This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Graph-Structured Referring Expression Reasoning in The Wild

Sibei Yang[1] Guanbin Li[2†] Yizhou Yu[1,3†]

[1]The University of Hong Kong [2]Sun Yat-sen University [3]Deepwise AI Lab

sbyang9@hku.hk, liguanbin@mail.sysu.edu.cn, yizhouy@acm.org

Abstract

Grounding referring expressions aims to locate in an image an object referred to by a natural language expression. The linguistic structure of a referring expression provides a layout of reasoning over the visual contents, and it is often crucial to align and jointly understand the image and the referring expression. In this paper, we propose a scene graph guided modular network (SGMN), which performs reasoning over a semantic graph and a scene graph with neural modules under the guidance of the linguistic structure of the expression. In particular, we model the image as a structured semantic graph, and parse the expression into a language scene graph. The language scene graph not only decodes the linguistic structure of the expression, but also has a consistent representation with the image semantic graph. In addition to exploring structured solutions to grounding referring expressions, we also propose Ref-Reasoning, a large-scale real-world dataset for structured referring expression reasoning. We automatically generate referring expressions over the scene graphs of images using diverse expression templates and functional programs. This dataset is equipped with real-world visual contents as well as semantically rich expressions with different reasoning layouts. Experimental results show that our SGMN[1] not only significantly outperforms existing state-of-the-art algorithms on the new Ref-Reasoning dataset, but also surpasses state-of-the-art structured methods on commonly used benchmark datasets. It can also provide interpretable visual evidences of reasoning.

1. Introduction

Grounding referring expressions aims to locate in an image an object referred to by a natural language expression, and the object is called the referent. It is a challenging problem because it requires understanding as well as performing reasoning over semantics-rich referring expressions and diverse visual contents including objects, attributes and relations.

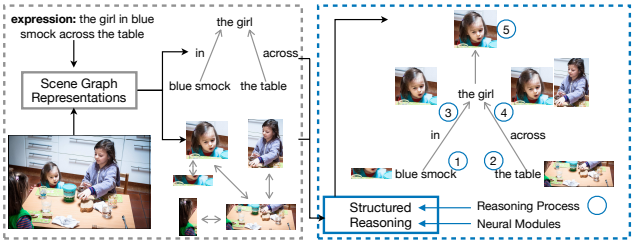

Figure 1. Scene Graph guided Modular Network (SGMN) for grounding referring expressions. SGMN first parses the expression into a language scene graph and models the image as a semantic graph, then it performs structured reasoning with neural modules under the guidance of the language scene graph.

Analyzing the linguistic structure of referring expressions is the key to grounding referring expressions because they naturally provide the layout of reasoning over the visual contents. For the example shown in Figure 1, the composition of the referring expression "the girl in blue smock across the table" (*i.e.*, triplets ("the girl", "in", "blue smock") and ("the girl", "across" , "the table")) reveals a tree-structured layout of finding the blue smock, locating the table and identifying the girl who is "in" the blue smock and meanwhile is "across" the table. However, nearly all the existing works either neglect linguistic structures and learn holistic matching scores between monolithic representations of referring expressions and visual contents [18, 29, 23] or neglect syntactic information and explore limited linguistic structures via self-attention mechanisms [26, 8, 24].

Consequently, in this paper, we propose a Scene Graph guided modular network (SGMN) to fully analyze the linguistic structure of referring expressions and enable reasoning over visual contents using neural modules under the guidance of the parsed linguistic structure. Specifically, SGMN first models the input image with a structured rep-

†Corresponding authors. This work was partially supported by the Hong Kong PhD Fellowship, the Guangdong Basic and Applied Basic Research Foundation under Grant No.2020B1515020048, the National Natural Science Foundation of China under Grant No.61976250 and No.U1811463.

[1]Data and code are available at https://github.com/sibeiyang/sgmn

resentation, which is a directed graph over the visual objects in the image. The edges of the graph encode the semantic relations among the objects. Second, SGMN analyzes the linguistic structure of the expression by parsing it into a language scene graph [21, 16] using an external parser, including the nodes and edges of which correspond to noun phrases and prepositional/verb phrases respectively. The language scene graph not only encodes the linguistic structure but is also consistent with the semantic graph representation of the image. Third, SGMN performs reasoning on the image semantic graph under the guidance of the language scene graph by using well-deigned neural modules [2, 22] including AttendNode, AttendRelation, Transfer, Merge and Norm. The reasoning process can be explicitly explained via a graph attention mechanism.

In addition to methods, datasets are also important for making progress on grounding referring expressions, and various real-world datasets have been released [12, 19, 27]. However, recent work [4] indicates dataset biases exist and they may be exploited by the methods. And methods accessing the images only achieve marginally higher performance than a random guess. Existing datasets also have other limitations. First, the samples in the datasets have unbalanced levels of difficulty. Many expressions in the datasets directly describe the referents with attributes due to the annotation process. Such an imbalance makes models learn shallow correlations instead of achieving joint image and text understanding, which defeats the original intention of grounding referring expressions. Second, evaluation is only conducted on final predictions but not on the intermediate reasoning process [17], which does not encourage the development of interpretable models [24, 15]. Thus, a synthetic dataset over simple 3D shapes with attributes is proposed in [17] to address these limitations. However, the visual contents in this synthetic dataset are too simple, which is not conducive to generalizing trained models on the synthetic dataset to real-world scenes.

