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

Long Chen1∗ Wenbo Ma1∗ Jun Xiao1 Hanwang Zhang2 Wei Liu3 Shih-Fu Chang4
1DCD Lab, College of Computer Science, Zhejiang University
2Nanyang Technological University 3Tencent AI Lab 4Columbia University

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.

1 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) [14, 16, 43, 44], where the referents are localized by bounding boxes (bboxes). 2) Referring Expression Segmentation (RES) [15, 24, 34, 30], where the referents are localized by segmentation masks. Both tasks are important for many downstream high-level applications such as visual question answering [3], navigation [6], and autonomous driving [20].

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 [7, 40, 22], 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., 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 (a), 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 considering its

∗Long Chen and Wenbo Ma are co-first authors with equal contributions. ([longc, mwb]@zju.edu.cn)

Preprint. Work in progress.
We demonstrate the significant performance gains of Ref-NMS on three challenging REG benchmarks.

## 2 Related Work

**Referring Expression Comprehension (REC).** The current overwhelming majority of REC methods are in a two-stage manner [45, 46, 37, 39, 42, 32]: proposal generation and referent grounding.

2 For the RefCOCO-like datasets, two-stage methods can use ground-truth regions in COCO [23] as proposals.
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 [42, 27, 25, 10], or more effective multi-modal interaction mechanism [39, 37]. 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 [7, 40, 22]. 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 focus on the two-stage REC methods, and try to rectify the overlooked mismatch.

Phrase grounding [1, 8] is another closely related task to the REC. Similarly, there are two types of solutions: proposal-free and proposal-driven methods. In theory, Ref-NMS can also be applied to the proposal-driven methods and further boost their performance. We leave this for future works.

Referring Expression Segmentation (RES). Unlike REC, most of RES works are one-stage methods [15, 24, 34, 30, 21]. They typically resort to FCN [28] and 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 [30], feature progressive refinement [21, 5], or novel attention mechanisms [34, 41]. Besides, with the development of two-stage instance segmentation networks (e.g., Mask R-CNN [9]), two-stage REC methods can be extended to solve the RES problem simply by replacing the object detection network at the second stage to an instance segmentation network. Analogously, Ref-NMS can be easily integrated into any two-stage RES method.

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 in the last few years. Generally, they can be categorized into three groups: 1) Criterion-based [18, 36, 35, 38]: they utilize other scores instead of classification confidence as the criterion to remove bboxes by NMS, e.g., IoU scores. 2) Learning-based [12, 13]: they directly learn an extra network to remove duplicate bboxes. 3) Heuristic-based [4, 26]: 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.

3 Approach

In this section, we first revisit the typical two-stage REG framework in Section 3.1, and then introduce the details about Ref-NMS in Section 3.2, including the relatedness module and training objectives.

3.1 Revisiting Two-Stage REG Framework

The two-stage framework is the most prevalent pipeline for REG. As shown in Figure 2, it consists of two separate stages: proposal generation at the first-stage and referent grounding at the second-stage.

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 [39, 37] or tree [25, 10] 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.

### 3.2 Ref-NMS

#### 3.2.1 Relatedness Module

An overview of the Ref-NMS model is shown in Figure 3. 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 $\delta$ to filter the bboxes with classification confidence, and obtain a filtered bbox set $B$. For each bbox $b_i \in B$, we use a region visual encoder $e_v$ (i.e., an RPN Pooling layer and a convolutional head network) to extract the bbox feature $v_i \in \mathbb{R}^v$. 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 $\{w_1, \ldots, w_{|Q|}\}$, where $w_j \in \mathbb{R}^q$ is the $j$-th word feature.

