

# Language-Led Visual Grounding and Future Possibilities

Zezhou Sui <sup>1,†</sup>, Mian Zhou <sup>1,\*</sup>, Zhikun Feng <sup>1</sup>, Angelos Stefanidis <sup>2</sup> and Nan Jiang <sup>3</sup>

<sup>1</sup> School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China; suizezhou@stud.tjut.edu.cn (Z.S.); kenjifeng1997@stud.tjut.edu.cn (Z.F.)

<sup>2</sup> School of AI and Advanced Computing, Xi'an Jiaotong-Liverpool University, Taicang, Suzhou 215412, China; angelos.stefanidis@xjtlu.edu.cn

<sup>3</sup> Department in Computing and Informatics, Bournemouth University, Poole BH12 5BB, UK; njiang@bournemouth.ac.uk

\* Correspondence: zhoumian@tjut.edu.cn; Tel.: +86-022-6021-6865

† These authors contributed equally to this work.

**Abstract:** In recent years, with the rapid development of computer vision technology and the popularity of intelligent hardware, as well as the increasing demand for human–machine interaction in intelligent products, visual localization technology can help machines and humans to recognize and locate objects, thereby promoting human–machine interaction and intelligent manufacturing. At the same time, human–machine interaction is constantly evolving and improving, becoming increasingly intelligent, humanized, and efficient. In this article, a new visual localization model is proposed, and a language validation module is designed to use language information as the main information to increase the model’s interactivity. In addition, we also list the future possibilities of visual localization and provide two examples to explore the application and optimization direction of visual localization and human–machine interaction technology in practical scenarios, providing reference and guidance for relevant researchers and promoting the development and application of visual localization and human–machine interaction technology.

**Keywords:** visual grounding; human–computer interaction; intelligent systems; user experience; interaction design



**Citation:** Sui, Z.; Zhou, M.; Feng, Z.; Stefanidis, A.; Jiang, N. Language-Led Visual Grounding and Future Possibilities. *Electronics* **2023**, *12*, 3142. <https://doi.org/10.3390/electronics12143142>

Academic Editor: Enzo Pasquale Scilingo

Received: 9 June 2023

Revised: 5 July 2023

Accepted: 17 July 2023

Published: 20 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Human–computer interaction (HCI) has evolved significantly over the years, transforming the way we interact with technology. One of the key challenges in HCI is bridging the gap between human perception and the machine’s understanding of the visual world. Visual grounding, a fundamental concept in computer vision, has emerged as a crucial approach to address this challenge. In this article, we will explore the concept of visual grounding and its applications in HCI, highlighting its potential to enhance our interactions with computers and devices.

One key emerging technology in computer vision [1,2] is visual grounding (VG), which refers to the process of establishing a correspondence between visual information and its associated meaning in the real world. It enables computers to recognize and interpret objects, actions, and scenes depicted in images or videos, facilitating meaningful interactions with users. By grounding visual information, computers can understand and respond to user queries, instructions, and gestures, opening up new possibilities for natural and intuitive interfaces. These technologies have many applications, including image retrieval [3], robot localization [4], image captioning [5], and so on. On the other hand, HCI technology focuses on developing interfaces that allow humans to interact with computers in a natural and intuitive way. This technology includes various modes such as voice, gesture, touch, and eye-tracking. These modes can be used alone or in combination to provide seamless and intuitive interaction between humans and computers.

In this article, we present a novel VG method within the context of HCI. Our work contributes in three main aspects:

- **Proposal of the new model:** We introduce a new model in the VG field that effectively processes visual and language features in a symmetric manner. This symmetric processing can better understand semantic information and improve the overall performance of VG systems. Additionally, we design a language-driven, multi-stage cross-modal decoder in the decoder section to iteratively locate targets based on language information, thereby increasing the model's interactivity;
- **Experimental validation:** We conducted extensive experiments to evaluate the efficacy of our proposed method. Through these experiments, we demonstrate the advantages and improvements achieved by our model on several well-established benchmarks [6–8] in the field of VG;
- **Linking between VG and HCI:** In addition to the empirical validation, we propose a connection between VG and human–computer interaction. By highlighting the synergies between these two fields, we propose feasible future applications of VG within the domain of HCI. These applications encompass various aspects around HCI, offering new possibilities for enhanced user experiences and intuitive interactions.

The structure of the paper is as follows. In Section 2, we will elaborate on previous work on VG, including the methods used. In Section 3, we introduce our model and the methods used in it. In Section 4, we will introduce the configuration, settings, and dataset used in our model experiments, the experimental setup, and the results obtained. In Section 5, we will discuss the connection between VG and HCI, as well as some potential future application scenarios.

## 2. The Previous Development in Visual Grounding

Existing VG methods can be divided into three categories: Two-stage methods [6,7,9,10], one-stage methods [11–13] and end-to-end transformer-based methods [14,15]. Both one-stage and two-stage methods treat VG as a ranking problem of detected candidate regions. One-stage methods [11–13,16] directly embed text and fuse image features to generate dense predictions, from which the one with the highest confidence is selected. Two-stage methods [10,17–24] first generate a set of object proposals and then match them with language queries to retrieve top-ranked proposals. Both methods rely on pre-detected proposals or predefined anchor box configurations for inference, and they match or fuse candidate objects with text embeddings based on region features (corresponding to predicted proposals) or point features (corresponding to dense anchor boxes). However, such feature representations may not be flexible enough to capture detailed visual concepts or context mentioned in language descriptions [14].

Although convolutional neural networks (CNNs) have achieved excellent performance in various visual tasks [25–30], the success of transformers in the fields of vision and language has attracted attention in the research community. Transformers have replaced CNNs in many visual tasks, such as image classification [2] and object detection [1,31]. The success of transformers in these areas has also driven the transformation of VG. In recent transformer-based methods, such as TransVG [14,15], a ResNet+transformer encoder and BERT are used in the backbone to extract visual and language features, respectively. The two features are projected into the same dimension using a linear projection layer, and a simple stack of transformer encoder blocks is used in the fusion stage to merge them. The output of the fusion module is directly fed into an MLP to generate the four-dimensional coordinates of the localized object. This approach achieves higher accuracy than the one- or two-stage methods on most datasets. However, the problem with TransVG [15] is that the fusion stage uses a simple stack of transformer encoder blocks, which, although effective, is too simple, resulting in suboptimal experimental accuracy. VLTVG [14] solves the problem of overly simple fusion in the fusion stage. It introduces a multi-stage cross-modal decoder with a query as the query, which calculates and outputs discriminative features based on the validation score input by the visual-language verification module and the visual

context feature and visual feature output by the language-guided context module. These features are then passed together with the language features into the multi-stage cross-modal decoder, and the final output after the last layer of features is fed into an FFN layer to generate the four-dimensional coordinates of the located object. VLTVG solves the problem of simple stacking of transformer encoder blocks in TransVG [15], but the model mainly processes visual features and neglects language feature processing, leading to a lack of semantic understanding of language features in the entire model. There are also other approaches that use transformers or attention mechanisms to tackle various types of vision and language tasks [32–35]. For instance, SCAN [33] addresses image-text matching by modeling correlations when proposing candidate bounding boxes. STVGBert [35] associates text embeddings with video frame features for video grounding. In contrast, we focus more on image-based grounding, using pixel-level modeling of visual-language correlations.

