Visual Grounding via Accumulated Attention

Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, Mingkui Tan

Abstract—Visual Grounding (VG) aims to locate the most relevant object or region in an image, based on a natural language query. Generally, it requires the machine to first understand the query, identify the key concepts in the image, and then locate the target object by specifying its bounding box. However, in many real-world visual grounding applications, we have to face with ambiguous queries and images with complicated scene structures. Identifying the target based on highly redundant and correlated information can be very challenging, and often leading to unsatisfactory performance. To tackle this, in this paper, we exploit an attention module for each kind of information to reduce internal redundancies. We then propose an accumulated attention (A-ATT) mechanism to reason among all the attention modules jointly. In this way, the relation among different kinds of information can be explicitly captured. Moreover, to improve the performance and robustness of our VG models, we additionally introduce some noises into the training procedure to bridge the distribution gap between the human-labeled training data and the real-world poor quality data. With this "noised" training strategy, we can further learn a bounding box regressor, which can be used to refine the bounding box of the target object. We evaluate the proposed methods on four popular datasets (namely ReferCOCO, ReferCOCO+, ReferCOCOg, and GuessWhat?!). The experimental results show that our methods significantly outperform all previous works on every dataset in terms of accuracy.

Index Terms—Visual Grounding, Accumulated Attention, Noised Training Strategy, Bounding Box Regression

1 Introduction

Visual Grounding (VG) has attracted a lot of attention in recent years [1,2,3,4,5,6]. Unlike object detection which aims to detect the objects or the regions of interest given the pre-defined class labels, VG aims to locate a specific object in the image, based on a query in the form of natural language. In practice, VG is an important technique for machine intelligence. For example, it can be widely used in the visual understanding system and dialogue system of new generation intelligence devices such as home robots and autonomous vehicles; it can also be embedded into virtual assistants in PCs and smartphones.

However, in many real-world VG applications, the queries can be very complex and the images often have complicated scene structures, making the joint reasoning between these two kinds of information very challenging. To illustrate the challenge, we show a practical example in Figure 1. In this example, the target object is the surfboard specified by the yellow box. There are many irrelevant concepts in the query, such as "beach", "woman", "dogs", "right". As a result, a model needs to understand the relation among those concepts and localize their positions so as to focus its attention on the correct image regions. Moreover, the image contains multiple objects, including four surfboards with similar shape, size and color, which requires the model to have a strong ability in dealing with noisy information.

In practice, VG is commonly formulated as a multiple-choice problem over a set of object proposals, where the proposal can be either human-labeled or detected by an object detector such as Faster RCNN [7] or SSD [8]. A general workflow for VG is to first construct a feature representation for each object proposal as well as the input query. After that, a matching score is obtained for each proposal-query pair based on their representations, and then the object proposal with the largest matching score is selected as the final prediction. Note that in general, the human-labeled object proposals have accurate and reliable bounding boxes. But in practice, we may have only detected object proposals using off-the-shelf detectors, and these proposals may contain substantial noisy bounding boxes, making the VG task

The surfboard on the beach with a woman and two dogs right next to it.

Fig. 1. A typical visual grounding example. Given the image and the query, we are asked to locate the target object in the image that is specified by the query, as the surfboard outlined by the yellow box.

- *Mingkui Tan, Chaorui Deng and Qingyao Wu are with the School of Software Engineering, South China University of Technology, China. E-mail: mingkuitan@scut.edu.cn; secrdyz@mail.scut.edu.cn; qyw@scut.edu.cn*
- *Qi Wu is with the School of Computer Science, The University of Adelaide, Australia. E-mail: qi.wu01@adelaide.edu.au*
- *Fan Lyu is with the College of Intelligence and Computing, Tianjin University. E-mail: fanlyu@tju.edu.cn*
- *Fuyuan Hu is with the School of Electronic & Information Engineering, Suzhou University of Science and Technology, China. E-mail: fuyuanhu@mail.usts.edu.cn*
- *Qi Wu is the co-first author, Mingkui Tan is the corresponding author.*

Manuscript received April 19, 2005; revised August 26, 2015.

more challenging in real-world scenario.

Many previous methods of VG put emphasis on the matching step. For example, in [1], the authors obtain the visual feature of each object proposal o, and feed them along with the global image feature I into a caption generator (*i.e.*, an LSTM [9]) to compute the conditional probability $P(s|o, I)$ of reconstructing the input query. The probabilities are then utilized as the matching score to rank the proposals. In [5], besides using a caption generator to perform the matching step, the author further trains two MLPs (Multi-Layer Perceptron) to project the paired object and query representations into the same dimensional space, where the distance between the paired representations is adopted as another ranking metric. However, these methods take fewer efforts in improving the feature representations for VG, *i.e.*, when constructing the feature of an object proposal, they simply adopt its CNN (Convolutional Neural Network) feature, without considering the relations among each object proposal. Similarly, they compress the whole input query into a static context vector through an LSTM, which has been shown to be problematic due to the loss of information [10, 11]. Moreover, the object proposal and query are processed separately before sent into the matching module, which may hamper the joint reasoning between the visual and textual information.

To this end, we propose an **Accumulated Attention (A-ATT) mechanism**, which provides an attention module for each kind of input information (*i.e.*, query, image, and object proposals) to focus on the essential elements of the information. Moreover, these attention modules transfer knowledge with each other to refine their attention in a circular manner, where the attended feature representation of one information source will serve as an "attention guidance" when updating the attention weights for other types of information. Hence, A-ATT mechanism explicitly captures the latent relations among different information sources and reduces the information redundancies, thus it is able to construct compact and informative feature representations.

Apart from this, existing methods suffer from another problem, *i.e.*, they train their VG models with the human-labeled object proposals (which have accurate bounding boxes). However, when applied on data with detected object proposals (the bounding boxes are very noisy), their performances degrade severely due to the huge distribution gap between the training and inference data. This problem deters the application of many VG algorithms in real-world scenarios because of the lack of human-labeled object proposals. Directly training these VG models with the detected object proposals, however, can only lead to poorer performance, due to the inferior performances of object detectors compared with human beings.

To alleviate this problem, we further propose to train VG models with "**noised**" human-labeled object proposals, where the bounding boxes of the object proposals are randomly shifted, scaled, and resized by a small extent to simulate the detected object proposals. Moreover, this noised bounding boxes further enable us to learn a bounding box regressor after the matching step of VG to refine the best-matched proposals, therefore improving the performance on VG data with detected object proposals. In this sense, we propose a noised training strategy, which contains three stages, *i.e.*, bounding box augmentation, bounding box regression, and end-to-end fine-tuning. In our experiments, the noised training strategy significantly improves the performance of our A-ATT models on VG tasks with detected object proposals. More critically, it can also be applied in many other VG methods and bring clear gains in the performance. More importantly, the noised training strategy increases only negligible computation complexity. Thus, it can be adopted as a general training paradigm for VG.

Our main contributions are summarized as follows: **Firstly**, we propose a novel Accumulated Attention (A-ATT) mechanism to jointly model the complex relations among multiple kinds of information, and apply it on the VG task. **Secondly**, we propose to use a "noised" training strategy to bridge the distribution gap between the training data and the real-world VG data, which can be adopted as the basic configuration for **general VG models** . **Thirdly**, we evaluate the proposed methods on four datasets, *i.e.*, ReferCOCO [4], ReferCOCO+ [4], ReferCOCOg [12] and GuessWhat?! [13] and show that our methods outperform previous best results by a large margin in terms of accuracy.

This paper extends our CVPR paper [14] with the following new contents: **1)** we extend A-ATT mechanism based on the idea of self-attention; **2)** we propose a "**noised**" training strategy to improve the generalization performance for general VG models on data with detected object proposals; **3)** More experimental results are provided, including extensive comparisons on testing data with detected object proposals and more ablation studies.

2 Related Works

As a research direction across vision and language, VG has benefited from the development of Convolutional Neural Networks (CNNs) [15, 16], Recurrent Neural Networks (RNNs) [9, 17], and other research areas such as Image Captioning [18, 19, 20], Visual Question Answering (VQA) [21, 22, 23, 24], and Object Detection [7, 8, 25].

Vision and Language The interplay of vision and language has been studied extensively in recent years, and lots of new tasks have been proposed to promote the research and push the boundaries of both fields, such as Image Captioning, Visual Grounding (VG), and Visual Question Answering (VQA). Research on these tasks has produced plenty of powerful methods for joint reasoning among the visual and the textual inputs, providing valuable insights and solid foundations for the following studies.

Image Captioning takes an image as input and aims to generate a natural language caption to describe it. Most methods in this research area adopt the encoder-decoder architecture [11, 20, 26], where the encoder encodes information from the image (with a CNN) into a context vector, and the decoder then decodes the context vector into a sequence of word tokens through an RNN.