To address the aforementioned limitations, we build a large-scale real-world dataset, named Ref-Reasoning. We generate semantically rich expressions over the scene graphs of images [14, 10] using diverse expression templates and functional programs, and automatically obtain the ground-truth annotations at all intermediate steps during the modularized generation process. Furthermore, we carefully balance the dataset by adopting uniform sampling and controlling the distribution of expression-referent pairs over the number of reasoning steps.

In summary, this paper has the following contributions:

• A scene graph guided modular neural network is proposed to perform reasoning over a semantic graph and a scene graph using neural modules under the guidance of the linguistic structure of referring expressions, which meets the fundamental requirement of grounding referring expressions.

• A large-scale real-word dataset, Ref-Reasoning, is constructed for grounding referring expressions. Ref-Reasoning includes semantically rich expressions describing objects, attributes, direct and indirect relations with a variety of reasoning layouts.

• Experimental results demonstrate that the proposed method not only significantly surpasses existing state-of-the-art algorithms on the new Ref-Reasoning dataset, but also outperforms state-of-the-art structured methods on common benchmark datasets. In addition, it can provide interpretable visual evidences of reasoning.

2. Related Work

2.1. Grounding Referring Expressions

A referring expression normally not only directly describes the appearance of the referent, but also its relations to other objects in the image, and its reference information depends on the meanings of its constituent expressions and the rules used to compose them [9, 24]. However, most of the existing works [29, 23, 25] neglect linguistic structures and learn holistic representations for the objects in the image and the expression. Recently, there are some works which involve the expression analysis into their models, and learn the components of expression and visual inference from end to end. The methods in [9, 26, 28] softly decompose the expression into different semantic components relevant to different visual evidences, and compute a matching score for every component. They use fixed semantic components, *e.g.* subject-relation-object triplets [9] and subject-location-relation components [26], which are not feasible for complex expressions. DGA [24] analyzes linguistic structures for complex expressions by iteratively attending their constituent expressions. However, they all resort to self-attention on the expression to explore its linguistic structure but neglect its syntactic information. Another work [3] grounds the referent using a parse tree, where each node of the tree is a word (or phrase) which can be a noun, preposition or verb.

2.2. Dataset Bias and Solutions

Recently, the dataset bias began to be discussed for grounding referring expressions [4, 17]. The work in [4] reveals even the linguistically-motivated models tend to learn shallow correlations instead of making use of linguistic structures because of the dataset bias. In addition, expression-independent models can achieve high performance. The dataset bias can have a significantly negative impact on the evaluation of a model's readiness for joint understanding and reasoning for language and vision.

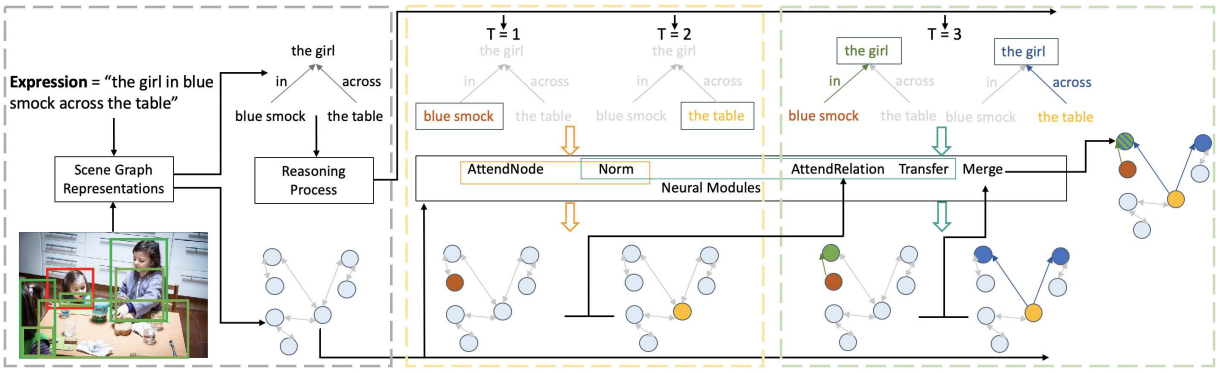

Figure 2. An overview of our Scene Graph guided Modular Network (SGMN)(better viewed in color). Different colors represent different nodes in the language scene graph and their corresponding nodes in the image semantic graph. SGMN parses the expression into a language scene graph and constructs an image semantic graph over the objects in the input image. Next, it performs reasoning under the guidance of the language scene graph. It first locates the nodes in the image semantic graph for the leaf nodes in the language scene graph using neural modules AttendNode and Norm. Then for the intermediate nodes in the language scene graph, it uses AttendRelation, Transfer and Norm modules to attend the nodes in the image semantic graph, and the Merge module to combine the attention results.

In order to address the above problem, the work in [17] proposes a new diagnostic dataset, called CLEVR-Ref+. Same as CLEVR [11] in visual question answering, it contains rendered images and automatically generated expressions. In particular, the objects in the images are simple 3D shapes with attributes (*i.e.*, color, size and material), and the expressions are generated using designed templates which include spatial and same-attribute relations. However, the models trained on this synthetic dataset cannot be easily generalized to real-world scenes because the visual contents (*i.e.*, simple 3D shapes with attributes and spatial relations) are too simple to jointly reason about language and vision.

Thanks for the scene graph annotations of real-world images provided in the Visual Genome datasets [14] and further cleaned in the GQA dataset [10], we generate semantically rich expressions over the scene graphs with objects, attributes and relations using carefully designed templates along with functional programs.

