SegVG: Transferring Object Bounding Box to Segmentation for Visual Grounding

Weitai Kang[1], Gaowen Liu[2], Mubarak Shah[3], and Yan Yan[1]

[1] Illinois Institute of Technology
[2] Cisco Research
[3] University of Central Florida

Abstract. Different from Object Detection, Visual Grounding deals with detecting a bounding box for each text-image pair. This one box for each text-image data provides sparse supervision signals. Although previous works achieve impressive results, their passive utilization of annotation, i.e. the sole use of the box annotation as regression ground truth, results in a suboptimal performance. In this paper, we present **SegVG**, a novel method transfers the box-level annotation as **Seg**mentation signals to provide an additional pixel-level supervision for **V**isual **G**rounding. Specifically, we propose the Multi-layer Multi-task Encoder-Decoder as the target grounding stage, where we learn a regression query and multiple segmentation queries to ground the target by regression and segmentation of the box in each decoding layer, respectively. This approach allows us to iteratively exploit the annotation as signals for both box-level regression and pixel-level segmentation. Moreover, as the backbones are typically initialized by pretrained parameters learned from unimodal tasks and the queries for both regression and segmentation are static learnable embeddings, a domain discrepancy remains among these three types of features, which impairs subsequent target grounding. To mitigate this discrepancy, we introduce the Triple Alignment module, where the query, text, and vision tokens are triangularly updated to share the same space by triple attention mechanism. Extensive experiments on five widely used datasets validate our state-of-the-art (SOTA) performance. Code is available at `https://github.com/WeitaiKang/SegVG`.

Keywords: Visual Grounding · Detection · Transformer

1 Introduction

Visual grounding [14,17,26,30,52] aims to localize a target object within an image based on a free-form natural language text expression. It is particularly important for numerous downstream multimodal reasoning systems, such as visual question answering [10,35,41] and image captioning [1,4,50]. Previous works can be broadly categorized into three distinct groups: two-stage methods [3,44,45,51], one-stage methods [47,48], and transformer-based ones [6,15,16,31,43,49]. Both

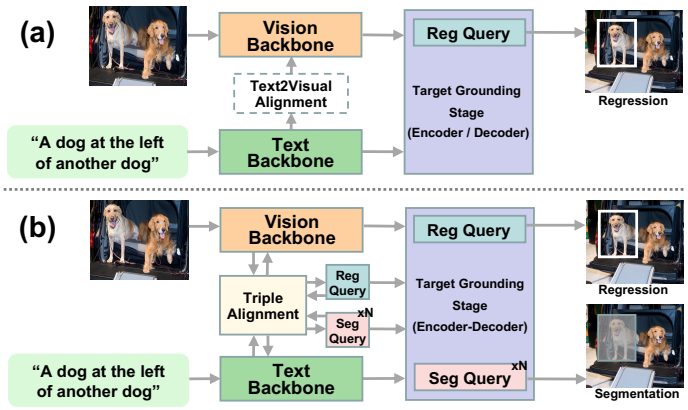

Fig. 1: The comparison of visual grounding frameworks. The block with a dashed border indicates that the module may not necessarily exist. (**a**) Previous baseline method consists of two backbones and additional transformer layers for target grounding, where a regression query is supervised to regress the box. Current SOTA methods further employ a text-to-visual module to align the visual features with text features. (**b**) Our method incorporates segmentation queries, which utilizes the box annotation at the pixel-level to segment the target. Additionally, we propose the Triple Alignment module to eliminate the domain discrepancy of the query, text, and vision features.

two-stage and one-stage approaches use convolutional neural networks for candidate proposals and the selection of the best-matching candidate. Nonetheless, these approaches rely on intricate modules that employ manually-crafted techniques for performing language inference and multi-modal integration.

Inspired by the success of the transformer [7, 8], TransVG [6] proposes a transformer-based pipeline. As shown in Fig. 1.(a), this pipeline extracts vision and text features via DETR [2] and BERT [7], respectively. To ground the target, they use the transformer encoder to fuse multimodal features along with a learnable regression query and decode the query through an MLP. To enhance the final target grounding stage, subsequent studies continue with some text-to-visual modules in the early stage to modulate the vision features to align with the text features. For example, QRNet [49] proposes a query-modulated method for extracting language-aware vision features within the vision backbone. VLTVG [43] introduces a verification map to activate the vision features to align with the text features before multimodal fusion.

Despite their advancements, the suboptimal annotation utilization, i.e., only using the box annotation as a regression annotation, limits their performance. As discussed in [37], Visual Grounding presents unique challenges compared to Object Detection due to its sparse supervision signals. Specifically, it provides only one box label for each text-image pair, while necessitatng detection within a multimodal setting. Therefore, it is essential to fully exploit the box annotation, by treating it as a segmentation mask (pixels within the bounding box are assigned a value of 1, while pixels outside the bounding box are assigned as 0).

In this paper, we introduce SegVG (see Fig. 1.(b)), a novel method that leverages the pixel-level details within the box annotation as segmentation signals to offer additional fine-grained supervision for Visual Grounding. Specifically, we propose the Multi-layer Multi-task Encoder-Decoder as the target grounding stage, where we learn a regression query and multiple segmentation queries to ground the target by regression and segmentation of the box in each decoding layer, respectively. The confidence score derived from the segmentation can further serve as a Focal Loss [21] scaling factor to adaptively emphasize the other losses of challenging training samples. This approach allows us to iteratively exploit the annotation as signals for both box-level regression and pixel-level segmentation. Furthermore, the initial parameters for model backbones, typically derived from pretrained unimodal tasks, along with data-agnostic static embeddings used as queries for decoding, result in a domain discrepancy among different sources of feature, affecting the effectiveness of target grounding. To tackle this problem, we present the Triple Alignment module, where we harmonize the domain of query, text, and vision features by implementing a triangular update process through a triple attention mechanism. As a result, we ensure that all features adapt and integrate within the same multimodal space, thereby enhancing subsequent target grounding. Our contributions are as follows:

- We propose the Multi-layer Multi-task Encoder-Decoder to maximize the utilization of the box annotation, which introduces an additional segmentation format for pixel-level supervision in Visual Grounding.
- To eliminate the domain discrepancy among the query, text, and vision, we introduce the Triple Alignment to update these three types of features into a sharing domain, which facilitates the subsequent target grounding.
- We conduct extensive experiments on five widely used datasets to show the performance superiority of our proposed methods compared with previous state-of-the-art methods and further investigate the reliability benefits derived from the segmentation output in real applications.
- We will release code and checkpoints for future research development.