### 3. Method

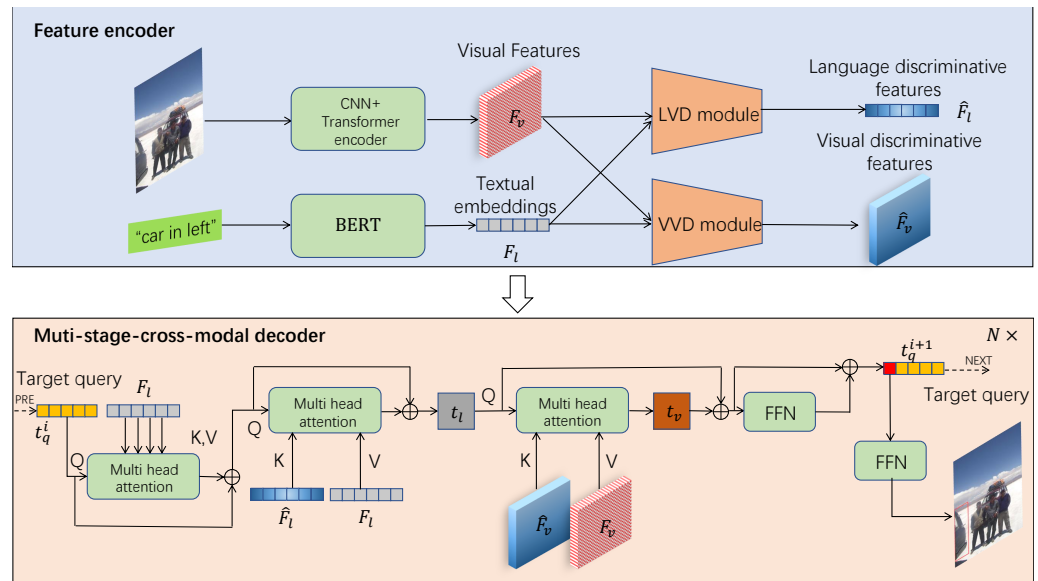
In this section, we will mainly introduce our proposed model. We will firstly present the overall structure of the model. Then we elaborate each module used in the model, including the verification module, context encoder module, and multi-stage cross-modal decoder module.

#### 3.1. The Overall Encoder and Decoder of the Model

Our proposed model follows the same approach as previous transformer-based models in directly locating objects based on their features. As shown in Figure 1, given an input image and language sentence, we first extract features separately by feeding images and sentences into two independent branches, respectively. For the image, we use a stack of transformer encoder layers on top of ResNet50 [25] to generate the 2D feature map  $F_v \in \mathbb{R}^{C \times H \times W}$ . For the language sentence, we use BERT [36] to encode it into a textual embedding sequence  $F_l \in \mathbb{R}^{C \times L}$ . Based on these two modalities, we use the visual verification discriminative (VVD) module and the language verification discriminative (LVD) module to encode them into discriminative features. In the VVD module, we employ a visual-linguistic verification module and a language-guided contextual module to encode the referenced features. In the LVD module, we use similar operations to those in the VVD module, but instead we validate the language features based on visual features, allowing the language features to focus more on the parts of visual information that are relevant to the language sentence. Finally, we apply a multi-stage cross-modal decoder to iteratively attend to the visual and language feature information encoded by the encoders for more accurate retrieval of object representations for object localization.

#### 3.2. Language Verification Discriminative (LVD) Module

The input language sentence is encoded by BERT into the feature  $F_l$ , which only contains semantic understanding, but not any prior visual knowledge. Without the prior knowledge, solely using language features to index object instances in images may lead to incorrect indexing or eventually failure to locate the object. Therefore, it is necessary to integrate visual features into the module. The LVD module integrates visual information into the language features by computing the language discriminative features.



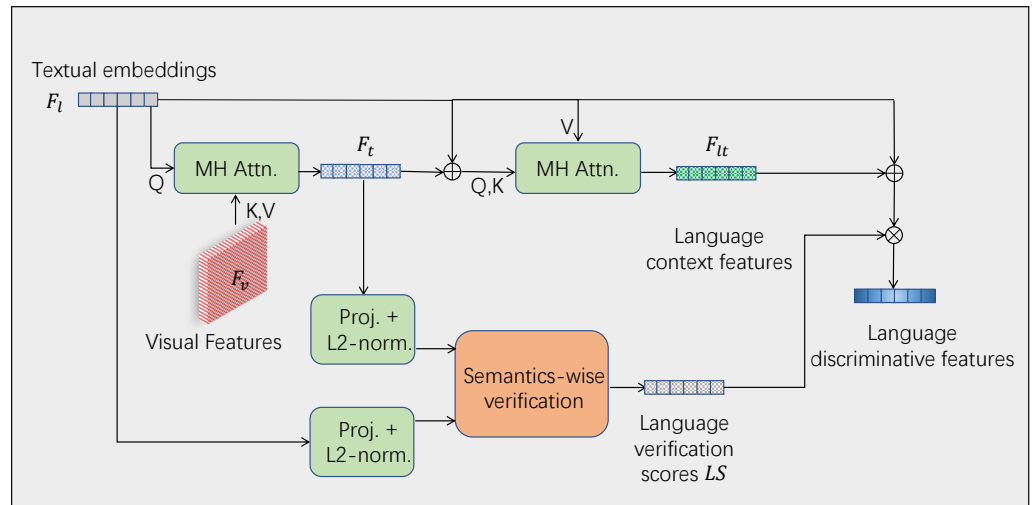
**Figure 1.** The overall encoder and decoder architecture of our proposed model is constructed as follows. Given the input image and language expression, the feature encoder first extracts visual features and textual embeddings separately. Then, the LVD module (language verification discriminative module) and VVD module (visual verification discriminative module) are used to process the features to generate discriminative features corresponding to the language and visual features, respectively. Finally, the multi-stage cross-modal decoder is utilized to iteratively infer the target location using all the generated visual and language features. The normalization layer is not indicated in the figure.