3. Approach

We now present the proposed scene graph guided modular network (SGMN). As illustrated in Figure 2, given an input expression and an input image with visual objects, our SGMN first builds a pair of semantic graph and scene graph representations for the image and expression respectively, and then performs structured reasoning over the graphs using neural modules.

3.1. Scene Graph Representations

Scene graph based representations form the basis of our structured reasoning. In particular, the image semantic graph flexibly captures and represents all the visual contents needed for grounding referring expressions in the input image while the language scene graph explores the linguistic structure of the input expression, which defines the layout of the reasoning process. In addition, these two types of graphs have consistent structures, where the nodes and edges of the language scene graph respectively correspond to a subset of the nodes and edges of the image semantic graph.

3.1.1 Image Semantic Graph

Given an image with objects $\mathcal{O} = \{o_i\}_{i=1}^N$, we define the image semantic graph over the objects $\mathcal{O}$ as a directed graph, $\mathcal{G}^o = (\mathcal{V}^o, \mathcal{E}^o)$, where $\mathcal{V}^o = \{v_i^o\}_{i=1}^N$ is the set of nodes and node v_i^o corresponds to object o_i; $\mathcal{E}^o = \{e_{ij}^o\}_{i,j=1}^N$ is the set of directed edges, and e_{ij}^o is the edge from v_j^o to v_i^o, which denotes the relation between objects o_j and o_i.

For each node v_i^o, we obtain two types of features, visual feature $\mathbf{v}_i^o$ extracted from a pretrained CNN model and spatial feature $\mathbf{p}_i^o = [x_i, y_i, w_i, h_i, w_i h_i]$, where (x_i, y_i), w_i and h_i are the normalized top-left coordinates, width and height of the bounding box of node v_i respectively. For each edge e_{ij}^o, we compute the edge feature $\mathbf{e}_{ij}^o$ by encoding the relative spatial feature $\mathbf{l}_{ij}^o$ between v_i^o and v_j^o and the visual feature $\mathbf{v}_j^o$ of node v_j^o together because relative spatial information between objects along with their appearance information is the key indicator of their semantic relation [5]. Specifically, the relative spatial feature is represented as $\mathbf{l}_{ij}^o = [\frac{x_j - x_{ci}}{w_i}, \frac{y_j - y_{ci}}{h_i}, \frac{x_j + w_j - x_{ci}}{w_i}, \frac{y_j + h_j - y_{ci}}{h_i}, \frac{w_j h_j}{w_i h_i}]$, where (x_{ci}, y_{ci}) are the normalized center coordinates of the bounding box of node v_i^o. And $\mathbf{e}_{ij}^o$ is the concatenation of an encoded version of $\mathbf{l}_{ij}^o$ and $\mathbf{v}_j^o$, *i.e.*, $\mathbf{e}_{ij}^o = [\mathbf{W}_o^T \mathbf{l}_{ij}^o, \mathbf{v}_j^o]$, where $\mathbf{W}_o$ is a learnable matrix.

3.1.2 Language Scene Graph

Given an expression S, we first use an off-the-shelf scene graph parser [21] to parse the expression into an initial lan-

guage scene graph, where a node and an edge of the graph correspond to an object and the relation between two objects mentioned in S respectively, and the object is represented as an entity with a set of attributes.

We define the language scene graph over S as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_m\}_{m=1}^{M}$ is a set of nodes and node v_m is associated with a noun or noun phrase, which is a sequence of words from S; $\mathcal{E} = \{e_k\}_{k=1}^{K}$ is a set of edges and edge $e_k = (v_{k_s}, r_k, v_{k_o})$ is a triplet of subject node $v_{k_s} \in \mathcal{V}$, object node $v_{k_o} \in \mathcal{V}$ and relation r_k, the direction of which is from v_{k_o} to v_{k_s}. Relation r_k is associated with a preposition/verb word or phrase from S, and e_k indicates that subject node v_{k_s} is modified by object node v_{k_o}.

3.2. Structured Reasoning

We perform structured reasoning on the nodes and edges of graphs using neural modules under the guidance of the structure of language scene graph $\mathcal{G}$. In particular, we first design the inference order and reasoning rules for its nodes $\mathcal{V}$ and edges $\mathcal{E}$. Then, we follow the inference order to perform reasoning. For each node, we adopt the AttendNode module to find its corresponding node in graph $\mathcal{G}^o$ or use the Merge module to combine information from its incident edges. For each edge, we execute specific reasoning steps using carefully designed neural modules, including AttendNode, AttendRelation and Transfer.

3.2.1 Reasoning Process

In this section, we first introduce the inference order, and then present specific reasoning steps on the nodes and edges respectively. In general, for every node in language scene graph $\mathcal{G}$, we learn its attention map over the nodes of image semantic graph $\mathcal{G}^o$ on the basis of its connections.

Given a language scene graph $\mathcal{G}$, we locate the node with zero out-degree as its referent node v_{ref} because the referent is usually modified by other entities rather than modifying other entities in a referring expression. Then, we perform breadth-first traversal of the nodes in graph $\mathcal{G}$ from the referent node v_{ref} by reversing the direction of all edges, meanwhile, push the visited nodes into a stack which is initially empty. Next, we iteratively pop one node from the stack and perform reasoning on the popped node. The stack determines the inference order for the nodes, and one node can reach the top of the stack only after all of its modifying nodes have been processed. This inference order essentially converts graph $\mathcal{G}$ into a directed acyclic graph. Without loss of generality, suppose node v_m is popped from the stack in the present iteration, and we carry out reasoning on node v_m on the basis of its connections to other nodes. There are two different situations: 1) If the in-degree of v_m is zero, v_m is a leaf node, which means node v_m is not modified by any other nodes. Thus, node v_m should be associated with the nodes of image semantic graph $\mathcal{G}^o$ independently; 2) otherwise, if node v_m has incident edges $\mathcal{E}_m \in \mathcal{E}$ starting from other nodes, v_m is an intermediate node, and its attention map over $\mathcal{V}^o$ should depend on the attention maps of its connected nodes and the edges between them.