VQA requires reasoning over visual concepts of the image and general knowledge to infer the correct answer for a natural language question. A simple baseline for VQA [27] is to use CNNs and RNNs to learn representations of images and sentences in a common feature space, following by feeding those representations jointly into a matching module that selects the correct answer. To improve the performance

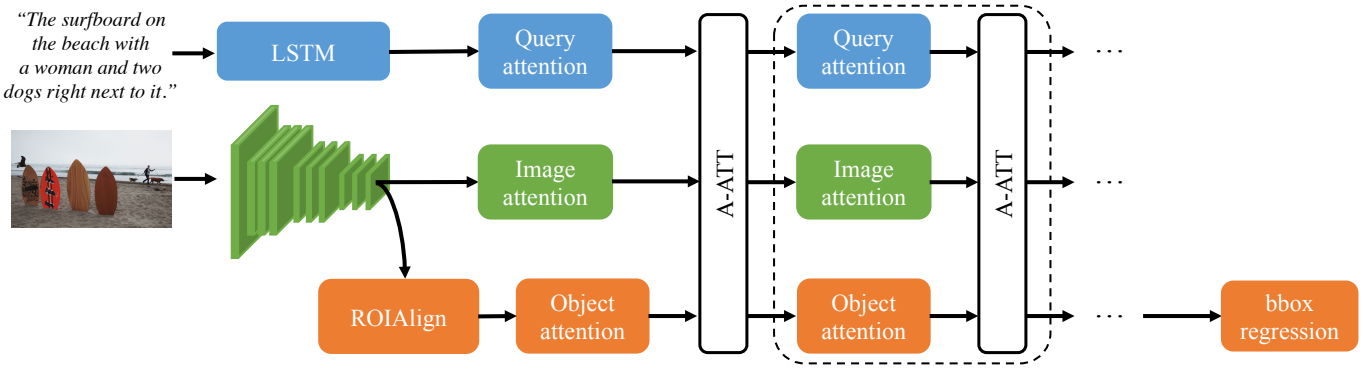


Fig. 2. The overall architecture of the proposed methods. The input object proposals are randomly jittered before sending into the object feature extractor to extract their feature. We use three attention modules to handle the attention problems for image, query, and object proposals. The A-ATT mechanism can be performed for multiple rounds to ensure a sufficient communication among different information, as shown in the dashed box. After the last round of A-ATT, we feed the attended proposal into a bounding box regressor to refine its bounding box.

on VQA, many novel methods have been proposed, such as Neural Module Networks [28, 29, 30], Dynamic Memory Networks [31, 32], and methods using external knowledge bases [33].

Visual Grounding The VG task requires a model to respond to a query by specifying its corresponding object in an image. Generally, VG is formulated as a multiple-choice problem over a set of pre-defined object proposals, thus a matching module is required to match the target object proposal with the input query and image. Some methods [1, 2, 4, 34, 35] view the VG task as a reverse of the image captioning task, where they feed each proposal into an image captioning model to calculate the probability of generating the input query. Then, the proposal that yields the largest probability is selected as the target object. Some methods [6,36,37], on the other hand, seek to directly embed the target object proposal with the input image and query into a close region in a multi-model representational space. Taking advantages of both the above mentioned approaches, Yu *et al.* [5] introduce a Speaker-Listener-Reinforcer model which takes the image captioning model as the speaker, the joint embedding model as the listener, and performs query-object matching separately in both the speaker module and the listener module.

More recently, Zhang *et al.* [38] propose a Variational Context model which exploits the relation between the image and query to improve the context information by a variational Bayesian method. In [39], the authors design three matching modules (*i.e.*, subject module, location module, and relationship module) to handle the matching problem in three aspects. They adopted more powerful feature extractors to process the input information. Unlike these methods, we perform cross-modal attention to reason among different kinds of information with multiple steps. This idea is later adopted in [40], where a Multi-hop FiLM model is proposed to perform multi-step reasoning between the image and language information. Different from Multi-hop FiLM, our A-ATT mechanism reasons among three types of information (*i.e.*, image, query, and object proposals) jointly and further considers the self-attention guidance to explore a more diversified interaction among multiple information sources.

Attention Mechanism The attention mechanism has become a research hot-spot, for its simplicity and effectiveness in dealing with multi-modal information. The general idea is that when constructing a representation for a sequence of information, maybe only a small subset of the sequence is relevant to the downstream tasks. Therefore, we can explicitly learn to pay our attention to the most relevant elements in the sequence according to some "attention guidance" so as to boost the performance of the downstream tasks. In practice, it has already been applied to a wide range of Computer Vision and Natural Language Processing tasks [10, 11, 41, 42, 43, 44, 45]. Many prior works [46, 47, 48, 49, 50, 51, 52] in these years seek to make some advancements in attention mechanism, where they focus on hierarchically-structured attention mechanism or memory-based attention mechanism. In [43,53], the authors proposed the Self Attention mechanism which captures the latent relations among the different elements of a sequence of information.

Object Detection Currently, VG still relies on a pre-trained object detector to generate a set of object proposals, and the performance of VG methods can be heavily affected by the quality of the detected object proposals. In an extreme case, a VG task will never succeed if its target object is missed by the pre-trained detector. Therefore, it is crucial to ensure the pre-trained detector to have a high recall rate, especially for the target object. Some object detectors [7,8,54] can generate relatively accurate object proposals for every object in the image. There are also many faster but less accurate object detectors, such as MultiBox [55], BING [56], and Selective Search [57], where they have to increase the number of proposals to improve the recall rate. Another strategy that can effectively improve the recall rate is to use a bounding box regressor [58] to predict a correction offset and use it to refine the predicted bounding box, so as to increase the Intersection over Union (IOU) scores between the predicted and the ground-truth bounding box of an object proposal.

3 Proposed Method

Given an image I, a query Q, and N object proposals $\{o_1, o_2, \ldots, o_N\}$, visual grounding aims to learn a hypoth-

esis $\mathcal{H}$ that maps Q and I to the target object o^*, *i.e.*, $\mathcal{H}(I,Q) \rightarrow o^*$. In practice, VG is a challenging task due to the complex correlations and heavy information redundancies in the image, query and object proposals.

To tackle this, we first extract a sequence of feature from Q, I and $\{o_i\}$, respectively. Then, for each sequence, we employ our Accumulated Attention (A-ATT) mechanism to assign attention weights for its elements with the other two feature sequences as the attention guidance. In this way, the information redundancies can be reduced, and the latent relations among different feature sequences can be explicitly captured. Afterward, the object proposal with the largest attention weight is selected as the prediction $\hat{o}$. We further refine its bounding box with a bounding box regressor. The overall model architecture is illustrated in Fig. 2.

In the following, we first introduce the feature encoding for image, query, and object proposals in Sec. 3.1. Then, our core algorithm, the **Accumulated Attention mechanism**, is introduced in Sec. 3.2. In Sec. 3.4, we demonstrate how to handle the inconsistent data distribution during training and inference with a "**noised**" training strategy.

3.1 Feature encoding

3.1.1 Query feature

In real-world visual grounding applications, the query may have multiple forms. For example, it can be a short phrase like "red hat", a sentence "the hat on a woman with a black jacket", or even a complex dialogue [13]. Take the sentences in ReferCOCO [4] as an example. A sentence consists of a sequence of words $Q = \{q_1, q_2, ..., q_T\}$ (T is the number of words). We first use a word embedding layer to encode each word q_t into a fix-length vector $\boldsymbol{q}_t$. Then, we feed the encoded sentence into an LSTM [9] and collect the hidden state $\boldsymbol{h}$ at every recurrent step as the sentence feature. For more complex queries (*e.g.*, multi-round dialogues), we can adopt a hierarchical recurrent architecture [46] to model the query in multiple levels. Besides, we can also use the bidirectional LSTM to obtain a better representation of the query information. Denote the query feature sequence as $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_T\}$, where $\boldsymbol{s}_t = \boldsymbol{h}_t$ is the feature representation for t-th word in the sentence.

3.1.2 Image feature

In practice, we can use Convolutional Neural Networks (CNNs), such as VGG-16 [16] and ResNet-101 [15], to extract the image feature. Without loss of generality, we construct the image feature sequence $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_L\}$ using the feature maps $\boldsymbol{M} \in \mathbb{R}^{w \times h \times c}$ at the last convolutional layer of the feature extractor, where each $\boldsymbol{v}_i \in \mathbb{R}^c$ corresponds to a region in the image, and the sequence length $L = w \times h$.

3.1.3 Object proposal feature

Following [13], we represent the object proposal with two kinds of feature: the local feature and the spatial feature. Unlike many previous methods [1,4,6,35] which extract the local feature $\boldsymbol{o}^l$ by cropping the corresponding image region of the object proposal and feeding it into a deep CNN, we use RoIAlign [25] for its efficiency and its comparable performance. Given the input feature map $\boldsymbol{M}$ and an object proposal, the bounding box of the proposal is first divided into $k \times k$ bins, then we average-pool the pixel values in each bin to generate a $k \times k$ output feature map:

$$\boldsymbol{o}^l(i,j) = \frac{1}{n_{ij}} \sum_{p \in \text{bin}(i,j)} \boldsymbol{M}(p), \quad i,j \in \{1,...,k\}, \tag{1}$$

Here, p indicates the coordinate of the pixels, and n_{ij} is the number of pixels in bin(i,j). Specifically, p is fractional to avoid the quantization of the boundary of the bins so that the misalignment between the extracted feature and the object proposal can be reduced. Then, we use bilinear interpolation to compute the exact pixel value at p.

Afterward, the spatial feature $\boldsymbol{o}^s$ is represented by:

$$\boldsymbol{o}^s = [\frac{x}{w_{\text{img}}}, \frac{y}{h_{\text{img}}}, \frac{w_{\text{box}}}{w_{\text{img}}}, \frac{h_{\text{box}}}{h_{\text{img}}}], \tag{2}$$

where (x,y) is the coordinate of the top-left corner of the bounding box, and w_{img} (h_{img}) and w_{box} (h_{box}) denote the width (height) of the image and the bounding box, respectively.

Note that the dimension of local feature $\boldsymbol{o}^l \in \mathbb{R}^{k \times k \times c}$ is usually orders of magnitude larger than the dimension of $\boldsymbol{o}^s \in \mathbb{R}^4$. Thus, the model tends to be dominated by the local information of the object proposal. To tackle this, we apply a depth-wise separable convolutional layer [59], denoted by f_{ds}, on $\boldsymbol{o}^l$ to reduce the spatial size and channel number. Specifically, the kernel size and output channel of f_{ds} are set to k and c' ($c' < c$), respectively, and the output of $f_{ds}(\boldsymbol{o}^l)$ is a c'-dimensional vector. One may also use a normal convolutional layer to replace f_{ds}, but it leads no clear gain in practice. Last, to obtain the feature of an object proposal, we follow [6,60] and fuse the local feature $\boldsymbol{o}^l$ and spatial feature $\boldsymbol{o}^s$ by

$$\boldsymbol{o} = \boldsymbol{W}[f_{ds}(\boldsymbol{o}^l) \otimes \boldsymbol{o}^s] \tag{3}$$

The above fuse strategy help to facilitate the element-wise interactions between $\boldsymbol{o}^l$ and $\boldsymbol{o}^s$. Here, $\otimes$ denotes outer product, $[\cdot]$ vectorizes the matrix in vector, and $\boldsymbol{W}$ is a linear transform. The whole process is illustrated in detail in Fig. 3. Denote the collection of the object proposal representations as $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_N\}$.

