Improving One-stage Visual Grounding by Recursive Sub-query Construction

Zhengyuan Yang[1] Tianlang Chen[1] Liwei Wang[2] Jiebo Luo[1]

[1]University of Rochester [2]Tencent AI Lab, Bellevue
{zyang39,tchen45,jluo}@cs.rochester.edu, liweiwang@tencent.com

Abstract. We improve one-stage visual grounding by addressing current limitations on grounding long and complex queries. Existing one-stage methods encode the entire language query as a single sentence embedding vector, *e.g.*, taking the embedding from BERT or the hidden state from LSTM. This single vector representation is prone to overlooking the detailed descriptions in the query. To address this query modeling deficiency, we propose a recursive sub-query construction framework, which reasons between image and query for multiple rounds and reduces the referring ambiguity step by step. We show our new one-stage method obtains $5.0\%, 4.5\%, 7.5\%, 12.8\%$ absolute improvements over the state-of-the-art one-stage approach on ReferItGame, RefCOCO, RefCOCO+, and RefCOCOg, respectively. In particular, superior performances on longer and more complex queries validates the effectiveness of our query modeling. Code is available at https://github.com/zyang-ur/ReSC.

Keywords: Visual grounding, Query modeling, Referring expressions

1 Introduction

Visual grounding aims to ground a natural language query onto a region of the image. There are mainly two threads of works in visual grounding: the two-stage approach [41,40,32,3,50,48] and one-stage approach [47,5,37]. Two-stage approaches first extract region proposals and then rank the proposals based on their similarities with the query. The recently proposed one-stage approach takes a different paradigm but soon becomes prevailing. The one-stage approach fuses visual-text features at image-level and directly predicts bounding boxes to ground the referred object. By densely sampling the possible object locations and reducing the redundant computation over region proposals, the one-stage methods [47,5,37] are simple, fast, and accurate.

In this paper, we improve the state-of-the-art one-stage methods by addressing their weaknesses in modeling long and complex queries. The overall advantage of our method is shown in Figure 1. Compared to the current state-of-the-art one-stage method [47], whose performance *drops dramatically on longer queries*, our approach achieves remarkably superior performance.

We analyze the limitations of current one-stage methods as follows. Existing one-stage methods [47,5,37] encode the entire query as a single embedding vector,

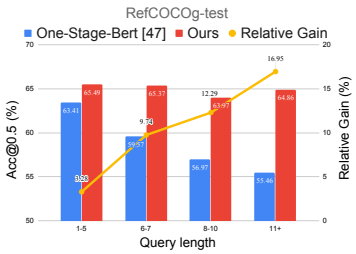

Fig. 1. The accuracy of previous one-stage methods (blue column) decreases on longer queries.

Fig. 2. Previous one-stage methods' representative failure cases of (a) overlooking detailed descriptions, (b) misinterpreting the query by keywords. Blue/ yellow boxes are the predicted regions [47]/ ground truths.

such as directly adopting the first token's embedding ([CLS]) from BERT [8,47] or aggregating hidden states from LSTM [12,47,5,37]. The single vector is then concatenated at all spatial locations with visual features to obtain the fused features for grounding box prediction. Modeling the entire language query as a single embedding vector tends to increase representation ambiguity, such as focusing on some words, yet overlooking other important ones. Such a problem potentially causes the loss of referring information, especially on those long and complex queries. For example in Figure 2 (a), the model seems to overlook detailed descriptions such as "sitting on the couch" or "looking tv," and grounds the wrong region with the head noun "man." As for Figure 2 (b), the model appears to look into the wrong word "mountain" and grounds the target without full consideration of the modifier of "water." Neglecting the query modeling problem, thus, causes the performance drop on long queries for the one-stage approach.

Several two-stage visual grounding works [42,45,46,24,48,27] have studied a similar query modeling problem. The main idea of these works is to link object regions with the parsed sub-queries to have a comprehensive understanding of the referring. Among them, MattNet [48] parses the query into the subject, location, and relationship phrases, and links each phrase with the related object regions for matching score computing. NMTREE [24] parses the query with a dependency tree parser [2] and links each tree node with a visual region. DGA [46] parses the query with text self-attention and links the text with regions via dynamic graph attention. Though elegant enough, these methods are designed intuitively for two-stage methods, requiring candidate region features to be extracted at the first stage. Since the main benefit of doing one-stage visual grounding is to avoid explicitly extracting candidate region features for the sake of computational cost, the query modeling in two-stage methods cannot be directly applied to the one-stage framework [47,5,37]. Therefore, in this paper, to address the query modeling problem in a unified one-stage framework, we propose the recursive sub-query construction framework.

When presented with a referring problem such as Figure 2 (a), humans tend to solve it by reasoning back-and-forth between the image and query for multiple

rounds and recursively reduce the referring ambiguity, *i.e.*, the possible region that contains the referred object. Inspired by this, we proposed to represent the intermediate understanding of the referring in each round as the *text-conditional visual feature*, which starts as the image feature and updated after multiple rounds, ending up as the fused visual-text feature ready for box prediction. In each round, the model constructs a new sub-query as a group of words attended with scores to refine the text-conditional visual feature. Gradually, with multiple rounds, our model reduces the referring ambiguity. Such a multi-round solution is in contrast to previous one-stage approaches that try to remember the entire query and ground the region in a single round.

Our framework recursively constructs sub-queries to refine the grounding prediction. Each round faces with two core problems that facilitate recursive reasoning, namely 1) how to construct the sub-query; and 2) how to refine the text-conditional visual feature with the sub-query. We propose a sub-query learner and a sub-query modulation network to solve the above two problems, respectively. They work alternately and recursively to reduce the referring ambiguity. Using the text-conditional visual features in the last round, a final output stage predicts bounding boxes to grounding the referred object.

