ScanFormer: Referring Expression Comprehension by Iteratively Scanning

Wei Su[1] Peihan Miao[2] Huanzhang Dou[1] Xi Li[1,3*]
[1]College of Computer Science and Technology, Zhejiang University
[2]School of Software Technology, Zhejiang University
[3]Zhejiang-Singapore Innovation and AI Joint Research Lab
{weisuzju, peihan.miao, hzdou, xilizju}@zju.edu.cn

Abstract

Referring Expression Comprehension (REC) aims to localize the target objects specified by free-form natural language descriptions in images. While state-of-the-art methods achieve impressive performance, they perform a dense perception of images, which incorporates redundant visual regions unrelated to linguistic queries, leading to additional computational overhead. This inspires us to explore a question: can we eliminate linguistic-irrelevant redundant visual regions to improve the efficiency of the model? Existing relevant methods primarily focus on fundamental visual tasks, with limited exploration in vision-language fields. To address this, we propose a coarse-to-fine iterative perception framework, called ScanFormer. It can iteratively exploit the image scale pyramid to extract linguistic-relevant visual patches from top to bottom. In each iteration, irrelevant patches are discarded by our designed informativeness prediction. Furthermore, we propose a patch selection strategy for discarded patches to accelerate inference. Experiments on widely used datasets, namely RefCOCO, RefCOCO+, RefCOCOg, and ReferItGame, verify the effectiveness of our method, which can strike a balance between accuracy and efficiency.

1. Introduction

As a fundamental task in vision-language understanding, Referring Expression Comprehension (REC) [9, 48, 53, 55] relies on free-form natural language descriptions to identify the referred target object. The development of REC not only can underpin various vision-language tasks [18, 30, 40, 56], but also potentially contribute to real-world applications such as human-computer interaction [39, 46].

In REC, images typically contain a substantial amount of redundant information when contrasted with highly concise and information-dense linguistic queries. For instance, as shown in Fig. 1, the image has considerable redundant visual regions that are weakly correlated or even unrelated to the language query, such as persons around the target catcher and extensive low-information background regions. However, state-of-the-art methods [9, 48, 53] adopt the form of dense perception to obtain visual features for subsequent cross-modal interaction. These methods use visual encoders such as ResNet [14], DarkNet [36], Swin Transformer [27], *etc.*, and traverse the entire spatial locations of the image using sliding windows or non-overlapping patches to extract features, as shown in Fig. 1 (a). Despite achieving impressive performance, the form of dense perception brings a significant amount of redundant information and increases computational overhead for the entire model. Especially in Transformer-based models [9, 48], the computational complexity of multi-head self-attention [43] is quadratic. This leads to a research question: **is it possible to discard linguistic-irrelevant redundant visual regions to enhance the efficiency of the model?**

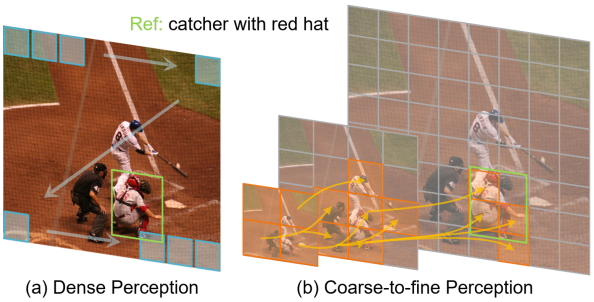

Figure 1. The comparison of dense perception and coarse-to-fine iterative perception. The dense perception extracts features with sliding windows or non-overlapping patches by traversing the image. In contrast, our iterative perception can identify and discard linguistic-irrelevant redundant regions from coarse to fine scales.

It is worth noting that there is an emerging trend [4, 6, 44, 45, 52] to explore the elimination of redundant visual features. Typical bottom-up merging methods [4, 44, 52]

*corresponding author.

initially divide the images into fine-grained patches and gradually merge the patches in subsequent multiple stages to reduce visual tokens. However, the initial abundance of tokens inevitably leads to a substantial computational cost in the early stages, especially when dealing with high-resolution images. In addition, the top-down coarse-to-fine methods [6, 45] start with coarse-grained partitioning using a large patch size, and gradually decrease the patch size to obtain fine-grained visual tokens. For instance, DVT [45] cascades multiple Transformers, and uses confident predictions to determine whether to divide the entire image into finer-grained patches using a smaller patch size. However, this method usually brings considerable redundant visual regions and increases computational overhead. CF-ViT [6] introduces a coarse-to-fine two-stage vision Transformer, which identifies informative patches in the coarse stage and further re-split them into finer patches in the second stage. Although impressive performance in classification, the heuristic informative region identification based on class attention limits its extension to other tasks and models without the [CLS] token. Furthermore, since it is non-learnable, applying regularization to control token sparsity is challenging. Therefore, existing efficient Transformer methods still have limitations, and focus on visual tasks while ignoring the exploration of the vision-language fields.

To address this, this paper proposes a coarse-to-fine iterative perception framework, termed **ScanFormer**, as shown in Fig. 1 (b). To be specific, using a pre-constructed image scale pyramid, the model initiates visual perception from the coarse-grained and low-resolution image at the top of the pyramid. By predicting the informativeness of finer-grained patches in the next iteration, the model adaptively eliminates redundant visual regions, ultimately reaching the fine-grained and high-resolution image at the bottom of the pyramid. We keep previous tokens in the cache without further updates, thus reducing computational resources. The new tokens extracted in each iteration interact with themselves and previous tokens contained in the cache via self-attention and cross-attention, respectively. In this process, multi-scale patch partitioning enables the model to aggregate scale-related information from different spatial positions. Furthermore, we propose a patch selection strategy for discarded patches to accelerate inference. A learnable token participates in the coarse-to-fine iterative perception process and is ultimately utilized for coordinate regression to directly predict the target box. Extensive experiments have demonstrated the effectiveness of our ScanFormer, which achieves state-of-the-art methods on widely-used datasets, *i.e.,* RefCOCO [49], RefCOCO+ [49], RefCOCOg [32], and ReferItGame [20].