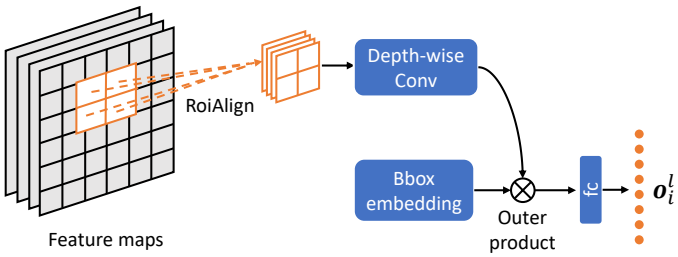


Fig. 3. We construct the local feature for an object proposal directly from the feature maps of the whole image with RoIAlign. We then use depth-wise separable convolution to reduce the dimension of the local feature, and fuse the spatial feature and the local feature by taking their outer product following by a linear transformation.

3.2 Accumulated Attention (A-ATT) mechanism

In this section, we demonstrate how to obtain compact and informative feature representations for multiple information

sources jointly by constructing extensive interactions among them through the A-ATT mechanism.

Accumulated Attention. Assume that we have extracted C feature sequence $\{\boldsymbol{X}_i\}_{i=1}^{C}$ from C kinds of information, where $\boldsymbol{X}_i = \{\boldsymbol{x}_{i1}, ..., \boldsymbol{x}_{in_i}\} \in \mathbb{R}^{d_i \times n_i}$ is a feature sequence with feature dimension d_i and sequence length n_i. The A-ATT mechanism employs an attention module for $\boldsymbol{X}_i$ to calculate attention weights for its elements, with the guidance from other attention modules:

$$\begin{aligned} H_{ij} &= \tanh(\boldsymbol{W}_i^\top \boldsymbol{x}_{ij} + \sum_{c \neq i}^{C} \boldsymbol{W}_c^\top \tilde{\boldsymbol{x}}_c + \boldsymbol{b}), \\ \alpha_{ij} &= \frac{\exp(\boldsymbol{W}_h^\top H_{ij})}{\sum_{k=1}^{n_i} \exp(\boldsymbol{W}_h^\top H_{ik})}, \quad j = 1, 2, ..., n_i. \end{aligned} \tag{4}$$

Here, α_{ij} is the attention weight for the j-th item in $\boldsymbol{X}_i$, and $\{\tilde{\boldsymbol{x}}_c | 1 \leq c \leq C, c \neq i\}$ denotes the "**attention guidance**" provided by other attention modules, which are initialized as zero tensors if they are not available. $\boldsymbol{W}_i \in \mathbb{R}^{d_i \times d_h}$, $\{\boldsymbol{W}_c \in \mathbb{R}^{d_c \times d_h}\}_{c \neq i}^{C}$, $\boldsymbol{W}_h$ and $\boldsymbol{b} \in \mathbb{R}^{d_h \times 1}$ are learn-able parameters related to each information source, and are shared across all attention modules (d_h is the dimension of H). Then, we compute the summarized representation $\tilde{\boldsymbol{x}}_i$ of feature sequence $\boldsymbol{X}_i$ by:

$$\tilde{\boldsymbol{x}}_i = \sum_{j=1}^{n_i} \alpha_{ij} \boldsymbol{x}_{ij}, \tag{5}$$

where $\tilde{\boldsymbol{x}}_i$ is then serving as the attention guidance for other attention modules.

Multi-round Accumulated Attention. For simplicity, we denote the attention process in Eqn. (4) and (5) as $\text{ATT}(\cdot)$. Then, for each feature sequence $\boldsymbol{X}_i$, with the input attention guidance $\{\tilde{\boldsymbol{x}}_c\}_{c \neq i}$, we can obtain its attended feature through:

$$\tilde{\boldsymbol{x}}_i = \text{ATT}_{\boldsymbol{X}_i}(\{\tilde{\boldsymbol{x}}_c\}_{c \neq i}). \tag{6}$$

Obviously, the computation in Eqn. (6) forms a circulation: the summarized representation of each feature sequence will be reused to refine attention weights of other feature sequences, and update their corresponding summarized representations. Therefore, we can perform Eqn. (6) for multiple rounds to facilitate the interaction among different information:

$$\tilde{\boldsymbol{x}}_i^r = \text{ATT}_{\boldsymbol{X}_i}(\{\tilde{\boldsymbol{x}}_c^r\}_{c<i}, \{\tilde{\boldsymbol{x}}_c^{r-1}\}_{c>i}), \tag{7}$$

where r indicates the round index. Ideally, we can obtain a better attention as the number of round grows, since the summarized representation $\tilde{\boldsymbol{x}}_i^r$ is incrementally refined after each round. However, it does not necessarily mean that the performance will be always improved with more rounds proceed. In other words, the performance may saturate once a sufficient interaction among multiple information sources is achieved.

To explore a more diversified interaction, we follow the idea of self-attention [43,53], and adopt the attended feature of $\boldsymbol{X}_i$ in all previous rounds, *i.e.*, $\{\tilde{\boldsymbol{x}}_i^t\}_{t=1}^{r-1}$, to guide the attention on $\boldsymbol{X}_i$ at current round:

$$\tilde{\boldsymbol{x}}_i^r = \text{ATT}_{\boldsymbol{X}_i}(\{\tilde{\boldsymbol{x}}_c^r\}_{c<i}, \{\tilde{\boldsymbol{x}}_i^t\}_{t=1}^{r-1}, \{\tilde{\boldsymbol{x}}_c^{r-1}\}_{c>i}). \tag{8}$$

In this way, for each $\boldsymbol{X}_i$, we can not only capture its relation with other feature sequences, but also discover the latent relationship among its own elements, which can provide a comprehensive modeling for all input information sources. Moreover, by utilizing $\{\tilde{\boldsymbol{x}}_i^t\}_{t=1}^{r-1}$ as the attention guidance for the r-th round, shortcut connections are constructed between the r-th round and all previous rounds, which facilitates the information propagation among the attention modules and ease the training of the whole model.

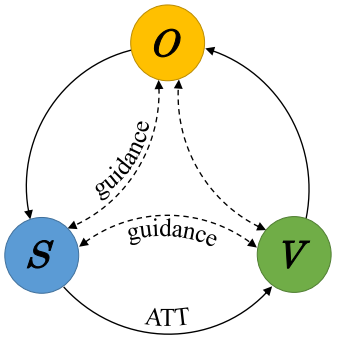


Fig. 4. The A-ATT mechanism for one-round VG. Bold lines denote the A-ATT process, and dash lines denote the attention guidance.

Similar to [14], the parameters $\{\boldsymbol{W}_c\}$, $\boldsymbol{W}_h$, and $\boldsymbol{b}$ can be shared across different rounds to avoid the over-fitting problem (see Sec. 4.5.4). However, this may also limit the representation ability of the model, since we have to use the same parameters to construct the correlations among the attention modules at different rounds. In the following sections, we will show how we prevent the model from over-fitting by adding noises into the training data and introducing regularization terms in the training objective. Thus, we can maintain a set of parameters for each round to improve the representation ability of our model.

3.3 Apply A-ATT on VG

Based on the proposed A-ATT mechanism, our VG model is built-in with three types of attention module to handle the feature sequences $\boldsymbol{S}$ (query), $\boldsymbol{V}$ (image) and $\boldsymbol{O}$ (object proposals) extracted in Sec. 3.1. Specifically, the attention module for $\boldsymbol{O}$ (denote as $\text{ATT}_{\boldsymbol{O}}$) is able to perform the matching step of VG, while the attention modules for $\boldsymbol{S}$ and $\boldsymbol{V}$ (denote as $\text{ATT}_{\boldsymbol{S}}$ and $\text{ATT}_{\boldsymbol{V}}$, respectively) help us obtain a better feature representation for the input image and query, which can ease the matching problem in $\text{ATT}_{\boldsymbol{O}}$.

According to Eqn. (8), we can obtain the attended features of $\boldsymbol{S}$, $\boldsymbol{V}$, $\boldsymbol{O}$ at r-th round through:

$$\begin{cases} \tilde{\boldsymbol{s}}^r = \text{ATT}_{\boldsymbol{S}}(\{\tilde{\boldsymbol{s}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{v}}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{v}}^r = \text{ATT}_{\boldsymbol{V}}(\tilde{\boldsymbol{s}}^r, \{\tilde{\boldsymbol{v}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\tilde{\boldsymbol{s}}^r, \tilde{\boldsymbol{v}}^r, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}), \end{cases} \tag{9}$$

where $\tilde{\boldsymbol{s}}^r$, $\tilde{\boldsymbol{v}}^r$ and $\tilde{\boldsymbol{o}}^r$ will then be passed to the next round of A-ATT (and keep flowing through the following rounds) to refine the attention weights of the corresponding feature sequences. During this circulation, the attention on the useful information in each feature sequence will be accumulated, while the attention on noises will fade out, leading to a improved summarized representation for each kind of information. We give a illustration of the A-ATT mechanism for VG in Fig. 4.

