# Visual Grounding for Multiple Instances

Bryce Yahn Jing Shi Ning Xu Chenliang Xu
University of Rochester Adobe Research

**Abstract:**
Referring expression grounding is an important machine vision task that incorporates language comprehension, object detection, and relational reasoning. The goal is to locate the objects in an image that are described in an expression called the referring expression. For instance, if the expression was "The plant on the table to the left of the orange lamp", the model should be able to draw a bounding box around the plant in the image. While the area of referring expression grounding is well explored for single instance referrents, few attempts have been made to ground multiple instances for one image and referring expression. The grounding of multiple instances is an interesting problem because it allows for different types of reasoning than single instance grounding, and is in general a more complex task. In this work, we generate a new version of the CLEVR-Ref+ dataset that includes expressions for multiple instance logic. We also modify the Language-Conditioned Graph Network proposed by Hu et al. to do context-aware visual grounding for multiple instances using two approaches. In the first approach, we consider each proposed region individually and select those to be included in the final set based on individual matching scores. In the second approach, we look at the combined features of all objects in the set and evaluate the sets of proposed regions directly. Preliminary results show that the first approach performs well but is subject to the threshold hyperparameter. The second approach directly output the best set without consideration of threshold.

## 1. Introduction

The task of visual grounding is incredibly interesting for artificial intelligence because it requires quality performance in natural language processing, object detection, and relational reasoning. The objective is that, given an image and a referring expression, the model will detect the objects referred to in the expression. For example, in Figure 1, if the expression is "The pink donut beneath the first chocolate donut from the left," the model will draw a bounding box around the pink striped donut. This requires the model to process the expression, find the objects in the image described in the expression, and do reasoning over the relationships in the expression to determine which object is the referent.

While visual grounding for single instances has been studied extensively, almost no work has done on grounding multiple instances. Visual grounding for multiple instances is a one to many problem where there is one image, one expression, and multiple boxes to be drawn. For

example, if the expression for figure one is “Chocolate donuts with sprinkles beneath the white donut,” the model should draw two bounding boxes, one for each donut that fits the description.

Figure 1: Example of single instance grounding (Box in red): “The pink donut beneath the first chocolate donut from the left.”
Example of visual grounding for multiple instances (Boxes in green): “Chocolate donuts with sprinkles beneath the white donut.”

Grounding for multiple instances provides its own unique problems as compared to grounding for single instances. First and foremost, it increases the overall complexity of the problem. In the single instance version of the problem, only the most probable image region is selected as the output. There is guaranteed to be one correct answer for every expression. However, for the case with multiple instances, each possible image region must be evaluated, and the output set varies in size. Additionally, grounding for multiple instances provides novel forms of logic unavailable in single instance grounding. The three main types of logic we consider here are counting, unions, and exclusions. Counting expressions involve numbers of some sort. This might be describing the number of objects in the image that fit the description, or a subset of objects where the size of the subset matters. For example “The four donuts that are farthest left.” Unions involve objects that fit either of two descriptions. While unions can in theory be applied to single instances, in that case, the selected object will only fit one of the descriptions. Grounding for multiple instances has the possibility of having multiple objects that fit different parts of the description in the output set. For example “Donuts that are yellow or on the bottom row.” The final logic type is exclusions. Exclusions allow for any objects that don’t match a certain description, and therefore can have wildly different visual characteristics or relationships. For example, “Donuts that aren’t pink.” Each of these types of logic provides an interesting new challenge for visual grounding models.

The task of visual grounding itself demonstrates the ability of artificial intelligence to perform multiple high level tasks in conjunction with each other. Beyond its theoretical significance, visual grounding for multiple instances can be used for real world applications. One clear use is using machine vision to identify subgroups of objects with certain features. For instance, it might be valuable to identify all donuts with nuts on them or all fans in a stadium wearing the home team’s colors. Another possible use is language guided image editing. For example, if you’re trying to edit your vacation photos, it might be of use to quickly select “all the cars and people.”

In this paper, we present two approaches to visual grounding for multiple instances based on the *Language-Conditioned Graph Network* and tested on a modified version of the *CLEVR-Ref+* dataset. In the first approach which we consider our baseline approach, we look at each proposed image region separately and use a matching score with a threshold to determine whether or not it should be included in the predictions set. For the second approach, we process all the image features of proposed sets simultaneously and select the set with the highest matching score. The baseline model appears to have comparable performance to the published single instance models, however the hyperparameter threshold is an undesirable aspect that limits performance. The set approach does not currently yield good results, but may be improved with experimentation and tuning.