As shown in Figure 2, the LVD module is based on multi-head attention [37]. It uses the language feature  $F_l \in \mathbb{R}^{C \times L}$  as a query, and the visual feature  $F_v \in \mathbb{R}^{C \times H \times W}$  as keys and values. Through multi-head attention, relevant visual semantic information is collected for the language feature reference, resulting in  $F_t$ , which is the language representation corresponding to the visual features. Then,  $F'_l$  and  $F'_t$  are projected onto the same semantic space through a projection layer and L2-norm layer, and their semantic correlation score  $LS$  are calculated as the verification score for each spatial position  $(x, y)$ :

$$LS(x, y) = \alpha \cdot \exp\left(-\frac{(1 - F'_l(x, y)^T F'_t(x, y))^2}{2\sigma^2}\right) \quad (1)$$

where  $\alpha$  and  $\sigma$  are learnable parameters. The verification score represents the semantic correlation score between the visual feature and the corresponding language semantic feature. For each language feature, the verification score models the correlation based on the visual feature. When the visual features are multiplied with the language feature, the features obtained will naturally suppress some stimulation which is irrelevant to visual information.

However, while modeling the correlation between language and visual features, the location information or other relevant information in the language representation may also be suppressed. This would lead to an incomplete understanding of the language representation by the model. Therefore, the language features should also have visual contextual information, such as interactive relationships and positional relationships. It combines the position and other information in the language representation with visual features, enabling the model to better understand information from the language extent. This is also crucial for modeling the objects or other parts in images.



**Figure 2.** The language verification discriminative (LVD) module is based on multi-head attention, which uses multiple attention heads to capture visual information and model the correlation between visual information and language expression, ultimately outputting language discriminative features. The normalization layer and normalization layer are not indicated in the figure.

As shown in the second multi-head attention of Figure 2,  $F_t$  is added to  $F_l$ , and the sum as the query and key.  $F_l$  serves as value for multi-head attention to obtain  $F_{lt}$ . The information in  $F_{lt}$  represents the visual semantic information collected from visual features through multi-head attention and the language contextual features obtained from the interaction between visual and language features. The attention formula is as follows:

$$\begin{cases} Q = W_Q^T(F_l + F_t) \\ K = W_K^T(F_l + F_t) \\ \text{attn}_{i,j} = \text{softmax}\left(\frac{Q(i)^T(K(j) + W_K^T R(i-j))}{\sqrt{d_k}}\right) \end{cases} \quad (2)$$

where  $W_Q$  and  $W_K$  are the projection weights for the query and key, respectively,  $d_k$  is the dimension of the projection channel, and  $R(\cdot)$  is the sine positional encoding for the relative position [37].

Finally, we combine the language verification score  $LS$  with  $F_{lt}$  to obtain the language discrimination feature  $\hat{F}_l$ :

$$\hat{F}_l = (F_l + F_{lt}) \cdot LS \quad (3)$$

The LVD feature  $\hat{F}_l$  is applied in the final multi-stage cross-modal decoder.

### 3.3. Visual Verification Discriminative (VVD) Module

In the VVD module, we employ a visual-linguistic verification module and a language-guided contextual module to encode the referenced features. The visual-linguistic verification module refines the visual features to focus on those that are relevant to the language sentence. The language-guided contextual module collects information-rich visual contextual information to aid in object recognition.

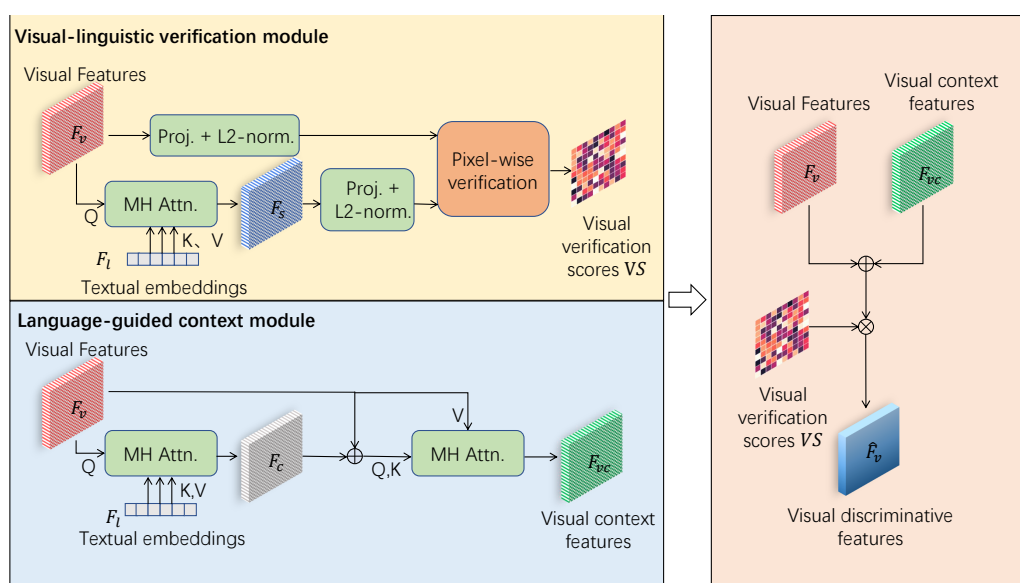
The input image is first encoded by a convolution neural network and then encoded into the visual feature map  $F_v$  by transformer encoder layers. This feature map contains the features of objects in the image, but with no prior knowledge of the language description. Retrieval without any prior knowledge is likely to be distracted by the features of other objects or regions, leading to inaccurate localization. Therefore, the visual language verification module allows the injection of language information into the visual features, enabling the visual features to possess rich prior language knowledge.

