# An Efficient and Effective Transformer Decoder-Based Framework for Multi-Task Visual Grounding

Wei Chen[1], Long Chen[2], and Yu Wu[1]*

[1]Wuhan University [2]The Hong Kong University of Science and Technology
{weichencs,wuyucs}@whu.edu.cn, longchen@ust.hk

**Abstract.** Most advanced visual grounding methods rely on Transformers for visual-linguistic feature fusion. However, these Transformer-based approaches encounter a significant drawback: the computational costs escalate quadratically due to the self-attention mechanism in the Transformer Encoder, particularly when dealing with high-resolution images or long context sentences. This quadratic increase in computational burden restricts the applicability of visual grounding to more intricate scenes, such as conversation-based reasoning segmentation, which involves lengthy language expressions. In this paper, we propose an efficient and effective multi-task visual grounding (EEVG) framework based on Transformer Decoder to address this issue, which reduces the cost in both language and visual aspects. In the **language aspect**, we employ the Transformer Decoder to fuse visual and linguistic features, where linguistic features are input as memory and visual features as queries. This allows fusion to scale *linearly* with language expression length. In the **visual aspect**, we introduce a parameter-free approach to reduce computation by eliminating background visual tokens based on attention scores. We then design a light mask head to directly predict segmentation masks from the remaining sparse feature maps. Extensive results and ablation studies on benchmarks demonstrate the efficiency and effectiveness of our approach. Code is available in https://github.com/chenwei746/EEVG.

**Keywords:** Visual Grounding · Transformer Decoder · Token Elimination

## 1 Introduction

Visual grounding [30, 47] is a task of locating visual objects based on language expressions, achieved by aligning visual features and linguistic features. According to the granularity of alignment between visual and linguistic information, it can be categorized into two sub-tasks: referring expression comprehension (REC) [3,6,34,49] which facilitates visual-language alignment at the region level, and referring expression segmentation (RES) [35,41,43] which grounds language

* Corresponding author

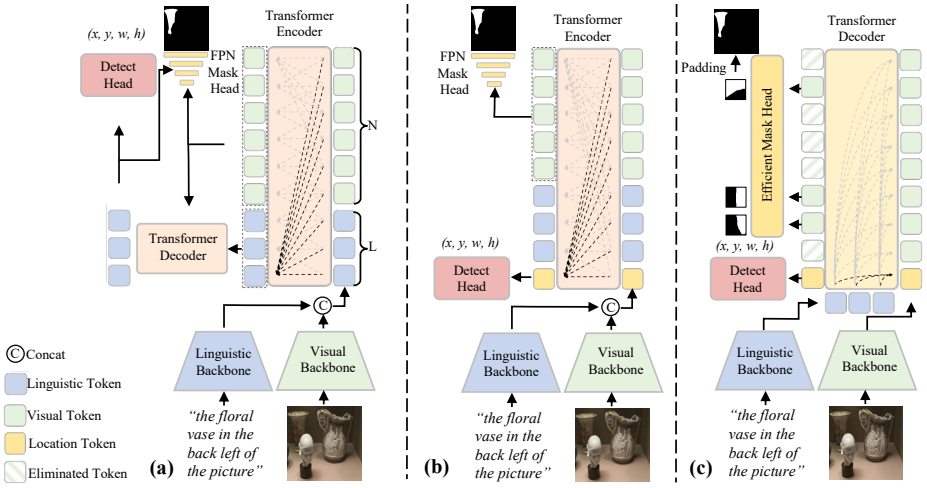


**Fig. 1:** Comparison of different frameworks: (a) Encoder-Decoder methods, (b) Encoder-only methods, and (c) our Decoder-only framework EEVG. In (a) and (b), Transformer Encoder is utilized, and all visual tokens are employed for mask generation. (c) Our method EEVG leverages the Transformer Decoder to integrate diverse modality information and remove background visual tokens during modalities fusion.

expression at the pixel level. Inspired by joint object detection and segmentation, several works [5,22,27,29,37,50] propose a multi-task collaborative learning framework to unify REC and RES, *i.e.*, multi-task visual grounding (MTVG). They demonstrate that REC aids RES in locating referents more accurately, while RES helps REC achieve better vision-language alignment. As a result, MTVG has become a prevailing way of visual grounding.

Current mainstream MTVG models consist of a visual encoder, a linguistic encoder, a cross-modal feature fusion module, and two task heads (*i.e.*, a detection head and a segmentation mask head). Recent state-of-the-art visual grounding works [6, 17, 18, 22, 27] resort to Encoder[1] for cross-modal vision-language fusion. As shown in Fig. 1, these existing methods can be categorized into two categories: (a) Encoder-Decoder [17, 22, 27, 34], where visual-linguistic features are fused by Encoder and target object features are output by Decoder, and (b) Encoder-only [6, 18], where the target object is directly predicted after vision-language feature fusion in Encoder.

However, these methods encounter two efficiency problems: 1) quadratic-increased cost in language length and 2) redundant visual token computation. *Firstly*, traditional methods concatenate visual and linguistic tokens together and input them into Encoder for self-attention, leading to a time complexity of $\mathcal{O}((N+L)^2)$. Thus the computation cost significantly increases as language expressions and context become longer and more complex in the era of Large Lan-

[1] We use *Encoder* and *Decoder* to refer to Transformer encoder and decoder, respectively.

guage Models (LLMs). This hinders the application of visual grounding to more complex scenes, such as conversation-based reasoning segmentation [21], which involves long language contexts. *Secondly*, different from general segmentation or detection tasks, visual grounding usually only aims at locating one referred object. Most visual pixels in the image are not in the region of interest and thus lead to redundant and unnecessary computations, and may distract/mislead the model's attention from the real target. To address these issues, we propose an efficient and effective multi-task visual grounding (EEVG) framework.

To alleviate the cost in the **language aspect** and to deal with longer complex language expressions such as long contextual dialog, we only use Decoder for the visual and language reasoning process. As depicted in Fig. 1 (c), we regard linguistic features as memory and visual features as queries in Decoder. This allows for the fusion of visual and linguistic modalities in the cross-attention module, resulting in a *linear* increase in computational cost with respect to the length of the language expressions. To the best of our knowledge, our method is the first Transformer-based framework with *no Encoder* for cross-modal fusion.

