Journal of Computers Vol. 35 No. 1, February 2024, pp. 83-95
doi: 10.53106/199115992024023501006

End-to-end Visual Grounding Based on Query Text Guidance and Multi-stage Reasoning

Chao Wang[1,2], Wei Luo[1,2], Jia-Rui Zhu[1,2], Ying-Chun Xia[1,2], Jin He[1,2], Li-Chuan Gu[1,2*]

[1] School of Information and Computer, Anhui Agricultural University, Hefei 230036, China
{wangchao_icle, glc}@ahau.edu.cn

[2] Key Laboratory of agricultural electronic commerce of the Ministry of Agriculture, Hefei, China

Received 1 January 2023; Revised 28 January 2023; Accepted 30 January 2023

Abstract: Visual grounding locates target objects or areas in the image based on natural language expression. Most current methods extract visual features and text embeddings independently, and then carry out complex fusion reasoning to locate target objects mentioned in the query text. However, such independently extracted visual features often contain many features that are irrelevant to the query text or misleading, thus affecting the subsequent multimodal fusion module, and deteriorating target localization. This study introduces a combined network model based on the transformer architecture, which realizes more accurate visual grounding by using query text to guide visual feature generation and multi-stage fusion reasoning. Specifically, the visual feature generation module reduces the interferences of irrelevant features and generates visual features related to query text through the guidance of query text features. The multi-stage fused reasoning module uses the relevant visual features obtained by the visual feature generation module and the query text embeddings for multi-stage interactive reasoning, further infers the correlation between the target image and the query text, so as to achieve the accurate localization of the object described by the query text. The effectiveness of the proposed model is experimentally verified on five public datasets and the model outperforms state-of-the-art methods. It achieves an improvement of 1.04%, 2.23%, 1.00% and +2.51% over the previous state-of-the-art methods in terms of the top-1 accuracy on TestA and TestB of the RefCOCO and RefCOCO+ datasets, respectively.

Keywords: visual grounding, query text guidance, Swin-transformer, attention module, multi-stage reasoning

1 Introduction

A deeper insight into unimodal information (e.g., text and images) and new possibilities of maching learning inspired recent studies on multi-modal tasks, including image captioning [1], cross-model retrieval [2], and visual question answering [3, 4]. In multi-modal tasks, learning the correspondence between text and images is vital. Visual grounding (also denoted as referring to expression comprehension [5, 6] or phrase localization [7, 8]) aims to locate the target object or area in the image according to natural language expression. Therefore, visual grounding is beneficial for accurately implementing other multi-modal tasks (e.g., image captioning and visual question answering).

Methods based on one-stage [9-11] and two-stage [5, 6, 12] model architectures are a common class of visual grounding methods, which transform the visual grounding task into the problem of ranking the detected candidate objects or areas. Methods based on the one-stage model, such as SSG [9] and FAOA [13], use pretrained fully convolutional networks to directly extract pixel-level visual features, fuse the extracted features with query text embedding to generate dense detections, and then select the detection target with the highest confidence score. These methods are effective in learning and reasoning about simple relationships between modalities but do not perform well for complex queries of various objects and relationships in images and text [10].

And methods based on the two-stage model, such as MAttNet [12] and DGA [14], use pretrained target detectors (e.g., Faster-RCNN [15]) to obtain a set of sparse region proposals, compute their similarity with query text features and then obtain the regional proposal that best matches the query text by ranking the similarity. Compared to the methods based on the one-stage model, the methods based on the two-stage model introduce a more complex multi-modal fusion and reasoning mechanism and thus perform better in visual grounding tasks [16, 17]. Nevertheless, the performance of target detectors and the quality of region proposals significantly im-

* Corresponding Author

pact the multi-modal reasoning performance in fused modules and limit the consideration of visual contextual information in the methods based on the two-stage model [18, 19].

Visual grounding methods based on a transformer model [18-21] perform multi-modal reasoning via pixel-level feature mapping and modeling of global visual information. By relying on the attention mechanism in the transformer architecture, intra- and intermodal interactions are achieved. Thus, the query objects can be localized with a more concise architecture and direct coordinate regression [19].

The methods based on the transformer model [19] use mutually independent visual and text encoders to extract their respective features, directly input these visual features and query text features into the coding layer architecture of the transformer, and then use the internal attention mechanism for encoding to achieve cross-modal fusion and direct location of the target object, as shown in Fig. 1. The pretrained visual encoder encodes only the information within the image, and the extracted features contain visual features irrelevant to the query text. These features may be redundant or even misleading, and transmitting them to the subsequent multi-modal fusion module may cause unreasonable reasoning. Meanwhile, the multi-modal fusion module directly adopts the transformer encoding architecture without considering the deeper interaction and reasoning between the query text and the visual objects, which would also affect the model's overall performance.

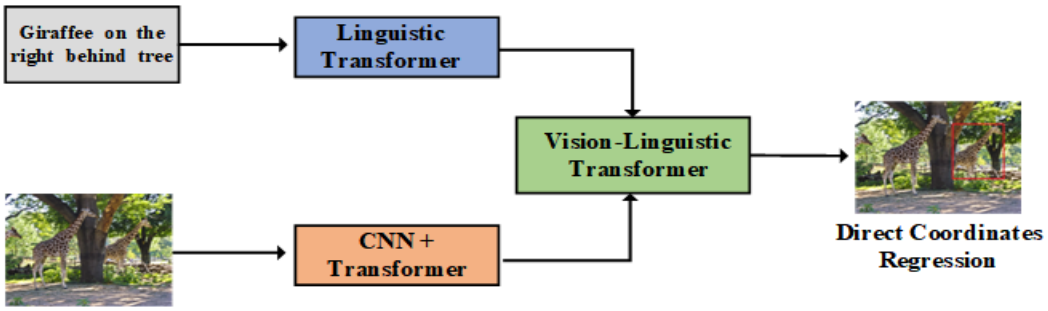

Fig. 1. Visual grounding with a transformer-based architecture

In this study, an end-to-end Visual Grounding model based on the guidance of Query text and Multi-stage reasoning, denoted as QMVG, was proposed for transformer-based architecture. As shown in Fig. 2, the contextual features of query text were obtained from the linguistic module and then introduced into the visual module to guide the generation of visual features closely related to the query text and suppress and reduce the generation of the visual features that are irrelevant to the query text or misleading; then, in the multi-stage reasoning module, multi-stage interactive reasoning was conducted for the visual features and the query text features to obtain an accurate representation of the query object gradually, thus achieving precise localization.