As shown in Figure 3, the visual-linguistic verification module (VLVM) is based on multi-head attention. The visual feature map  $F_v \in \mathbb{R}^{C \times H \times W}$  is used as the query and the

textual embedding  $F_l \in \mathbb{R}^{C \times L}$  is used as the key and value. Though multi-head attention, relevant semantic information is collected for some visual features. Then, the projected feature maps  $F'_v$  and  $F'_s$  are obtained by projection and L2-norm layers into the same semantic space. Their semantic correlation score is calculated as the verification score for each spatial location  $(x, y)$

$$S(x, y) = \alpha \cdot \exp\left(-\frac{(1 - F'_v(x, y)^T F'_s(x, y))^2}{2\sigma^2}\right) \tag{4}$$

where  $\alpha$  and  $\sigma$  are learnable parameters. These verification scores measure the correlation between each visual feature and textual embedding. When the embedding is multiplied with the visual feature, the resulting feature naturally suppresses the textual parts that are irrelevant.



**Figure 3.** Visual verification discriminative (VVD) module. The VVD module is also based on multi-head attention and consists of two sub-modules: the visual-linguistic verification module and the language-guided context module. The visual-linguistic verification module models the correlation between the language information and visual information pixel-by-pixel through multi-head attention to obtain visual verification scores. The language-guided context module uses multi-head attention to obtain contextual information of the language features and visual context features. Finally, the visual verification scores, visual context features, and visual features are combined to calculate the visual discriminative features.

While using the VLD module to model the correlation, the visual features should also contain the visual context, such as interactive relationships and positional relationships, which are crucial for modeling the target object and other parts. The language-guided context module effectively combines the interaction, location, and other information about objects in the language expression with visual features, forming the visual features with contextual information.

As shown in Figure 3, the language-guided context module is also based on multi-head attention.  $F_v$  is used as the query, and  $F_l$  is used as the key and value to obtain  $F_c$ , which contains semantic information and is the corresponding language representation of the visual features. Then,  $F_c$  is added with  $F_v$  as the query and key, and  $F_v$  is used as the value to again obtain the visual context feature  $F_{vc}$ , which contains the visual representation of

the contextual information in the language description. The attention formula for  $F_{vc}$  is as follows:

$$\begin{cases} Q = W_Q^T(F_v + F_c) \\ K = W_K^T(F_v + F_c) \\ \text{attn}_{i,j} = \text{softmax}\left(\frac{Q(i)^T(K(j)+W_K^TR(i-j))}{\sqrt{d_k}}\right) \end{cases} \quad (5)$$

where  $W_Q$  and  $W_K$  are the projecting weights for the query and key, respectively,  $d_k$  is the dimension of the projecting channel, and  $R(\cdot)$  is the sine encoding of the relative position.

To establish more discriminative features for the target object, we use the same approach to fuse the contextual features  $F_{vc}$  and the visual verification scores  $S$  with the visual features  $F_v$ :

$$\hat{F}_v = (F_v + F_{vc}) \cdot S \quad (6)$$

The generated visual discriminative feature  $\hat{F}_v$  is also applied in the final multi-stage cross-modal decoder.

### 3.4. Multi-Stage-Cross-Modal Decoder

We use a multi-stage cross-modal decoder that takes language features as queries to perform the final object localization task by leveraging the established visual feature map and text embedding. The decoder can repeatedly process visual and language information, thereby distinguishing the target object from other objects.

In Figure 1, we illustrate the architecture of the multi-stage cross-modal decoder that we use language features as queries to perform final target localization task with the established visual feature maps and text embedding. The decoder is composed of  $N$  stages, each of which consists of the same network architecture (weights are not shared) for iterative cross-modal reasoning. In the first stage, we use a learnable target query  $t_q^1 \in \mathbb{R}^{c \times x}$  as the initial representation of the target object, where  $x$  is set to five. The target query is fed into the decoder to extract visual features based on the language expression and update its feature representation to  $t_q^i (1 \leq i \leq N)$  at the beginning of each subsequent stage. The feature updating process for each decoder is shown in Figure 1. Specifically, in the  $i$ -th stage, the target query  $t_q^i$  is used as a query and fed into the first multi-head attention module, where language features are used as keys and values for multi-head attention, aiming to imbue the target query with language information that can query the relevant parts in both language and visual information encoded by the text embedding. After obtaining the language information, it is fed into the second multi-head attention module, where the previously computed language discrimination feature  $\hat{F}_l$  is used as the keys and the language feature  $F_l$  is used as the values for multi-head attention to compute the relevance with the previous language discrimination feature, resulting in  $t_l \in \mathbb{R}^{c \times 5}$ , which contains the semantic information of the language expression corresponding to the visual region of interest. Then, in the third multi-head attention module, we use  $t_l$  as the query,  $\hat{F}_v$  as the keys, and  $F_v$  as the values to compute the relevance with the previous visual discrimination feature and collect the interested region from the visual feature map  $F_v$  based on the semantic description collected in  $t_l$ , generating the collected visual feature  $t_v \in \mathbb{R}^{c \times 5}$  for the referenced object. Finally,  $t_v$  is used to update the target query  $t_q^i$ :

$$\begin{cases} t'_q = LN(t_q^i + t_v) \\ t_q^{i+1} = LN(t'_q + FFN(t'_q)) \end{cases} \quad (7)$$

where  $LN(\cdot)$  stands for layer normalization, and  $FFN(\cdot)$  is a feed-forward neural network consisting of two linear projection layers with a ReLU activation layer. The updated  $t_q^{i+1}$  is then fed into the next level decoder for iterative cross-modal inference and feature representation updates.

We used five target queries to represent the target object, but only the first target query was used for prediction and back-propagation updates during the final prediction

process. This way, other interfering information is concentrated in the last four target queries, making it simple and efficient, and eliminating the need to use a classification head to detect whether the feature representation in the target query is a representation of the query object.

Based on this multi-level decoder, using target queries can focus on different descriptions of the referring expressions, enabling us to collect more complete features of the target objects. Furthermore, we can use the collected features to further refine and improve the target queries  $t_q^i$ , forming a more accurate representation of the target objects.

Finally, we predict the bounding box of the referenced object by attaching a three-layer MLP with a ReLU activation function to the output of each stage's target query, and supervise all predicted bounding boxes equally to facilitate multi-level decoder training.

## 4. Experiments

In this section, we first talk about how to implement the proposed methods, and display the results subsequently.

### 4.1. Implementation Detail

We set the size of the input image to  $640 \times 640$  and the maximum length of the language expression to 40. During inference, we dynamically adjust the size of the input image according to the input image, so that the longer edge equals 640 and the shorter edge is padded to 640. At the beginning and end of the language expression, we added [CLS] and [SEP] tokens, respectively, and then processed them using BERT [36]. We perform data augmentation during training, following previous work [12,13,15,16].