To mitigate the cost in the **visual aspect** and further improve efficiency and efficacy, we introduce a parameter-free strategy to eliminate redundant and distracting image tokens for the visual grounding task. The core idea is dynamically eliminating visual tokens with low attention scores, making the visual feature map to be sparse, and concentrating more on the referred target to remove distracting noise. After that, instead of utilizing the traditional feature pyramid network (FPN) [19] that is widely used in previous MTVG works, we devise a very light-weight and efficient mask head to directly project the remaining sparse tokens into region masks. Previous works' [10, 15, 16, 18, 22, 37, 43] FPN module constitutes 42.8% of the Decoder's parameters (8.1M *versus* 18.9M), acting like an independent network to predict mask. Differently, we use a light-weight two-layer MLP (0.79M) to directly transfer 1-D feature channels of Decoder tokens to the 2-D spatial segmentation mask prediction of the corresponding patch. In this way, we migrate the segmentation prediction workload from the add-on head to the main Decoder. This is consistent with the detection head of REC which also transfers the detection workload to Decoder via a light-weight MLP.

By doing so, Decoder gains a better understanding of the multi-tasks and their mutual improvement, as the location and pixel information are directly embedded in the Decoder token feature. Experiments validate this by the fact that our mask head improves the detection performance on REC by about 2.0%, even though the detection prediction does not go through the mask head. We conduct extensive experiments on several challenging benchmarks including RefCOCO [47], RefCOCO+ [47], and RefCOCOg [30]. Our EEVG is faster than state-of-the art method PolyFormer [27] by 28.19%. Benefiting from light-weight mask head and elimination of disturbing tokens, EEVG also shows enhanced performance. Particularly in the RefCOCOg dataset, which encompasses longer complex language expressions, our method exhibits a notable increase of 3.93% on the RES.

In summary, our contributions are three-fold:

- We propose a Decoder-only framework for MTVG, which reduces computation cost from quadratic to linear increase with regards to language length.
- We propose a dynamic eliminating strategy to reduce redundant and distracting visual tokens, together with a lightweight mask head to directly project the remaining sparse tokens to masks.
- Comprehensive results show that EEVG surpasses state-of-the-art approaches in both speed and performance.

## 2 Related Work

**Referring Expression Comprehension (REC).** REC can be categorized into one-stage and two-stage approaches. Two-stage [12, 13, 48] methods rely on ranking region proposal scores based on language expressions as a crucial component. However, their performance is limited to the pre-trained object detector. While one-stage methods [17, 23, 44, 45] focus on directly predicting the target bounding box guided by language expressions. Yang *et al.* [44] propose a recursive sub-query construction framework to reason between image and query for multiple rounds. To further align modalities, RCCF [23] maps the language domain to the visual domain and performs correlation filtering on the image feature map. Inspired by DETR's [2] success in object detection, MDETR [17] extends it to multi-modal understanding for REC.

**Referring Expression Segmentation (RES).** As RES needs to predict pixel-level results, it heavily relies on accurate vision-language feature extraction and alignment. Previous studies [10, 15, 16, 43] have explored various approaches for cross-modal interaction. EFN [10] utilizes a co-attention mechanism to promote the consistency of the cross-modal information representation in the semantic space. On the other hand, LTS [16] leverages visual-textual features to accurately localize the referenced object by incorporating position priors before facilitating segmentation. Recent work LAVT [43] aligns visual and linguistic representations within the visual backbone using a pixel-word attention module. Current approaches [10, 15, 16, 18, 22, 37, 43] typically employ an FPN-like architecture to generate binary masks from fused visual features. In contrast, our proposed method introduces a lighter mask head based on MLP.

**Multi-task Visual Grounding (MTVG).** To promote consistency between REC and RES, it is natural to integrate them using a shared backbone. In a pioneering effort, MCN [29] proposes a novel multi-task collaborative network that enables joint learning of REC and RES. With the widespread adoption of Transformer [38], follow-up works [18, 22, 37] have employed Transformer as a unified backbone, employing different task heads for REC and RES. Alternatively, SeqTR [50] approaches the problem differently by treating MTVG as a sequence prediction task, representing bounding boxes and masks as discrete coordinate tokens. Moreover, Polyformer [27] leverages precise floating-point coordinates and multi-polygon generation to achieve finer segmentation. Tasks similar to MTVG, Open-Vocabulary Object Detection/Segmentation [11, 42] need to identify all objects across all categories using the vocabularies.

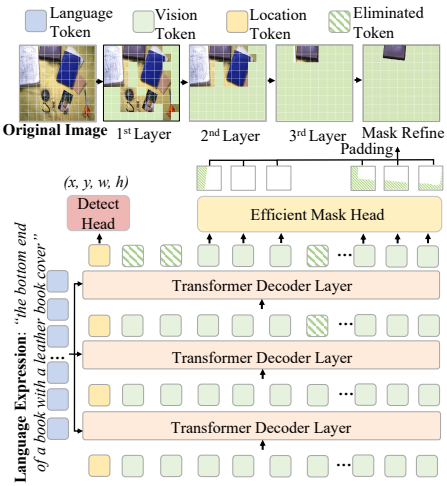


**Fig. 2:** Overview of our method. The language tokens and visual tokens are extracted by a linguistic backbone and a visual backbone which are not shown in the figure.

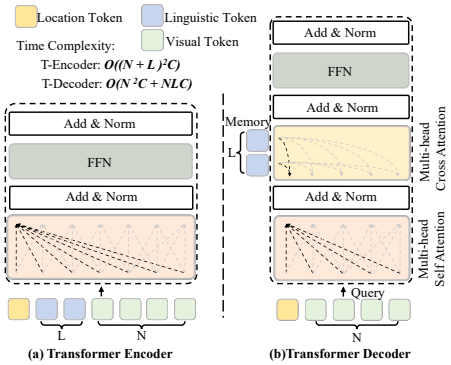


**Fig. 3:** Time complexity comparison between Encoder and Decoder. There is only one input for Encoder while Decoder has two inputs: query and memory. $N$ denotes the number of visual tokens, $L$ means the number of linguistic tokens, $C$ is the dimension of tokens, and "Add & Norm" refers to residual connection and normalization.

**Transformer for Vision-Language Tasks.** Transformer model [38], initially proposed for natural language processing tasks, has demonstrated its effectiveness in the computer vision domain as well, as evidenced by the success of Vision Transformer (ViT) [8,40]. Leveraging the Transformer's exceptional performance in both vision and natural language, researchers have extensively explored its potential as a unified model for vision-language tasks [4,28,39]. Recently, there have been a growing number of methods [6,22,27,37,50] based on the Transformer architecture in visual grounding. TransVG [6], which employs Transformer Encoder for cross-modal fusion, introduces the pioneering transformer-based framework for visual grounding.

## 3 Method

In this section, we first formulate the multi-task visual grounding (MTVG) task and review the prevalent Encoder-based framework in Sec. 3.1. Then, we elaborate on our new Decoder-based framework as shown in Fig. 2, which utilizes Decoder for vision-language fusion (Sec. 3.2), and includes a parameter-free strategy to eliminate visual tokens (Sec. 3.3) and an efficient mask head (Sec. 3.4).