**2. Related Works**

Visual grounding for single instances have been studied extensively, but little consideration has been given to a multi-instance version of the task. Within single instance grounding, multiple approaches and many models have been suggested. Most of the models can be put into one of three categories: monolithic, modular, and graph-based. Monolithic models combine the language features with the visual features of different regions and then process them together with an LSTM as they would process a normal sentence[6, 7]. Modular models are rather popular [1, 9, 11]. These models break down the expressions into different language components (generally some variation of "entity," "attribute," and "relation"). A different module attends over the image using the text from each component to find visual matches in the image. Finally, the network combines the output and selects a final bounding box as the answer. The final approach is more recent. Graph based models tend to represent objects as nodes in a graph with the relationships between objects represented by edges. They then use the language to create, navigate, or update the graph [3,4,8,10]**.** There are models that cross over between these categories such as *Referring Image Segmentation via Cross-Modal Progressive Comprehension* which uses a two stage approach. It begins with a modular analysis of entity and attribute features. It then forms a graph from proposed regions and uses message passing to do relational reasoning guided by the "relation" words in the expression[4].

Almost no published models do visual grounding for multiple instances. There are not even many datasets that provide the possibility of multi-instance grounding. One dataset published in 2019 is the *PhraseCut* dataset[9]. This is a real image data set where ~20% of image-referring expression pairs have multiple referents. In the associated paper, the authors propose HULANet, which is a modular approach that considers attribute, category, and relationships. The model performs comparably with the baselines and better than some state of the art models, but the authors point out that there is room for improvement and encourage further development of multi-instance models. This dataset provides complex scenes with many objects and potential

relations, but the expressions are very simple and the reasoning is limited. Most expressions are three to four words long, and the reasoning is entirely positional.

In 2019, Liu et al published the *CLEVR-Ref+* dataset, which is a synthetic dataset which provides the opportunity for grounding multiple instances[5]. In their paper, they propose a model called IEP-REF which was adapted from a visual question answering model. While the dataset allows for grounding multiple instances, they only test their detection task on samples with a single referent for easy comparison to state of the art models.

**3. Method**

We demonstrate two approaches built off of the Language-Conditioned Graph Network and tested on a modified version of the *CLEVR-Ref+* dataset. This first involved modifying the Clevr-Ref+ dataset to include examples of logic for multiple instances. Then we designed an approach that looked at each individual proposed region, assigned a probability of it being in the output set, and selected all proposals with a probability above a certain threshold. This is considered our baseline approach because it is the simplest adjustment. We then designed a set-based approach where we looked at the features of all proposals in a set and selected the most probable set.

3.1 The Dataset

The *Clevr-Ref+* dataset is a diagnostic synthetic dataset for referring expression comprehension. The images are composed of simple three dimensional objects that all have a shape, color, texture, visibility, and ordinality (ex: second from the left). These objects are in relationships with each other including (to the front of, to the left of, to the right of, and to the back of). These images are paired with referring expressions that fit into different logic families. There are four original logic families. Same-relate which looks for objects with the same attributes as a selected object. The N-hop family includes expressions with up to N relationships where N ranges from zero to three. The single-and family involves objects that fit both of two descriptions, and the single-or family involves objects that fit either of two descriptions.

When working with this dataset, I balanced it so that some of the features were better represented. For instance, “ordinality” showed up in almost ninety percent of expressions, so I reduced the prevalence of it so that other attributes were more present. I also added two additional expression types. “Count-returns” requests a specific number of items that fit the description, and “check_farthest” requests the N things farthest left, right, front, or back. In the future, I would like to add logic for exclusions, however, that was not included in the experiments and results listed below.

### 3.2 The Language-Conditioned Graph Network

We built our models off of the *Language-Conditioned Graph Network (LCGN)* published by Hu et al in 2019. The main goal of this model is to build contextualized feature representations of image regions using a graph network conditioned on the text. Each image region is represented by a local feature vector and a context feature vector. The local feature vector remains constant, while the context feature vector is update over multiple rounds of message passing. The message passing is directed by textual commands extracted from the expression using an LSTM. After the message passing is complete, the local and context features are combined and output as the feature vector for each region which may then be used for visual grounding or visual question answering. For the visual grounding, a matching score is calculated between each image region and the expression using an MLP on the dot product of the region output features and the encoded expression. The model uses multi-class cross entropy loss with softmax to select which region is most likely to be the referred object. It then uses bounding box regression to adjust the bounding box coordinates and outline the selected object. We chose this as the backbone of our model because the context is necessary for the complex reasoning involved in our task, and additionally, the LCGN was trained and tested on single instance examples from the *CLEVR-Ref+* dataset.

