$

where $\boldsymbol{C} \in \mathbb{R}^{M\times N}$ is an assignment cost matrix whose element $\boldsymbol{C}_{ij} = d(\boldsymbol{u}_i^L, \boldsymbol{u}_j^I)$ represents the distance between $\boldsymbol{u}_i^L$ and $\boldsymbol{u}_j^I$. Because $\boldsymbol{C}$ encodes both node similarity and edge similarity, the QAP problem can be relaxed as a linear assignment problem.

For phrase localization, we introduce a permutation loss to directly optimize the assignment matrix $\boldsymbol{A}$ for minimizing Eq. (2) because the ground-truth node correspondences are available. For referring expression comprehension, we introduce a self-supervised semantic cycle-consistency loss to learn appropriate node representations for matching to minimizing Eq. (2). The cycle-consistency loss enforces all nodes in the language graph to satisfy the semantic cycle-consistency constraint to guarantee that the minimal element of each row is also the minimal element of the corresponding column, in the assignment cost matrix. Thus by assigning 1 to the corresponding positions of the cost matrix $\boldsymbol{C}$ and 0 to other positions, we can obtain the assignment matrix $\boldsymbol{A}$ such that Eq. (2) is minimized. In the following sections, we illustrate how we learn the unified node representations and obtain the assignment matrix.

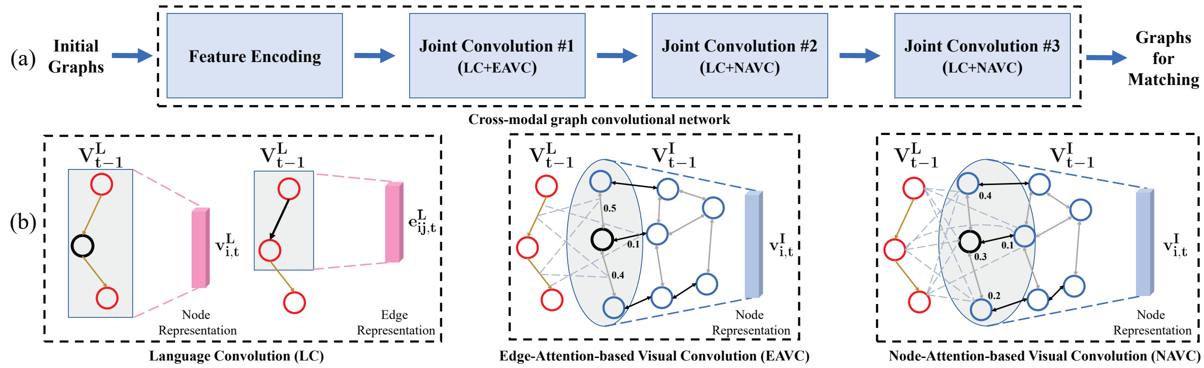

Figure 3: Illustration of the cross-modal graph convolutional network. (a) shows the architecture of the network. It consists of a feature encoding module to project representations of the constructed graphs into a common space, and three joint convolution modules to learn context-aware representations. (b) shows the language convolution operation and two kinds of visual convolution operations. Red nodes and blue nodes show nodes in a language scene graph or a visual scene graph, respectively. The black node/edge represents a node/edge selected for convolution and the corresponding dark area is the receptive field.

3.2 Graph Construction

For the image I, we construct a visual scene graph $G^I = \{V^I, E^I\}$ where each node v_i^I represents a corresponding object o_i and each edge e_{ij}^I denotes the visual relationship between o_i and o_j. For each node, we concatenate two types of features, an appearance feature $\boldsymbol{a}_i$ extracted by a pre-trained CNN and a spatial feature $\boldsymbol{s}_i$ encoding its location information and size, to obtain its representation $\boldsymbol{v}_i^I$. We establish two types of edges, intra-class edges $E^{I,intra}$ and inter-class edges $E^{I,inter}$ according to categories of the linked objects. For each node v_i^I, we rank other objects based on their distances to v_i^I, select the top-5 ranked intra-class objects and top-5 ranked inter-class objects, and establish corresponding edges between these objects and v_i^I. The relative spatial information between two nodes is used as the edge representation $\boldsymbol{e}_{ij}^I$.

For the sentence L, we use a rule-based scene graph parser [36] to construct the graph G^L. The nodes of the language scene graph are nouns with modifiers such as determinants or adjectives, and the edges are relations between nouns. We directly concatenate the modifiers with the nouns to represent the nodes. To obtain the embeddings for the nodes and the edges, we use Bi-LSTM [35] to encode the sentence L and represent each word by concatenating corresponding forward and backward hidden vectors. The embeddings of words for each node or edge are averaged to obtain the node representation $\boldsymbol{v}_i^L$ or edge representation $\boldsymbol{e}_{ij}^L$.