Leaf node. We learn an embedding for the words associated with the nodes of the language scene graph $\mathcal{G}$ in advance. Then, for node v_m, suppose its associated phrase consists of words $\{w_t\}_{t=1}^{T}$, and the embedded feature vectors for these words are $\{\mathbf{f}_t\}_{t=1}^{T}$. We use a bi-directional LSTM [7] to compute the context of every word in this phrase, and define the concatenation of the forward and backward hidden vectors of a word w_t as its context, denoted as $\mathbf{h}_t$. Meanwhile, we represent the whole phrase using the concatenation of the last hidden vectors of both directions, denoted as $\mathbf{h}$. In a referring expression, an individual entity is often described by its appearance and spatial location. Therefore, we learn feature representations for node v_m from both appearance and spatial location. In particular, inspired by self-attention in [9, 23, 26], we first learn the attention over each word on the basis of its context, and obtain feature representations $\mathbf{v}_m^{look}$ and $\mathbf{v}_m^{loc}$ at node v_m by aggregating attention weighted word embedding as follows,

$$\begin{aligned} \alpha_{t,m}^{look} &= \frac{\exp(\mathbf{W}_{look}^T \mathbf{h}_t)}{\sum_{t=1}^{T} \exp(\mathbf{W}_{look}^T \mathbf{h}_t)}, \mathbf{v}_m^{look} = \sum_{t=1}^{T} \alpha_{t,m}^{look} \mathbf{f}_t \\ \alpha_{t,m}^{loc} &= \frac{\exp(\mathbf{W}_{loc}^T \mathbf{h}_t)}{\sum_{t=1}^{T} \exp(\mathbf{W}_{loc}^T \mathbf{h}_t)}, \mathbf{v}_m^{loc} = \sum_{t=1}^{T} \alpha_{t,m}^{loc} \mathbf{f}_t, \end{aligned} \tag{1}$$

where $\mathbf{W}_{look}$ and $\mathbf{W}_{loc}$ are learnable parameters, and $\mathbf{v}_m^{look}$ and $\mathbf{v}_m^{loc}$ correspond to the appearance and spatial location of node v_m. Then, we feed these two features into the **AttendNode** neural module to compute attention maps $\{\lambda_{n,m}^{look}\}_{n=1}^{N}$ and $\{\lambda_{n,m}^{loc}\}_{n=1}^{N}$ over the nodes of image semantic graph $\mathcal{G}^o$. Finally, we combine these two attention maps to obtain the final attention map for node v_m. A noun phrase may place emphasis on appearance, spatial location or both of them. We flexibly adapt to the variations of noun phrases by learning a pair of weights at node v_m for the attention maps related to appearance and spatial location. The weights (*i.e.* β^{look} and β^{loc}) and the final attention map $\{\lambda_{n,m}\}_{n=1}^{N}$ for node v_m are computed as follows,

$$\begin{aligned} \beta^{look} &= \text{sigmoid}(\mathbf{W}_0^T \mathbf{h} + b_0) \\ \beta^{loc} &= \text{sigmoid}(\mathbf{W}_1^T \mathbf{h} + b_1) \\ \lambda_{n,m} &= \beta^{look} \lambda_{n,m}^{look} + \beta^{loc} \lambda_{n,m}^{loc} \\ \{\lambda_{n,m}\}_{n=1}^{N} &= \text{Norm}(\{\lambda_{n,m}\}_{n=1}^{N}), \end{aligned} \tag{2}$$

where $\mathbf{W}_0^T$, b_0, $\mathbf{W}_1^T$ and b_1 are learnable parameters, and the **Norm** module is used to constrain the scale of the attention map.

Intermediate node. As an intermediate node, v_m is connected to other nodes that modify it, and such connections are actually a subset of edges, $\mathcal{E}_m \in \mathcal{E}$, incident to v_m. We compute an attention map over the edges of image semantic graph $\mathcal{G}^o$ for each edge in this subset, then transfer and combine all these attention maps to obtain a final attention map for node v_m.

For each edge $e_k = (v_{ks}, r_k, v_{ko})$ in $\mathcal{E}_m$ (where v_{ks} is exactly v_m), we first form a sentence associated with e_k by concatenating the words or phrases associated with v_{ks}, r_k and v_{ko}. Then, we obtain the embedded feature vectors $\{\mathbf{f}_t\}_{t=1}^T$ and word contexts $\{\mathbf{h}_t\}_{t=1}^T$ for the words $\{w_t\}_{t=1}^T$ in this sentence and the feature representation of the whole sentence by following the same computation for leaf nodes. Next, we compute the attention map for node v_{ks} from two different aspects, *i.e*. subject description and relation-based transfer, because e_k not only directly describes subject v_{ks} itself but also its relation to object v_{ko}. From the aspect of subject description, same as the computation for leaf nodes, we obtain attention maps corresponding to the appearance and spatial location of v_{ks} (*i.e*. $\{\lambda_{n,k_s}^{look}\}_{n=1}^N$ and $\{\lambda_{n,k_s}^{loc}\}_{n=1}^N$) and weights (*i.e*. $\beta_{k_s}^{look}$ and $\beta_{k_s}^{loc}$) to combine them. From the aspect of relation-based transfer, we first compute a relational feature representation for edge e_k as follows,

