Language-Conditioned Graph Networks for Relational Reasoning

Ronghang Hu[1] Anna Rohrbach[1] Trevor Darrell[1] Kate Saenko[2]
[1]University of California, Berkeley [2]Boston University

Abstract

Solving grounded language tasks often requires reasoning about relationships between objects in the context of a given task. For example, to answer the question "What color is the mug on the plate?" we must check the color of the specific mug that satisfies the "on" relationship with respect to the plate. Recent work has proposed various methods capable of complex relational reasoning. However, most of their power is in the inference structure, while the scene is represented with simple local appearance features. In this paper, we take an alternate approach and build contextualized representations for objects in a visual scene to support relational reasoning. We propose a general framework of Language-Conditioned Graph Networks (LCGN), where each node represents an object, and is described by a context-aware representation from related objects through iterative message passing conditioned on the textual input. E.g., conditioning on the "on" relationship to the plate, the object "mug" gathers messages from the object "plate" to update its representation to "mug on the plate", which can be easily consumed by a simple classifier for answer prediction. We experimentally show that our LCGN approach effectively supports relational reasoning and improves performance across several tasks and datasets. Our code is available at http://ronghanghu.com/lcgn.

1. Introduction

Grounded language comprehension tasks, such as visual question answering (VQA) or referring expression comprehension (REF), require finding the relevant objects in the scene and reasoning about certain relationships between them. For example in Figure 1, to answer the question *is there a person to the left of the woman holding a blue umbrella*, we must locate the relevant objects – *person*, *woman* and *blue umbrella* – and model the specified relationships – *to the left of* and *holding*.

How should we build a model to perform reasoning in grounded language comprehension tasks? Prior works have explored various approaches from learning joint visual-textual representations (*e.g.* [8, 31]) to pooling over pairwise relationships (*e.g.* [35, 43]) or constructing explicit reasoning steps with modular or symbolic representations (*e.g.* [2, 42]). Although these models are capable of performing complex relational inference, their scene representations are built upon local visual appearance features that do not contain much contextual information. Instead, they tend to rely heavily on manually designed inference structures or modules to perform reasoning about relationships, and are often specific to a particular task.

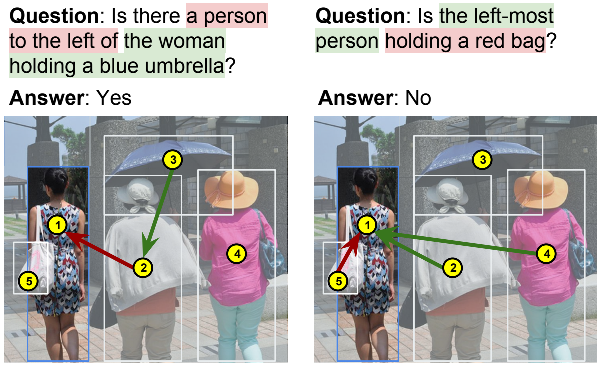
Question: Is there a person to the left of the woman holding a blue umbrella?
Answer: Yes
Question: Is the left-most person holding a red bag?
Answer: No
1
2
3
4
5
1
2
3
4
5

Figure 1. In this work, we create context-aware representations for objects by sending messages between relevant objects in a dynamic way that depends on the input language. In the left example, the first round of message passing updates object 2 with features of object 3 based on *the woman holding a blue umbrella* (green arrow), and the second round updates object 1 with object 2's features based on *person to the left* (red arrow). The final answer prediction can be made by a single attention hop over the most relevant object (blue box).

In this work, we propose an alternative way to facilitate reasoning with a context-aware scene representation, suitable for multiple tasks. Our proposed Language-Conditioned Graph Network (LCGN) model augments the local appearance feature of each entity in the scene with a relational contextualized feature. Our model is a graph network built upon visual entities in the scene, which collects relational information through multiple iterations of message passing between the entities. It dynamically determines which objects to collect information from on each round, by weighting the edges in the graph, and sends messages through the graph to propagate just the right amount of relational information. The key idea is to condition the message passing on the specific contextual relationships described in the input text. Figure 1 illustrates this process,

2380-7504/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCV.2019.01039

where the *person* would be represented not only by her local appearance, but also by contextualized features indicating her relationship to other relevant objects in the scene, e.g., *left of a woman*. Our contextualized representation can be easily plugged into task-specific models to replace standard local appearance features, facilitating reasoning with rich relational information. E.g. for the question answering task, it is sufficient to perform a single attention hop over the relevant object, whose representation is contextualized (e.g. blue box in Figure 1).

Importantly, our scene representation is constructed with respect to the given reasoning task. An object in the scene may be involved in multiple relations in different contexts: in Figure 1, the person can be simultaneously *left of a woman holding a blue umbrella*, *holding a white bag*, and *standing on a sidewalk*. Rather than building a complete representation of all the first- and higher-order relational information for each object (which can be enormous and unnecessary), we focus the contextual representation on relational information that is helpful to the reasoning task by conditioning on the input text (Figure 1 left vs. right).

We apply our Language-Conditioned Graph Networks to two reasoning tasks with language inputs—Visual Question Answering (VQA) and Referring Expression Comprehension (REF). In these tasks, we replace the local appearance-based visual representations with the context-aware representations from our LCGN model, and demonstrate that our context-aware scene representations can be used as inputs to perform complex reasoning via simple task-specific approaches, with a consistent improvement over the local appearance features across different tasks and datasets. We obtain state-of-the-art results on the GQA dataset [19] for VQA and the CLEVR-Ref+ dataset [25] for REF.

2. Related work

We first provide an overview of the reasoning tasks addressed in this paper. Then we review related work on graph networks and other contextualized representations. Finally, we discuss alternative approaches to reasoning problems.