### 3.1 Preliminary

**Formultation.** Given an image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ and a text query $\mathcal{T} \in \mathbb{R}^{L}$, MTVG needs to predict a bounding box $B$ and a binary mask $M$ simultaneously, which

corresponding to the referent. The current prevailing MTVG framework, typically, first utilizes a visual backbone (*e.g.*, ViT [8]) and a linguistic backbone (*e.g.*, BERT [7]) to extract visual features $\boldsymbol{F}_v \in \mathbb{R}^{N \times C_v}$ and linguistic features $\boldsymbol{F}_l \in \mathbb{R}^{L \times C_l}$. After fusing them in the cross-modal interaction module, two task heads are used to predict results.

**Encoder-based Framework.** Existing works [6, 18, 27] directly adopt Encoder as the cross-modal interaction module. As shown in Fig. 3 (a), after linearly projecting $\boldsymbol{F}_v$ and $\boldsymbol{F}_l$ into the same dimension $C$ ($\widetilde{\boldsymbol{F}}_v$ and $\widetilde{\boldsymbol{F}}_l$), $\widetilde{\boldsymbol{F}}_v$ and $\widetilde{\boldsymbol{F}}_l$ are concatenated with a learnable location token $\widetilde{\boldsymbol{F}}_{loc} \in \mathbb{R}^{1 \times C}$ and fed into Encoder:

$$[\hat{\boldsymbol{F}}_{loc}, \hat{\boldsymbol{F}}_l, \hat{\boldsymbol{F}}_v] = \texttt{Encoder}([\widetilde{\boldsymbol{F}}_{loc}; \widetilde{\boldsymbol{F}}_l; \widetilde{\boldsymbol{F}}_v]), \tag{1}$$

where $[\cdot;\ \cdot;\ \cdot]$ denotes the concatenation operation.

After getting fused features, for the detection task, a two-layer MLP is used to project $\hat{\boldsymbol{F}}_{loc}$ into four dimensions (*i.e.*, $(x, y, w, h)$). For the segmentation results, $\hat{\boldsymbol{F}}_v$ is reshaped from sequence to square (*i.e.*, $\mathbb{R}^{N \times C} \rightarrow \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times C}$ where $P$ is the patch size) and uses FPN-like [19] architecture built on convolution layers to generate masks. This architecture typically needs the entire visual features because convolution layers can only work in a square feature map which contains redundant costs in the visual features corresponding to the background. Therefore, we utilize our elimination strategy and efficient head to alleviate this.

**Time Complexity Analysis.** The computational cost of aligning different modalities in the Encoder-based methods primarily lies in the multi-head self-attention (MSA) which can be formulated as:

$$\texttt{MSA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \texttt{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{C}})\boldsymbol{V}, \tag{2}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent query, key, and value, respectively. They are obtained through linear projections of the input features. Specifically, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V} \in \mathbb{R}^{(N+L+1) \times C}$. According to Eq. (2), its time complexity can be calculated as $\mathcal{O}((N+L)^2 C)$. In our method, we aim to alleviate the quadratic increase in burden by reducing it to a linear one with respect to $L$, as well as minimizing the number of $N$.

### 3.2 Transformer Decoder for Modalities Fusion

**Module Details.** First, we separately project visual feature $\boldsymbol{F}_v$ and linguistic feature $\boldsymbol{F}_l$ into the same channel dimension $C$ ($\widetilde{\boldsymbol{F}}_v$ and $\widetilde{\boldsymbol{F}}_l$). Then we adopt Transformer Decoder [38] for vision-language fusion, as depicted in Fig. 3 (b), where each decoder layer consists of an MSA layer, a multi-head cross-attention (MCA) layer, and a feed-forward network. MCA has a similar architecture to MSA, but the difference is that MCA has two inputs: the first one is input as Q and another one is input as K and V in Eq. (2). We concatenate $\widetilde{\boldsymbol{F}}_v$ with a learnable location token $\widetilde{\boldsymbol{F}}_{loc}$ and input them into the MSA layer:

$$[\boldsymbol{F}'_{loc}, \boldsymbol{F}'_v] = \texttt{LN}(\texttt{MSA}([\widetilde{\boldsymbol{F}}_{loc}; \widetilde{\boldsymbol{F}}_v]) + [\widetilde{\boldsymbol{F}}_{loc}; \widetilde{\boldsymbol{F}}_v]), \tag{3}$$

where $\texttt{LN}(\cdot)$ refers to layer normalization. After that, we input $[\boldsymbol{F}'_{loc}; \boldsymbol{F}'_v]$ as the query of MCA and input linguistic feature $\widetilde{\boldsymbol{F}}_l$ as the key and value of MCA:

$$[\boldsymbol{F}^*_{loc}, \boldsymbol{F}^*_v] = \texttt{LN}(\texttt{MCA}([\boldsymbol{F}'_{loc}; \boldsymbol{F}'_v], \widetilde{\boldsymbol{F}}_l) + [\boldsymbol{F}'_{loc}; \boldsymbol{F}'_v]), \tag{4}$$

Finally, $[\boldsymbol{F}^*_{loc}, \boldsymbol{F}^*_v]$ is passed into the feed-forward network:

$$[\hat{\boldsymbol{F}}_{loc}, \hat{\boldsymbol{F}}_v] = \texttt{LN}(\texttt{FFN}([\boldsymbol{F}^*_{loc}; \boldsymbol{F}^*_v]) + [\boldsymbol{F}^*_{loc}; \boldsymbol{F}^*_v]), \tag{5}$$

where $\texttt{FFN}(\cdot)$ means the feed-forward network, which is a two-layer MLP.

**Time Complexity Analysis.** The computational cost mainly lies in the MSA and MCA, with time complexities of $\mathcal{O}(N^2C)$ and $\mathcal{O}(NLC)$, respectively. Thus, the overall time complexity is $\mathcal{O}(N^2C + NLC)$, which increases linearly with respect to $L$.

### 3.3 Parameter-free Token Elimination Strategy

Visual grounding usually aims at locating one referent and most referents only occupy a small percentage of the visual tokens where most visual tokens are not in the region of interest. Therefore, there exist redundant costs in background visual tokens. As a result, we attempt to address this issue by eliminating background visual tokens. During our analysis, we discovered that the attention scores between the location token and the visual tokens are notably higher for those corresponding to the target object. This finding led us to the conclusion that we can effectively eliminate visual tokens with low attention scores which are depicted in Fig. 4.

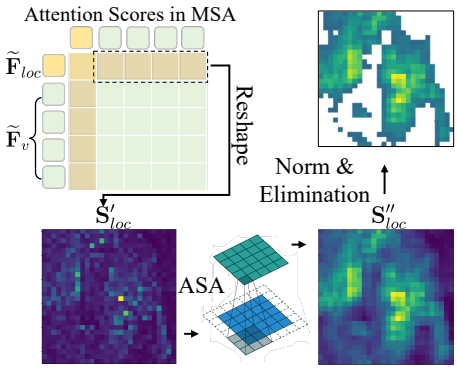


**Fig. 4:** We conduct the visual tokens elimination process in each Decoder layer. "ASA" denotes adaptive spatial attention and "Norm & Elimination" means normalization and eliminating visual tokens according to Eq. (7).

