Look Before You Leap: Learning Landmark Features for One-Stage Visual Grounding

Binbin Huang [1] Dongze Lian [1] Weixin Luo [1] Shenghua Gao[†1,2]

[1]ShanghaiTech University
[2]Shanghai Engineering Research Center of Intelligent Vision and Imaging
{huangbb, liandz, luowx, gaoshh}@shanghaitech.edu.cn

Abstract

An LBYL ('Look Before You Leap') Network is proposed for end-to-end trainable one-stage visual grounding. The idea behind LBYL-Net is intuitive and straightforward: we follow a language's description to localize the target object based on its relative spatial relation to 'Landmarks', which is characterized by some spatial positional words and some descriptive words about the object. The core of our LBYL-Net is a landmark feature convolution module that transmits the visual features with the guidance of linguistic description along with different directions. Consequently, such a module encodes the relative spatial positional relations between the current object and its context. Then we combine the contextual information from the landmark feature convolution module with the target's visual features for grounding. To make this landmark feature convolution light-weight, we introduce a dynamic programming algorithm (termed dynamic max pooling) with low complexity to extract the landmark feature. Thanks to the landmark feature convolution module, we mimic the human behavior of 'Look Before You Leap' to design an LBYL-Net, which takes full consideration of contextual information. Extensive experiments show our method's effectiveness in four grounding datasets. Specifically, our LBYL-Net outperforms all state-of-the-art two-stage and one-stage methods on ReferitGame. On RefCOCO and RefCOCO+, Our LBYL-Net also achieves comparable results or even better results than existing one-stage methods. Code is available at https://github.com/svip-lab/LBYLNet.

1. Introduction

Humans often refer to objects in an image by describing their relationships with other entities, *e.g.* "laptop on table", and understanding their relationships is vital to comprehend referential expressions. *Visual grounding*, aiming to localize the entities described by referential expressions, inherently requires contextual information for grounding the target. By considering relationship of objects, several recent studies have achieved promising results [44, 24, 9, 43]. In particular, these methods usually leverage a two-stage paradigm, where they first extract region proposals as candidates and then rank the region-expression pairs as a way of *metric learning*.

Query expression: the guy in brown on the right.

Figure 1. Illustration on how LBYL-Net uses contextual cues. On the left figure, the target location (green) perceives information from landmarks (red) to localize itself. In this case, landmarks attend to the attribute *brown* to differ from the other *guy*. The right figure shows our predicted result (blue box) and the ground-truth (yellow box).

Although effective, these two-stage methods have the following defects: (i) two stages bring time complexity, which hinders these methods from being real-time. (ii) since only objects in the pre-defined categories are considered, the contextual cues in the whole scene may not be fully exploited. Motivated by the success of one-stage detection [28, 19], one-stage based visual grounding has gained great interest, where the pipeline is simplified, and the inference is accelerated with a *detecting and matching simultaneously* paradigm [39, 29]. These detection-based one-stage approaches, however, still perform localization on grid features *indivisually*. The contextual information in the whole scene, especially relationships between objects,

† Corresponding author

is not thoroughly investigated yet, making them inferior to their two-stage counterparts.

From this perspective, it is desirable to enable relationship modeling in one-stage visual grounding since the object requires perceiving the relational entities mentioned by the language to localize itself, *e.g.* "the chair with owl on it". We enable the grid features to capture rich contextual cues for better localization by introducing the concepts of *Landmark Features* and *Landmark Feature Convolution*.

To begin with, in our real life, we usually judge our location or positions of other buildings by using an easily noticed building, which is called *Landmark*. Similarly, in the image domain of visual grounding, the landmarks can be regarded as those locations that are helpful for object localization. Figure 1 shows the visualization of landmarks in an image given the query language. These landmarks might fall on the background, other objects or the object itself to be located as long as they have helpful semantic cues. The network could extract the *Landmark Features*, which contains the global contextual information from these landmarks. To fully integrate this contextual information to improve the localization, these landmark features are propagated to the target object from different orientations to characterize relative positions by an efficient dynamic programming algorithm, termed *Dynamic Max Pooling*. By aggregating landmark features with a standard convolution operation, the grid features are equipped with (i) global receptive field (ii) direction-awareness. We call the whole process as *Landmark Feature Convolution*.

Considering the long-range context, we propose a novel one-stage visual grounding framework. Our network first applies feature pyramid network (FPN) [15] to extract visual features of objects from different scales, of which effectiveness has been proven for better object localization. A landmark feature convolution is then employed to extract contextual information of objects from different orientations, for a better characterizing relationship to objects mentioned by expressions. Since we mimic the 'Look Before You Leap' behavior of us humans in visual grounding, we term our method as LBYL-Net.

We summarize our main contributions as follows:

- We propose a novel LBYL-Net for one-stage visual grounding, which combines the visual feature of objects mentioned in the description as well as landmark features of the spatial relationships between different objects for target localization;
- A landmark features convolution is proposed, which has a global receptive field but without introducing extra parameter and complexity. We showcase it's superiority over related convolutional modules, *i.e.* dilated convolution [41], deformable convolution [4] and Non-Local module [34].
- Extensive experiments show the effectiveness and efficiency of our LBYL-Net on four grounding datasets. Especially, our method achieves state-of-the-art performance on ReferitGame.