Visual question answering (VQA) and referring expression comprehension (REF) VQA and REF are two popular tasks that require reasoning about image content. While in VQA the goal is to answer a question about an image [3], in REF one has to localize an image region that corresponds to a referring expression [28]. While the real-world VQA dataset [3, 12] focuses more on perception than complex reasoning, the more recent synthetic CLEVR [20] dataset is a standard benchmark for relational reasoning. An even more recent GQA dataset [19] brings together the best of both worlds: real images and relational questions. It is built upon the Visual Genome dataset [22] and construct the balanced question-answer pairs from scene graphs.

For REF, there are a number of standard benchmarks such as RefCOCO [44] and RefCOCOg [28], with natural language referring expressions and images from the COCO dataset [24]. However, many of the expressions in these datasets do not require resolving relations. Recently, a new CLEVR-Ref+ dataset [25] has been proposed for REF. It is built using the CLEVR environment and involves very complex queries, aiming to assess the reasoning capabilities of existing models and find their limitations.

In this work we tackle both VQA and REF tasks on three datasets in total. Notably, in all cases, we use the same approach, Language-Conditioned Graph Network (LCGN), to build contextualized representations of objects/image regions. This shows the generality and effectiveness of our approach for various visual reasoning tasks.

Graph networks and contextualized representations Graph networks are powerful models that can perform relational inference through message passing [4, 11, 21, 23, 38, 46]. The core idea is to enable communication between image regions to build contextualized representations of these regions. Graph networks have been successfully applied to various tasks, from object detection [26] and region classification [7] to human-object interaction [32] and activity recognition [14]. Besides, self-attention models [37] and non-local networks [40] can also be cast as graph networks in a general sense. Below we review some of the recent works that rely on graph networks and other contextualized representations for VQA and REF.

A prominent work that introduced relational reasoning in VQA is [35], which proposes Relation Networks (RNs) for modeling relations between all pairs of objects, conditioned on a question. [6] extends RNs with the Broadcasting Convolutional Network module, which globally broadcasts objects' visuo-spatial features. The first work to use graph networks in VQA is [36], which combines dependency parses of questions and scene graph representations of abstract scenes. [47] proposes modeling structured visual attention over a Conditional Random Field on image regions. A recent work, [29], conditions on a question to learn a graph representation of an image, capturing object interactions with the relevant neighbours via spatial graph convolutions. Later, [5] extends this idea to modeling spatial-semantic pairwise relations between all pairs of regions.

For the REF task, [39] proposes Language-guided Graph Attention Networks, where attention over nodes and edges is guided by a referring expression, which is decomposed into subject, intra-class and inter-class relationships.

Our work is related to, yet distinct from, the approaches above. While [29] predicts a sparsely connected graph (conditioned on the question) that remains fixed for each step of graph convolution, our LCGN model predicts dynamic edge weights to focus on different connections in each message passing iteration. Besides, [29] is tailored to VQA and is non-trivial to adapt to REF (since it includes max-pooling over node representations). Compared to [5], instead of max-pooling over explicitly constructed pairwise vectors, our model predicts normalized edge weights that both improve computation efficiency in message passing and make it easier to visualize and inspect connections. Finally, [39] is tailored to REF by modeling specific subject attention and inter- and intra-class relations, and does not gather higher-order relational information in an iterative manner. We propose a more general approach for scene representation that is applicable to both VQA and REF.

Reasoning models A multitude of approaches have been recently proposed to tackle visual reasoning tasks, such as VQA and REF. Neural Module Networks (NMNs) [2, 16] are multi-step models that build question-specific layouts and execute them against an image. NMNs have also been applied to REF, *e.g.* CMN [17] and Stack-NMN [15]. MAC [18] performs multi-step reasoning while recording information in its memory. FiLM [31] is an approach which modulates image representation with the given question via conditional batch normalization, and is extended in [41] with a multi-step reasoning procedure where both modalities can modulate each other. QGHC [10] predicts question-dependent convolution kernels to modulate visual features. DFAF [9] introduces self-attention and co-attention mechanisms between visual features and question words, allowing information to flow across modalities. The Neural-Symbolic approach [42] disentangles reasoning from image and language understanding, by first extracting symbolic representations from images and text, and then executing symbolic programs over them. MAttNet [43], a state-of-the-art approach to REF, uses attention to parse an expression and ground it through several modules.

Our approach is not meant to substitute the aforementioned reasoning models, but to complement them. Our contextualized visual representation can be combined with other reasoning models to replace the local feature representation. A prominent reasoning model capable of addressing both VQA and REF is Stack-NMN [15], and we empirically compare to it in Section 4.

3. Language-Conditioned Graph Networks

Given a visual scene and a textual input for a reasoning task such as VQA or REF, we propose to construct a contextualized representation for each entity in the scene that contains the relational information needed for the reasoning procedure specified in the language input.

This contextualized representation is obtained in our novel Language-Conditioned Graph Networks (LCGN) model, through iterative message passing conditioned on the language input. It can be then used as input to a task-specific output module such as a single-hop VQA classifier.

3.1. Context-aware scene representation

For an image I and a textual input Q that represents a reasoning task, let N be the number of entities in the scene, where each entity can be a detected object or a spatial location on the convolutional feature map of the image. Let x_i^{loc} (where $i = 1, ..., N$) be the local feature representation of the i-th entity, *i.e.* the i-th detected object's visual feature or the convolutional feature at the i-th location on the feature grid. We would like to output a context-aware representation x_i^{out} for each entity i conditioned on the textual input Q that contains the relational context associated with entity i. This is obtained through iterative message passing over T iterations with our Language-Conditioned Graph Networks, as shown in Figure 2.