In the visual feature extraction branch, we use ResNet50 as our CNN backbone, followed by six transformer encoder layers of the visual feature extraction branch, which we initialize using the corresponding weights of the DETR model [1]. In the text embedding extraction branch, we initialize the corresponding weights using BERT [36].

During the training process, we used the AdamW optimizer [38] to train our model with a batch size of eight. We trained for a total of 90 epochs, and in the first 10 epochs, we froze the weights of the feature extraction branch (i.e., the CNN+transformer encoder layers and BERT). This allowed our model to be trained in a more stable manner.

We set the initial learning rate of the network to  $10^{-4}$ , the initial learning rate of the feature extraction layers to  $10^{-5}$ , and decay the learning rate by a factor of 10 after 60 epochs of training.

We use the same loss function as previously used in transformer-based methods. Since our network directly regresses to the coordinates of bounding boxes, we avoid positive/negative sample assignment and directly use the predicted bounding boxes to calculate the loss:

$$L = \sum_{i=1}^N \lambda_{\text{giou}} L_{\text{giou}}(b, \hat{b}^i) + \lambda_{L1} L_{L1}(b, \hat{b}^i) \quad (8)$$

where  $b$  represents the ground-truth bounding box, and  $\{\hat{b}^i\}_{i=1}^N$  represents the predicted bounding boxes from stage 1 to stage  $N$ .  $\lambda_{\text{giou}}$  and  $\lambda_{L1}$  denote the GIoU loss [39] and  $L1$  loss, respectively, and  $\lambda_{\text{giou}}$  and  $\lambda_{L1}$  are hyper-parameters that balance the two losses, which are set to 2 and 5, respectively.

We follow the evaluation metrics used in previous works [15,16]. Given an image and a language expression, a predicted bounding box is considered correct if its intersection over union (IoU) with the ground-truth bounding box is greater than 0.5.

### 4.2. Results

In Table 1, we report the performance comparison of our method with other state-of-the-art methods on three popular benchmark visual localization datasets: RefCOCO [6], RefCOCO+ [6], and RefCOCOg [7]. Our method outperforms other methods on some of the datasets. Table 2 also shows the performance of our method on the test set of



ReferItGame [8]. Our method is significantly better than the one- and two-stage methods and also leads in transformer-based methods. As the ReferItGame dataset is annotated through the refer it game, which has strong interactivity in terms of data formatting and annotation, the performance comparison on the ReferItGame dataset indicates that our model is superior to other methods in terms of interaction.

**Table 1.** Comparison of our method with other state-of-the-art methods on RefCOCO [6], RefCOCO+ [6], and RefCOCOg [7]. For the data not given in the model paper, we use “-” instead.

Models	Venue	BackBone	RefCOCO			RefCOCO+			RefCOCOg		
			Val	TestA	TestB	Val	TestA	TestB	Val-g	Val-u	Test-u
Two-stage Models											
CMN [18]	CVPR'17	VGG16	-	71.03	65.77	-	54.32	47.76	57.47	-	-
VC [23]	CVPR'18	VGG16	-	73.33	67.44	-	58.40	53.18	62.30	-	-
ParalAttn [24]	CVPR'18	VGG16	-	75.31	65.52	-	61.34	50.86	58.03	-	-
MAttNet [22]	CVPR'18	ResNet-101	76.65	81.14	69.99	65.33	71.62	56.02	-	66.58	67.27
LGRANs [20]	CVPR'19	VGG16	-	76.60	66.40	-	64.00	53.40	61.78	-	-
DGA [21]	ICCV'19	VGG16	-	78.42	65.53	-	69.07	51.99	-	-	63.28
RvG-Tree [17]	TPAMI'19	ResNet-101	75.06	78.61	69.85	63.51	67.45	56.66	-	66.95	66.51
NMTree [19]	ICCV'19	ResNet-101	76.41	81.21	70.09	66.46	72.02	57.52	64.62	65.87	66.44
Ref-NMS [40]	AAAI'21	ResNet-101	80.70	84.00	76.04	68.25	73.68	59.42	-	70.55	70.62
One-stage Models											
SSG [11]	arXiv'18	DarkNet-53	-	76.51	67.50	-	62.14	49.27	47.47	58.80	-
FAOA [13]	ICCV'19	DarkNet-53	72.54	74.35	68.50	56.81	60.23	49.60	56.12	61.33	60.36
RCCF [12]	CVPR'20	DLA-34	-	81.06	71.85	-	70.35	56.32	-	-	65.73
ReSC-Large [16]	ECCV'20	DarkNet-53	77.63	80.45	72.30	63.59	68.36	56.81	63.12	67.30	67.20
LBYL-Net [41]	CVPR'21	DarkNet-53	79.67	82.91	74.15	68.64	73.38	59.49	62.70	-	-
Transformer-based Models											
TransVG [15]	ICCV'21	ResNet-50	80.32	82.67	78.12	63.50	68.15	55.63	66.56	67.66	67.44
TransVG [15]	ICCV'21	ResNet-101	81.02	82.72	78.35	64.82	70.70	56.94	67.02	68.67	67.73
ours		ResNet-50	<b>84.00</b>	<b>87.64</b>	<b>79.31</b>	<b>72.67</b>	<b>78.17</b>	<b>63.51</b>	<b>71.71</b>	<b>74.63</b>	<b>73.36</b>

**Table 2.** Comparison with the state-of-the-art methods on the test sets of ReferItGame [8].

Models	BackBone	ReferItGame Test
Two-stage models		
CMN [18]	VGG16	28.33
VC [23]	VGG16	31.13
MAttNet [22]	ResNet-101	29.04
Similarity Net [10]	ResNet-101	34.54
CITE [42]	ResNet-101	35.07
DDPN [43]	ResNet-101	63.00
One-stage models		
SSG [11]	DarkNet-53	54.24
ZSGNet [44]	ResNet-50	58.63
FAOA [13]	DarkNet-53	60.67
RCCF [12]	DLA-34	63.79
ReSC-Large [16]	DarkNet-53	64.60
LBYL-Net [41]	DarkNet-53	67.47
Transformer-based models		
TransVG [15]	ResNet-50	69.76
TransVG [15]	ResNet-101	70.73
VLTVG [14]	ResNet-50	71.60
VLTVG [14]	ResNet-101	71.84
ours	ResNet-50	<b>72.45</b>