For each bbox $b_i$, we use a soft-attention mechanism to calculate a unique expression feature $q_i$ by:

$$v_i^a = \text{MLP}_a(v_i), \quad a_{ij} = \text{FC}_s([v_i^a; w_j]), \quad \alpha_{ij} = \text{softmax}_j(a_{ij}), \quad q_i = \sum_j \alpha_{ij} w_j,$$

where $\text{MLP}_a$ is a two-layer MLP mapping $v_i \in \mathbb{R}^v$ to $v_i^a \in \mathbb{R}^a$, $\text{FC}_s$ is a FC layer to calculate the similarity between bbox feature $v_i^a$ and word feature $w_j$, and $[;]$ is a concatenation operation. Then, we combine the two modal features and predict the relatedness score $r_i$:

$$v_i^b = \text{MLP}_b(v_i), \quad m_i = \text{L2Norm}(v_i^b \odot q_i), \quad \hat{r}_i = \text{FC}_r(m_i), \quad r_i = \text{sigmoid}(\hat{r}_i),$$

where $\text{MLP}_b$ is a two-layer MLP mapping $v_i \in \mathbb{R}^v$ to $v_i^b \in \mathbb{R}^b$, $\odot$ is the element-wise multiplication, L2Norm represents $l_2$ normalization, and $\text{FC}_r$ is a FC layer mapping $m_i \in \mathbb{R}^b$ to $\hat{r}_i \in \mathbb{R}$.

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

#### 3.2.2 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 tager [11] and extract all nouns in the expression. Then, we calculate the cosine similarity between GloVe embeddings [33] of extracted nouns and categories of ground-truth regions in COCO. Lastly, we use threshold \( \gamma \) 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 following types of training objectives for the Ref-NMS:

**Binary XE Loss.** For each bbox \( b_i \in B \), 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:

\[
L = - \frac{1}{|B|} \sum_{i=1}^{|B|} r^*_i \log(r_i) + (1 - r^*_i) \log(1 - r_i).
\] (3)

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

\[
L = \frac{1}{N} \sum_{(b_i, b_j)} \max(0, r_i - r_j - \alpha),
\] (4)

where \( \rho_i \) denotes the largest IoU value between bbox \( b_i \) and foreground bboxes, \( N \) is the total number of pos-neg training pairs, and \( \alpha \) is a constant to control the ranking margin, set as 0.1. To select the pos-neg pair \((b_i, b_j)\), we follow the sampling-after-splitting strategy [35]. Specifically, we first divide the bbox set \( B \) into 6 subsets based on a quantization \( q \)-value: \( q_i = \left\lceil \max(0, \rho_i - 0.5) / 0.1 \right\rceil \), 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.

3.2.3 Implementation Details

**Language Settings.** We build a vocabulary for each dataset by filtering the words less than 2 times, and exploit the 300-d GloVe embeddings [33] 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 \( e_e \) is set to 256.

**Visual Settings.** For visual encoder \( e_e \), we use the same head network of the Mask R-CNN with ResNet-101 backbone\(^3\) as prior works [42], and utilize the pre-trained weights as initialization. The weights of the original detector (i.e., the gray part in Figure 3) are fixed during the training phase.

**Parameter Settings.** The whole model is trained with Adam optimizer. The initial 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 \( \delta \) and \( \gamma \) are set to 0.05 and 0.4, respectively. For ranking loss, the top-\( h \) is set to 100.

4 Experiments

4.1 Datasets and Evaluation Metrics

**Datasets.** We evaluate the Ref-NMS on three challenging REG datasets collected from the COCO images [23]: 1) RefCOCO [43]: It consists of 142,210 referring expressions for 50,000 objects in 19,994 images. These expressions are collected in an interactive game interface [19], and the average length of each expression is 3.5 words. All expression-referent pairs are split into train, validation, testA, and testB sets. The testA set contains the images with multiple objects, and the testB set contains images with multiple objects. 2) RefCOCO+ [43]: It consists of 141,564 referring expressions for 49,856 objects in 19,992 images. Similar to RefCOCO, these expressions are collected from the same game interface, and have train, val, testA, and testB splits. Compared to RefCOCO, these expressions don’t include absolute locations. 3) RefCOCOg [29]: 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 [31].