We use a fully-connected graph over the scene, where each node corresponds to an entity i as defined above, and there is a directed edge $i \rightarrow j$ between every pair of entities i and j. Each node i is represented by a local feature x_i^{loc} that is fixed during message passing, and a context feature $x_{i,t}^{ctx}$ that is updated during each iteration t. A learned parameter is used as the initial context representation $x_{i,0}^{ctx}$ at $t = 0$ for all nodes, before the message passing starts.

Textual command extraction To incorporate the textual input in the iterative message passing, we build a textual command vector for each iteration t (where $t = 1, ..., T$). Given a textual input Q for the reasoning task, such as a question in VQA or a query in REF, we extract a set of vectors $\{c_t\}$ from the text Q, using the same multi-step textual attention mechanism as in Stack-NMN [15] and MAC [18]. Specifically, Q is encoded into a sequence $\{h_s\}_{s=1}^S$ and a summary vector q with a bi-directional LSTM as:

$$[h_1, h_2, ..., h_S] = \text{BiLSTM}(Q) \quad \text{and} \quad q = [h_1; h_S] \quad (1)$$

where S is the number of words in Q, and $h_s = [\overrightarrow{h_s}; \overleftarrow{h_s}]$ is the concatenation of the forward and backward hidden states for word s from the bi-directional LSTM output. At each iteration t, a textual attention $\alpha_{t,s}$ is computed over the words, and the textual command c_t is obtained from the textual attention as follows:

$$\alpha_{t,s} = \underset{s}{\text{Softmax}}\left(W_1\left(h_s \odot \left(W_2^{(t)}\,\text{ReLU}\left(W_3 q\right)\right)\right)\right) \quad (2)$$

$$c_t = \sum_{s=1}^{S} \alpha_{t,s} \cdot h_s \quad (3)$$

where $\odot$ is element-wise multiplication. Each c_t can be seen as a textual command supplied during the t-th iteration. Unlike all other parameters that are shared across iterations, here $W_2^{(t)}$ is learned separately for each iteration t.

Language-conditioned message passing At the t-th iteration where $t = 1, ..., T$, we first build a joint representation of each entity. Then, we compute the (directed) connection

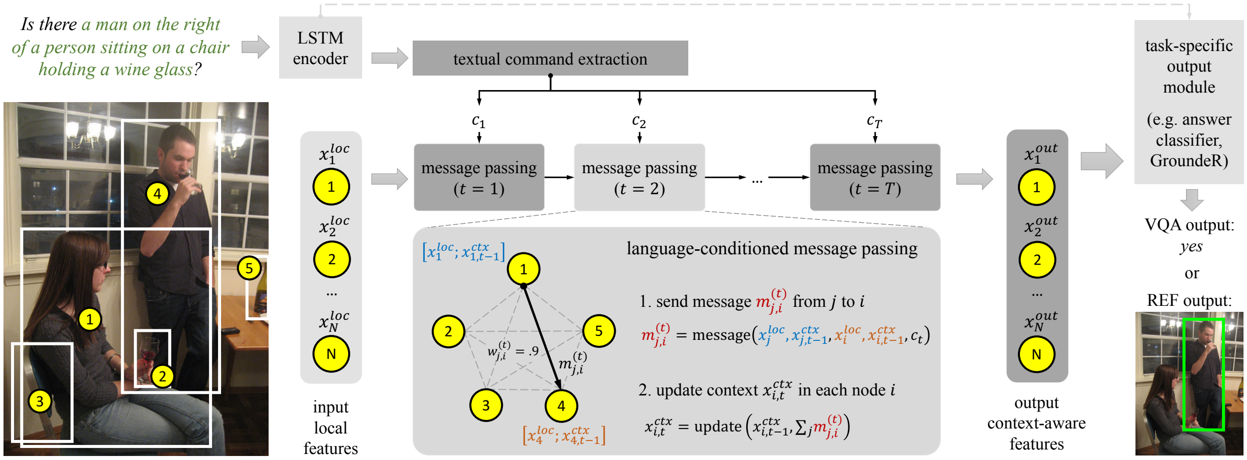
Is there a man on the right of a person sitting on a chair holding a wine glass?
LSTM encoder
textual command extraction
c_1
c_2
c_T
task-specific output module
(e.g. answer classifier, GroundeR)
x_1^{loc}
x_2^{loc}
x_N^{loc}
input local features
message passing ($t = 1$)
message passing ($t = 2$)
...
message passing ($t = T$)
language-conditioned message passing
$[x_1^{loc}; x_{1,t-1}^{ctx}]$
$w_{j,i}^{(t)} = .9$
$m_{j,i}^{(t)}$
$[x_4^{loc}; x_{4,t-1}^{ctx}]$
1. send message $m_{j,i}^{(t)}$ from j to i
$m_{j,i}^{(t)} = \text{message}(x_j^{loc}, x_{j,t-1}^{ctx}, x_i^{loc}, x_{i,t-1}^{ctx}, c_t)$
2. update context $x_{i,t}^{ctx}$ in each node i
$x_{i,t}^{ctx} = \text{update}\left(x_{i,t-1}^{ctx}, \Sigma_j m_{j,i}^{(t)}\right)$
x_1^{out}
x_2^{out}
x_N^{out}
output context-aware features
VQA output: yes
or
REF output:
1
2
3
4
5
N

Figure 2. We propose Language-Conditioned Graph Networks (LCGN) to address reasoning tasks such as VQA and REF. Our model constructs a context-aware representation x_i^{out} for each object i through iterative message passing conditioned on the input text. During message passing, each object i is represented by a local feature x_i^{loc} and a context feature $x_{i,t}^{ctx}$. In every iteration, each object j sends a message vector $m_{j,i}^{(t)}$ to each object i, which is collected by i to update its context feature $x_{i,t}^{ctx}$. The local feature x_i^{loc} and the final context feature $x_{i,T}^{ctx}$ are combined into a joint context-aware feature x_i^{out}, which is used in simple task-specific output modules for VQA or REF.

weights $w_{j,i}^{(t)}$ from every entity j (the sender, $j = 1, ..., N$) to every entity i (the receiver, $i = 1, ..., N$). Finally, each entity j sends a message vector $m_{j,i}^{(t)}$ to each entity i, and each entity i sums up all of its incoming messages $m_{j,i}^{(t)}$ to update its contextual representation from $x_{i,t-1}^{ctx}$ to $x_{i,t}^{ctx}$ as described below.