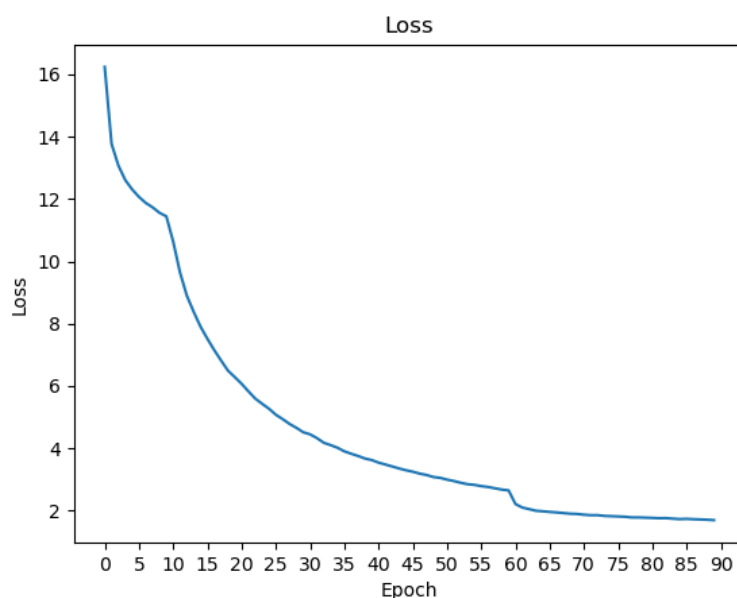
Table 3 shows the ablation experiments conducted to verify the effectiveness of our proposed method. In the first row, we did not use our improved module and achieved

an accuracy of 71.62%. In the second row, we modified the number of queries in the multi-stage decoder to five, resulting in an accuracy of 71.81%. In the last row, we used the complete model and achieved the best performance of 72.45% among all ablation variants.

**Table 3.** The ablation studies of the proposed components in our network. We evaluated the accuracy of the visual grounding, and reported the model size and computational complexity.

Query5	LVD	#Params	Gflops	Acc (%)
		152.18M	41.79	71.62
✓		152.19M	41.84	71.81
✓	✓	152.31M	42.13	72.45

Figure 4 shows the loss curve of our training process. We unfroze the parameters of the network layers for image and text feature extraction after the 10th epoch to involve them in the training process. We also decreased the learning rate by a factor of ten at the 60th epoch. These modifications can be clearly observed in the figure, demonstrating their effectiveness.



**Figure 4.** The loss curve of the model training.

Figure 5 showcases the output results of our model, revealing a high degree of accuracy in the majority of the localization outcomes. The predicted results of our model closely match the ground-truths, indicating its proficiency in localizing objects within images. However, we also present instances of localization failures in Figure 5, accompanying the language expressions associated with the respective images. In these particular examples, the model tends to mislocate the objects, potentially due to a limited understanding of the language expressions relevant to the given image. As a result, the model heavily relies on visual cues alone for localization, leading to inaccuracies.

For example, in the second-to-last image in Figure 5, the requested localization phrase is “girl with mic”, but the model’s localization result does not match the ground-truth. Instead, it localizes to the girl on the left. This could be due to the model not encountering the term “mic” during training, leading to a lack of understanding of the complete meaning of the language expression and resulting in localization to the most prominent girl.



**Figure 5.** The comparison between our model's localization results and the ground-truth bounding boxes is shown. The green boxes represent the ground-truths, while the red boxes indicate the predicted bounding boxes by our model. Additionally, we present three examples of localization errors, with the language expressions associated with the erroneous examples marked with an asterisk (\*).

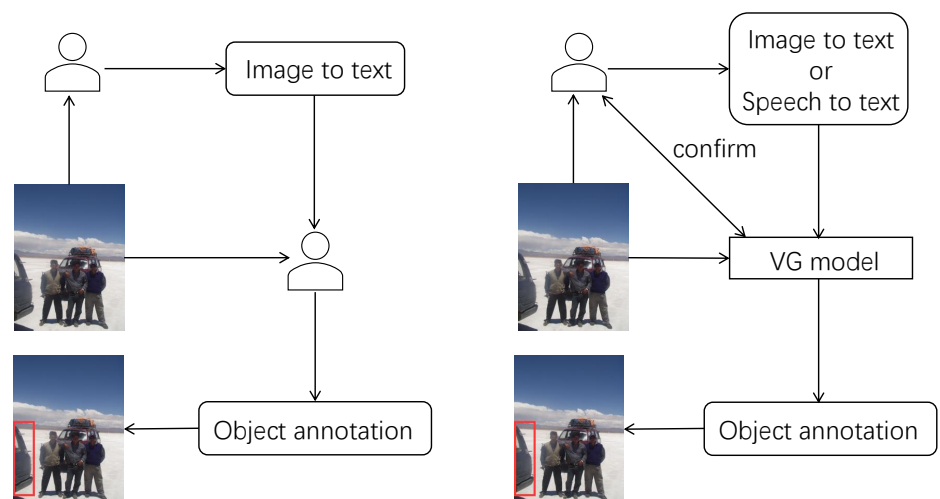
## 5. Feasible Work of VG to Enhance HCI

VG is a field closely related to HCI. Using VG to enhance HCI is a future research area because VG can enable computers to better understand the relationship between images and language, making it easier for them to understand human intentions during the interaction process, resulting in simpler and more efficient HCI. Here we present potential application scenarios that VG could be applied to in HCI.

### 5.1. Image Dataset Annotation

There is a growing trend towards the utilization of larger visual and multi-modal models in recent years. However, training these models on large datasets necessitates a significant amount of data to facilitate effective learning of the given task. Unfortunately, the process of annotating image data is both laborious and costly. One potential solution to address this challenge is to integrate VG as a fundamental component with other algorithms, thereby creating an automatic image dataset annotation tool. As shown in Figure 6a, traditional dataset annotation consists of two main parts. Firstly, a dataset organizer/manager generator describes the objects to be annotated in the images in terms

of language expressions. Then, workers annotate/label the objects in the images according to the language expressions. This purely manual approach naturally brings with it several major problems. Firstly, it requires hiring a large amount of personnel to annotate the data, when the dataset becomes large. Secondly, there is no specific standard for workers to manually annotate data, the position and size of objects are annotated at workers' will. This leads to inconsistency when different workers have different annotation styles. In deep learning, the high-quality annotation of a dataset is crucial for accurate classification, detection, and segmentation, while an inconsistent dataset degrades the performance of deep learning systems. Lastly, employing humans to perform such repetitive and tedious tasks is not in line with the principles of HCI.