We benchmark our framework on the ReferItGame [17], RefCOCO [49], RefCOCO+ [49], RefCOCOg [29] datasets, with $5.0\%, 4.5\%, 7.5\%, 12.8\%$ absolute improvements over the state-of-the-art one-stage method [47]. Meanwhile, our method runs fast at 38 FPS ($26ms$). Moreover, the relative gain curve according to the query length changes in Figure 1 shows the effectiveness of our approach in solving the aforementioned query modeling problem.

Our main contributions are:

- We improve one-stage visual grounding by addressing previous one-stage methods' limitations on grounding long and complex queries.
- We propose a recursive sub-query construction framework that recursively reduces the referring ambiguity with different constructed sub-queries.
- Our proposed method shows significantly improved results on multiple datasets and meanwhile maintains the real-time inference speed. Extensive experiments and ablations validate the effectiveness of our method.

2 Related Work

There exists two major categories of visual grounding methods: phrase localization [17,33,40] and referring expression comprehension [29,49,48,15,18]. Most previous visual grounding methods are composed of two stages. In the first stage, a number of region proposals are generated by an off-line module such as Edge-Box [53], selective search [39] or pretrained detectors [26,35,13]. In the second stage, each region is compared to the input query and outputs a similarity score. During inference, the region with the highest similarity score is output as the final prediction. Under the two-stage framework, various works explore different aspects to improve visual grounding, such as better exploiting attributes [25,48,27], object relationships [42,45,46,24], phrase co-occurrences [4,9,1], *etc.*

Recently, several works [47,5,37,20,28] propose a different paradigm of one-stage visual grounding. The primary motivation is to solve the two limitations of two-stage methods, *i.e.*, the performance cap caused by the sparsely sampled region proposals, and the slow inference speed caused by the region feature computation. Instead of explicitly extracting the features for all proposed regions, one-stage methods fuse the visual-text feature densely at all spatial locations, and directly predict bounding boxes to ground the target. Previous one-stage methods usually encode the query as a single language vector and concatenate the feature along the channel dimension of the visual feature. Despite the effectiveness of one-stage methods, modeling the language query as a single vector could lead to the loss of referring information, especially on long and complex queries. Though two-stage methods [42,45,46,24,48,27] had studied the similar problem of language query modeling, the explorations can not be directly applied to the one-stage approach given the intrinsic difference between two paradigms.

Besides, an intuitive alternative is to model the query phrase by the attention mechanism. Lin *et al.* [22] propose to extract sentence embedding with self-attention. Modeling query with attention mechanism is also explored in various vision-language tasks [19,48]. In experiments, we observe that our proposed multi-round solution performs better than the simple query attention method (*cf.* "Single/ Multi-head attention" and "Sub-query learner (ours)" in Table 3).

3 Approach

In this section, we will introduce our query modeling in a unified one-stage grounding framework. Previous one-stage grounding methods encode the language query as a C_l-dimension language feature and concatenate the text feature at all spatial locations with the visual feature $v \in R^{H\times W\times C_v}$. H, W, C_v are the height, width, and dimension of the visual feature. The visual feature and the text feature are usually mapped to the same dimension C before the concatenation. Extra convolutional layers then further refine the fused feature $f \in R^{H\times W\times 2C}$ and predict bounding boxes at each spatial location $H \times W$ to ground the target. Such single-round query modeling tends to overlook important query details and lead to incorrect predictions. The problems become increasingly severe on longer and more complex queries, as shown quantitatively in Figure 1.

To address this problem in a unified one-stage grounding system, we propose a recursive sub-query construction framework that step by step refines the visual-text feature $v^{(k)}$ to get better prediction. As shown in Figure 3, the initial feature $v^{(0)}$ is the image feature $v \in R^{H\times W\times C}$ encoded by the visual encoder [34]. In each round k, the framework constructs a new sub-query as a group of words attended with score vector $\vec{\alpha}^{(k)}$, and obtains the sub-query embedding $q^{(k)}$ to refine the visual-text feature. The framework ends up with the refined feature $v^{(K)}$ after K rounds, and predicts bounding boxes on each spatial location of $v^{(K)}$ to ground the target. We name $v^{(k)}$ the text-conditional visual feature.

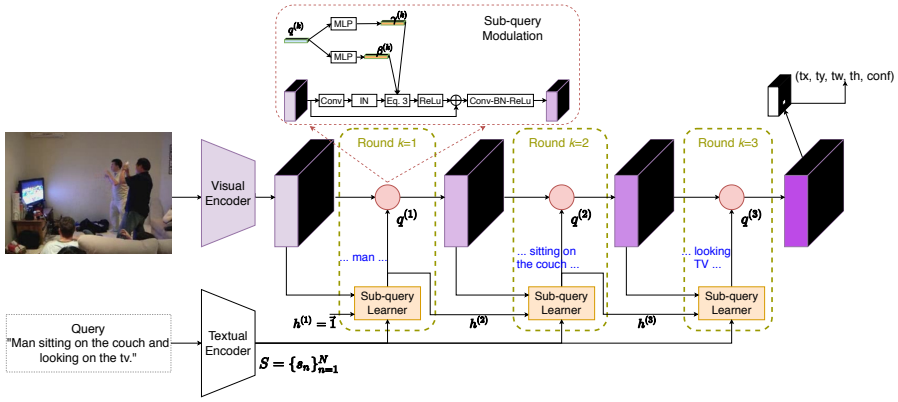

Fig. 3. The architecture for the recursive sub-query construction framework. In each round, the framework constructs a new sub-query to refine the text-conditional visual feature $v^{(k)}$ (shown in purple). $q^{(k)}$ is the feature for the constructed sub-query.