Step 1. We build a joint representation $\tilde{x}_{i,t}$ for each node, by concatenating x_i^{loc} and $x_{i,t-1}^{ctx}$ and their element-wise product (after linear mapping) as

$$\tilde{x}_{i,t} = \left[x_i^{loc}; x_{i,t-1}^{ctx}; \left(W_4 x_i^{loc}\right) \odot \left(W_5 x_{i,t-1}^{ctx}\right)\right] \qquad (4)$$

Step 2. We compute the directed connection weights $w_{j,i}^{(t)}$ from node j (the sender) to node i (the receiver), conditioning on the textual command c_t at iteration t. Here, the connection weights are normalized with a softmax function over j, so that the sender weights sum up to 1 for each receiver, *i.e.* $\sum_{j=1}^{N} w_{j,i}^{(t)} = 1$ for all $i = 1, ..., N$ as follows:

$$w_{j,i}^{(t)} = \operatorname*{Softmax}_{j}\left(\left(W_6 \tilde{x}_{i,t}\right)^T \left(\left(W_7 \tilde{x}_{j,t}\right) \odot \left(W_8 c_t\right)\right)\right) \qquad (5)$$

Step 3. Each node j sends a message $m_{j,i}^{(t)}$ to each node i conditioning on the textual input c_t and weighted by the connection weight $w_{j,i}^{(t)}$. Then, each node i sums up the incoming messages and updates its context representation:

$$m_{j,i}^{(t)} = w_{j,i}^{(t)} \cdot \left(\left(W_9 \tilde{x}_{j,t}\right) \odot \left(W_{10} c_t\right)\right) \qquad (6)$$

$$x_{i,t}^{ctx} = W_{11}\left[x_{i,t-1}^{ctx}; \sum_{j=1}^{N} m_{j,i}^{(t)}\right] \qquad (7)$$

A naive implementation would involve N^2 pairwise vectors $m_{j,i}^{(t)}$, which is inefficient for large N. We implement it more efficiently by building an N-row matrix M containing N unweighted messages $\tilde{m}_j^{(t)} = (W_9 \tilde{x}_{j,t}) \odot (W_{10} c_t)$ in Eqn. 6, which is left multiplied by the edge weight matrix E (where $E_{ij} = w_{j,i}^{(t)}$) to obtain the sums $\sum_{j=1}^{N} m_{j,i}^{(t)}$ in Eqn. 7 for all nodes in a single matrix multiplication. With this implementation, we can train our LCGN model efficiently with N as large as 196 in our experiments.

Final representation We combine each entity's local feature x_i^{loc} and context feature $x_{i,T}^{ctx}$ (after T iterations) as its final representation x_i^{out}:

$$x_i^{out} = W_{12}\left[x_i^{loc}; x_{i,T}^{ctx}\right] \qquad (8)$$

The x_i^{out} can be used as input to subsequent task-specific modules such as VQA or REF models, instead of the original local representation x_i^{loc}.

3.2. Application to VQA and REF

To apply our LCGN model to language-based reasoning tasks such as Visual Question Answering (VQA) and Referring Expression Comprehension (REF), we build simple task-specific output modules based on the language input and the contextualized representation of each entity. Our LCGN model and the subsequent task-specific modules are jointly trained end-to-end.

A single-hop answer classifier for VQA The VQA task requires outputting an answer for an input image I and a question Q. We adopt the commonly used classification approach and build a single-hop attention model as a classifier to select one of the possible answers from the training set.

First, the question Q is encoded into a vector q with the Bi-LSTM in Eqn. 1. Then a single-hop attention β_i is used over the objects to aggregate visual information, which is fused with q to predict the score vector y for each answer.

$$\beta_i = \operatorname*{Softmax}_i \left(W_{13}\left(x_i^{out} \odot (W_{14}q)\right)\right) \tag{9}$$

$$y = W_{15}\,\mathrm{ReLU}\left(W_{16}\left[\sum_{i=1}^{N} \beta_i x_i^{out}; q\right]\right) \tag{10}$$

During training, a softmax or sigmoid classification loss is applied on the output scores y for answer classification.

GroundeR [34] for REF The REF task requires outputting a target bounding box as the grounding result for an input referring expression Q. Here, we use a retrieval approach as in previous works and select one target entity from the N candidate entities in the scene (either object detection results or spatial locations on a convolutional feature map). To select the target object p from the N candidates, we encode expression Q to vector q as in Eqn 1 and build a model similar to the fully-supervised version of GroundeR [34] to output a matching score r_i for each entity i. In the case of using spatial locations on a convolutional feature map, we further output a 4-dimensional vector u to predict the bounding box offset from the feature grid location.

$$r_i = W_{17}\left(x_i^{out} \odot (W_{18}q)\right) \tag{11}$$

$$p = \arg\max_i r_i \tag{12}$$

$$u = W_{19}x_p^{out} \tag{13}$$

During training, we use a softmax loss over the scores r_i among the N candidates to select the target entity p, and an L2 loss over the box offset u to refine the box location.

