One for All: One-stage Referring Expression Comprehension with Dynamic Reasoning

Zhipeng Zhang[a,b], Zhimin Wei[a,b], Zhongzhen Huang[a], Rui Niu[a], Peng Wang[a,b,*]

[a] *School of Computer Science, Northwestern Polytechnical University, Xi'an, China*
[b] *National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, China*

Abstract

Referring Expression Comprehension (REC) is one of the most important tasks in visual reasoning that requires a model to detect the target object referred by a natural language expression. Among the proposed pipelines, the one-stage Referring Expression Comprehension (OSREC) has become the dominant trend since it merges the region proposal and selection stages. Many *state-of-the-art* OSREC models adopt a multi-hop reasoning strategy because a sequence of objects is frequently mentioned in a single expression which needs multi-hop reasoning to analyze the semantic relation. However, one unsolved issue of these models is that the number of reasoning steps needs to be pre-defined and fixed before inference, ignoring the varying complexity of expressions. In this paper, we propose a *Dynamic Multi-step Reasoning Network*, which allows the reasoning steps to be dynamically adjusted based on the reasoning state and expression complexity. Specifically, we adopt a Transformer module to memorize & process the reasoning state and a Reinforcement Learning strategy to dynamically infer the reasoning steps. The work achieves the state-of-the-art performance or significant improvements on several REC datasets, ranging from RefCOCO (+, g) with short expressions, to Ref-Reasoning, a dataset with long and complex compositional expressions.

Keywords: Referring Expression Comprehension, Dynamic Reasoning, Reinforcement Learning

[*] Corresponding author

1. Introduction

Referring Expression Comprehension (REC) aims to detect a correct target region in an image described by a natural language expression. Therefore, we argue that the key to this task is to understand the expression and image jointly and correctly and aggregate the extracted relevant features appropriately. Only with the correct generation of the best bounding boxes in the task of the referring expression comprehension, will it benefit the downstream tasks like Visual Question Answering [57, 9, 20, 56, 19, 11] and Vision-Language Navigation [44, 27, 26, 38].

There are mainly two threads of work in Referring Expression Comprehension: two-stage approaches [42, 41, 32, 3, 54, 52] and one-stage approaches [50, 49, 4, 33, 21]. The two-stage methods first generate a series of candidate bounding boxes and then gradually select the best. While the one-stage methods attempt to combine the region proposal and selection stage. The one-stage method is not only simple and effective but also achieves great performance and therefore has become a major approach for the REC task.

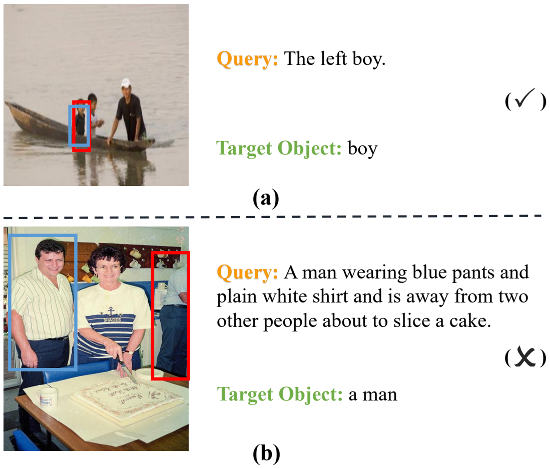

Query: The left boy.
(✓)
Target Object: boy
(a)
Query: A man wearing blue pants and plain white shirt and is away from two other people about to slice a cake.
(✗)
Target Object: a man
(b)

Figure 1: The red and blue boxes represent the ground-truths and predicted regions by the current *state-of-the-art* one-stage method respectively. This task requires the model to ground the ground-truth region (red box) by using the input query. To get the ground-truth bounding box, the existing method needs multi-steps to reason. Existing current methods, such as *ReSC*[49], with fixed reasoning steps will search the wrong object if the expression is long and complex. For example, **(a).** The expression is short, so we only need a few steps for reasoning the correct object. **(b).** When facing the longer and more difficult expression, it needs more steps to reason the target object so existing methods get the wrong bounding box.

However, there are also obvious flaws and shortcomings in the existing one-stage referring expression comprehension (OSREC) approaches [50, 21]. Most of the data processing way is encoding the expression and image as two embedding vectors and then fusing these features via concatenation for further bounding box prediction. A series of OSREC models [13, 31, 48, 49, 37] adopt a multi-hop fusion strategy, to reason over multiple objects mentioned in a complex expression and their spatial relations. Nevertheless, one unsolved issue of these models is that the number of reasoning steps is predefined and fixed during inference, without considering the varying complexities of different expressions. As is shown in Figure 1, with the increase in expression complexity, it accordingly needs more reasoning steps to reach the correct object. In this case, the ***Dynamic Multi-step Reasoning Network*** proposed by us can solve the above problems primely by automatically determining the number of reasoning steps.

As [34, 39, 25], CNNs are the most fundamental backbone for general computer vision tasks, which cover more local information through local convolution within the receptive fields, so as to effectively extract high-frequency representations. Transformer has strong capability of building long-range dependencies and fusing associations, but powerless in capturing high-frequency local information. There are a lot of researches [46, 15, 23] in the field of multimodal on how to combine them. Therefore, we follow their strengths to design the model.

In order to solve dynamic reasoning problems, we use reinforcement learning [35, 18, 5, 8] to monitor the model and make decisions about whether to continue reasoning. Different from the previous method of fixed reasoning times, our dynamic reasoning method can automatically determine the reasoning steps according to the characteristics of the data. As a result, our model can solve the problem that the previous model faced with complex query and image relationship reasoning times are not enough, leading to prediction errors.