3.3 Cross-modal Graph Convolutional Network

The cross-modal graph convolutional network as shown in Figure 3 (a) consists of a feature encoding module and three cascaded joint convolution modules. The feature encoding module is to project node representations and edge representations of the two graphs into a common space. The cascaded joint convolution modules, where the two graphs are jointly updated via graph convolution, are used to learn context-aware representations.

3.3.1 *Feature Encoding Module.*

The feature encoding module uses specific transformation matrices to project the node representations and edge representations of the two graphs into a common space $\mathbb{R}^d$ as

$$\begin{aligned} \boldsymbol{v}_{i,0}^I &= \boldsymbol{W}_{node}^I \boldsymbol{v}_i^I, \quad \boldsymbol{e}_{ij,0}^I = \boldsymbol{W}_{edge}^I \boldsymbol{e}_{ij}^I, \\ \boldsymbol{v}_{i,0}^L &= \boldsymbol{W}_{node}^L \boldsymbol{v}_i^L, \quad \boldsymbol{e}_{ij,0}^L = \boldsymbol{W}_{edge}^L \boldsymbol{e}_{ij}^L, \end{aligned} \tag{3}$$

where $\boldsymbol{W}_{node}^I$, $\boldsymbol{W}_{edge}^I$, $\boldsymbol{W}_{node}^L$, and $\boldsymbol{W}_{edge}^L$ are learnable matrices. The obtained representations are used as the input of the first joint convolution module.

3.3.2 *Cascaded Joint Convolution Modules.*

The joint convolution module enables joint updating of the two graphs for context-aware representation learning. Specifically, in each joint convolution module, we first perform *language graph convolution* to update the representations of the language scene graph. The updated language scene graph is further exploited to guide the visual graph convolution to minimize the influence of the irrelevant objects and relationships in images. We devise *edge-attention-based visual convolution* and *node-attention-based visual convolution* to exploit edges and nodes of the language graph as explicit guidance to guide the visual graph convolution process, respectively. The two visual convolution operations use graph-level edge/node attention to assign different weights for different edges/nodes in visual graph convolution. In each joint convolution module only one kind of visual convolution operation is exploited. All the graph convolution operations are shown in Figure 3 (b).

Language Graph Convolution. We devise specific graph convolution operations for edges and nodes of the language graph to obtain context-aware representations. In the t-th joint convolution module, for an edge e_{ij}^L, we enrich its representations $\boldsymbol{e}_{ij,t-1}^L$ with the representations of the nodes it connects, $\boldsymbol{v}_{i,t-1}^L$ and $\boldsymbol{v}_{j,t-1}^L$, via

$$\boldsymbol{e}_{ij,t}^L = \boldsymbol{W}_{rel}^L \left[\boldsymbol{v}_{i,t-1}^L; \boldsymbol{e}_{ij,t-1}^L; \boldsymbol{v}_{j,t-1}^L\right], \tag{4}$$

where $\boldsymbol{W}_{rel}^L$ is a learnable weight matrix and $[\cdot;\cdot]$ denotes the concatenation operation of two vectors. Since a node v_i^L can be the "subject" and the "object" simultaneously in different relationships, two transformation matrices are introduced and its context-aware representation is computed by

$$\boldsymbol{v}_{i,t}^{L} = \frac{1}{M_i}\Big(\sum_{j} \boldsymbol{W}_{sub}^{L}[\boldsymbol{v}_{i,t-1}^{L}; \boldsymbol{e}_{ij,t-1}^{L}; \boldsymbol{v}_{j,t-1}^{L}] + \sum_{k} \boldsymbol{W}_{obj}^{L}[\boldsymbol{v}_{k,t-1}^{L}; \boldsymbol{e}_{ki,t-1}^{L}; \boldsymbol{v}_{i,t-1}^{L}]\Big), \tag{5}$$

where $\boldsymbol{W}_{sub}^{L}$ and $\boldsymbol{W}_{obj}^{L}$ are learnable weight matrices. M_i is the number of relationships where v_i^L appears.