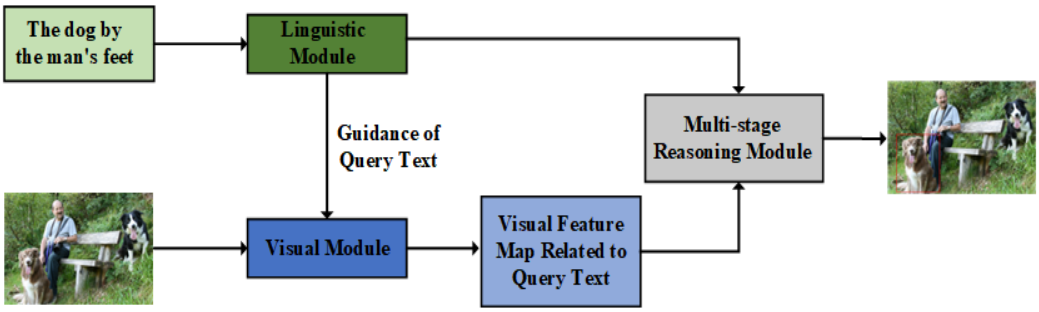

Fig. 2. End-to-end visual grounding framework based on the guidance of query text and multi-stage reasoning

The main contributions of this study can be summarized as follows:

(1) An end-to-end visual grounding model for transformer-based architecture was proposed. Through the

guidance of query text, the visual encoding was focused on the feature areas related to query text, and query text embedding was combined for multi-stage interactive reasoning, thus achieving accurate localization of query targets.

(2) Multiple sets of experiments were designed to verify the proposed model's performance, the proposed model's operation mechanism was analyzed in detail, and the model's good performance was verified on five public datasets.

2 Related Work

As a fundamental task in multi-modal learning tasks, visual grounding is used to locate relevant object instances in an image by the natural language expression of the described object. Most tasks can benefit from good localization between linguistic descriptions and visual objects. Therefore, visual grounding can provide reliable and effective support for multi-modal learning tasks such as visual question answering [4, 22] and visual language navigation [23]. The existing visual grounding methods can be broadly classified into one-stage, two-stage, and transformer-based methods [19-21].

One-stage methods extract visual features from images directly by a feature extractor, perform a complex transmodal fusion of query text embeddings and visual features, and then use the fused features for bounding box prediction. Two-stage methods split the visual grounding task into two stages: the stage of generating a set of candidate object proposals and the stage of sorting the proposals. Transformer-based methods achieve intramodal and intermodal interactions by relying on the attention mechanism in the transformer architecture, and visual grounding tasks are implemented in an end-to-end form. Specifically, Table 1 summarizes the work related to the three types of visual grounding methods.

However, one-stage methods are highly efficient but have inflexible models, and they cannot associate detailed descriptions in the query text and may ignore local information in images. Two-stage methods rely heavily on the performance of pretrained target detectors and only consider objects in predefined categories, making them unable to take full advantage of the contextual information in the scene. Given the excellent performance of transformers in visual grounding tasks [21], this study adopted Swin-transformer [29] as the visual feature extraction backbone network and used the hierarchical structure and moving window of the Swin-transformer to obtain different scales of features and global information of the images for modeling visual features. Based on the work of [20] and [21], an end-to-end visual grounding model was designed based on query text guidance and multi-stage reasoning.

Table 1. Related work of the visual grounding

Method types	References	Descriptions
One-stage methods	Yang et al. [13]	It exploited encoding the query text to obtain the text embeddings and further fusing the obtained text embeddings into the YOLOv3 [24] target detector and enhancing them with spatial features to achieve rapid and accurate localization of query objects.
	Yang et al. [10]	It designed a recursive subquery framework to iteratively adjust the sentence embedding to solve the problem of complex query statements, while the embedding of each subquery still remained a single vector.
	Huang et al. [25]	It made use of the relative spatial relationship between the target object and landmarks and the background information of landmarks to achieve localization.
Two-stage methods	Yu et al. [12]	It constructed the similarity between modals in terms of fine granularity by introducing modular components of topics, locations, and relations related to the query text description.
	Hong et al. [26]	It decomposed query text sentences into semantic components in a recursive way to construct a binary semantic tree, and then performed visual reasoning along the tree structure in a bottom-up manner.
	Chen et al. [27]	It used query text features to guide the nonmaximum suppression of object proposals in the first stage to increase the recall of key objects, which solved the problem of mismatch between the proposals generated based on the detection confidence and the query text.

Transformer-based methods	Deng et al. [19]	It proposed a transformer-based end-to-end visual grounding framework TransVG. This method used the DETR [28] encoder to extract visual features, incorporating the extracted visual features and text features into the coding layer of a transformer for intermodal interaction, with a final direct output of the target location through an MLP layer.
	Yang et al. [20]	It used the feature extraction module in the TransVG [19] network to encode and iteratively decode the obtained visual features and text features through crossed multi-head attention blocks to achieve the localization of query objects.
	Ye at al. [21]	It proposed a query-aware dynamic attention mechanism called QRNet, including a query-aware multiscale fused module, which was incorporated into the transformer in the visual backbone network to solve the inconsistency problem between intermediate features and query text features in the visual backbone network.