In each round, we address how to construct the sub-query and how to refine the feature $v^{(k)}$ with the sub-query embedding to better ground the target. For the first problem, we propose a sub-query learner in Section 3.1. The objective is to construct the sub-query that could best resolve the current referring ambiguity. We find it important to refer to the text-conditional visual feature $v^{(k)}$ in each round. For the second problem, we propose a sub-query modulation network that scales and shifts the feature $v^{(k)}$ with the sub-query feature. We introduce the details in Section 3.2. The sub-query learner and the modulation network operate alternately for multiple rounds, and recursively reduce the referring ambiguity. The feature $v^{(K)}$ in the last round contains the full object referring information and is used for target box prediction.

3.1 Sub-query Learner

Our method addresses the visual grounding problem as a multi-round reasoning process. In each round, the sub-query learner refers to the text-conditional visual feature $v^{(k)}$ and constructs the sub-query that could gradually reduce the referring ambiguity.

Given a language query of N words, the language encoder extracts the per-word query representation $S = \{s_n\}_{n=1}^{N}$, each with the dimension of C. As shown in Figure 3, the sub-query learner constructs a sub-query in each round k as a group of words attended with score vector $\vec{\alpha}^{(k)} = \{\alpha_n^{(k)}\}_{n=1}^{N}$ of length N. Apart from the query word features S, we find it particularly important to reference the current text-conditional visual feature $v^{(k-1)} \in R^{H \times W \times C}$ in sub-query construction, and thus take the average-pooled feature $\bar{v}^{(k-1)}$ of dimension C as an additional input to the learner. Moreover, the history of the previous sub-queries could help to avoid overemphasizing certain keywords. Given the

history of the previous sub-queries $\{\vec{\alpha}^{(i)}\}_{i=1}^{k-1}$, the history vector $\vec{h}^{(k)}$ represents the words that have been previously attended on and is calculates as

$$\vec{h}^{(k)} = \vec{1} - \min\left(\sum_{i=1}^{k-1} \vec{\alpha}^{(i)}, \vec{1}\right),$$

where $\vec{1}$ is the all-ones vector. Both $\vec{h}^{(k)}$ and $\vec{\alpha}^{(i)}$ are N-Dimension vectors with values range from 0 to 1. The sub-query learner takes the query word feature $\{s_n\}_{n=1}^{N}$, the text-conditional visual feature $\bar{v}^{(k-1)}$, and the history vector $\vec{h}^{(k)} = \{h_n^{(k)}\}_{n=1}^{N}$ to construct the sub-query for round k by predicting score vector $\vec{\alpha}^{(k)}$:

$$\alpha_n^{(k)} = softmax\left[W_{a1}^{(k)} \tanh\left(W_{a0}^{(k)} h_n^{(k)} (\bar{v}^{(k-1)} \odot s_n) + b_{a0}^{(k)}\right) + b_{a1}^{(k)}\right], \quad (1)$$

where $\odot$ represents hadamard product, and $W_{a0}, b_{a0}, W_{a1}, b_{a1}$ are learnable parameters. We compute the softmax over the N attention scores.

To guide the multi-round reasoning, explicit regularization is imposed on the word attention scores. Intuitively, the constructed sub-queries at each round should focus on different elements of the query, and in the end, most words in the query should have been looked at. Therefore, we add two regularization terms:

$$L_{div} = \left\|A^T A \odot (\mathbb{1} - I)\right\|_F^2, \quad L_{cover} = \left\|\vec{1} - \min\left(\sum_{i=1}^{K} \vec{\alpha}^{(i)}, \vec{1}\right)\right\|_1, \quad (2)$$

where matrix A is the predicted attention score matrix $A = \{\alpha_n^{(k)}\}_{n,k=1,1}^{N,K}$, $\mathbb{1}$ is the matrix of ones and I is an identity matrix. L_{div} avoids any words being focused on in more than one round and thus enforces the diversity. L_{cover} helps the model looks at all words in the query and thus improves the coverage.

The adopted technique of attention-based sub-query learning is related to previous compositional reasoning studies [16,46]. The major difference is that our method refers to the text-conditional visual feature $v^{(k)}$ to recursively construct the sub-query in each round. In contrast, the sub-query learning in previous studies [16,46] is purely based on the word feature $\{s_n\}_{n=1}^{N}$, and generates all sub-queries in prior to visual-text fusion.

3.2 Sub-query Modulation

In each round, the sub-query learner constructs a sub-query as a group of words attended by a score vector $\vec{\alpha}^{(k)}$, and generates the sub-query feature $q^{(k)}$ as

$$q^{(k)} = \sum_{n=1}^{N} \alpha_n^{(k)} s_n.$$

The goal for the sub-query modulation is to refine the text-conditional visual feature $v^{(k-1)}$ with the new sub-query feature $q^{(k)}$, such that the refined feature $v^{(k)}$ performs better in grounding box prediction.

Inspired by conditional normalization on image-level tasks [7,10,31], we encode the sub-query representation $q^{(k)}$ to modulate the previous visual-text representation $v^{(k-1)}$ by scaling and shifting. To be specific, $q^{(k)}$ is projected into a scaling vectors $\gamma^{(k)}$ and a shifting vector $\beta^{(k)}$ with two MLPs respectively:

$$\gamma^{(k)} = \tanh\left(W_{\gamma}^{(k)} q^{(k)} + b_{\gamma}^{(k)}\right), \quad \beta^{(k)} = \tanh\left(W_{\beta}^{(k)} q^{(k)} + b_{\beta}^{(k)}\right).$$

The text-conditional visual feature $v^{(k)}$ is then refined from $v^{(k-1)}$ with the two modulation vectors and extra learnable parameters:

$$v^{(k)}(i,j) = f_2\left\{ReLU\left[f_1(v^{(k-1)}(i,j)) \odot \gamma^{(k)} + \beta^{(k)}\right] + v^{(k-1)}(i,j)\right\}, \tag{3}$$

where (i,j) are the spatial coordinates, f_1, f_2 are learnable mapping layers as shown in Figure 3. f_1 consists of a 1×1 convolution followed by an instance normalization layer. f_2 consists of a 3×3 convolution followed by a batch normalization layer and ReLU activation. The grounding module takes the text-conditional visual feature in the final round $v^{(K)}$ as input, and predicts bounding boxes to ground the referred object. With the extra referring information in each sub-query $q^{(k)}$, we expect the modulation in each round to strength the feature of the referred object, and meanwhile suppress the ones for distracting objects and the background.

