Ref-NMS: Breaking Proposal Bottlenecks in Two-Stage Referring Expression Grounding

Long Chen,[2*] **Wenbo Ma,**[1*] **Jun Xiao,**[1†] **Hanwang Zhang,**[3] **Shih-Fu Chang**[4]

[1] College of Computer Science, Zhejiang University, Hangzhou
[2] Tencent AI Lab, Shenzhen
[3] MReaL Lab, Nanyang Technological University, Singapore
[4] DVMM Lab, Columbia University, New York

Abstract

The prevailing framework for solving referring expression grounding is based on a two-stage process: 1) detecting proposals with an object detector and 2) grounding the referent to one of the proposals. Existing two-stage solutions mostly focus on the grounding step, which aims to align the expressions with the proposals. In this paper, we argue that these methods overlook an obvious *mismatch* between the roles of proposals in the two stages: they generate proposals solely based on the detection confidence (*i.e.*, expression-agnostic), hoping that the proposals contain all right instances in the expression (*i.e.*, expression-aware). Due to this mismatch, current two-stage methods suffer from a severe performance drop between detected and ground-truth proposals. To this end, we propose Ref-NMS, which is the first method to yield expression-aware proposals at the first stage. Ref-NMS regards all nouns in the expression as critical objects, and introduces a lightweight module to predict a score for aligning each box with a critical object. These scores can guide the NMS operation to filter out the boxes irrelevant to the expression, increasing the recall of critical objects, resulting in a significantly improved grounding performance. Since Ref-NMS is agnostic to the grounding step, it can be easily integrated into any state-of-the-art two-stage method. Extensive ablation studies on several backbones, benchmarks, and tasks consistently demonstrate the superiority of Ref-NMS. Codes are available at: https://github.com/ChopinSharp/ref-nms.

Introduction

Referring Expression Grounding (REG), *i.e.*, localizing the targeted instance (referent) in an image given a natural language description, is a longstanding task for multimodal understanding. Considering different granularities of localization, there are two sub-types of REG: 1) **Referring Expression Comprehension (REC)** (Hu et al. 2017, 2016; Yu et al. 2016, 2017), where the referents are localized by bounding boxes (bboxes). 2) **Referring Expression Segmentation (RES)** (Hu, Rohrbach, and Darrell 2016; Liu et al. 2017; Shi et al. 2018; Margffoy-Tuay et al. 2018), where the referents are localized by segmentation masks. Both two tasks are important for many downstream high-level applications such as VQA (Antol et al. 2015), navigation (Chen et al. 2019b), and autonomous driving (Kim et al. 2019).

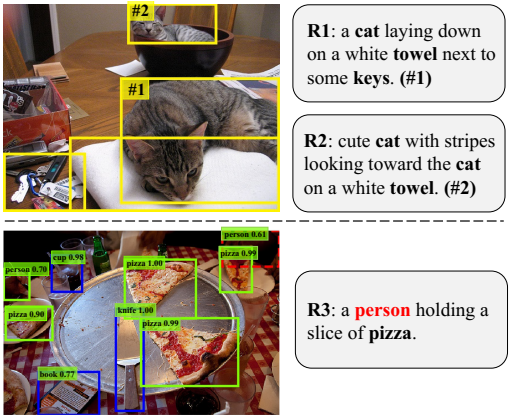

Figure 1: Upper: A typical REC example from RefCOCOg. The two "similar" expressions (R1 and R2) refer to different objects. Below: An example of proposals from the first-stage of prevailing MAttNet (Yu et al. 2018a). The proposals only contain bboxes with high detection confidence ($>$ 0.65) regardless of the content of expression (*e.g.*, The candidates `knife`, `book`, and `cup` are not mentioned in R3). Red dashline bbox denotes the missing referent.

State-of-the-art REG methods can be classified into two major categories: one-stage, proposal-free methods and two-stage, proposal-driven methods. For the one-stage methods (Chen et al. 2018; Yang et al. 2019b; Liao et al. 2020), they regard REG as a generalized object detection (or segmentation) task, and the whole textual expression is treated as a specific object category. Although these one-stage methods achieve faster inference speed, their grounding performance, especially for complex expressions (*e.g.*, in dataset RefCOCOg), is still behind the two-stage counterpart. The main reasons for the differences are two-fold: 1) The one-stage methods naturally focus on the local content, *i.e.*, they fail to perform well in the expressions which need global reasoning. For example in Figure 1, when grounding "*a cat laying down on a white towel next to some keys*", it is even difficult for humans to identify the referent `cat` without

*indicates equal contribution ({longc, mwb}@zju.edu.cn).
†indicates corresponding author (junx@zju.edu.cn).

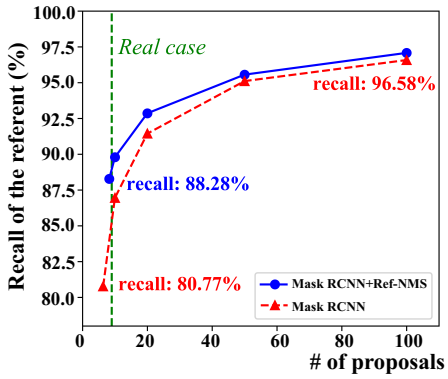

Figure 2: The recall of the referent (IoU>0.5) vs. number of proposals on the RefCOCO testB set. The real case denotes the actual situation in all SOTA two-stage methods.

considering its contextual objects `towel` and `keys`. 2) The one-stage methods do not exploit the linguistic structure of expressions, *i.e.*, they are not sensitive to linguistic variations in expressions. For instance, when changing the expression in Figure 1 to "*cute cat with stripes looking toward the cat on a white towel*", they tend to refer to the same object (#1) (Akula et al. 2020). On the contrary, the two-stage methods (Yu et al. 2018a; Liu et al. 2019a,c) intuitively are more similar to the human way of reasoning: 1) detecting proposals with a detector, and then 2) grounding the referent to one of the proposals. In general, two-stage methods with perfect proposals (*e.g.*, all human-annotated object regions) can achieve more accurate and explainable grounding results than the one-stage methods.