$$\alpha_{t,k}^{rel} = \frac{\exp(\mathbf{W}_{rel}^T \mathbf{h}_t)}{\sum_{t=1}^T \exp(\mathbf{W}_{rel}^T \mathbf{h}_t)}, \mathbf{r}_k = \sum_{t=1}^T \alpha_{t,k}^{rel} \mathbf{f}_t \quad (3)$$

where $\mathbf{W}_{rel}$ is a learnable parameter. Then we feed the relational representation $\mathbf{r}_k$ to the **AttendRelation** neural module to attend the relation $\mathbf{r}_k$ over the edges $\mathcal{E}_{ij}^o$ of graph $\mathcal{G}^o$, and the computed attention weights are denoted as $\{\gamma_{ij,k}\}_{i,j=1}^N$. Moreover, we use the **Transfer** module and the **Norm** module to transfer the attention map $\{\lambda_{n,k_o}\}_{n=1}^N$ for object node v_{ko} to node v_m by modulating $\{\lambda_{n,k_o}\}_{n=1}^N$ with the attention weights on edges $\{\gamma_{ij,k}\}_{i,j=1}^N$, and the transferred attention map for node v_m is denoted as $\{\lambda_{n,k_s}^{rel}\}_{n=1}^N$. It is worth mentioning that object node v_{ko} has been accessed before and the attention map $\{\lambda_{n,k_o}\}_{n=1}^N$ for node v_{ko} has been computed. Next, we estimate the weight of relation at edge e_k and integrate the attention maps for node v_{ks} related to subject description and relation-based transfer to obtain attention map $\{\lambda_{n,k_s}\}_{n=1}^N$ for node v_{ks} contributed by edge e_k, and $\{\lambda_{n,k_s}\}_{n=1}^N$ is defined as follows.

$$\begin{aligned} \beta_k^{rel} &= \text{sigmoid}(\mathbf{W}_2^T \mathbf{h} + b_2) \\ \lambda_{n,k_s} &= \beta_{k_s}^{look} \lambda_{n,k_s}^{look} + \beta_{k_s}^{loc} \lambda_{n,k_s}^{loc} + \beta_{n,k_s}^{rel} \lambda_{n,k_s}^{rel} \\ \{\lambda_{n,k_s}\}_{n=1}^N &= \text{Norm}(\{\lambda_{n,k_s}\}_{n=1}^N), \end{aligned} \quad (4)$$

where $\mathbf{W}_2$ and b_2 are learnable parameters.

Finally, we combine the attention maps $\{\{\lambda_{n,k_s}\}_{n=1}^N\}$ for node v_m contributed by all edges in $\mathcal{E}_m$ using the **Merge** module followed by the **Norm** module to obtain the final attention map $\{\lambda_{n,m}\}_{n=1}^N$ for node v_m.

3.2.2 Neural Modules

We present a series of neural modules to perform specific reasoning steps, inspired by the neural modules in [22]. In particular, the AttendNode and AttendRelation modules are used to connect the language mode with the vision mode. They receive feature representations of linguistic contents from the language scene graph and output attention maps of the features defined over visual contents in the image semantic graph. The Merge, Norm and Transfer modules are adopted to further integrate and transfer attention maps over the nodes and edges of the image semantic graph.

AttendNode [appearance query, location query] module aims to find relevant nodes among the nodes of the image semantic graph $\mathcal{G}^o$ given an appearance query and location query. It takes the query vectors of the appearance query and location query as inputs and generate attention maps $\{\lambda_n^{look}\}_{n=1}^N$ and $\{\lambda_n^{loc}\}_{n=1}^N$ over the nodes $\mathcal{V}^o$, where every node $v_n^o \in \mathcal{V}^o$ has two attention weights, *i.e*., $\lambda_n^{look} \in [-1, 1]$ and $\lambda_n^{loc} \in [-1, 1]$. The query vectors are linguistic features at nodes of the language scene graph, denoted as $\mathbf{v}^{look}$ and $\mathbf{v}^{loc}$. For node v_n^o in graph $\mathcal{G}^o$, its attention weights λ_n^{look} and λ_n^{loc} are defined as follows,

$$\begin{aligned} \lambda_n^{look} &= \langle \text{L2Norm}(\text{MLP}_0(\mathbf{v}_n^o)), \text{L2Norm}(\text{MLP}_1(\mathbf{v}^{look})) \rangle, \\ \lambda_n^{loc} &= \langle \text{L2Norm}(\text{MLP}_2(\mathbf{p}_n^o)), \text{L2Norm}(\text{MLP}_3(\mathbf{v}^{loc})) \rangle, \end{aligned} \quad (5)$$

where $\text{MLP}_0()$, $\text{MLP}_1()$, $\text{MLP}_2()$ and $\text{MLP}_3()$ are multilayer perceptrons consisting of several linear and ReLU layers, L2Norm() is the L2 normalization, and $\mathbf{v}_n^o$ and $\mathbf{p}_n^o$ are the visual feature and spatial feature at node v_n^o respectively, which are mentioned in Section 3.1.1.