**Dynamic Elimination.** There are some related works [1,9] eliminating visual tokens in image classification. However, these methods eliminate a fixed number in each model layer and typically retain only the essential object features. These approaches are suitable for coarse granularity tasks like image classification. In contrast, the task of RES requires the preservation of the entire object and deals with objects of varying sizes, each demanding a different number of tokens to be eliminated. To tackle this challenge, we propose a dynamic elimination strategy. First, attention scores $\boldsymbol{S}_{loc} \in \mathbb{R}^{1\times N}$ between the location token and visual tokens are calculated as:

$$\begin{aligned} \boldsymbol{Q}_{loc} &= \widetilde{\boldsymbol{F}}_{loc}\boldsymbol{W}_{\boldsymbol{Q}}, \boldsymbol{K}_v = \widetilde{\boldsymbol{F}}_v\boldsymbol{W}_{\boldsymbol{K}}, \\ \boldsymbol{S}_{loc} &= \texttt{softmax}(\frac{\boldsymbol{Q}_{loc}\boldsymbol{K}_v^T}{\sqrt{C}}), \end{aligned} \tag{6}$$

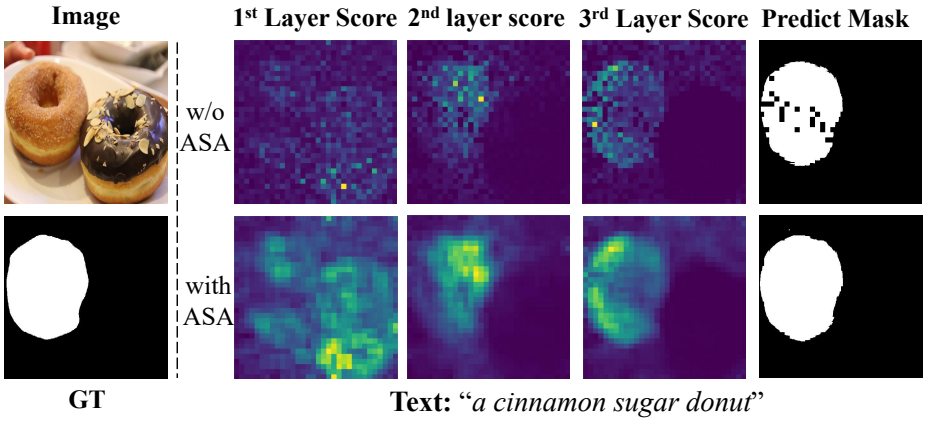


**Fig. 5:** Visualization of attention scores $S_{loc}$ of location token and visual tokens. GT refers to ground truth, "w/o ASA" denotes without using adaptive spatial attention, and "$1^{st}$Layer Score" means the first Decoder layer attention scores between location token and visual tokens.

where $\boldsymbol{W_Q} \in \mathbb{R}^{C\times C}$ and $\boldsymbol{W_K} \in \mathbb{R}^{C\times C}$ are parameter matrices in MSA. Then we normalize the attention scores $\boldsymbol{S}_{loc}$ and remove those visual tokens whose attention scores are smaller than $\alpha$:

$$\begin{aligned} \widetilde{\boldsymbol{S}}_{loc} &= \frac{\boldsymbol{S}_{loc} - \min \boldsymbol{S}^i_{loc}}{\max \boldsymbol{S}^i_{loc} - \min \boldsymbol{S}^i_{loc}}, 0 \le i < N \\ \overline{\boldsymbol{F}}_v &= \{\hat{\boldsymbol{F}}^i_v \mid \widetilde{\boldsymbol{S}}^i_{loc} \ge \alpha,\ 0 \le i < N\}, \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{F}}^i_v$ is the $i$-th visual token, $\overline{\boldsymbol{F}}_v \in \mathbb{R}^{N'\times C}$ denotes the remained tokens, and $N'$ means the number of remained visual tokens. Our strategy offers two advantages over prior works [1,9]: 1) it dynamically eliminates a different number of tokens for different objects in different sizes, and 2) the number of eliminated tokens gradually increases as the loss converges, preventing error elimination.

**Adaptive Spatial Attention.** As shown in the predicted mask in Fig. 5, we found that some visual tokens corresponding to target objects are eliminated incorrectly. Therefore, we propose the attention weight average strategy to make the attention weights spatially aware to maintain the original shape of the target object. We reshape $\boldsymbol{S}_{loc}$ into $\boldsymbol{S}'_{loc} \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}}$, then calculate as follows:

$$\boldsymbol{S}''_{loc}[i,j] = \frac{\sum_{u=-k}^{k}\sum_{v=-k}^{k} \boldsymbol{S}'_{loc}[i+u, j+v]}{(2k+1)^2}, \tag{8}$$

where $0 \le i < \frac{H}{P}$ and $0 \le j < \frac{W}{P}$. After that, each attention score is averaged with the surrounding attention scores, then we reshape $\boldsymbol{S}''_{loc}$ back to $\boldsymbol{S}_{loc} \in \mathbb{R}^{1\times N}$ and following Eq. (7) to eliminate visual tokens. So we can alleviate the problem of incorrectly eliminating tokens in the referent.

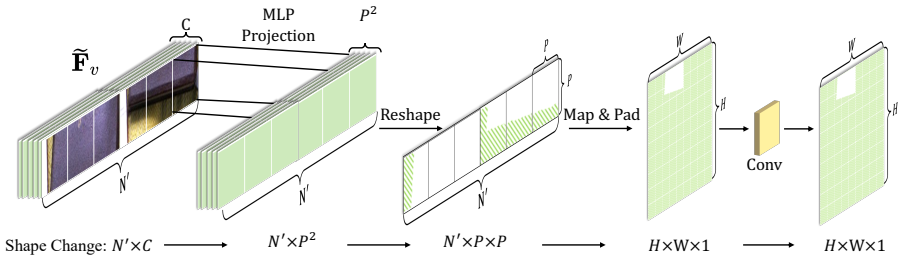


**Fig. 6:** Our mask head utilizes MLP to project tokens from channel to spatial region masks and adopts a 1-channel convolution layer to establish spatial relationships among pixels. "Map & Pad" refers to mapping each patch to the original position in the image and padding the eliminated position with 0.