2 Related Work

Visual grounding methods can be roughly classified into three pipelines: two-stage methods, one-stage methods, and transformer-based methods.

Two-stage methods Two-stage approaches [3,51] treat visual grounding as first generating candidate object proposals and then finding the best match to the text. In the first stage, an off-the-shelf detector processes the image and proposes regions that may contain the target. In the second stage, a ranking network calculates the similarity between candidate regions and processed text features, selecting the region with the highest similarity score as the final result. Training losses include binary classification loss [29] or maximum-margin ranking loss [51]. To better understand the text and cross-modality matching, MattNet [51] focuses

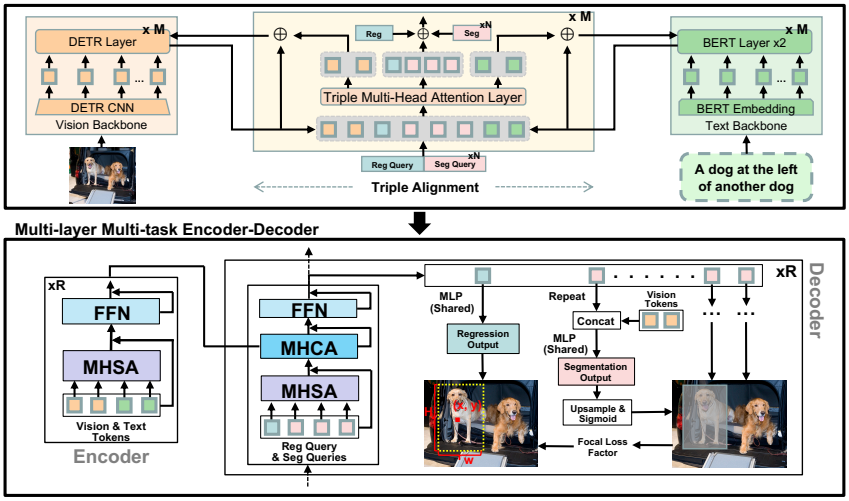

Fig. 2: SegVG: **The upper figure** includes the vision and text backbone. Our proposed Triple Alignment module is iteratively inserted into intermediate layers to eliminate domain discrepancy. **The lower figure** shows our Multi-layer Multi-task Encoder-Decoder, which adopts a transformer encoder-decoder to update multimodal features and ground the target. In this architecture, we make the best of the box annotation as a segmentation ground truth and integrate an additional segmentation task into Visual Grounding. Additionally, the segmentation output serves as a Focal Loss factor, allowing adaptive emphasis on challenging cases for the regression loss. M = 6, R=6.

on decomposing the text into subject, location, and relationship components. [3] introduces an expression-aware score for improved candidate region ranking.

One-stage methods One-stage approaches [47, 48] directly concatenate vision and text features in the channel dimension and rank confidence values for candidate regions proposed based on the concatenated multimodal features. For example, FAOA [48] predicts bounding boxes using a YOLOv3 detector [32] on the concatenated features. ReSC [47] further improves the ability to ground complex queries by introducing a recursive sub-query construction module.

Transformer-based methods Transformer-based approach is first introduced by TransVG [6]. Unlike previous methods, TransVG concatenates the regression query (a learnable embedding), vision tokens, and text tokens and uses transformer encoders [38] to perform cross-modal fusion and target grounding. The query is then processed through an MLP to decode the box. Benefiting from the flexible structure of transformer modules in processing multimodal features, recent works continue adopting this pipeline and propose novelties regarding feature extraction. VLTVG [43] develops a visual-linguistic verification module before the target grounding stage to modulate the vision features with the relationship between vision and text features. QRNet [49] proposes a Query-modulated Refinement Network to mitigate the gap between features from the unimodal vision backbone and those needed for multi-modal reasoning.

Multi-task Visual Grounding Multi-task learning is extensively utilized in object detection and segmentation [2,11], often capitalizing on a shared backbone and task-specific heads. Expanding upon this idea, several studies [19, 25, 36] have proposed solutions to the Multi-task Visual Grounding problem. In this problem, they jointly tackle Referring Expression Comprehension (REC, also known as Visual Grounding) and Referring Expression Segmentation (RES), requiring both box annotations and segmentation annotations. It is important to note that, unlike those approaches, even though we incorporate segmentation losses in our method, **we do not need segmentation annotations** but only box annotations, focusing specifically on the Visual Grounding task.

3 Methodology

In this section, we present the components of our SegVG in the order of the data flow: starting with the backbones, followed by our proposed Triple Alignment, and finally our Multi-layer Multi-task Encoder-Decoder.

3.1 Backbones

As shown in Fig. 2 (upper), our vision backbone consists of ResNet and transformer encoder from DETR [2], with parameters pretrained on Object Detection task using the MSCOCO dataset [22], excluding the validation and test sets of Visual Grounding dataset. The text backbone is the base model of BERT [7].

