LaLGA: Multi-Scale Language-Aware Visual Grounding on Remote Sensing Data

Ke Li 1
1Xidian University

October 31, 2023

Abstract

Visual grounding of remote sensing (RSVG) aims to detect the specific objects associated with query expressions in remote sensing (RS) images.
LaLGA: Multi-Scale Language-Aware Visual Grounding on Remote Sensing Data

Ke Li, Di Wang, Haojie Xu, Quan Wang, Gang Liu, Weiping Ni, and Xiao Gao

Abstract—Visual grounding of remote sensing (RSVG) aims to detect the specific objects associated with query expressions in remote sensing (RS) images. Existing methods typically integrate features from pre-trained visual backbone with text embeddings to ground the referred target. However, due to the lack of linguistic guidance in the visual backbone, the extracted visual features may not match the linguistic features, limiting detection performance. To avoid this, we propose a novel visual grounding framework for RS Data, named LaLGA, which guides the visual backbone to focus on the referred target. Specifically, a language adaptive weight generator (LA WG) is proposed to dynamically generate multi-scale language-adaptive weights, enabling the visual backbone to learn expression-relevant visual features layer by layer. Additionally, a multi-level language guided alignment (MLGA) module is devised to aggregate visual contextual information of the referred target and filter complex background noise to enhance the uniqueness of the object, improving the accuracy of localization. Furthermore, a new large-scale benchmark dataset of visual grounding named OPT-RSVG is constructed based on optical RS data to further advance research on RSVG. We compared the proposed method with state-of-the-art methods and conducted in-depth analyses and discussions. Experiments show that the proposed method pushes the accuracy score to 82.27% (6.29% point absolute improvement) on the DIOR-RSVG dataset and 74.69% on the OPT-RSVG dataset, setting new records. The source code of the proposed method and the OPT-RSVG dataset are available at https://github.com/like413/OPT-RSVG.

Index Terms—LaLGA, language adaptive weight generator, multi-level language guided alignment, visual grounding on remote sensing data (RSVG).

I. INTRODUCTION

VISUAL grounding of remote sensing (RSVG) aims to localize specific objects based on a given query expression. Query expressions consist of phrases and sentences describing targets in remote sensing (RS) data, and localizations of objects are marked using bounding boxes. Compared to general object detection trained only on fixed categories, visual grounding is more flexible [1], [2]. Free-formed query expressions allow users to specify visual attributes of the target objects, such as category, color, absolute position in the RS data, relative position to other objects. Moreover, RSVG enables non-experts to easily access information contained in RS data, which has important research significance and wide application prospects in scenarios such as military target localization and recognition [3], military intelligence analysis [4], infrastructure management [5], disaster response [6], urban planning [7] and agricultural monitoring [8].

Research on RSVG is still limited as it is a new topic. Sun et al. [9] first proposed the concept of RSVG and put forward a novel RSVG method named GeoVG. It consists of three components: an image encoder, a language encoder, and a fusion module. The image encoder introduces an adaptive region attention module that distills key visual content from a large-scale scene. The language encoder builds a geospatial relationship graph based on target objects and geographical coordinate distances between objects in the textual description, to understand the complex expression content. Finally, a fusion module is designed to combine the constructed geographic relationship graph with the visual features to locate target. Recently, Zhan et al. [10] developed a novel RSVG method named MGVLF. It utilized large-scale language and vision pre-training transformer-based models to extract language and visual features, respectively. Compared to GeoVG, MGVLF first utilizes multiscale visual features and multigranularity textual embeddings to guide the visual feature refining and achieve multigranularity visual language learning.
Despite some progress has been made in RSVG, visual backbones of the aforementioned methods are not well explored. Specifically, feature extraction from visual backbone relies exclusively on visual information, without taking into account the potential relevance of language and visual information, as shown in Fig. 1(a) and (b). Neglecting language information in visual feature extraction may cause visual backbone passively obtains visual features, which may not match the objects referred in the query expression. Taking Fig. 2(b) as an example, without referring language information, visual backbone exhibits attention on both airplanes, which can lead to imprecise matching of query expressions like “left airplane” or “right airplane”. Ideally, visual backbone should actively focus on extracting features of desired objects most relevant to queries. As expressions can provide information about the color, size, location and other attributes of the desired visual objects, incorporating language information into visual backbones is a feasible way to extracting precise visual features. Moreover, compared to natural scene images, the limited texture and structural information in RS data poses challenges in achieving accurate recognition and localization of objects. Factors such as variations in object scales and cluttered backgrounds further exacerbate these difficulties. Therefore, RSVG needs perceive objects of different scales to effectively reduce complex background and noise interference and enhance the robustness of localization.

To meet these challenges, we design a novel model named LaLGA, which enhances the effectiveness of visual feature extraction through multi-scale language perception. It dynamically adjusts the behavior of the visual backbone by generating multi-scale weights based on query expression information. Specifically, a language adaptive weight generator (LAWG) is designed to enable visual backbone extracting expression-relevant visual features with the guidance of language information. Additionally, a multi-level language guided alignment (MLGA) module is designed to aggregate the visual contextual information of the target object and enhance its distinctiveness.