### 3.3 The Baseline Model

Our first approach involved looking at each individual proposal and independently determining whether or not it belonged in the output set. To do this, we used the LCGN framework to generate a probability of each proposal being a ground truth box and then used a threshold (0.9) to select which boxes were in the final set. We then used bounding box regression to calculate the box coordinates. To accomplish this, we had to modify the representation of the data throughout the model to keep track of a varying number of objects for each sample. For our loss function, we used Binary Cross Entropy loss with a sigmoid on each proposed region. This loss was weighted by the representation of ground truth positive and negative boxes in the sample. On average, about 99% of the proposals were ground truth negative while 1% were ground truth positive. Like the original LCGN, we use L2 loss for the bounding box regression, and we use the Adam optimizer.

### 3.4 The Set-Based Model

The baseline model looked at each proposal independently, but with visual grounding for multiple instances, the output is one correct set of boxes. Additionally, the baseline has a threshold hyperparameter which can drastically affect the performance of the model. We wanted to see if we could evaluate sets directly and remove the need for a threshold. Since the total number of possible sets grows exponentially with the number of proposals, it's infeasible to consider all possible sets. We therefore developed separate training and testing approaches for considering sets without considering all possible combinations of proposals.

For training, we run the message passing as normal so that we're working on the context represented boxes. We create four set representations, one positive, and three negative. The sets were created using average pooling of the feature vectors of the proposals in the set as seen in equation 1,

$$\text{Eq 1: } X_s = \frac{1}{len(S)} * \sum_{i \in S} X_i$$

Where $X_s$ is the feature vector of the set S, and $X_i$ is the feature vector of the $i^{th}$ proposal in the set S. The positive set is made up of all of the proposals that are in the ground truth set. Then there are three negative sets of different sizes. One is neg_same which is the same size as the ground truth set but which is composed of random proposals. One is neg_plus which is composed of all proposals in the ground truth positive set plus one extra proposal not in the ground truth positive set. The final negative set is a subset of the ground truth positive set with one missing element. We use an MLP to calculate the matching scores between the expression and set features, and take the sigmoid of that to treat it as a probability as seen in equation 2

$$\text{Eq 2: } P_s = \sigma(W_1(X_s \odot W_s q))$$

Where q is the summary vector of the BiLSTM and $P_s$ is the probability that the set is the ground truth set. Finally, we train using triplet loss so that the probability of the positive set is greater than the average probability of the negative sets by some margin (0.5) as seen in equation 3

$$\text{Eq 3: } L = max(0, P_{s_n} - P_{s_p} + margin)$$

Where $P_{sp}$ is the probability of the positive set and $P_{sn}$ is the average probability of the negative sets. This loss is added to the L2 loss for the bounding box regression.

During testing, the model takes the output of the message passing and uses the MLP from equation 2 to calculate the probability of each set of size one. It then uses beam search to continuously add boxes and calculate probabilities until the probabilities no longer increase. It finally takes the set with the highest probability and treats that as the output. For instance, if there are three ground truth boxes, and the beam width is k, it will calculate the likelihood of each individual box being the output set. It then selects the k most likely boxes and uses equation 1 to combine their features into a set of size two. It calculates the probability of each of these new sets and then selects the top k most likely sets of size two. It compares these probabilities to the sets of size one. If the probability increases, it will use equation 1 to consider all unique sets of size three of which the top k sets are a subset. It will calculate the probability of each of these sets, pick the top k, and continue the process. Ideally, it when it gets to sets of size 4, the probabilities will all decrease, and it will select the set of size three with the highest probability.

| Table 1 | | | |
|---|---|---|---|
| Model\Metric | IOU Accuracy | Index Accuracy | Top Accuracy |
| Stack-NMN [2] | 56.5% | / | / |
| SLR [12] | 57.7% | / | / |
| MAttNet [11] | 60.9% | / | / |
| LCGN [3] | 74.8% | / | / |
| Baseline (ours) | 66.4% | 68.4% | 82.9% |
| Set-Model (ours) | 28.4% | 36.5% | 37.8% |

Table 1: The first four rows are the results of the LCGN and three state of the art models on the original CLEVR-Ref+ dataset for single instance grounding. The last two rows (below the bold line) are our models tested on the modified CLEVR-Ref+ dataset for multi-instance grounding.This is not a direct comparison, but it gives a general idea of the performance of our models.