Edge-attention-based visual convolution. The graph-level edge attention aims to highlight important edges for each node in the graph convolution. It consists of two types of edge attention mechanisms, intra-class edge attention and inter-class edge attention. Suppose there are M_e^L edges in the language graph and $\boldsymbol{e}_{k,t-1}^{L}$ is the representation of the k-th edge e_k^L. In the t-th joint convolution module, for a node v_i^I, each type of edge attention computes the attention weight $\alpha_{ij,t}^{k,type}$ between each corresponding edge $\boldsymbol{e}_{ij,t-1}^{I,type}$ and each edge $\boldsymbol{e}_{k,t-1}^{L}$ in the language graph. The attention weights $\alpha_{ij,t}^{k,type}$ are first normalized via the softmax function over j, and then averaged over k to obtain the final attention weight $A_{ij,t}^{type}$, which is given by

$$\begin{aligned}\alpha_{ij,t}^{k,type} &= \boldsymbol{w}_{a,type}^{\top} \tanh\left(\boldsymbol{W}_{L,type}^{a}\boldsymbol{e}_{k,t-1}^{L} + \boldsymbol{W}_{I,type}^{a}\boldsymbol{e}_{ij,t-1}^{I,type}\right), \\ A_{ij,t}^{type} &= \frac{1}{M_e^L}\sum_{k}^{M_e^L} Softmax_j(\alpha_{ij,t}^{k,type}),\end{aligned} \tag{6}$$

where $\boldsymbol{W}_{L,type}^{a}$, $\boldsymbol{W}_{I,type}^{a}$, and $\boldsymbol{w}_{a,type}$ are learnable weights. To update the representation $\boldsymbol{v}_{i,t-1}^{I}$ for v_i^I, we aggregate its two types of edges respectively, and concatenate the obtained representations with the input representation. Concretely, the *edge-attention-based visual convolution* is performed by

$$\boldsymbol{v}_{i,t}^{I} = \Big[\boldsymbol{v}_{i,t-1}^{I}; \sum_{j} A_{ij,t}^{intra}\boldsymbol{e}_{ij,t-1}^{I,intra}; \sum_{k} A_{ik,t}^{inter}\boldsymbol{e}_{ik,t-1}^{I,inter}\Big]. \tag{7}$$

Node-attention-based visual convolution. Similarly, for each node, the graph-level node attention aims to highlight its neighborhoods relevant to the sentence. The node attention weight for a node v_i^I in the t-th joint convolution module is computed by

$$\begin{aligned}\alpha_{i,t}^{k,node} &= \boldsymbol{w}_{a,node}^{\top} \tanh\left(\boldsymbol{W}_{L,node}^{a}\boldsymbol{v}_{k,t-1}^{L} + \boldsymbol{W}_{I,node}^{a}\boldsymbol{v}_{i,t-1}^{I}\right), \\ A_{i,t}^{node} &= \frac{1}{M}\sum_{k}^{M} Softmax_i(\alpha_{i,t}^{k,node}),\end{aligned} \tag{8}$$

where $\boldsymbol{W}_{L,node}^{a}$, $\boldsymbol{W}_{I,node}^{a}$, and $\boldsymbol{w}_{a,node}$ are learnable weights, and $\boldsymbol{v}_{k,t-1}^{L}$ is the representation of the k-th node in the language graph. For each node v_i^I, we aggregate the representations of its neighborhoods and concatenate the obtained representation with the input representation. Concretely, the *node-attention-based visual convolution* is performed by

$$\boldsymbol{v}_{i,t}^{I} = \Big[\boldsymbol{v}_{i,t-1}^{I}; \sum_{j=1}^{N} A_{j}^{node,t}\boldsymbol{v}_{j,t}^{I}\Big]. \tag{9}$$

Note that in each joint convolution module, only one kind of language-guided visual graph convolution is used. Specifically, the

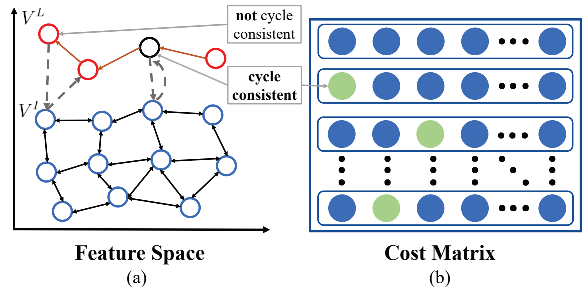

Figure 4: Illustration of the semantic cycle-consistency. (a) shows a language graph and a visual graph in the common feature space. The black cycle represents a cycle-consistent node in the language scene graph while the red cycle in the left corner is a non-cycle-consistent node. (b) shows the cost matrix of the two graphs. For the cycle-consistent node, the corresponding element in the cost matrix is the minimal in both the row and the column it belongs to.

first joint convolution module uses edge-attention-based visual convolution to enrich the node representations with relative location information. The following modules use the node-attention-based visual convolution to fully capture the interaction among objects relevant to the expression for modeling high-order relationships. The output of the last module, $\boldsymbol{v}_{i,3}^{I}$ and $\boldsymbol{v}_{i,3}^{L}$, which characterize rich structure information for graph matching, is regarded as the unified contextual representations $\boldsymbol{u}_i^I$ and $\boldsymbol{u}_i^L$, respectively.