2. Related Work

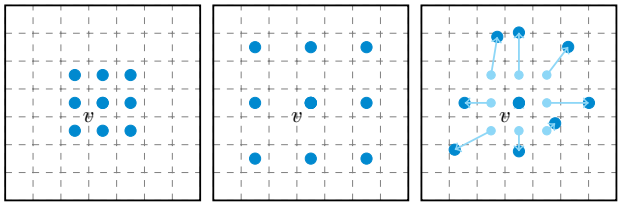
Two-Stage Visual Grounding. Probably motivated by the evidence that regions of interest can provide better localization of individual entities and the ease to build their relational connections, the two-stage have become the de-facto approaches over a period of time. Typically, different approaches differ in how they represent the context. Mao *et al.* [23] and Hu *et al.* [10] use the whole image as a global context, while Yu *et al.* [44] directly pool visual feature from nearby objects as a way of modeling visual differences, showing that focus on the relationship between objects can achieve better results. Furthermore, the context in [24, 9] is regarded as weak supervision signals of unannotated objects, and multiple instances learning [7] is then adopted to maximize the joint likelihood of all object pairs. However, the above modeling may oversimplify the number of contextual objects to a fixed size, *e.g.*, one object as the contextual information. To this end, Zhang *et al.* [47] generates an attention map over all the objects as contextual information to approximate the combinatorial context configurations using a Variational Bayesian framework. For more detailed visual-language alignment, attention mechanisms are also widely adopted to fragment language to match the targets or contextual objects [5, 43, 48]. Unlike them, we think context can present arbitrarily within the whole scene and fully integrate them into a one-stage framework.

One-Stage Visual Grounding. Before using one-stage object detectors, several methods have attempted to directly regress the bounding box from the whole image. However, these frameworks often suffer from a lower recall of objects, making them inferior to the two-stage counterparts. Some attention-based techniques are employed to enhance the local features of the target [8]. Besides, Yeh *et al.* [40] use a subwindow search to find the location that minimizes the energy function. Encouraged by the prominent one-stage detectors (*e.g.*, YOLO [28], SSD [19]), many recent one-stage approaches have regarded proposals as grids in the feature map and directly regress the bounding box from the grid features responsible for detection [39, 29]. While achieving a large-margin improvement compared with that try to directly regress object from the entire image, such progress may attribute to the robust local features representation on the grid. Another line to improve one-stage visual grounding is to apply complex language modeling, such as decomposing the longer phrase into multiple parts [38]. In this work, we do not use complex techniques for language modeling. We show that by simply considering the context within the scene, our network can show competitive results.

3. Landmark Feature Convolution

We first summarize the most common convolutions, which we categorize into the family of *point-based sampling* strategy, and along the way, discuss their relations, advantages and limitations. After that, we introduce our proposed *region-based sampling* strategy, followed by the landmark feature convolution as well as its formulation and implementation.

3.1. Point-based Sampling



(a) Standard Conv. (b) Dilated Conv. (c) Defrormable Conv.

Figure 2. A graphical view of different *point-based* convolutions.

Given an input feature map $X = \{x_v : v \in V\}$ with node feature $x_v \in \mathbb{R}^c$, a *point-based* convolution learns a representation vector y_v by

$$y_v = \text{SUM}(\{W_{(u,v)} \cdot x_u : \forall u \in \mathcal{N}(v) \cup \{v\}\}), \quad (1)$$

where $v \in V$ is the location of the node, $\mathcal{N}(v)$ is the neighborhood of node v and $W_{(u,v)}$ parameterizes the spatial relation of node u and node v. In the context of image feature maps, a node is the same as a location, so that we use both notations interchangeably. Different convolutions have different sampling strategies $\mathcal{N}(v)$. That is, how we sample nodes for convolution to represent the output vector y_v.

Standard Convolution. In a 3×3 convolution kernel, a regurlar grid window $\mathcal{R}$ is used, which can be represented as a list of offests. Then the sampled neighbor $\mathcal{N}(v)$, or we called receptive field, is equal to $\{v + o : \forall o \in \mathcal{R}\}$, as shown in Figure 2 (a). Notably, the parameter W is not shared among sampled locations, such that the spatial relations between node v and its neighborhood nodes $u \in \mathcal{N}(v)$ can be explicitly captured. This property enables the convolution to detect meaningful patterns, such as line segments and corners. Theoretically, the receptive field grows as a convolutional layers stack, allowing deep CNNs to perform various high-level semantic tasks, such as object recognition, face detection, and semantic segmentation. However, the *effecive* receptive field often occupies a fraction of the full *theoratical* receptive field and converges to a Gaussian, making recognizing large objects and long-range modeling still challenge [22].

Dilated Convolution. To model long-range context, one solution is to increase the number of sampling points to enlarge receptive field, such as morphing the kernel size from 3×3 to 5×5. However, this substantially improves the number of parameters and brings the risk of over-fitting. To this end, the sampling window of a 3×3 kernel is dilated to a 5×5 grid window, resulting in a dilated convolution (in this case, dilation is 2) [41], as shown in Figure 2 (b).