Our dynamic reasoning framework, ***Dynamic Multi-step Reasoning Network***, is composed of *Data encoder*, *Fusion Module* and *Dynamic reward Module*, as **Figure 2** shows. Our framework dynamically chooses to continue to reason or not, according to the reasoning state and expression complexity. By introducing reinforcement learning, the recursive reasoning will be adjusted to judge automatically by our *Dynamic Reward Module.* The information of expression and image will be preferably fused by our *Fusion Module* regardless of the length of the expression. We use transformer to associate all features globally, to pay more attention to each detail in the image and expression.

The main contributions of our work are as follows:

(1). We propose a ***Dynamic Multi-step Reasoning Network*** to solve referring expression comprehension tasks of diverse complexities. By adaptively selecting the number of reasoning steps during inference, our model is capable of dealing with varying-complexity expressions using the same set of hyper-parameters without manually tuning.

(2). With no need of manually tuning the reasoning steps, our model achieves the *state-of-the-art* (*SOTA*) performances or significant improvements on a number of REC datasets with different complexities, including RefCOCO+, RefCOCOg and Ref-Reasoning.

2. Related Works

2.1. Referring Expression Comprehension

Referring Expression Comprehension (REC) is a visual-linguistic cross-modal understanding problem. It aims to detect the target object described by a natural language expression in an image. Most previous methods are two-stage methods [42, 41, 32, 3, 54, 52], which generate several region proposals in the first stage. The second stage is to retrieve the objected region matched with the input expression by calculating the similarity. Inspired by *attention mechanism*, A-ATT [6] reasons between information jointly and further considers the self-attention guidance to explore a more diversified interaction among multiple information sources. A graph-based language-guided attention network was proposed by *Wang* [43] to highlight the inter-object and intra-object relationships that are closely associated with the expression for better performance. *Niu* [30] developed a variational Bayesian framework to exploit the reciprocity between the referent and context. Owing to the inaccuracy of region proposals and slow computation speed, more and more work concentrate on the one-Stage method [50, 49, 4, 33, 21]. Compared to the two-stage method, one-stage methods directly predict bounding boxes of the object by densely fusing the features of visual-text at all spatial locations.

Recently, many works have made some advancements in the one-stage method. In the work [50], the author puts forward a one-stage model that fuses an expression's embedding into YOLOv3 object detector augmented by spatial features, and then it uses the merged features to localize the corresponding region. [21] reformulates the REC as a correlation filtering process and puts forward CenterNet [55]. The expression is first mapped from the language domain to the visual domain, and then it is treated as a template (kernel) to perform correlation filtering on the image feature maps. Besides, the peak value in the heatmap is used to predict the center of the target box. Yang [49], uses a recursive *sub-query learner* to enhance the model's ability to integrate text-image features through more steps. Although the one-stage method is effective, it also results in the loss of text features when facing a long and complicated expression. These studies have found that it needs different times instead of more times for a better result when dealing with different lengths of expressions. Hence, we propose a ***Dynamic Multi-step Reasoning Network*** to reason dynamically and customize the most appropriate number of iterations for reasoning the final result. In addition, the previous attention mechanism is local, which is hard to pay attention to all detail. To solve this problem, we use *transformer* to fuse features and propose *Attention Module* for variable-length queries.

2.2. Dynamic Reasoning

Dynamic reasoning has been proposed to solve reasoning tasks. Neural Module Networks (NMNs) [13] are multi-step models that build question-specific layouts and execute. In work [31], the author raised FiLM, a multi-step reasoning procedure that influences neural network computation via a simple and feature-wise affine transformation based on conditioning information. *Hu* [14] proposed Language Conditioned Graph Network (LCGN) model that dynamically determines which objects to collect information from each round by weighting the edges in the graph. At the same time, since reinforcement learning is concerned with how intelligent agents ought to take actions in an environment to maximize the notion of cumulative reward, some works applied it to various vision tasks and obtained remarkable success. Recently, *He* [10] has extended reinforcement learning to the reasoning task that formulates the task of video grounding as a problem of sequential decision making by learning an agent and improving by multi-task learning.

In recent years, more and more one-stage Referring Expression Comprehension (OSREC) have been proposed to tackle the Referring Expression Comprehension (REC) task. Moreover, in consideration of previous one-stage methods' limitations on long and complex expressions, many OSREC models adopt a multi-hop reasoning strategy because a sequence of objects is frequently mentioned in a single long and complex expression. The work in [48] argues to learn the representations from expression and image regions in a progressive manner and performs multi-step reasoning for better matching performance. Inspired by the strategy, *Luo* proposed a Multi-hop FiLM [31] model that recursively reduces the referring ambiguity with different constructed sub-queries to perform multi-step reasoning between the image and language information. However, there is an unsolved issue of these models that the number of reasoning steps needs to be pre-defined, and the results appear to drop significantly when these models process long and complex expressions. Hence, we propose dynamically reason, which allows the reasoning steps to be dynamically adjusted based on the reasoning state and expression complexity for deducing the final result.

3. The approach

In this section, we mainly introduce our one-stage ***Dynamic Multi-step Reasoning Network*** for referring expression task. Given a natural language expression $\{q\}_{n=1}^{N}$, which q is the word of the expression and N is the length of the referring expression, the model is required to detect the described object O in the given image I. Previous one-stage reasoning models [13, 31, 48, 49, 37] with multi-hop reasoning strategies either need to set the same reasoning steps for different instances or have to set different steps for various datasets through several advanced experiments. Nevertheless, detecting the correct objects from various referring expressions with different complexities needs different numbers of reasoning iterations. In addition, it's unrealizable to frequently adjust reasoning iterations and other hyper-parameters in practical use. However, the reasoning iteration steps of our model can change dynamically when the input expressions change.