Vision Backbone Given an input image $\mathbf{I_0}$ ($\mathbb{R}^{3\times H_0\times W_0}$), we employ ResNet to generate a 2D feature map $\mathbf{I}\in\mathbb{R}^{C\times H\times W}$ ($C=2048, H=\frac{H_0}{32}, W=\frac{W_0}{32}$). A 1x1 convolutional layer is then used to reduce the channel dimension of $\mathbf{I}$ to $C_v=256$, resulting in $\mathbf{I}^{'}$. We further flatten $\mathbf{I}^{'}$ into $\mathbf{Z}_v\in\mathbb{R}^{C_v\times N_v}$ ($N_v=H\times W$). Position embedding is then added to $\mathbf{Z}_v$ to preserve sensitivity to the original 2D spatial locations. $\mathbf{Z}_v$ is then iteratively processed through DETR's encoder layer (total 6 transformer layers) and the Triple Alignment to obtain the output $\mathbf{Z}_v$.

Text Backbone Given a text, we initially utilize the BERT's embedding layer to convert it into N_t language tokens with C_t channel dimension. In alignment with [6], we prepend a [CLS] token and append a [SEP] token to the beginning and end positions of the tokenized language, respectively. Following this, we iteratively input the language tokens into BERT's layers (total 12 transformer layers) and the Triple Alignment, generating language embedding $\mathbf{Z}_t$.

3.2 Triple Alignment

Given that the text and vision backbones are pretrained from unimodal tasks and the queries are data-agnostic, the subsequent target grounding stage faces the challenge of aligning these three types of features into the same space before performing multimodal fusion for target grounding. Additionally, considering

that the backbones usually contribute the majority of the overall parameters, solely using them for extracting unimodal features without incorporating multi-modal alignment is sub-optimal. Therefore, an optimal solution is to address the domain discrepancy before moving on to the subsequent target grounding stage.

Triple Alignment (Fig. 2 (upper)) utilizes an attention mechanism to perform triangular feature sampling, aiming to ensure domain consistency among the query, text, and vision features. The queries, $\mathbf{Z}_o$, are first initialized by N learnable embeddings, where one embedding is for the regression query and the rest of the embeddings are for multiple segmentation queries. The data flow is:

$$\mathbf{Z}_v^{i+1} = \text{DETRLayer}_i(\mathbf{Z}_v^i), i \in \{0, 1, .., L-1\} \tag{1}$$

$$\mathbf{Z}_t^{i+1} = \text{BERTLayer}_{2i+1}(\text{BERTLayer}_{2i}(\mathbf{Z}_t^i)), \tag{2}$$

$$[\mathbf{Z}_o^{'}, \mathbf{Z}_t^{'}, \mathbf{Z}_v^{'}] = \text{Tri-MHA}(\mathbf{Z}_o, \mathbf{Z}_t^{i+1}, \mathbf{Z}_v^{i+1}), \tag{3}$$

$$\mathbf{Z}_o = \mathbf{Z}_o + \mathbf{Z}_o^{'}, \mathbf{Z}_t^{i+1} = \mathbf{Z}_t^{i+1} + \mathbf{Z}_t^{'}, \mathbf{Z}_v^{i+1} = \mathbf{Z}_v^{i+1} + \mathbf{Z}_v^{'}, \tag{4}$$

where L is the number of layers, BERTLayer is the layer of BERT and DETRLayer is the layer of DETR's encoder. The vision and text features are first encoded by Eq. 1 and Eq. 2. Subsequently, the three types of tokens (query, text, and vision) are updated by our Triple Multi-Head Attention Layer (Tri-MHA) using Eq. 3. The output tokens are merged back to their original branches respectively by Eq. 4. Within each head of the Triple Multi-Head Attention Layer (Tri-MHA), each type of the features simultaneously computes its updated representation by attending to both the others and itself:

$$\begin{aligned} S = [\mathbf{Z}_o W^{(o,S)}, &\mathbf{Z}_t W^{(t,S)}, \mathbf{Z}_v W^{(v,S)}], S \in \{Q, K, V\} \\ [\mathbf{Z}_o, \mathbf{Z}_t, &\mathbf{Z}_v] = \text{SoftMax}(QK^T/\sqrt{d_k})V, \\ \mathbf{Z}_e^{'} = &\ \mathbf{Z}_e W^e, e \in \{o, t, v\}, \end{aligned} \tag{5}$$

where $\{W^{(e,S)}, W^e : e \in \{o, t, v\}, S \in \{Q, K, V\}\}$ are trainable parameter. As a result, each of the output features is triangular sampling from all of the three types of features, which alleviates the domain discrepancy.

3.3 Multi-layer Multi-task Encoder-Decoder

The Multi-layer Multi-task Encoder-Decoder serves as the target grounding stage, where we use a transformer encoder-decoder for cross-modal fusion and target grounding to perform a box regression task and a box segmentation task.

Encoder As shown in Fig. 2 (lower left), given the aligned output text and vision features from the backbones, the encoder fuses the two modalities into the multimodal features by a stack of transformer layers. In each layer, the concatenated text and vision tokens go through the Multi-Head Self-Attention layer (MHSA) and the Feed Forward Network (FFN) with the residual connection.