### 3.4 Efficient Mask Head

After eliminating tokens, we get some sparse visual tokens. To further reduce the redundant cost, we propose a lightweight and efficient mask head to generate masks from sparse visual tokens instead of using the FPN-like mask head which needs to pad the eliminated tokens. The procedure of our mask head is shown in Fig. 6. We denote the rest of indexes $\mathbb{I}$ as follows:

$$\mathbb{I} = \{i \mid \widetilde{\boldsymbol{S}}_{loc}^{i} \geq \alpha,\ 0 \leq i < N\}, \tag{9}$$

We utilize MLP to transfer the remaining visual tokens from 1-D feature channels into 2-D spatial binary masks ($C \rightarrow P^2$ dimension) and pad the eliminated tokens with 0:

$$\mathcal{M}^i = \begin{cases} MLP(\overline{\boldsymbol{F}}_v^{f(i)}), & \text{if } i \in \mathbb{I} \\ 0^{1 \times P^2}, & \text{if } i \notin \mathbb{I} \end{cases} \tag{10}$$

where $0 \leq i < N$ and $f(i)$ means the index in remained tokens corresponding to the $i$-th original token. We reshape and permute $\mathcal{M} \in \mathbb{R}^{N \times P^2}$ into $\mathcal{M}' \in \mathbb{R}^{H \times W}$ and since the pixels projected by MLP are not spatially related to each other, we use the local context processing method to make neighboring pixels relate to each other. In the actual implementation, we use a 5*5 convolutional kernel $Conv$, and mask $M$ is generated as follows:

$$M = \texttt{sigmoid}(\texttt{Conv}(\mathcal{M}')), \tag{11}$$

### 3.5 Multi-task Training

Our method is an end-to-end framework that unifies REC and RES, incorporating two distinct types of loss: detection loss and segmentation loss.

**Detection Loss.** For REC task, we denote the predicted bounding box $B = (\hat{x}, \hat{y}, \hat{w}, \hat{h})$ and the ground truth $B_{gt} = (x, y, w, h)$, and the detection loss

**Table 1:** Comparison with state-of-the-art methods on RefCOCO [47], RefCOCO+ [47], and RefCOCOg [32] for **REC** task. **Bold** denotes the best performance. Swin-B and ViT-B are abbreviations for Swin-Transformer Base and ViT Base.

| Method | Backbone | Multi-task | RefCOCO | | | RefCOCO+ | | | RefCOCOg | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | val | test A | test B | val | test A | test B | val(U) | test(U) |
| MAttNet [46] | MRCNN-Res101 | ✗ | 76.65 | 81.14 | 69.99 | 65.33 | 71.62 | 56.02 | 66.58 | 67.27 |
| NMTree [26] | MRCNN-Res101 | ✗ | 76.41 | 81.21 | 70.09 | 66.46 | 72.02 | 57.52 | 65.87 | 66.44 |
| LBYL [14] | DarkNet53 | ✗ | 79.67 | 82.91 | 74.15 | 68.64 | 73.38 | 59.49 | - | - |
| MCN [29] | DarkNet53 | ✓ | 80.08 | 82.29 | 74.98 | 67.16 | 72.86 | 57.31 | 66.46 | 66.01 |
| TransVG [6] | ResNet101 | ✗ | 81.02 | 82.72 | 78.35 | 64.82 | 70.70 | 56.94 | 68.67 | 67.73 |
| TRAR [49] | DarkNet53 | ✗ | - | 81.40 | 78.60 | - | 69.10 | 56.10 | 68.90 | 68.30 |
| SeqTR [50] | DarkNet53 | ✓ | 81.23 | 85.00 | 76.08 | 68.82 | 75.37 | 58.78 | 71.35 | 71.58 |
| PVD [5] | DarkNet53 | ✓ | 82.51 | 86.19 | 76.81 | 69.48 | 76.83 | 59.68 | 68.40 | 69.57 |
| PVD [5] | Swin-B | ✓ | 84.52 | 87.64 | 79.63 | 73.89 | 78.41 | 64.25 | 73.81 | 74.13 |
| VG-LAW [37] | ViT-B | ✓ | 86.62 | 89.32 | 83.16 | 76.37 | 81.04 | 67.50 | 76.90 | 76.96 |
| EEVG **(Ours)** | MRCNN-Res101 | ✓ | 82.19 | 85.34 | 77.18 | 71.35 | 76.76 | 60.73 | 70.18 | 71.28 |
| EEVG **(Ours)** | DarkNet53 | ✓ | 81.82 | 86.02 | 74.67 | 69.72 | 76.26 | 57.95 | 71.38 | 70.93 |
| EEVG **(Ours)** | Swin-B | ✓ | 86.79 | 89.52 | 83.12 | 77.52 | **83.05** | 66.93 | 78.15 | 78.11 |
| EEVG **(Ours)** | ViT-B | ✓ | **88.08** | **90.33** | **85.50** | **77.97** | 82.44 | **69.15** | **79.60** | **80.24** |

is define as follows:

$$\mathcal{L}_{det} = \mathcal{L}_{smooth-L1}(B, B_{gt}) + \mathcal{L}_{giou}(B, B_{gt}), \tag{12}$$

where $\mathcal{L}_{smooth-L1}(\cdot,\cdot)$ and $\mathcal{L}_{giou}(\cdot,\cdot)$ are the smooth L1 loss and GIoU loss [36].

**Segmentation Loss.** For RES task, the segmentation loss is calculated using the predicted segmentation mask denoted as $M \in \mathbb{R}^{H\times W}$, and the ground truth denoted as $M_{gt} \in \mathbb{R}^{H\times W}$, according to the following formula:

$$\mathcal{L}_{seg} = \mathcal{L}_{focal}(M, M_{gt}) + \mathcal{L}_{dice}(M, M_{gt}), \tag{13}$$

where $\mathcal{L}_{focal}(\cdot,\cdot)$ and $\mathcal{L}_{dice}(\cdot,\cdot)$ represent focal loss [24] and dice loss [31]. Finally, the joint training loss function is defined as follows:

$$\mathcal{L} = \lambda_{det}\mathcal{L}_{det} + \lambda_{seg}\mathcal{L}_{seg}. \tag{14}$$

where $\lambda_{det}$ and $\lambda_{seg}$ represent the weight coefficients for the detection loss and segmentation loss, respectively.