The main contributions can be summarized as follows:

- We propose ScanFormer, a coarse-to-fine iterative perception framework that progressively discards linguistic-irrelevant redundant visual regions in each iteration to enhance the efficiency of the model.
- To achieve patch selection, we propose to select tokens by constant token replacement, where the unselected tokens are replaced by a constant token and merged finally for real acceleration.
- Extensive experiments demonstrate the effectiveness of our ScanFormer, which strikes a balance between accuracy and efficiency compared to state-of-the-art methods.

2. Related Work

2.1. Referring Expression Comprehension

Most conventional methods [5, 17, 26, 50, 51] explore REC through a two-stage framework. Concretely, in the first stage, numerous candidate proposals for the input image are pre-generated using a pre-trained object detector [37]. In the second stage, the proposal that best matches the given referring expression is considered the referred target box. However, two-stage methods are constrained by the accuracy and speed of the object detector. To this end, one-stage methods [30, 46, 47, 54] based on dense anchors [36] are proposed, which can achieve faster speed and comparable performance to the two-stage methods. In recent years, the success of the transformer [43] in vision-language fields has attracted researchers, leading to the emergence of REC methods [3, 9, 41, 48, 53, 55] based on the transformer. Due to the multi-head attention mechanism [43], transformer-based REC methods can effectively capture cross-modal relationships. While achieving impressive performance, these methods incur additional computational overhead due to their dense perception of images. To this end, this paper proposes a coarse-to-fine iterative perception framework to enhance the efficiency of the model.

2.2. Efficient Vision Transformer

The self-attention mechanism [43] is the primary reason for the inefficiency of ViT [11], as its computational complexity grows quadratically with the number of visual tokens. Recently, several methods [4, 6, 44, 45, 52] have have emerged to improve the efficiency of ViT by reducing the number of computed visual tokens. These methods can be broadly categorized into bottom-up token merging methods [4, 44, 52] and top-down coarse-to-fine methods [6, 45]. To be specific, bottom-up token merging methods [4, 44, 52] initially divide the high-resolution image into fine-grained patches, and gradually merge these patches in multiple stages to reduce the number of visual tokens. In addition, top-down coarse-to-fine methods [6, 45] start with coarse-grained partitioning, *i.e.* large patch size but a small number of tokens, and progressively reduce the patch size while performing fine-grained partitioning. However, existing relevant methods focus on efficient vision Transform-

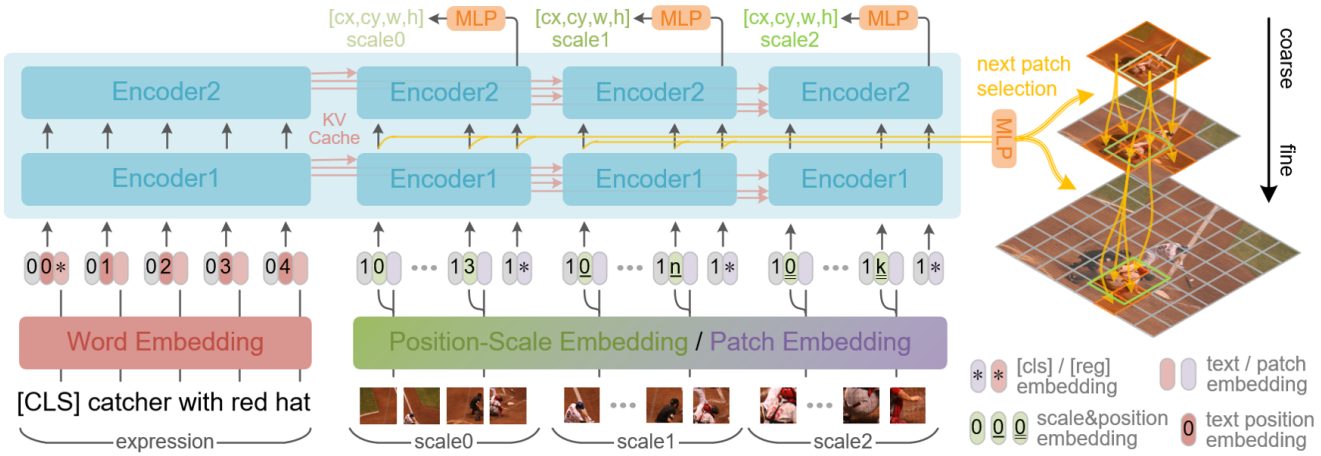

Figure 2. The overall architecture of ScanFormer. The text inputs and image patches at each scale share the encoder. The outputs of the first half part of the encoder, *i.e.* Encoder1, are used to select finer-grained patches for the next level. The [REG] tokens output by the second half of the encoder, *i.e.* Encoder2, are used to predict the coordinates of the referred object at the corresponding scale. The key and value features generated in the encoder are cached and propagated from left to right.

ers, with a limited exploration into efficient vision-language Transformers. In this paper, we explore an efficient vision-language Transformer framework for REC.

3. Method

In this section, we give a detailed description of our ScanFormer for REC. First, we briefly introduce the overview of our framework in Sec. 3.1. Then, we elaborate on the patch selection strategy in Sec. 3.2. Next, we describe our prediction head in Sec. 3.3. Finally, we detail the training objectives of the whole framework in Sec. 3.4.