3.4 Graph Matching

By learning unified contextual node representations, the QAP problem is relaxed as a linear assignment problem. Traditional methods to solve linear assignment problem given by Eq. (2) either optimize the cost matrix $\boldsymbol{C}$ with the assignment matrix $\boldsymbol{A}$ fixed or optimize $\boldsymbol{C}$ and $\boldsymbol{A}$ iteratively [48]. By contrast, we solve this problem in an end-to-end manner for better compatibility by introducing specific loss functions for the cross-modal graph convolutional network.

3.4.1 Graph Matching for Referring Expression Comprehension.

For referring expression comprehension, we exploit the semantic cycle-consistency between the two graphs to learn node representations appropriate for matching in an end-to-end manner, inspired by the temporal cycle-consistency used in video alignment [7]. We introduce a semantic cycle-consistency loss, which forces all nodes in the language scene graph to be cycle-consistent nodes, to encourage one-to-one mapping between the two graphs. Figure 4 (a) shows a cycle-consistent node and a non-cycle-consistent node in the common feature space. For a node v_i^L in the language scene graph, its nearest neighbor in the visual graph is denoted as $v_j^I = \arg\min_{v_l^I \in V^I} d(\boldsymbol{u}_l^I, \boldsymbol{u}_i^L)$ and the nearest neighbor of v_j^I in the language graph is denoted as $v_k^L = \arg\min_{v_l^L \in V^L} d(\boldsymbol{u}_l^L, \boldsymbol{u}_j^I)$. The node v_i^L is a cycle-consistent node if and only if $i = k$.

To guarantee the differentiability of the cycle-consistency loss, for each node v_i^L, we first find its soft nearest neighbor $\widetilde{v}_j^I$ in the visual graph and treat the identification of the nearest neighbor of

v_j^I as a classification task. The soft nearest neighbor of the selected point v_i^L is computed via the softmax function given by

$$\widetilde{\boldsymbol{u}}_j^{\boldsymbol{I}} = \sum_{l}^{N} \alpha_l \boldsymbol{u}_l^I, \quad \alpha_l = Softmax_l\big(\cos\big(\boldsymbol{u}_l^I, \boldsymbol{u}_i^L\big)\big). \tag{10}$$

Then we measure the similarity between the $\widetilde{\boldsymbol{u}}_j^{\boldsymbol{I}}$ and all the nodes in the language scene graph, and obtain the predicted labels as

$$\hat{\boldsymbol{y}}_{i,l} = softmax_l\big(x_{i,l}\big), \quad x_{i,l} = \cos\big(\boldsymbol{u}_l^L, \widetilde{\boldsymbol{u}}_j^I\big). \tag{11}$$

The semantic cycle-consistency loss of an image and a sentence is thus given by

$$\mathcal{L}_{cycle} = -\sum_{i}^{M} \boldsymbol{y}_i \log\big(\hat{\boldsymbol{y}}_i\big), \tag{12}$$

where $\boldsymbol{y}_i$, whose the i-th element is 1 and others are 0, is the ground-truth label for the classification task of v_i^L.

Apart from the self-supervised cycle-consistent loss, we incorporate a supervised matching loss function to make full use of referent object annotations. Considering that the referent object is usually the center node modified by others, we regard the node in the language graph whose in-degree is zero as the referent node, denoted as v_*^L. Its similarity with each node v_i^I in the visual graph is measured as

$$s_i = \tanh\big(\boldsymbol{W}_L \boldsymbol{u}_*^L\big) \cdot \tanh\big(\boldsymbol{W}_I \boldsymbol{u}_i^I\big), \tag{13}$$

where $\boldsymbol{W}_L$ and $\boldsymbol{W}_I$ are two learnable weight matrices. The supervised matching loss of an image and a sentence is given by

$$\mathcal{L}_{match} = -\boldsymbol{l} \log\big(softmax(\boldsymbol{s})\big), \tag{14}$$

where $\boldsymbol{l}$ is the one-hot label whose element representing the ground-truth object is 1 and others are 0. Note that here we use the representation of the referent node rather than holistic linguistic representations of the sentence. The overall training objective for referring expression comprehension is given by

$$\mathcal{L}_{refer} = \lambda \mathcal{L}_{cycle} + \mathcal{L}_{match}, \tag{15}$$

where λ is a hyper-parameter which balances the two loss terms.