Decoder In each decoder layer, we aim to fully exploit the box annotation. We propose the **bbox2seg** paradigm to transform the box annotation into a segmentation mask, which classifies all pixels within the box as foreground (with a value of one) and those outside the box as background (with a value of zero). As shown in Fig. 2 (lower right), one regression query aims to regress the box, while the remaining segmentation queries aim to segment the box. Different segmentation queries are endowed with different learnable positional embeddings to enhance the robustness of each decoder layer, since the decoder layer, when confronted with various queries, is required to segment the same box. Following that, the queries pass through the Multi-Head Self-Attention layer to exchange information about the same target, prompting each other to better locate the target. Subsequently, the queries undergo the Multi-Head Cross-Attention layer and the Feed Forward Network, where multimodal features serve as the Key and Value to ground the target. Finally, a shared MLP across all decoder layers decodes the regression query into the box result, supervised by L1 loss and Giou loss [33]. Each segmentation query is repeated N_v times and concatenated with visual tokens along the channel dimension. Another shared MLP decodes the concatenated feature into the segmentation mask, supervised by Focal loss [21] and Dice loss [27]. It is noteworthy that **our segmentation paradigm shares the same semantic foundation as the regression paradigm, i.e., to distinguish bounding box**, rather than instance segmentation. Therefore, incorporating non-object pixels in the segmented foreground does not introduce ambiguity to the model. We provide qualitative results 4.8 to demonstrate this feature. To alleviate multi-task optimization challenges, we freeze the backbones for the initial k epochs to stabilize the training process.

Confidence score Since both the regression output and segmentation output share the same aim, we can additionally obtain the confidence score for the foreground by averaging values inside the ground truth box of the segmentation output to reflect the confidence of the regression output. In the training process, we can transform this confidence score as the Focal loss factor [21] to adaptively emphasize the other losses of challenging training samples. The final loss function of each decoder layer is formulated as follows:

$$\begin{aligned} L = \lambda_1 c_{focal} L_1 + \lambda_{giou} c_{focal} L_{giou} + \\ \lambda_{dice} c_{focal} L_{dice} + \lambda_{focal} L_{focal}, \end{aligned} \tag{6}$$

where λ_1, λ_{giou}, λ_{focal} and λ_{dice} are hyperparameters. L_1 is the L1 loss. L_{giou} is the GIoU loss [33]. L_{focal} is the Focal loss [21]. L_{dice} is the Dice loss [27]. c_{focal} is the above Focal loss factor averaged across all segmentation outputs.

In the real-world application perspective, the visual grounding task can be viewed as open-vocabulary object detection [54], where target objects lack predetermined categories. Therefore, previous transformer-based methods directly regress the box without confidence scores, since there is no candidate proposal or selection stage in transformer-based pipeline. However, confidence scores are valuable for enhancing the control or reliability of predictions by filtering out low-confidence predictions. This feature could benefit the future integration of visual

Table 1: Comparisons with state-of-the-art methods on widely used datasets. We highlight the best and second best performance in **red** and blue, and bold our model.

Models	Backbone	RefCOCO			RefCOCO+			RefCOCOg			ReferItGame
		val	*testA*	*testB*	*val*	*testA*	*testB*	*val-g*	*val-u*	*test-u*	*test*
Two-stage:											
CMN [12]	VGG16	-	71.03	65.77	-	54.32	47.76	57.47	-	-	28.33
VC [55]	VGG16	-	73.33	67.44	-	58.40	53.18	62.30	-	-	31.13
ParalAttn [57]	VGG16	-	75.31	65.52	-	61.34	50.86	58.03	-	-	-
MAttNet [51]	ResNet-101	76.65	81.14	69.99	65.33	71.62	56.02	-	66.58	67.27	29.04
Similarity Net [39]	ResNet-101	-	-	-	-	-	-	-	-	-	34.54
CITE [29]	ResNet-101	-	-	-	-	-	-	-	-	-	35.07
DDPN [53]	ResNet-101	-	-	-	-	-	-	-	-	-	63.00
LGRANs [40]	VGG16	-	76.60	66.40	-	64.00	53.40	61.78	-	-	-
DGA [44]	VGG16	-	78.42	65.53	-	69.07	51.99	-	-	63.28	-
RvG-Tree [29]	ResNet-101	75.06	78.61	69.85	63.51	67.45	56.66	-	66.95	66.51	-
NMTree [23]	ResNet-101	76.41	81.21	70.09	66.46	72.02	57.52	64.62	65.87	66.44	-
Ref-NMS [3]	ResNet-101	80.70	84.00	76.04	68.25	73.68	59.42	-	70.55	70.62	-
One-stage:											
SSG [5]	DarkNet-53	-	76.51	67.50	-	62.14	49.27	47.47	58.80	-	54.24
ZSGNet [34]	ResNet-50	-	-	-	-	-	-	-	-	-	58.63
FAOA [48]	DarkNet-53	72.54	74.35	68.50	56.81	60.23	49.60	56.12	61.33	60.36	60.67
RCCF [20]	DLA-34	-	81.06	71.85	-	70.35	56.32	-	-	65.73	63.79
ReSC-Large [47]	DarkNet-53	77.63	80.45	72.30	63.59	68.36	56.81	63.12	67.30	67.20	64.60
LBYL-Net [13]	DarkNet-53	79.67	82.91	74.15	68.64	73.38	59.49	62.70	-	-	67.47
Transformer-based:											
TransVG [6]	ResNet-101	81.02	82.72	78.35	64.82	70.70	56.94	67.02	68.67	67.73	70.73
QRNet [49]	Swin-S	84.01	85.85	82.34	72.94	76.17	63.81	71.89	73.03	72.52	74.61
VLTVG [43]	ResNet-101	84.77	87.24	80.49	74.19	78.93	65.17	72.98	76.04	74.18	71.98
SegVG (ours)	ResNet-101	**86.84**	**89.46**	**83.07**	**77.18**	**82.63**	**67.59**	**76.01**	**78.35**	**77.42**	**75.59**

grounding models into downstream multimodal reasoning systems or real-world applications. To meet the requirements of this feature, our approach incorporates a confidence score derived from the segmentation output during inference. Specifically, we calculate the model's confidence by averaging values greater than or equal to 0.35 (adopted from [42]) in the segmentation output of one segmentation query. Analyses 4.7 in the Experiment section demonstrate the faithfulness and benefits of incorporating this additional confidence score.