3.1. Framework

ScanFormer utilizes a unified Transformer-like structure for linguistic and visual modalities, as illustrated in Fig. 2. Concretely, the framework consists of word embedding, patch embedding, position-scale embedding, and encoders. Word embedding and patch embedding extract features from texts and images, respectively. Position-scale embedding is used to encode the spatial position and scale size of each image patch. The encoder consists of N layers, each comprising a Multi-Head Attention (MHA) layer and a Feed-Forward Network (FFN). In addition, each encoder layer is equipped with a cache to store the output features. The query for MHA comes from the input features, while the key and value are composed of features from the input features and the previous cached features, as illustrated in Fig. 3. The causality in scale not only reduces the amount of calculations but also leverages previous linguistic and multi-scale visual information to update features.

The input of linguistic modality is initially encoded by the framework, and the extracted linguistic features are stored in the cache. Subsequently, for the visual modality, an image scale pyramid with S scales is constructed based on the input image I. From top to bottom, for each iteration, selected patches are extracted and processed through the framework, where intermediate features are used to generate the selection of sub-patches in the next pyramid layer. In addition, the cache at each layer of the encoder stores the visual features obtained after each iteration. The features corresponding to the [REG] token in each iteration are used to predict the coordinates of the referred object at the corresponding scale. In particular, for the image at the top of the pyramid, all the patches are selected to ensure that the model captures global information. As the scale increases, ScanFormer incorporates finer-grained features to achieve accurate predictions, while discarding irrelevant patches to save substantial computing resources.

Specifically, for the linguistic modality, the referring text $t \in \mathbb{R}^{L\times|V|}$ is embedded with the word embedding matrix $T \in \mathbb{R}^{|V|\times d}$, prepended the [CLS] embedding $T^{cls} \in \mathbb{R}^{d}$, and then added with the text position embedding matrix $T^{pos} \in \mathbb{R}^{(L+1)\times d}$ and the type embedding $T^{type} \in \mathbb{R}^{d}$. The embedded linguistic features are first fed into the framework, and the updated linguistic features are stored in the cache at each layer of the encoder. For the visual modality, from top to bottom of the image scale pyramid, taking level-i as an example, the N_i selected patches with (P, P) resolution and C channels are first flattened to $v \in \mathbb{R}^{N_i\times(P^2\cdot C)}$ and then projected to $E \in \mathbb{R}^{N_i\times d}$ with a linear projection layer. After that, the patch features are added with the spatial embedding $E^{spatial} \in \mathbb{R}^{N_i\times d}$ and the type embedding $E^{type} \in \mathbb{R}^{d}$. $E^{spatial}$ is produced by the position-scale embedding $PSE : [0,1]^3 \rightarrow \mathbb{R}^d$ with the normalized patch coordinates and scales $[cx, cy, s]$

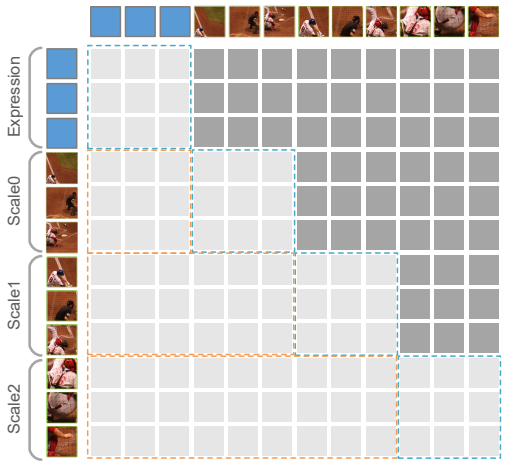

Figure 3. Token interaction of different modalities and scales. "dark" color means blocking interaction. The regions surrounded by blue dotted lines represent the interaction in each iteration, and the regions surrounded by orange dotted lines represent the interaction with the K&V cache.

as inputs. After that, The embedding of the [REG] token $E^{reg} \in \mathbb{R}^d$ is appended, which is used to regress the bounding box $[cx_i, cy_i, w_i, h_i]$ of the object at level i.

3.2. Patch Selection by Constant Replacement

To facilitate learning to select informative patches through back-propagation, a selection factor s_i is generated for the i-th patch. There are two options for using s_i: **(1)** Apply s_i to every head of the MHA on every Transformer layer. This is achieved by weighting the key and value, and gradually decaying s_i to 0.0 to minimize its impact on the remaining tokens. However, for a Transformer with N layers and H heads, obtaining clear gradient signals to optimize s_i is challenging, making it difficult to achieve ideal learning choices. **(2)** Apply s_i directly to the inputs of the Transformer, *i.e.* patch embedding, in a weighted manner. Since s_i is only used at this location, it is easier to train. Therefore, this paper adopts the second candidate.

Furthermore, it is worth noting that even if the input patch embedding is set to zero, it still becomes non-zero in subsequent layers due to the bias terms of FFN and MHA, and the dot product attention. Fortunately, when the token sequence contains many identical tokens, the calculation of MHA can be simplified, leading to practical inference acceleration. To improve the model's adaptability, the paper suggests replacing the patch embedding with a learnable constant token rather than directly setting it to zero. Therefore, the patch selection problem is transformed into a patch replacement problem. Next, the constant patch replacement and merging for acceleration will be introduced.