**4. Results**

The results below are largely preliminary results, as the set-model requires more tuning, improvement, and possibly debugging.

4.1 Metrics

Due to the imbalance of negative and positive samples in our data, the area under the ROC and the overall accuracy (percentage of proposals correctly classified as positive or negative) are not representative metrics. We use three metrics to assess our model's performance. The bounding box IOU accuracy is the percentage of bounding boxes with an intersection of union with a ground truth box over 50%. The bounding box index accuracy is the percentage of proposed positive boxes that were ground truth positive whether or not the bounding box overlapped. The final metric is top accuracy which is the percentage of the proposals with the top k probabilities that were ground truth positive where k is the total number of ground truth positive boxes.

4.2 Baseline

Since there aren't any published examples of visual grounding for multiple instances on the Clevr-Ref+ dataset, it's difficult to make comparisons on the effectiveness of our baseline. We provide the published performance of the LCGN doing single instance grounding on the original Clevr-Ref+ dataset for comparison, but we acknowledge that this is not a direct comparison. The bounding box IOU accuracy and bounding box index accuracy is comparable to the performance

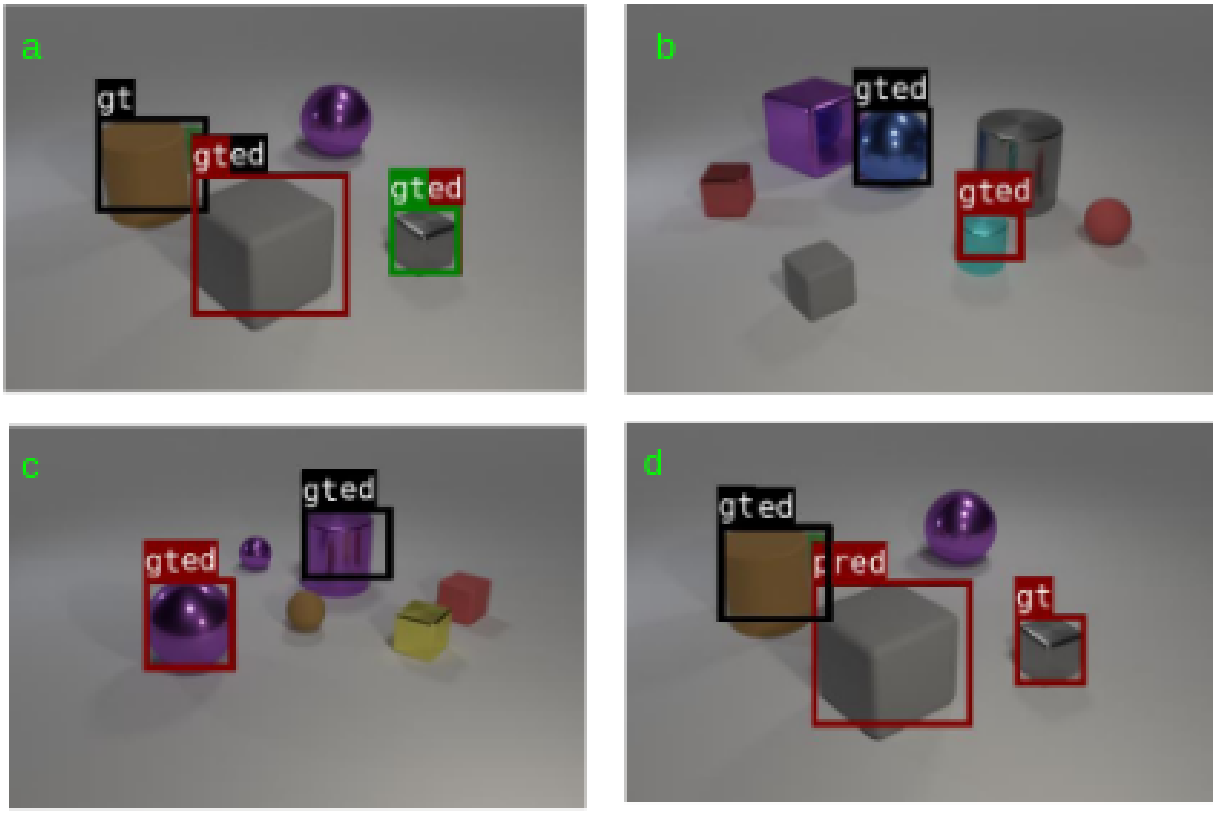


Figure 2: Qualitative results for the baseline model. "Gt" stands for ground truth box. "Pred" stands for predicted box. And gted is where the ground truth and predicted boxes overlap. a) Expression "The three fully visible things that are farthest front" Accuracy 0.667. b) Expression "The metallic things that are to the left of the large gray cylinder and to the right of the purple metallic cube" Accuracy 1.0. c) Expression "Both big metal things" Accuracy 1.0. d) Expression "Objects that are either the third one of the rubber objects from the right or gray objects that are to the right of the third one of the matte things from the left" Accuracy 0.5.

of the published LCGN. The top accuracy (0.829) is much higher than the performance of the published model. This is encouraging, and one might suggest simply lowering the threshold to capture more of these correct boxes, however this number does not account for false positives. By lowering the threshold, the number of falsely identified boxes increases, which is not necessarily better performance. Ultimately, the accuracy appears to be competitive with state of the art models for single instance grounding, but the threshold hyperparameter is a factor that limits performance.

4.3 Set-Model

The performance of the set-model was below what we expected, but there is still much tuning and experimentation to be done before rejecting this approach. For the training, the goal is that the probability of the positive set is larger than the average probabilities of the negative sets by