\(^3\)Two-stage methods always use a detector pretrained on COCO. Thus, we don’t use extra or more annotations.

\(^4\)https://github.com/lichengunc/mask-faster-rcnn
we calculate the recall of pseudo ground-truths to approximate the recall of contextual objects. The results are reported in Table 2.

To evaluate the effectiveness and generality of Ref-NMS to boost the grounding performance of different backbones, we incorporated the Ref-NMS into multiple state-of-the-art two-stage methods. Since the Ref-NMS model is agnostic to the second stage network, it can be easily integrated into any referent grounding architectures.

### 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       | Ref-NMS | Referring Expression Segmentation |
|--------------|---------|-----------------------------------|
| MAiNet [42]  | 90.14   | 79.54                             |
| MAttNet e    | 90.00   | 79.62                             |
| +Ref-NMS B   | 90.12   | 79.75                             |
| +Ref-NMS R   | 90.12   | 79.75                             |
| NMTree [25]  | 90.12   | 79.75                             |
| +Ref-NMS B   | 90.12   | 79.75                             |
| +Ref-NMS R   | 90.12   | 79.75                             |
| CM-A-E [27]  | 90.12   | 79.75                             |
| +Ref-NMS B   | 90.12   | 79.75                             |
| +Ref-NMS R   | 90.12   | 79.75                             |
| CM-A-E [27]  | 90.12   | 79.75                             |
| +Ref-NMS B   | 90.12   | 79.75                             |
| +Ref-NMS R   | 90.12   | 79.75                             |

### 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 according formats. † denotes our implementation.

**Evaluation Metrics.** For the REC task, we use the top-1 accuracy as the 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) metrics.

### 4.2 Ablation Studies

#### 4.2.1 Recall Analyses of the Referent and Contextual 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 (≈ 97%) for the referent and acceptable recall (≈ 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 RefCOCO+).

#### 4.2.2 Architecture Agnostic

**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 state-of-the-art two-stage methods including: MAiNet [42], NMTree [25], and CM-A-E [27]. All results are reported in Table 2.

---

*Due to the limited space, all RES results with the Pr@X metric are provided in the supplementary materials.*
We illustrate the qualitative results between CM-A-E+Ref-NMS and baseline CM-A-E on REC in Figure 4. From the results in line (b), we can observe that Ref-NMS can assign high attention weights to areas with common features and significantly improve 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.

### 4.3 Comparisons with State-of-the-Arts

We incorporate Ref-NMS (with the binary XE loss) into the model CM-A-E [27], which is dubbed CM-A-E+Ref-NMS, and compare it against the state-of-the-art methods on both REC and RES.

**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 [45], ParalAttn [46], LGRANs [37], DGA [39], NMTree [25], MAttNet [42], RVG-Tree [10], and CM-A-E [27]; 2) one-stage methods: SSG [7], FAOA [40], and RCCF [22]. Analogously, for the state-of-the-art RES methods, we group them into: 1) Two-stage methods: MAttNet [42], NMTree [25], and CM-A-E [27]; 2) one-stage methods: RMI [24], DMN [30], RNN [21], CMSA [41], ParalAttn [46], BRINet [17], and STEP [5], denoted as CM-A-E+Ref-NMS.

**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.

We illustrate the qualitative results between CM-A-E+Ref-NMS and baseline CM-A-E on REC in Figure 4. From the results in line (b), we can observe that Ref-NMS can assign high attention weights...
Figure 4: 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 visualisation of word attention weights $\alpha$ (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.

On more relevant words 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.

5 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. We validated the effectiveness through extensive comparative and ablative experiments. Moving forward, we plan to 1) extend Ref-NMS into bounding boxes from Selective Search or EdgeBox, instead of pre-trained detectors; and 2) apply Ref-NMS into other proposal-drive tasks which suffer from the same mismatch issue, e.g., phrase grounding, visual relationship detection, and VQA.
References

[1] Hassan Akbari, Svebor Karaman, Surabhi Bhargava, Brian Chen, Carl Vondrick, and Shih-Fu Chang. Multi-level multimodal common semantic space for image-phrase grounding. In CVPR, 2019.