Our proposed sub-query modulation in Equation 3 has shared modulation vectors $\gamma^{(k)}, \beta^{(k)}$ over all spatial locations (i,j). One intuitive alternative is to predict different modulation vectors $\gamma^{(k)}(i,j), \beta^{(k)}(i,j)$ for each spatial location. This can be done by constructing sub-queries $\vec{\alpha}^{(k)}(i,j)$ for each location with the corresponding text-conditional visual feature $v^{(k-1)}(i,j)$. Despite using different modulation parameters at different spatial locations seems more intuitive, we show that the modulation along the channel dimension achieves the same objective and meanwhile is computationally efficient (*cf.* "Spatial-independent sub-query" and "Sub-query learner (ours)" in Table 3).

3.3 Framework Details

Visual and text feature encoder. We resize the input image to $3\times256\times256$ and use Darknet-53 [34] pretrained on COCO object detection [21] as the visual encoder. We adopt the visual feature from the 102-th convolutional layer that has a dimension $32\times32\times256$. We map the raw visual feature into the visual input $v^{(0)}$ with a 1×1 convolutional layer with batch normalization and ReLU. We set the shared dimension $C = 512$.

We encode the each word in the query as a $768D$ vector with the uncased base version of BERT [8,43]. We sum the representations for each word in the last four layers and map the features with two fully connected layers to obtain the representation $S = \{s_n\}_{n=1}^{N}$. N is the number of query words and does not include special tokens such as [CLS], [SEP] and [PAD].

Grounding module. The grounding module takes the visual-text feature $v^{(K)}$ as input and generates object prediction at each spatial location to ground the

target. We use the same two 1×1 convolutional layers as in One-Stage-BERT [47] for box prediction. There are $32 \times 32 = 1024$ spatial locations, and we predict 9 anchor boxes at each location. We follow the anchor selection steps in a previous one-stage grounding method [47] with the same anchor boxes used. For each one of the $1024 \times 9 = 9216$ anchor boxes, we predict the relative offset and confidence score. A cross-entropy loss between the softmax over all the 9216 boxes and the one-hot target center vector, a regression loss of the relative location and size offset, and the regularization in Equation 2 are used to train the model. We use the same classification and regression losses as in One-Stage-BERT [47].

4 Experiments

4.1 Datasets

RefCOCO/ RefCOCO+/ RefCOCOg. RefCOCO [49], RefCOCO+ [49], and RefCOCOg [29] are three visual grounding datasets with images and referred objects selected from MSCOCO [21]. The referred objects are selected from the MSCOCO object detection annotations and belong to 80 object classes. RefCOCO [49] has 19,994 images with 142,210 referring expressions for 50,000 object instances. RefCOCO+ has 19,992 images with 141,564 referring expressions for 49,856 object instances. RefCOCOg has 25,799 images with 95,010 referring expressions for 49,822 object instances. On RefCOCO and RefCOCO+, we follow the split [49] of train/ validation/ testA/ testB that has 120,624/ 10,834/ 5,657/ 5,095 expressions for RefCOCO and 120,191/ 10,758/ 5,726/ 4,889 expressions for RefCOCO+, respectively. Images in "testA" are of multiple people and the ones in "testB" contain all other objects. The queries in RefCOCO+ contains no absolute location words, such as "on the right" that describes the object's location in the image. On RefCOCOg, we experiment with the splits of RefCOCOg-google[29] and RefCOCOg-umd [30], and refer to the splits as the val-g, val-u, and test-u in Table 1. The queries in RefCOCOg are generally longer than those in RefCOCO and RefCOCO+: the average lengths are 3.61, 3.53, 8.43, respectively, for RefCOCO, RefCOCO+, RefCOCOg.

ReferItGame. The ReferItGame dataset [17] has 20,000 images from SAIAPR-12 [11]. We follow a cleaned version of split [15,36,4], which has 54,127, 5,842, and 60,103 referring expressions in train, validation, and test set, respectively.

Flickr30K Entities. Flickr30K Entities [33] has 31,783 images with 427K referred entities. We follow the same split in previous works [33,32,40]. We note that the queries in Flickr30K Entities are mostly short noun phrases and do not well reflect the difficulty of comprehensive phrase understanding. We still benchmark our method on Flickr30K Entities and compare it with other baselines for experiments completeness.

4.2 Implementation Details

Training. Following the standard setting [34,47], we keep the aspect ratio of the input image and resize the long edge to 256. We then pad the resized image to a

Table 1. The performance comparisons (Acc@0.5%) on RefCOCO, RefCOCO+, RefCOCOg (upper table), and ReferItGame, Flickr30K Entities (lower table). We highlight the best one-stage performance with **bold** and the best two-stage performance with underline. The *COCO-trained detector* generates ideal proposals only for images in COCO. This leads to the two-stage methods' good performance on RefCOCO, RefCOCO+, RefCOCOg, as well as the accuracy drop on other datasets (lower table).