By enlarging the receptive field without introducing extra parameters, dilated convolution has become the de-facto technique to aggregate multi-scale context and hence has advanced a variety of researches [2, 3]. However, due to the sparse topology of sampling locations, dilated convolution can suffer from *gridding* artifacts [42, 35]. In visual grounding, this can hinder relation modeling of objects since their spatial positions can be arbitrary.

Deformable Convolution. Due to the fixed topology of sampling locations, the aforementioned CNNs are inherently limited to model large, unkown transformations [4]. Deformable convolution relieves this issue by adding learnable 2D offsets to the regular grid via additional convolutional layers [4]. That is, transforming $\mathcal{N}(v) = \{v + o : \forall o \in \mathcal{R}\}$ to $\mathcal{N}(v) = \{v + o + \Delta o : \forall o \in \mathcal{R}\}$, where Δo is a leraned offset. After that, the node features are sampled from the transformed $\mathcal{N}(v)$ via bilinear interpolation. The illustrasion is shown in Figure 2 (c).

While deformable convolution excels in recognize objects by morphing the kernels to *intra-object* geometries, little do we know that such deformation can generalize to model *inter-object* relations, particularly in the context of visual grounding. One potential is that it may fail when modeling relations across very long distances since the learned offsets are expected to be constrained by the receptive field of their producers, *i.e.* the standard CNNs.

Graph Convolution. By regarding any pair of points have an edge, one can apply graph convolution on node v to have a global receptive field. For example, Non-Local module [34] update y_v by

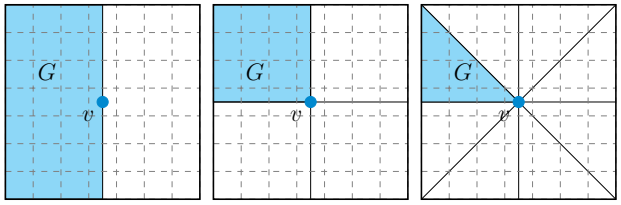
$$y_v = \text{SUM}\{f(x_v, x_u) \cdot W \cdot x_u : \forall u \in V\}, \quad (2)$$

where $f(x_v, x_u)$ is a affinity between x_v, x_u, and W is shared for all locations. Since W is shared, the ability to represent spatial relations relies on $f(x_v, x_u)$, which requires V has a suitable positional embedding. While applicable, how to effectively represent relative positional embedding still remains unclear.

3.2. Region-based Sampling

To overcome the gridding artifacts and receptive field limitations of *point-based* convolutions, we proposed a *region-based* sampling strategy for convolution. That is, we set some axes on the node v to part the whole feature map into several sub-regions and update the representation of v by aggregating representations from each region. Formally, we update the representation of node v by

$$y_v = \text{SUM}(\{W_{(v,G)} \cdot h_G : \forall G \in \mathcal{P}_v(V)\}), \quad (3)$$



(a) $\mathcal{P} = 2$. (b) $\mathcal{P} = 4$. (c) $\mathcal{P} = 8$.

Figure 3. Some variants of our *region-based* convolution. $\mathcal{P} = k$ denotes that we divide the whole feature map V into k groups, *i.e.* $\|\mathcal{P}(V)\| = k$. For clarity, only one group (G) is highlighted.

where $\mathcal{P}_v$ denotes the partition function $\mathcal{P}$ over the input feature map V based on node v, and G is a group of nodes that shares similar spatial relation to node v, and h_G is yielded from G, which we call *landmark feature*. There are a variety of partitions $\mathcal{P}$, as shown in Figure 3. For $\mathcal{P} = 2$ in Figure 3 (a), the nodes are parted into two groups according to the vertical axis, such that one group is to the left of node v and the other is to the right. By parameterizing two groups differently, the convolution can specialize in detecting horizontal spatial relations and hence help to ground the target. Namely, given the expression "man to the right of car", the likelihood of the nodes to the right of "car" will be raised.

To extract *landmark feature* h_G, we apply a simple permutation invariant function, sharing the same spirit to those obtaining the entire graph's representation in graph classification [13, 36]. We use Max Pooling as the readout function, following the concept of landmark (namely, the most noticeable position). To make the *landmark feature* more descrimitive and spatial-aware, we can also use MLP or CNN to embed h_G. We find an additional one-layer MLP is sufficient. The h_G is yielded as follow:

$$h_G = \text{MAX}(\{\text{ReLU}(W_G \cdot x_u) : \forall u \in G\}), \qquad (4)$$

where W_G is the embedding parameter that is *not* shared among different groups. Since W_G is exclusive to each pariticular group, we do not need position embedding once choosing a suitable partition $\mathcal{P}$. In this paper, we empirically adopt $\mathcal{P} = 4$ for modeling the most common relations (*i.e.* "left, right, on, below"), as shown in Figure 3 (b). Since our module update the representation of node v with landmark features $\{h_G, G \in \mathcal{P}(V)\}$, we call it *Landmark Feature Convolution*.