[2] Arjun R Akula, Spandana Gella, Yaser Al-Onaizan, Song-Chun Zhu, and Siva Reddy. Words aren’t enough, their order matters: On the robustness of grounding visual referring expressions. In ACL, 2020.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In ICCV, 2015.

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms—improving object detection with one line of code. In ICCV, 2017.

[5] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In ICCV, 2019.

[6] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In CVPR, 2019.

[7] Xinpeng Chen, Lin Ma, Jingyuan Chen, Zequn Jie, Wei Liu, and Jiebo Luo. Real-time referring expression comprehension by single-stage grounding network. In arXiv, 2018.

[8] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (seqground). In CVPR, 2019.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In ICCV, 2017.

[10] Richang Hong, Daqing Liu, Dipendra Misra, Noah Snavely, and Yoav Artzi. Learning to compose and reason with language tree structures for visual grounding. TPAMI, 2019.

[11] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[12] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In CVPR, 2017.

[13] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In CVPR, 2018.

[14] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In CVPR, 2017.

[15] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In ECCV, 2016.

[16] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In CVPR, 2016.

[17] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In CVPR, 2020.

[18] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In ECCV, 2018.

[19] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In EMNLP, 2014.

[20] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In CVPR, 2019.

[21] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In CVPR, 2018.

[22] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In CVPR, 2020.

[23] 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 ECCV, 2014.
[24] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In ICCV, 2017. 1, 3, 7

[25] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In ICCV, 2019. 2, 3, 4, 6, 7, 12, 13

[26] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In CVPR, 2019. 3

[27] Xihui Liu, Zhihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In CVPR, 2019. 2, 3, 6, 7, 12, 13, 14

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015. 3

[29] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In CVPR, 2016. 5

[30] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In ECCV, 2018. 1, 3, 7

[31] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In ECCV, 2016. 5

[32] Yulei Niu, Hanwang Zhang, Zhiwu Lu, and Shih-Fu Chang. Variational context: Exploiting visual and textual context for grounding referring expressions. TPAMI, 2019. 2

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In EMNLP, 2014. 5

[34] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In ECCV, 2018. 1, 3

[35] Zhiyu Tan, Xuecheng Nie, Qi Qian, Nan Li, and Hao Li. Learning to rank proposals for object detection. In ICCV, 2019. 3, 5

[36] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In CVPR, 2018. 3

[37] 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 CVPR, 2019. 2, 3, 4, 7

[38] Chenhongyi Yang, Vitaly Ablavsky, Kaihong Wang, Qi Feng, and Margrit Betke. Learning to separate: Detecting heavily-occluded objects in urban scenes. In arXiv, 2019. 3

[39] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In ICCV, 2019. 2, 3, 4, 7

[40] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In ICCV, 2019. 1, 3, 7

[41] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In CVPR, 2019. 3, 7

[42] 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 CVPR, 2018. 2, 3, 5, 6, 7, 12, 13, 14

[43] Licheng Yu, Patrick Poisson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In ECCV, 2016. 1, 5

[44] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In CVPR, 2017. 1

[45] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In CVPR, 2018. 2, 7

[46] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton van den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In CVPR, 2018. 2, 7
Appendix

This supplementary document is organized as follows:

- Section A reports detailed results about the recall of the referent and contextual objects.
- Section B reports detailed referring expression segmentation results with the Pr@X metric.
- Section C shows more qualitative results on RefCOCO, RefCOCO+ and RefCOCOg.
- Section D validates the effectiveness of using pseudo ground truth for contextual objects in the training phase.