Method	Feature	RefCOCO			RefCOCO+			RefCOCOg			Time
		val	testA	testB	val	testA	testB	val-g	val-u	test-u	(ms)
Two-stage Methods											
MMI [29]	VGG16-Imagenet	-	64.90	54.51	-	54.03	42.81	45.85	-	-	-
Neg Bag [30]	VGG16-Imagenet	-	58.60	56.40	-	-	-	-	-	49.50	-
CMN [14]	VGG16-COCO	-	71.03	65.77	-	54.32	47.76	57.47	-	-	-
ParallelAttn [52]	VGG16-Imagenet	-	75.31	65.52	-	61.34	50.86	58.03	-	-	-
VC [51]	VGG16-COCO	-	73.33	67.44	-	58.40	53.18	62.30	-	-	-
LGRAN [42]	VGG16-Imagenet	-	76.6	66.4	-	64.0	53.4	61.78	-	-	-
SLR [50]	Res101-COCO	69.48	73.71	64.96	55.71	60.74	48.80	-	60.21	59.63	-
MAttNet [48]	Res101-COCO	76.40	80.43	69.28	64.93	70.26	56.00	-	66.67	67.01	320
DGA [46]	Res101-COCO	-	78.42	65.53	-	69.07	51.99	-	-	63.28	341
One-stage Methods											
SSG [5]	Darknet53-COCO	-	76.51	67.50	-	62.14	49.27	47.47	58.80	-	25
One-Stage-BERT [47]	Darknet53-COCO	72.05	74.81	67.59	55.72	60.37	48.54	48.14	59.03	58.70	23
One-Stage-BERT*	Darknet53-COCO	72.54	74.35	68.50	56.81	60.23	49.60	56.12	61.33	60.36	23
Ours-Base	Darknet53-COCO	76.59	78.22	**73.25**	63.23	66.64	55.53	60.96	64.87	64.87	26
Ours-Large	Darknet53-COCO	**77.63**	**80.45**	72.30	**63.59**	**68.36**	**56.81**	**63.12**	**67.30**	**67.20**	36

Method	Feature	ReferItGame	Flickr30K Entities	Time
		test	test	(ms)
Two-stage Methods				
CMN [14]	VGG16-COCO	28.33	-	-
VC [51]	VGG16-COCO	31.13	-	-
MAttNet [48]	Res101-COCO	29.04	-	320
Similarity Net [40]	Res101-COCO	34.54	60.89	184
CITE [32]	Res101-COCO	35.07	61.33	196
One-stage Methods				
SSG [5]	Darknet53-COCO	54.24	-	25
ZSGNet [37]	Res50-FPN	58.63	63.39	-
One-Stage-BERT [47]	Darknet53-COCO	59.30	68.69	23
One-Stage-BERT*	Darknet53-COCO	60.67	68.71	23
Ours-Base	Darknet53-COCO	64.33	69.04	26
Ours-Large	Darknet53-COCO	**64.60**	**69.28**	36

size of 256×256 with the mean pixel value. We follow the data augmentation in previous one-stage studies [34,47]. The RMSProp [38] optimizer, with an initial learning rate of 10^{-4} is used to train the model with a batch size of 8. The learning rate decreases by half every 10 epochs for a total of 100 epochs. We set the weight for L_{div}, L_{cov} as 1. We select $K = 3$ as the default number of rounds and defer the related ablation studies to supplementary materials.

Evaluation. We follow the same *Acc*@0.5 evaluation protocol in prior works [33,36]. Given a language query, this metric is to consider the predicted region correct if its IoU is at least 0.5 with the ground truth bounding box.

4.3 Quantitative Results

Experiment settings. Table 1 reports visual grounding results on RefCOCO, RefCOCO+, RefCOCOg (the upper table), and ReferItGame, Flickr30K Entities (the lower table). The *top part* of each table contain results of the state-of-the-art two-stage visual grounding methods [29,30,14,50,48,27,52,51,42,46,40,32]. The "*Feature*" column lists the backbone and pretrained dataset used for proposal

feature extraction. COCO-trained Faster-RCNN [35] detector is used for region proposal generation for the experiments on RefCOCO [49], RefCOCO+ [49] and RefCOCOg [29]. We quote the two-stage methods' results on ReferItGame and Flickr30K Entities reported by SSG [5] and One-Stage-BERT [47] where Edgebox [53] is used for proposal generation.

The *bottom part* of Table 1 compares the performance of our method to other state-of-the-art one-stage methods [5,37,47]. The "*Feature*" column shows the adopted visual backbone and its pretrained dataset, if any. For a fair comparison, we modify One-Stage-BERT [47] to have the exact same training details as ours, and observe a small accuracy improvement by the modification. Specifically, we 1). encode the query as the averaged BERT word embedding instead of the BERT sentence embedding at the first token's position ([CLS]), 2). remove the feature pyramid network, and 3). follow the implementation details in Section 4.2. We refer to the modified version "*One-Stage-BERT**." Other than the state-of-the-art, we design and compare to additional alternatives to our methods such as "*single/ multi-head attention query modeling*," "*per-word sub-query*," *etc.*, in Section 4.5 and Table 3.

We obtain our main results by the method described in Section 3 and refer to it as "*Ours-Base*" in Table 1. Furthermore, we observe that a larger input image size of 512 and a ConvLSTM [44,6] grounding module increase the accuracy, but meanwhile slightly slow the inference speed. We refer to the corresponding model as "*Ours-Large*" and analyze each modification in supplementary materials.

Visual grounding results. Our proposed method outperforms the state-of-the-art one-stage grounding methods [47,5,37] by over 5% absolute accuracy on all experimented datasets.