As shown in Figure 4 (b), by encouraging a node v_i^L to be cycle-consistent, we obtain an element $\boldsymbol{C}_{ij}$ that is the minimal in both the row and the column it belongs to. Intuitively, by forcing all nodes in the language graph to be cycle-consistent, we can guarantee that in $\boldsymbol{C}$ the minimal element of each row is also the minimal in the corresponding column. Thus by assigning 1 to the corresponding positions (green filled circles in the figure) of $\boldsymbol{C}$ and 0 to other positions, we can obtain the assignment matrix $\boldsymbol{A}$.

3.4.2 *Graph Matching for Phrase Localization.*

For phrase localization, the ground-truth node correspondences are available, thus we introduce a permutation loss to directly optimize the assignment matrix $\boldsymbol{A}$ for minimizing Eq. (2) as

$$\mathcal{L}_{perm} = -\sum_{i,j}\big(\boldsymbol{A}_{i,j}^{GT} \log \hat{\boldsymbol{A}}_{i,j} + \big(1 - \boldsymbol{A}_{i,j}^{gt}\big)\log\big(1 - \hat{\boldsymbol{A}}_{i,j}\big)\big), \tag{16}$$

where $\boldsymbol{A}^{GT}$ is the ground-truth assignment matrix and $\hat{\boldsymbol{A}}$ is the predicted assignment matrix. We compute pairwise similarity matrix of contextual representations $\boldsymbol{U}^I$ and $\boldsymbol{U}^L$ and transform the obtained matrix via a differentiable Sinkhorn layer as [42] to obtain $\hat{\boldsymbol{A}}$.

Apart from the permutation loss, we further use an extra bounding box regression loss to estimate a 4-d offset vector of each object proposal as

$$\mathcal{L}_{reg} = \sum_{i \in \{x,y,w,h\}} \mathrm{SmoothL}_1\big(\hat{\boldsymbol{R}}_i - \boldsymbol{R}_i\big), \tag{17}$$

where $\boldsymbol{R}$ is the ground-truth offset vector. $\hat{\boldsymbol{R}}$ is the predicted vector computed by $\hat{\boldsymbol{R}} = \boldsymbol{W}_{reg}[\boldsymbol{u}^L; \boldsymbol{u}^I]$, where $\boldsymbol{u}^L$ and $\boldsymbol{u}^I$ represent the visual and textual nodes, respectively. The overall training objective for phrase localization is given by

$$\mathcal{L}_{pl} = \mathcal{L}_{perm} + \eta \mathcal{L}_{reg}, \tag{18}$$

where η is a hyper-parameter to balance the two loss terms.

4 EXPERIMENTS

We apply the proposed method on referring expression comprehension and phrase localization to evaluate its effectiveness.

4.1 Referring Expression Comprehension

4.1.1 *Experimental setting.*

Datasets. We conduct experiments on three widely-used referring expression comprehension datasets based on MS-COCO dataset [20]: RefCOCO [50], RefCOCO+ [50], and RefCOCOg [27]. The RefCOCO [50] consists of 142, 210 referring expressions for 50, 000 objects in 19, 994 images. The RefCOCO+ [50] consists of 141, 564 referring expressions for 49, 856 objects in 19, 992 images. The two datasets were collected in an interactive game [16] and thus the referring expressions are usually short phrases. The difference between them is that absolute location words are not allowed in the referring expressions of the RefCOCO+. Both the two datasets have four splits: "train", "val", "testA", and "testB". The "testA" split evaluates images containing multiple people, while the "testB" evaluates images containing multiple instances of all other objects. The RefCOCOg [27] was collected in a non-interactive setting and consists of 95, 010 long declarative referring expressions for 49, 822 objects in 21, 899 images. We adopt the partition in [28], where objects are divided into "train" split, "val" split, and "test" split by restricting all objects of an image to appear in only one split.