Implementation details. We pay close attention to the efficiency of our proposed module. The biggest bottleneck is that we need to perform k times Max Pooling to update $\{y_v : v \in V\}$, for $\mathcal{P} = k$. Noticing that *landmark features* of adjacent nodes have overlapping sub-regions, computations can be reduced by dynamic programming. Assuming the input is the embedded feature map $\mathcal{X}_G$, we show how we compute $\mathcal{H}_G = \{h_v : \forall v \in V\}$ for the group highlighted in Figure 3 (b) with a few lines in Algorithm 1, termed *Dynamic Max Pooling*. Computing for other groups or partitions $\mathcal{P}$ can be implemented as straightforward. We also accelerate it with CUDA since each channel can run in parallel, which distinguishes our algorithm from those running RNNs over the feature map [1, 18].

Algorithm 1: Dynamic Max Pooling

Input: An input $\mathcal{X} = \{x_{i,j}\}^{M \times N}$ where $x_{i,j} \in \mathbb{R}^c$.
Output: An output $\mathcal{H} = \{h_{i,j}\}^{M \times N}$, where $h_{i,j} \in \mathbb{R}^c$.

```
H ← X
for i ∈ [1, M] do
    for j ∈ [1, N] do
        h_{i,j} ← MAX({h_{i,j-1}, h_{i-1,j}})
    end for
end for
return H
```

Overall, our algorithm has linear time-space complexity with respect to the number of nodes, *i.e.* $\Theta(k\|V\|)$ where k represents the number of partitions. Although having sequential operations, our implementation demonstrates its superiority over graph convolution layers, such as Non-Local layer [34] or self-attention layer [31] whose time-space complexity is $\Theta(\|V\|^2)$, by simulations, as shown in Figure 4.

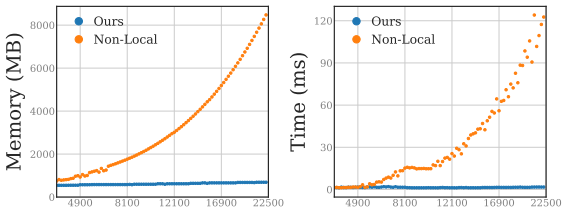


Figure 4. Real time simulation of memory usage and running time. Different from Non-Local layer [34], our LFC is linear time-space complexity *w.r.t.* $\|V\|$, and still enjoy a global receptive field.

4. LBYL-Net

Based on landmark feature convolution, we propose LBYL-Net. LBYL-Net consists of four components: a visual and language encoder, a fusion module, a landmark feature convolution module, and a localization module, which are introduced in the following, respectively.

Visual and language encoder. In Figure 5, LBYL-Net firstly forwards the given image through a backbone network, where we use DarkNet-53 based Feature Pyramid Network (FPN) [15] to extract features from different scales. We choose the outputs from P3 to P5 levels of FPN as visual features $v \in \mathbb{R}^{c_d \times h_d \times w_d}$, where $d = 3, 4, 5$ shows the d-th level. After that, we utilize a 1×1 convo-

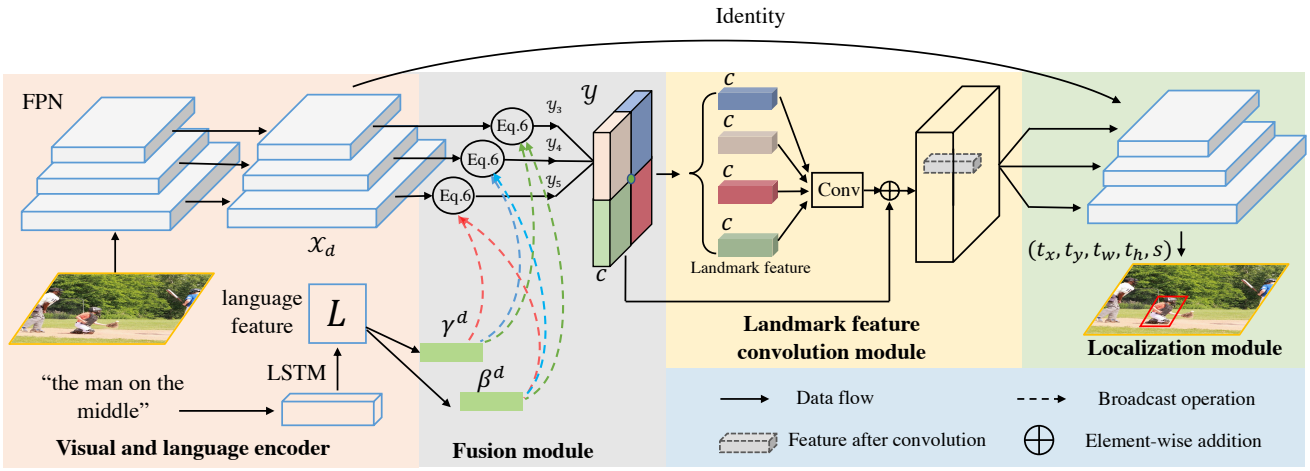


Figure 5. Our proposed LBYL-Net, which consists of four components: a visual and language encoder, a fusion module, a landmark feature convolution module and a localization module.

lution in v to obtain feature maps with the same channel c_v and concatenate the coordinate features with a 8 dimension position embedding vector, which is the same with prior work [39, 38], such that we generate the fused feature maps $\mathcal{X}_d \in \mathbb{R}^{(c)\times h_d \times w_d}$, where $c = c_v + 8$.