4. Experiments

We apply our LCGN model to two tasks – VQA and REF – for language-conditioned reasoning. For the VQA task, we evaluate on the GQA dataset [19] and the CLEVR dataset [20], which both require resolving relations between objects. For the REF task, we evaluate on the CLEVR-Ref+ dataset [25]. In particular, the CLEVR and CLEVR-Ref+ datasets contain many complicated questions or expressions with higher-order relations, such as *the ball on the left of the object behind a blue cylinder*.

4.1. Visual Question Answering (VQA)

Evaluation on the GQA dataset We first evaluate our LCGN model on the GQA dataset [19] for visual question answering. The GQA dataset is a large-scale visual question answering dataset with real images from the Visual Genome dataset [22] and balanced question-answer pairs. Each training and validation image is also associated with scene graph annotations describing the classes and attributes of those objects in the scene, and their pairwise relations. Along with the images and question-answer pairs, the GQA dataset provides two types of pre-extracted visual features for each image – convolutional grid features of size $7 \times 7 \times 2048$ extracted from a ResNet-101 network [13] trained on ImageNet, and object detection features of size $N_{det} \times 2048$ (where N_{det} is the number of detected objects in each image with a maximum of 100 per image) from a Faster R-CNN detector [33].

Method	Accuracy[1]		
	val	test-dev	test
CNN+LSTM [19]	49.2%	–	46.6%
Bottom-Up [1]	52.2%	–	49.7%
MAC [18]	57.5%	–	54.1%
single-hop	62.0%	53.8%	54.4%
single-hop + LCGN (ours)	**63.9%**	**55.8%**	**56.1%**

Table 1. VQA performance on the GQA dataset.[1]

We apply our LCGN model together with the single-hop classifier (“**single-hop + LCGN**”) in Sec. 3.2 for answer prediction. We use $T = 4$ rounds of message passing in our LCGN model, which takes approximately 20 hours to train using a single Titan Xp GPU. As a comparison to our LCGN model, we also train the single-hop classifier with only the local features x^{loc} in Eqn. 9 (“**single-hop**”).

We first experiment with using the released object detection features in the GQA dataset as our local features x^{loc}, which is shown in [19] to perform better than the convolutional grid features, and compare with previous works.[1] Similar to MAC [18], we initialize question word embedding from GloVe [30] and maintain an exponential moving average of model parameters during training. To facilitate spatial reasoning, we concatenate the Faster R-CNN object detection features with their corresponding box coordinates. The results are shown in Table 1. By comparing “single-hop + LCGN” with “single-hop” in the last two rows, it can be seen that our LCGN model brings around 2% (absolute) improvement in accuracy, indicating that our LCGN model facilitates reasoning by replacing the local features x^{loc} with the contextualized features x^{out} containing rich relational information for the reasoning task. Figure 3 shows question answering examples from our model on this dataset.

We compare with three previous approaches in Table 1. CNN+LSTM [19] and Bottom-Up [1] are simple fusion approaches between the text and the image, using the released GQA convolutional grid features or object detection features respectively. The MAC model [18] is a multi-step attention and memory model with specially designed control, reading and writing cells, and is trained on the same object detection features as our model. Our approach outperforms the MAC model that performs multi-step inference, obtain-

[1] We learned from the GQA dataset authors that its *test-dev* and *test* splits were collected differently from its *train* and *val* splits, with a noticeable domain shift from *val* to *test-dev* and *test*. We train on the *train* split and report results on three GQA splits (*val*, *test-dev* and *test*). The performance of previous work on *val* was obtained from the dataset authors.

Method	Local features	Accuracy	
		val	test-dev
single-hop	convolutional grid features	55.0%	48.6%
single-hop + LCGN		**55.3%**	**49.5%**
single-hop	object features from detection	62.0%	53.8%
single-hop + LCGN		**63.9%**	**55.8%**
single-hop	GT objects and attributes[2]	87.0%	n/a
single-hop + LCGN		**90.2%**	n/a

Table 2. Ablation on different local features on the GQA dataset.

ing the state-of-the-art results on the GQA dataset.

We further apply our LCGN model to other types of local features, and experiment with using either the same $7\times7\times2048$-dimensional convolutional grid features (where each x_i^{loc} is a feature map location and $N = 49$) as used in CNN+LSTM in Table 1 or an "oracle" symbolic local representation at both training and test time, based on a set of ground-truth objects along with their class and attribute annotations ("GT objects and attributes") in the scene graph data of the GQA dataset. In the latter setting with symbolic representation, we construct two one-hot vectors to represent each object's class and attributes, and concatenate them as each object's x_i^{loc}.[2] The results are shown in Table 2, where our LCGN model delivers consistent improvements over all three types of local feature representations.

Evaluation on the CLEVR dataset We also evaluate our LCGN model on the CLEVR dataset [20], a dataset for VQA with complicated relational questions, such as *what number of other objects are there of the same size as the brown shiny object*. Following previous works, we use the $14 \times 14 \times 1024$ convolutional grid features extracted from the C4 block of an ImageNet-pretrained ResNet-101 network [13] as the local features x^{loc} on the CLEVR dataset (*i.e.* each x_i^{loc} is a feature map location and $N = 196$).

Similar to our experiments on the GQA dataset, we apply our LCGN model together with the single-hop answer classifier and compare it with using only the local features in the answer classifier. We also compare with previous works that use only question-answer pairs as supervision (*i.e.* without relying on the functional program annotations in [20]).