Unfortunately, when using the results from off-the-shelf detectors as proposals, two-stage methods' performance all drops dramatically. This is also the main weakness of two-stage solutions often criticized by competing methods in the literature, *i.e.*, the performance of two-stage methods is heavily limited by the proposal quality. In this paper, we argue that this huge performance gap between the detected and ground-truth proposals is mainly caused by the **mismatch** between the roles of proposals in the two stages: *the first-stage network generates proposals solely based on the detection confidence, while the second-stage network just assumes that the generated proposals will contain all right instances in the expression*. More specifically, for each image, a well pre-trained detector can detect hundreds of detections with a near-perfect recall of the referent and contextual objects (*e.g.*, as shown in Figure 2, recall of the referent can reach up to 96.58% with top-100 detections). However, to relieve the burden of the referent grounding step in the second stage, current two-stage methods always filter proposals simply based on their detection confidences. These heuristic rules result in a sharp reduction of the recall (*e.g.*, decrease to 80.77% as in Figure 2), and bring in the mismatch negligently. To illustrate this further, we show a concrete example in Figure 1. To ground the referent at the second stage, we hope that the proposals contain the referent `person` and its contextual object `pizza`. In contrast, the first-stage network only keeps bboxes with high detection confidence (*e.g.*, `knife`, `book`, and `cup`) as proposals, but actually misses the critical referent `person` (*i.e.*, the red bbox).

In this paper, we propose a novel algorithm Ref-NMS, to rectify the mismatch of detected proposals at the conventional first stage. In particular, for each expression, Ref-NMS regards all nouns in the expression as critical objects, and introduces a lightweight relatedness module to predict a probability score for each proposal to be a critical object. The higher predicted score denotes the higher relevance between a proposal and the expression. Then, we fuse the relatedness scores and classification scores, and exploit the fused scores as the suppression criterion in Non-Maximum Suppression (NMS). After NMS, we can filter out the proposals with little relevance to the expression. Finally, all proposals and the expression are fed into the second-stage grounding network, to obtain the referent prediction.

We demonstrate the significant performance gains of Ref-NMS on three challenging REG benchmarks. It's worth noting that the Ref-NMS can be generalized and easily integrated into any state-of-the-art two-stage method to further boost its performance on both REC and RES. Our method is robust and efficient, opening the door for many downstream applications such as multimodal summarization.

Related Work

Referring Expression Comprehension (REC). Current overwhelming majority of REC methods are in a two-stage manner: proposal generation and referent grounding. To the best of our knowledge, existing two-stage works all focus on the second stage. Specifically, they tend to design a more explainable reasoning process by structural modeling (Yu et al. 2018a; Liu et al. 2019c,a,b; Hong et al. 2019; Niu et al. 2019), or more effective multi-modal interaction mechanism (Wang et al. 2019; Yang, Li, and Yu 2020). However, their performance is strictly limited by the proposals from the first stage. Recently, another emerging direction to solve REC is in a one-stage manner (Chen et al. 2018; Yang et al. 2019b; Liao et al. 2020; Luo et al. 2020; Yang et al. 2020). Although one-stage methods achieve faster inference speed empirically, they come at a cost of lost interpretability and poor performance in composite expressions. In this paper, we rectify the overlooked mismatch in two-stage methods.

Referring Expression Segmentation (RES). Unlike REC, most of RES works are one-stage methods. They typically utilize a "concatenation-convolution" design to combine the two different modalities: they first concatenate the expression feature with visual features at each location, and then use several conv-layers to fuse the multimodal features for mask generation. To further improve mask qualities, they usually enhance their backbones with more effective features by multi-scale feature fusion (Margffoy-Tuay et al. 2018), feature progressive refinement (Li et al. 2018; Chen et al. 2019a; Huang et al. 2020), or novel attention mechanisms (Shi et al. 2018; Ye et al. 2019; Hu et al. 2020). Besides, with the development of two-stage instance segmentation (*e.g.*, Mask R-CNN (He et al. 2017)), two-stage REC methods can be extended to solve RES simply by replacing the object detection network at the second stage to an

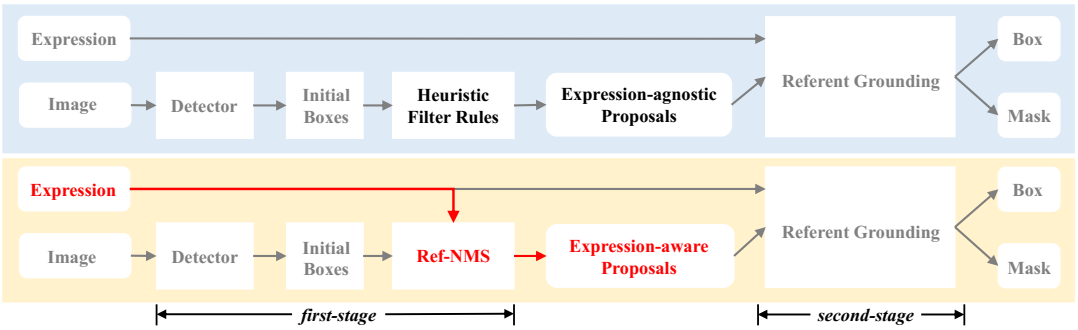

Figure 3: Upper: A typical two-stage REG framework, which uses heuristic filter rules to obtain expression-agnostic proposals at the first-stage, and feeds them into the second stage for referent grounding. Below: The Ref-NMS module can generate expression-aware proposals by considering the expression at the first stage.

instance segmentation network. Analogously, Ref-NMS can be easily integrated into any two-stage RES method.