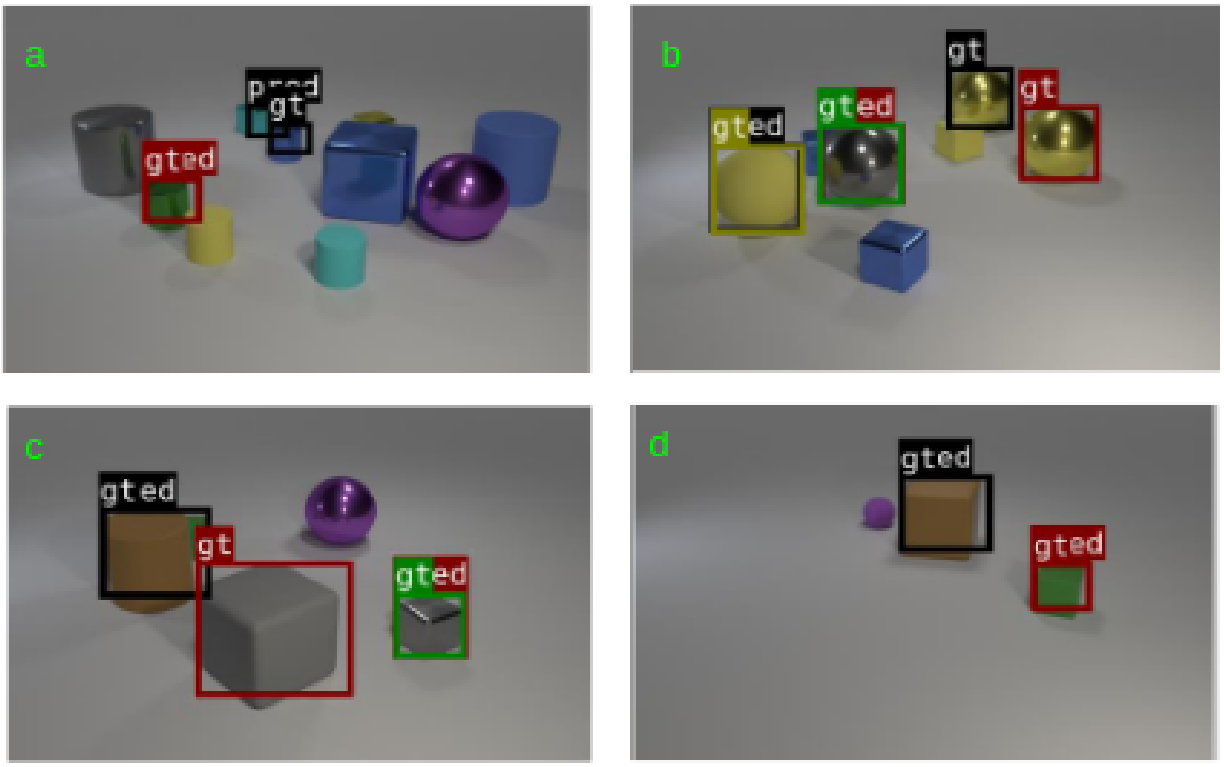


Figure 3: Qualitative results for the set-model. "Gt" stands for ground truth box. "Pred" stands for predicted box. And gted is where the ground truth and predicted boxes overlap. a) Expression "Both small metallic cylinders" Accuracy 0.5. b) Expression: "The things that are either large objects or shiny objects that are to the back of the small rubber block" Accuracy 0.5. c) Expression "The fully visible objects that are to the front of the shiny object that is to the back of the cylinder that is to the front of the partially visible rubber cylinder" Accuracy 0.667. d) Expression "Find the first one of the tiny things from the left; the things that are to the right of it" Accuracy 1.0.

some margin. During the final batch of training, the MLP output a probability of 0.763 for the positive set, ~0 for the random negative set, 0.031 for the neg_plus set, and 0.352 for the neg_minus set. This is exactly what we are hoping to see. The positive set should be most probable by a wide margin. We want to punish wrong answers, so the random negative set which is composed solely of incorrect boxes should have an incredibly low probability. We also want the probability to decrease drastically if incorrect boxes are added to the ground truth set. Finally, we want the probability to increase as correct boxes are added, but we want a subset of the ground truth set to still have a relatively high probability so that it stands out as a likely candidate during beam search.

The testing stage of the set-model did not reflect the promising results in the training phase. This is unideal, but we believe this is mainly due to the testing strategy. We hope to develop a more appropriate testing strategy that can accurately predict the correct set. This may involve

updating the loss function so that the probability of subsets of different sizes are compared, so that the probability consistently increases as correct boxes are added, and it's clear which boxes are good candidates for the initial set of size n=1.

**5. Conclusion and Future Work**

In this work, we explore two approaches to grounding multiple instances. In the first approach, we look at each box individually and use a threshold to predict whether or not it belongs in the output set. This performs satisfactorily well, but is limited by the threshold. In the second approach, we analyze sets of proposals and attempt to choose the optimal set. This showed promising results in the training, but did not meet expectations in the test phase. We hope that we can improve upon this and apply our model to real world image datasets such as the Phrase-cut dataset.