3 Method

In this section, we first introduce the QMVG model architecture. Then, we describe the three modules of our model: linguistic module, visual module and multi-stage reasoning module. Finally, we introduce the location of query objects.

3.1 Model Architecture and Process

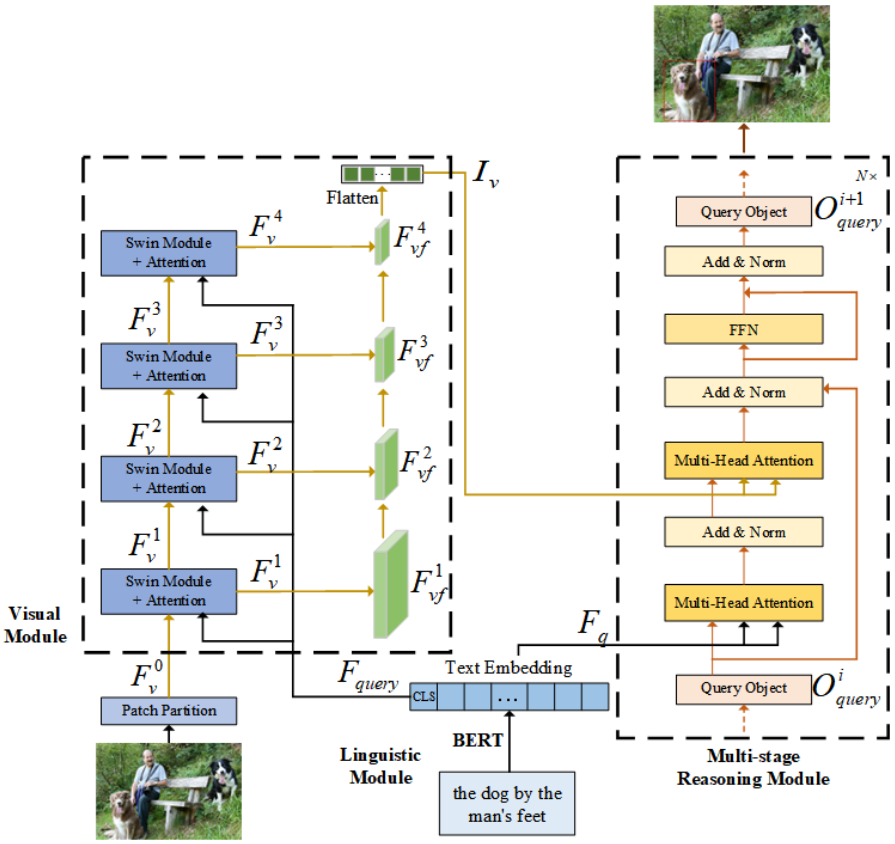

Fig. 3. The overall model framework of the QMVG

The QMVG model depicted in Fig. 3 mainly consisted of three modules: linguistic, visual, and multi-stage reasoning ones. The query text features are extracted by the linguistic module to guide the feature extraction of the visual module, so as to suppress the generation of the visual features that are irrelevant to the query text. Then the two modal features are interactively reasoned through the multi-stage reasoning module to achieve more accurate localization.

Specifically, the image and the query text are two inputs of the QMVG. The linguistic module encoded the query text to generate text embeddings. The visual module introduced the query text contextual information encoded by the linguistic module into each layer of the Swin-transformer architecture, guided the learning of visual features at different scales with the help of the attention mechanism, and aggregated the visual features at different scales to obtain the visual features related to the query text. Then, the query text features and visual features obtained from the first two modules were incorporated into the multi-stage reasoning module; thus, the accurate localization of the query objects could be gradually obtained.

3.2 Linguistic Module

For the query text, the BERT model [30] was used in the linguistic module to extract query text features. First, query text was tokened. Then, the tokened query text expressions were added with the [CLS] token at the beginning and the [SEP] token at the end. After that, the token query text was used as input to the text feature extractor and encoded to obtain the token of the query text contextual information $F_{query} \in R^{Cq\times 1}$ (contextual information was tokened by [CLS]) and the token of each word in the query text $F_q \in R^{Cq\times Nq}$ as the query text features, where the channel size C_q is 768 dimensions and N_q is the number of word tokens.

3.3 Visual Module

The image $I \in R^{H\times W\times 3}$ was given as the input of the visual module, where H and W represent the height and width of the image, respectively. QMVG used the query text guiding network to extract relevant visual features and flattened them into feature sequence $I_v \in R^{Cv\times Nv}$, where the channel dimension $C_v = 256$ and the number of input tokens $N_v = H \times W$. The visual module extracted visual features under the guidance of the query text features through the attention mechanism. It fused the visual features of different scales to obtain only those closely related to the query text.

Visual Feature Map. As the backbone network of visual module, the Swin-transformer outputted a hierarchical list of visual feature maps $[F_v^1, F_v^2, F_v^3, F_v^4]$. Each stage in the QMVG was composed of multiple Swin-transformer blocks (i.e., a Swin module) and an attention module, and the visual feature map of each stage was extracted as shown in the visual module in Fig. 3. Through the patch partition operation,the image I was embedded into $F_v^0 \in R^{\frac{H}{4}\times\frac{W}{4}\times C}$, where C is the dimension of embedding. Then, F_v^0 and query text feature F_{query} were inputted into the Swin-transformer architecture to guide the visual feature extraction at the four stages through the attention module. That is, at the m-th stage ($1 \le m \le 4$) stage, the visual feature map of the previous stage F_v^{m-1} and F_{query} were incorporated into the Swin-transformer blocks. The attention module realized the guided learning of F_{query} for visual features. Then, the visual feature map F_v^m at each stage was obtained.

By adopting the QRNet concept [21], visual feature extraction under the guided learning of query text used a dynamic linear layer to compute the channel and spatial attention maps related to query text.