To further promote the research on RSVG, we propose a new dataset OPT-RSVG. The OPT-RSVG dataset is sampled from object detection datasets HRRSD [11], DIOR [12], and SPCD (Swimming Pool And Car Detection) [13], containing 25452 images and 48952 image-query pairs. The query expressions include not only basic category information but also object attributes and relationship attributes such as color, shape, size, and position. Compared to the currently largest dataset DIOR-RSVG, our dataset has more image-query pairs, a larger spatial resolution span, more balanced categories, and more abundant objects of each category. In addition, the query expressions in OPT-RSVG have both Chinese and English versions.

The main contributions can be summarized as follows:
1) We propose a novel RSVG framework, named LaLGA, which consists of two modules: LAWG and MLGA. LAWG utilizes language information to enable the visual backbone to learn expression-relevant visual features layer by layer. MLGA can aggregate visual contextual information of the target object to enhance its uniqueness.
2) To foster the research of RSVG, we construct a new large-scale dataset, termed OPT-RSVG. Specifically, the dataset contains 25452 RS images and 48952 image-query pairs, with expressions provided in both Chinese and English versions.
3) Experiments demonstrated the effectiveness of our framework, achieving accuracies of 74.69% and 82.27% on the OPT-RSVG and DIOR-RSVG datasets, respectively, and establishing a new state-of-the-art (SOTA).

The rest of this article is organized as follows. Section II introduces relevant works of RSVG. In Section III, the construction procedure of the new dataset is described, and its characteristics are analyzed. Section IV elaborates on the proposed method. The extensive experiments and analyses are presented in Section V. Finally, the conclusions of this work and some discussions on possible future research directions are drawn in Section VI.

II. RELATED WORK

In this section, we comprehensively review the closely related works about natural image visual grounding and visual grounding on remote sensing. To be more specific, Section II-A provides a detailed summary of two-stage, one-stage, and transformer-based methods in natural image visual grounding. Existing RSVG methods and datasets are reported in Section II-B.

A. Natural Image Visual Grounding

Two-Stage Methods. Inspired by the success of region based object detectors, two-stage visual grounding methods are characterized by generating region proposals in the first stage and then selecting the best matching one corresponding to the language expression in the second stage. Generally, region proposals are generated by a pre-trained object detector [14], [15]. The main efforts of approaches in this direction are devoted to the second stage [16], [17], [18], addressing visual grounding as text-region matching. One representative work in this category is RegionCLIP [19], which utilizes pseudo region text for pre-training on region visual language and transfers the backbone network to the localization task. Lu et al. [20] proposed using text descriptions to predict the object category without modifying Faster R-CNN [21] to alleviate confusion in classification. Recent studies further improve the two-stage methods by better modeling the object relationships [22], [23], or making use of phrase co-occurrences [24], [25]. However,

Fig. 2. Attention visualization of the visual backbone. (a) Input image. (b) Without language guidance, the visual backbone focuses on “airplane”. (c) and (d) With the language guidance, the visual backbone focuses on the objects referred in the text (e.g., “left airplane” and “right airplane”), respectively.
the two-stage approaches still require filtering a large number of candidate boxes, resulting in increased time overhead and reduced efficiency.

**One-Stage Methods.** To address the aforementioned issues of two-stage methods, simpler and faster one-stage visual grounding methods were proposed [26], [27], [28]. The pioneering work FAOA [29] encoded language expressions into a textual embedding and fused it into the YOLOv3 detector [30] to ground the referred instance. The model generates dense object detections with confidence scores and selects the top-ranked one as reference objects for prediction. Yang et al. [31] proposed a recursive sub-query construction (ReSC) framework to address the limitations of FAOA on grounding complex queries by multi-round fusion. Liao et al. [32] proposed a language-customized visual feature learning mechanism, which further improved the reasoning ability of the one-stage grounding. Although one-stage methods are efficient, they typically use point features as object representations, which may not be flexible enough to capture detailed descriptions in expressions.

**Transformer-Based Methods.** Recently, transformer-based visual grounding methods have gained significant research attention due to their efficiency in processing multimodal data. TransVG [33] established multi-modal correspondences by Transformers and localized referred regions by directly regressing box coordinates. However, the core fusion Transformer in TransVG is stand-alone against uni-modal encoders, and thus should be trained from scratch on limited visual grounding data, which makes it hard to be optimized and leads to sub-optimal performance. To this end, Deng et al. [34] proposed TransVG++, which is a purely transformer-based framework that includes BERT [35], Vision Transformer [36], and multimodal fusion transformers. Recent methods have started to focus on the design of the visual branch, utilizing cross-modal attention mechanisms to adjust visual features. For example, VLTGV [37] employs an encoder-decoder architecture that aims to adjust visual features using a visual-linguistic verification module and aggregate visual context using a language-guided context encoder.

**B. Visual Grounding on Remote Sensing**

**RSVG Methods.** Compared to query expressions in natural images, expressions in RSVG often involve complex geographical spatial relationships, and objects of interest are usually not visually prominent. Despite the extensive research on visual grounding of natural images, RSVG has not been fully explored. Sun et al. [9] first proposed the concept of RSVG and put forward a novel RSVG method named GeoVG. The method consists of three components: an image encoder, a language encoder, and a fusion module. The image encoder introduces an adaptive region attention module that distills key visual content from a large-scale scene. The language encoder builds a geospatial relationship graph based on target objects and geographical coordinate distances between objects in the textual description, to understand the complex expression content. Another recent work was proposed by Zhan et al. [10], in which they developed a novel transformer-based RSVG method called MGVLF. Compared to GeoVG, it utilized multi-scale visual features and multi-granularity language features to address the scale variation problem. On the other hand, MGVLF dynamically filters out irrelevant noise and strengthens salient features to deal with cluttered backgrounds. However, the lack of language guidance in the visual branches of these two methods can result in a mismatch between the visual focus and the referent of the language, leading to a decrease in the model’s performance.