To ground the query into the image, at the last round of A-ATT, we directly adopt the attention module for object proposal $\text{ATT}_{\boldsymbol{O}}(\cdot)$ to perform the matching step. *I.e.*, matching the summarized representations of image and query ($\tilde{\boldsymbol{v}}^r$ and $\tilde{\boldsymbol{s}}^r$) as well as the summarized representations of object proposals at previous rounds ($\{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}$) with the representation of each object proposal $\boldsymbol{o}_i$. The best-matched proposal is selected as the prediction.

In practice, however, it is not necessary to perform A-ATT for too many rounds [14], since increasing the number of round may incur difficulties in model optimization due to the vanishing of training signal. Besides, we may also encounter a learning plateaus as the round of accumulation grows. Last, a large number of round will incur increased computation complexity.

3.4 Noised training strategy

As mentioned in Sec. 1, there exists a huge distribution gap between the training data and the real-world data in previous VG methods: their models are trained with human-labeled object proposals, which have accurate bounding boxes for every object in the image. However, in practice, we may only have the object proposals detected by off-the-shelf detectors such as Faster RCNN [7] and SSD [8], thus the bounding boxes can be noisy. More critically, the object detectors may even fail to detect the target object (*i.e.*, the IOU score between the detected bounding box and the target bounding box is less than 0.5), in which case the VG task will definitely fail. To tackle this problem, in this section, we propose a noised training strategy, which typically consists three stages, *i.e.*, bounding box augmentation, bounding box regression, and end-to-end fine-tuning.

3.4.1 Bounding box augmentation

First, we perform bounding box augmentation onto the bounding boxes of the human-labeled object proposals, which *i.e.*, randomly shifts, scales, and resizes these bounding boxes to a small extent to approximate the bounding boxes of detected proposals. Formally, we represent the original bounding box of an object proposal as a four-tuple $\boldsymbol{b} = (x, y, w, h)$, where (x, y) denote the top-left corner; w and h denote its width and height. Then, we also represent the noise as a four-tuple $\boldsymbol{\epsilon} = (\epsilon_x, \epsilon_y, \epsilon_w, \epsilon_h)$, where

$$\begin{aligned} \frac{\epsilon_x}{w} &\sim \mathcal{N}(0,\ \sigma_x), \quad \frac{\epsilon_y}{h} \sim \mathcal{N}(0,\ \sigma_y), \\ \frac{\epsilon_w}{w} &\sim \mathcal{N}(0,\ \sigma_w), \quad \frac{\epsilon_h}{h} \sim \mathcal{N}(0,\ \sigma_h). \end{aligned} \tag{10}$$

The noise level is jointly controlled by σ and the width or height of the corresponding bounding box. Afterward, the noised bounding box is obtained by $\boldsymbol{b}_\epsilon = (x+\epsilon_x, y+\epsilon_y, w+\epsilon_w, h+\epsilon_h)$. Moreover, we adopt the truncated version of Eqn. (10) to make sure the noised bounding box of each object proposal has the largest IOU (Intersection over Union) score with its original ground-truth box rather than the ground-truth boxes of other object proposals.

By adding noises into the training data, a VG model can learn to handle inaccurate bounding boxes during training. Therefore, the model can generalize better on testing data with detected object proposals. In Fig. 5, we show a comparison of the detected object proposal, the human-labeled object proposal and its noised counterpart.

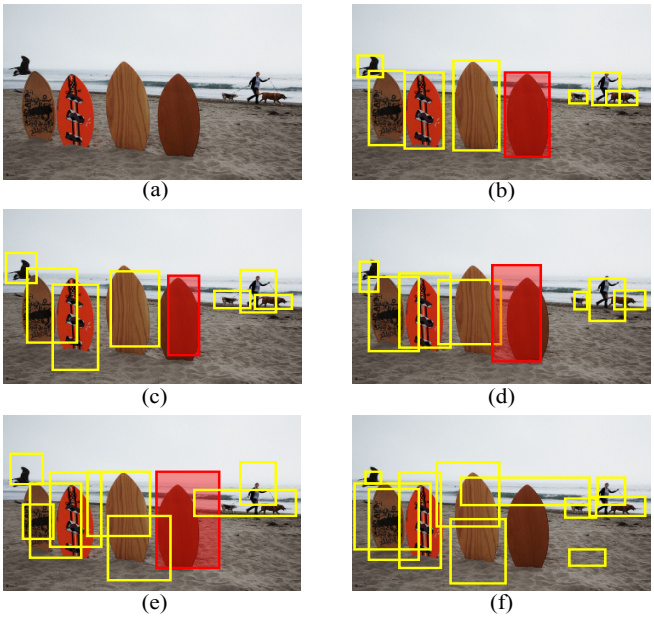


Fig. 5. Different kinds of proposals for a VG example. (a) the original image; (b) the human-labeled object proposals; (c) and (d) two set of noised human-labeled proposals with different random noises; (e) detected object proposals, the target are successfully detected (the red box); (f) detected object proposals, a "definitely fail" case for VG where the target objects are missed.

After we perform bounding box augmentation, we use the noised human-labeled proposals to train the A-ATT mechanism. Specifically, we minimize the cross-entropy loss between the model prediction $\hat{o}$ and the target o^*:

$$L_{cls} = -\frac{1}{m}\sum_{i=1}^{m} \log \alpha_{o^*}(i). \tag{11}$$

Here, m is the batch size, i is the index of the training sample. The attention weight α_{o^*} is calculated by Eqn. (4), and represents the predicted probability $P(\hat{o} = o^*)$.

In addition, we follow [11] and add a regularization term for each attention module, *i.e.*,

$$L_{reg}^x = \lambda_{\boldsymbol{x}} \sum_{i}^{n_x} (1 - \sum_{r}^{R} \alpha_i^r)^2, \tag{12}$$

where α_i^r denotes the attention weights computed for the i-th elements in the feature sequence $\boldsymbol{X}$ at the r-th round of A-ATT mechanism, n_x denotes the length of $\boldsymbol{X}$, and $\lambda_{\boldsymbol{x}}$ is a hyper-parameter. By introducing these regularization terms, we encourage the model to pay equal attention to every part of the information at different round so as to prevent the model from over-fitting. We train the A-ATT mechanism with the following loss function until it is converged:

$$L_{att} = L_{cls} + \sum_{x} L_{reg}^x + \frac{\lambda}{2}||\boldsymbol{W}||^2. \tag{13}$$

Here, the term $\frac{\lambda}{2}||\boldsymbol{W}||^2$ is a weight decay term, with λ being a weight decay parameter.

3.4.2 Bounding box regression

The noises introduced in the bounding box augmentation stage also enable us to learn a bounding box regressor with the human-labeled object proposals and their noised counterparts. Specifically, the bounding box regressor takes the feature $\boldsymbol{o}^*$ of a noised object proposal (see Sec 3.1.3) as input and aims to predict an offset tuple $\boldsymbol{\delta} = (\delta_x, \delta_y, \delta_w, \delta_h)$ so that we can restore the original bounding box of the proposal. Thanks to this bounding box regressor, when the VG model is evaluated on testing data with detected object proposals, the IOU between the bounding boxes of predicted and target object proposals can be significantly increased, and the "definitely fail" case can be effectively avoided.

Denote the ground-truth bounding box of the target object proposal o^* and its noised version as $\boldsymbol{b}^* = (x^*, y^*, w^*, h^*)$ and $\boldsymbol{b}^*_\epsilon = (x^*_\epsilon, y^*_\epsilon, w^*_\epsilon, h^*_\epsilon)$, respectively. Then, the regression loss is defined as:

$$L_{bbox} = \sum_{k \in \{x,y,w,h\}} \frac{1}{\eta} \cdot L_{\text{hu}}(k^*_\epsilon + \delta_k,\ k^*), \tag{14}$$

where $\eta = w$ for $k \in \{x, w\}$ and $\eta = h$ for $k \in \{y, h\}$. L_{hu} indicates the smooth$_{L_1}$ loss [58]:

$$L_{\text{hu}}(\hat{t}, t) = \begin{cases} \frac{1}{2}(\hat{t} - t)^2 & \text{for } |\hat{t} - t| \leq 1, \\ |\hat{t} - t| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{15}$$

In practice, the smooth$_{L1}$ loss is more robust than the commonly used L_2 loss, since it is less sensitive to outliers.

3.4.3 End-to-end fine-tuning

Lastly, like in [7], we can combine the Eqn. (13) and (14) together to fine-tune the whole model in an end-to-end manner to obtain a joint optimal solution:

$$L = L_{att} + \gamma L_{bbox}, \tag{16}$$

where γ is a hyper-parameter to balance L_{att} and L_{bbox}. Typically, for some training samples, our A-ATT model may make wrong predictions ($\hat{o} \neq o^*$), then the difference between $\hat{k} + \delta_k$ and k^* in Eqn. (14) can be very large, resulting in large gradient values that may damage the bounding box regressor. Therefore, we adopt gradient clipping for the bounding box regressor during end-to-end fine-tuning.

4 Experiments

In this section, we first evaluate the performance of the proposed A-ATT mechanism on human-labeled object proposals. Then, we evaluate the performance of our "**noised**" training strategy on detected object proposals. Lastly, we conduct several ablation studies on the components in A-ATT, and visualize the attention within our A-ATT model.

4.1 Datasets

We test our methods on four datasets: ReferCOCO, ReferCOCO+, ReferCOCOg and GuessWhat?!. In ReferCOCO and ReferCOCO+ [61], the average length of queries is around 3.6, indicating that their queries are mostly short phrases. The difference is that the queries in ReferCOCO+ are not supposed to contain any location words, such as "left", "front". In ReferCOCOg, the queries are normal sentences, which have an average length of 8.43. Moreover, in ReferCOCO(+), the average number of objects of the same type is about 3.9, whilst in ReferCOCOg, this number is limited to 1.6. GuessWhat?! [13] is collected by a two-player game, where the queries are all multi-round dialogues. The average number of question-answer pairs in each dialogue in GuessWhat?! 5.4. Both ReferCOCO and ReferCOCO+ contain nearly 20k images, 142k queries, and 50k target object proposals. ReferCOCOg has about 26.7k images, 85.5k queries, and 54.8k target object proposals. In GuessWhat?!, there are 66.5k images, 155.3k dialogues, and 134.1k targets.