First, the dynamic linear layer adopted a query text feature F_{query} to guide the mapping from a given input vector $z_{in} \in R^{Cin}$ to an output vector $z_{out} \in R^{Cout}$. The formula is as follows.

$$z_{out} = DyLinear_{M_{query}}(z_{in}) = DyLinear_{\Psi*(F_{query})}(z_{in}). \quad (1)$$

where $M_{query} = \{W_{query}, b_{query}\} = \Psi*(F_{query})$, linear layer parameter $W_{query} \in R^{Cin\times Cout}$, bias $b_{query} \in R^{Cout}$, and $\Psi*(F_{query})$ indicates that M_{query} is calculated by matrix decomposition.

Then, a visual feature map $F^m \in R^{H\times W\times C_v}$ was generated in the Swin module for calculating a channel attention map at each stage. Firstly, average and maximum poolings were used to gather spatial information and generate the corresponding feature $F^c_{\max}, F^c_{mean} \in R^{1\times 1\times C_v}$. Secondly, the pooled visual features were processed through the dynamic linear layer and ReLU function. After that, the sigmoid function was used to sum the processed visual features with average pooling and maximum pooling to obtain the channel attention map A^{cq}. The calculation process was as follows.

$$F^{cq}_{\max} = DyLinear_1(\mathrm{Re}\,LU(DyLinear_2(F^c_{\max}))). \tag{2}$$

$$F^{cq}_{mean} = DyLinear_1(\mathrm{Re}\,LU(DyLinear_2(F^c_{mean}))). \tag{3}$$

$$A^{cq} = Sigmoid(F^{cq}_{\max} + F^{cq}_{mean}). \tag{4}$$

By calculating the product of the visual feature map F^m and A^{cq}, the visual feature F^m_c in the channel was obtained. The calculation formula is as follows.

$$F^m_c = A^{cq} \otimes F^m. \tag{5}$$

For calculating a spatial attention map, the dynamic linear layer was used to reduce the dimensionality on the channel instead of compressing the channel dimensionality to obtain the areas related to query text. Then, the sigmoid function was used to generate the spatial attention map.

$$A^{sq} = Sigmoid(DyLinear(F^m_c)). \tag{6}$$

$$F^m_v = A^{sq} \otimes F^m_c. \tag{7}$$

where $A^{sq} \in R^{H\times W\times 1}$ refers to the spatial attention map, and F^m_v is the final output of the attention module.

Multiscale Fusion. Multiscale visual features are helpful to detecting objects of different scales. Through the hierarchal architecture of the Swin-transformer, QMVG obtained four visual feature maps of different scales, with a resolution of $[\frac{H}{4}\times\frac{W}{4}\times\frac{H}{8}\times\frac{W}{8}\times\frac{H}{16}\times\frac{W}{16}\times\frac{H}{32}\times\frac{W}{32}]$. To effectively fuse the visual feature maps obtained from different stages, QMVG performed average pooling for the multiscale visual features using a convolutional block with a convolutional kernel of 2×2. That is, average pooling was conducted for the visual feature map F^m_{vf} generated at the m-th stage ($1 \le m \le 3$) so that the map had the same dimension as that generated at the (m+1) stage, and the two visual feature maps were averaged to obtain F^{m+1}_{vf}. Finally, the visual features map F^4_{vf} was flattened into the sequence I_v, which was utilized as the input for the multi-modal reasoning module.

3.4 Multi-stage Reasoning Module

Under the guidance of the query text contextual features introduced in the visual module, the relevance of the generated visual features and the query text was coarse-grained. The construction of fine-grained relevance was required to obtain accurate localization. The QMVG applied a multi-stage decoder for iterative reasoning and achieved iterative interactions between visual information and linguistic information using a cross-attention mechanism to reduce ambiguity in reasoning and thus gradually locate the final target object location.

According to the multi-stage reasoning module shown in Fig. 2, referring to the settings of the number of layers in VLTVG [20], the number of layers of the decoder in QMVG was set to six, i.e., corresponding to six

stages, and each stage consisted of the same network architecture. Additionally, the feature output of the decoder at each stage was used as the feature input of the target query object in the next stage, and iterative reasoning was performed.

In the first stage, a learnable query object $O^1_{query} \in R^{Cv\times 1}$ was preset as the initial representation of the target object and inputted into the first layer of the decoder. Then, through the multi-head cross-attention module, interactive learning was performed between O^1_{query} and text embedding F_q and visual features I_v to collect the features related to query text object (O^1_v) from the visual feature I_v. After that, through the feed forward neural network (FFN) and residual connection and layer normalization (Add&Norm), the target object feature of the first stage $O^2_{query}$was obtained. Then, the visual object feature O^2_{query} generated in the first stage was used as the representation of the query object to input into the decoder in the second stage, which process was consistent with the first stage. Finally, the optimal representation of the query object was obtained through the iterative reasoning of the six stages. The target object $O^i_{query}(1 \le i \le 6)$ at each stage was updated as follows:

$$O'_{query} = LN(O^i_{query} + O^i_v). \tag{8}$$

$$O^{i+1}_{query} = LN(O'_{query} + FFN(O'_{query})). \tag{9}$$

where $L_N(\cdot)$ is the layer normalization, and $FFN(\cdot)$ comprises two linear projection layers and one ReLU activation function.

Through the dynamic updating of the query object O^i_{query} at different stages of the decoder, more attention could be paid to the various descriptions of the query text at each stage. This helped find the target object more finely, aggregate more complete features of the target object, and thus obtain a more accurate visual representation of the target object described by the query text.

3.5 Location of Query Objects

The QMVG inputted the target object features output at each stage of the multi-modal reasoning module to an MLP with a ReLU activation function. The target objects' output coordinate positions at each intermediate stage were used for calculating the loss function. The output of the last stage was used as the coordinate position of the final target object.