For the language encoder, we firstly encode each word to dimension c_l with a one-hot embedding given a language expression, and then a Bi-LSTM is applied to extract language feature $L \in \mathbb{R}^{c_l}$ to encode the whole expression. We also use BERT [6] in place of LSTM to enhance language representation, following [39, 38].

Fusion Module. Given the generated language feature, we aim to obtain the maximum response of visual information conditioned on language. Therefore, we fuse visual and language features through a FiLM module [26] and a 1×1 convolution. FiLM applies an affine transformation in visual features $\mathcal{X}_d$ under the guidance of language L. The specific operations are as follows:

$$\gamma^d = \text{MLP}^d_\gamma(L), \beta^d = \text{MLP}^d_\beta(L), \tag{5}$$

and

$$\mathcal{Y}_d = \text{ReLU}(\text{Conv}(\text{ReLU}(\gamma^d \odot \mathcal{X}_d \oplus \beta^d))), \tag{6}$$

where MLP^d_γ and MLP^d_β are two one-layer MLP that maps language vector L to coefficients γ^d and β^d. Then we apply these coefficients to visual feature $\mathcal{X}_d$ from different FPN level followed by convolution and ReLU operations, yielding the output $\mathcal{Y}_d \in \mathbb{R}^{(c)\times h_d \times w_d}$, where $\odot$ and $\oplus$ represent the broadcast element-wise multiplication and addition, respectively. After that, the feature of each position in $\mathcal{Y}_d$ might be adaptively responsible for different fine-grained properties, such as colors, positions, categories conditioned on language [26].

Prior to spatial relation modeling, we observe that FPN can hurt the performance. It could be that FPN distributes objects into different feature maps based on their scales, making modeling across-scale relation difficult. For example, given the relation "painting over bed", the "painting" hardly stands a chance of perceiving the "bed" if they are assigned to two separate feature maps. Simply summing the feature map to the intermediate size cures that problem, the same technique as BFPN [25]. In particular, we achieve this through max downsampling and bilinear upsampling, and finally:

$$\mathcal{Y} = \frac{1}{3}\sum_{d=3}^{d=5} \mathcal{Y}_d. \tag{7}$$

Landmark feature convolution module and localization module. To consider the landmark features, we apply a landmark feature convolution in feature map $\mathcal{Y}$, where we choose $\mathcal{P} = 4$ (as shown in Figure 3 (b)). By DMP (*dynamic max pooling*) and convolution, landmark features from four sub-regions are aggregated, yielding a direction-aware feature map. Afterward, we distribute the features to different FPN levels to account for the scale problem in general object detection.

Finally, we feed them into the localization module, where we adopt an anchor-based box regression head in YOLOv3 as a detection head. The final output of LBYL-Net has a dimension of $KA \times h_d \times w_d$, where $A = 3$ is the number of anchors and $K = 5$ for (t_x, t_y, t_w, t_h, s), where the first four values mean the bounding box offset relative to the pre-defined anchor and the last one is the confidence score indicating whether there is an object in this position. Following [28], only the anchor with the largest IoU with the ground-truth bounding box is assigned as a positive sample; the rest are negative samples. Therefore, there

Methods	Visual Encoder	Language Encoder	RefCOCO			RefCOCO+			RefCOCOg	Time (ms)
			val	testA	testB	val	testA	testB	val	
Two-stage methods										
MMI [23]	VGG-16	-	-	64.9	54.51	-	54.03	42.8	-	-
Neg Bag [24]	VGG-16	-	-	58.6	56.4	-	-	-	49.5	-
CMN [9]	VGG-16	LSTM	-	71.03	65.77	-	54.32	47.76	-	-
VC [47]	VGG-16	LSTM	-	73.33	67.44	-	50.86	58.03	-	
ParallelAttn [48]	VGG-16	LSTM	-	75.31	65.52	-	61.34	50.86	-	-
LGRAN [33]	VGG-16	LSTM	-	76.6	66.4	-	64.00	53.40	61.78	-
SLR [45]	ResNet-101	LSTM	69.48	73.71	64.96	55.71	60.74	48.80	-	-
MAttNet [43]	ResNet-101	LSTM	76.40	80.43	69.28	64.93	70.26	56.00	-	320
DGA [37]	ResNet-101	LSTM	-	78.42	65.53	-	69.07	51.99	-	341
CM-Att-Erase [20]	ResNet-101	LSTM	78.35	83.14	71.32	68.09	73.65	58.03	68.67	-
NMTree [17]	ResNet-101	TreeLSTM [30]	76.41	81.21	70.09	66.46	72.02	57.52	64.62	-
One-stage methods										
RCCF [14]	DLA-34	LSTM	-	81.06	71.85	-	70.35	56.32	**65.73**	25
YOLO-VG† [39]	DarkNet-53	BERT	72.05	74.35	68.5	56.81	60.23	49.6	56.12	23
SQC-Base [38]	DarkNet-53	BERT	76.59	78.22	73.25	63.23	66.64	55.53	60.96	26
SQC-Large [38]	DarkNet-53	BERT	77.63	80.45	72.3	63.59	68.36	56.81	63.12	36
Baseline [39]	DarkNet-53	LSTM	72.36	73.86	65.93	57.98	63.97	48.31	47.25	24
LBYL-Net w/o LFC	DarkNet-53	LSTM	77.43	80.75	70.68	64.84	70.24	54.71	56.17	25
LBYL-Net	DarkNet-53	LSTM	78.76	82.18	71.91	66.67	73.21	56.23	58.72	28
LBYL-Net	DarkNet-53	BERT	**79.67**	**82.91**	**74.15**	**68.64**	**73.38**	**59.49**	62.70	30

† indicates the result is adopted from [38].