**RSVG Datasets.** Since RSVG is a relatively new topic, RSVG datasets are very limited. Sun et al. [9] pioneered the RSVG task and proposed the first dataset RSVG-H\(^1\), which contains 4239 remote sensing images and 7933 text expressions for 5994 objects. The average length of the expressions is 28.41 words. Expressions in RSVG-H contain a large number of distance relationships (e.g., “Find a ground track field, located approximately 295 meters southeast of a baseball field.”). However, obtaining accurate target location information is usually not easy, resulting in limitations of RSVG-H in practical applications. Another recent work was proposed by Zhan et al. [10]. They constructed a larger-scale dataset with 17402 RS images and 38320 language expressions, covering 20 target categories. The DIOR-RSVG dataset has more RS images and its textual descriptions are comparatively more ambiguous than those of RSVG-H, which further promotes the development of RSVG. However, the DIOR-RSVG dataset exhibits extreme class imbalance, with a clear long-tail distribution. For example, there are 7161 descriptions for the vehicle category, but only 578 descriptions for the harbor category. This imbalance can have a detrimental impact on the model’s performance, causing it to be biased towards predicting frequent categories while ignoring categories with fewer instances.

## III. DATASET CONSTRUCTION

In this section, we will introduce the construction process of the new dataset in Section III-A. The statistical analysis of our OPT-RSVG will be presented in Section III-B.

### A. OPT-RSVG: A New Large-Scale Dataset for RSVG

Due to the similarity between RSVG and object detection tasks, we collected RS images and location information from existing object detection datasets HRSSD, DIOR, and SPCD to construct OPT-RSVG dataset. Textual descriptions are then automatically generated according to the potentially relevant properties of the object. The textual descriptions for the RSVG should take into account both the attributes of the objects and the relationships between objects [38]. Object attributes include size, color, shape, spatial position, etc. Relationship attributes include relative size, relative position, etc. To achieve this, we designed a two-stage process for automated description generation and manual verification. The following provides a detailed explanation:

**Automatic Description Generation.** To achieve the automatic generation of query expressions, we divided the process into four steps: data preprocessing, extraction of object

\(^1\)We use RSVG-H to represent the RSVG dataset proposed in [9], in order to distinguish it from the visual grounding of remote sensing (RSVG).
attributes, extraction of object relationship attributes, and generation of query expressions. The specific label space can be found in Table I.

**Step 1:** Data preprocessing. HRRSD, DIOR, and SPCD include in total 47927 images. The spatial resolution range from 0.15m to 30m. Following the same processing method as [10], we removed bounding boxes that were less than 0.02% or greater than 99% of the image size, and ensured that no more than 5 objects of the same category were sampled in each RS image.

**Step 2:** Extraction of object attributes. We used a combination of prior knowledge and algorithmic recognition to extract object attributes.

1) `<category>`. The object category information is already provided in original object detection datasets.
2) `<size>`. According to the ratio of the bounding box area to the image size, the object’s size is divided into three types: large, medium and small.
3) `<color>`. Some objects have relatively fixed color attributes, such as gray roads and blue swimming pools. So we pre-set these objects’ color attribute. For colorful objects such as cars and airplanes, the HSV method is used to extract color information of objects.
4) `<shape>`. For some objects with fixed shapes, such as a fan-shaped baseball field or a rectangular basketball court, the shape attribute is set in advance. For other objects with various shapes, no shape attribute is set.
5) `<absolute location>`. The image is divided into a 3×3 grid, and the absolute position of the object in the image is determined based on the center coordinates of the bounding box.

Incorporating prior knowledge in the expression generation process greatly reduces the error rate and facilitates subsequent manual verification.

**Step 3:** Extraction of relationship attributes. Relationship attributes are generally used to describe scenes with multiple objects in an image. The relative position relationship is obtained by comparing the coordinates of the center points of different object bounding boxes. The relative size relationship is determined by comparing the ratio of the bounding box area of two objects to the image size.

**Step 4:** Generation of query expressions. To make the generated query expressions more natural, we have pre-defined text templates for the dataset. The text templates include object templates and relationship templates. The object template in the following form:

There is a/an ⟨a2/a3/a4⟩ ⟨a1⟩ in/on the ⟨a5⟩.

The relationship template in the following form:

Reference object’s feature template, searching for a /an ⟨a6⟩ ⟨a1⟩ in the ⟨a7⟩ of the above object.

We chose textual templates and populate attributes to generate a query expression for each bounding box. The generation algorithm can be summarized as the following few steps.

1) First, we determine if the object category is unique in the RS image. If so, we populated the object attribute template with the category and randomly selecte object properties.
2) If the object category is not unique, we look for a unique attribute of the object that distinguishes it from other objects of the same category. If such a attribute exists, we combine it with the category name to fill the object attribute template. We prioritize using fuzzy attributes such as color, size, and shape, and if none of these are available, we use absolute position. (fuzzy attributes are more valuable in practical applications.)
3) If there is no such unique object attribute, we use relationship attribute for positioning. We determine if there are other objects that have already been described by 1) or 2) around the object, and if so, we combine the two objects using relationship attribute to populate the relationship attribute template.
4) If all the above fails, the object is discarded.