The two-stage methods [48,27,52,51,42,46] also show good performance on COCO-series datasets (*i.e.*, RefCOCO, RefCOCO+, and RefCOCOg) by using the COCO-trained detector [35]. For example, we notice that MAttNet [48] achieves comparable performance with our best model in RefCOCO+, though our best model obviously surpasses MAttNet on RefCOCO, RefCOCOg, and the testB of RefCOCO+. However, in the ReferItGame dataset, as listed in the lower part of Table 1, MAttNet's accuracy drops dramatically. The findings of previous one-stage work [47,5] show that two-stage visual grounding methods rely highly on the region proposals quality. Since RefCOCO/ RefCOCO+/ RefCOCOg are subsets of COCO and have shared images and objects, the COCO-trained detector generates nearly perfect region proposals on COCO-series datasets. When used in other datasets, e.g., ReferItGame and Flickr30K Entities datasets, their proposal quality and grounding accuracy drop, such as the MAttNet's degraded performance in ReferItGame. Nonetheless, our method performs stably across all datasets and, meanwhile being significantly faster.

Inference time. The real-time inference speed is one major advantage of the one-stage visual grounding method. We conduct all the experiments on a single NVIDIA 1080TI GPU. We observe our method achieves a real-time inference speed of $26ms$. The method is more than 10 times faster than typical two-stage methods such as the MattNet [48] of $320ms$.

Table 2. The performance break-down with query lengths. The first row shows the experimented dataset and the number of query words in each sub-set.

RefCOCO	1-2	3	4-5	6+
Percent (%)	36.22	23.87	25.60	14.30
One-Stage-BERT	77.68	76.04	66.98	55.59
Ours-Base	79.35	79.28	72.65	66.19
Relative Gain	2.15	4.26	8.46	19.07

RefCOCO+	1-2	3	4-5	6+
Percent (%)	37.79	19.48	27.40	15.33
One-Stage-BERT	66.59	55.42	47.40	39.03
Ours-Base	71.08	60.01	56.24	49.35
Relative Gain	6.74	8.28	18.65	26.44

RefCOCOg	1-5	6-7	8-10	11+
Percent (%)	23.54	22.80	28.30	25.37
One-Stage-BERT	63.41	59.57	56.97	55.46
Ours-Base	65.49	65.37	63.97	64.86
Relative Gain	3.28	9.74	12.29	16.95

ReferItGame	1	2	3-4	5+
Percent (%)	25.78	16.76	31.53	25.93
One-Stage-BERT	82.33	66.66	56.64	34.89
Ours-Base	82.12	69.46	61.43	46.84
Relative Gain	-0.26	4.20	8.46	34.25

4.4 Performance break-down studies

We show the effectiveness of our method in modeling long queries by breaking down the test set. We split the test set of ReferItGame [17], RefCOCO [49], RefCOCO+ [49], and RefCOCOg [29] each into four sub-sets based on the query lengths (we combine the testA and testB for RefCOCO and RefCOCO+). Table 2 compares our method to One-Stage-BERT [47] on the generated sub-sets. We adopt "Ours-Base" instead of "Ours-Large" for comparison because the inference speed of "Ours-Base" is more comparable with One-Stage-BERT. The first row shows the experimented dataset and the number of query words in each sub-set. The second row shows the portion of samples in each subset. We generate sub-sets that are roughly with the same size. The middle two rows compare the accuracy of our method to the state-of-the-art one-stage grounding method [47]. The last row computes the relative gain obtained by our method as $(Ours - Base)\,/\,Base$. We observe a larger relative gain of our method on longer queries. The relative gain is around 20% on the longest query sub-set. The consistent increases in the relative gain on all experimented datasets suggest the effectiveness of our recursive sub-query construction framework in modeling and grounding long queries.

4.5 Ablation studies

In this section, we conduct ablation studies to understand our method better. We perform the study on RefCOCOg-google [29] as it has, on average, longer queries than other datasets, which can better reflect the query modeling problem.

Query modeling. Table 3 shows the ablation studies on different query modeling choices. Specifically, we systematically study the following settings.

- **Average vector.** We average the BERT embedded word features $S = \{s_n\}_{n=1}^{N}$ to form a single $512D$ vector as the query representation.
- **Per word sub-query.** We consider each word as a sub-query. The per-word sub-query modeling is used by RMI [23] for referring image segmentation.

Table 3. Ablation studies on query modeling. The sub-query modulation introduced in Section 3.2 is used for fusion.

Query Modeling	Acc@0.5
Average vector	59.24
Per word sub-query	58.36
Single-head attention	59.43
Multi-head attention	59.25
Spatial-independent sub-query	60.81
Sub-query learner (ours)	**60.96**

- **Single-head attention.** A set of self-attention scores of size N is learned from the word features $\{s_n\}_{n=1}^{N}$. We obtain a single query feature vector by weighted sum the BERT embedded word features. We use the same self-attention method as in Equation 1 to obtain the attention scores [22], expect only the text feature s_n is used as the input.
- **Multi-head attention.** Self-attention scores of size $N \times K$ are learned from the word features. One unique sub-query feature vector is formed as the input to each round.
- **Spatial-independent sub-query.** We discuss one alternative to our approach in the end of Section 3.2. We refer to it as "Spatial-independent sub-query" as shown in the last row of Table 3.
- **Sub-query learner (ours).** Instead of jointly predicting the sub-queries for all steps, our proposed sub-query learner, as introduced in Section 3.1, recursively constructs the sub-query by referring to the current text-conditional visual feature $v^{(k)}$.

Our proposed sub-query learner boosts the baseline accuracy with no attention by 1.7% (*cf.* "Average vector" and "Sub-query learner (ours)"). The query attention without the visual contents shows limited improvements over the no attention baseline (*cf.* "Average vector" and "Single/ Multi-head attention"). Instead, by referring to the text-conditional visual feature in each round, our proposed sub-query learner further improve the attention baseline by 1.5% (*cf.* "Single/ Multi-head attention" and "Sub-query learner (ours)"). This shows the importance of recursive sub-query construction. Furthermore, "spatial-independent sub-query" constructs the sub-query independently at each spatial location. This alternative leads to extra computation while is not more accurate.