(a) Traditional methods for annotating datasets      (b) Improved method using VG model

**Figure 6.** The previous dataset annotation process and the process of dataset annotation using VG technology are shown in the flowchart. Figure (a) represents the previous dataset annotation process, while Figure (b) represents the process of dataset annotation using VG technology.

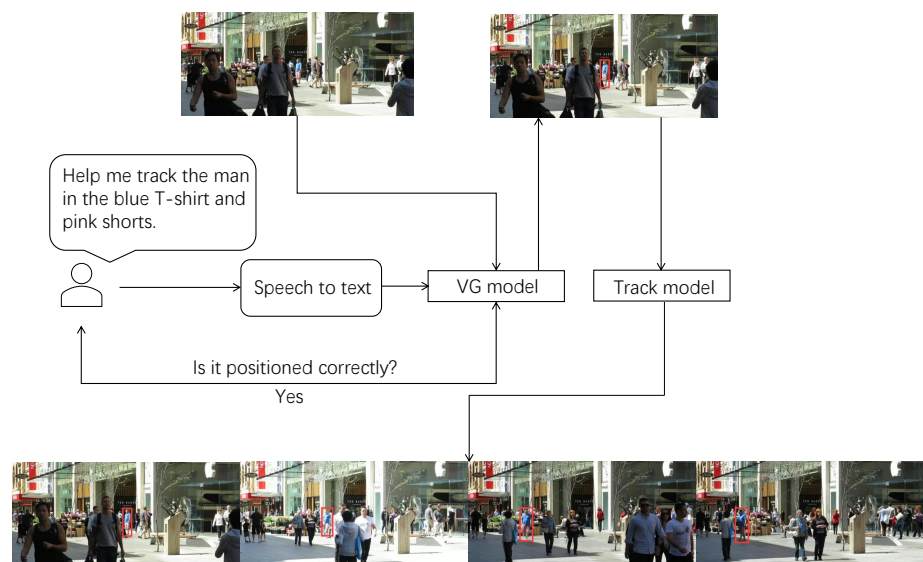
To address the limitations of traditional data annotation methods mentioned above, we have made improvements by incorporating VG technology as a replacement for extensive manual operations. We present a process flow diagram illustrating the data annotation procedure using VG technology, as shown in Figure 6b. Our approach also consists of two main parts: the generation of language expressions as the first step, localization annotation using VG technology as the second step. The first step involves manually describing the objects that need to be annotated to generate language expressions. We only rely on human input in this step because the selection and description of language expressions are subjective tasks that are challenging for current technology to replace with automated processes. When generating language expressions, there are two options for the input: direct text input and voice input using devices with language recognition capabilities. After generating the language expressions, we feed the images and language expressions into our VG model to perform object localization annotation. In this step, we utilize VG technology to replace manual annotation because it involves repetitive labeling tasks that do not require strong subjectivity. Additionally, using machines for annotation leads to faster processing, ensures a consistent labeling style, and improves the overall quality of the dataset.

After the model completes the annotation process, we interact with the user to verify the accuracy of the annotations. If the annotations fail, we either reposition and re-annotate them or ask the organizer to provide a more precise language expression for the model to perform the repositioning and annotation.

### 5.2. Interactive Object Tracking

VG can be integrated with other technologies to enable real-time object detection and tracking, which has numerous applications in HCI. For example, cameras can use this technology to track people or robots. In the paradigm of tracking by learning, when tracking pedestrians in a crowded scene, initial object detectors may present multiple objects as tracking candidates at the start of tracking. A human operator may then select an object of interest to track. However, it can be challenging for the human operator to communicate to the computer which object to track, and it may even be impossible to switch to a different object during the tracking process. Furthermore, there is currently no provision for interactive operations during target selection.

We propose an interactive object-tracking system to overcome the limitations of traditional object-tracking approaches. As illustrated in Figure 7, when a human operator encounters a scene with multiple pedestrians and decides on a target to track, an oral instruction is released by the operator and transformed into text using a speech-to-text module, such as a voice recognition device. The first frame image and the language expression are then input into a VG model for target localization. Once the VG model completes the localization, the operator is asked to verify whether it is correct. If the positioning is not accurate, the user needs to provide more precise language descriptions, such as the interactive relationship or location of the object in addition to the characteristics of the object itself, because the model has the ability to understand object interactions and positions. The best wording should include the characteristics of the positioned object and its location relationship, etc. For example, “the car on the left” is better than just “car”. Once the target localization is completed, the VG-based results and the remaining video frames are fed into a tracking model. The tracking model utilizes the VG localization results to track the target in the subsequent video frames.



**Figure 7.** Interactive object-tracking process diagram. The images in the diagram represent different frames of a video to illustrate the tracking process.

The interactive object-tracking system we have designed is highly adaptable and can be used in various working environments. For instance, if the tracking task involves recognizing a single object, the system can easily be adapted by replacing the input expression with a specific expression for that object. This flexibility allows the system to be customized for different tracking tasks and makes it more versatile and widely applicable.

## 6. Conclusions

We proposed a transformer-based VG framework that establishes discriminative features for both language and vision, and performs iterative cross-modal reasoning for accurate object localization. Our language discriminative feature module and vision discriminative feature module enable the collection of semantically related information from language and vision, respectively, which is used for localization in a multi-stage cross-modal decoder. Extensive experiments on public datasets demonstrate the superiority of our method. Finally, we propose some solutions for enhancing HCI in VG tasks, aiming to promote the development and application of VG and HCI technology.

Although VG has made significant advancements, there are still several challenges that need to be addressed. Some of the key challenges include dealing with ambiguity, incorporating temporal information for dynamic scenes, and achieving robustness to changes in lighting, viewpoints, or occlusions. Additionally, ethical considerations, such as privacy, bias, and fairness, must be carefully addressed when deploying VG technologies in HCI. To promote responsible and inclusive HCI, it is essential to ensure transparency, accountability, and user consent. Handling these challenges will be critical in realizing the full potential of VG and ensuring that it benefits society as a whole.