The results are shown in Table 3. It can be seen that the single-hop classifier only achieves 72.6% accuracy when using the local convolutional grid features x^{loc} ("**single-hop**"), which is unsurprising since the CLEVR dataset often involves resolving multiple and higher-order relations beyond the capacity of the single-hop classifier alone. However, when trained together with the context-aware representation x^{out} from our LCGN model, this same single-hop classifier ("**single-hop + LCGN**") achieves a significantly higher accuracy of 97.9% comparable to several state-of-the-art approaches on this dataset, showing that our LCGN model is able to embed relational context information in its output scene representation x^{out}. Among previous works, Stack-NMN [15] and MAC [18] rely on multi-step inference procedures to predict an answer. RN [35] pools over all N^2 pairwise object-object vectors to collect relational information in a single step. FiLM [31] modulates the batch normalization parameters of a convolutional network with the input question. NS-CL [27] learns symbolic representations of the scene and uses quasi-logical reasoning. Except for Stack-NMN [15], most previous works are tailored to the VQA task, and it is non-trivial to apply them to other tasks such as REF, while our LCGN model provides a generic scene representation applicable to multiple tasks. Figure 4 shows question answering examples of our model.

Method	Accuracy
Stack-NMN [15]	93.0%
RN [35]	95.5%
FiLM [31]	97.6%
MAC [18]	98.9%
NS-CL [27]	**99.2%**
single-hop	72.6%
single-hop + LCGN (ours)	97.9%

Table 3. VQA performance on the test split of the CLEVR dataset. We use $T = 4$ rounds of message passing in our LCGN model.

We further experiment with varying the number T of message passing iterations in our LCGN model. In addition, to isolate the effect of conditioning on textual inputs during message passing, we also train and evaluate a restricted version of LCGN without text conditioning ("**single-hop + LCGN w/o txt**"), by replacing the c_t's from Eqn 3 with a vector of all ones. The results are shown in Table 4, where it can be seen that using multiple rounds of iterations ($T > 1$) leads to a significant performance increase, and it is crucial to incorporate the textual information c_t into the message passing procedure. This is likely because the CLEVR dataset involves complicated questions that need multi-step context propagation. In addition, it is more efficient to collect the specific relational context relevant to the input question, instead of building a scene representation with a complete and unconditional knowledge base of all relational information that any input questions can query from.

Given that multi-round message passing ($T > 1$) works better than using only a single round ($T = 1$), we further study whether it is beneficial to have dynamic connection weights $w_{j,i}^{(t)}$ in Eqn. 5 that can be different in each iteration t to allow an object i to focus on different context objects j in different rounds. As a comparison, we train a restricted version of LCGN with static connection weights $w_{j,i}$ ("**single-hop + LCGN w/ static** $w_{j,i}$"), where we only predict the weights $w_{j,i}^{(1)}$ in Eqn. 5 for the first round $t = 1$, and reuse it in all subsequent rounds (*i.e.* setting

[2] In this setting, we can only evaluate on the *val* split with public scene graph annotations. We note that this is the only setting where we use the scene graphs in the GQA dataset. In all other settings, we only use the images and question-answer pairs to train our models. Also, our model does not rely on the GQA question semantic step annotations in any settings.

Method	Steps T	Accuracy
single-hop	n/a	72.6%
single-hop + LCGN	$T = 1$	94.0%
single-hop + LCGN	$T = 2$	94.5%
single-hop + LCGN	$T = 3$	96.4%
single-hop + LCGN	$T = 4$	**97.9%**
single-hop + LCGN	$T = 5$	96.9%
single-hop + LCGN w/o txt	$T = 4$	78.6%
single-hop + LCGN w/ static $w_{j,i}$	$T = 4$	96.5%

Table 4. Ablation on iteration steps T and whether to condition on the text or have dynamic weights, on the CLEVR validation split.

$w_{j,i}^{(t)} = w_{j,i}^{(1)}$ for all $t > 1$). From the last row of Table 4 it can be seen that there is a performance drop when restricting to static connection weights $w_{j,i}$ predicted only in the first round, and we also observe a similar (but larger) drop for the REF task in Sec. 4.2 and Table 5. This suggests that it is better to have dynamic connections during each iteration, instead of first predicting a fixed connection structure on which iterative message passing is performed (*e.g.* [29]).

4.2. Referring Expression Comprehension (REF)

Our LCGN model provides a generic approach to building context-aware scene representations and is not restricted to a specific task such as VQA. We also apply our LGCN model to the referring expression comprehension (REF) task, where given a referring expression that describes an object in the scene, the model is asked to localize the target object with a bounding box.

We experiment with the CLEVR-Ref+ dataset [25], which contains similar images as in the CLEVR dataset [20] and complicated referring expressions requiring relation resolution. On the CLEVR-Ref+ dataset, we evaluate with the bounding box detection task in [25], where the output is a bounding box of the target object and there is only one single target object described by the expression. A localization is consider correct if it overlaps with the ground-truth box with at least 50% IoU. Same as in our VQA experiments on the CLEVR dataset in Sec. 4.1, here we also use the $14 \times 14 \times 1024$ convolutional grid features from ResNet-101 C4 block as our local features x^{loc} (*i.e.* each x_i^{loc} is a feature map location and $N = 196$), with $T = 4$ rounds of message passing. The final target bounding box is predicted with a 4-dimensional bounding box offset vector u in Eqn. 13 from the selected grid location p in Eqn. 12.

We apply our LCGN model to build a context-aware representation x^{out} conditioned on the input referring expression, which is used as input to our implementation of the GroundeR approach [34] (Sec. 3.2) for bounding box prediction ("**GroundeR + LCGN**"). As a comparison, we train and evaluate the GroundeR model without our context-aware representation ("**GroundeR**"), using local features x^{loc} as inputs in Eqn. 11. Similar to our experiments on the CLEVR dataset for VQA in Sec. 4.1, we also ablate our LCGN model with not conditioning on the input expression in message passing ("**GroundeR + LCGN w/o txt**") or using static connection weights $w_{j,i}$ predicted from the first round ("**GroundeR + LCGN w/ static** $w_{j,i}$").