Following [4], we split ReferCOCO(+) into 40,000 training, 5,000 validation, and 5,000 testing samples, where the testing set are further split into "TestA" and "TestB". More precisely, images containing multiple people are put into "TestA", while images containing multiple instances of all other categories are in "TestB". ReferCOCOg is split into 49,822 training and 5,000 validation samples. For GuessWhat?!, we follow [13] and split it into training, validation, and testing set by a fixed proportion of 70%, 15%, and 15%.

4.2 Implementation details

In our basic implementation, we use VGG-16 as our image feature extractor for fair comparisons, and pre-train it on the ImageNet [64] dataset. The input image is resized to have a short side of 448 pixels before fed into the VGG-16. We adopt the output of the last convolutional layer (C5) as the image feature, which has a spatial size of 14×14. To obtain the feature for an object proposal, we perform RoIAlign on the output of the fourth convolutional block (C4), and we set the number of bins $k^2 = 7 \times 7$. We use an LSTM to encode the query feature, where the word embeddings (300-D) are pre-trained with GloVe model on the LM-1B [65] corpus, and the hidden state dimension of the LSTM is 512. If a query has tokens that are not in the pre-trained word embeddings (*e.g.*, misspelled words), we initialize them as random vectors.

To determine the value of the hyper-parameter σ in Eqn. (10), we adopt a widely used object detector (SSD) to perform object detection on images in ReferCOCO dataset, and analyze the distribution of the normalized offsets $\frac{(x'-x)}{w}, \frac{(w'-w)}{w}, \frac{(y'-y)}{h}, \frac{(h'-h)}{h}$ between the bounding boxes of the ground-truth object proposals (x, y, w, h) and their best matched detected object proposals (x', y', w', h'). We find that setting $\sigma_x = 0.06$, $\sigma_y = 0.06$, $\sigma_w = 0.12$, $\sigma_h = 0.10$ for the object proposals can be good for our setting.

During training, we use Adam [66] with an initial step size of 0.001 for training the A-ATT mechanism, and use momentum SGD to optimize the bounding box regressor, where the step size is 0.01 and the momentum is 0.9. The hyper-parameter λ_x and λ in Eqn. (13) is set to 0.05 and 5e-4, respectively. The γ in Eqn. (16) is set to 0.5. We train the whole model for 60 epochs (30 epochs for the first stage, 15 for the second stage, and 15 for end-to-end fine-tuning), and we use 8 GPUs with 2 training sample on each GPU, resulting in a batch size of 16.

We implement 4 versions of the proposed A-ATT mechanism, namely, A-ATT-1, A-ATT-2, A-ATT-3, and A-ATT-4

TABLE 1
Comparisons on ReferCOCO, ReferCOCO+ and ReferCOCOg on human-labeled object proposals. The parameters at different rounds are shared in A-ATT-{1,2,3,4} and their noised versions. All comparing methods use VGG16 features.

Methods	ReferCOCO			ReferCOCO+			ReferCOCOg
	Val acc	TestA acc	TestB acc	Val acc	TestA acc	TestB acc	Val acc
visdif [4]	-	67.57	71.19	-	52.44	47.51	59.25
MMI [2]	-	71.72	71.09	-	58.42	51.23	62.14
[62]	-	74.14	71.46	-	59.87	54.35	63.39
Neg Bag [34]	76.90	75.60	78.00	-	-	-	68.40
speaker+listener+reinforcer+MMI [5]	79.56	78.95	80.22	62.26	64.60	59.62	72.63
Variational Context [38]	-	78.98	**82.39**	-	62.56	**62.90**	**73.98**
MattN [39]	**80.94**	**79.99**	82.30	**63.07**	**65.04**	61.77	73.08
A-ATT-1 [14]	79.19	79.67	78.16	64.45	66.51	58.84	72.33
A-ATT-2 [14]	80.68	81.37	79.79	65.35	68.36	60.19	72.67
A-ATT-3 [14]	80.98	81.67	79.96	65.50	67.92	60.69	72.94
A-ATT-4 [14]	81.27	81.17	80.01	65.56	68.76	60.63	73.18
A-ATT-1	78.97	79.71	78.24	64.51	66.77	58.85	72.19
A-ATT-2	81.03	81.56	80.72	65.93	68.67	61.04	73.74
A-ATT-3	**81.67**	**82.33**	81.32	**66.59**	**68.84**	**62.31**	**74.79**
A-ATT-4	81.45	82.00	**81.89**	65.82	68.47	62.03	74.61
A-ATT-1 (bbox augmentation)	79.02	79.54	78.26	64.63	66.77	58.72	72.15
A-ATT-2 (bbox augmentation)	81.14	81.62	80.99	66.27	68.86	61.67	73.95
A-ATT-3 (bbox augmentation)	**82.10**	**82.98**	81.78	67.08	**69.49**	62.01	**75.81**
A-ATT-4 (bbox augmentation)	81.95	82.85	**82.16**	**67.13**	69.12	**62.39**	75.50

TABLE 2
Comparisons on GuessWhat?! with human-labeled object proposals.

Model	Validation error	Testing error
HRED [13]	38.2	39.0
Parallel Attention [63]	36.2	36.6
A-ATT-1	36.7	37.9
A-ATT-2	34.3	34.6
A-ATT-3	33.2	33.9
A-ATT-4	**33.0**	**33.3**

w.r.t different rounds. Here, the first round, *i.e.*, A-ATT-1, is used for **warming-up**, since no attention guidance is available in the beginning.

4.3 Performance of the A-ATT mechanism

In this section, we evaluate the performance of the proposed A-ATT mechanism. We train and evaluate our model on data with human-labeled object proposals. Moreover, in this setting, we share the parameters among different rounds by default to avoid the over-fitting problem.

4.3.1 Results on human-labeled object proposals

Here, we present the results on ReferCOCO, ReferCOCO+, ReferCOCOg, and GuessWhat?! dataset with human-labeled object proposals. As shown in Table 1, A-ATT-3 achieves the best results on the Val split of ReferCOCO (81.67 vs. 80.94), ReferCOCO+ (66.59 vs. 63.07) and ReferCOCOg, as well as the best results on the TestA split of ReferCOCO (82.33 vs. 79.99) and ReferCOCO+ (68.84 vs. 65.04). Moreover, on these dataset splits, the performances of A-ATT-2 and A-ATT-4 are also very competitive, which outperform the previous best results on most of the splits by a considerable margin. On the TestB split of ReferCOCO and ReferCOCO+, A-ATT-3 and A-ATT-4 also perform comparably to the previous best results. On GuessWhat?!, the A-ATT mechanism surpasses the previous best results significantly, see Table 2. These empirical results verify the superiority of our proposed A-ATT mechanism.

Note that the warm-up phase, *i.e.*, A-ATT-1, is able to produce comparable or slightly improved results with the previous state-of-the-art methods on some dataset splits, indicating the strong ability of our A-ATT mechanism in joint reasoning among multi-model information. After the warm-up phase, the attention guidance generated by A-ATT-1 will be adopted to refine the attention weights in A-ATT-2. That is, the attention begins to accumulate. As a result, A-ATT-2 significantly improves the performance on top of A-ATT-1 (*e.g.*, +2.06% on ReferCOCO Val).

The attention accumulation process, however, may yield only slight improvement after three or four rounds of A-ATT process. For example, A-ATT-3 and A-ATT-4 may only lead to less than 1% gains in accuracy on the top of A-ATT-2. As we discussed in Section 3.3, this may because of that, for an easy setting of VG, like using human-labeled object proposals in the above VG datasets, two rounds are already sufficient for the proposed A-ATT mechanism to achieve good interactions among all types of information. As a result, A-ATT-2 performs competitively on all dataset under this setting, while further rounds (A-ATT-3 and A-ATT-4) only bring small gains, *i.e.*, the performance saturates. In the following sections, we will show that on a more challenging VG task, *i.e.*, using detected object proposals, attention accumulation still bring further gains.

We have two modifications on our CVPR Version [14]: 1) We exploit the self-attention guidance to improve the A-ATT mechanism; 2) We adopt bounding box augmentation to train our A-ATT mechanism. From Table 1, we see a clear performance improvement for these two modifications.

4.3.2 Results on detected object proposals

Here, we analyze the performance of the A-ATT mechanism on ReferCOCO, ReferCOCO+, and ReferCOCOg with de-

TABLE 3
Comparisons on ReferCOCO, ReferCOCO+ and ReferCOCOg with detected object proposals. The parameters at different rounds are shared in A-ATT-{1,2,3,4}, and unshared in all their noised training versions. All methods use VGG16 features and SSD-detected object proposals.

Methods	ReferCOCO (detected)		ReferCOCO+ (detected)		ReferCOCOg (detected)
	TestA acc	TestB acc	TestA acc	TestB acc	Val acc
[62]	67.94	55.18	57.05	43.33	49.07
speaker+listener+reinforcer [5]	72.34	63.24	**59.36**	48.72	58.70
speaker+listener+reinforcer+MMI [5]	72.88	63.43	60.43	48.74	59.51
Variational Context [38]	**73.33**	**67.44**	58.40	**53.18**	**62.30**
A-ATT-1	73.43	64.11	59.35	50.08	58.84
A-ATT-2	74.98	66.02	61.03	51.31	61.94
A-ATT-3	**76.45**	67.57	**62.17**	**52.71**	**62.33**
A-ATT-4	76.32	**67.93**	61.66	52.15	61.70
A-ATT-1 (noised training)	76.42	66.75	62.63	53.54	60.32
A-ATT-2 (noised training)	79.59	70.14	63.97	54.66	63.57
A-ATT-3 (noised training)	**80.87**	**71.55**	**65.13**	**55.01**	**63.84**
A-ATT-4 (noised training)	80.60	71.26	65.10	54.94	63.13

TABLE 4
Evaluate "noised" training strategy on previous methods. "origin" denotes the baseline models in the original paper; "ours" denotes applying our training strategy on the baseline models.