The QMVG outputted the coordinates of the final target object's bounding box through the final MLP, calculated the losses between the predicted bounding box and the ground-truth box for each decoder stage, and summed the calculated losses. Herein, $\{\hat{b}\}^N_{i=1} = (x_i, y_i, w_i, h_i)$ denotes the predicted coordinates of the target box from Stage 1 to Stage N, and $b = \{x, y, w, h\}$ denotes the ground-truth box. The training target was as follows.

$$L = \sum_{i=1}^{N} (\lambda_{giou} L_{giou}(b, \widehat{b^i}) + \lambda_{L1} L_{L1}(b, \widehat{b^i})). \tag{10}$$

where $L_{giou}(\cdot)$ and $L_{L1}(\cdot)$ are the GIoU and L1 loss functions, respectively, and λ_{giou} and λ_{L1} are the hyperparameters that balance the two losses during training.

4 Experimental

First, the datasets used in the experiments and the relevant settings of the model were introduced. Then, the performance of the proposed model on five public datasets was analyzed in detail and compared with other state-of-the-art methods. After that, the effectiveness of the proposed model was evaluated and verified through relevant ablation experiments and qualitative visual analysis.

4.1 Datasets and Implementation Details

Datasets. Each referred object in the RefCOCO/RefCOCO+/RefCOCOg datasets corresponded to multiple referring expressions. Herein, the samples in the RefCOCO [6] dataset were split into the training set, validation set, TestA set, and TestB set, containing 120624, 10834, 5657, and 5095 referring expressions, respectively. The RefCOCO+ [6] dataset was subdivided in the same way, with each subset containing 120191, 10758, 5726, and 4889 referring expressions, respectively. The referred objects included multiple identical classes in the subsets obtained by the division of the two datasets. The referred objects in the TestA and TestB sets were people and ordinary objects, respectiveky. The difference between RefCOCO+ and RefCOCO was that the referring expressions in the former dataset contain no "absolute position" -indicating words, such as "left" and "right".

However, compared to the above two datasets, the length of expressions in RefCOCOg [5] was usually longer (the average lengths of RefCOCO, RefCOCO+, and RefCOCOg were 3.61, 3.53, and 8.43, respectively). Additionally, the RefCOCOg dataset had two splitting conventions, namely RefCOCOg-google(Val-g) [5] and RefCOCOg-umd [31] (herienafter abbreviated as Val-u and Test-u, respectively). This study conducted a comprehensive experimental comparison of the two conventions.

The details on the used experimenal data are summarized in Table 2. The images in ReferItGame [8] were extracted from the SAIAPR-12 dataset [32], each image containing one or several areas with corresponding referring expressions. By following the normal method [19], the dataset was partitioned into three subsets: the training set, testing set, and validation set, which had 54127, 5842, and 60103 referring expressions, respectively.

Most referred entities in Flickr 30K Entities [33] were short noun phrases. Meanwhile, 29783 of these images were used for training, 1000 for validation, and 1000 for testing [33, 34].

Table 2. Experimental data details

Dataset	Number of images	Number of referred objects	Number of referring expressions
RefCOCO [9]	19994	50000	142210
RefCOCO+ [9]	19992	49856	141564
RefCOCOg [8]	25799	49822	95010
ReferItGame [12]	20000	96654	120072
Flickr30K Entites [41]	31783	275775	427193

Implementation Details Settings for Model Input. Settings for the model inputs, the size of the input images was set to 640x640, and the maximum length of the query text was set to 40. In resizing the images, their long edges were resized to 640, and the shorter ones were filled to 640 to maintain the original aspect ratio of each image. If the query text length exceeded the maximum allowlable length, the query text was truncated from the end, and then the [CLS] and [SEP] tokens were appended to the beginning and end of the text, respectively. Otherwise, empty tokens were filled after the [SEP] token to make the input length of each batch the same.

During training, QMVG used the AdamW optimizer [35] for end-to-end optimization, and the batch size was set to 16. The initial learning rate of the visual and text feature extraction modules was set to $10^{-5,}$ and the learning rate of other modules was set to 10^{-4}. The proposed visual module was built based on the Swin-transformer and initialized with the corresponding weights obtained from training on MSCOCO [36]. The linguistic module was initialized with $\text{BERT}_{\text{base}}$ (uncase).

Xavier initialization strategy [37] was used to randomly initialize the parameters for the other components in the network. For all datasets, the proposed model was trained for 90 epochs, and the learning rate was reduced by a factor of approximately ten after 60 epochs. The weights of the visual and text feature extraction modules were frozen in the first ten epochs to stabilize the training. The common data augmentation strategy was used, which detailed description can be found elsewhere [10, 13, 19].

4.2 Comparative Analysis Versus Other State-of-the-art Methods

Table 3 compares QMVG and one-stage, two-stage, and other transformer-based visual grounding models on three datasets: RefCOCO, RefCOCO+, and RefCOCOg. In compliance with the consistent standard protocol [19], the top-1 accuracy (%) was used as the comparison metric of model performance, i.e., the prediction was

considered correct if the value of IoU between the predicted area and the ground-truth bounding box exceeded 0.5.

QMVG adopted an end-to-end form, making it possible to avoid the limitations incurred by the performance of auxiliary tools as much as possible. It filtered out the irrelevant object features through the guidance of the query text, and further identified the target object from the visual objects related to the query text by iterative reasoning, thus achieving the accurate location of query objects.

The QMVG model outperformed the compared models on all split subsets of the three datasets. Herein, compared with best-performing one-stage LBYL-Net [25] and HFRN [38] models, the proposed model's accuracy on the split subsets Val, TestA, and TestB of RefCOCO and RefCOCO+ was higher by 7.15%, 5.61%, 9.06%, and 7.62%, 6.25%, 8.19%, respectively. Here, the split subset TestB exhibited the most significant improvement.