Sub-query modulation. We compare with the "Concat-Conv" fusion used in One-Stage-BERT [47]. To be specific, the query feature is duplicated spatially and is concatenated with the visual and spatial features to form a $512+512+8 = 1032D$ feature. One 1×1 and one 3×3 convolution layers then generate a $512D$ fused feature. In contrast, the sub-query modulation introduced in Section 3.2 converts the sub-query feature into scaling and shifting parameters to refine the text-conditional visual feature $v^{(k)}$. Our proposed sub-query modulation improves the accuracy by 1.8% with the similar amount of fusion parameters (*cf.* "Concat-Conv": 59.20% and "Ours": 60.96%).

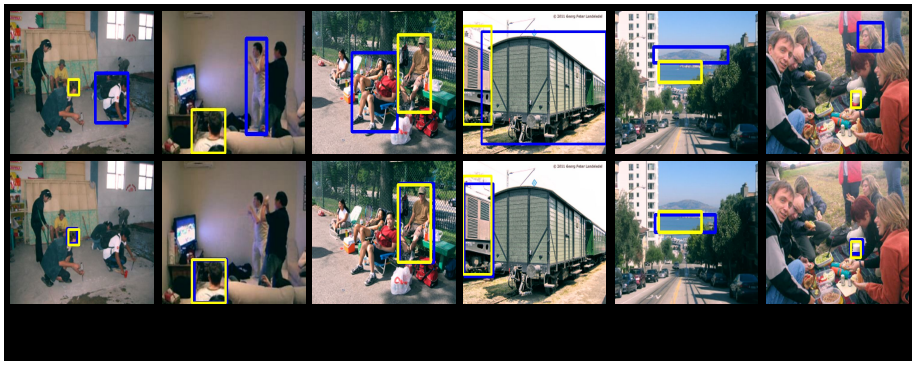

Fig. 4. Failure cases of One-Stage-BERT [47] (top row) that can be corrected by our method (bottom row). Blue/ yellow boxes are the predicted regions/ ground truths. Constructed sub-queries and per-round visualizations are in Figure 5.

4.6 Qualitative results

Figure 4 shows the failure cases made by a previous one-stage method [47] that can be correctly predicted by our method. We observe that previous methods appear to fail because of neglecting those detailed descriptions or modifiers (*e.g.*, Figures 4 (a)-(d)), or attending on the wrong keyword (*e.g.*, Figures 4 (e),(f)). In contrast, our method corrects these errors by better modeling the query.

More importantly, in order to understand what happened inside our model and explain why it works, we show the visualization of intermediate results of our model in Figure 5. The left column in Figure 5 shows the constructed sub-query in each round. The right three columns visualize the intermediate text-conditional visual feature $v^{(k)}$. For visualization, we adopt an extra output head over the feature $v^{(k)}$ in all steps and obtain the confidence score heatmaps. The confidence score indicates the probability of the object center. Therefore, the heatmap contains the peaks of object centers instead of the object contours. We highlight the referred object and the major distracting object with the green up arrow and red down arrow, respectively. We note that the intermediate prediction is just for visualization purpose, and is not in our proposed framework.

We observe that the model tends to focus on all head nouns in the first round, *e.g.*, "man", "head", "car",*etc.*, because such keywords are the most informative sub-query when given a raw image. Then, in the next few rounds, our method can refine the intermediate text-conditional visual representation and reduce the referring ambiguity. For example, in the first row of Figure 5, the model first focuses on the head-noun "head". Our model refines its prediction by the constructed sub-queries "in the middle" and "with drill" in the following rounds. Accordingly, from the heatmap visualization, we observe such a disambiguation process that the refined visual-text feature step by step generates more accurate and confident predictions. In the first round with the sub-query "persons head," the model predicts four peaks in the heatmap, each centering at an appeared

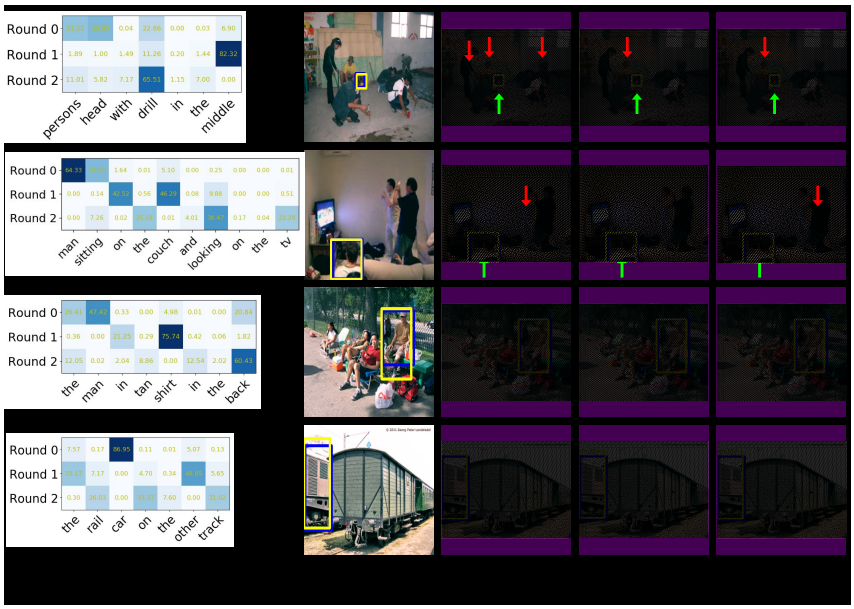

Fig. 5. Visualization of the constructed sub-queries and the intermediate text-conditional visual feature at each round. The green up arrow and the red down arrow point to the target and the major distracting object on heatmaps, respectively. Bested viewed in color. More examples and detailed analyses are in supplementary materials.

person. In the second round with sub-query "in the middle," the model focuses on the two person in the middle and eliminates two distracting objects. In the final round with the sub-query "with drill," the model successfully focuses on the referred person, and the heatmap values for all other distracting objects are greatly suppressed. We observe similar recursive disambiguation processes in other examples in Figure 5 and supplementary materials.