## 4 Experiments

### 4.1 Experiment Settings

**Datasets.** The commonly used datasets in visual grounding are RefCOCO [47], RefCOCO+ [47], and RefCOCOg [30], which are collected from MS-COCO [25]. RefCOCO contains 19,994 images with 142,210 referring expressions for 50,000 objects which is split into the training set, the validation set, the testA set, and the testB set. RefCOCO+, excluding absolute-location words, consists of 19,992 images with 49,856 referred objects and 141,564 referring expressions. There are

**Table 2:** Comparison with state-of-the-art methods on RefCOCO [47], RefCOCO+ [47], and RefCOCOg [32] for **RES** task. **Bold** denotes the best performance. Swin-B and ViT-B are abbreviations for Swin-Transformer Base and ViT Base.

| Method | Backbone | Multi-task | RefCOCO | | | RefCOCO+ | | | RefCOCOg | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | val | test A | test B | val | test A | test B | val(U) | test(U) |
| MAttNet [46] | MRCNN-Res101 | ✗ | 56.51 | 62.37 | 51.70 | 46.67 | 52.39 | 40.08 | 47.64 | 48.61 |
| NMTree [26] | MRCNN-Res101 | ✗ | 56.59 | 63.02 | 52.06 | 47.40 | 53.01 | 41.56 | 46.59 | 47.88 |
| MCN [29] | DarkNet53 | ✓ | 62.44 | 64.20 | 59.71 | 50.62 | 54.99 | 44.69 | 49.22 | 49.40 |
| CRIS [41] | CLIP-ResNet50 | ✗ | 69.52 | 72.72 | 64.70 | 61.39 | 67.10 | 52.48 | 59.87 | 60.36 |
| SeqTR [50] | DarkNet53 | ✓ | 67.26 | 69.79 | 64.12 | 54.14 | 58.93 | 48.19 | 55.67 | 55.64 |
| PVD [5] | DarkNet53 | ✓ | 68.87 | 70.53 | 65.83 | 54.98 | 60.12 | 50.23 | 57.81 | 57.17 |
| LAVT [43] | Swin-B | ✗ | 74.46 | 76.89 | 70.94 | 65.81 | 70.97 | 59.23 | 63.34 | 63.62 |
| PVD [5] | Swin-B | ✓ | 74.82 | 77.11 | 69.52 | 63.38 | 68.60 | 56.92 | 63.13 | 63.62 |
| VG-LAW [37] | ViT-B | ✓ | 75.62 | 77.51 | 72.89 | 66.63 | 70.38 | 59.89 | 65.53 | 66.08 |
| EEVG **(Ours)** | MRCNN-Res101 | ✓ | 71.28 | 73.87 | 67.49 | 61.96 | 66.25 | 53.74 | 59.94 | 60.72 |
| EEVG **(Ours)** | DarkNet53 | ✓ | 70.66 | 74.16 | 65.60 | 60.76 | 65.38 | 50.70 | 59.79 | 59.93 |
| EEVG **(Ours)** | Swin-B | ✓ | 75.79 | 77.86 | 72.78 | 67.62 | 71.48 | 59.12 | 67.40 | 67.30 |
| EEVG **(Ours)** | ViT-B | ✓ | **78.23** | **79.27** | **76.58** | **69.04** | **72.65** | **62.33** | **69.15** | **70.01** |

25,799 images with 49,856 referred objects and 141,564 referring expressions in RefCOCOg whose descriptions are longer and more complex. We use the umd-splits [32] for RefCOCOg. Implementation details can be found in the Appendix.

**Evaluation Metrics.** For REC, we utilize the accuracy of the grounding results as the evaluation metric. The predicted region is deemed correct if the intersection over union (IoU) between the predicted region and the ground truth exceeds 0.5. As for RES, we employ the mean Intersection over Union (mIoU) between predicted masks and ground truth as the evaluation metric.

### 4.2 Quantitative Results

**Results of RefCOCO Series.** To validate the effectiveness of our method, we conduct experiments and report our performance on RefCOCO series datasets (*i.e.*, RefCOCO/+/g). The results of REC and RES are reported in Table 1 and Table 2, respectively. Our method outperforms previous state-of-the-art approaches in both REC and RES. Particularly on RefCOCOg, which includes longer and more complex language expressions, our method exhibits even greater improvement (3.62% in the val set and 3.93% in the test set), showcasing its effectiveness in handling intricate scenes.

**Table 3:** Speed comparison with SOTA methods. "↓" means lower is better, "↑" refers to upper is better, and "FPS" denotes frames per second. The batch size is 20 and all experiments are conducted in one RTX 4090.

| Method | LAVT [43] | PolyFormer [27] | **Ours** |
|---|---|---|---|
| Runtime (ms) ↓ | 285.97 | 318.36 | **248.35** |
| FPS ↑ | 69.94 | 62.82 | **80.53** |

**Speed Comparison.** In order to demonstrate the efficiency of our proposed method, as shown in Table 3, we conduct a speed comparison with LAVT [43] and PolyFormer [27]. We use the same settings with LAVT and PolyFormer, *i.e.*, Swin-B & BERT-base as backbones, and the length of linguistic tokens is

**Table 4:** Comparison with pre-trained state-of-the-art methods on RefCOCO [47], RefCOCO+ [47], and RefCOCOg [32] for **REC** and **RES** task. **Bold** denotes the best performance. Swin-B and ViT-B are Swin-Transformer Base and ViT Base.

| Method | Backbone | Task Type | RefCOCO | | | RefCOCO+ | | | RefCOCOg | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | val | test A | test B | val | test A | test B | val(U) | test(U) |
| RefTr [22] | ResNet101 | REC | 85.65 | 88.73 | 81.16 | 77.55 | 82.26 | 68.99 | 79.25 | 80.01 |
| SeqTR [50] | DarkNet53 | REC | 87.00 | 90.15 | 83.59 | 78.69 | 84.51 | 71.87 | 82.69 | 83.37 |
| PolyFormer [27] | Swin-B | REC | 89.73 | 91.73 | 86.03 | **83.73** | **88.60** | **76.38** | 84.46 | **84.96** |
| EEVG **(Ours)** | Swin-B | REC | 89.63 | 92.00 | 86.40 | 82.24 | 87.34 | 74.00 | 83.99 | 84.53 |
| EEVG **(Ours)** | ViT-B | REC | **90.47** | **92.73** | **87.72** | 81.79 | 87.80 | 74.94 | **85.19** | 84.72 |
| RefTr [22] | ResNet101 | RES | 74.34 | 76.77 | 70.87 | 66.75 | 70.58 | 59.40 | 66.63 | 67.39 |
| SeqTR [50] | DarkNet53 | RES | 71.70 | 73.31 | 69.82 | 63.04 | 66.73 | 58.97 | 64.69 | 65.74 |
| PolyFormer [27] | Swin-B | RES | 75.96 | 77.09 | 73.22 | 70.65 | 74.51 | 64.64 | 69.36 | 69.88 |
| EEVG **(Ours)** | Swin-B | RES | 77.52 | 79.63 | 75.25 | 71.35 | 75.56 | 64.58 | 71.48 | 71.90 |
| EEVG **(Ours)** | ViT-B | RES | **79.49** | **80.87** | **77.39** | **71.86** | **76.67** | **66.31** | **73.56** | **73.47** |