Table 3. Comparison of QMVG with other state-of-the-art methods on RefCOCO, RefCOCO+, and RefCOCOg in terms of top-1 accuracy (%)

Model	Backbone	RefCOCO			RefCOCO+			RefCOCOg		
		Val	TestA	TestB	Val	TestA	TestB	Val-g	Val-u	Test-u
One-stage methods										
SSG [9]	DarkNet-53	-	76.51	67.50	-	62.14	49.27	47.47	58.80	-
FAOA [13]	DarkNet-53	72.54	74.35	68.50	56.81	60.23	49.60	56.12	61.33	60.36
ReSC [10]	DarkNet-53	77.63	80.45	72.30	63.59	68.38	56.81	63.12	67.30	67.20
HFRN [38]	ResNet-101	79.76	83.12	75.51	66.80	72.53	59.09	-	69.71	69.08
ISRL [39]	ResNet-101	-	74.27	68.10	-	71.05	58.25	-	-	70.05
LBYL-Net [25]	DarkNet-53	79.67	82.91	74.15	68.64	73.68	59.49	62.70	-	-
Two-stage methods										
MAttNet [12]	ResNet-101	76.65	81.14	69.99	65.33	71.62	56.02	-	66.58	67.27
DGA [14]	VGG16	-	78.42	65.53	-	69.07	51.99	-	-	63.28
RvG-Tree [26]	ResNet-101	75.06	78.61	69.85	63.51	67.45	56.66	-	66.95	66.51
NMTree [40]	ResNet-101	76.41	81.21	70.09	66.46	72.02	57.52	64.62	65.87	66.44
Ref-NMS [27]	ResNet-101	80.70	84.00	76.04	68.25	73.68	59.42	-	70.55	70.62
Transformer-based methods										
TransVG [19]	ResNet-50	80.32	82.67	78.12	63.50	68.15	55.63	66.56	67.66	67.44
TransVG [19]	ResNet-101	81.02	82.72	78.35	64.82	70.70	56.94	67.02	68.67	67.73
VLTVG [20]	ResNet-50	84.53	87.69	79.22	73.60	78.37	64.53	72.53	74.90	73.88
VLTVG [20]	ResNet-101	84.77	87.24	80.49	74.19	78.93	65.17	72.98	76.04	74.18
QRNet [21]	Swin-S	84.01	85.85	82.34	72.94	76.17	63.81	71.89	73.03	72.52
This study	**Swin-S**	**86.91**	**88.73**	**84.57**	**76.26**	**79.93**	**67.68**	**75.88**	**76.88**	**75.64**

Note: The symbol "-" indicates that the respective indicator was not reported in the original literature. The symbol "_" represents the maximum value of the indicator in the existing one-stage, two-stage and transformer-based methods.

Compared with the best-performing two-stage model Ref-NMS [27] and the mainstream transformer-based models VLTVG [20] and QRNet [21], the proposed model also showed a significant improvement in terms of accuracy. As shown in Table 3, compared with VLTVG with the best overall performance among reference models, the proposed model showed improvements of 2.14%, 2.23%, and 2.07%, 2.51% on Val and TestB of the RefCOCO and RefCOCO+ datasets, respectively, and offered a rise of 1.04% and 1.00% on TestA. For the longer query text dataset RefCOCOg, the QMVG model also achieved the best performance, verifying its effectiveness in processing complex queries.

Table 4 shows the performance of QMVG compared with other state-of-the-art models on the testing sets of ReferItGame and Flickr30k Entities. Compared with the one-stage and two-stage methods, the proposed model achieves a significant improvement. However, being applied to the Flickr30k Entities dataset, the proposed model outperformed the best-performing QRNet model among the transformer-based methods only by 0.83%. Such a slight improvement may be related to the fact that the query text in the dataset is mainly short noun phrases, and short query text expresses limited contextual information, inhibiting the interactive learning between visual features and textual features, thus deteriorating the processing of phrase queries.

Table 4. Comparison of QMVG with other state-of-the-art models on the ReferItGame and Flickr30k Entities test sets in terms of top-1 accuracy (%)

Models	Backbone	ReferItGame test	Flickr30K test
One-stage methods			
SSG [9]	DarkNet-53	54.24	-
FAOA [13]	DarkNet-53	60.67	68.71
ReSC [10]	DarkNet-53	64.60	69.28
LBYL_Net [25]	DarkNet-53	67.47	-
SAFF [11]	DarkNet-53	66.01	70.71
Two-stage methods			
MAttNet [12]	ResNet-101	29.04	-
DIGN [16]	VGG16	65.15	78.73
Transformer-based methods			
TransVG [19]	ResNet-50	69.76	78.47
TransVG [19]	ResNet-101	70.73	79.10
VLTVG [20]	ResNet-50	71.60	79.18
VLTVG [20]	ResNet-101	71.98	79.84
QRNet [21]	Swin-S	74.61	81.95
This study	**Swin-S**	**75.83**	**82.78**

4.3 Ablation Study

In this section, the RefCOCOg (Val-g) dataset is used to study the ablation of the QMVG model. The long referring expressions in the dataset pose more challenges to the understanding and reasoning capabilities of the proposed model.

Table 5 shows the results of the ablation experiments on the two modules proposed in the QMVG model to verify their effectiveness. The first row of Table 5 shows the baseline, i.e., no query text was introduced in the visual module for guidance. Only a single-stage decoder was used for reasoning localization, achieving 73.11% accuracy. Based on this baseline, query text was introduced in the visual module to guide the generation of visual features, as shown in the second row of Table 5, with an accuracy improvement of 1.29%. Then, no query text was introduced in the visual module to verify the multi-stage reasoning module. The result is shown in the third row of Table 5, with an accuracy improvement of 1.90% compared to the baseline. The last row of Table 5 shows the performance of the entire model, implying accuracy improvements of 2.77%, 1.48%, and 0.87%, compared to the baseline and the two modules alone, respectively. This verifies the proposed model feasibility.