5 Conclusions

We have proposed a recursive sub-query construction framework to address the limitation of previous one-stage methods when understanding complex queries. We recursively construct sub-queries to refine the visual-text feature for grounding box prediction. Extensive experiments and ablation studies have validated the high effectiveness of our method. Our proposed framework significantly outperforms the state-of-the-art one-stage methods by over 5% in absolute accuracy on multiple datasets while still maintaining a real-time inference speed.

Acknowledgment

This work is supported in part by NSF awards IIS-1704337, IIS-1722847, and IIS-1813709, Twitch Fellowship, as well as our corporate sponsors.

References

1. Bajaj, M., Wang, L., Sigal, L.: G3raphground: Graph-based language grounding. In: ICCV (2019)
2. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: EMNLP. pp. 740–750 (2014)
3. Chen, K., Kovvuri, R., Gao, J., Nevatia, R.: Msrc: Multimodal spatial regression with semantic context for phrase grounding. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. pp. 23–31. ACM (2017)
4. Chen, K., Kovvuri, R., Nevatia, R.: Query-guided regression network with context policy for phrase grounding. In: ICCV (2017)
5. Chen, X., Ma, L., Chen, J., Jie, Z., Liu, W., Luo, J.: Real-time referring expression comprehension by single-stage grounding network. arXiv preprint arXiv:1812.03426 (2018)
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016)
7. De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.C.: Modulating early visual processing by language. In: Advances in Neural Information Processing Systems. pp. 6594–6604 (2017)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Dogan, P., Sigal, L., Gross, M.: Neural sequential phrase grounding (seqground). In: CVPR (2019)
10. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: ICLR (2017)
11. Escalante, H.J., Hernández, C.A., Gonzalez, J.A., López-López, A., Montes, M., Morales, E.F., Sucar, L.E., Villaseñor, L., Grubinger, M.: The segmented and annotated iapr tc-12 benchmark. CVIU (2010)
12. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: A search space odyssey. IEEE transactions on neural networks and learning systems (2016)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
14. Hu, R., Rohrbach, M., Andreas, J., Darrell, T., Saenko, K.: Modeling relationships in referential expressions with compositional modular networks. In: CVPR (2017)
15. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: CVPR (2016)
16. Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. In: ICLR (2018)
17. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: EMNLP (2014)
18. Li, J., Wei, Y., Liang, X., Zhao, F., Li, J., Xu, T., Feng, J.: Deep attribute-preserving metric learning for natural language object retrieval. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 181–189. ACM (2017)
19. Li, S., Xiao, T., Li, H., Zhou, B., Yue, D., Wang, X.: Person search with natural language description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1970–1979 (2017)
20. Liao, Y., Liu, S., Li, G., Wang, F., Chen, Y., Qian, C., Li, B.: A real-time cross-modality correlation filtering method for referring expression comprehension. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10880–10889 (2020)

21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
22. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: ICLR (2017)
23. Liu, C., Lin, Z., Shen, X., Yang, J., Lu, X., Yuille, A.: Recurrent multimodal interaction for referring image segmentation. In: ICCV (2017)
24. Liu, D., Zhang, H., Wu, F., Zha, Z.J.: Learning to assemble neural module tree networks for visual grounding. In: ICCV (2019)
25. Liu, J., Wang, L., Yang, M.H.: Referring expression generation and comprehension via attributes. In: ICCV (2017)
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
27. Liu, X., Wang, Z., Shao, J., Wang, X., Li, H.: Improving referring expression grounding with cross-modal attention-guided erasing. In: CVPR (2019)
28. Luo, G., Zhou, Y., Sun, X., Cao, L., Wu, C., Deng, C., Ji, R.: Multi-task collaborative network for joint referring expression comprehension and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10034–10043 (2020)
29. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR (2016)
30. Nagaraja, V.K., Morariu, V.I., Davis, L.S.: Modeling context between objects for referring expression understanding. In: ECCV (2016)
31. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: AAAI (2018)
32. Plummer, B.A., Kordas, P., Kiapour, M.H., Zheng, S., Piramuthu, R., Lazebnik, S.: Conditional image-text embedding networks. In: ECCV (2018)
33. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. International journal of computer vision (2017)
34. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
36. Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: ECCV (2016)
37. Sadhu, A., Chen, K., Nevatia, R.: Zero-shot grounding of objects from natural language queries. In: ICCV (2019)
38. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning (2012)
39. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision (2013)
40. Wang, L., Li, Y., Huang, J., Lazebnik, S.: Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
41. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: CVPR (2016)
42. Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., Hengel, A.v.d.: Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In: CVPR (2019)

43. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
44. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015)
45. Yang, S., Li, G., Yu, Y.: Cross-modal relationship inference for grounding referring expressions. In: CVPR (2019)
46. Yang, S., Li, G., Yu, Y.: Dynamic graph attention for referring expression comprehension. In: ICCV (2019)
47. Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., Luo, J.: A fast and accurate one-stage approach to visual grounding. In: ICCV (2019)
48. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: CVPR (2018)
49. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: ECCV (2016)
50. Yu, L., Tan, H., Bansal, M., Berg, T.L.: A joint speaker-listener-reinforcer model for referring expressions. In: CVPR (2017)
51. Zhang, H., Niu, Y., Chang, S.F.: Grounding referring expressions in images by variational context. In: CVPR (2018)
52. Zhuang, B., Wu, Q., Shen, C., Reid, I., van den Hengel, A.: Parallel attention: A unified framework for visual object discovery through dialogs and queries. In: CVPR (2018)
53. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV (2014)