Constant Token Replacement. To implement the token replacement, a constant token $E^{const} \in \mathbb{R}^d$ is introduced and the selection logits $r_i \in \mathbb{R}$ for i-th patch is yielded from the Transformer. We follow the improved semantic hashing [19] to learn r_i by back-propagation. To encourage exploration, noises are added to r_i, *i.e.* $r_i^n = r_i + n$. During training, $n \sim \mathcal{N}(0, 1)$, and $n = 0$ when evaluation and inference. Then, two variables $v_1 = \sigma'(r_i^n)$, and $v_2 = \mathbb{I}(r_i^n \geq 0)$ can be calculated.

$$\sigma'(x) = clamp(1.2\sigma(x) - 0.1, 0, 1), \tag{1}$$

where $\mathbb{I}(\cdot)$ and $\sigma(\cdot)$ are the indicative function and $sigmoid$ respectively. During training, in the forward pass, we uniformly sample v_1 and v_2 as the selection factor s_i.

$$s_i = \mathbb{I}(n_s \geq 0.5) \cdot v_1 + \mathbb{I}(n_s < 0.5) \cdot v_2, \tag{2}$$

where $n_s \sim Uniform[0, 1]$ represents the random sample weight. In the backward pass, the gradients always flow to v_1, even if v_2 is used in the forward computation. The weighted patch embedding $\overline{E}_i$ can be calculated as:

$$\overline{E}_i = s_i \cdot E_i + (1 - s_i) \cdot E^{const}. \tag{3}$$

During training, s_i is regularized to 0, *i.e.* the i-th token is replaced by the constant token E^{const}.

Merging Constant Tokens. Although the redundant tokens are replaced by the constant tokens and are still included in the forward computation of the encoder, they can not be discarded directly without any impact. However, it can be proved that these constant tokens can be merged to reduce the computation effectively. Taking a key and value sequences with N tokens and N_c constant tokens:

$$\begin{aligned} K &= [\underbrace{k_1, k_2, \cdots, k_i}_{N-N_c}, \underbrace{k^c, \cdots, k^c}_{N_c}] \\ V &= [\underbrace{v_1, v_2, \cdots, v_i}_{N-N_c}, \underbrace{v^c, \cdots, v^c}_{N_c}] \end{aligned} \tag{4}$$

The keys and values of N_c tokens can be reduced to only one key and value by concatenating a constant vector to keys, which can be illustrated as:

$$\begin{aligned} K' &= concat([\underbrace{k_1, k_2, \cdots, k_i}_{N-N_c}, k^c], [\underbrace{0, 0, \cdots, 0}_{N-N_c}, log(N_c)]) \\ V' &= [\underbrace{v_1, v_2, \cdots, v_i}_{N-N_c}, v^c] \end{aligned} \tag{5}$$

According to the scaled dot-product attention mechanism, the attention values $A \in \mathbb{R}^N$ for one query $q \in \mathbb{R}^d$ relative to K can be calculated as:

$$A = softmax(\frac{qK^T}{\sqrt{d}}). \tag{6}$$

It can be concluded that the same attention-weighted value can be derived using Eq. (4) and Eq. (5) according to Eq. (6). Therefore, $N_c - 1$ tokens are eventually dropped and the computations brought by them can be saved. The Pytorch [34] implementation is illustrated in Algorithm 1.

Algorithm 1. Dot-product Attention with Constant Tokens

```
"""
The features of constant tokens are stored to
the last element of key and value.
Input:
  query: (bsz, dst_len, ndim)
  key, value: (bsz, src_len, ndim)
  num_const: (bsz, 1)
Output:
  attn_out: (bsz, dst_len, ndim)
"""
pad_query = torch.ones(bsz, dst_len, 1)
pad_key = torch.zeros(bsz, src_len, 1)
pad_key[:, -1] = torch.log(num_const)

q = torch.cat([query, pad_query], dim=-1)
k = torch.cat([key, pad_key], dim=-1)

attn = torch.softmax(q@k.t(-1, -2), dim=-1)
attn_out = attn @ value
```

3.3. Prediction Head

The referred object may exist at various scales. Similar to the object detection methods[36, 37], where multi-scale predictions are conducted at different feature levels, for each scale level in ScanFormer, we apply direct coordinate regression [9] to predict the bounding box of the referred object. The regression token [REG] is introduced to gather features of image patches across the Transformer. The output features corresponding to the [REG] token is fed to a shared multi-layer perception (MLP), followed by the Sigmoid function to predict the normalized bounding box $\hat{b} = (\hat{x}, \hat{y}, \hat{w}, \hat{h})$ of the referred objects.

3.4. Training Objectives

We optimize the proposed coarse-to-fine iterative perception framework end-to-end. For the l-th image scale, we can obtain predicted bounding box $\hat{b}_l = (\hat{x}_l, \hat{y}_l, \hat{w}_l, \hat{h}_l)$. Given the ground truth $b = (x, y, w, h)$, the detection loss function is defined as follows:

$$\mathcal{L}_{bbox} = \sum_{l=0}^{2} \lambda_{L1}^{l} \mathcal{L}_{L1}(b, \hat{b}_l) + \sum_{l=0}^{2} \lambda_{giou}^{l} \mathcal{L}_{giou}^{l}(b, \hat{b}_l), \quad (7)$$

where $\mathcal{L}_{L1}(\cdot,\cdot)$ and $\mathcal{L}_{giou}(\cdot,\cdot)$ represent L1 loss and Generalized IoU loss [38], respectively, and λ_{L1}^{l} and λ_{giou}^{l} are the relative weights to control the detection loss penalty for the l-th image scale.

In addition, to control the sparsity of selected patches, we add a regularization loss function as follows:

$$\mathcal{L}_{sparse} = \lambda_{sparse} \sum_{l=1}^{2} (\frac{1}{N_l} \sum_{i=1}^{N_l} s_i^l - \beta^l)^2, \quad (8)$$

where λ_{sparse} represents the relative weights to control the sparsification penalty, and s_i^l represents the selection factor for the i-th patch in Eq. (2) in the l-th image scale. β^l is the hyperparameter to control the ratio of selected tokens from the l-th image scale. The total loss function of our ScanFormer is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{bbox} + \mathcal{L}_{sparse}, \quad (9)$$

The trained ScanFormer can strike a balance between accuracy and efficiency. The experimental analysis of the ScanFormer will be elaborated in Sec. 4.