Table 5. Evaluation of the top-1 accuracy (%) of visual grounding in the ablation experiments on modules in the proposed framework

Import query-text in the visual module	Multi-stage reasoning module	Acc (%)
		73.11
√		74.40
	√	75.01
√	√	75.88

4.4 Qualitative Results

Fig. 4 shows the visualized heat attention maps of QMVG at different stages of the localization process. (A) represents the input of QMVG, (B) and (C) represent the heat attention maps in the visual module, where (B) is the heat attention map without the guidance of query text and (C) is the heat attention map with the guidance of query text, (D) represents the visualized localization heat attention maps of some stages in the multi-stage reasoning process, and (E) represents the localization result of the final target object of QMVG. In the visual module, many visual features extracted by the feature extractor without the guidance of query text are irrelevant to the query text.

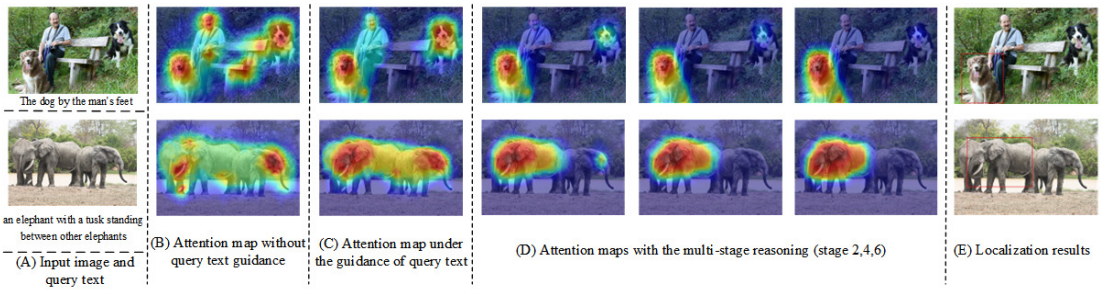

Fig. 4. Visualized heat attention maps of the QMVG at different stages of the localization process

Visual attention is also focused on irrelevant visual object features. In contrast, under the guidance of query text, visual attention pays more attention to visual object features related to the query text, which reduces the interference of irrelevant features. Then, the visual attention map is derived from multi-stage reasoning, and multi-stage interactive reasoning is performed between the relevant visual features obtained in the previous stage and the query text features, which can gradually shift the focus of the visual attention to the target object, thus achieving the localization of the target object.

For instance, in the first row of Fig. 4, given the query text "the dog by the man's feet" and the images, under the guidance of the query text, the visual attention is focused on the object area related to the query text, with a particular bias, focusing more attention on the dog. However, in the absence of guidance of query text, visual attention is focused on more objects, including some objects irrelevant to the query text, which is less biased. Additionally, the heat attention maps obtained after the multi-stage reasoning show that after the interactive reasoning between the query text features and the visual object features, the visual attention is gradually focused on the dog at the bottom left of the image to locate the target object.

5 Conclusions

In this study, a combined network model QMVG based on the Swin-transformer architecture was designed. This model mainly comprised (i) the visual feature generation module based on the guidance of query text and (ii) the multi-stage fused reasoning module. The former introduced the query text information in the visual feature extractor. It used the attention mechanism to guide the learning of visual features. In contrast, the latter used the visual features related to the query text information obtained from the former and the query text information for multiple interactive learnings to locate the target object. The effectiveness of QMVG was experimentally evaluated on five public datasets, outperforming that of twenty-three state-of-the-art methods, including eight one-stage, twelve two-stage, and three transformer-based ones. Additionally, the effectiveness of the query text-guided visual feature generation module and the multi-stage fused reasoning module in QMVG and the process feasibility were verified through an ablation study and qualitative analysis.

In practice, abstract object expressions exist in linguistic expressions, and abstract expressions are confusing for object localization in visual grounding, which may lead to inaccurate object localization. However, the proposed model was trained on a general corpus containing a few abstract language expressions, and the abstract expressions were not processed separately. This might deteriorate the proposed model's performance for abstract language expressions. The follow-up study envisages building datasets containing more abstract language expressions and making the proposed model more adaptive by designing specialized modules to process abstract language expressions.

6 Acknowledgement

This work was supported by the National Natural Science Foundation of China (31771679, 31671589), Major Scientific and Technological Projects in Anhui Province, China (201903a06020009), Anhui Natural Science Foundation (2108085MF209), the Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture

of China under Grant (AEC2021001), Natural Science Research Project of Anhui Provincial Department of Education (KJ2020A0107, KJ2021A1550), The University Synergy Innovation Program of Anhui Province (GXXT-2022-046, GXXT-2022-055).