**Manual verification and validation.** Due to the large differences in object scales and complex backgrounds in RS images, errors may occur in feature attribute extraction, especially in object-specific feature properties. Therefore, manual verification is required for OPT-RSVG to help correct errors in language expressions. To improve efficiency, we have developed a dataset visualization correction system.

It is worth noting that we have also thoughtfully provided Chinese query expressions for OPT-RSVG to better promote the use of the dataset.

**B. Analysis of the OPT-RSVG**

We constructed a large-scale OPT-RSVG, where each object instance in the RS image corresponds to a unique language expression. Our constructed OPT-RSVG consists of 25452 RS images and 48952 image-query pairs and contains 14 object
categories. We now present a more detailed statistical analysis of the OPT-RSVG dataset.

Fig. 3(a) shows the proportion of the number of each object category. We can see that vehicles have the highest proportion of 16.73% with 8188 objects, ports account for the least but also have 1920 objects, and the proportions of the remaining 12 categories are relatively uniform, with the number between 2500 and 5000. Fig. 3(b) shows the proportion of described object attributes. The inner circle represents object attributes and relationship attributes, while the outer circle represents the proportion of each type of attribute in quantity. Fig. 3(c) shows the length distribution of English query expressions. The average query expression length is 10.10 words, with a minimum of 3 words and a maximum of 32 words. Fig. 3(d) shows the length distribution of Chinese query expressions. The average query expression length is 9.67 characters, with a minimum of 2 words and a maximum of 36 words. Fig. 3(e)(f) and (g) shows the distribution of average width, height, and area of all object bounding boxes to the images. Fig. 3(h)(i)(j) and (k) shows the word clouds of object categories and all attributes in English and Chinese. The most common object names are vehicles, bridges, and crossroad. The most common attributes are color (e.g., blue, and white), size (e.g., small), and position (e.g., left, right, top, and bottom), and shape (e.g., rectangle and round).

IV. METHODOLOGY

In this section, we will introduce the multi-scale language-aware framework for RSVG, including the language adaptive weight generation (LAWG) and multi-level language guided alignment (MLGA) module. We first overview the overall framework in Section IV-A. The design of the LAWG module is detailed in Section IV-B. Section IV-C details the designs of the MLGA module.

A. Overview

Passive perceptual extraction of RS data features by the visual backbone may lead to a mismatch problem with linguistic features, limiting the potential for performance improvement despite subsequent carefully designed multimodal fusion modules. Considering that query expressions already provide a guidance for the desired RS data features, we propose a multi-scale RSVG framework based on the language adaptive weights, called LaLGA, as illustrated in Fig. 4. In this framework, the visual backbone can actively extract RS data features related to query expressions using language-adaptive weights, without needing to manually modify the visual backbone architecture.

Specifically, the LaLGA comprises six components: the visual backbone, the language backbone, the language adaptive weight generator, the multi-level language guided alignment module, the multimodal fusion module, and the localization module. Given a query expression, the BERT-based [35] language backbone extracts linguistic features $F_L \in \mathbb{R}^{l \times d}$, where $l$ and $d$ represent the number of linguistic feature tokens and their dimensionality, respectively. The linguistic features $F_L$ are then input to the language adaptive weight generator to generate weights for the DETR-based [39] visual backbone. Next, given the RS data $I \in \mathbb{R}^{H \times W \times 3}$, the visual backbone extracts language-aware visual features $F_V \in \mathbb{R}^{h \times w \times c}$, where $H, W$ and $h, w$ represent the height and width of the RS data and visual features, $c$ represents the dimensions of the
visual feature. These language-perceived visual features \( F_V \) and the linguistic features \( F_L \) are then passed to the multi-level language guided alignment module to obtain \( F_{MLGA} \). This module aggregates visual contextual information of the target object while reducing the interference of irrelevant noise. Subsequently, \( F_{MLGA} \) and \( F_L \) are separately mapped to the same dimensionality through two linear projections. These mapped features, along with a learnable label, are then jointly fed into a multimodal fusion transformer module. Finally, \( F_{token} \) is passed to the localization module to predict the coordinates of bounding box. Following the previous method [33], we apply the same loss function for training our proposed network.

### B. Language Adaptive Weight Generation

After extracting linguistic features, language adaptive weights are generated to guide the active perception of the visual backbone. As shown in Fig. 5, the LAWG essentially functions as an attention mechanism, and its calculation process is similar to linear projection. It consists of two stages: linguistic weight generation and linguistic bias generation.

#### Linguistic-Based Weight and Bias Generation

Considering that each stage of the visual backbone focuses on different scale features, we aim to generate dynamic language features independently for each stage. Inspired by the multi-head attention mechanism [40], we introduce two learnable embeddings \( e_{i,j}^r \in \mathbb{R}^d \) and \( e_{i,j}^l \in \mathbb{R}^d \) for the \( j \)-th features of \( i \)-th stage of the visual backbone to extract layer-specific linguistic features dynamically. For each stage \( i \), the token-wise attention \( \omega_i \in [0, 1]^L \) and \( \beta_i \in [0, 1]^L \) are assigned to the normalized dot product of \( e \) and \( F_L \), which is denoted as:

\[
\omega_i = \text{Softmax} \left( \left[ e_{i,1}^r \cdot F_L^1, e_{i,2}^r \cdot F_L^2, \ldots, e_{i,d}^r \cdot F_L^d \right] \right) \quad (1)
\]

\[
\beta_i = \text{Softmax} \left( \left[ e_{i,1}^l \cdot F_L^1, e_{i,2}^l \cdot F_L^2, \ldots, e_{i,d}^l \cdot F_L^d \right] \right) \quad (2)
\]