Method	Accuracy
Stack-NMN [15]	56.5%
SLR [45]	57.7%
MAttNet [43]	60.9%
GroundeR [34]	61.7%
GroundeR + LCGN w/o txt	65.0%
GroundeR + LCGN w/ static $w_{j,i}$	71.4%
GroundeR + LCGN (ours)	**74.8%**

Table 5. Performance on the CLEVR-Ref+ dataset for REF.

The results are shown in Table 5, where our context-aware scene representation from LCGN leads to approximately 13% (absolute) improvement in REF accuracy. Consistent with our observation on the VQA task, for the REF task we find it important for the message passing procedure to depend on the input expression, and allowing the model to have dynamic connection weights $w_{j,i}^{(t)}$ that can differ for each round t. Our model outperforms previous work by a large margin, achieving the state-of-the-art performance for REF on the CLEVR-Ref+ dataset. Figure 5 shows example predictions of our model on the CLEVR-Ref+ dataset.

In previous works, SLR [45] and MAttNet [43] are specifically designed for the REF task. SLR jointly trains an expression generation model (speaker) and an expression comprehension model (listener), and MAttNet relies on modular structure for subject, location and relation comprehension. While Stack-NMN [15] is also a generic approach that is applicable to both the VQA task and the REF task, the major contribution of Stack-NMN is to construct an explicit step-wise inference procedure with compositional modules, and it relies on hand-designed module structures and local appearance-based scene representations. On the other hand, our work augments the scene representation with rich relational context. We show that our approach outperforms Stack-NMN on both the VQA and the REF tasks.

5. Conclusion

In this work, we propose Language-Conditioned Graph Networks (LCGN), a generic approach to language-based reasoning tasks such VQA and REF. Instead of building task-specific inference procedures, our LCGN model constructs rich context-aware *representations* of the scene through iterative message passing. Experimentally, we show that the context-aware representations from our LCGN model can improve over the local appearance-based representations across various types of local features and multiple datasets, and it is crucial for the message passing procedure to depend on the language inputs.

Acknowledgements. This work was partially supported by the Berkeley AI Research, the NSF and DARPA XAI.

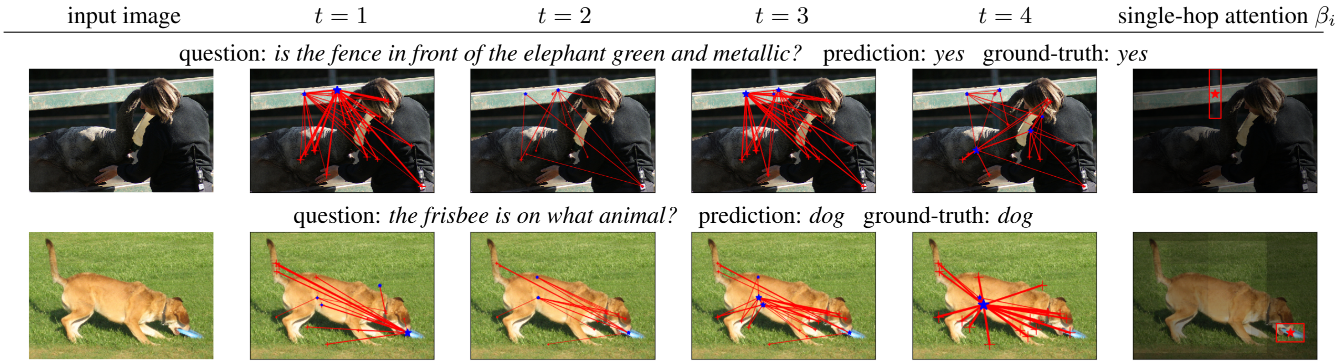
input image
$t = 1$
$t = 2$
$t = 3$
$t = 4$
single-hop attention β_i
question: *is the fence in front of the elephant green and metallic?* prediction: *yes* ground-truth: *yes*
question: *the frisbee is on what animal?* prediction: *dog* ground-truth: *dog*

Figure 3. Examples from our LCGN model on the validation split of the GQA dataset for VQA. In the middle 4 columns, each red line shows an edge $j \rightarrow i$ along the message passing paths (among the N detected objects) where the connection edge weight $w_{j,i}^{(t)}$ exceeds a threshold. The blue star on each line is the sender node j, and the line width corresponds to its connection weight. In the upper example, the person, the elephant and the fence propagate messages with each other, and fence receives messages from the elephant in $t = 4$. In the lower example, the frisbee collect messages from the dog as contextual information in multiple rounds, and is picked up by the single-hop classifier. The red star (along with the box) in the last column shows the object with the highest single-hop attention β_i in Eqn. 9.

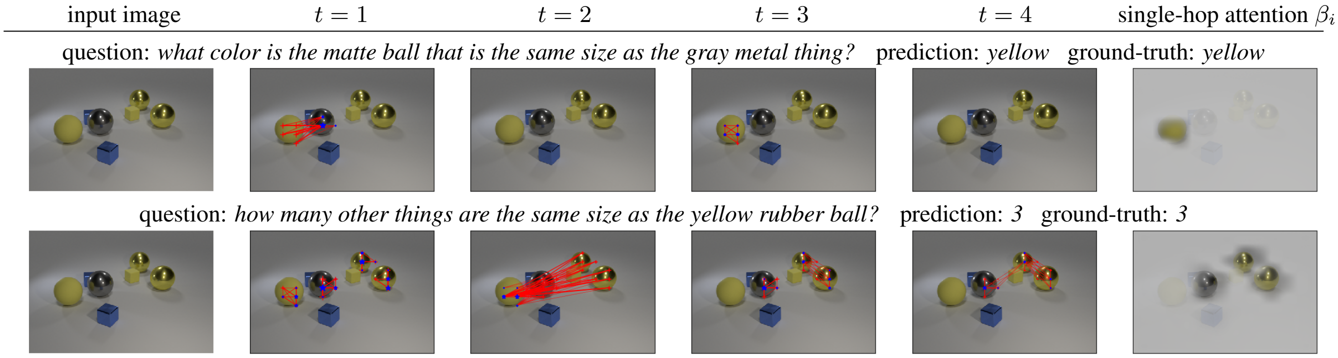
input image
$t = 1$
$t = 2$
$t = 3$
$t = 4$
single-hop attention β_i
question: *what color is the matte ball that is the same size as the gray metal thing?* prediction: *yellow* ground-truth: *yellow*
question: *how many other things are the same size as the yellow rubber ball?* prediction: *3* ground-truth: *3*