Implementation details. In our implementations, we use ground-truth object regions contained in the MS-COCO dataset. Same as [41], we use VGG16 [37] as the backbone to extract features with the dimension of 512 for objects in images. For linguistic input, we pre-process the referring expression to a maximum of 10 words for RefCOCO and RefCOCO+, and 20 words for RefCOCOg. The extra words are discarded and the shorter language expressions are padded with vectors of zeros. We build our model based on the PyTorch framework [29]. The batch size is fixed as 30. All sentences associated with these images are fed into the model. We use Adam [17] as the training optimizer and set the initial learning rate as 0.001, which decays by a factor of 10 every 6000 iterations. In training, the trade-off parameter λ is fixed as 0.01.

4.1.2 *Results.*

Comparisons with the State-of-the-Art. The results of the proposed method and the state-of-the-art are listed in Table 1. We

Table 1: Results on referring expression comprehension datasets. All methods use VGG16 features.

Methods	*RefCOCO*			*RefCOCO+*			*RefCOCOg*	
	val	testA	testB	val	testA	testB	val	test
MMI [27]	-	71.72	71.09	-	58.42	51.23	-	-
NegBag [28]	76.90	75.60	78.00	-	-	-	-	68.40
CMN [11]	-	75.94	79.57	-	59.29	59.34	-	-
listener [51]	77.48	76.58	78.94	60.5	61.39	58.11	69.93	69.03
VariContxt [54]	-	78.98	82.39	-	62.56	62.90	-	-
MAttNet [49]	80.94	79.99	82.30	63.07	65.04	61.77	73.04	72.79
ParallelAttn [56]	81.67	80.81	81.32	64.18	66.31	61.46	-	-
RVGTREE [9]	79.04	78.82	80.53	62.38	62.82	61.28	72.32	71.95
AccumulateAttn [5]	81.27	81.17	80.01	65.56	68.76	60.63	-	-
LGRAN [41]	82.0	81.2	84.0	66.6	67.6	65.5	75.4	74.7
Ours	**82.68**	**82.06**	**84.24**	**67.70**	**69.34**	**65.74**	**75.73**	**75.31**

Table 2: Ablation studies of the proposed method on referring expression comprehension datasets.

Methods	*RefCOCO+*			*RefCOCOg*	
	val	testA	testB	val	test
Ours w/o cycle	66.72	69.04	64.78	74.29	74.76
Ours w/o JC	62.97	63.93	60.13	70.15	70.53
Ours JC(#1)	65.53	68.46	63.39	73.71	73.89
Ours JC(#1+#2)	67.21	69.04	65.47	75.25	74.83
Ours JC(#1+#2+#3+#4)	67.46	**69.40**	65.02	75.02	74.60
Ours	**67.70**	69.34	**65.74**	**75.73**	**75.31**

didn't compare with [45] and [46], because they use the Visual Genome [18] as an additional dataset to train the object detector, but most existing methods only use the MSCOCO dataset [20]. As shown in the table, benefiting from the representation capacity of graphs and full alignments of textual semantics and visual concepts, the proposed method outperforms the others in all datasets. In particular, the LGRAN [41] performs reasoning over the visual scene graph while the RVGTREE [9] performs reasoning over the dependency parsing tree in a bottom-up manner. By contrast, the proposed method can achieve joint reasoning over the language graph and the visual graph via the cross-modal graph convolutional network for more comprehensive context modeling. Besides, the self-supervised semantic cycle-consistency loss guarantees that our method can fully capture fine-grained correspondences between the two modalities as extra supervision information. Thus our method outperforms the LGRAN and the RVGTREE.

Ablation Studies. To evaluate the effectiveness of several important components of our method, we re-train different versions of our model by ablating certain components. The results of those models on the RefCOCO and the RefCOCOg are listed in Table 2.

Firstly, to investigate the influence of the joint convolution module, we cascade different numbers of joint convolution modules in the cross-modal graph convolutional network. It can be found that the number of joint convolution modules is critical to our method. Taking into account no relative information among objects, "Ours w/o JC" performs much worse than "Ours JC(#1)", which only contains the first joint convolution module. By cascading more modules with node attention, the performance can be further improved since contextual information of both the image and the language expression is fully modeled to better understand and ground the multi-order relationships. However, we also find that if more than three joint convolution modules are cascaded, the accuracy of the proposed method will decrease. A possible reason is that the overly complex convolution process brings redundant information.

Secondly, we study the effectiveness of the cycle-consistency loss in the proposed method. We use only the matching loss to train a model denoted as "Ours w/o cycle". We observe that the cycle-consistency loss improves the accuracy of our method although no additional supervision information is introduced. Particularly, it brings more improvement in the RefCOCOg because language expressions in this dataset contain more context objects.