**Table 5:** Performance between Encoder and Decoder for cross-modal interaction.

| Cross-modal Module | Backbone | Task Type | RefCOCO | | | RefCOCO+ | | | RefCOCOg | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | val | test A | test B | val | test A | test B | val(U) | test(U) |
| Encoder | ViT-B | REC | 86.59 | 89.59 | 84.43 | 75.82 | 81.11 | 68.48 | 78.43 | 78.08 |
| Decoder | ViT-B | REC | **87.55** | **90.03** | **84.71** | **77.31** | **81.84** | **68.89** | **79.27** | **79.26** |
| Encoder | ViT-B | RES | 76.97 | 78.72 | 75.53 | 67.40 | 71.25 | 60.70 | 68.12 | 68.19 |
| Decoder | ViT-B | RES | **77.49** | **78.99** | **75.73** | **68.19** | **71.53** | **61.03** | **68.16** | **68.60** |

20. To ensure a fair comparison, we exclude the time cost of point generation in PolyFormer, as it utilizes a point sequence approach to generate masks after visual-linguistic feature alignment. Compared with PolyFormer which utilizes Encoder for modalities fusion, we are faster than it 28.19% in FPS (80.53 *versus* 62.82). Compared with LAVT which devises an interaction module to fuse linguistic features in the visual backbone, we are still faster than it 15.14% in FPS (80.53 *versus* 69.94).

**Results of Pre-trained setting.** Table 4 presents the results of our proposed method pre-trained on a large corpus of visual referring expression data and fine-tuned on the RefCOCO, RefCOCO+, and RefCOCOg datasets. The large corpus consists of the combination of Visual Genome [20], RefCOCO [47], RefCOCO+ [47], RefCOCOg [30], and Flickr30k [33] datasets. Following PolyFormer [27], we pre-train our model using the REC task on this large corpus and subsequently fine-tune on the combined training sets of RefCOCO, RefCOCO+, and RefCOCOg in both REC and RES task, with all validation and test images removed. For the RES task, our method achieves state-of-the-art performance across all validation and test splits, outperforming previous methods by a considerable margin. In the REC task, our approach demonstrates superior or comparable performance to prior state-of-the-art techniques.

### 4.3 Ablation Studies

**Comparison between Encoder and Decoder.** To demonstrate the effectiveness and efficiency of Decoder, we conduct experiments to compare Decoder

**Table 6:** Runtime (ms) comparison between Encoder and Decoder, both of them with 3 layers. The batch size is 20 and all experiments are conducted in one RTX 4090. N and L denote the number of visual and linguistic tokens, respectively. The lower value is better in the table.

| $N$ (Visual Tokens) | 196 | | | | | 784 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $L$ (Linguistic Tokens) | 60 | 100 | 150 | 200 | 300 | 60 | 100 | 150 | 200 | 300 |
| Encoder | 6.41 | 7.99 | 9.64 | 11.14 | 14.85 | 28.56 | 30.33 | 33.12 | 35.39 | 40.51 |
| Decoder | **5.80** | **6.06** | **6.59** | **6.93** | **8.09** | **28.10** | **28.74** | **29.93** | **30.81** | **33.00** |

with Encoder for vision language fusion. Table 5 demonstrates that Decoder outperforms Encoder, highlighting Decoder's ability to facilitate vision language alignment. Additionally, Table 6 indicates that Decoder exhibits faster speed compared to Encoder, particularly when handling longer language expressions. It should be noted that Decoder incorporates an additional multi-head cross-attention module. To ensure a fair comparison, we set the dimension of the feed-forward network to 1024 in Decoder and 2048 in Encoder, thereby aligning their parameter quantities.

**Table 7:** Ablation study of eliminating tokens on RefCOCOg. In the static eliminating strategy, we remove 96 tokens from each layer. ASA denotes adaptive spatial attention, "↓" means lower is better, and "↑" refers to higher is better. The runtime of token elimination module is tested in one RTX 4090 and the batch size is 20.

| Elimination Strategy | ASA | Runtime (ms) ↓ | RES | | REC | |
|---|---|---|---|---|---|---|
| | | | val ↑ | test ↑ | val ↑ | test ↑ |
| No | ✗ | 27.85 | 68.16 | 68.60 | 79.27 | 79.26 |
| Static | ✗ | 24.72 | 66.38 | 66.94 | 78.33 | 79.18 |
| Static | ✓ | 24.88 | 67.75 | 68.21 | 78.72 | 79.20 |
| Dynamic | ✗ | **24.68** | 68.89 | 68.95 | 79.27 | 79.34 |
| Dynamic | ✓ | 24.83 | **69.15** | **70.01** | **79.60** | **80.24** |

**Token Elimination.** To validate the advantages of our token elimination strategy, we compare it with the results obtained without elimination as well as the results achieved using a static elimination strategy [9]. As depicted in Table 7, employing the static token elimination strategy leads to a decline in performance when compared to the absence of an elimination strategy. Conversely, the dynamic token elimination strategy not only reduces computational costs but also exhibits performance improvements. Furthermore, the adoption of adaptive spatial attention demonstrates enhancement in performance, thereby mitigating the issue of erroneously eliminating certain visual tokens associated with target objects, as illustrated in Fig. 5.

**Table 8:** Ablation study of mask head on RefCOCOg. The FPN-like is the mask head based on convolution layers within the FPN framework, which has prevailed in recent VG works [10, 22, 37, 43]. "↓" means lower is better and "↑" refers to higher is better. The runtime of mask head is tested in one RTX 4090 and batch size is 20.

| Mask Head | Parameter Number ↓ | Runtime (ms) ↓ | RES | | REC | |
|---|---|---|---|---|---|---|
| | | | val ↑ | test ↑ | val ↑ | test ↑ |
| FPN-like | 8.11M | 23.56 | 68.64 | 69.12 | 77.68 | 78.26 |
| **Ours** | **0.79M** | **0.87** | **69.15** | **70.01** | **79.60** | **80.24** |

**Efficient Mask Head.** In comparison to the FPN-like mask head [19], as reported in Table 8, our mask head has fewer parameters and faster speed. Moreover, our mask head not only demonstrates improvements in the RES task but also exhibits enhancements in the REC task. We believe this is because our

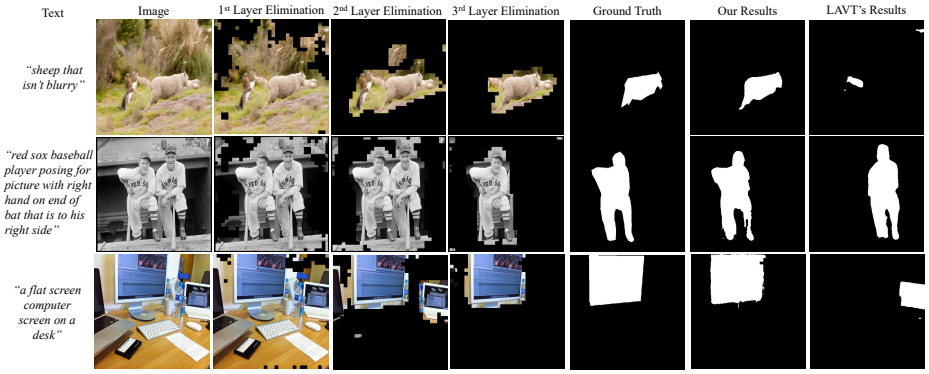


**Fig. 7:** Visualization of our eliminating process, our predicted results, and LAVT's [43].

light mask head offloads the spatial prediction workload from the task head add-ons to Decoder which is consistent with the light MLP-based detection head. Consequently, Decoder benefits from improved vision-language fusion, as the location and pixel information are now embedded within the Decoder token feature channels.