Figure 4. Examples from our LCGN model on the validation split of the CLEVR dataset for VQA. The middle 4 columns show the connection edge weights $w_{j,i}^{(t)}$ similar to Figure 3, where the blue stars are the sender nodes. The last column shows the single-hop attention β_i in Eqn. 9 over the $N = 14 \times 14$ feature grid. In the upper example, in $t = 1$ the matte ball (leftmost) collects messages from the gray metal ball (of the same size), and then in $t = 3$ messages are propagated within the convolutional grids on the matte ball, possibly to refine the collected context from the gray ball. In the lower example, in $t = 1$ all four balls try to propagate messages within the convolutional grids of each ball region, and in $t = 2$ the three other balls (of the same size) receive messages from the rubber ball (leftmost) and are picked up by the single-hop classifier.

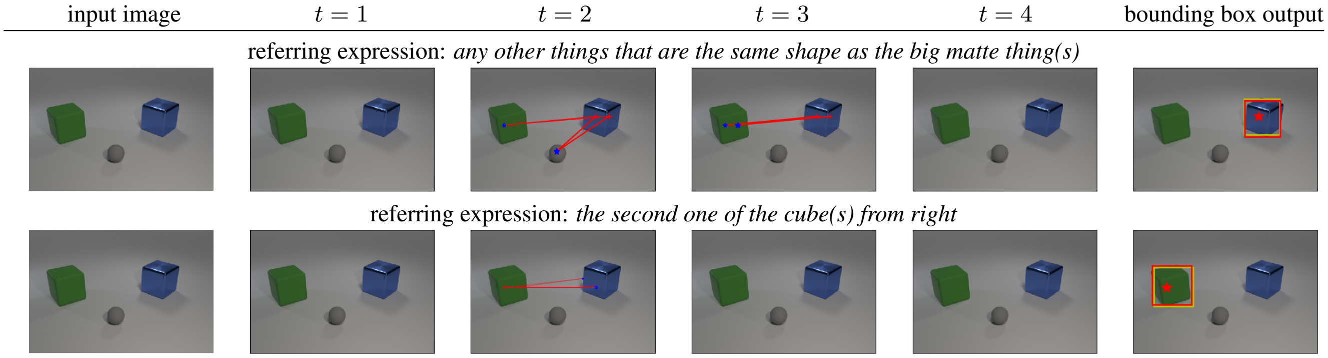
input image
$t = 1$
$t = 2$
$t = 3$
$t = 4$
bounding box output
referring expression: *any other things that are the same shape as the big matte thing(s)*
referring expression: *the second one of the cube(s) from right*

Figure 5. Examples from our LCGN model on the validation split of the CLEVR-Ref+ dataset for REF. The middle 4 columns show the connection edge weights $w_{j,i}^{(t)}$ similar to Figure 3, where the blue stars are the sender nodes. The last column shows the selected target grid location p on the $N = 14 \times 14$ spatial grid (the red star) in Eqn. 12, along with the ground-truth (yellow) box and the predicted box (red box from bounding box regression u in Eqn. 13). In the upper example, the blue cube (the target object) collects messages from the two other objects in $t = 2$, and then the blue cube further collects messages from the big matte green cube on the left (which has the same shape) in $t = 3$. In the lower example, the green cube checks for other cubes by collecting messages from things on its right in $t = 2$.

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*, pages 6077–6086, 2018.

[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 (CVPR)*, 2016.

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

[4] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[5] Remi Cadene, Hedi Ben-younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[6] Simyung Chang, John Yang, SeongUk Park, and Nojun Kwak. Broadcasting convolutional network for visual relational reasoning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 754–769, 2018.

[7] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, 2018.

[8] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[9] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6639–6648, 2019.

[10] Peng Gao, Hongsheng Li, Shuang Li, Pan Lu, Yikang Li, Steven CH Hoi, and Xiaogang Wang. Question-guided hybrid convolution for visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 469–485, 2018.

[11] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[13] 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*, pages 770–778, 2016.

[14] Roei Herzig, Elad Levi, Huijuan Xu, Eli Brosh, Amir Globerson, and Trevor Darrell. Classifying collisions with spatio-temporal action graph networks. *arXiv preprint arXiv:1812.01233*, 2018.

[15] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018.

[16] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] 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.

[18] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2018.

[19] Drew A Hudson and Christopher D Manning. Gqa: a new dataset for compositional question answering over real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] 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 *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

[21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[22] 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.

[23] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[25] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[26] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: object detection using scene-level context and instance-level relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6985–6994, 2018.

[27] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *ICLR*, 2019.

[28] 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.

[29] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems*, pages 8344–8353, 2018.

[30] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[31] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[32] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–417, 2018.

[33] 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.

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

[35] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4974–4983, 2017.

[36] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

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

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[41] Yiqun Yao, Jiaming Xu, Feng Wang, and Bo Xu. Cascaded mutual modulation for visual reasoning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[42] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050, 2018.

[43] 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.

[44] 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.

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

[46] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[47] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. Structured attentions for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1291–1300, 2017.