References

[1] K. Arora, A. Raj, A. Goel, S. Susan, A Hybrid Model for Combining Neural Image Caption and k-Nearest Neighbor Approach for Image Captioning, in: Proc. 2022 International Conference on Soft Computing and Signal Processing, 2022.
[2] M. Zheng, Y. Jia, H. Jiang, Fine-grained image-text retrieval via complementary feature learning, in: Proc. 2021 International Conference on Multimedia Modeling, 2021.
[3] W. Tian, Y. Zhang, B. He, J. Zhu, Z. Zhao, Visual-Textual Semantic Alignment Network for Visual Question Answering, in: Proc. 2021 International Conference on Artificial Neural Networks, 2021.
[4] G.J. Burghouts, W. Huizinga, Coarse-to-Fine Visual Question Answering by Iterative, Conditional Refinement, in: Proc. 2022 International Conference on Image Analysis and Processing, 2022.
[5] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, K. Murphy, Generation and comprehension of unambiguous object descriptions, in: Proc. 2016 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.
[6] Li. Yu, P. Poirson, S. Yang, A.C. Berg, T.L. Berg, Modeling context in referring expressions, in: Proc. 2016 the European Conference on Computer Vision, 2016.
[7] K. Chen, R. Kovvuri, R. Nevatia, Query-guided regression network with context policy for phrase grounding, in: Proc. 2017 the IEEE/CVF International Conference on Computer Vision, 2017.
[8] S. Kazemzadeh, V. Ordonez, M. Matten, T. Berg, ReferItGame: Referring to objects in photographs of natural scenes, in: Proc. 2014 the conference on empirical methods in natural language processing (EMNLP), 2014.
[9] X. Chen, L. Ma, J. Chen, Z. Jie, W. Liu, J. Luo, Real-time referring expression comprehension by single-stage grounding network, arXiv preprint arXiv:1812.03426 (2018).
[10] Z. Yang, T. Chen, L. Wang, J. Luo, Improving one-stage visual grounding by recursive sub-query construction, in: Proc. 2020 the European Conference on Computer Vision, 2020.
[11] J. Ye, X. Lin, L. He, D. Li, Q. Chen, One-stage visual grounding via semantic-aware feature filter, in: Proc. 2021 the 29th ACM International Conference on Multimedia, 2021.
[12] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, T.L. Berg, MAttNet: Modular attention network for referring expression comprehension, in: Proc. 2018 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[13] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, J. Luo, A fast and accurate one stage approach to visual grounding, in: Proc. 2019 the IEEE/CVF International Conference on Computer Vision, 2019.
[14] S. Yang, G. Li, Y. Yu, Dynamic graph attention for referring expression comprehension, in: Proc. 2019 the IEEE/CVF International Conference on Computer Vision, 2019.
[15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28(2015) 91–99.
[16] Z. Mu, S. Tang, J. Tan, Q. Yu, Y. Zhuang, Disentangled motif-aware graph learning for phrase grounding, in: Proc. 2021 the AAAI Conference on Artificial Intelligence, 2021.
[17] S. Yang, G. Li, Y. Yu, Graph-structured referring expression reasoning in the wild, in: Proc. 2020 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[18] Y. Du, Z. Fu, Q. Liu, Y. Wang, Visual grounding with transformer. <https://arxiv.org/abs/2105.04281> 2021 (accessed 09.07.2022).
[19] J. Deng, Z. Yang, T. Chen, W. Zhou, H. Li, TransVG: End-to-end visual grounding with transformers, in: Proc. 2021 the IEEE/CVF International Conference on Computer Vision, 2021.
[20] L. Yang, Y. Xu, C. Yuan, W. Liu, B. Li, W. Hu, Improving visual grounding with visual-linguistic verification and iterative reasoning, in: Proc. 2022 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[21] J. Ye, J. Tian, M. Yan, X. Yang, X. Wang, J. Zhang, L. He, X. Lin, Shifting More Attention to Visual Backbone: Query-modulated refinement networks for end-to-end visual grounding, in: Proc. 2022 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[22] H. Tan, M. Bansal, LXMERT: Learning cross-modality encoder representations from transformers, in: Proc. 2019 the Conference on the Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
[23] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sunderhauf, I. Reid, S. Gould, A.V.D. Hengel, Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments, in: Proc. 2018 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[24] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement. <https://arxiv.org/abs/1804.02767>, 2018 (accessed 12.05.2020).
[25] B. Huang, D. Lian, W. Luo, S. Gao, Look Before You Leap: Learning landmark features for one-stage visual grounding, in: Proc. 2021 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[26] R. Hong, D. Liu, X. Mo, X. He, H. Zhang, Learning to compose and reason with language tree structures for visual grounding. IEEE transactions on pattern analysis and machine intelligence 44(2)(2019) 684-696.
[27] L. Chen, W. Ma, J. Xiao, H. Zhang, S.-F. Chang, Ref-NMS: Breaking proposal bottlenecks in two-stage referring expression grounding, in: Proc. 2021 the AAAI Conference on Artificial Intelligence, 2021.
[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Proc. 2020 the European Conference on Computer Vision, 2020.
[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical vision transformer using shifted windows, in: Proc. 2021 the IEEE/CVF International Conference on Computer Vision, 2021.
[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding. <https://arxiv.org/abs/1810.04805>, 2018 (accessed 10.07.2020).
[31] V.K. Nagaraja, V.I. Morariu, L.S. Davis, Modeling context between objects for referring expression understanding, in: Proc. 2016 the European Conference on Computer Vision, 2016.
[32] H.J. Escalante, C.A. Hernández, J.A. Gonzalez, M. Montes, E.F. Morales, L.E. Sucar, L. Villasenor, M. Grubinger. The segmented and annotated IAPR TC-12 benchmark. Computer vision and image understanding 114(4)(2010) 419-428.
[33] B.A. Plummer, L. Wang, C.M. Cervantes, J.C. Caicedo, J. Hockenmaier, S. Lazebnik, Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, in: Proc. 2015 the IEEE/CVF International Conference on Computer Vision, 2015.
[34] B.A. Plummer, P. Kordas, M.H. Kiapour, S. Zheng, R. Piramuthu, S. Lazebnik, Conditional image-text embedding networks, in: Proc. 2018 the European Conference on Computer Vision, 2018.
[35] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).
[36] T.Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick, P. Dollar, Microsoft COCO: Common objects in context, in: Proc. 2014 the European Conference on Computer Vision, 2014.
[37] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proc. 2010 the thirteenth international conference on artificial intelligence and statistics, 2010.
[38] H. Qiu, H. Li., Q. Wu, F. Meng, H. Shi, T. Zhao, K.N. Ngan, Language-aware fine-grained object representation for referring expression comprehension, in: Proc. 2020 the 28th ACM International Conference on Multimedia, 2020.
[39] M. Sun, J. Xiao, E.G Lim, Iterative shrinking for referring expression grounding using deep reinforcement learning, in: Proc. 2021 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[40] D. Liu, H. Zhang, Z.-J. Zha, F. Wu, Learning to assemble neural module tree networks for visual grounding, in: Proc. 2019 the IEEE/CVF International Conference on Computer Vision, 2019.