A Detailed Results on Recall of the Referent and Contextual Objects

Table 5: Recall (%) of the referent with different number of proposals.

| # proposals | Model       | RefCOCO val | RefCOCO+ val | RefCOCOg val |
|------------|-------------|-------------|--------------|--------------|
|            |             | testA | testB | testA | testB | test   |
| N=100      | Baseline    | 97.60 | 97.81 | 96.58 | 97.79 | 97.78 | 96.99 | 97.18 | 96.91 |
|            | + Ref-NMS B | 97.75 | 98.59 | 97.08 | 97.96 | 98.39 | 97.50 | 97.61 | 97.44 |
|            | + Ref-NMS R | 97.62 | 98.02 | 96.78 | 97.71 | 98.06 | 97.14 | 97.18 | 97.08 |
| N=50       | Baseline    | 96.63 | 97.14 | 95.11 | 96.89 | 97.15 | 95.56 | 95.98 | 95.53 |
|            | + Ref-NMS B | 97.16 | 97.91 | 95.56 | 97.23 | 97.73 | 96.26 | 96.49 | 96.20 |
|            | + Ref-NMS R | 96.58 | 97.21 | 95.51 | 96.74 | 97.22 | 96.13 | 96.14 | 95.67 |
| N=20       | Baseline    | 94.54 | 95.85 | 91.44 | 94.81 | 95.90 | 91.92 | 93.42 | 92.96 |
|            | + Ref-NMS B | 95.63 | 96.89 | 92.86 | 95.53 | 96.65 | 93.56 | 94.04 | 94.25 |
|            | + Ref-NMS R | 94.69 | 96.32 | 91.93 | 94.92 | 96.37 | 92.66 | 93.85 | 93.19 |
| N=10       | Baseline    | 91.34 | 93.94 | 86.95 | 91.73 | 94.04 | 87.58 | 89.83 | 89.15 |
|            | + Ref-NMS B | 93.09 | 95.63 | 89.79 | 93.36 | 95.72 | 89.79 | 91.36 | 91.19 |
|            | + Ref-NMS R | 92.20 | 94.73 | 88.68 | 92.51 | 94.81 | 89.51 | 90.52 | 90.20 |
| Real case  | Baseline    | 88.84 | 93.99 | 80.77 | 90.71 | 94.34 | 84.11 | 87.83 | 87.88 |
|            | + Ref-NMS B | 92.51 | 95.56 | 88.28 | 93.42 | 95.86 | 88.95 | 90.28 | 90.34 |
|            | + Ref-NMS R | 90.50 | 94.75 | 83.87 | 91.62 | 95.14 | 86.42 | 89.01 | 88.96 |

Table 6: Recall (%) of the contextual objects with different number of proposals.

| # proposals | Model       | RefCOCO val | RefCOCO+ val | RefCOCOg val |
|------------|-------------|-------------|--------------|--------------|
|            |             | testA | testB | testA | testB | test   |
| N=100      | Baseline    | 90.14 | 89.85 | 90.53 | 89.53 | 88.47 | 90.69 | 90.56 | 90.30 |
|            | + Ref-NMS B | 90.38 | 90.31 | 90.64 | 89.67 | 88.88 | 91.04 | 90.36 | 90.37 |
|            | + Ref-NMS R | 90.22 | 89.83 | 90.63 | 89.70 | 88.62 | 90.71 | 90.67 | 90.30 |
| N=50       | Baseline    | 88.72 | 88.37 | 88.78 | 88.09 | 86.90 | 88.85 | 88.87 | 88.25 |
|            | + Ref-NMS B | 88.69 | 88.45 | 88.51 | 87.81 | 87.62 | 88.74 | 88.58 | 88.36 |
|            | + Ref-NMS R | 88.76 | 88.57 | 88.81 | 88.12 | 87.15 | 89.15 | 89.10 | 88.34 |
| N=20       | Baseline    | 84.10 | 84.74 | 84.18 | 83.35 | 82.67 | 84.32 | 84.68 | 84.17 |
|            | + Ref-NMS B | 84.51 | 85.39 | 83.81 | 83.69 | 83.47 | 83.74 | 84.65 | 84.40 |
|            | + Ref-NMS R | 84.55 | 85.01 | 84.21 | 83.87 | 83.12 | 84.73 | 85.10 | 84.24 |
| N=10       | Baseline    | 76.10 | 76.56 | 76.79 | 75.06 | 73.76 | 77.02 | 78.39 | 78.37 |
|            | + Ref-NMS B | 78.26 | 80.24 | 77.38 | 77.32 | 77.84 | 77.17 | 79.12 | 79.07 |
|            | + Ref-NMS R | 77.79 | 78.87 | 77.57 | 76.75 | 76.30 | 78.39 | 79.49 | 79.05 |
| Real case  | Baseline    | 74.97 | 78.60 | 70.19 | 76.34 | 77.45 | 73.52 | 75.69 | 75.87 |
|            | + Ref-NMS B | 78.75 | 80.14 | 76.47 | 78.44 | 78.82 | 77.49 | 76.12 | 76.57 |
|            | + Ref-NMS R | 76.79 | 79.12 | 72.99 | 77.66 | 78.44 | 75.59 | 76.68 | 76.73 |