Then, the linguistic-based weights \( w_i \in \mathbb{R}^d \) and linguistic-based biases \( b_i \in \mathbb{R}^d \) can be derived by concatenating:

\[
w_i = \sum_{l=1}^{L} \omega_i F_L^l, \quad b_i = \sum_{l=1}^{L} \beta_i F_L^l \quad (3)
\]

Finally, we use a fully-connected layer \( fc \) to reduce the dimension of the \( w_i \) and \( b_i \) for the \( i \)-th layer of the visual backbone, which are indicated as:

\[
W_i = \text{GeLU} (fc(w_i)), \quad B_i = \text{GeLU} (fc(b_i)) \quad (4)
\]

where \( W_i \in \mathbb{R}^{c_i}, B_i \in \mathbb{R}^{c_i}, c_i \) represents number of visual feature channels of stage \( i \)-th and GeLU represents GeLU activation function.

In practice, it is easier to refer to \( fc(w_i) \) and \( fc(b_i) \) as a single function that outputs one \((W_i, B_i)\) vector, since, for example, it is often beneficial to share parameters across \( W_i \) and \( B_i \) for more efficient learning.

#### Multi-Scale Guide Visual Feature Extraction

We generate language-adaptive weights and biases for applying an affine transformation to the intermediate features of the network to guide the multi-scale active perception of the visual backbone, which can be represented as:

\[
\text{LAWG}(F_i || W_i, B_i) = W_i F_i + B_i \quad (5)
\]
Since each feature map is independently tuned, giving the LAWG moderately fine-grained control over the activations in each stage of the backbone network.

Furthermore, as LAWG only requires two parameters per modulated feature map, it is a scalable and computationally efficient conditioning method. In particular, LAWG has a computational cost that does not scale with the image resolution.

C. Multi-Level Language Guided Alignment

In order to fully utilize linguistic information to enhance target object information, a MLGA module is carefully designed to fuse visual and linguistic features. Specifically, the MLGA contains left and right branches, as shown in Fig. 6. The left branch reduces the interference of background or other noise through Cross Attention Layer (CAL) to improve the uniqueness of the target object. The right branch is used to aggregate target object visual context information through two visual-linguistic guidances (specially designed two CALs in series). Due to the strategy of guiding before fusing, the semantic discrepancy between the two modalities of data is reduced, resulting in a more robust fusion. Finally, the output of these two branches are added to get the result of multi-level language guidance.

Cross Attention Layer. The CAL contains an multi-head cross attention (MCA) module, a feed-forward network (FFN) and two layer normalization (LN) blocks with residual connection, as follows:

\[
Q = W_qX, \quad K = W_kX, \quad V = W_vX
\]

where \(W_q, W_k, W_v \in \mathbb{R}^{d_{out} \times d_{in}}\) are learnable weights, used to generate the query \(Q\), key \(K\), and value \(V\), respectively. \(d_{in}\) and \(d_{out}\) are the dimension of feature \(X\) and query/key/value, respectively.

\[
\text{Att}_i(\cdot) = \text{Softmax} \left( \frac{Q_i K_i}{\sqrt{d}} \right) V_i,
\]

\[
\text{MCA}(\cdot) = \text{Concat}(\text{Att}_1, \text{Att}_2, \cdots, \text{Att}_h),
\]

\[
\text{LN}_1(\cdot) = \text{norm}(X + X_{\text{MCA}}),
\]

\[
\text{FFN}(\cdot) = fc(fc(X_{\text{LN}_1})),
\]

\[
\text{LN}_2(\cdot) = \text{norm}(X_{\text{FFN}} + X_{\text{LN}_1})
\]

\[
\text{CAL}(\cdot) = \text{MCA}(\cdot) \Rightarrow \text{LN}_1(\cdot) \Rightarrow \text{FFN}(\cdot) \Rightarrow \text{LN}_2(\cdot)
\]

where \(i\) represents the number of attention heads, \(d\) represents the dimension of \(K_i\), \(fc\) represents the fully connected layer, and \(\text{norm}\) represents layer normalization. It is worth noting that \(X\) here is just for convenience, in actual situations \(X\) may come from different features.

Multi-Level Guided Alignment. The visual feature map \(F_V \in \mathbb{R}^{d_{out} \times d_{in}}\) and linguistic feature vector \(F_L \in \mathbb{R}^{L \times d}\) are passed through two linear projections respectively to obtain the same dimension and add multimodal position embedding, as shown in the formula below:

\[
\tilde{F}_V = fc(F_V) + \text{PosEmbedding}(F_V)
\]

\[
\tilde{F}_L = fc(F_L) + \text{PosEmbedding}(F_L)
\]

where \(\text{PosEmbedding}(\cdot)\) denotes the function to get positional embedding, \(F_V, \tilde{F}_L \in \mathbb{R}^{L \times d}\).