Phrase Grounding. It is a task closely related to REC. There are also two types of solutions: proposal-free and proposal-driven methods. Different from REC, the queries in phrase grounding have two characteristics: 1) Simple. This relieves two-stage methods from complicated relational reasoning and allow them to accept more proposals (*e.g.*, $>$ 200 proposals)[1] at the second stage, which means two-stage phrase grounding methods doesn't suffer from the aforementioned recall drop problem. 2) Diverse. Efforts have been taken to address this problem by either using a object detector pre-trained on another large-scale dataset (Yu et al. 2018b) or re-generate proposals with respect to queries and mentioned objects (Chen, Kovvuri, and Nevatia 2017).

Non-Maximum Suppression (NMS). NMS is a de facto standard post-processing step adopted by numerous modern object detectors, which removes duplicate bboxes based on detection confidence. Except for the most prevalent GreedyNMS, multiple improved variants have been proposed recently. Generally, they can be categorized into three groups: 1) Criterion-based (Jiang et al. 2018; Tychsen-Smith and Petersson 2018; Tan et al. 2019; Yang et al. 2019a): they utilize other scores instead of classification confidence as the criterion to remove bboxes by NMS, *e.g.*, IoU scores. 2) Learning-based (Hosang, Benenson, and Schiele 2017; Hu et al. 2018): they directly learn an extra network to remove duplicate bboxes. 3) Heuristic-based (Bodla et al. 2017; Liu, Huang, and Wang 2019): they dynamically adjust the thresholds for suppression according to some heuristic rules. In this paper, we are inspired by the criterion-based NMS, and design the Ref-NMS, which uses both expression relatedness and detection confidence as the criterion.

Approach

Revisiting Two-Stage REG Framework

The two-stage framework is the most prevalent pipeline for REG. As shown in Figure 3, it consists of two separate stages: proposal generation at the first-stage and referent grounding at the second-stage.

[1]In contrast, the average number of proposals in REC is 10.

Proposal Generation. Given an image, current two-stage methods always resort to a well pre-trained detector to obtain a set of initially detected bboxes, and utilize an NMS to remove duplicate bboxes. However, even after NMS operation, there are still thousands of bboxes left (*e.g.*, each image in RefCOCO has an average of 3,500 detections). To relieve the burden of the following referent grounding step, all existing works further filter these bboxes based on their detection confidences. Although this heuristic filter rule can reduce the number of proposals, it also results in a drastic drop in the recall of both the referent and contextual objects (Detailed results are reported in Table 1.).

Referent Grounding. In the training phase, two-stage methods usually use the ground-truth regions in COCO as proposals, and the number is quite small (*e.g.*, each image in RefCOCO has an average of 9.84 ground-truth regions). For explainable grounding, state-of-the-art two-stage methods always compose these proposals into graph (Yang, Li, and Yu 2019; Wang et al. 2019) or tree (Liu et al. 2019a; Hong et al. 2019) structures, *i.e.*, as the number of proposals increases linearly, the number of computation increases exponentially. Therefore, in the test phase, it is a must for them to filter detections at the first stage.

Relatedness Module

An overview of the Ref-NMS model is shown in Figure 4. The core of Ref-NMS is the relatedness module. Given an image and a pre-trained detector, we can receive thousands of initial bboxes. To reduce the computation of the relatedness module, we first use a threshold δ to filter the bboxes with classification confidence, and obtain a filtered bbox set $\mathcal{B}$. For each bbox $\boldsymbol{b}_i \in \mathcal{B}$, we use a region visual encoder e_v (*i.e.*, an RoI Pooling layer and a convolutional head network) to extract the bbox feature $\boldsymbol{v}_i \in \mathbb{R}^v$. Meanwhile, for the referring expression Q, we use an expression encoder e_q (*i.e.*, a Bi-GRU) to output a set of word features $\{\boldsymbol{w}_1, ..., \boldsymbol{w}_{|Q|}\}$, where $\boldsymbol{w}_j \in \mathbb{R}^q$ is the j-th word feature. For each bbox $\boldsymbol{b}_i$, we use a soft-attention mechanism (Chen et al. 2017) to calculate a unique expression feature $\boldsymbol{q}_i$ by:

$$\begin{aligned} \boldsymbol{v}_i^a &= \text{MLP}_a(\boldsymbol{v}_i), \quad a_{ij} = \text{FC}_s([\boldsymbol{v}_i^a; \boldsymbol{w}_j]), \\ \alpha_{ij} &= \text{softmax}_j(a_{ij}), \quad \boldsymbol{q}_i = \textstyle\sum_j \alpha_{ij} \boldsymbol{w}_j, \end{aligned} \tag{1}$$

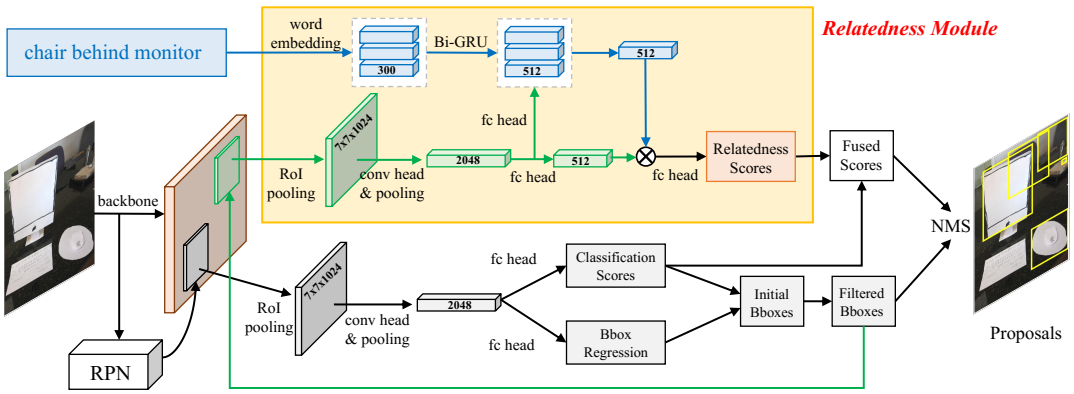

Figure 4: The overview of Ref-NMS model. Given an image, the model uses a pre-trained detector to generate thousands of initial bboxes. Then, hundreds of filtered bboxes and the expression are fed into the relatedness module to predict their relatedness scores. Lastly, the fused scores are used as the suppression criterion of NMS.