4 Experiments

4.1 Metric and Datasets

Metric A predicted bounding box is considered accurate if its Intersection over Union (IoU) with the ground-truth bounding box exceeds 0.5. In accordance with the established practices in preceding studies [6, 43], we employ top-1 accuracy (measured in percentage) as the primary metric to assess our method.

Datasets There are five standard benchmarks: RefCOCO [52], RefCOCO+ [52], RefCOCOg-g [26], RefCOCOg-umd [26], and ReferItGame [17]. Four of them (RefCOCO, RefCOCO+, and RefCOCOg-(g/umd)) are all derived from MSCOCO [22]. RefCOCO consists of 19,994 images and 142,210 referring texts, which is divided into four subsets: a training set with 120,624 texts, a validation set with 10,834 texts, and two test sets (testA and testB) containing 5,657

and 5,095 texts, respectively. RefCOCO+ includes 19,992 images and 141,564 referring texts, which is partitioned into four subsets: a training set with 120,191 texts, a validation set with 10,758 texts, and two test sets (testA and testB) containing 5,726 and 4,889 texts, respectively. RefCOCOg contains 25,799 images and 95,010 longer texts. Two widely accepted splitting methods—RefCOCOg-g [26] and RefCOCOg-umd [28]—are employed for this dataset, and we perform experiments using both RefCOCOg-g (val-g) and RefCOCOg-umd (val-u and test-u) splitting conventions. The ReferItGame dataset, featuring 20,000 images from SAIAPR-12 [9], is divided into three segments: a training set with 54,127 texts, a validation set with 5,842 texts, and a testing set comprising 60,103 texts.

4.2 Implementation

Our approach uses an input image size of 640 x 640 and sets the maximum expression length at 40. When resizing images, we preserve the original aspect ratio. The longer edge is resized to 640, and the shorter edge is padded to 640 with the value of zero. Texts exceeding 38 tokens are truncated, reserving a start position and an end position of the characters for the [CLS] and [SEP] token, respectively. If the text is shorter, empty tokens are added after the [SEP] token to reach an input length of 40. Paddings for the input image are not tracked by masks, while empty tokens of text employ masks.

We use the AdamW optimizer. The initial learning rate of 1e-5 is assigned to backbones, and 1e-4 to the rest parameters. Weight decay is set at 1e-4. The visual backbone is initialized with the DETR's backbone and encoder, while the language branch uses the basic BERT model. For the final results, our model is trained for 90 epochs, with the learning rate decreasing by a factor of 10 after 60 epochs. The k hyperparameter in Multi-layer Multi-task Encoder-Decoder is set to 10. We use a batch size of 64. For the ablation studies presented in Table 3, the models are trained for 60 epochs with k equal to 20, and the learning rate drops after 40 epochs. We set $\lambda_1 = 5$, $\lambda_{giou} = 2$, $\lambda_{focal} = 1$ and $\lambda_{dice} = 1$. We adhere to previous practices [6, 43] for data augmentation during training.

4.3 Quantitative Results

We report the performance of our SegVG on all benchmark datasets. As presented in Table 1, our SegVG model demonstrates superiority across all of the datasets. This indicates the effectiveness and generalizability of our approach. It is worth noting that RefCOCO+ and RefCOCOg are relatively more challenging datasets, as RefCOCO+ does not include location terms in its language expressions, and RefCOCOg has longer language expressions compared to other datasets. Despite these challenges, our model exhibits significant improvements on these two difficult datasets. Specifically, on RefCOCO+, our model outperforms the previous SOTA models with +2.99%, +3.7%, and +2.42% on the val, testA, and testB subsets, respectively. On RefCOCOg, our model also surpasses the previous SOTA models with +3.03%, +2.31%, and +3.24% on the val-g,

Table 2: Compare transformer-based models on Parameter count and GFLOPS.

	Backbone	Parameter count (M)	GFLOPS (G)
TransVG [6]	ResNet101	141.55	72.61
VLTVG [43]	ResNet101	141.61	69.87
QRNet [49]	Swin-S	247.06	80.12
SegVG	ResNet101	155.28	73.48

Table 3: Ablation study on RefCOCOg test-u. Encoder and Decoder are the encoder and decoder of Multi-layer Multi-task Encoder-Decoder, respectively. MMDecoder represents Multi-layer and Multi-task supervision in the Decoder. N is the number of segmentation queries. Triple means Triple Alignment. Excluding Query in Triple (g) means using bidirectional alignment with concatenated text-vision tokens.

Id	Model	Acc(%)
(a)	Backbones + Encoder + Decoder	66.97
(b)	Backbones + Triple + Encoder + Decoder	75.75
(c)	Backbones + Encoder + MMDecoder (N=1)	72.61
(d)	Backbones + Encoder + MMDecoder (N=5)	76.21
(e)	Backbones + Triple + Encoder + MMDecoder (N=5)	77.29
(f)	(e) w/o Encoder	76.65(-0.64)
(g)	(e) w/o Query in Triple	76.37(-0.92)
(h)	(e) w/o Focal loss	76.83(-0.46)

val-u, and test-u subsets, respectively. These results suggest that under the reinforcement of Triple Alignment and Multi-layer Multi-task Encoder-Decoder, the query, text, and vision tokens are triangularly updated to share the same space, and the model fully exploits the bounding box as fine-grained pixel-level supervision for comprehensive end-to-end learning.

We also conduct a comparison of the number of parameters and GFLOPS across transformer-based models to evaluate computational costs. As depicted in Table 2, the computational cost of SegVG falls within a reasonable range.

4.4 Ablation Study

In this section, we aim to validate the efficacy of each proposed module. We conduct ablation studies on the RefCOCOg-umd test dataset. Specifically, we start by evaluating a basic structure, i.e., the backbones with encoder-decoder structure. After that, we systematically incorporate the Triple Alignment module into the backbones and introduce the Multi-layer Multi-task supervision into the decoder through the controlled variable approach. Meanwhile, we conduct additional ablation experiments on specific details, including assessing the efficacy of incorporating the encoder, introducing Query in the Triple Alignment, and introducing the Focal loss from the segmentation output to the other losses.