For the left branch of MLGA module, the visual feature map \(F_V\) acts as the query \(Q_1\). The linguistic feature vector \(F_L\) serves as the \(K_1\) and \(V_1\), as shown in the formula below:

\[
\tilde{Q}_1 = W_{q_1} \tilde{F}_V, \quad \tilde{K}_1 = W_{k_1} \tilde{F}_L, \quad \tilde{V}_1 = W_{v_1} \tilde{F}_L
\]

where \(W_{q_1} \in \mathbb{R}^{d_{out} \times \tilde{d}_1}, W_{k_1} \in \mathbb{R}^{d_{LCA} \times \tilde{d}_1}, W_{v_1} \in \mathbb{R}^{d_{LCA} \times \tilde{d}_1}\) are learnable weights.

Next, we use the CAL to compute \(\tilde{Q}_1, \tilde{K}_1, \text{and } \tilde{V}_1\) to obtain the left branch output’s feature map \(F_{left}\), as follows:

\[
F_{left} = \text{CAL}(\tilde{Q}_1, \tilde{K}_1, \tilde{V}_1)
\]

For the right branch of MLGA module, the input of the first CAL is completely opposite to that of the left branch’s CAL. Linguistic feature vector are used to generate \(\tilde{Q}_2\), while visual feature map are used to generate \(\tilde{K}_2\) and \(\tilde{V}_2\), as follows:

\[
\tilde{Q}_2 = W_{q_2} \tilde{F}_V, \quad \tilde{K}_2 = W_{k_2} \tilde{F}_L, \quad \tilde{V}_2 = W_{v_2} \tilde{F}_L
\]

where \(W_{q_2} \in \mathbb{R}^{d_{out} \times \tilde{d}_2}, W_{k_2} \in \mathbb{R}^{d_{LCA} \times \tilde{d}_2}, W_{v_2} \in \mathbb{R}^{d_{LCA} \times \tilde{d}_2}\) are learnable weights.

We feed \(\tilde{Q}_2, \tilde{K}_2, \text{and } \tilde{V}_2\) into another CAL for fusion to obtain middle guided features \(F_{middle}\), which can be obtained by the following equation:

\[
F_{middle} = \text{CAL}(\tilde{Q}_2, \tilde{K}_2, \tilde{V}_2)
\]

Then, we take the sum of \(\tilde{F}_V, F_{middle}\) as the query and key of the CAL and input the visual feature map \(F_V\) as the value, to compute the feature correlations in both visual and linguistic representations, as shown in the formula below:

\[
\tilde{Q}_3 = W_{q_3} (\tilde{F}_L + F_{middle}),
\]

\[
\tilde{K}_3 = W_{k_3} (\tilde{F}_L + F_{middle}),
\]

\[
\tilde{V}_3 = W_{v_3} \tilde{F}_L
\]
Algorithm 1 LaLGA

Require: Language expression, RS data, a learnable label, ground-truth bounding box.

Ensure: Bounding box Bbox.

1: Initialize all weights and bias terms.
2: for epoch < epochs do
3: Perform the linguistic backbone to extract linguistic features $F_L \in R^{d_L \times d}$.
4: Perform the LAWG module to dynamically generate multi-scale language-adaptive weights $W_l$ and biases $B_i$ by Eqs. (1) to (4).
5: Apply an affine transformation to the intermediate features of the network to guide the multi-scale active perception of the visual backbone by Eq. (5).
6: Perform the language-guided visual backbone to extract visual features $F_V \in R^{h \times w \times c}$.
7: Perform the MLGA module to enhance the uniqueness of the object by Eqs. (6) to (17).
8: Perform the Transformer-based Multimodal Fusion Module to obtain $F_{token}$.
9: Input the $F_{token}$ to the localization module of LaLGA framework to predict the coordinates of bounding box.
10: Use a loss function to calculate the difference between the predicted bounding box and the ground-truth bounding box.
11: end for
12: Obtain predicted bounding box Bbox.

where $W_{q_3} \in R^{d_{q_3} \times \tilde{I}}, W_{k_3} \in R^{d_{k_3} \times \tilde{I}}, W_{v_3} \in R^{d_{v_3} \times \tilde{I}}$ are learnable weights.

$F_{right}$ can be obtained by the following equation:

$$F_{right} = CAL(\tilde{Q}_3, \tilde{K}_3, \tilde{V}_3)$$ (16)

Finally, to establish more discriminative feature for the target object, we fuse $F_{left}$ and $F_{right}$ as:

$$F_{MLGA} = F_{left} + F_{right}$$ (17)

The overall training process of the proposed LaLGA is shown in Algorithm 1.

V. EXPERIMENTS AND ANALYSIS

In this section, we present extensive experiments to validate the merits of our proposed LaLGA. Section V-A introduces the dataset used for experiments is presented. The details of the experimental setup are described in Section V-B and corresponding analyses are performed in Section V-C. In Section V-D, the experimental results of LaLGA are presented, along with a comparison with SOTA methods. Ablation experiments are performed in Section V-E.

A. Data Description

We validate the proposed LaLGA on OPT-RSVG dataset and Dior-RSVG dataset. Dior-RSVG [10] dataset was proposed by Northwestern Polytechnical University in 2023. The dataset is constructed based on the Dior dataset [12], which consists of 38320 language expressions across 17402

RS images, with a pixel size of 800 x 800. The average length of expressions is 7.47, and the size of the vocabulary is 100. This dataset depicts 20 object categories. Fig. 7 shows samples of RS images and language expressions, where the red bounding box visualizes the object’s positioning information. The names of the object categories, the number of training samples, the number of validation samples, and the number of test samples for OPT-RSVG and Dior-RSVG dataset are listed in Table III.