The overall architecture of our ***Dynamic Multi-step Reasoning Network*** is exhibited in Figure 2. The model can be generally divided into three components: (1) *Data Encoder*: CNN models and BERT are adopted to extract image and expression features, respectively. (2) *Fusion Module*: this module uses *Transformer Encoder* to fuse visual and text features better and learns richly contextual vision-text representations. (3) *Dynamic Reward Module*: a reinforcement learning strategy determining the max reasoning iterations and outputting the detected object O in image I.

3.1. Encoder Module

Given an image $I \in R^{W \times H \times 3}$ and a referring expression $Q = \{q_n\}_{n=1}^{N}$, where q_n represents the n-th word and $W \times H \times 3$ denotes the spatial scale

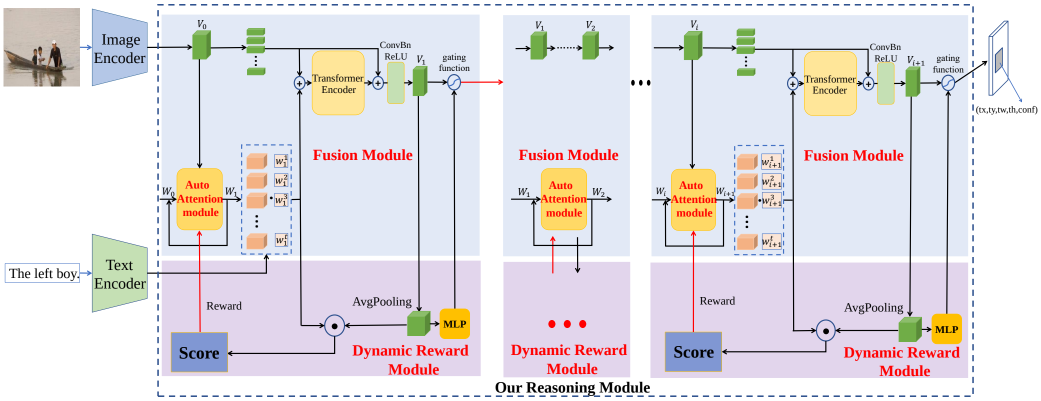


Figure 2: The overall architecture of our model, which contains three main parts: the *encoder module*, the *fusion module* and the *dynamic reward module*. The gating function decides to continue reasoning or output the vision vector, which is activated by MLP in *Dynamic Reward Module*. There are seriatim elaborate descriptions of the three modules in Section 3. Additionally, the sign + in our model means concatenating two vectors, and · means dot-product.

of the image, our goal is to find one sub-region I_S within the image I that corresponds to the semantic meaning of the referring expression Q.

3.1.1. Image Encoder

Following the work [49], our image encoder uses Darknet-53 pertained on COCO object detection dataset as our module's backbone to extract visual features of input images. In the experiment, We use the same initial parameters with *yang* [6, 50] for fairness. Specifically, the input image was fed into the visual encoder network after being resized to the size of 256 × 256 × 3. We select the encoded visual feature from the convolution layer with a dimension 256 × 256 × 3, and then add 1 × 1 convolution layer with batch normalization and *ReLU* to map them all into the same dimension 512. The output feature map is $V = \{v_i\}_{i=1}^{W \times H}$, where W and H mean the weight and height of the image respectively and v_i is the feature V of different regions i, which denotes different local regions for the input image.

3.1.2. Expression Encoder

In this paper, we use the pre-trained BERT model as our expression encoder and train with the same initial parameters [49]. In our module, each word in a referring expression which length is N is firstly mapped to a corresponding word embedding vector with a linear function, giving us $X = \{x_n\}_{n=1}^{N}$. Then, each x_n and its absolute position in the sentence n are fed into a pre-trained BERT model. We summarize the representation of each word in the last layer as the vector of expression and each sentence also comes with special tokens such as [CLS], [SEP] and [PAD]. Finally, we obtain word-level features $E = \{e_n\}_{n=1}^{N}$, $e_n \in R^d$, which the dimension size $d = 768$ and the maximum value of $N = 20$.

3.2. Fusion Module

The visual features of the images are extracted by DarkNet [1]. The referring expression Q is firstly fed into a pre-trained BERT-Base [7] to obtain the original word features $\{e_n\}_{n=1}^{N}$ (dimension, $N \times 768$) and then we feed the word features into an MLP, whose output dimension is 512. Our overall ***Dynamic Multi-step Reasoning Network*** updates both visual and word features into V^t and $\{e_n^t\}_{n=1}^{N}$ step by step which t means the number of current iterations and the max reasoning iteration T is decided by the *Dynamic Reward Module*.

By the inspiration from the previous work [50, 49], we also introduce the *Attention module* into our model to construct a score vector $\{w_n^t\}_{n=1}^{N}$ for word features in each iteration. Otherwise, the history of score vector is recorded to compute a history vector $\{h_n^t\}$ which helps to avoid ignoring some small tips in the referring expression. To be specific, the *Attention module* takes the word feature $\{e_n\}_{n=1}^{N}$, the visual feature V^{t-1} in the last iteration, and the history vector $\{h_n^t\}$ to compute the score.

$$\begin{aligned} h_n^t &= 1 - \mathbf{min}(\sum_{i=1}^{t-1} w_n^i, 1), \\ w_n^t &= \mathbf{softmax}[\mathbf{W}_1^t \mathbf{tanh}(\mathbf{W}_0^t h_n^t (\overline{V}^{t-1} \cdot\ e_n) + b_0^t) + b_1^t] \end{aligned} \tag{1}$$

where $\cdot$ is dot product. $\overline{V}^{t-1}$ is the average pooling of V^{t-1}. $\mathbf{W}_0^t$, b_0^t, $\mathbf{W}_1^t$ and b_1^t are learnable parameters in the *Attention module*. Given the history of previous score vectors, h_n^t can represent how much attention each word in the expression has got so far. Both h_n^t and w_n^i are N-Dimension vectors with values ranging from 0 to 1.