The results on recall of the referent and contextual objects are reported on Table 5 and Table 6, respectively. For Table 5, as we can see, both two variants of Ref-NMS can help boost the recall of the referent on all dataset splits. More specifically, Ref-NMS B consistently achieves the best recall under all settings. Analogously, two variants can also improve the recall of contextual objects. However, for the contextual objects (cf. Table 6), it’s hard to pick a winner between the two variants but it’s safe to say the best recall is always achieved by either of them. In conclusion, these results concretely validate the effectiveness of the proposed Ref-NMS in boosting the recall of the referent and contextual objects.
The detailed referring expression segmentation results with Pr@X metric on RefCOCO, RefCOCO+ and RefCOCOg are shown in Table 7, Table 8, and Table 9, respectively. We can observe that the Ref-NMS can consistently improve the grounding performance of all baselines and dataset splits over most of the metric thresholds. In particular, it is worth noting that the Ref-NMS significantly outperform the baseline on RefCOCO testB split, RefCOCO+ testB split, where the category of the referent is more diverse, and all splits of RefCOCOg, where the referring expressions are more complex. We attribute these performance gains to the richer contextual information conserved in the proposals generated by Ref-NMS, which is consistent with our motivation.

### C More Qualitative Results on RefCOCO, RefCOCO+, and RefCOCOg

Qualitative results including the comparisons of proposals between the baseline (i.e., CM-A-E) and CM-A-E+Ref-NMS as well as the visualization of REC and RES predictions on RefCOCO and RefCOCOg are illustrated in Figure 5 and Figure 6, respectively. The RES mask prediction results on RefCOCOg is shown in Figure 7.

From these qualitative results, we have the following observations: 1) The generated pseudo ground-truth boxes can almost contain all contextual objects in the expression. 2) The baseline model tends to detect more false-positive proposals, and misses some critical objects. 3) Even in the failed cases a CM-A-E+Ref-NMS, Ref-NMS still generates express-aware proposals, and the grounding errors mainly come from the referent grounding step (i.e., second stage).

### D Effectiveness of Pseudo Ground Truths

To validate the effectiveness of pseudo ground truths, we further design a strong baseline: Ref-NMS without pseudo ground truth. Both methods are trained using the same set of hyper-parameters and tested with CM-A-E. We can observe that Ref-NMS performs better on all splits but RefCOCO testB, where the performance between these two methods is trivial (0.05%). These results validate the effectiveness of using pseudo ground truth for contextual objects in the training phase. Meanwhile, in
another perspective, Ref-NMS trained with single ground truth can be regarded as a specific one-stage referring expression comprehension model, which suggests that current one-stage REC methods are not qualified for generating proposals for two-stage REG methods.