4. Experiment

In this section, we provide a detailed experimental analysis of the entire framework, including the datasets, evaluation protocol, training and inference implementation details, comparisons with state-of-the-art methods, early exiting, and qualitative results.

4.1. Datasets and Evaluation Protocol

Datasets. To demonstrate the effectiveness of our method, we conduct experiments on the widely used REC dataset, which includes RefCOCO [49], RefCOCO+ [49], RefCOCOg [33], and ReferItGame [20]. RefCOCO, RefCOCO+ and RefCOCOg are constructed based on MS-COCO [25]. To be specific, RefCOCO and RefCOCO+ are collected from interactive games, including train, val, testA, and testB sets. In contrast to RefCOCO, expressions in RefCOCO+ do not contain words related to the absolute position of the referred objects. Unlike RefCOCO and RefCOCO+, RefCOCOg is collected on Amazon Mechanical Turk in a non-interactive setting, which results in longer and more complex referring expressions. Following the common split version [33], RefCOCOg consists of train, val, and test sets. In addition, ReferItGame is constructed based on SAIAPR-12 [12], including train and test sets. We also pre-train ScanFormer with a large-scale pre-training dataset, which contains 174k images with approximately 6.1M distinct referring expressions by combining the train sets of RefCOCO/+/g, ReferItGame, Visual Genome regions [22], and Flickr entities [35].

Evaluation Protocol. Following the previous works [9, 48, 55], we choose $Acc@0.5$ as the metric to evaluate the accuracy of positioning the referred objects, where $Acc@0.5$ represents the percentage of predicted correct samples among all test samples. For each sample, if the

Methods	Venue	Backbone	RefCOCO			RefCOCO+			RefCOCOg		ReferItGame
			val	testA	testB	val	testA	testB	val	test	test
Two-stage:											
MAttNet[50]	CVPR18	ResNet-101/LSTM	76.65	81.14	69.99	65.33	71.62	56.02	66.58	67.27	29.04
RvG-Tree [17]	TPAMI19	ResNet-101/LSTM	75.06	78.61	69.85	63.51	67.45	56.66	66.95	66.51	-
CM-Att-Erase [26]	CVPR19	ResNet-101/LSTM	78.35	83.14	71.32	68.09	73.65	58.03	67.99	68.67	-
Ref-NMS[5]	AAAI21	ResNet-101/GRU	80.70	84.00	76.04	68.25	73.68	59.42	70.55	70.55	-
One-stage:											
FAOA [46]	ICCV19	DarkNet-53/BERT	72.54	74.35	68.50	56.81	60.23	49.60	61.33	60.36	60.67
ReSC-Large [47]	ECCV20	DarkNet-53/BERT	77.63	80.45	72.30	63.59	68.36	56.81	67.30	67.20	64.60
MCN [30]	CVPR20	DarkNet-53/GRU	80.08	82.29	74.98	67.16	72.86	57.31	66.46	66.01	-
RealGIN [54]	TNNLS21	DarkNet-53/GRU	77.25	78.70	72.10	62.78	67.17	54.21	62.75	62.33	-
PLV-FPN [24]	TIP22	ResNet-101/BERT	81.93	84.99	76.25	71.20	77.40	61.08	70.45	71.08	71.77
Transformer-based:											
TransVG [9]	ICCV21	ResNet-101/BERT	81.02	82.72	78.35	64.82	70.70	56.94	68.67	67.73	70.73
RefTR [23]	NeurIPS21	ResNet-101/BERT	82.23	85.59	76.57	71.58	75.96	62.16	69.41	69.40	71.42
PFOS [42]	TMM22	ResNet-101/BERT	78.44	81.94	73.61	65.86	72.43	55.26	67.89	67.63	67.90
Word2Pix [53]	TNNLS22	ResNet-101/BERT	81.20	84.39	78.12	69.74	76.11	61.24	70.81	71.34	-
SeqTR [55]	ECCV22	DarkNet-53/GRU	81.23	85.00	76.08	68.82	75.37	58.78	71.35	71.58	69.66
QRNet [48]	CVPR22	Swin-S/BERT	84.01	85.85	82.34	72.94	76.17	63.81	71.89	73.03	74.61
M-DGT [7]	CVPR22	ResNet-101/BERT	85.37	84.82	87.11	70.02	72.26	68.92	79.21	79.06	-
LADS [41]	AAAI23	ResNet-50/BERT	82.85	86.67	78.57	71.16	77.64	59.82	71.56	71.66	71.08
Ours:											
ScanFormer	-	Unified Transformer	83.40	85.86	78.81	72.96	77.57	62.50	74.10	74.14	68.85

Table 1. Comparison with state-of-the-art methods on RefCOCO [49], RefCOCO+ [49], RefCOCOg [33] and ReferItGame [20].

Method	RefCOCO		RefCOCO+		RefCOCOg
	testA	testB	testA	testB	test
ViLBERT [29]	-	-	78.52	62.61	-
UNITER_L [8]	87.04	74.17	81.45	66.70	75.77
VILLA_L [13]	87.48	74.84	81.54	66.84	76.71
MDETR [1]	89.58	81.41	84.09	70.62	80.89
OFA_B [2]	90.67	83.30	87.15	74.29	82.31
ScanFormer	89.99	82.89	84.04	70.63	82.75

Table 2. Comparison with the large-scale pre-training methods.