Considering that transformer [40] can make good use of global information, we adopt a 6-layer *Transformer Encoder* which each layer has 8 heads into our *Fusion Module* to update visual features according to the word features. By this connection, we find that the output image feature section will have a strong association with expression by using transformer construct. Multi-modal information is associated globally, which shows superior performance. The original word feature $\{e_n\}_{n=1}^{N}$ is weighted by multiplying the learnable score vector $\{w_n^t\}_{n=1}^{N}$:

$$\widetilde{e}_n = w_n^t * e_n, n = 1, 2, 3, ..., N. \tag{2}$$

We resize the visual feature V^{t-1} to $\widetilde{V}^{t-1}$. The size of feature $\widetilde{V}^{t-1}$ is 256×512.

Then the weighted word features $\{\widetilde{e_n}\}_{n=1}^{N}$ and the visual feature $\widetilde{V}^{t-1}$ are fed into the *Transformer Encoder* [36, 2] to get updated and fused visual features v^t:

$$\begin{aligned} \widetilde{V}^t &= \textbf{TransformerEncoder}([\{\widetilde{e_n}\}_{n=1}^{N} : \widetilde{V}^{t-1}]) \\ \widetilde{V}^t &= [\widetilde{V}^t : \widetilde{V}^{t-1}] \\ V^t &= \textbf{Resize}(\textbf{ConvBNReLU}(\widetilde{V}^t)) \end{aligned} \tag{3}$$

where [:] indicates concatenation. Then the updated and fused visual features are used to calculate the weights of all words when the step is $t + 1$. In the last step, the gating function decides to continue reasoning or output the vision vector, which is activated by the *Dynamic Reward Module.* Our *Fusion Module* uses the visual-text contextual feature as input to output the predicted grounding for the referring expression in the final iteration. Following the work [49], We adopt the same two 1×1 convolutional layers to predict 9 anchor boxes at each location where there are $32 \times 32 = 1024$ spatial locations. For each box, our module predicts five values $t_x, t_y, t_w, t_h, conf$. The last parameter is the confidence score and other parameters represent the relative offsets.

3.3. Dynamic Reward Module

In *Dynamic Reward Module*, we employ policy gradient (PG) [35, 18, 5] to set up reward and punishment mechanisms to optimize the final decision of the model. We resort to the policy gradient for outputting every action's probability value after each iteration to decide whether to continue reasoning or output the V^t.

The *Dynamic Reward Module* comprises two kinds of actions helping to decide whether the system will continue reasoning or not. The action states are dependent on current visual feature V^t, word features $\{e_n\}_{n=1}^N$ and $\{w_n^t\}_{n=1}^N$. The probability values are calculated by:

$$\begin{aligned}
&\hat{e} = e_{cls}, \\
&\hat{V}^t = \mathbf{AvgPooling}(V^t), \\
&actions\_prob = \mathbf{softmax}[\mathbf{W_2}^t\mathbf{tanh}(\mathbf{W_1}^t[\hat{V}^t : \hat{e}] + b_1^t) + b_2^t] \\
&action = \mathbf{argmax}(actions\_prob)
\end{aligned} \tag{4}$$

where *argmax* returns the indices of the maximum values in *actions_prob*. The feature of vision vector V^t passes through the **AvgPooling**. The parameters of the *Attention module* **W** and b are updated and learned under this section reward. They work to compute *actions_prob* and *action*. The *action* includes two kinds of signals, which 0 represents stopping reasoning and 1 represents continuing reasoning. The *actions_prob* represents the predicted probabilities of two actions.

We deal with the input expression at the first token's position and the last token's location ([CLS],[SEP]). We also join the [CLS] as the expression vector representation in MLP from BERT-encoder into our *Dynamic Reward Module* to be better rewarded. The judgment of MLP works with the connected vector.

To train *Dynamic Reward Module*, we set two kinds of reward: the *ultimate reward* and the *immediate reward* and adopt both of them. The details about them are as follows.

3.3.1. Ultimate Reward

The *ultimate reward* represents the reward to the result of reasoning. We use the output of the last reasoning step to regress the bounding box of the target object O. The *ultimate reward* is defined as:

$$r^t_{ultimate} = \begin{cases} 1, & IoU >= 0.5, \\ -1, & otherwise, \end{cases}, \tag{5}$$

where IoU is calculated by comparing prediction bounding boxes with the ground truth bounding box. The *ultimate reward* is 1 if the IoU is greater than 0.5. Otherwise, the *ultimate reward* is -1.

3.3.2. Immediate Reward

The *Immediate Reward* represents the reward to the positive effect of features during reasoning. The reasoning module concentrates on the most related words and visual field and makes visual features and weighted word features more and more relative step by step. Accordingly, we calculate the *immediate reward* between the visual features and weighted word features during all the reasoning steps. The reward is calculated as:

$$\begin{aligned}
&L^t = \sum_{n=1}^{N} w_n^t * e_n, \\
&\hat{V}^t = \mathbf{AvgPooling}(V^t), \\
&Score^t = L^t \cdot \hat{V}^t, \\
&r_{imm}^t = \left\{ \begin{array}{cc} 1, & Score^t - Score^{t-1} >= 0, \\ -1, & otherwise, \end{array} \right. .
\end{aligned} \tag{6}$$

Where $Score^t$ is the relevancy degree of weighted word features and visual features in the t-th reasoning step. The *immediate reward* is 1 if the relevancy degree is increasing. Otherwise, the *immediate reward* is -1.