B. Experimental Settings

1) Evaluation Indicators: Given an RS image-query pair, the predicted bounding box is considered correct if the intersection over union (IoU) with the ground-truth bounding box is above a threshold. We follow the evaluation indicators of MGVLF [10], including $Pr@0.5$, $Pr@0.6$, $Pr@0.7$, $Pr@0.8$, $Pr@0.9$, mean IoU, and cumulative IoU (cmIoU).

2) Configurations: The experiments with the proposed method and other deep-learning methods were all implemented.
in the PyTorch platform, using a DELL EMC DSS8440 server with an Inter Xeon Silver 4210R 2.4 GHz CPU, 128 GB RAM, and four NVIDIA GeForce RTX 3090 GPU with 24 GB of VRAM. We split the dataset by randomly assigning 40%, 10%, and 50% of the expressions and their corresponding images to the training, validation, and test sets, respectively. We use the pre-trained weights of DETR [39] to initialize ResNet-50 for visual feature extraction. The linguistic backbone use the pre-training weights of the BERT model [35]. The hidden size $b$ of BERT is 768. To optimize the network, the AdamW [41] with weight decay $10^{-4}$ was selected as the initial optimizer. The learning rate is set to 1e-4. For the training stage, the mini-batch size, and the number of training epochs are set to 16 and 100, respectively.

### C. Parameter Analysis

Before conducting the comparative experiment, we analyze several hyperparameters that may affect the classification performance, including the $[CLS]$ token, and the number of multi-head attention heads $\text{head}_\text{num}$.

$[CLS]$ token: For language expressions, we typically append a learnable label token, $[CLS]$ [35], to the language backbone. This token can capture the sentence-level linguistic features, while other tokens represent word-level linguistic features. To study the impact of multi-granularity linguistic features on RSVG, we conducted related experiments as shown in Fig. 8. Experimental results show that using $[CLS]$ token is beneficial to our model.

Attention Heads: The $CAL$ module uses multiple attention heads to learn the correspondence between different represen-
Regression subspaces, where each head corresponds to an independent subspace with respect to different feature representations. Therefore, the number of attention heads used can affect the final feature representation capability of the transformer. We selected different head num from the set \{2, 4, 8, 16, 32\} to evaluate their impact on Pr@0.5. Fig. 8 show that the optimal Pr@0.5 value was achieved on OPT-RSVG datasets when head num = 8.

D. Comparisons With State-of-the-Art Methods

The comparison results of our method with the natural image VG method and the RSVG method are presented in Table IV and Table VI. For example, on OPT-RSVG dataset, our method achieves the highest score across all metrics, achieving 74.69\% in terms of Pr@0.5. Furthermore, LaLGA has surpassed the highest score achieved on DIOR-RSVG dataset, improving all performance indicators by 3.9\% to 7.02\%. Generally, in the natural image VG method, one-stage methods perform better than two-stage methods because two-stage methods [14], [15] generate object proposals using a pre-trained object detector, which does not consider the information provided by textual expression. However, one-stage methods [27], [29], [28] require manually designed, complex visual-language fusion modules to generate bounding boxes. In contrast, our method, which utilizes a transformer-based architecture, achieves simpler and more comprehensive fusion of visual-language features. We also compared our method to other transformer-based methods. TransVG [33] uses a simple stacking of transformers to design an effective VG architecture. However, its performance is limited because it ignores the differences in multimodal information. VLTVG [37] achieves an accurate VG by establishing text-conditioned discriminative features and performing multi-stage cross-modal reasoning. However, its effectiveness heavily relies on the quality of visual feature extraction. When the visual backbone is ResNet-50, the performance of VLTVG is lower than that of TransVG. MGVLF [10] processes the final output of the visual backbone at multiple scales to learn more discriminative visual representations for fusion with the text modality. However, excessive downsampling can lead to a further reduction in resolution, resulting in the loss of small objects in the RS data. Our proposed LaLGA utilizes adaptive linguistic weights to multi-scale guide the active perception of the visual backbone from RS data. Additionally, our method can further filter out interference from unrelated regions and aggregate contextual information of the target object by aligning visual-language features multiple times before fusion, enhancing the uniqueness of the target object. Fig. 9 and Fig. 10 are visualizations of the bounding boxes and attention maps predicted by LaLGA on the OPT-RSVG and DIOR-RSVG test set.

We conducted experiments on the Chinese version of OPT-RSVG, where the language encoder uses the same BERT as the English version. Taking Pr@0.5 as an example, compared with the English version, the accuracy rate of the Chinese version is reduced by 10.82\%. This is not an isolated case. The phenomenon is also observed in MGVLF. An intuitive explanation is that the semantic information learned at the word-level is relatively limited in the Chinese description of RS data.