Methods	TestA acc	TestB acc
speaker+**listener** [5]	72.23	62.92
speaker+**listener** + noised training	75.07	65.36
speaker+**listener**+MMI [5]	72.95	62.43
speaker+**listener**+MMI+noised training	75.48	64.62
speaker+listener+MMI [5]	72.95	63.10
speaker+listener+MMI+noised training	75.32	65.51

tected object proposals. The results are recorded in Table 3, denoted by "A-ATT-1 " to "A-ATT-4". Similar to the observation in the last section, simply performing the A-ATT mechanism for one round is able to produce comparable results with previous best results on TestA split of ReferCOCO and ReferCOCO+. Adding another round of A-ATT (*i.e.*, A-ATT-2) leads to a big gain in accuracy (1.55% on ReferCOCO TestA split and 1.89% on TestB split). Moreover, A-ATT-3 further improves the performance significantly (1.47% on ReferCOCO TestA split and 1.55% on TestB split) on the top of A-ATT-2, yielding the new state-of-the-art results on almost all splits. These observations verify the effectiveness of the proposed A-ATT mechanism on the dataset with detected object proposals.

4.4 Performance of the "noised" training strategy

We then evaluate the effectiveness of the "noised" training strategy on ReferCOCO, ReferCOCO+, and ReferCOCOg. Different from the settings in Sec. 4.3, we do not share the parameters among different rounds for models trained with noised data in Table 3 so as to increase the representation power of the model. However, for the models marked with "noised training" in Table 1, the parameters in different rounds are shared for fair comparisons.

4.4.1 Results on detected object proposals

We use SSD-detected [8] object proposals provided by [5] for all comparisons. See Table 3, performing the noised training strategy significantly improves the performance of all four versions of the proposed A-ATT model on all dataset splits. Moreover, with noised training strategy, the proposed A-ATT mechanism achieves at least 1.5% and up to **7.5%** performance improvement over the previous best result. These observations clearly demonstrate the superiority of the proposed noised training strategy.

We further show how each training stage in our noised training strategy contributes to the performance improvement in Fig. 6. We observe that all three training stage contribute to the improved final performance. Besides, we can also observe the benefit brought by multi-round A-ATT, as the performance of A-ATT grows steadily and clearly when increasing the number of rounds from one to three.

4.4.2 Results on human-labeled object proposals

We also evaluate the noised training strategy on human-labeled object proposals. Note that the bounding box regressor is not needed, since we have already the ground-truth bounding boxes. So we only use bounding box augmentation in these cases. As shown in Table 1, the noised training strategy improves the performance of the A-ATT mechanism when inferred with human-labeled object proposals, since it augments the training data which benefits the generalization ability of the model. To be more specific, A-ATT-3 and A-ATT-4 with noised training strategy set up new state-of-the-art performance on the Val and TestA splits of all datasets (improve the accuracy by 1% to 4.5%). On TestB split, our methods also yield very close results to the current state-of-the-art.

4.4.3 Apply our training strategy in other VG methods

To show the effectiveness of our noised training strategy (*i.e.*, bounding box augmentation + bounding box regression + end-to-end fine-tuning), we apply it on some other VG baselines. Here, we consider three baselines from [5], namely, speaker+**listener**, speaker+**listener**+MMI, and **speaker**+listener+MMI. As in [5], the module being used for grounding the query are highlighted in **bold**. The results are recorded in Table 4. From the table, we find that the proposed noised training strategy boosts the performance for all baselines. This suggests that the proposed training strategy can be adopted as the basic configuration for general VG methods, since it is very beneficial for the model

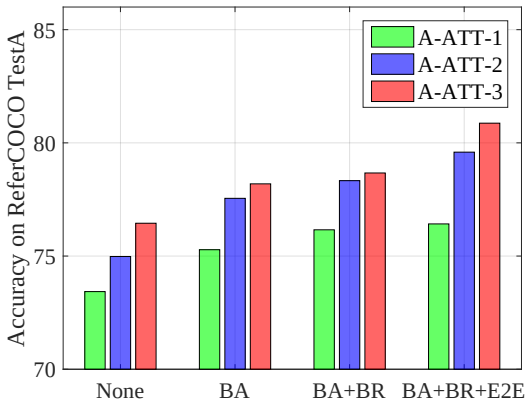


Fig. 6. Effect of different training stages on the performance of three A-ATT models, including A-ATT-1, A-ATT-2, and A-ATT-3. The training procedure consists of three stages, namely, "bbox augmentation" (BA), "bbox regression" (BR), and "e2e finetuning" (E2E). "None" denotes models trained without noised training strategy.

performance while only brings tiny extra computation, and is easy to use.

4.5 Ablation studies

In this section, we conduct extensive ablation studies on each component in our VG model, including the attention modules, the connections between attention modules, the weight sharing strategy in attention modules, as well as the feature extractors for each kind of input information. Without specification, we conduct experiments based on A-ATT-3 model without noised training strategy.

4.5.1 How do the connections between each attention module affect the A-ATT mechanism?

We break the connections between attention modules to analyze their importance. Specifically, we consider the following options:

1) A-ATT(w/o VO). Break the connections between the object attention module and the image attention module, *i.e.*, we remove the summarized representation of image feature sequence from the object attention module and vice versa:

$$\begin{cases} \tilde{\boldsymbol{s}}^r = \text{ATT}_{\boldsymbol{S}}(\{\tilde{\boldsymbol{s}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{v}}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{v}}^r = \text{ATT}_{\boldsymbol{V}}(\tilde{\boldsymbol{s}}^r, \{\tilde{\boldsymbol{v}}^t\}_{t=1}^{r-1}) \\ \tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\tilde{\boldsymbol{s}}^r, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}). \end{cases}$$

2) A-ATT(w/o SO). Similarly, we break the connections between the object attention module and the query attention module:

$$\begin{cases} \tilde{\boldsymbol{s}}^r = \text{ATT}_{\boldsymbol{S}}(\{\tilde{\boldsymbol{s}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{v}}^{r-1}) \\ \tilde{\boldsymbol{v}}^r = \text{ATT}_{\boldsymbol{V}}(\tilde{\boldsymbol{s}}^r, \{\tilde{\boldsymbol{v}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\tilde{\boldsymbol{v}}^r, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}). \end{cases}$$

3) A-ATT(w/o SV). Break the connections between the image attention module and the query attention module:

$$\begin{cases} \tilde{\boldsymbol{s}}^r = \text{ATT}_{\boldsymbol{S}}(\{\tilde{\boldsymbol{s}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{v}}^r = \text{ATT}_{\boldsymbol{V}}(\{\tilde{\boldsymbol{v}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\tilde{\boldsymbol{s}}^r, \tilde{\boldsymbol{v}}^r, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}). \end{cases}$$

4) A-ATT(w/o Self). Remove all self-attention connections, *i.e.*, the original A-ATT mechanism in our conference paper [14]:

$$\begin{cases} \tilde{\boldsymbol{s}}_i^r = \text{ATT}_{\boldsymbol{S}}(\tilde{\boldsymbol{v}}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{v}}_i^r = \text{ATT}_{\boldsymbol{V}}(\tilde{\boldsymbol{s}}^r, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{o}}_i^r = \text{ATT}_{\boldsymbol{O}}(\tilde{\boldsymbol{s}}^r, \tilde{\boldsymbol{v}}^r). \end{cases}$$

The results are shown in Fig. 7. From the figure, we find that building connections between any two types of attention modules within the A-ATT mechanism are beneficial for solving the VG task. Moreover, by breaking the self-attention connections for each attention module, the model performs even worse, which verifies the effectiveness of the self-attention mechanism for VG tasks.

4.5.2 How do the image & query attention modules affect the A-ATT mechanism?

When removing the attention modules on image or query, we need to obtain the summarized feature representation directly from the input image or query like in many previous works [1,2,4,6], instead of first extracting a feature sequence from them and then summarizing the sequence. Following these works, we obtain the image feature from the fc7 of VGG-16 (4096-D), and adopt a linear layer to transform it into a 512-D vector, denote as $\boldsymbol{v}_{cnn}$. For the query feature, we use the hidden state in the last time step of the LSTM, $\boldsymbol{s}_{rnn}$. Particularly, we consider the following options:

1) A-ATT(w/o Query). Remove the query attention module:

$$\begin{cases} \tilde{\boldsymbol{v}}^r = \text{ATT}_{\boldsymbol{V}}(\boldsymbol{s}_{rnn}, \{\tilde{\boldsymbol{v}}^t\}_{t=1}^{r-1}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\boldsymbol{s}_{rnn}, \tilde{\boldsymbol{v}}^r, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}). \end{cases}$$

2) A-ATT(w/o Image). Remove the image attention module:

$$\begin{cases} \tilde{\boldsymbol{s}}^r = \text{ATT}_{\boldsymbol{S}}(\{\tilde{\boldsymbol{s}}^t\}_{t=1}^{r-1}, \boldsymbol{v}_{cnn}, \tilde{\boldsymbol{o}}^{r-1}) \\ \tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\tilde{\boldsymbol{s}}^r, \boldsymbol{v}_{cnn}, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}). \end{cases}$$

3) Simple Matching. Remove both the query and image attention modules, results in a simple matching model:

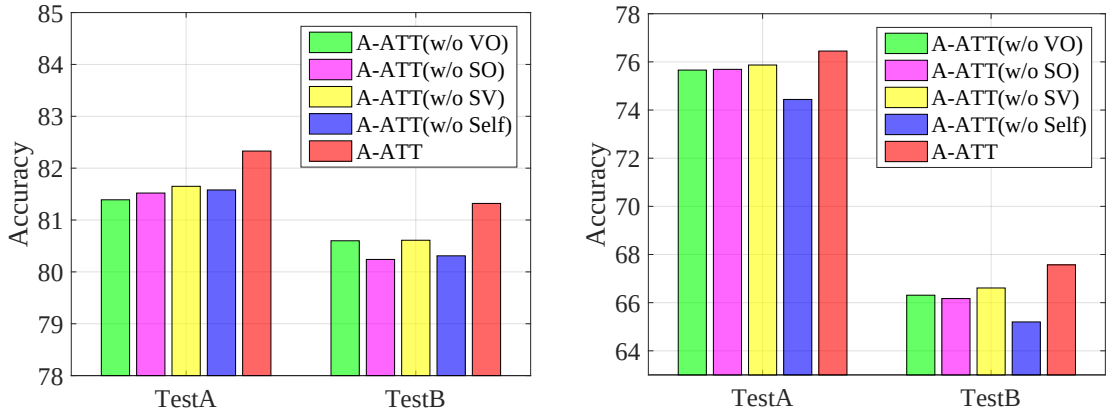
$$\tilde{\boldsymbol{o}}^r = \text{ATT}_{\boldsymbol{O}}(\boldsymbol{s}_{rnn}, \boldsymbol{v}_{cnn}, \{\tilde{\boldsymbol{o}}^t\}_{t=1}^{r-1}).$$

The results are shown in Fig. 8. From the figure, we find that each attention module plays an indispensable role in our A-ATT mechanism, showing the effectiveness of the attention module in improving the feature representation of the input image and query.

4.5.3 How do the feature extractors affect the model performance?

In our previous experiments, we use ImageNet pre-trained VGG-16 to extract the image feature, and use an LSTM to extract the query feature. In this section, we further explore some alternate feature extractors for our VG model. More precisely, we replace the VGG-16 model with ResNet-101 as the image feature extractor, and pre-train it on MS COCO [67] object detection task. We also replace LSTM with bidirectional LSTM as the query feature extractor.

We test these alternatives on ReferCOCO with detected and human-labeled object proposals, based on A-ATT-3 with noised training strategy, and show the results in Table 5. From the table, we observe that employing a better feature extractor for image or query can indeed boost the



(a) Performance comparison on ReferCOCO (b) Performance comparison on ReferCOCO (detected)

Fig. 7. Effect of different connections between different modules on the proposed A-ATT method. We compare the performance of the A-ATT-3 models with different settings on ReferCOCO and ReferCOCO (detected) datasets.

TABLE 5
The empirical results of different feature extractors. Note that R-101 and R-152 denote ResNet-101 and ResNet-152, respectively. '†' denotes **pre-trained** on ImageNet image recognition task, '‡' denotes **pre-trained** on MS COCO object detection task.

Methods	feature extractor			ReferCOCO			ReferCOCO (detected)	
	Image	Query	Object	Val acc	TestA acc	TestB acc	TestA acc	TestB acc
MattN [39]	R-101‡	bi-LSTM	RoIAlign	**85.65**	85.26	**84.57**	**81.14**	**69.99**
Multi-hop FiLM [40]	R-152†	bi-GRU	R-152†	84.90	**87.40**	83.10	-	-
A-ATT-3 (noised training)	VGG-16†	LSTM	RoIAlign	82.10	82.98	81.78	80.87	71.55
A-ATT-3 (noised training)	R-101‡	LSTM	RoIAlign	84.34	85.12	83.97	82.65	73.29
A-ATT-3 (noised training)	VGG-16†	bi-LSTM	RoIAlign	82.72	83.44	82.54	81.12	71.59
A-ATT-3 (noised training)	R-101‡	bi-LSTM	RoIAlign	**85.39**	**85.56**	**84.51**	**82.97**	73.48
A-ATT-3 (noised training)	R-152†	bi-LSTM	RoIAlign	84.97	84.69	84.12	82.24	**73.75**

TABLE 6
The impact of the regularization term in Eqn. (12) and the parameter sharing strategy among different rounds of A-ATT. The **bold** numbers are related to the original settings and are obtained from Table 3.

settings	ReferCOCO (detected)		ReferCOCO+ (detected)	
	TestA acc	TestB acc	TestA acc	TestB acc
A-ATT-3				
shared	**76.45**	**67.57**	**61.87**	**52.51**
unshared	68.28	60.14	57.07	47.10
w/o L_{reg}^{x}	75.39	65.74	60.95	51.30
A-ATT-3 with noised training strategy				
shared	79.63	70.22	64.26	54.58
unshared	**80.87**	**71.55**	**65.13**	**55.01**
w/o L_{reg}^{x}	80.15	71.12	64.06	53.69

performance of the A-ATT-3 model. By using the same feature extractors, our A-ATT-3 model trained with noised training strategy achieves comparable results with the previous state-of-the-art model MattN [39] on ReferCOCO with human-labeled object proposals. Moreover, when using detected object proposals for evaluation, our methods significantly outperform MattN (+1.83% on TestA, +3.49% on TestB), indicating the superiority of the proposed methods in dealing with inaccurate object proposals.

4.5.4 How do the regularization term and parameter sharing strategy alleviate the over-fitting problem?

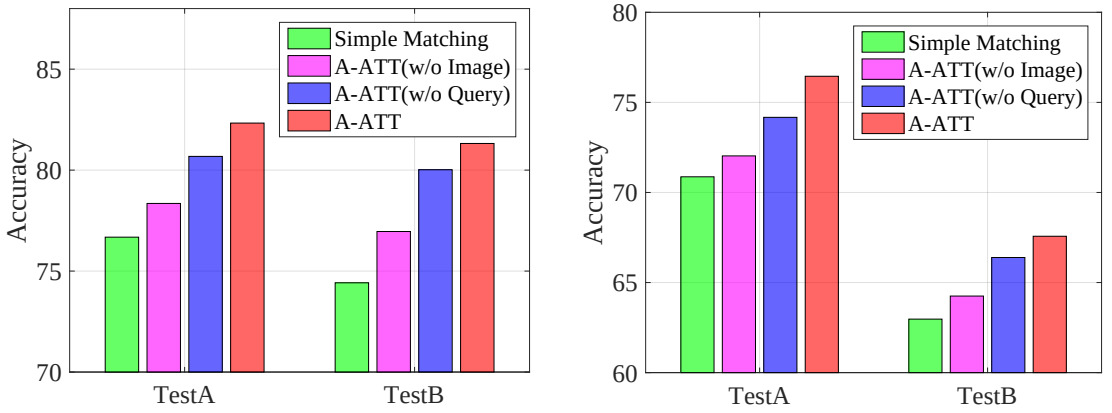
In this section, we analyse the impact of the regularization term in Eqn. (12) and the parameter sharing strategy among different rounds of A-ATT. We evaluate A-ATT-3 trained with/without noised training strategy on ReferCOCO(+) with detected object proposals.

As shown in Table 6, after removing the regularization term in the training objective, the performance of A-ATT-3 models consistently drop by a notable margin, indicating that the regularizer used in our training objective has a positive effect the model performance.

Apart from this, we also find that A-ATT models trained without noised training strategy can be very sensitive to the parameter sharing strategy, *e.g.*, the performance of A-ATT-3 declines severely on both ReferCOCO and ReferCOCO+ when each round of A-ATT keeps its own parameters. However, after switching to noised training, we observe a significant performance improvement (about %10) for the same model setting. These observations show that our noised training strategy is able to effectively address data distribution gap issue for the proposed A-ATT mechanism on VG tasks.

4.6 Visualization of the attention

We further visualize the attention weights on query and image for ReferCOCO, ReferCOCO+, and ReferCOCOg, as shown in Figure 9. We observe that the attention for different types of information tend to focus on the items that are correlated semantically or spatially. For example, on the ReferCOCO dataset, in the first column of the visualization results, the most focused word in the query is "person",



(a) Performance comparison on ReferCOCO. (b) Performance comparison on ReferCOCO (detected).

Fig. 8. Effect of the query attention module and the image attention module on the proposed A-ATT method. We compare the performance of the A-ATT-3 models with different settings on ReferCOCO and ReferCOCO (detected) datasets.

Fig. 9. Visualization of attention of our A-ATT-3 model on ReferCOCO, ReferCOCO+, and ReferCOCOg. In the image, the brighter regions correspond to a larger attention weights; in the query, we mark the words with attention weighs larger than 0.3 or 0.1 with yellow blocks or blue blocks, respectively. The target object for each image-query pair is outlined by a red box.

while in the image the relevant regions are assigned with larger attention weights. This means that different kinds of information can provide useful guidance for each other through the proposed A-ATT mechanism.

Moreover, to illustrate the effect of the attention accumulating, we visualize the attention on ReferCOCOg at different rounds of the A-ATT mechanism. The results are shown in Figure 10. Obviously, from the first round (A-ATT-1) to the fourth round (A-ATT-4), the attention weights on both image and the query tend to concentrate on more relevant parts of the information (*i.e.*, the target regions in the image, and keywords in the query).