The data results of these different reward policies are shown in Table 3. The final model framework is compatible with the use of both kinds of rewards in order to achieve the best possible results. The gating function $\mathbf{r^t}$ is also computed in this step. The model will continue to reason if the gating function $\mathbf{r^t}>0$, which means that $r_{imm}^t = r_{ultimate}^t = 1$.

To train the Dynamic Reasoning globally, we use *actions_prob* as the prediction and the *action* is used as our label. Accordingly, we use the weighted CrossEntropyLoss between the softmax over all boxes and a one-hot vector — the anchor box, which has the highest IoU with the ground truth region, is labeled 1 and all the others are labeled 0 as the loss function. The weight is calculated by:

$$\begin{aligned}
&r^t = r_{ultimate}^t + r_{imm}^t \\
&weight^t = \sum_{t=i}^{T} 0.9^{t-i} r^t, i = 1, 2, 3, ..., T
\end{aligned} \tag{7}$$

4. Experiment

In this section, we conduct experiments to analyze our model. Firstly, the proposed model is compared with a variety of REC models on different datasets. Especially, we compare our model with other excellent one-stage methods to analyze the effectiveness of our reasoning architecture and reinforcement learning. Then, a series of ablation experiments are performed to analyze the effectiveness of different rewards, the impact of different iterations and the impact of different transformers.

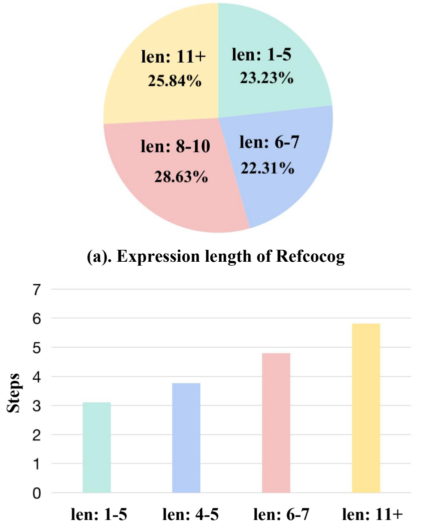
All the experiments are conducted on 8 Nvidia TitanX GPUs. The proposed model is implemented with PyTorch. Following the universal setting [49, 50], we keep the original image ratio and resize the long edge to 256 and then pad the resized image to 256 × 256 with the mean pixel value. The RMSProp optimizer with a learning rate of $1e^{-4}$ initially which decreases by half every 10 epochs is used to train the model. We adopt a batch size of 8 and train the model with 100 epochs. Consistent with previous work, we also use accuracy as the evaluation metric, which is calculated by checking whether the target object is correctly selected or not. Given a language expression, the predicted region is considered as the correct grounding if the Intersection-over-Union(IoU) score with the ground-truth bounding box is greater than 0.5.

4.1. Datasets and Analysis Benchmark

RefCOCO/RefCOCO+/RefCOCOg. RefCOCO [53], RefCOCO+ [53], and RefCOCOg [28] are three Referring Expression Comprehension datasets with images and referred objects selected from MSCOCO [22]. The referred objects are selected from the MSCOCO object detection annotations and divided into 80 object classes. RefCOCO 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. It was collected in a non-interactive setting thereby producing longer expressions than that of the other three datasets which were collected in an interactive game interface. On RefCOCO and RefCOCO+, we follow the general standard split 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 while images in “testB” contain all other objects. RefCOCO+ is similar to RefCOCO where however, absolute location words are forbidden to use , so it takes more effort obviously. To be specific, the expressions of RefCOCO+ do not contain words with positional relationship attributes. For example, “on the right” describes the object’s location in the image. On RefCOCOg, we experiment with the splits of RefCOCOg-umd [29] and refer to the splits as the 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 for RefCOCO, RefCOCO+, RefCOCOg respectively. In fact, with the development of REC tasks, the descriptions become longer and more complex and similar to human real-world social language expressions. So our mission is of great significance.

Ref-Reasoning. Ref-Reasoning [48] is built on the scenes from the GQA dataset [16] and includes semantically rich expressions, which describe objects, attributes, direct relations and indirect relations with different layouts. The numbers of the expression-referent pairs for training, validation and test on the dataset are 721,164, 36,183 and 34,609 respectively. The referring expressions for each image are generated based on the image scene graph using a set of templates and diverse reasoning layouts. In total, there are 1,664 object classes, 308 relation classes and 610 attribute classes in the adopted scene graphs.

We present the statistical visualization results in **Figure 3** using the common division criteria. As shown in **Figure 3**, there is no significant distinction between the percentage of sentences of different range lengths in Refcocog.



(a). Expression length of Refcocog

(b). The average number of inference steps for different lengths.

Figure 3: The proportion of different referring expressions length in RefCOCOg and the analysis of the average number of inference steps required for different lengths.

4.2. Comparison with State-of-the-arts

In the following, previous models are evaluated on different datasets and compared with our proposed model.

The overall accuracy of all evaluated models as well as the proposed model is presented in Table 1. In the table, we can see that compared with previous one-stage methods, our method achieves remarkable improvements. Especially, our model outperforms Sub-query-Base [49] (one-stage baseline) by nearly 2.69%, 2.00% and 1.10% on RefCOCOg-umd [29], RefReasoning dataset [48] and RefCOCO split [53] datasets. Since the expressions in RefCOCOg are generally longer, the great improvements on RefCOCOg mean that our model is very capable of handling long and difficult expressions. For fair comparisons, we don't take account of two-stage methods and Sub-query-Large [49, 45]. As [49] says, two-stage methods highly rely on the region proposals' quality. The COCO-trained detector can generate nearly perfect region proposals on COCO-series datasets for two-stage methods, but not for one-stage methods. When dealing with other datasets, their performances drop dramatically. Relatively speaking, our model is stable across all datasets. As for Sub-query-Large [49], it has a larger image size and another BERT-large in the process of image encoder and text encoder which differ from us. Therefore, we choose Sub-query-Base as the comparison baseline of the experiments. In total, our model achieves *SOTA* performances in REF-reasoning and performs better generally compared with previous one-stage methods and achieves nearly sota performances in RefCOCO(+, g) [17] .