where MLP_a is a two-layer MLP mapping $\boldsymbol{v}_i \in \mathbb{R}^v$ to $\boldsymbol{v}_i^a \in \mathbb{R}^q$, FC_s is a FC layer to calculate the similarity between bbox feature $\boldsymbol{v}_i^a$ and word feature $\boldsymbol{w}_j$, and $[;]$ is a concatenation operation. Then, we combine the two modal features and predict the relatedness score r_i:

$$\begin{aligned} \boldsymbol{v}_i^b &= \text{MLP}_b(\boldsymbol{v}_i), \quad \boldsymbol{m}_i = \text{L2Norm}(\boldsymbol{v}_i^b \odot \boldsymbol{q}_i), \\ \hat{r}_i &= \text{FC}_r(\boldsymbol{m}_i), \quad r_i = \text{sigmoid}(\hat{r}_i), \end{aligned} \tag{2}$$

where MLP_b is a two-layer MLP mapping $\boldsymbol{v}_i \in \mathbb{R}^v$ to $\boldsymbol{v}_i^b \in \mathbb{R}^q$, $\odot$ is the element-wise multiplication, L2Norm represents l_2 normalization, and FC_r is a FC layer mapping $\boldsymbol{m}_i \in \mathbb{R}^q$ to $\hat{r}_i \in \mathbb{R}$.

Score Fusion. After obtaining the relatedness score r_i for bbox $\boldsymbol{b}_i$, we multiply r_i with the classification confidence c_i for bbox $\boldsymbol{b}_i$ from the original detector, and utilize the multiplication of two scores s_i as the suppression criterion of the NMS operation, *i.e.*, $s_i = r_i \times c_i$.

Training Objectives for Ref-NMS

To learn the relatedness score for each bbox, we need the ground-truth annotations for all mentioned instances (*i.e.*, both referent and contextual objects) in the expression. However, current REG datasets only have annotations about the referent. Thus, we need to generate pseudo ground-truths for contextual objects. Specifically, we first assign POS tags to each word in the expression using the spaCy POS tagger and extract all nouns in the expression. Then, we calculate the cosine similarity between GloVe embeddings of extracted nouns and categories of ground-truth regions in COCO[2]. Lastly, we use threshold γ to filter regions as the pseudo ground-truths.

In the training phase, we regard all the pseudo ground-truth bboxes and annotated referent bboxes as foreground bboxes. And we use two types of training objectives:

Binary XE Loss. For each bbox $\boldsymbol{b}_i \in \mathcal{B}$, if it has a high overlap (*i.e.*, IoU>0.5) with any foreground bbox, its ground-truth relatedness score r^* is set to 1, otherwise $r^* = 0$. Then the relatedness score prediction becomes a binary classification problem. We can use the binary cross-entropy (XE) loss as the training objective:

$$L = -\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} r_i^* \log(r_i) + (1 - r_i^*)\log(1 - r_i). \tag{3}$$

Ranking Loss. Generally, if a bbox has a higher IoU with foregound bboxes, the relatedness between the bbox and expression should be higher, *i.e.*, we can use the ranking loss as the training objectives:

$$L = \frac{1}{N}\sum_{(\boldsymbol{b}_i, \boldsymbol{b}_j), \rho_i < \rho_j} \max(0, r_i - r_j + \alpha), \tag{4}$$

where ρ_i denotes the largest IoU value between bbox $\boldsymbol{b}_i$ and foreground bboxes, N is the total number of pos-neg training pairs, and α is a constant to control the ranking margin, set as 0.1. To select the pos-neg pair $(\boldsymbol{b}_i, \boldsymbol{b}_j)$, we follow the sampling-after-splitting strategy (Tan et al. 2019). Specifically, we first divide the bbox set $\mathcal{B}$ into 6 subsets based on a quantization q-value: $q_i = \lceil \max(0, \rho_i - 0.5)/0.1 \rceil$, *i.e.*, the bbox with higher IoU value has larger q-value. Then, all bboxes with $\rho > 0.5$ are selected as positive samples. For each positive sample, we rank the top-h bboxes as negative samples based on predicted relatedness scores from the union of subsets with smaller q-value.

Experiments

Experimental Settings and Details

Datasets. We evaluate the Ref-NMS on three challenging REG benchmarks: 1) **RefCOCO** (Yu et al. 2016): It consists of 142,210 referring expressions for 50,000 objects in 19,994 images. These expressions are collected in an interactive game interface (Kazemzadeh et al. 2014), and the average length of each expression is 3.5 words. All expression-referent pairs are split into train, val, testA, and testB sets. The testA set contains the images with multiple people and the testB set contains the images with multiple objects. 2) **RefCOCO+** (Yu et al. 2016): It consists of 141,564 referring expressions for 49,856 objects in 19,992 images. Similar to

[2]Two-stage methods always use an object detector pretrained on COCO dataset. Thus, we don't use extra or more annotations.

	Ref-NMS	Referent								Contextual Objects							
		RefCOCO			RefCOCO+			RefCOCOg		RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val	test	val	testA	testB	val	testA	testB	val	test
N=100		97.60	97.81	96.58	97.79	97.78	96.99	97.18	96.91	90.14	89.85	90.53	89.53	88.47	90.69	90.56	90.30
	B	97.75	98.59	97.08	97.96	98.39	97.50	97.61	97.44	90.38	90.31	90.64	89.67	88.88	91.04	90.36	90.37
	R	97.62	98.02	96.78	97.71	98.06	97.14	97.18	97.08	90.22	89.83	90.63	89.70	88.62	90.71	90.67	90.30
Real		88.84	93.99	80.77	90.71	94.34	84.11	87.83	87.88	74.97	78.60	70.19	76.34	77.45	73.52	75.69	75.87
	B	**92.51**	**95.56**	**88.28**	**93.42**	**95.86**	**88.95**	**90.28**	**90.34**	**78.75**	**80.14**	**76.47**	**78.44**	**78.82**	**77.49**	76.12	76.57
	R	90.50	94.75	83.87	91.62	95.14	86.42	89.01	88.96	76.79	79.12	72.99	77.66	78.44	75.59	**76.68**	**76.73**

Table 1: Recall (%) of the referent and contextual objects. The baseline detector is the ResNet-101 based Mask R-CNN with plain GreedyNMS. B denotes the Ref-NMS with binary XE loss, R denotes the Ref-NMS with ranking loss. Real denotes the real case used in the state-of-the-art two-stage methods.