Table 1. Performance comparisons on the RefCOCO, RefCOCO+, RefCOCOg. The best two-stage performance is highlighted with underline, and the best one-stage performance is highlighted with **bold**.

Methods	Visual Encoder	Language Encoder	Pr@0.5 (%)	Time (ms)
Two-stage methods				
CMN[9]	VGG-16	LSTM	28.33	-
VC [47]	VGG-16	LSTM	31.13	-
Similarity Net [32]	ResNet-101	-	34.54	184
CITE [27]	ResNet-101	-	35.07	196
MAttNet [43]	ResNet-101	LSTM	29.04	320
DDPN‡ [46]	ResNet-101	LSTM	63.00	-
One-stage methods				
ZSGNet [29]	ResNet-50	LSTM	58.63	25
RCCF [14]	DLA-34	LSTM	63.79	25
YOLO-VG [39]	DarkNet-53	LSTM	58.76	21
YOLO-VG [39]	DarkNet-53	BERT	59.30	38
SQC-Base [38]	DarkNet-53	BERT	64.33	26
SQC-Large [38]	DarkNet-53	BERT	64.60	36
Baseline [39]	DarkNet-53	LSTM	59.28	24
LBYL-Net w/o LFC	DarkNet-53	LSTM	62.59	25
LBYL-Net	DarkNet-53	LSTM	65.48	28
LBYL-Net	DarkNet-53	BERT	**67.47**	30

Table 2. Performance comparisons on the ReferitGame [11].

is only one positive sample because we only want to find an object referred to by sentence. For the ranking loss, we maximize the distance between the positive sample and the negative samples. Thus a cross-entropy loss is employed, which can be viewed as MMI training defined in [23]. For the bounding box regression loss, we use an MSE loss to minimize the distance between the predicted bounding box and the ground-truth. The whole loss function consists of a localization term and a regression term:

$$\ell = \ell_{loc} + \beta \ell_{reg}, \tag{8}$$

where β is the hyper-parameter to balance two terms, and we empirically set $\beta = 5$. The whole network is optimized with Adam [12] in an end-to-end manner.

5. Experiments

5.1. Implementation and Evaluation

Training. A DarkNet-53 pre-trained on COCO is used as our backbone, and a cosine annealing strategy [21] is employed for optimization. We train our network with a learning rate $1e^{-4}$, weight decay $1e^{-4}$, batch size 64, with GPUs. We do not use very high resolution for speed, although it could be helpful for the performance. The input images are resized 256×256 and employed on two GTX TITAN X. The total numbers of epochs are 100 for ReferitGame, RefCOCO, RefCOCO+ datasets, and 30 for the RefCOCOg dataset.

The standard data augmentation methods in object detection are employed. We use random horizontal flip, random affine operations, and random color jitter. When horizontally flipping images, we need to flip the expressions simultaneously. *e.g.*, replacing 'left' with 'right' and vice versa.

Evaluation. We evaluate our method on ReferitGame [11], RefCOCO [44], RefCOCO+ [44] and RefCOCOg [23] visual grounding datasets. The evaluation metric is the same as that in [16]. Specifically, given a regressed bounding box of the referring object, we treat the regression as right if IoU > 0.5 between the ground-truth bounding box and prediction, termed as Pr@0.5. We also use Pr@0.75 for ana-

Module	Pr@0.5(%)	Pr@0.75(%)
Baseline	59.28	40.02
+ *FiLM*	60.99 (+1.71)	40.24 (+0.22)
+ *FiLM* + *BFPN*	62.59 (+1.71+1.60)	41.00 (+0.22+0.76)
+ *FiLM* + *BFPN* + *LFC*	65.48 (+1.71+1.60+2.87)	44.31 (+0.22+0.76+3.31)

Table 3. Ablation studies on ReferitGame. The numbers inside parentheses show the improvement upon the baseline.

Ablation	Pr@0.5(%)	Pr@0.75(%)	Time (ms)
1 × 1 Conv	62.59	41.00	25
Nonlocal NN	63.59	42.46	29
Dilated Conv	63.85	42.90	26
Deform Conv	63.99	43.72	29
LFC	**65.48**	**44.31**	28

Table 4. Performance comparison to related convolution operations on ReferitGame.

lyzing certain experiments.

Settings. We re-implement YOLO-VG [39] with an LSTM language encoder as our baseline, which perform grounding on gird features individually, *i.e.* only using 1×1 convolution in fusion module. We have small modifications to keep the same training scheme like ours, like learning rate and optimizer. We see a slight improvement in accuracy in ReferitGame compared to the result reported in [39]. This will serve as our baseline for all of our experiments. We mainly report results with LSTM, unless specified.