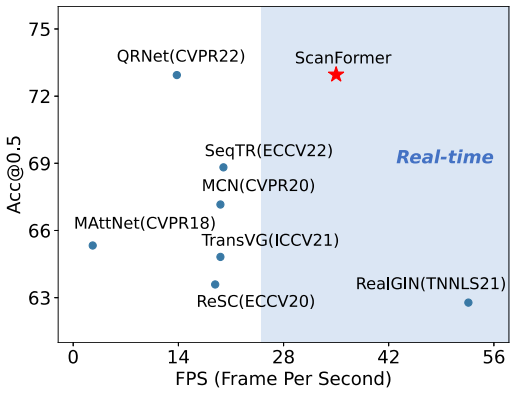

Figure 4. Comparison of the performance and inference speed on the val set of RefCOCO+ [49]. The real-time speed threshold is set to 25 FPS and all inference speeds all tested on the 1080 Ti.

intersection-over-union (IoU) between the predicted bounding box and the ground truth is greater than 0.5, it indicates that the predicted bounding box is correct.

4.2. Implementation Details

Training. Unlike conventional methods [9, 48, 55] that require additional visual encoder (such as ResNet [14], Swin [27]) and linguistic encoder (such as LSTM [16], BERT [10]), the proposed model extract visual and linguistic features using one unified Transformer [43], which is initialized from the ViLT [21] pre-training weights. The resolution of the input image is resized to 640×640, and the referring expressions are truncated or padded to a length of 40. Data augmentation operations during training include random color space jittering, Gaussian blur, random horizontal flipping, random cropping, and random resizing. We set $\lambda_{L1}^{l} = \lambda_{giou}^{l} = 4^{l-2}$ in Eq. (7), and $\lambda_{sparse} = 0.05$ and $\beta^{l} = 2^{-l}$ in Eq. (8). The model is optimized end-to-end for 80 epochs using AdamW [28], with a batch size of 384, and weight decay set to $1e^{-4}$. The learning rate is gradually increased to $1.5e^{-4}$ in the first 800 iterations using a warm-up strategy, and then the learning rate is decayed with a linear strategy. To test the performance improvement brought by large-scale pre-training, we also pre-train ScanFormer for 40 epochs on the large-scale pre-training dataset and then fine-tune the pre-trained model on the specific data set for 20 epochs. We implement the framework using PyTorch and conduct experiments using NVIDIA A100 GPUs.

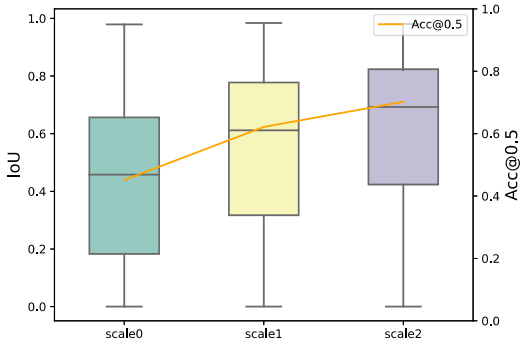

Figure 5. The Acc@0.5 and IoUs between predicted bounding boxes and ground truth of three scales, which are evaluated on the val set of RefCOCOg[33].

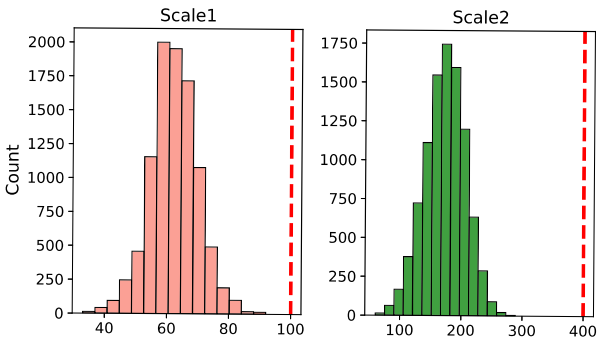

Figure 6. Distribution of the number of selected tokens from different scales in RefCOCOg [33]. The red dotted line indicates the total number of tokens for the corresponding scale.

Inference. In the inference stage, each input sample consists of an image and a referring expression, where the image is resized to 640×640, and the maximum length of the referring expression is 40. Our framework can directly output the bounding boxes specified by referring expressions without any post-processing operations.

4.3. Comparisons with State-of-the-art Methods

To verify the effectiveness of the ScanFormer proposed in this paper, we compare with state-of-the-art methods on widely used datasets, *i.e.,* RefCOCO [49], RefCOCO+ [49], RefCOCOg [31], and ReferItGame [20].

Concretely, we compare the performance of our ScanFormer with state-of-the-art REC methods, including two-stage methods [5, 17, 26, 50], one-stage methods [24, 30, 46, 47, 54], and transformer-based methods [7, 9, 23, 41, 42, 48, 53, 55]. The comparison results are shown in Tab. 1. It can be observed that our ScanFormer achieves a significant performance improvement compared to state-of-the-art one-stage method PLV-FPN [24] and two-stage method Ref-NMS [5]. Compared to state-of-the-art transformer-based methods, such as LADS [41], SeqTR [55], and Word2Pix [53], it can be found that the proposed ScanFormer also achieved good performance. In particular, compared to QRNet [48], our ScanFormer achieves comparable performance. In addition, unlike previous methods that use additional visual and linguistic backbones [10, 14, 16, 27], ScanFormer only utilizes a unified Transformer to achieve accurate language-to-vision localization.