Models	Referring Expression Comprehension								Referring Expression Segmentation							
	RefCOCO			RefCOCO+			RefCOCOg		RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test	val	testA	testB	val	testA	testB	val	test
MAttNet (Yu et al. 2018a)	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61
MAttNet†	76.92	81.19	69.58	65.90	**71.53**	57.23	67.52	67.55	57.14	62.34	51.48	47.30	**52.37**	41.14	48.28	49.01
+Ref-NMS B	**78.82**	**82.71**	**73.94**	**66.95**	71.29	**58.40**	*68.89*	**68.67**	**59.75**	**63.48**	**55.66**	**48.39**	51.57	**42.56**	*49.54*	**50.38**
+Ref-NMS R	*77.98*	*82.02*	*71.64*	*66.64*	*71.36*	*58.01*	**69.16**	*67.63*	*58.32*	*62.96*	*53.68*	*47.87*	*51.85*	*41.41*	**50.13**	*49.07*
NMTree (Liu et al. 2019a)	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44	56.59	63.02	52.06	47.40	53.01	41.56	46.59	47.88
NMTree†	76.54	81.32	69.66	66.65	71.48	57.74	65.65	65.94	56.99	*62.88*	51.90	47.75	*52.36*	41.86	46.19	47.41
+Ref-NMS B	**78.67**	**82.09**	**73.78**	**67.15**	*71.76*	*58.70*	**67.30**	**66.93**	**59.95**	**63.25**	**55.64**	**48.68**	52.30	**42.64**	**48.14**	**48.59**
+Ref-NMS R	*77.81*	*81.69*	*71.78*	*67.03*	**71.78**	**58.79**	*66.81*	*66.31*	*58.42*	62.69	*53.60*	*48.27*	**52.65**	*42.18*	*47.72*	*48.09*
CM-A-E (Liu et al. 2019c)	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67	—	—	—	—	—	—	—	—
CM-A-E†	78.35	83.12	71.32	68.19	73.04	58.27	69.10	69.20	58.23	64.60	53.14	49.65	**53.90**	41.77	49.10	50.72
+Ref-NMS B	**80.70**	**84.00**	**76.04**	*68.25*	**73.68**	**59.42**	**70.55**	**70.62**	**61.46**	**65.55**	**57.41**	*49.76*	*53.84*	**42.66**	**51.21**	**51.90**
+Ref-NMS R	*79.55*	*83.58*	*73.62*	**68.51**	*73.14*	*58.38*	*69.77*	*70.01*	*59.72*	*64.87*	*55.63*	**49.86**	52.62	*41.87*	*50.13*	*51.44*

Table 2: Performances of different architectures on REC and RES. The metrics are top-1 accuracy (%) for REC and overall IoU (%) for RES. All baselines use the ResNet-101 based Mask R-CNN as first-stage networks. The best and second best methods under each setting are marked in bold and italic fonts, respectively. † denotes the results from our implementations.

RefCOCO, these expressions are collected from the same game interface, and have train, val, testA, and testB splits. 3) **RefCOCOg** (Mao et al. 2016): It consists of 104,560 referring expressions for 54,822 objects in 26,711 images. These expressions are collected in a non-interactive way, and the average length of each expression is 8.4 words. We follow the same split as (Nagaraja, Morariu, and Davis 2016).

Evaluation Metrics. For the REC task, we use the top-1 accuracy as evaluation metric. When the IoU between bbox and ground truth is larger than 0.5, the prediction is correct. For the RES task, we use the overall IoU and Pr@X (the percentage of samples with IoU higher than X)[3] as metrics.

Implementation Details. We build a vocabulary for each dataset by filtering the words less than 2 times, and exploit the 300-d GloVe embeddings as the initialization of word embeddings. We use an "unk" symbol to replace all words out of the vocabulary. The largest length of sentences is set to 10 for RefCOCO and RefCOCO+, 20 for RefCOCOg. The hidden size of the encoder e_q is set to 256. For encoder e_v, we use the same head network of the Mask R-CNN with ResNet-101 backbone[4] as prior works (Yu et al. 2018a), and utilize the pre-trained weights as initialization. The weights of the original detector (*i.e.*, the gray part in Figure 4) are fixed during training. The whole model is trained with Adam optimizer. The learning rate is initialized to 4e-4 and 5e-3 for the head network and the rest of network. We set the batch size as 8. The thresholds δ and γ are set to 0.05 and 0.4, respectively. For ranking loss, the top-h is set to 100.

Recall Analyses of Critical Objects

Settings. To evaluate the effectiveness of the Ref-NMS to improve the recall of both referent and contextual objects, we compare Ref-NMS with plain GreedyNMS used in the baseline detector (*i.e.*, ResNet-101 based Mask R-CNN). Since we only have annotated ground-truth bboxes for the referent, we calculate the recall of pseudo ground-truths to approximate the recall of contextual objects. The results are reported in Table 1, and more detailed results are provided in the supplementary materials.

Results. From Table 1, we have the following observations. When using top-100 bboxes as proposals, all three methods can achieve near-perfect recall ($\approx 97\%$) for the referent and acceptable recall ($\approx 90\%$) for the contextual objects, respectively. However, when the number of proposals decreases to a very small number (*e.g.*, < 10 in the real case), the recall of the baseline all drops significantly (*e.g.*, 15.81% for the referent and 20.34% for the contextual objects on RefCOCO testB). In contrast, Ref-NMS can help narrow the gap over all dataset splits. Especially, the improvement is more obvious in the testB set (*e.g.*, 7.51% and 4.85% absolute gains for the recall of referent on RefCOCO and Ref-

[3]Due to the limited space, all RES results with the Pr@X metric are provided in the supplementary materials.