E. Ablation Study

In this section, we conducted extensive experiments to systematically analyze the proposed LaLGA. Table VII shows the results of our ablation analysis on the three main components that constitute LaLGA, ie., Language Adaptive Weight Generation (LAWG), Multiple Language Guided Alignment (MLGA) module, and Multi-Modal Fusion Transformer (MFT). Case1 represents the result without LAWG and MLGA, which achieved only 71.56\% Pr@0.5 on the DIOR-RSVG test set. In Case2, we removed only the MLGA module, resulting in a performance drop of 2.24\%. Case3 involved removing LAWG, which caused the visual backbone to fail to actively perceive the target object, and the performance is dropped by 5.86\%. In Case4, we removed MFT, and the model achieved 78.52\% Pr@0.5. This results indicate that only
TABLE VI
COMPARISON WITH THE SOTA METHODS FOR LALGA ON THE TEST SET OF DIOR-RSVG. THE BEST PERFORMANCE IS WITH BOLD

<table>
<thead>
<tr>
<th>Methods</th>
<th>Venue</th>
<th>Visual Encoder</th>
<th>Language Encoder</th>
<th>Pr@0.5 (%)</th>
<th>Pr@0.6 (%)</th>
<th>Pr@0.7 (%)</th>
<th>Pr@0.8 (%)</th>
<th>Pr@0.9 (%)</th>
<th>meanIoU (%)</th>
<th>cmIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGVLF [10]</td>
<td>TGRS’23</td>
<td>ResNet-50</td>
<td>BERT</td>
<td>75.98</td>
<td>72.06</td>
<td>65.23</td>
<td>54.89</td>
<td>35.65</td>
<td>67.48</td>
<td>78.63</td>
</tr>
<tr>
<td>LaLGA (Ours)</td>
<td>-</td>
<td>ResNet-50</td>
<td>BERT</td>
<td><strong>82.27</strong></td>
<td><strong>77.44</strong></td>
<td><strong>72.25</strong></td>
<td><strong>60.98</strong></td>
<td><strong>39.55</strong></td>
<td><strong>72.35</strong></td>
<td><strong>85.11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( +6.29)</td>
<td>( +5.38)</td>
<td>( +7.02)</td>
<td>( +6.09)</td>
<td>( +3.90)</td>
<td>( +4.87)</td>
<td>( +6.48)</td>
</tr>
</tbody>
</table>

Fig. 10. Visualization of the final grounding results and the attention maps of our proposed LaLGA on DIOR-RSVG dataset.

TABLE VII
ABLATION STUDIES OF OUR NETWORK

<table>
<thead>
<tr>
<th>Cases</th>
<th>LAWG</th>
<th>MLGA</th>
<th>MFT</th>
<th>Pr@0.5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>71.56 (-10.71)</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>80.03 (-2.24)</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>76.41 (-5.86)</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>78.52 (-3.75)</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td><strong>82.27</strong></td>
</tr>
</tbody>
</table>

“Gray overpass in the middle” (1) (2) (3) (4) (5)

Fig. 11. Visualization of LaLGA main component ablation experiments (red boxes are ground-truth regions, and textual description indicates the target object). (1) attention map of Case1, (2) attention map of Case2, (3) attention map of Case3, (4) attention map of Case4, (5) attention map of Ours.

with early guidance from language features, the model can still achieve good performance, which reflects the significant contributions of LAWG and MLGA. Finally, the last row of Table VII shows the result of the complete LaLGA, which achieved an effect of 82.27%. The relevant visualization results are shown in Fig. 11.

To conduct a thorough analysis of the effect of LAWG, we carried out a comparative study using different stages of the visual backbone. The detailed results are shown in Table VIII. The visualization of LAWG ablation experiments are shown in Fig. 12. Our visual backbone employs ResNet-50, which consists of i-th stages, i ∈ {0, 1, 2, 3, 4}. When we added language adaptive weights only to the stage4’s output, the model performance Pr@0.5 decreases by 1.32%. In Case2, we used adaptive weights for the last two stages of the visual backbone, which resulted in a Pr@0.5 of 81.59%. We then applied LAWG to the last four stages and all stages, respectively. The results, as shown in the third and fourth rows, drop by 2.52% and 4.44%, respectively. Finally, the LaLGA, which only employs LAWG in the 3-th to 5-th stages, achieved the best Pr@0.5. The above experiments demonstrate that the use of LAWG in the early stages leads to a decline in performance. This is because the language expression tends to destroy the underlying visual features, and the VG task is to select the target object within those visual features.

In order to explore the influence of MLGA on performance, we conducted an ablation experiment on its structure. The
results are shown in Table IX and Fig. 13. MLGA consists of two branches: a left branch and a right branch, and the right branch contains two Cross Attention Layers. Case1 is the result of deleting MLGA as a whole, which has been mentioned in Table VII. In Case2 and Case4, we deleted the left and right branches respectively. The results, as shown in the second and fourth rows, dropped by 1.06% and 0.58%, respectively. In Case3, we deleted the second CAL of the right branch, and the performance dropped by 0.58%.

VI. CONCLUSION

In this paper, we propose a novel RSVG framework named LaLGA, which can guide the visual backbone to focus on the referred target. It consists of two modules: (1) LAWG is proposed to dynamically generate multi-scale language-adaptive weights, enabling the visual backbone to learn expression-relevant visual features layer by layer. (2) MLGA module is devised to aggregate the visual contextual information of the target object to enhance its uniqueness. Experimental results show that our proposed method outperforms existing natural image VG and RSVG methods, demonstrating its effectiveness and superiority. Additionally, we build a large-scale benchmark termed OPT-RSVG. OPT-RSVG dataset has more image-query pairs, a larger spatial resolution span, more balanced categories, and more abundant objects of each category. In the future, we plan to conduct more in-depth research on the characteristics of RS data, such as large image sizes and small targets, to facilitate the growth of the RS community.

REFERENCES