We also compare ScanFormer with state-of-the-art large-scale pre-training methods, *i.e.* MDETR [1] and OFA [2], as shown in Tab. 2. Compared with training directly on specific datasets, large-scale pre-training greatly improves the performance of the model. In addition, ScanFormer achieves comparable performance to MDETR and OFA, but the unified Transformer structure is simpler.

Furthermore, we compare the performance and inference speed of state-of-the-art methods with our ScanFormer on the RefCOCO+ val set, as shown in Fig. 4. The inference speed is tested on 1080 Ti, and the proposed ScanFormer achieves a real-time inference speed of 34.9 FPS. Compared with state-of-the-art methods [9, 30, 48, 54, 55], we achieve high accuracy and fast inference speed, benefiting from the unified Transformer. In particular, compared to QRNet [48] with comparable performance, Scanformer achieves an inference speed of more than twice as fast.

4.4. Early Exit in Image Pyramid

Considering that our ScanFormer iteratively conducts visual perception from coarse granularity to fine granularity, we show the prediction results of the model at different scales, as shown in Figure Fig. 5. It can be observed that as the scale level increases, the performance of the model improves significantly, and better bounding box location can be predicted, as shown by the increasing IoU (Intersection over Union) values. The model can achieve acceptable performance even at the smallest scale. We also presented the distribution of IoU between predicted boxes and ground-truth boxes at different scales. It can be found that as the number of iterations increases, the upper and lower bounds of the iou distribution are significantly improved.

Although the gradually improved performance, it is not trivial to select a proper metric to decide when to stop iteration across the scale pyramids like [45]. We made a preliminary attempt by adding an extra branch sibling to the regression head to predict the exit metrics, *e.g.* IoU, GIoU, or the variance of the predicted bounding box [15]. The experimental results are not ideal, where the predicted exit metrics have a poor correlation with localization accuracy. So we left the early exit metric for further exploration.

4.5. Qualitative Results

We propose to reduce computational overhead by replacing not-selected tokens with constant tokens and then merging

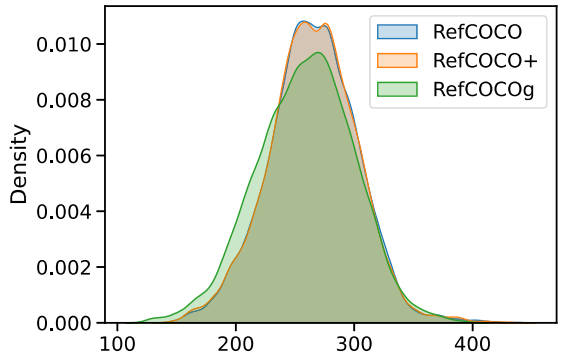

Figure 7. Distribution of the number of selected tokens for samples from RefCOCO [49], RefCOCO+ [49], and RefCOCOg [33].

them. According to Fig. 6, massive tokens are merged in Scale 1 and Scale 2. The distributions in Scale 0 are not visualized as all the tokens are selected. There are 40 and 220 tokens replaced with constant tokens on average in Scale 1 and Scale 2, respectively. Token merging can remove 39 and 219 tokens by merging them into one. Therefore, token merging can increase speed and significantly reduce FLOPs. We also visualize the distributions of the number of selected tokens per sample in Fig. 7. For some samples in RefCOCOg [33], only 100 tokens are selected for predicting the localization. Relative to all 400 tokens, each sample has an average of 270 selected tokens participating in the calculation. In addition, causal attention across scales can further reduce computational overhead.

The qualitative results are shown in Fig. 8. It can be observed that our model can successfully locate the referred objects, and the regressed bounding boxes are gradually refined with the increasing scales. Furthermore, based on images reconstructed from selected patches, the model can leave regions with low texture or no relation to the reference representation at coarse scales.

5. Conclusions and Liminations

In this paper, we explore an efficient vision-language Transformer and propose a coarse-to-fine iterative perception framework, called ScanFormer. It can continuously discard linguistic-irrelevant redundant visual regions in each iteration to enhance the efficiency of the model. Extensive experiments on widely used datasets verify the effectiveness of our ScanFormer, which can strike a balance between accuracy and efficiency. The limitations of our method are two-fold: (1) The current method localizes the referred objects through all the scales, and a flexible early exit method can be studied to further improve model efficiency. (2) The current framework only predicts one target object at a time, limiting its extension to phrase grounding.

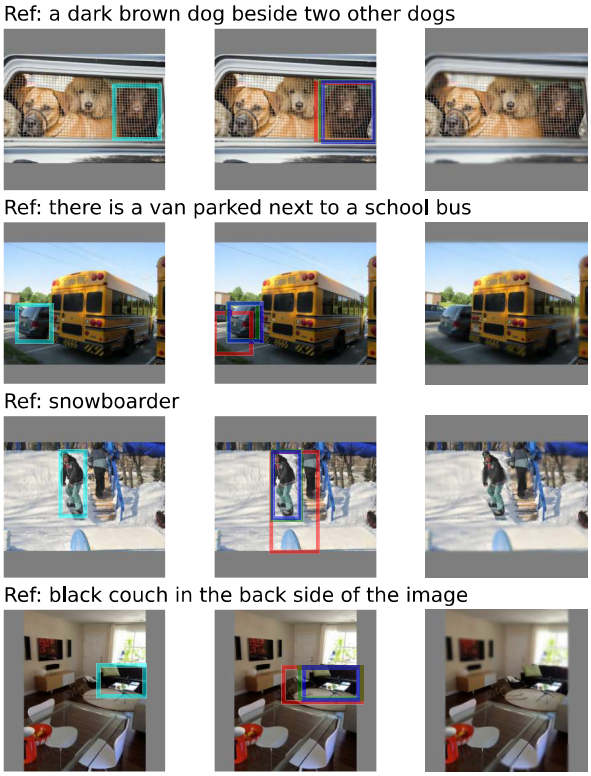

Figure 8. Visualization of examples from the val set of RefCOCOg [33]. From left to right: the input image and ground truth bounding box, the detection results at three scales (red, green, and blue represent the result from scale 0, 1, and 2, respectively), and the image reconstructed from selected patches from different scales.