[4]https://github.com/lichengunc/mask-faster-rcnn

	Models	Venue	Backbone	RefCOCO			RefCOCO+			RefCOCOg	
				val	testA	testB	val	testA	testB	val	test
one-s.	SSG (Chen et al. 2018)	*arXiv'18*	darknet53	—	76.51	67.50	—	62.14	49.27	58.80	—
	FAOA (Yang et al. 2019b)	*ICCV'19*	darknet53	71.15	74.88	66.32	56.86	61.89	49.46	59.44	58.90
	RCCF (Liao et al. 2020)	*CVPR'20*	dla34	—	81.06	71.85	—	70.35	56.32	—	65.73
	RSC-Large (Yang et al. 2020)	*ECCV'20*	darknet53	77.63	80.45	72.30	63.59	68.36	56.81	67.30	67.20
two-s.	VC (Zhang, Niu, and Chang 2018)	*CVPR'18*	vgg16	—	73.33	67.44	—	58.40	53.18	—	—
	ParalAttn (Zhuang et al. 2018)	*CVPR'18*	vgg16	—	75.31	65.52	—	61.34	50.86	—	—
	LGRANs (Wang et al. 2019)	*CVPR'19*	vgg16	—	76.60	66.40	—	64.00	53.40	—	—
	DGA (Yang, Li, and Yu 2019)	*ICCV'19*	vgg16	—	78.42	65.53	—	69.07	51.99	—	63.28
	NMTree (Liu et al. 2019a)	*ICCV'19*	vgg16	71.65	74.81	67.34	58.00	61.09	53.45	61.01	61.46
	MAttNet (Yu et al. 2018a)	*CVPR'18*	res101	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27
	RvG-Tree (Hong et al. 2019)	*TPAMI'19*	res101	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51
	NMTree (Liu et al. 2019a)	*ICCV'19*	res101	76.41	81.21	70.09	66.46	72.02	57.52	65.87	66.44
	CM-A-E (Liu et al. 2019c)	*CVPR'19*	res101	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67
	CM-A-E+Ref-NMS	*AAAI'21*	res101	**80.70**	**84.00**	**76.04**	**68.25**	**73.68**	**59.42**	**70.55**	**70.62**

Table 3: Top-1 accuracies (%) of state-of-the-art models on referring expression comprehension.

	Models	Venue	RefCOCO			RefCOCO+			RefCOCOg	
			val	testA	testB	val	testA	testB	val	test
one-s.	STEP (Chen et al. 2019a)	*ICCV'19*	60.04	63.46	57.97	48.19	52.33	40.41	—	—
	BRINet (Hu et al. 2020)	*CVPR'20*	60.98	62.99	59.21	48.17	52.32	42.41	—	—
	CMPC (Huang et al. 2020)	*CVPR'20*	61.36	64.53	59.64	49.56	53.44	43.23	—	—
	MCN (Luo et al. 2020)	*CVPR'20*	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40
two-s.	MAttNet (Yu et al. 2018a)	*CVPR'18*	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61
	NMTree (Liu et al. 2019a)	*ICCV'19*	56.59	63.02	52.06	47.40	53.01	41.56	46.59	47.88
	CM-A-E† (Liu et al. 2019c)	*CVPR'19*	58.23	64.60	53.14	49.65	**53.90**	41.77	49.10	50.72
	CM-A-E+Ref-NMS	*AAAI'21*	**61.46**	**65.55**	**57.41**	**49.76**	53.84	**42.66**	**51.21**	**51.90**

Table 4: Overall IoU (%) of state-of-the-art models on referring expression segmentation. All methods utilize ResNet-101 as backbone. † denotes that the results are from our reimplementation. Note that since one-stage RES and two-stage RES models are always pretrained on different datasets, the comparision between one-stage and two-stage models are not absolutely fair.

COCO+), where the categories of referents are more diverse and the recalls are relatively lower.

Architecture Agnostic Generalization

Settings. Since the Ref-NMS model is agnostic to the second stage network, it can be easily integrated into any referent grounding architectures. To evaluate the effectiveness and generality of Ref-NMS to boost the grounding performance of different backbones, we incorporated the Ref-NMS into multiple SOTA two-stage methods: **MAttNet** (Yu et al. 2018a) , **NMTree** (Liu et al. 2019a), and **CM-A-E** (Liu et al. 2019c). All results are reported in Table 2.

Results. From Table 2, we can observe that both variants of Ref-NMS can consistently improve the grounding performance over all three backbones on both REC and RES. The improvement is more significant on the testB set (*e.g.*, 4.72% and 3.23% absolute performance gains for CM-A-E in REC and RES), which meets our expectation, *i.e.*, the improvements of grounding performance have a strong positive correlation with the improvements of the recall of critical objects. Compared between two variants of Ref-NMS, in most of cases, Ref-NMS B achieves better grounding performance. We argue that the reason may come from the imbalance of positive and negative samples in each level.

Comparisons with State-of-the-Arts

We incorporate Ref-NMS (with binary XE loss) into model CM-A-E, which is dubbed **CM-A-E+Ref-NMS**, and compare it against the state-of-the-art REC and RES methods.

Settings. For the state-of-the art REC methods, from the viewpoint of one-stage and two-stage, we can group them into: 1) Two-stage methods: **VC** (Zhang, Niu, and Chang 2018), **ParalAttn** (Zhuang et al. 2018),**LGRANs** (Wang et al. 2019), **DGA** (Yang, Li, and Yu 2019), **NMTree** (Liu et al. 2019a), **MAttNet** (Yu et al. 2018a), **RvG-Tree** (Hong et al. 2019), and **CM-A-E** (Liu et al. 2019c); 2) one-stage methods: **SSG** (Chen et al. 2018), **FAOA** (Yang et al. 2019b), **RCCF** (Liao et al. 2020), and **RSC-Large** (Yang et al. 2020). Analogously, for the state-of-the-art RES methods, we group them into: 1) Two-stage methods: **MAttNet** (Yu et al. 2018a), **NMTree** (Liu et al. 2019a), and **CM-A-E** (Liu et al. 2019c); 2) one-stage methods: **STEP** (Chen et al. 2019a), **BRINet** (Hu et al. 2020), **CMPC** (Huang et al. 2020), and **MCN** (Luo et al. 2020).