5.2. Quantitative Results

Comparisons with baselines. In summary, our LBYL-Net has about 6.2%, 7.5%, 8.6%, 12.4% absolute improvement on ReferitGame, RefCOCO, RefCOCO+, RefCOCOg, respectively, which demonstrates the effectiveness of our LBYL-Net. When a stronger language encoder is adopted, the performance can be further improved. The advance of our modification will be detailed in ablation studies.

Comparisons with state-of-the-art results. We compare our proposed LBYL-Net with state-of-the-art results of both one-stage and two-stage methods on ReferitGame, RefCOCO, RefCOCO+, RefCOCOg. The comparisons on ReferitGame are listed in Table 2 and those on RefCOCO, RefCOCO+, RefCOCOg are listed in Table 1. Stronger visual and language representation can boost performance. For fair comparisons, we list the visual encoders and language encoders of these methods.

In ReferitGame, it is worth noting that the two-stage methods usually obtain poor results because they have no qualified proposals. We attribute the poor performance to the off-the-shelf detector that is not trained on ReferitGame. The evidence is that by using an end-to-end trainable RPN (region proposal network), the best result in two-stage methods can be achieved [46]. In COCO series datasets, *e.g.*, RefCOCO, RefCOCO+, RefCOCOg, top results are usually achieved by two-stage methods since they adopt powerful detectors for COCO datasets. The detector helps to filter out irrelevant or noise regions prior to performing reasoning. However, our one-stage LBYL-Net still achieves competitive results among all the SOTA methods on RefCOCO, RefCOCO+, and the best result on ReferitGame. We show that not only is a one-stage pipeline advantageous to efficiency but can also achieve very strong performance by modeling long-range spatial relations.

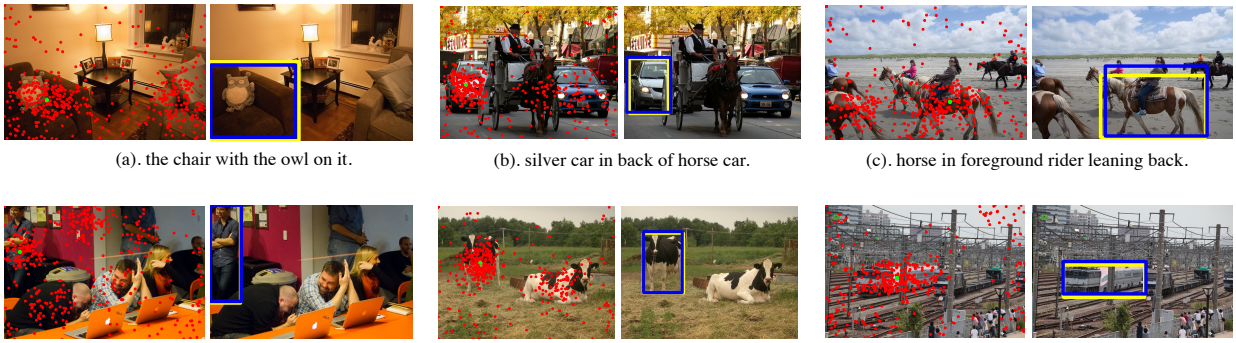
Another line to improve one-stage visual grounding is to better comprehend longer expression, especially for RefCOCOg, which contains more complex sentences. Although decomposing the expressions can achieve significant improvement [14, 38], we adopt a gloabl language representation for the sake of simplicity. On RefCOCOg, our model still improves the performance upon our baseline [39] by 12% and 6%, with LSTM and BERT, respectively, showing that modeling long-range spatial relations can help to comprehend longer sentences since these cases require more spatial relational cues to localize the target.

5.3. Ablation Studies

We conduct several ablation studies on ReferitGame [11] to reveal the effectiveness of our proposed LBYL-Net as well as our proposed *Landmark Feature Convolution Module* (LFC). We additionally train three models upon this baseline by gradually replacing the 'Concat-Conv' with FiLM [26], replacing the FPN with BFPN [25] and adding LFC, respectively. The results are shown in Table 3. Thanks to the capacity of FiLM of fusing language and visual features, the performance is improved by 1.71% under the metric Pr@0.5. The performance is further boosted by aligning visual features from different scales with BFPN by 1.6%. However, the major improvement should be attributed to landmark feature convolution because it significantly raises the precision to 65.48%. This can be more clearly validated under Pr@0.75. In this metric, LFC significantly improves the accuracy by more than 3%, while the improvement of FiLM plus BFPN is marginally close to 1%.

5.4. Effectiveness of LFC

We first compare the performance of the LFC and the point-based convolutions discussed in Sec. 3.1. We compare to Gaussian embedded Non-Local layer [34], Deformable convolution with kernel size 3 [4], and Dilated convolution with dilation 3. The result is shown in Table 4. By comparing to 1×1 convolution, we show that a large receptive field is of central importance. We also see that



(a). the chair with the owl on it. (b). silver car in back of horse car. (c). horse in foreground rider leaning back.

(d). guy by the red wall with arms crossed. (e). the cow standing on the left. (f). train dark and light gray left.