AttendRelation [relation query] module aims to find relevant edges in the image semantic graph $\mathcal{G}^o$ given a relation query. The purpose of a relation query is to establish connections between nodes in graph $\mathcal{G}^o$. Given query vector $\mathbf{e}$, the attention weights $\{\gamma_{ij}\}_{i,j=1}^N$ on edges $\{\mathbf{e}_{ij}^o\}_{i,j=1}^N$ are defined as follows,

$$\gamma_{ij} = \sigma(\langle \text{L2Norm}(\text{MLP}_5(\mathbf{e}_{ij}^o)), \text{L2Norm}(\text{MLP}_1(\mathbf{e})) \rangle) \quad (6)$$

where $\text{MLP}_5()$, $\text{MLP}_5()$ are multilayer perceptrons, and the ReLU activation function σ ensures the attention weights are larger than zero.

Transfer module aims to find new nodes by passing attention weights $\{\lambda_n\}_{n=1}^N$ on nodes that modify those new nodes along attended edges $\{\gamma_{ij}\}_{i,j=1}^N$. The updated attention weights $\{\lambda_n^{new}\}_{n=1}^N$ are calculated as follows,

$$\lambda_n^{new} = \sum_{j=1}^N \gamma_{n,j} \lambda_j. \quad (7)$$

Merge module aims to combine multiple attention maps generated from different edges of the same node, where the attention weights over edges are computed individually. Given the set of attention maps Λ for a node, the merged attention map $\{\lambda_n\}_{n=1}^N$ is defined as follows,

$$\lambda_n = \sum_{\{\lambda'_n\}_{n=1}^N \in \Lambda} \lambda'_n. \quad (8)$$

Norm module aims to set the range of weights in attention maps to $[-1, 1]$. If the maximum absolute value of an attention map is larger than 1, the attention map is divided by the maximum absolute value.

3.3. Loss Function

Once all the nodes in the stack have been processed, the final attention map for the referent node of the language scene graph is obtained. This attention map is denoted as $\{\lambda_{n,ref}\}_{n=1}^N$. As in previous methods for grounding referring expressions [9], during the training phase, we adopt the cross-entropy loss, which is defined as

$$p_i = \exp(\lambda_{i,ref}) / \sum_{n=1}^{N} \exp(\lambda_{n,ref}), \text{loss} = -\log(p_{gt}) \quad (9)$$

where p_{gt} is the probability of the ground-truth object. During the inference phase, we predict the referent by choosing the object with the highest probability.

4. Ref-Reasoning Dataset

The proposed dataset is built on the scenes from the GQA dataset [10]. We automatically generate referring expressions for every image on the basis of the image scene graph using a diverse set of expression templates.

4.1. Preparation

Scene Graph. We generate referring expressions according to the ground-truth image scene graphs. Specifically, we adopt the scene graph annotations provided by the Visual Genome dataset [14] and further normalized by the GQA dataset. In a scene graph annotation of an image, each node represents an object with about 1-3 attributes, and each edge represents a relation (*i.e.*, semantic relation, spatial relation and comparatives) between two objects. In order to use the scene graphs for referring expression generation, we remove some unnatural edges and classes, *e.g.*, "nose left of eyes". In addition, we add edges between objects to represent same-attribute relations between objects, *i.e.*, "same material", "same color" and "same shape". In total, there are 1,664 object classes, 308 relation classes and 610 attribute classes in the adopted scene graphs.

Expression Template. In order to generate referring expressions with diverse reasoning layouts, for each specified number of nodes, we design a family of referring expression templates for each reasoning layout. We generate expressions according to layouts and templates using functional programs, and the functional program for each template can be easily obtained according to the layout. In particular, layouts are sub-graphs of directed acyclic graphs, where only one node (*i.e.*, the root node) has zero out-degree and other nodes can reach the root node. The functional program for a layout provides a step-wise plan for reaching the root node from leaf nodes (*i.e.*, the nodes with zero in-degree) by traversing all the nodes and edges in this layout, and templates are parameterized natural language expressions, where the parameters can be filled in. Moreover, we set the constraint that the number of nodes in a template ranges from one to five.

4.2. Generation Process

Given an image, we generate dozens of expressions from the scene graph of the image, and the generation process for one expression is summarized as follows,

- Randomly sample the referent node and randomly decide the number of nodes, denoted as C.
- Randomly sample a sub-graph with C nodes including the referent node in the scene graph.
- Judge the layout of the sub-graph and randomly sample a referring expression template from the family of templates corresponding to the layout.
- Fill in the parameters in the template using contents of the sub-graph, including relations and objects with randomly sampled attributes.
- Execute the functional program with filled parameters and accept the expression if the referred object is unique in the scene graph.

Note that we perform extra operations during the generation process: 1) If there are objects that have same-attribute relations in the sub-graph, we avoid choosing the attributes that appear in such relations for these objects. This restriction intends to make the modified node identified by the relation edge instead of the attribute directly. 2) To help balance the dataset, during the process of random sampling, we decrease the chances of nodes and relations whose classes most commonly exist in the scene graphs. In addition, we increase the chances of multi-order relationships with $C = 3$ or $C = 4$ to reasonably increase the level of difficulty for reasoning. 3) We define a difficulty level for a referring expression. We find its shortest sub-expression which can identify the referent in the scene graph, and the number of objects in the sub-expression is defined as the difficulty level. For example, if there is only one bottle in an image, the difficulty level of "the bottle on a table beside

	Number of Objects				Split	
	one	two	three	>= four	val	test
CNN	10.57	13.11	14.21	11.32	12.36	12.15
CNN+LSTM	75.29	51.85	46.26	32.45	42.38	42.43
DGA	73.14	54.63	48.48	37.63	45.37	45.87
CMRIN	79.20	56.87	50.07	35.29	45.43	45.87
Ours SGMN	**79.71**	**61.77**	**55.57**	**41.89**	**51.04**	**51.39**

Table 1. Comparison with baselines and state-of-the-art methods on Ref-Reasoning dataset. The best performing method is marked in bold.

a plate" is still one even though it describes three objects and their relations. Then, we obtain the balanced dataset and its final splits by randomly sampling expressions of images according to their difficulty level and the number of nodes described by them.