As shown in Table 3, we can draw the following conclusions when comparing the experimental results under controlled variables: **1) [(a) v.s. (b)]**: Incorporating the Triple Alignment can effectively eliminate the domain discrepancy

Table 4: Ablation study on RefCOCOg test-u regarding the number of segmentation queries. ΔHour and ΔAcc are the additional time cost and accuracy improvement compared to ID(i) experiment, respectively.

ID	num_query	Acc	Time Cost	ΔHour / ΔAcc (v.s. ID(i))
i	N = 1	72.61	16.32h	-
ii	N = 3	73.02	18.97h	6.46
iii	N = 5	76.21	20.10h	1.05
iv	N = 7	74.38	22.24h	3.34
v	N = 9	73.91	24.07h	5.96

among the query, text, and vision features, thereby facilitating subsequent target grounding. **2) [(a) v.s. (c)]**: Introducing Multi-layer Multi-task supervision can iteratively make the best of the annotations in the target grounding stage, thereby enhancing the learning of query representations. **3) [(c) v.s. (d)]**: Increasing the number of segmentation queries can further improve the robustness of the decoder when provided with different queries and required to segment the same box. **4) [(a), (b), (d), and (e)]**: Combining the Triple Alignment and Multi-layer Multi-task Encoder-Decoder can effectively enhance the overall performance, achieving the optimal result. **5) [(e) v.s. (f)]**: Even if we include the Triple Alignment supporting multimodal communication, it remains necessary for the subsequent encoder to update the unimodal features generated by the backbones into multimodal features. **6) [(e) v.s. (g)]**: It is necessary to involve queries in Triple Alignment to transfer the data-agnostic embedding into data-related queries. Otherwise, using only Bidirectional Alignment (BA) for text and vision tokens, similar to approaches like *Deep Fusion* in GLIP [18] and Grounding DINO [24], and *WPA* in CoupAlign [56], causes a noticeable decline (-0.92%). **7) [(d), (e) and (g)]**: The slight gain of (g)(76.37%) over (d)(76.21%) stems from our MMDecoder's pixel-level signals, which already boosts BA in the Encoder. Thus, extra BA effort in the backbone is marginal, failing to solve the unaligned query issue. Instead,Tri-Align (e)(77.29%) can solve this issue, showing its novelty. Notably, (d)(76.21%), with a basic encoder-decoder, already achieves SOTA performance, emphasizing our bbox2seg paradigm's simplicity and effectiveness. **8) [(e) v.s. (h)]**: The segmentation output can further derive the confidence score of the model's prediction, which is transformed into the Focal loss factor to adaptively scale the other losses to focus more attention on challenging cases.

We further conduct a more detailed abaltion study about the improvement and corresponding cost of adding more segmentation queries of Tab.3**(c)-(d)**. As shown in Table 4, the best performance is observed with five segmentation queries. Adding more than that increases the burden from pixel-level constraint without benefits, also increasing computational cost per accuracy improvement.

4.5 Triple Alignment Analysis

In additional to the improvement results in ablation study, we further deepen our understanding of the Triple Alignment by analyzing the attention behavior.

Table 5: Attention values from query to text-referred visual region in Triple Alignment.

Layer	RefCOCO			RefCOCO+			RefCOCOg			ReferItGame
	val	*testA*	*testB*	*val*	*testA*	*testB*	*val-g*	*val-u*	*test-u*	*test*
layer2	17%	18%	16%	19%	21%	18%	20%	18%	17%	25%
layer4	34%	36%	33%	29%	31%	28%	27%	33%	33%	30%
layer6	61%	63%	60%	55%	58%	50%	55%	59%	59%	38%

Table 6: Comparison with alternative method using RIS to provide pseudo segmentation labels for supervision.

ReferItGame	SegVG+RIS (LAVT)	SegVG
val	74.91	76.85
test	73.76	75.06

Table 7: AP50 using different segmentation queries for confidence score calculation on RefCOCOg test-u.

Seg Query	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
AP50	84.65	84.57	84.73	84.78	84.63

Specifically, we calculate the sum of the attention values from the query to the text-referred visual region (target bbox) as a percentage of the total attention (which includes attention to the query, text, and visual tokens) to illustrate the extent of alignment across these three modalities in the second, fourth, and final layer of Triple Alignment. We average the percentage across all the attention heads and queries, and perform the analysis across all the datasets. As shown in Table 5, in all the datasets, the attention values increase as the layer progresses, indicating that Triple Alignment progressively aligns the query to comprehend the text and then focus on the referred visual region.

4.6 Comparison with alternative method

Given the development of Referring Expression Segmentation (RES), a natural alternative method would be using a RES method to generate pseudo segmentation labels to substitute our bbox2seg paradigm. Therefore, to mimick real-world scenarios, we use LAVT [46] trained on RefCOCO to obtain pseudo segmentation labels on ReferItGame. We follow the same training setting in ablation study to conduct the comparison on ReferItGame. As shown in Table 6, our SegVG outperforms the alternative method. This demonstrates that our bbox2seg paradigm is more effective than using a RES model to provide pseudo segmentation labels which might suffer from the errors from the RES model.

4.7 Confidence Score Analysis

Selection of segmentation query To show the effect of different selection of segmentation query for the calculation of confidence score, we calculate AP50 on RefCOCOg test-u. As shown in Table 7, the performance variations are slight among them. We use the first one to calculate confidence score for simplicity.