**Author Contributions:** Conceptualization, Z.S. and M.Z.; methodology, Z.S.; software, Z.S.; validation, Z.S. and M.Z.; formal analysis, Z.S.; investigation, N.J.; resources, A.S.; data curation, Z.F.; writing—original draft preparation, Z.S.; writing—review and editing, M.Z.; visualization, Z.S.; supervision, M.Z.; project administration, M.Z.; funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (61872270).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** An implementation example is available at <https://github.com/fest007/LVVVG>, accessed on 8 June 2023.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

VG	Visual grounding
HCI	Human computer interaction
CNN	Convolution neural network
LD	Linear dichroism

## References

1. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 213–229.
2. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
3. Datta, R.; Joshi, D.; Li, J.; Wang, J.Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv. (Csur)* **2008**, *40*, 1–60. [[CrossRef](#)]
4. Betke, M.; Gurvits, L. Mobile robot localization using landmarks. *IEEE Trans. Robot. Autom.* **1997**, *13*, 251–263. [[CrossRef](#)]
5. You, Q.; Jin, H.; Wang, Z.; Fang, C.; Luo, J. Image captioning with semantic attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4651–4659.
6. Yu, L.; Poirson, P.; Yang, S.; Berg, A.C.; Berg, T.L. Modeling context in referring expressions. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 69–85.
7. Mao, J.; Huang, J.; Toshev, A.; Camburu, O.; Yuille, A.L.; Murphy, K. Generation and Comprehension of Unambiguous Object Descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

8. Kazemzadeh, S.; Ordonez, V.; Matten, M.; Berg, T. Referitgame: Referring to objects in photographs of natural scenes. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 787–798.
9. Nagaraja, V.K.; Morariu, V.I.; Davis, L.S. Modeling Context Between Objects for Referring Expression Understanding. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 792–807.
10. Wang, L.; Li, Y.; Huang, J.; Lazebnik, S. Learning Two-Branch Neural Networks for Image-Text Matching Tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 394–407. [[CrossRef](#)] [[PubMed](#)]
11. Chen, X.; Ma, L.; Chen, J.; Jie, Z.; Liu, W.; Luo, J. Real-time referring expression comprehension by single-stage grounding network. *arXiv* **2018**, arXiv:1812.03426.
12. Liao, Y.; Liu, S.; Li, G.; Wang, F.; Chen, Y.; Qian, C.; Li, B. A real-time cross-modality correlation filtering method for referring expression comprehension. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10880–10889.
13. Yang, Z.; Gong, B.; Wang, L.; Huang, W.; Yu, D.; Luo, J. A fast and accurate one-stage approach to visual grounding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 4683–4693.
14. Yang, L.; Xu, Y.; Yuan, C.; Liu, W.; Li, B.; Hu, W. Improving visual grounding with visual-linguistic verification and iterative reasoning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9499–9508.
15. Deng, J.; Yang, Z.; Chen, T.; Zhou, W.; Li, H. Transvg: End-to-end visual grounding with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1769–1779.
16. Yang, Z.; Chen, T.; Wang, L.; Luo, J. Improving one-stage visual grounding by recursive sub-query construction. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 387–404.
17. Hong, R.; Liu, D.; Mo, X.; He, X.; Zhang, H. Learning to compose and reason with language tree structures for visual grounding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *44*, 684–696. [[CrossRef](#)] [[PubMed](#)]
18. Hu, R.; Rohrbach, M.; Andreas, J.; Darrell, T.; Saenko, K. Modeling relationships in referential expressions with compositional modular networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1115–1124.
19. Liu, D.; Zhang, H.; Wu, F.; Zha, Z.J. Learning to assemble neural module tree networks for visual grounding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4673–4682.
20. Wang, P.; Wu, Q.; Cao, J.; Shen, C.; Gao, L.; Hengel, A.V.D. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1960–1968.
21. Yang, S.; Li, G.; Yu, Y. Dynamic graph attention for referring expression comprehension. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4644–4653.
22. Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; Berg, T.L. Mattnet: Modular attention network for referring expression comprehension. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1307–1315.
23. Zhang, H.; Niu, Y.; Chang, S.F. Grounding referring expressions in images by variational context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4158–4166.
24. Zhuang, B.; Wu, Q.; Shen, C.; Reid, I.; Van Den Hengel, A. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4252–4261.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
26. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)] [[PubMed](#)]
27. Xu, Y.; Huang, Z.; Lin, K.Y.; Zhu, X.; Shi, J.; Bao, H.; Zhang, G.; Li, H. Selfvoxelo: Self-supervised lidar odometry with voxel-based deep neural networks. In Proceedings of the Conference on Robot Learning. PMLR, London, UK, 8–11 November 2021; pp. 115–125.
28. Xu, Y.; Lin, J.; Shi, J.; Zhang, G.; Wang, X.; Li, H. Robust self-supervised lidar odometry via representative structure discovery and 3d inherent error modeling. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1651–1658. [[CrossRef](#)]
29. Xu, Y.; Lin, K.Y.; Zhang, G.; Wang, X.; Li, H. RNNPose: Recurrent 6-DoF object pose refinement with robust correspondence field estimation and pose optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14880–14890.
30. Xu, Y.; Zhu, X.; Shi, J.; Zhang, G.; Bao, H.; Li, H. Depth completion from sparse lidar data with depth-normal constraints. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2811–2820.

31. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
32. Chen, Y.; Gong, S.; Bazzani, L. Image search with text feedback by visiolinguistic attention learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3001–3011.
33. Lee, K.H.; Chen, X.; Hua, G.; Hu, H.; He, X. Stacked cross attention for image-text matching. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 201–216.
34. Lu, J.; Batra, D.; Parikh, D.; Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 13–23.
35. Su, R.; Yu, Q.; Xu, D. Stvgbert: A visual-linguistic transformer based framework for spatio-temporal video grounding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1533–1542.
36. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
38. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
39. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
40. Chen, L.; Ma, W.; Xiao, J.; Zhang, H.; Chang, S.F. Ref-nms: Breaking proposal bottlenecks in two-stage referring expression grounding. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 1036–1044.
41. Huang, B.; Lian, D.; Luo, W.; Gao, S. Look before you leap: Learning landmark features for one-stage visual grounding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16888–16897.
42. Plummer, B.A.; Kordas, P.; Kiapour, M.H.; Zheng, S.; Piramuthu, R.; Lazebnik, S. Conditional image-text embedding networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 249–264.
43. Yu, Z.; Yu, J.; Xiang, C.; Zhao, Z.; Tian, Q.; Tao, D. Rethinking diversified and discriminative proposal generation for visual grounding. *arXiv* **2018**, arXiv:1805.03508.
44. Sadhu, A.; Chen, K.; Nevatia, R. Zero-shot grounding of objects from natural language queries. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4694–4703.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.