We show the distribution of relatives between iterations of reasoning and the length of referring expression on the RefCOCOg test-umd split in **Figure 3**. According to our statistics, we can conclude that longer and more complex expressions do require more inference steps to make the module integrate features fully. Without a suitable and sufficient number of steps, the model is not able to fuse features adequately, which will affect the subsequent decision in the candidate regions. So one of the reasons for the improvement of accuracy is that we can adapt it to automatically determine the inference step for all expression lengths.

We also visualize some experimental results in **Figure 4** to show the process of reasoning confronted with the strength of expressions. The result shows, **(a).** that the advantage of our model will gradually show up as the length of the expression increases. **(b).** We visualize the two results between ours and Resc's [49]. Our *Dynamic Multi-step Reasoning Network* shows the superior performance when tackling longer expression. The previous work

Table 1: Performance (Acc%) comparison with the state-of-the-art methods and our proposed model on the RefCOCO, RefCOCO+, RefCOCOg and RefReasoning datasets especially with one-stage methods. Our proposed model achieves excellent performance among one-stage models.

Type	Method	Backbone	RefCOCO			RefCOCO+			RefCOCOg		RefReasoning
			val	testA	testB	val	testA	testB	val-u	test-u	val
Two-stage Models	LGARNs [43]	VGG16	–	76.60	66.40	–	64.00	53.40	–	–	–
	MAttNet [52]	Res101	76.40	80.43	69.28	64.93	70.26	56.00	66.67	67.01	–
	DGA [47]	Res101	–	78.42	65.53	–	69.07	51.99	–	63.28	–
	RvG-Tree [12]	Res101	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51	–
	NMTree [24]	Res101	76.41	81.21	70.09	66.46	72.02	7.52	65.87	66.44	–
	SGMN [48]	FasterR-CNN	–	–	–	–	–	–	–	–	25.5
One-Stage Models	SSG[4]	DarkNet53	–	76.51	67.50	–	62.14	49.27	58.80	–	–
	FAOA[51]	DarkNet53	72.54	74.35	68.50	56.81	60.23	49.60	61.33	60.36	–
	One-Stage Bert [50]	DarkNet53	72.05	74.81	67.59	55.72	60.37	48.54	59.03	58.70	–
	Sub-query-Base[49]	DarkNet53	76.74	78.61	71.86	**63.21**	65.94	**56.08**	64.89	64.01	29.5
	Our model	DarkNet53	**76.99**	**79.71**	**72.67**	61.58	**66.60**	54.00	**66.03**	**66.70**	**31.5**

always predicts the wrong bounding box. We believe that in the fusion between image feature and expression feature, without correct reasoning steps, the model can't choose the truth by comparing the various regions at a fixed inference step setting, especially for long expression descriptions.

In 4.3 Ablation Studies, through these, the role played by each part of our model. In the following, we will analyze the role of each part in detail.

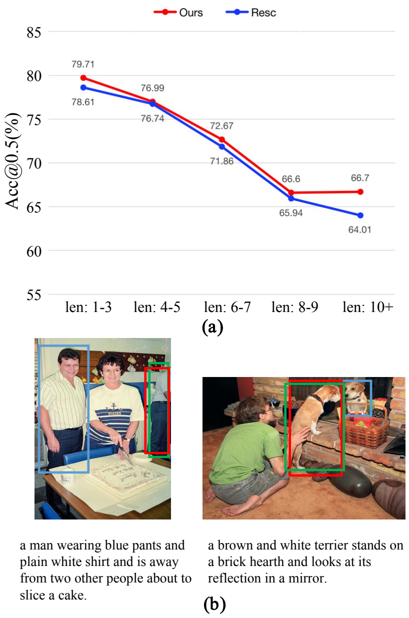
4.3. Ablation Studies

In this section, we elaborate on the details of the ablation studies to prove the superiority and validity of our model. First, we illustrate the impact of different iterations by setting a range of numbers of iterations to show how iterations influence the model's performance. Then, we demonstrate the effectiveness of rewards adopted in our method and the impact of the parameters of the *transformer encoder* by a sequence of comparison experiments.

4.3.1. Impact of Different Iterations

Experiments are conducted on RefCOCOg, which has a wider distribution of referring expressions' length as **Figure 3** shows. We can see how the model with different iterations performs when processing datasets of various referring expression lengths.

Each row in Table 2 displays the result of the experiments where the numbers of iterations are set to 1, 3, 5, 8, 10 respectively. Table 2 shows



a man wearing blue pants and plain white shirt and is away from two other people about to slice a cake.

a brown and white terrier stands on a brick hearth and looks at its reflection in a mirror.

Figure 4: The comparison of accuracy on different lengths of expressions between *ours* and *Resc*. The red region is the true target area and our module proposal region is green. The blue circled region is detected by the previous work Resc model. We can see that the error will easily happen when facing the long and difficult expression.

that, with the maximal iterations increasing, the result will improve in general. But if the maximal iteration increases too much, the performance will decrease instead. It is worth mentioning that all the current works also confirm the phenomenon in our experiment, i.e., the bigger number of iterations does not mean better performance. If the number of iterations is set too much like 8, 10, etc. for better performance in long queries, it will sacrifice the reasoning time and performance in short queries. In the meanwhile, the experiment further demonstrates that our *Dynamic Multi-step Reasoning Network* can make correct decisions for variable-length queries and customize the most appropriate number of iterations for deducing the final result, which has stronger robustness than the previous models with a fixed number of iterations.