5 Conclusion

In this paper, we have proposed a novel accumulated attention (A-ATT) mechanism to ground the natural language query into the image. Our model utilizes three kinds of information, *i.e.*, query, image and objects proposals, and provide an attention module to handle the attention problem for each information. Moreover, the A-ATT mechanism builds rich connections among the attention modules for knowledge transferring and accumulation. In this way, the noises and redundancy will decrease gradually, leading to an improved performance. On top of A-ATT mechanism, we further propose to use a "noised" training strategy to boost the performance of our VG models. Our model is able to deal with various types of queries, ranging from short

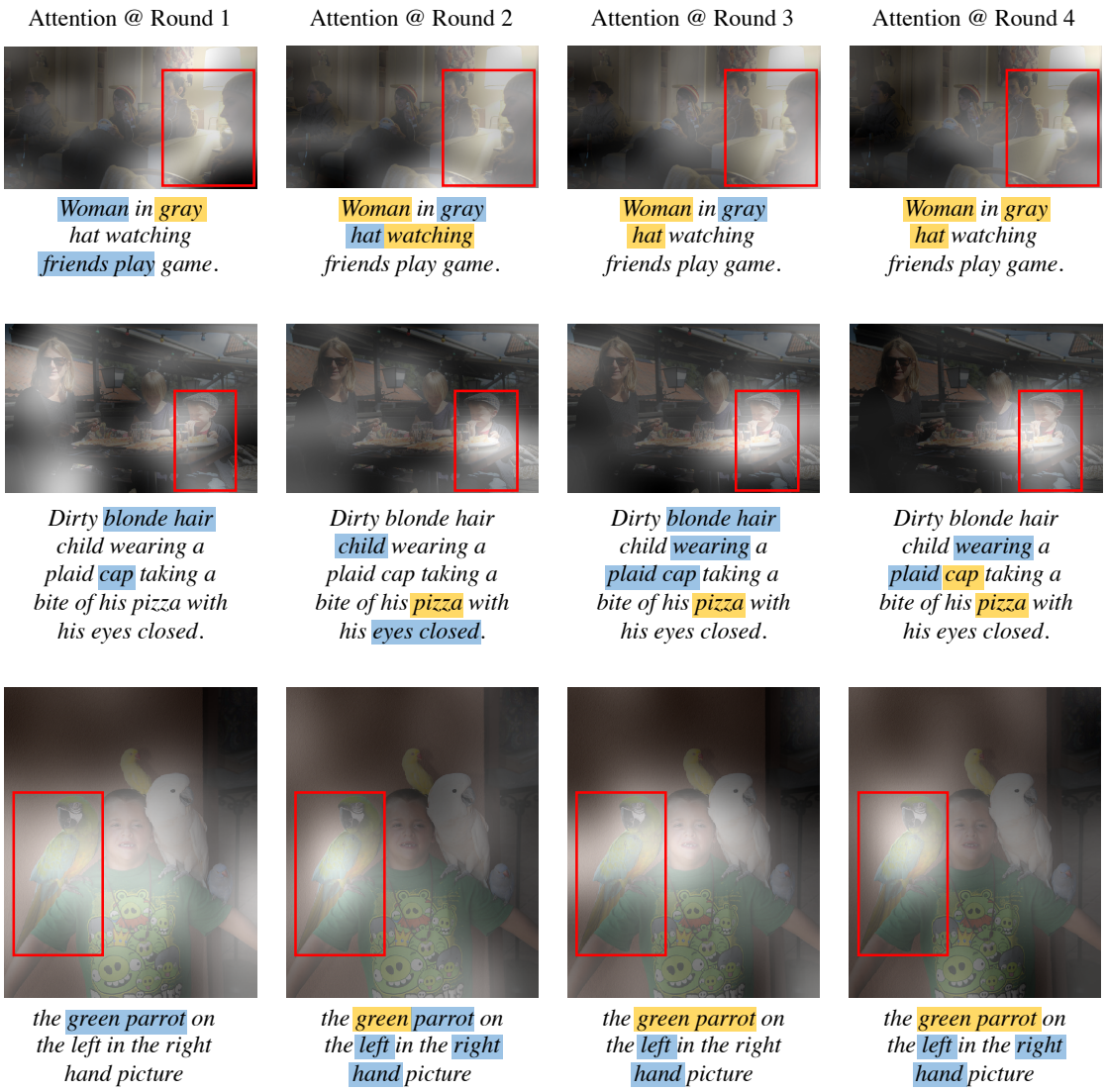


Fig. 10. Evolution of attention accumulating. We visualize the attention for A-ATT-{1,2,3,4}.

phrases to long dialogues. We evaluate the effectiveness of the proposed method on four popular datasets. Extensive experiments demonstrate the superior performance of the proposed methods over existing methods.

REFERENCES

[1] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4555–4564.

[2] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 11–20.

[3] F. Xiao, L. Sigal, and Y. J. Lee, "Weakly-supervised visual grounding of phrases with linguistic structures," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 5253–5262.

[4] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," 2016, pp. 69–85.

[5] L. Yu, H. Tan, M. Bansal, and T. L. Berg, "A joint speaker-listener-reinforcer model for referring expressions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 3521–3529.

[6] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *empirical methods in natural language processing*, pp. 457–468, 2016.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *International Conference on Neural Information Processing Systems*, 2015, pp. 91–99.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[9] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[11] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image

caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[12] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.

[13] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. C. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 4466–4475.

[14] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan, "Visual grounding via accumulated attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7746–7755.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[18] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *arXiv preprint arXiv:1412.6632*, 2014.

[19] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[20] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[21] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? dataset and methods for multilingual image question," in *Advances in neural information processing systems*, 2015, pp. 2296–2304.

[22] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 451–466.

[23] P. Wang, Q. Wu, C. Shen, and A. van den Hengel, "The vqa-machine: Learning how to use existing vision algorithms to answer new questions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 3909–3918.

[24] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[26] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[27] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[28] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to compose neural networks for question answering," *arXiv preprint arXiv:1601.01705*, 2016.

[29] ——, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 39–48.

[30] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," *CoRR, abs/1704.05526*, vol. 3, 2017.

[31] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International Conference on Machine Learning*, 2016, pp. 1378–1387.

[32] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International conference on machine learning*, 2016, pp. 2397–2406.

[33] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel, "Fvqa: Fact-based visual question answering," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[34] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Modeling context between objects for referring expression understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 792–807.

[35] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 817–834.

[36] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1960–1968.

[37] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, "Language-conditioned graph networks for relational reasoning," *arXiv preprint arXiv:1905.04405*, 2019.

[38] H. Zhang, Y. Niu, and S.-F. Chang, "Grounding referring expressions in images by variational context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4158–4166.

[39] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "Mattnet: Modular attention network for referring expression comprehension," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1307–1315.

[40] F. Strub, M. Seurin, E. Perez, H. De Vries, J. Mary, P. Preux, and A. CourvilleOlivier Pietquin, "Visual reasoning with multi-hop feature modulation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 784–800.

[41] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[42] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[45] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[46] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

[47] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Aˆ 2-nets: Double attention networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 352–361.

[48] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[49] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, "Attention-over-attention neural networks for reading comprehension," *arXiv preprint arXiv:1607.04423*, 2016.

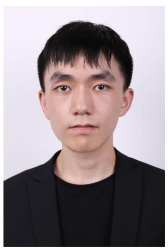
[50] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[51] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[52] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," *arXiv preprint arXiv:1803.03067*, 2018.

[53] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[54] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR, abs/1703.06211*, vol. 1, no. 2, p. 3, 2017.
[55] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.
[56] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3286–3293.
[57] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.
[58] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
[59] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, pp. 1610–02 357, 2017.
[60] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.
[61] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, "Referitgame: Referring to objects in photographs of natural scenes." in *EMNLP*, 2014, pp. 787–798.
[62] R. Luo and G. Shakhnarovich, "Comprehension-guided referring expressions," *Conference on Computer Vision and Pattern Recognition*, pp. 3125–3134, 2017.
[63] B. Zhuang, Q. Wu, C. Shen, I. Reid, and A. van den Hengel, "Parallel attention: A unified framework for visual object discovery through dialogs and queries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4252–4261.
[64] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[65] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *arXiv preprint arXiv:1312.3005*, 2013.
[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[67] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.



Chaorui Deng is a Master student in School of Software Engineering, South China University of Technology, Guangzhou, China. His research interests include deep learning, reinforcement learning.

Dr. Qi Wu is a Lecturer (Assistant Professor) in the University of Adelaide and he is an Associate Investigator in the Australia Centre for Robotic Vision (ACRV). He is the ARC Discovery Early Career Researcher Award (DECRA) Fellow between 2019-2021. Prior to joining the ACRV, Dr.Wu worked as a Senior Research Associate at the Australia Centre for Visual Technology. He obtained his PhD degree in 2015 and MSc degree in 2011, in Computer Science from the University of Bath, United Kingdom. Dr. Wu's research interests are mainly in Computer Vision and Machine Learning. His previous research projects include modeling visual objects regardless of depictive styles and image understanding using contextual cues. He is currently working on the Vision and Language problems, incuding Image Captioning, Visual Question Answering, Visual Dialog, etc.

Qingyao Wu received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore, from 2014 to 2015. He is currently an Associate Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include machine learning, data mining, big data research.

Fuyuan Hu was a postdoctoral researcher at Vrije Universiteit Brussel,Belgium, a Ph.D. student at Northwestern Polytechnical University, and a visiting Ph.D. student at the City University of Hong Kong. He is a professor in computer vision and machine learning at Suzhou University of Science and Technology. His research interests include graphical models, structured learning, and tracking.

Fan Lyu is a PhD student in College of Intelligence and Computing, Tianjin University. He received the MS degree in Electronic & Information Engineering, Suzhou University of Science and Technology, China. His research interests include deep learning, multi-modal learning.

Mingkui Tan received his Bachelor Degree in Environmental Science and Engineering in 2006 and Master degree in Control Science and Engineering in 2009, both from Hunan University in Changsha, China. He received the PhD degree in Computer Science from Nanyang Technological University, Singapore, in 2014. From 2014-2016, he worked as a Senior Research Associate on computer vision in the School of Computer Science, University of Adelaide, Australia. Since 2016, he has been with the School of Software Engineering, South China University of Technology, China, where he is currently a Professor. His research interests include machine learning, sparse analysis, deep learning and large-scale optimization.