Confidence Score Faithfulness To evaluate the faithfulness of our confidence score, i.e., whether a higher confidence score indicates better performance, we

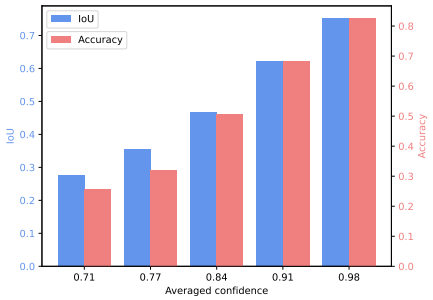

Fig. 3: IoU and Accuracy of different confidence scores on RefCOCOg test-u.

Table 8: Performance of different confidence levels on RefCOCOg test-u.

Confidence	IoU	Acc(%)	Proportion (%)
≥ 0.65	0.7067	77.41%	100.00%
≥ 0.70	0.7072	77.48%	99.89%
≥ 0.75	0.7091	77.69%	99.44%
≥ 0.80	0.7143	78.37%	97.98%
≥ 0.85	0.7226	79.29%	95.04%

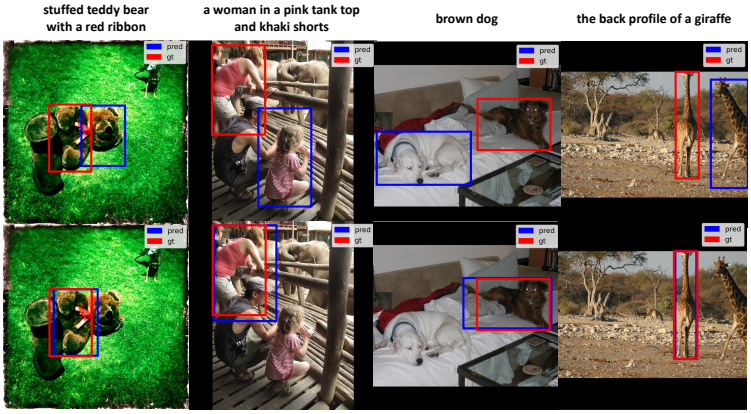

Fig. 4: Qualitative comparison between **(c)** (the first line) and **(d)** (the second line) of Table. 3. Red boxes are ground truth. Blue boxes are model predictions.

assess the relationship of our confidence score and model performance metrics (IoU and Accuracy) as shown in Fig 3. We sort the RefCOCOg-umd test set by confidence score, split it into five equal parts, and calculate each part's average score and performance. We observe a positive correlation between the performance metrics and our confidence score, which confirms its faithfulness.

Confidence Score Application In real applications, the confidence score can be used to enhance the model's reliability. Specifically, we can apply different confidence thresholds to achieve different predictions, as shown in Table 8. First, we observe that accuracy increases with higher thresholds, indicating that adjusting the threshold can enhance the model's localization ability. Furthermore, the mean IoU also increases with the increasing threshold. Therefore, in downstream applications, such as using the model to provide pseudo-labels, we can increase the threshold to obtain a more accurate box. Since low-confidence outputs are excluded, the output proportion is slightly reduced, i.e., yields fewer outputs.

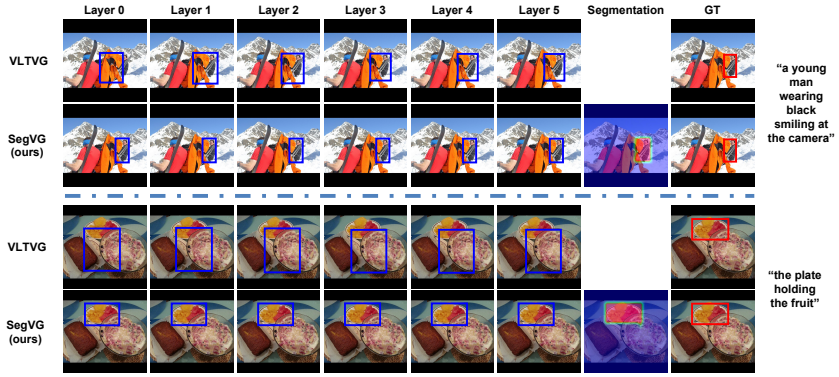

Fig. 5: Layer i refers to the output of the i-th layer of the decoder. The blue boxes represent the models' predictions, while the red boxes denote the ground truth. In the segmentation mask shown in the second column from the right, red indicates high confidence for the foreground. Note that VLTVG does not provide segmentation output.

4.8 Qualitative Results

We compare Tab. 3 (c) and (d) qualitatively to show the robustness enhancement. As shown in Fig. 4, adding segmentation queries improves the robustness to distinguish the target from the distractor, e.g. the case of two dogs.

As depicted in Fig. 5, we compare the box prediction quality by each decoding layer of SegVG with VLTVG [43] which also involves multi-layer supervision. As seen in the upper two rows of Fig. 5, VLTVG initially misses the target "young man" but improves its prediction gradually and finally makes the correct prediction. In contrast, due to the full exploitation of the annotations and the domain alignment in our Triple Alignment, SegVG successfully identifies the location of the target in the early decoding layer and consistently makes the correct prediction in each layer. Another example can be observed in the lower two rows of Fig. 5. In this image, due to the complex colors, VLTVG fails to locate the target "plate" and consistently repeats the same mistake. Instead, SegVG correctly detects the target, even in the first decoder layer. Additionally, we visualize the segmentation mask obtained by SegVG in Fig. 5, which accurately identifies the target box with high confidence. This behavior aligns with the box regression, demonstrating their shared objective, i.e., to distinguish the box.

5 Conclusion

We propose SegVG, a visual grounding model, where we iteratively make the best of box annotations to involve pixel-level supervision and address the domain discrepancy among queries, text, and vision by Triple Alignment module. Experiments show the superior performance of SegVG. Furthermore, we explore the reliability benefits of our segmentation output in real-world applications.