4.3.2. Effectiveness of Different Rewards

To validate the effectiveness of *ultimate reward* and *immediate reward*, we compare it with a model without any rewards and a model without *im-*

Table 2: The performance (Acc%) of the model under different iterations.

Method	RefCOCOg	
	val-u	test-u
our model (without RL, 1 pass)	60.40	60.73
our model (without RL, 3 pass)	65.81	66.15
our model (without RL, 5 pass)	**66.01**	**66.56**
our model (without RL, 8 pass)	65.72	66.08
our model (without RL, 10 pass)	65.43	65.87

mediate reward. As is shown in Table 3, the result improves by 0.53% on the test-umd split with the *ultimate reward* and the result improves by 0.14% with *immediate reward*, which as the whole contributes to an improvement of **0.77%**. It suggests that compared with *ultimate reward* and *immediate reward* can fulfill predominant improvement, which simultaneously indicates that our reasoning module can focus on the most related words and visual field, and make visual features and weighted expression features more and more relative step by step.

Table 3: The performance (Acc%) of the model with different rewards. The final model with both two reward mechanisms applied presents superior performance.

Method	RefCOCOg	
	val-u	test-u
our model + without rewards	65.24	65.56
our model + ultimate reward	66.01	66.59
our model + immediate reward	65.81	66.23
our final model	**66.03**	**66.70**

4.3.3. Impact of the Parameters of Transformer Encoder

We also conduct experiments on the RefCOCOg dataset with different numbers of the layers and heads of Transformer [40] to figure out the influence of different *Transformer Encoders*. Table 4 lists out the results of different settings of the parameters, which show that the performance will improve with the increment of the layer and head. But for the maximum setting, the performance will also decrease. Additionally, by contrasting the results with the same number of heads and a different number of layers (for example, row 1 and row 2 or row 3 and row 4), we observe that the increment of layers is more crucial for the performance improvement. This experimental result clearly indicates that our proposed *Transformer Encoder* is capable of extracting and fusing the information from images and expressions, which plays a vital part in the subsequent dynamic inference process. Meanwhile, *Transformer Encoder* can achieve great performance for further extraction of the fused information at each iteration. But to some extent, the constant increment will cause performance degradation, like the result of row 5 in Table 4.

Table 4: The performance (Acc%) of our model under different Transformer Encoders with different parameters without containing the Dynamic Reward module.

Method	RefCOCOg	
	val-u	test-u
our model(without RL, 1 layer 1 head)	48.87	48.93
our model(without RL, 6 layer 1 head)	64.09	64.25
our model(without RL, 1 layer 8 head)	49.89	49.95
our model(without RL, 6 layer 8 head)	**65.52**	**65.70**
our model(without RL, 8 layer 16 head)	64.19	64.61

4.4. Qualitative Results Analyses

In order to analyze the advantages and weaknesses of our method, we present a visualization of the model processing as a way to explain how our model works in every step for the task in **Figure 5**. On the left of Figure 5, we can see that the performance of our BERT-Based attention mechanism expression encoder vector is enhanced gradually in the processing of representations. In addition, we can see that our attention region is getting smaller and more precise with multiple rounds of inference on the right. It will be very important for downstream tasks like robotics, UAV vision language navigation [44, 27, 26, 38], etc.

Finally, Figure 6 shows the successful and failed cases of our method and comparison. Our method performs better when the expressions are long and complicated in most instances. As is shown in (a) (b) (c), the effect of our

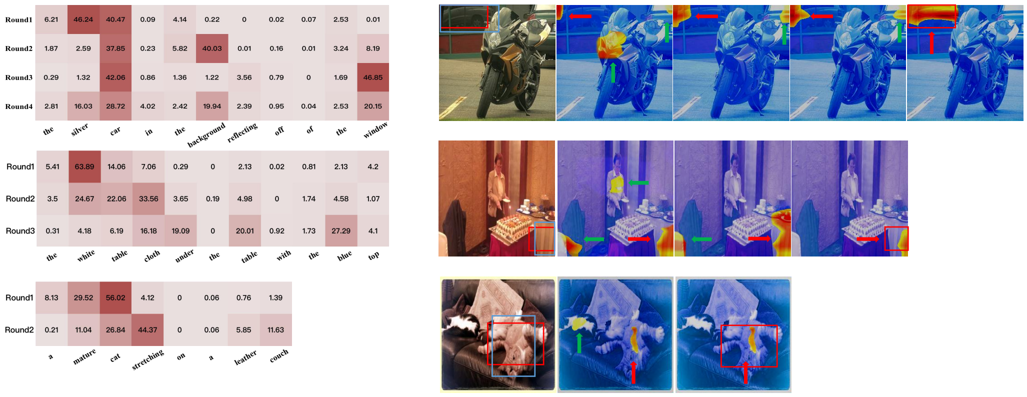
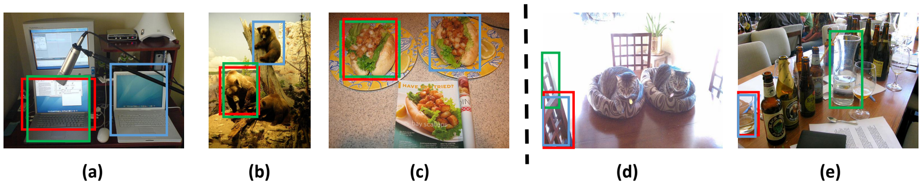


Figure 5: Visualization of one-stage referring expression comprehension with dynamic reasoning and visual feature at each step. Red/ Blue boxes are the predicted regions/ ground truth. The red arrow and the green arrow point to the target and the major distracting object on heatmaps respectively. Our model can make decisions to use different rounds for reasoning the target in the situation of various lengths of queries.

method is obviously improved. The right pictures (d) and (e) are two failure cases of our method. In (d), the error occurs because the multiple relations of the objects are blurred and the spatial position information is adjacent. In (e), the detection fails to result from the ambiguity of the objects' description.