Results. The REC and RES results are reported in Table 3 and Table 4. For the REC, CM-A-E+Ref-NMS achieves a new record-breaking performance that is superior to all existing REC methods on three benchmarks. Ref-NMS improves the strong baseline CM-A-E with an average of 2.64%, 0.53%, and 2.26% absolute performance gains over RefCOCO, RefCOCO+, and RefCOCOg, respectively. For the RES, CM-A-E+Ref-NMS achieves a new state-of-the-art performance of two-stage methods over most of the dataset splits. Similarly, Ref-NMS improves CM-A-E with an average of 2.82%, 0.31%, and 1.65% absolute performance gains over the three datasets.

Figure 5: Qualitative REC results on RefCOCOg showing comparisons between correct (green tick) and wrong referent grounds (red cross) by CM-A-E and CM-A-E+Ref-NMS. (a): The input image and referring expressions. (b): The visualiation of word attention weights α (*cf.*, Eq. (1)) for each referent object. (c): The annotated referent ground-truth bbox (red) and generated pseudo ground-truth bboxes for contextual objects (green). (d) and (e) denote the proposals and final grounding results from two methods. We only show the proposals and the final predicted referent bbox is illustrated in dash line. The denotations of bbox colors are as follows. Red: The bbox hits (IoU>0.5) the referent ground-truth bbox; Green: The bboxes hit the pseudo ground-truth bboxes; Blue: The false positive proposal predictions.

Qualitative Results

We illustrate the qualitative results between CM-A-E+Ref-NMS and baseline CM-A-E on REC in Figure 5. From the results in line (b), we can observe that Ref-NMS can assign high attention weights to words that are more relevant to individual referents (*e.g.*, umbrella, man, and zebra). The results in line (c) show that the generated pseudo ground-truth bboxes can almost contain all contextual objects in the expression, except a few objects whose categories are far different from the categories of COCO (*e.g.*, sweater, armrest, and grass). By comparing the results between line (d) and line (e), we have the following observations: 1) The baseline method always detects more false-positive proposals (*i.e.*, the blue bboxes), and misses some critical objects (*i.e.*, the red and green bboxes). Instead, Ref-NMS helps the model generate more expression-aware proposals. 2) Even for the failed cases in CM-A-E+Ref-NMS (*i.e.*, the last two columns), Ref-NMS still generates more reasonable proposals (*e.g.*, with less false positive proposals), and the grounding errors mainly come from the second stage.

Conclusions and Future Works

In this paper, we focused on the two-stage referring expression grounding, and discussed the overlooked mismatch problem between the roles of proposals in different stages. Particularly, we proposed a novel approach dubbed Ref-NMS to calibrate this mismatch. Ref-NMS tackles the problem by considering the expression at the first stage, and learns a relatedness score between each detected proposal and the expression. The multiplication of the relatedness scores and classification scores serves as the suppression criterion for the NMS operation. Meanwhile, Ref-NMS is agnostic to the referent grounding step, and can be integrated into any state-of-the-art two-stage method. Moving forward, we plan to apply Ref-NMS into other proposal-drive tasks which suffer from the same mismatch issue, *e.g.*, video grounding (Xiao et al. 2021; Chen et al. 2020a), VQA (Chen et al. 2020b) and scene graph generation (Chen et al. 2019c).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U19B2043, 61976185), Zhejiang Natural Science Foundation (LR19F020002, LZ17F020001), Major Project of Zhejiang Social Science Foundation (21XXJC01ZD), and the Fundamental Research Funds for the Central Universities.

References

Akula, A. R.; Gella, S.; Al-Onaizan, Y.; Zhu, S.-C.; and Reddy, S. 2020. Words aren't enough, their order matters: On the Robustness of Grounding Visual Referring Expressions. In *ACL*.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *ICCV*, 2425–2433.

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-NMS–improving object detection with one line of code. In *ICCV*.

Chen, D.-J.; Jia, S.; Lo, Y.-C.; Chen, H.-T.; and Liu, T.-L. 2019a. See-through-text grouping for referring image segmentation. In *ICCV*, 7454–7463.

Chen, H.; Suhr, A.; Misra, D.; Snavely, N.; and Artzi, Y. 2019b. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*, 12538–12547.

Chen, K.; Kovvuri, R.; and Nevatia, R. 2017. Query-guided regression network with context policy for phrase grounding. In *ICCV*, 824–832.

Chen, L.; Lu, C.; Tang, S.; Xiao, J.; Zhang, D.; Tan, C.; and Li, X. 2020a. Rethinking the Bottom-Up Framework for Query-Based Video Localization. In *AAAI*, 10551–10558.

Chen, L.; Yan, X.; Xiao, J.; Zhang, H.; Pu, S.; and Zhuang, Y. 2020b. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, 10800–10809.

Chen, L.; Zhang, H.; Xiao, J.; He, X.; Pu, S.; and Chang, S.-F. 2019c. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, 4613–4623.

Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; and Chua, T.-S. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 5659–5667.

Chen, X.; Ma, L.; Chen, J.; Jie, Z.; Liu, W.; and Luo, J. 2018. Real-time referring expression comprehension by single-stage grounding network. In *arXiv*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*, 2961–2969.

Hong, R.; Liu, D.; Mo, X.; He, X.; and Zhang, H. 2019. Learning to compose and reason with language tree structures for visual grounding. *TPAMI* .

Hosang, J.; Benenson, R.; and Schiele, B. 2017. Learning non-maximum suppression. In *CVPR*, 4507–4515.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *CVPR*, 3588–3597.