### 4.4 Qualitative Results

We present qualitative results achieved using our proposed method and compare them with the results obtained from LAVT, as shown in Fig. 7. These results demonstrate the effectiveness of our approach in managing complex scenes with similar and potentially distracting objects. In such situations, LAVT tends to make incorrect predictions because of these distracting objects. However, our method removes these distracting objects in various Decoder layers, allowing for accurate predictions of the target object. More visualization examples can be found in the Appendix.

## 5 Conclusion

In this paper, we present a novel approach to visual grounding that achieves superior performance while requiring less computational resources. By incorporating visual and linguistic features through cross-attention in the Transformer Decoder, our method effectively handles longer language expressions without significantly increasing the computational cost. To further enhance efficiency, we introduce a parameter-free strategy to remove unnecessary visual tokens during cross-modal fusion. This strategy not only reduces computations but also improves overall performance by eliminating irrelevant objects. After obtaining sparse visual tokens, we propose an efficient mask head that directly generates masks without the need for padding. Extensive experiments conducted on benchmark datasets validate that our method surpasses state-of-the-art techniques in both referring expression comprehension and segmentation tasks.

## Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under Grant 62372341. Long Chen was supported by HKUST Special Support for Young Faculty (F0927) and HKUST Sports Science and Technology Research Grant (SSTRG24EG04).

## References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. ICLR (2022)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (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: AAAI (2021)
4. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: ECCV (2020)
5. Cheng, Z., Li, K., Jin, P., Ji, X., Yuan, L., Liu, C., Chen, J.: Parallel vertex diffusion for unified visual grounding. arXiv preprint arXiv:2303.07216 (2023)
6. Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. In: ICCV (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. In: ICLR (2020)
9. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: ECCV (2022)
10. Feng, G., Hu, Z., Zhang, L., Lu, H.: Encoder fusion network with co-attention embedding for referring image segmentation. In: CVPR (2021)
11. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921 (2021)
12. Hong, R., Liu, D., Mo, X., He, X., Zhang, H.: Learning to compose and reason with language tree structures for visual grounding. TPAMI (2019)
13. Hu, R., Rohrbach, M., Andreas, J., Darrell, T., Saenko, K.: Modeling relationships in referential expressions with compositional modular networks. In: CVPR (2017)
14. Huang, B., Lian, D., Luo, W., Gao, S.: Look before you leap: Learning landmark features for one-stage visual grounding. In: CVPR (2021)
15. Huang, S., Hui, T., Liu, S., Li, G., Wei, Y., Han, J., Liu, L., Li, B.: Referring image segmentation via cross-modal progressive comprehension. In: CVPR (2020)
16. Jing, Y., Kong, T., Wang, W., Wang, L., Li, L., Tan, T.: Locate then segment: A strong pipeline for referring image segmentation. In: CVPR (2021)
17. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: ICCV (2021)
18. Kim, N., Kim, D., Lan, C., Zeng, W., Kwak, S.: Restr: Convolution-free referring image segmentation using transformers. In: CVPR (2022)

19. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR (2019)
20. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV (2017)
21. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. arXiv preprint arXiv:2308.00692 (2023)
22. Li, M., Sigal, L.: Referring transformer: A one-step approach to multi-task visual grounding. NeurIPS (2021)
23. 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: CVPR (2020)
24. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
25. 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: ECCV (2014)
26. Liu, D., Zhang, H., Wu, F., Zha, Z.J.: Learning to assemble neural module tree networks for visual grounding. In: ICCV (2019)
27. Liu, J., Ding, H., Cai, Z., Zhang, Y., Satzoda, R.K., Mahadevan, V., Manmatha, R.: Polyformer: Referring image segmentation as sequential polygon generation. In: CVPR (2023)
28. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. NeurIPS (2019)
29. 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: CVPR (2020)
30. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR (2016)
31. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016)
32. Nagaraja, V.K., Morariu, V.I., Davis, L.S.: Modeling context between objects for referring expression understanding. In: ECCV (2016)
33. 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: ICCV (2015)
34. 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: ECCV (2022)
35. Qu, M., Wu, Y., Wei, Y., Liu, W., Liang, X., Zhao, Y.: Learning to segment every referring object point by point. In: CVPR (2023)
36. 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: CVPR (2019)
37. Su, W., Miao, P., Dou, H., Wang, G., Qiao, L., Li, Z., Li, X.: Language adaptive weight generation for multi-task visual grounding. In: CVPR (2023)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
39. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: ICML (2022)

40. Wang, W., Chen, W., Qiu, Q., Chen, L., Wu, B., Lin, B., He, X., Liu, W.: Crossformer++: A versatile vision transformer hinging on cross-scale attention. IEEE TPAMI (2023)
41. Wang, Z., Lu, Y., Li, Q., Tao, X., Guo, Y., Gong, M., Liu, T.: Cris: Clip-driven referring image segmentation. In: CVPR (2022)
42. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2945–2954 (2023)
43. Yang, Z., Wang, J., Tang, Y., Chen, K., Zhao, H., Torr, P.H.: Lavt: Language-aware vision transformer for referring image segmentation. In: CVPR (2022)
44. Yang, Z., Chen, T., Wang, L., Luo, J.: Improving one-stage visual grounding by recursive sub-query construction. In: ECCV (2020)
45. Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., Luo, J.: A fast and accurate one-stage approach to visual grounding. In: ICCV (2019)
46. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: CVPR (2018)
47. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: ECCV (2016)
48. Zhang, H., Niu, Y., Chang, S.F.: Grounding referring expressions in images by variational context. In: CVPR (2018)
49. Zhou, Y., Ren, T., Zhu, C., Sun, X., Liu, J., Ding, X., Xu, M., Ji, R.: Trar: Routing the attention spans in transformer for visual question answering. In: ICCV (2021)
50. Zhu, C., Zhou, Y., Shen, Y., Luo, G., Pan, X., Lin, M., Chen, C., Cao, L., Sun, X., Ji, R.: Seqtr: A simple yet universal network for visual grounding. In: ECCV (2022)