Query:

(a). a laptop of grey color which is kept on on the table with pages opened.
(b). bear in the woods with a cub on a rock and a cub climbing a tree.
(c). the sandwich which has lettuce hanging out of it on the top left of the bun.
(d). an empty chair directly to the left of the leftmost of two cats and behind a second empty chair that is closer to the foreground.
(e). a glass vase with clear water in it , next to a wine glass with brown beer in it on a brown wood table.

Figure 6: The successful and failed cases of our method and the comparison between the performance of our method and the current *state-of-the-art* one-stage method in handling long and complex queries. Green/blue boxes are ground-truths/predicted regions by the current *state-of-the-art* one-stage method, and the red ones represent predicted boxes by our model. The three pictures on the left are successful, and the right pictures are some failures in our method.

5. Conclusion

In this work, we propose a *Dynamic Multi-step Reasoning Network*, which solves the issue that exists in one-stage methods, the unsure numbers of reasoning steps. We use the same hyper-parameters for all of the above Datasets tasks without tuning the number of reasoning steps for each. Experiments show that with the effective integration of Transformer module and Reinforcement Learning strategy, our model can automatically determine the number of inferences and achieve the *state-of-the-art* performance or significant improvement on several REC datasets universally in spite of the length and complexity of expression. We hope this method will enlighten the community to move forward with the research of one-stage referring expression comprehension.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[3] Kan Chen, Rama Kovvuri, Jiyang Gao, and Ram Nevatia. Msrc: Multimodal spatial regression with semantic context for phrase grounding. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 23–31, 2017.

[4] Xinpeng Chen, Lin Ma, Jingyuan Chen, Zequn Jie, Wei Liu, and Jiebo Luo. Real-time referring expression comprehension by single-stage grounding network. *arXiv preprint arXiv:1812.03426*, 2018.

[5] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2951–2960, 2017.

[6] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7746–7755, 2018.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Saite Fan, Xinmin Zhang, and Zhihuan Song. Reinforced knowledge distillation: Multi-class imbalanced classifier based on policy gradient reinforcement learning. *Neurocomputing*, 463:422–436, 2021.

[9] Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1811–1820, 2017.

[10] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8393–8400, 2019.

[11] Jongkwang Hong, Sungho Park, and Hyeran Byun. Selective residual learning for visual question answering. *Neurocomputing*, 402:366–374, 2020.

[12] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[13] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813, 2017.

[14] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10294–10303, 2019.

[15] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002, 2020.

[16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.

[17] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr – modulated detection for end-to-end multi-modal understanding. 04 2021.

[18] Shauharda Khadka and Kagan Tumer. Evolution-guided policy gradient in reinforcement learning. *arXiv preprint arXiv:1805.07917*, 2018.

[19] Mingrui Lao, Yanming Guo, Nan Pu, Wei Chen, Yu Liu, and Michael S. Lew. Multi-stage hybrid embedding fusion network for visual question answering. *Neurocomputing*, 423:541–550, 2021.

[20] Qing Li, Jianlong Fu, Dongfei Yu, Tao Mei, and Jiebo Luo. Tell-and-answer: Towards explainable visual question answering using attributes and captions. *arXiv preprint arXiv:1801.09041*, 2018.

[21] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10880–10889, 2020.

[22] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Xudong Lin, Gedas Bertasius, Jue Wang, Shih-Fu Chang, Devi Parikh, and Lorenzo Torresani. Vx2text: End-to-end learning of video-based text generation from multimodal inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7005–7015, 2021.

[24] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. Learning to assemble neural module tree networks for visual grounding. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4672–4681, 2019.

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[26] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*, 2019.

[27] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6732–6740, 2019.

[28] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[29] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016.

[30] Yulei Niu, Hanwang Zhang, Zhiwu Lu, and Shih-Fu Chang. Variational context: Exploiting visual and textual context for grounding referring expressions. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):347–359, 2019.

[31] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[32] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–264, 2018.

[33] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4694–4703, 2019.

[34] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *arXiv preprint arXiv:2205.12956*, 2022.

[35] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[36] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[37] Wei Suo, Mengyang Sun, Peng Wang, and Qi Wu. Proposal-free one-stage referring expression via grid-word cross-attention. 05 2021.

[38] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*, 2019.

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[41] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2018.

[42] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1960–1968, 2019.

[44] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–53, 2018.

[45] Yuechen Wang, Wengang Zhou, and Houqiang Li. Fine-grained semantic alignment network for weakly supervised temporal language grounding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 89–99, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[46] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[47] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4644–4653, 2019.

[48] Sibei Yang, Guanbin Li, and Yizhou Yu. Graph-structured referring expression reasoning in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9952–9961, 2020.

[49] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. *arXiv preprint arXiv:2008.01059*, 2020.

[50] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4683–4693, 2019.

[51] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4682–4692, 2019.

[52] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018.

[53] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 69–85, Cham, 2016. Springer International Publishing.

[54] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7282–7290, 2017.

[55] Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. Reasoning visual dialogs with structural and partial observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6669–6678, 2019.

[56] Qi Zhu, Chenyu Gao, Peng Wang, and Qi Wu. Simple is not easy: A simple strong baseline for textvqa and textcaps. *arXiv preprint arXiv:2012.05153*, 2020.

[57] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016.