Hu, R.; Rohrbach, M.; Andreas, J.; Darrell, T.; and Saenko, K. 2017. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 1115–1124.

Hu, R.; Rohrbach, M.; and Darrell, T. 2016. Segmentation from natural language expressions. In *ECCV*, 108–124.

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T. 2016. Natural language object retrieval. In *CVPR*, 4555–4564.

Hu, Z.; Feng, G.; Sun, J.; Zhang, L.; and Lu, H. 2020. Bi-Directional Relationship Inferring Network for Referring Image Segmentation. In *CVPR*, 4424–4433.

Huang, S.; Hui, T.; Liu, S.; Li, G.; Wei, Y.; Han, J.; Liu, L.; and Li, B. 2020. Referring Image Segmentation via Cross-Modal Progressive Comprehension. In *CVPR*, 10488–10497.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of localization confidence for accurate object detection. In *ECCV*, 784–799.

Kazemzadeh, S.; Ordonez, V.; Matten, M.; and Berg, T. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 787–798.

Kim, J.; Misu, T.; Chen, Y.-T.; Tawari, A.; and Canny, J. 2019. Grounding human-to-vehicle advice for self-driving vehicles. In *CVPR*, 10591–10599.

Li, R.; Li, K.; Kuo, Y.-C.; Shu, M.; Qi, X.; Shen, X.; and Jia, J. 2018. Referring image segmentation via recurrent refinement networks. In *CVPR*, 5745–5753.

Liao, Y.; Liu, S.; Li, G.; Wang, F.; Chen, Y.; Qian, C.; and Li, B. 2020. A Real-Time Cross-modality Correlation Filtering Method for Referring Expression Comprehension. In *CVPR*, 10880–10889.

Liu, C.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; and Yuille, A. 2017. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, 1271–1280.

Liu, D.; Zhang, H.; Wu, F.; and Zha, Z.-J. 2019a. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 4673–4682.

Liu, D.; Zhang, H.; Zha, Z.-J.; Wang, M.; and Sun, Q. 2019b. Joint Visual Grounding with Language Scene Graphs. In *arXiv*.

Liu, S.; Huang, D.; and Wang, Y. 2019. Adaptive nms: Refining pedestrian detection in a crowd. In *CVPR*, 6459–6468.

Liu, X.; Wang, Z.; Shao, J.; Wang, X.; and Li, H. 2019c. Improving referring expression grounding with cross-modal attention-guided erasing. In *CVPR*, 1950–1959.

Luo, G.; Zhou, Y.; Sun, X.; Cao, L.; Wu, C.; Deng, C.; and Ji, R. 2020. Multi-task Collaborative Network for Joint Referring Expression Comprehension and Segmentation. In *CVPR*, 10034–10043.

Mao, J.; Huang, J.; Toshev, A.; Camburu, O.; Yuille, A. L.; and Murphy, K. 2016. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 11–20.

Margffoy-Tuay, E.; Pérez, J. C.; Botero, E.; and Arbeláez, P. 2018. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, 630–645.

Nagaraja, V. K.; Morariu, V. I.; and Davis, L. S. 2016. Modeling context between objects for referring expression understanding. In *ECCV*, 792–807.

Niu, Y.; Zhang, H.; Lu, Z.; and Chang, S.-F. 2019. Variational Context: Exploiting Visual and Textual Context for Grounding Referring Expressions. *TPAMI* .

Shi, H.; Li, H.; Meng, F.; and Wu, Q. 2018. Key-word-aware network for referring expression image segmentation. In *ECCV*, 38–54.

Tan, Z.; Nie, X.; Qian, Q.; Li, N.; and Li, H. 2019. Learning to rank proposals for object detection. In *ICCV*, 8273–8281.

Tychsen-Smith, L.; and Petersson, L. 2018. Improving object localization with fitness nms and bounded iou loss. In *CVPR*, 6877–6885.

Wang, P.; Wu, Q.; Cao, J.; Shen, C.; Gao, L.; and Hengel, A. v. d. 2019. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *CVPR*, 1960–1968.

Xiao, S.; Chen, L.; Zhang, S.; Ji, W.; Shao, J.; Ye, L.; and Xiao, J. 2021. Boundary Proposal Network for Two-Stage Natural Language Video Localization. In *AAAI*.

Yang, C.; Ablavsky, V.; Wang, K.; Feng, Q.; and Betke, M. 2019a. Learning to Separate: Detecting Heavily-Occluded Objects in Urban Scenes. In *arXiv*.

Yang, S.; Li, G.; and Yu, Y. 2019. Dynamic graph attention for referring expression comprehension. In *ICCV*, 4644–4653.

Yang, S.; Li, G.; and Yu, Y. 2020. Graph-Structured Referring Expression Reasoning in The Wild. In *CVPR*, 9952–9961.

Yang, Z.; Chen, T.; Wang, L.; and Luo, J. 2020. Improving One-stage Visual Grounding by Recursive Sub-query Construction. In *ECCV*.

Yang, Z.; Gong, B.; Wang, L.; Huang, W.; Yu, D.; and Luo, J. 2019b. A fast and accurate one-stage approach to visual grounding. In *ICCV*, 4683–4693.

Ye, L.; Rochan, M.; Liu, Z.; and Wang, Y. 2019. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 10502–10511.

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L. 2018a. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 1307–1315.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *ECCV*, 69–85.

Yu, L.; Tan, H.; Bansal, M.; and Berg, T. L. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 7282–7290.

Yu, Z.; Yu, J.; Xiang, C.; Zhao, Z.; Tian, Q.; and Tao, D. 2018b. Rethinking diversified and discriminative proposal generation for visual grounding. In *IJCAI*.

Zhang, H.; Niu, Y.; and Chang, S.-F. 2018. Grounding referring expressions in images by variational context. In *CVPR*, 4158–4166.

Zhuang, B.; Wu, Q.; Shen, C.; Reid, I.; and van den Hengel, A. 2018. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*, 4252–4261.