Table 9: RES Performance (%) of different architectures with Pr@X metric on RefCOCOg. † denotes that the results are from our reimplementation using official released codes.

| Models                | val     | test     | val     | test     |
|-----------------------|---------|----------|---------|----------|
|                       | 0.5     | 0.6      | 0.7     | 0.8      | 0.9     | 0.5     | 0.6      | 0.7     | 0.8      | 0.9     |
| MAttNet [42]†         | 65.28   | 62.57    | 56.96   | 44.36    | 14.67   | 65.93   | 63.14    | 57.51   | 44.62    | 12.61   |
| +Ref-NMS B            | 66.03   | 62.70    | 57.09   | 44.34    | 14.73   | 66.58   | 63.49    | 57.82   | 45.21    | 13.03   |
| +Ref-NMS R            | 66.26   | 63.38    | 57.74   | 44.93    | 15.09   | 65.86   | 62.83    | 57.24   | 45.26    | 13.12   |
| NMTree [25]†          | 63.32   | 60.21    | 55.23   | 42.99    | 14.56   | 64.33   | 61.58    | 56.31   | 43.88    | 12.57   |
| +Ref-NMS B            | 64.40   | 61.01    | 55.54   | 43.10    | 14.69   | 64.94   | 61.87    | 56.34   | 44.28    | 12.98   |
| +Ref-NMS R            | 64.01   | 61.07    | 55.86   | 43.67    | 15.20   | 64.54   | 61.41    | 56.09   | 44.62    | 13.25   |
| CM-A-E [27]†          | 66.32   | 63.01    | 57.86   | 44.71    | 14.85   | 67.63   | 64.62    | 59.12   | 45.77    | 13.21   |
| +Ref-NMS B            | 67.75   | 64.11    | 58.35   | 45.40    | 15.16   | 68.49   | 65.28    | 59.51   | 46.50    | 13.52   |
| +Ref-NMS R            | 66.52   | 63.52    | 58.03   | 45.04    | 15.28   | 68.29   | 65.06    | 59.48   | 46.96    | 13.82   |

Table 10: Performance (%) on REC between baseline (i.e., Ref-NMS without pseudo ground-truths) and Ref-NMS.

| Model                  | RefCOCO |          | RefCOCO+ |          | RefCOCOg |          |
|------------------------|---------|----------|----------|----------|----------|----------|
| CM-A-E                 | 80.56   | 83.63    | 76.09    | 67.33    | 71.83    | 58.64    |
| +Ref-NMS w/o Pseudo GT| 69.14   | 68.76    | 69.14    | 68.76    | 69.14    | 68.76    |
| CM-A-E                 | 80.70   | 84.00    | 76.04    | 68.25    | 73.68    | 59.42    |
| +Ref-NMS               | 70.55   | 70.62    | 70.55    | 70.62    | 70.55    | 70.62    |

6Compared to the typical one-stage REC methods, this baseline is more lightweight and adaptable.
Figure 5: Qualitative REC and RES results on RefCOCO. (a): The referring expression and input image. (b): The annotated ground-truth bbox of the referent (marked in red) and the generated pseudo ground-truth bboxes of the contextual objects (marked in green). (c): The upper row demonstrates the proposals generated by off-the-shelf object detector and the REC predictions of the downstream CM-A-E[27]; The lower row demonstrates the two-stage RES predictions acquired using the REC predictions from the upper row, the detailed method of which is fully described in [42]. (d): Ref-NMS proposals, the REC and RES predictions from the downstream CM-A-E, arranged in the same format as (e). The predicted bbox in REC is shown in dashed line. The denotations of bbox colors are as follows. Red: The bbox hits (IoU>0.5) ground-truth bbox of the referent; Green: The bbox hits one of the pseudo ground-truth bboxes of the contextual objects; Blue: The false positive proposals.
Figure 6: Qualitative REC and RES results on RefCOCO+, arranged in the same style as Figure 5.

Figure 7: Qualitative RES results on RefCOCOg. The visualization of the proposals and REC results of the same examples used in Figure 4 in the main article.