4.2 Phrase Localization

4.2.1 *Experimental setting.*

Dataset. We conduct experiments on the Flickr30k Entities dataset [32] to evaluate the effectiveness of our method for phrase localization. The Flickr30k Entities dataset contains 31,783 images, over 275K bounding boxes and over 360K phrases. Each image is associated with 5 captions. We use 29,783/1000/1000 images for training/validation/testing. For each phrase, the grounding is regarded as correct if the IoU (intersection over union) between the predicted region and the ground-truth region is higher than 0.5. If a phrase is associated with multiple ground-truth bounding boxes, we merge them into a new enclosing box as prior work [23, 24, 34].

Implementation details. For visual input, we use the Faster R-CNN [33] from Anderson *et al.* [1] to generate 100 proposals for each image. Note that for phrase localization we perform proposal pruning to select a small set of high-quality proposals for each phrase as [24, 32, 34]. For textual input, we use the 1024-d contextualized word embeddings from the last layer of ELMo [30] to initialize the word embeddings as [23]. In training, the trade-off parameter η is fixed as 10.

4.2.2 *Results.*

Comparisons with the State-of-the-Art. The results of state-of-the-art methods and our method on the Flickr30k Entities dataset are listed in Table 3. As shown in the table, our method outperforms all other methods. The main reason is that our model can thoroughly characterize contextual information of images and sentences and achieve full alignments of textual semantics and visual concepts. Note that the LCMCG [24] and the G3RAPHGROUND++ [4] also

Table 3: Results of our method and the state-of-the-art on the Flickr30k Entities dataset

Methods	Overll	People	Clothing	Bodyparts	Animal	Vehicles	Instruments	Scene	Other
SMPL [40]	42.08	57.89	34.61	15.87	55.98	52.25	23.46	34.22	26.23
GroundeR [34]	47.81	61.00	38.12	10.33	62.55	68.75	36.42	58.18	29.08
SPC [26]	55.49	71.69	50.95	25.24	76.23	66.50	35.80	51.51	35.98
CITE [31]	59.27	73.20	52.34	30.59	76.25	75.75	48.15	55.64	42.83
SeqGROUND [6]	61.60	76.02	56.94	26.18	75.56	66.00	39.36	68.69	40.60
G3RAPHGROUND++ [4]	66.93	78.86	68.34	39.80	81.38	76.58	42.35	68.82	45.08
DDPN [52]	73.30	-	-	-	-	-	-	-	-
SL-CCRF [23]	74.69	84.41	78.51	46.74	88.89	81.41	64.97	75.95	57.57
LCMCG [24]	76.74	**86.82**	79.92	**53.54**	90.73	84.75	**63.58**	77.12	58.65
Ours	**76.87**	86.57	**79.92**	52.77	**91.89**	**85.25**	58.64	**78.78**	**59.04**

Figure 5: Qualitative examples from the test split of the Flickr30k Entities dataset. The predicted objects are marked via bounding boxes whose color is the same as corresponding noun phrases in descriptions. In the third column, the boxes with thinner lines represent predictions of models without the constraint of one-to-one mapping. The last column shows failure cases, where black boxes are the incorrect predictions and white boxes are ground-truths.

use graphs to represent the contextual structure of the image and the sentence. However, the G3RAPHGROUND++ fuse visual graphs and language graphs to get the final grounding decision rather than finding the correspondences between graphs. The LCMCG only uses the graph matching as a post-processing procedure. By contrast, our method learns informative node representations and finds their correspondences in a unified framework.

Qualitative Results. We show some qualitative examples of the proposed method in Figure 5 to demonstrate the effectiveness of our framework for localizing multiple noun phrases in an image. Specifically, examples in the first column show our method can localize different kinds of entities with huge overlaps among the corresponding objects. The second column shows the effectiveness of our method in grounding noun phrases that corresponding to several objects in the image, such as "a group of kids", "a couple bicycles", and "people". In the third column, the boxes with thinner lines represent predictions of a model via a KL-divergence loss without the one-to-one mapping constraint. In the upper sample of the third column, the model without the constraint associates the shirt of the man with two entities ("a gray shirt" and "a blue sweatshirt") while our method can correctly ground the two entities. This demonstrates that our method can avoid matching ambiguity benefiting from the graph matching. Several failure cases are shown in the last column. Our method may fail to ground objects or background in images accurately with huge occlusions.