5. Experiments

5.1. Datasets

We have conducted extensive experiments on the proposed Ref-Reasoning dataset as well as on three commonly used benchmark datasets (*i.e.*, RefCOCO[27], RefCOCO+[27] and RefCOCOg[19]). Ref-Reasoning contains 791,956 referring expressions in 83,989 images. It has 721,164, 36,183 and 34,609 expression-referent pairs for training, validation and testing, respectively. Ref-Reasoning includes semantically rich expressions describing objects, attributes, direct relations and indirect relations with different layouts. RefCOCO and RefCOCO+ datasets includes short expressions collected from an interactive game interface. RefCOCOg collects from a non-interactive settings and it has longer complex expressions.

5.2. Implementation and Evaluation

The performance of grounding referring expressions is evaluated by accuracy, *i.e.*, the fraction of correct predictions of referents.

For the Ref-Reasoning dataset, we use a ResNet-101 based Faster R-CNN [20, 6] as the backbone, and adopt a feature extractor which is trained on the training set of GQA with an extra attribute loss following [1]. Visual features of annotated objects are extracted from the pool5 layer of the feature extractor. For the three common benchmark datasets (*i.e.*, RefCOCO, RefCOCO+ and RefCOCOg), we follow CMRIN [23] to extract the visual features of objects in images. To keep the image semantic graph sparse and reduce computational cost, we connect each node in the image semantic graph to its five nearest nodes based on the distances between their normalized center coordinates. We set the mini-batch size to 64. All the models are trained by the Adam optimizer [13] with the learning rate set to 0.0001 and 0.0005 for the Ref-Reasoning dataset and other benchmark datasets respectively.

5.3. Comparison with the State of the Art

We conduct experimental comparisons between the proposed SGMN and existing state-of-the-art methods on both the collected Ref-Reasoning dataset and three commonly used benchmark datasets.

Ref-Reasoning Dataset. We evaluate two baselines (*i.e.*, a CNN model and a CNN+LSTM model), two state-of-the-art methods (*i.e.*, CMRIN [23] and DGA [24]) and the proposed SGMN on the Ref-Reasoning dataset. The CNN model is allowed to access objects and images only. The CNN+LSTM model embeds objects and expressions into a common feature space and learns matching scores between them. For CMRIN and DGA, we adopt their default settings [23, 24] in our evaluation. For a fair comparison, all the models use the same visual object features and the same setting in LSTMs.

Table 1 shows the evaluation results on the Ref-Reasoning dataset. The proposed SGMN significantly outperforms the baselines and existing state-of-the-art models, and it consistently achieves the best performance on all the splits of the testing set, where different splits need different numbers of reasoning steps. The CNN model has a low accuracy of 12.15%, which is much lower than the accuracy (*i.e.*, 41.1% [4]) of the image-only model for the RefCOCOg dataset, which demonstrates that joint understanding of images and text is required on Ref-Reasoning. The CNN+LSTM model achieves a high accuracy of 75.29% on the split where expressions directly describe the referents. This is because relation reasoning is not required in this split and LSTM may be qualified to capture the semantics of expressions. Compared with the CNN+LSTM model, DGA and CMRIN achieve higher performance on the two-, three- and four-node splits because they learn a language-guided contextual representation for objects.

Common Benchmark Datasets. Quantitative evaluation results on RefCOCO, RefCOCO+ and RefCOCOg datasets are shown in Table 2. The proposed SGMN consistently outperforms existing structured methods across all the datasets, and it improves the average accuracy over the testing sets achieved by the best performing existing structured method by 0.92%, 2.54% and 2.96% respectively on the RefCOCO, RefCOCO+ and RefCOCOg datasets. Moreover, it also surpasses all the existing models on the

	RefCOCO		RefCOCO+		RefCOCOg
	testA	testB	testA	testB	test
Holistic Models					
CMN [9]	75.94	79.57	59.29	59.34	-
ParallelAttn [29]	80.81	81.32	66.31	61.46	-
MAttNet* [26]	85.26	84.57	75.13	66.17	78.12
CMRIN* [23]	**87.63**	84.73	**80.93**	68.99	80.66
DGA* [24]	86.64	84.79	78.31	68.15	80.26
Structured Models					
MattNet* + parser [26]	79.71	81.22	68.30	62.94	73.72
RvG-Tree* [8]	82.52	82.90	70.21	65.49	75.20
DGA* + parser [24]	84.69	83.69	74.83	65.43	76.33
NMTree* [15]	85.63	85.08	75.74	67.62	78.21
MSGL* [16]	85.45	85.12	75.31	67.50	78.46
Ours SGMN*	86.67	**85.36**	78.66	**69.77**	**81.42**

Table 2. Comparison with state-of-the-art methods on RefCOCO, RefCOCO+ and RefCOCOg. We use * to indicate this model uses resnet101 features. None-superscript indicates that model uses vgg16 features. The best performing method is marked in bold.