Figure 6. Visualizations of landmark positions and the grounding results. The images on the left side show the landmark positions (red dots) and the center of the predicted box (green dot). Notably, the predicted center receives information from landmark positions. The images on the right side show the ground-truths (yellow boxes) and the predictions (blue boxes).

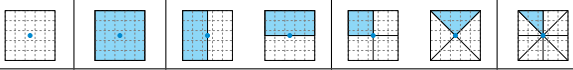
Conv	LFC					
w/o $\mathcal{P}$	$\mathcal{P}=1$	$\mathcal{P}=2$		$\mathcal{P}=4$		$\mathcal{P}=8$
						
62.59	63.90	65.05	65.36	65.48	65.26	65.58

Table 5. Performance on ReferitGame (Pr@0.5) with different $\mathcal{P}$.

Non-Local is inferior to other convolutions, probably due to its limitation of modeling relative spatial relation. With the merits of the global receptive field as well as spatial awareness, our LFC outperforms all of them.

5.5. The Effect of Different Partitions

We study the effect of various partitions, as shown in Table 5. When $\mathcal{P} = 1$, our formulation of DMP (dynamic max pooling) degenerates to a global max pooling, yielding globally equal representations. We are surprised to find that such a simple global representation can boost performance. By considering spatial information, the performance can be further improved but reach saturation when $\mathcal{P} = 4$. We hypothesis that this is dataset-related.

5.6. Visualizations of Results and Landmarks

Beyond effectiveness, the design of the *landmark feature* $h \in \mathbb{R}^c$, which is max pooled from the sub-region, allows us to see where are focused over the whole feature map. In this way, we are able to take a small step toward interpretability in one-stage visual grounding. In particular, we can decode the landmark locations by *argmax*. Since the *landmark features* are pooled from a coarse feature map, to reflect on the original image size, we add a gaussian $\mathcal{G}(\mu, \sigma)$ for each landmark positions, where we choose $\mu = 0, \sigma = 1/3$. It is worth noting that there could be c landmark positions since the dimension of h is c.

We visualize several examples of landmark features of the grounding centers, as well as the grounding results in Figure 6. Many two-stage methods are typically motivated by the fact that the ROI-pooled features can provide better localization for individual objects and filter irrelevant background noise. We show that simply using Max-pooled features has a similar effect, *i.e.* focusing on useful cues, but without resorting to extra supervision. In addition, while two-stage methods hold a strong assumption that contextual cues only come from a pre-defined set of objects, *e.g.* 80 objects in COCO, we show that some of the cues outside of this distribution are also important, such as "red wall" as shown in Figure 6 (d). We return such a degree of freedom to the data itself.

6. Conclusion

In this work, we place emphasis on the relation modeling in one-stage visual grounding, and along this line of thought, propose a novel and simple LBYL-Net that shows competitive results over all state-of-the-art one-stage and two-stage methods. Central to our idea is to model long-range and spatial-aware features with *Dynamic Max Pooling* (DMP) and *Landmark Feature Convolution* (LFC), showing its superiority over related modules. We hope that our proposed LFC can also accelerate related researches, such as visual relation detection.

Acknowledgement. The work was supported by National Key R&D Program of China (2018AAA0100704), NSFC #61932020, Science and Technology Commission of Shanghai Municipality (Grant No. 20ZR1436000), and "Shuguang Program" supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission.

References

[1] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, 2016. 4

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 3

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 3, 7

[5] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *CVPR*, 2018. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 5

[7] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 1997. 2

[8] Ko Endo, Masaki Aono, Eric Nichols, and Kotaro Funakoshi. An attention-based regression model for grounding textual phrases in images. In *IJCAI*, 2017. 2

[9] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017. 1, 2, 6

[10] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, 2016. 2

[11] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 6, 7

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[14] 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. 6, 7

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 4

[16] 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. 6

[17] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 2019. 6

[18] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NeurIPS*, 2017. 4

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 2

[20] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *CVPR*, 2019. 6

[21] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 6

[22] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NeurIPS*, 2016. 3

[23] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 2, 6

[24] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 1, 2, 6

[25] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019. 5, 7

[26] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 5, 7

[27] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *ECCV*, 2018. 6

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2, 5

[29] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *ICCV*, 2019. 1, 2, 6

[30] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015. 6

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[32] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *TPAMI*, 2018. 6

[33] 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. 6

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2, 3, 4, 7

[35] Zhengyang Wang and Shuiwang Ji. Smoothed dilated convolutions for improved dense prediction. In *SIGKDD*, 2018. 3

[36] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019. 4

[37] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *ICCV*, 2019. 6

[38] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *ECCV*, 2020. 2, 5, 6, 7

[39] 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, 2, 5, 6, 7

[40] Raymond Yeh, Jinjun Xiong, Wen-Mei Hwu, Minh Do, and Alexander Schwing. Interpretable and globally optimal prediction for textual grounding using image concepts. In *NeurIPS*, 2017. 2

[41] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2, 3

[42] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017. 3

[43] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 1, 2, 6

[44] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 1, 2, 6

[45] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 6

[46] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. In *IJCAI*, 2018. 6, 7

[47] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, 2018. 2, 6

[48] 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, 6