5 CONCLUSION

In this paper, we have presented a visual-semantic graph matching method for visual grounding. Our method achieves full alignments between textual semantics and visual concepts by solving the graph matching between a visual scene graph and a language scene graph. Using a cross-modal graph convolutional network, the proposed method learns unified visual-semantic node representations for the two heterogeneous graphs. The introduction of a permutation loss and a self-supervised semantic cycle-consistency loss further enables one-to-one mapping between the two graphs with or without ground-truth correspondences. Experimental results on referring expression comprehension and phrase localization demonstrate that our method can effectively associate the noun phrases in sentences with the corresponding objects in images.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grants No. 61773062 and No. 61702037, and Alibaba Group through Alibaba Innovative Research Program. Qi Wu is not supported by any of the projects above.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6077–6086.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3674–3683.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2425–2433.

[4] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. 2019. G3raphGround: Graph-Based Language Grounding. In *The IEEE International Conference on Computer Vision (ICCV)*. 4281–4290.

[5] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. 2018. Visual grounding via accumulated attention. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7746–7755.

[6] Pelin Dogan, Leonid Sigal, and Markus Gross. 2019. Neural sequential phrase grounding (seqground). In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4175–4184.

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2019. Temporal Cycle-Consistency Learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1801–1810.

[8] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2712–2719.

[9] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. 2019. Learning to Compose and Reason with Language Tree Structures for Visual Grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2019).

[10] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. 2019. Language-Conditioned Graph Networks for Relational Reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 10294–10303.

[11] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2017. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1115–1124.

[12] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4555–4564.

[13] Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. In *Advances in Neural Information Processing Systems (NeurIPS)*. 5901–5914.

[14] Chenchen Jing, Yuwei Wu, Xiaoxun Zhang, Jia Yunde, and Qi Wu. 2020. Overcoming Language Priors in VQA via Decomposed Linguistic Representations. In *Thirty-Forth AAAI Conference on Artificial Intelligence*. 11181–11188.

[15] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3668–3678.

[16] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 787–798.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123, 1 (2017), 32–73.

[19] Marius Leordeanu and Martial Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1482–1489.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 740–755.

[21] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Fanglin Wang. 2019. Referring Expression Grounding by Marginalizing Scene Graph Likelihood. *arXiv preprint arXiv:1906.03561* (2019).

[22] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. 2019. Learning to Assemble Neural Module Tree Networks for Visual Grounding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 4673–4682.

[23] Jiacheng Liu and Julia Hockenmaier. 2019. Phrase Grounding by Soft-Label Chain Conditional Random Field. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 5115–5125.

[24] Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He. 2020. Learning Cross-modal Context Graph for Visual Grounding. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*. 11645–11652.

[25] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. 2007. A survey for the quadratic assignment problem. *European journal of operational research* 176, 2 (2007), 657–690.

[26] Ruotian Luo and Gregory Shakhnarovich. 2017. Comprehension-guided referring expressions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7102–7111.

[27] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11–20.

[28] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. 2016. Modeling context between objects for referring expression understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 792–807.

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[30] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[31] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. 2018. Conditional image-text embedding networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 249–264.

[32] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2641–2649.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. 91–99.

[34] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 817–834.

[35] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[36] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*. 70–80.

[37] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[38] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3156–3164.

[39] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5005–5013.

[40] Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. 2016. Structured matching for phrase localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 696–711.

[41] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. 2019. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1960–1968.

[42] Runzhong Wang, Junchi Yan, and Xiaokang Yang. 2019. Learning Combinatorial Embedding Networks for Deep Graph Matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 3056–3065.

[43] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6629–6638.

[44] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding* 163 (2017), 21–40.

[45] Sibei Yang, Guanbin Li, and Yizhou Yu. 2019. Cross-Modal Relationship Inference for Grounding Referring Expressions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4145–4154.

[46] Sibei Yang, Guanbin Li, and Yizhou Yu. 2019. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 4644–4653.

[47] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-encoding scene graphs for image captioning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 10685–10694.

[48] Mang Ye, Jiawei Li, Andy J Ma, Liang Zheng, and Pong C Yuen. 2019. Dynamic graph co-matching for unsupervised video-based person re-identification. *IEEE Transactions on Image Processing (TIP)* 28, 6 (2019), 2976–2990.

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

[50] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 69–85.

[51] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7282–7290.

[52] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. 2018. Rethinking Diversified and Discriminative Proposal Generation for Visual Grounding. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[53] Andrei Zanfir and Cristian Sminchisescu. 2018. Deep learning of graph matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2684–2693.

[54] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. 2018. Grounding referring expressions in images by variational context. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4158–4166.

[55] Feng Zhou and Fernando De la Torre. 2012. Factorized graph matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 127–134.

[56] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton van den Hengel. 2018. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4252–4261.