RefCOCOg dataset which has relatively longer complex expressions with an average length 8.43, and achieves a performance comparable to the best performing holistic method on the other two common benchmark datasets. Note that holistic models usually have higher performance than structured models on the common benchmark datasets because those datasets include many simple expressions describing the referents without relations, and holistic models are prone to learn shallow correlations without reasoning and may exploit this dataset bias [4, 17]. In addition, the inference mechanism of holistic methods has poor interpretability.

5.4. Qualitative Evaluation

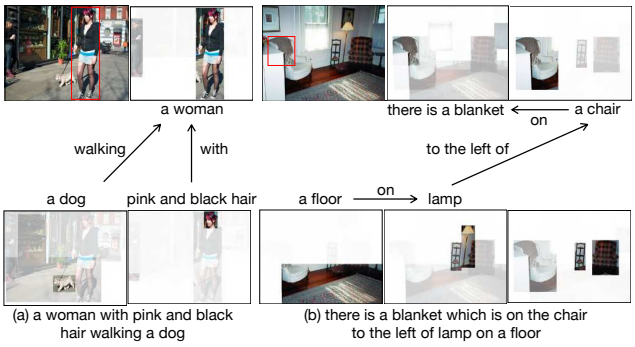

Figure 3. Qualitative results showing the attention maps over the objects along the language scene graphs predicted by the SGMN.

Visualizations of two examples along with their language scene graphs and attention maps over the objects in images at every node of the language scene graphs are shown in Figure 3. This qualitative evaluation results demonstrate that the proposed SGMN can generate interpretable visual evidences of intermediate steps in the reasoning process. In Figure 3(a), SGMN parses the expression into a tree structure and finds the referred "woman" who is walking "a dog" and meanwhile is with "pink and black hair". Figure 3(b) shows a more complex expression which describes four objects and their relations. SGMN first successfully changes from the initial attention map (bottom-right) to the final attention map (top-right) at the node "a chair" by performing relational reasoning along the edges (*i.e.*, triplets ("a chair", "to the left of" , "lamp") and ("lamp", "on", "a floor")), and then identifies the target "blanket" on that chair.

5.5. Ablation Study

	Number of Objects				Split	
	one	two	three	>= four	val	test
w/o transfer	79.14	48.51	45.97	31.57	40.66	41.88
w/o norm	79.37	49.44	45.61	31.57	40.80	41.93
max merge	78.71	54.00	50.34	34.76	44.50	45.27
min merge	78.83	53.83	51.11	35.79	45.25	46.00
Ours SGMN	**79.71**	**61.77**	**55.57**	**41.89**	**51.04**	**51.39**

Table 3. Ablation study on Ref-Reasoning dataset. The best performing method is marked in bold.

To demonstrate the effectiveness of reasoning under the guidance of scene graphs inferred from referring expressions as well as the design of neural modules, we train four additional models for comparison. The results are shown in Table 3. All the models have similar performance on the split of expressions directly describing the referents. For the other splits, SGMN without the Transfer module and SGMN without the Norm module have much lower performance than the original SGMN because the former treats the referent as an isolated node without performing relation reasoning while the latter unfairly treats different relational edges and the nodes connected by them. Next, we explore different options of the function (*i.e.*, max, min and sum) used in the Merge module. Compared to SGMN with sum-merge, its performance with min-merge and max-merge drops because max-merge only captures the most significant relation for each intermediate node and min-merge is sensitive to parsing errors and recognition errors.

6. Conclusion

In this paper, we present a scene graph guided modular network (SGMN) for grounding referring expressions. It performs graph-structured reasoning over the constructed graph representations of the input image and expression using neural modules. In addition, we propose a large-scale real-world dataset for structured referring expression reasoning, named Ref-Reasoning. Experimental results demonstrate that SGMN not only significantly outperforms existing state-of-the-art algorithms on the new Ref-Reasoning dataset, but also surpasses state-of-the-art structured methods on commonly used benchmark datasets. Moreover, it can generate interpretable visual evidences of reasoning via a graph attention mechanism.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 7

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016. 2

[3] Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[4] Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. Visual referring expression recognition: What do systems actually learn? *arXiv preprint arXiv:1805.11818*, 2018. 2, 7, 8

[5] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. 7

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[8] R. Hong, D. Liu, X. Mo, X. He, and H. Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 1, 8

[9] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1115–1124, 2017. 2, 4, 6, 8

[10] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019. 2, 3, 6

[11] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 3

[12] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 7

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 2, 3, 6

[15] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Wu Feng. Learning to assemble neural module tree networks for visual grounding. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 8

[16] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Fanglin Wang. Referring expression grounding by marginalizing scene graph likelihood, 2019. 2, 8

[17] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4185–4194, 2019. 2, 3, 8

[18] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7102–7111, 2017. 1

[19] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 2, 7

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 7

[21] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 2, 3

[22] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8376–8384, 2019. 2, 5

[23] Sibei Yang, Guanbin Li, and Yizhou Yu. Cross-modal relationship inference for grounding referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4145–4154, 2019. 1, 2, 4, 7, 8

[24] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 7, 8

[25] Sibei Yang, Guanbin Li, and Yizhou Yu. Relationship-embedded representation learning for grounding referring expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[26] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 1, 2, 4, 8

[27] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 2, 7

[28] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4158–4166, 2018. 2

[29] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton van den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4252–4261, 2018. 1, 2, 8