Acknowledgements

This research is supported by NSF IIS-2309073 and ECCS-2123521. This article solely reflects the opinions and conclusions of its authors and not the funding agencies.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6077–6086 (2018)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
3. 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. vol. 35, pp. 1036–1044 (2021)
4. Chen, S., Jin, Q., Wang, P., Wu, Q.: Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9962–9971 (2020)
5. Chen, X., Ma, L., Chen, J., Jie, Z., Liu, W., Luo, J.: Real-time referring expression comprehension by single-stage grounding network. arXiv preprint arXiv:1812.03426 (2018)
6. 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. pp. 1769–1779 (2021)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. 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 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Escalante, H.J., Hernández, C.A., Gonzalez, J.A., López-López, A., Montes, M., Morales, E.F., Sucar, L.E., Villasenor, L., Grubinger, M.: The segmented and annotated iapr tc-12 benchmark. Computer vision and image understanding **114**(4), 419–428 (2010)
10. Gan, C., Li, Y., Li, H., Sun, C., Gong, B.: Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1811–1820 (2017)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
12. 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. pp. 1115–1124 (2017)

13. 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. pp. 16888–16897 (2021)
14. Kang, W., Qu, M., Kini, J., Wei, Y., Shah, M., Yan, Y.: Intent3d: 3d object detection in rgb-d scans based on human intention. arXiv preprint arXiv:2405.18295 (2024)
15. Kang, W., Qu, M., Wei, Y., Yan, Y.: Actress: Active retraining for semi-supervised visual grounding. arXiv preprint arXiv:2407.03251 (2024)
16. Kang, W., Zhou, L., Wu, J., Sun, C., Yan, Y.: Visual grounding with attention-driven constraint balancing. arXiv preprint arXiv:2407.03243 (2024)
17. 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). pp. 787–798 (2014)
18. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10965–10975 (2022)
19. Li, M., Sigal, L.: Referring transformer: A one-step approach to multi-task visual grounding. Advances in neural information processing systems **34**, 19652–19664 (2021)
20. 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. pp. 10880–10889 (2020)
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
23. 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. pp. 4673–4682 (2019)
24. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
25. Luo, G., Zhou, Y., Sun, X., Cao, L., Wu, C., Deng, C., Ji, R.: Multi-task collaborative network for joint referring expression comprehension and segmentation. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 10034–10043 (2020)
26. 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. pp. 11–20 (2016)
27. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. Ieee (2016)
28. Nagaraja, V.K., Morariu, V.I., Davis, L.S.: Modeling context between objects for referring expression understanding. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 792–807. Springer (2016)

29. 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). pp. 249–264 (2018)
30. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. pp. 2641–2649 (2015)
31. Qu, M., Wu, Y., Liu, W., Gong, Q., Liang, X., Russakovsky, O., Zhao, Y., Wei, Y.: Siri: A simple selective retraining mechanism for transformer-based visual grounding. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV. pp. 546–562. Springer (2022)
32. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
33. 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. pp. 658–666 (2019)
34. 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. pp. 4694–4703 (2019)
35. Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: Llava-prumerge: Adaptive token reduction for efficient large multimodal models. arXiv preprint arXiv:2403.15388 (2024)
36. Su, W., Miao, P., Dou, H., Wang, G., Qiao, L., Li, Z., Li, X.: Language adaptive weight generation for multi-task visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10857–10866 (2023)
37. Sun, J., Luo, G., Zhou, Y., Sun, X., Jiang, G., Wang, Z., Ji, R.: Refteacher: A strong baseline for semi-supervised referring expression comprehension. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19144–19154 (2023)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
39. Wang, L., Li, Y., Huang, J., Lazebnik, S.: Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(2), 394–407 (2018)
40. 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. pp. 1960–1968 (2019)
41. Wang, X., Liu, Y., Shen, C., Ng, C.C., Luo, C., Jin, L., Chan, C.S., Hengel, A.v.d., Wang, L.: On the general value of evidence, and bilingual scene-text visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10126–10135 (2020)
42. Wang, Z., Lu, Y., Li, Q., Tao, X., Guo, Y., Gong, M., Liu, T.: Cris: Clip-driven referring image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11686–11695 (2022)
43. 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. pp. 9499–9508 (2022)
44. Yang, S., Li, G., Yu, Y.: Dynamic graph attention for referring expression comprehension. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4644–4653 (2019)
45. Yang, S., Li, G., Yu, Y.: Graph-structured referring expression reasoning in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9952–9961 (2020)
46. Yang, Z., Wang, J., Tang, Y., Chen, K., Zhao, H., Torr, P.H.: Lavt: Language-aware vision transformer for referring image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18155–18165 (2022)
47. Yang, Z., Chen, T., Wang, L., Luo, J.: Improving one-stage visual grounding by recursive sub-query construction. In: European Conference on Computer Vision. pp. 387–404. Springer (2020)
48. 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. pp. 4683–4693 (2019)
49. Ye, J., Tian, J., Yan, M., Yang, X., Wang, X., Zhang, J., He, L., Lin, X.: Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15502–15512 (2022)
50. 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. pp. 4651–4659 (2016)
51. 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. pp. 1307–1315 (2018)
52. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 69–85. Springer (2016)
53. Yu, Z., Yu, J., Xiang, C., Zhao, Z., Tian, Q., Tao, D.: Rethinking diversified and discriminative proposal generation for visual grounding. arXiv preprint arXiv:1805.03508 (2018)
54. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14393–14402 (2021)
55. 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. pp. 4158–4166 (2018)
56. Zhang, Z., Zhu, Y., Liu, J., Liang, X., Ke, W.: Coupalign: Coupling word-pixel with sentence-mask alignments for referring image segmentation. arXiv preprint arXiv:2212.01769 (2022)
57. 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. pp. 4252–4261 (2018)