**Acknowledgements**. This work is funded by the NSF.

**References**


[1] Hu, R., Andreas, J., Rohrbach, M., Darrell, T., & Saenko, K. (2017). Learning to Reason: End-to-End Module Networks for Visual Question Answering. *2017 IEEE International Conference on Computer Vision (ICCV)*.

[2] Hu, R., Andreas, J., Darrell, T., & Saenko, K. (2018). Explainable Neural Computation via Stack Neural Module Networks. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science,* 55-71.

[3] Hu, R., Rohrbach, A., Darrell, T., & Saenko, K. (2019). Language-Conditioned Graph Networks for Relational Reasoning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

[4] Huang, S., Hui, T., Liu, S., Li, G., Wei, Y., Han, J., . . . Li, B. (2020). Referring Image Segmentation via Cross-Modal Progressive Comprehension. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Liu, R., Liu, C., Bai, Y., & Yuille, A. L. (2019). CLEVR-Ref : Diagnosing Visual Reasoning With Referring Expressions. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., & Murphy, K. (2016). Generation and Comprehension of Unambiguous Object Descriptions. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Nagaraja, V. K., Morariu, V. I., & Davis, L. S. (2016). Modeling Context Between Objects for Referring Expression Understanding. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science,* 792-807.

[8] Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., & Hengel, A. V. (2019). Neighbourhood Watch: Referring Expression Comprehension via Language-Guided Graph Attention Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Wu, C., Lin, Z., Cohen, S., Bui, T., & Maji, S. (2020). PhraseCut: Language-Based Image Segmentation in the Wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Yang, S., Li, G., & Yu, Y. (2020). Graph-Structured Referring Expression Reasoning in the Wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., & Berg, T. L. (2018). MAttNet: Modular Attention Network for Referring Expression Comprehension. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[12] Yu, L., Tan, H., Bansal, M., & Berg, T. L. (2017). A Joint Speaker-Listener-Reinforcer Model for Referring Expressions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.