6. Acknowledgments

This work is supported in part by National Natural Science Foundation of China under Grant U20A20222, National Science Foundation for Distinguished Young Scholars under Grant 62225605, CCF-Zhipu AI Large Model Fund (CCF-Zhipu202302), Zhejiang Key Research and Development Program under Grant 2023C03196, Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020016, SupreMind, and The Ng Teng Fong Charitable Foundation in the form of ZJU-SUTD IDEA Grant, 188170-11102.

References

[1] Mdetr-modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021. 6, 7

[2] Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, 2022. 6, 7

[3] Transvg++: End-to-end visual grounding with language conditioned vision transformer. *IEEE TPAMI*, 2023. 2

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2

[5] Long Chen, Wenbo Ma, Jun Xiao, Hanwang Zhang, and Shih-Fu Chang. Ref-nms: Breaking proposal bottlenecks in two-stage referring expression grounding. In *AAAI*, pages 1036–1044, 2021. 2, 6, 7

[6] Mengzhao Chen, Mingbao Lin, Ke Li, Yunhang Shen, Yongjian Wu, Fei Chao, and Rongrong Ji. Cf-vit: A general coarse-to-fine method for vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7042–7052, 2023. 1, 2

[7] Sijia Chen and Baochun Li. Multi-modal dynamic graph transformer for visual grounding. In *CVPR*, pages 15534–15543, 2022. 6, 7

[8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, pages 104–120, 2020. 6

[9] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *ICCV*, pages 1769–1779, 2021. 1, 2, 5, 6, 7

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 2

[12] Hugo Jair Escalante, Carlos A Hernández, Jesus A Gonzalez, Aurelio López-López, Manuel Montes, Eduardo F Morales, L Enrique Sucar, Luis Villasenor, and Michael Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer vision and image understanding*, 114(4): 419–428, 2010. 5

[13] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *NeurIPS*, 33:6616–6628, 2020. 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 6, 7

[15] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, pages 2888–2897, 2019. 7

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 6, 7

[17] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE TPAMI*, 2019. 2, 6, 7

[18] Wenhui Jiang, Minwei Zhu, Yuming Fang, Guangming Shi, Xiaowei Zhao, and Yang Liu. Visual cluster grounding for image captioning. *IEEE TIP*, 2022. 1

[19] Łukasz Kaiser and Samy Bengio. Discrete autoencoders for sequence models. *arXiv preprint arXiv:1801.09797*, 2018. 4

[20] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 2, 5, 6, 7

[21] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, pages 5583–5594, 2021. 6

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 5

[23] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. In *NeurIPS*, 2021. 6, 7

[24] Yue Liao, Aixi Zhang, Zhiyuan Chen, Tianrui Hui, and Si Liu. Progressive language-customized visual feature learning for one-stage visual grounding. *IEEE TIP*, 31:4266–4277, 2022. 6, 7

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 5

[26] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *CVPR*, pages 1950–1959, 2019. 2, 6, 7

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 1, 6, 7

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 32, 2019. 6

[30] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, pages 10034–10043, 2020. 1, 2, 6, 7

[31] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 7

[32] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 2

[33] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, pages 792–807, 2016. 5, 6, 7, 8

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[35] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 5

[36] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2, 5

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 2, 5

[38] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 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*, pages 658–666, 2019. 5

[39] Xuejian Rong, Chucai Yi, and Yingli Tian. Unambiguous scene text segmentation with referring expression comprehension. *IEEE TIP*, 29:591–601, 2019. 1

[40] Amaia Salvador, Xavier Giró-i Nieto, Ferran Marqués, and Shin'ichi Satoh. Faster r-cnn features for instance search. In *CVPR*, pages 9–16, 2016. 1

[41] Wei Su, Peihan Miao, Huanzhang Dou, Yongjian Fu, and Xi Li. Referring expression comprehension using language adaptive inference. *arXiv preprint arXiv:2306.04451*, 2023. 2, 6, 7

[42] Mengyang Sun, Wei Suo, Peng Wang, Yanning Zhang, and Qi Wu. A proposal-free one-stage framework for referring expression comprehension and generation via dense cross-attention. 2022. 6, 7

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 6

[44] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 1, 2

[45] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, 34:11960–11973, 2021. 1, 2, 7

[46] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *ICCV*, pages 4683–4693, 2019. 1, 2, 6, 7

[47] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *ECCV*, pages 387–404. Springer, 2020. 2, 6, 7

[48] Jiabo Ye, Junfeng Tian, Ming Yan, Xiaoshan Yang, Xuwu Wang, Ji Zhang, Liang He, and Xin Lin. Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding. In *CVPR*, pages 15502–15512, 2022. 1, 2, 5, 6, 7

[49] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 2, 5, 6, 7, 8

[50] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, pages 1307–1315, 2018. 2, 6, 7

[51] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. In *IJCAI*, pages 1114–1120, 2018. 2

[52] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022. 1, 2

[53] Heng Zhao, Joey Tianyi Zhou, and Yew-Soon Ong. Word2pix: Word to pixel cross-attention transformer in visual grounding. *TNNLS*, 2022. 1, 2, 6, 7

[54] Yiyi Zhou, Rongrong Ji, Gen Luo, Xiaoshuai Sun, Jinsong Su, Xinghao Ding, Chia-Wen Lin, and Qi Tian. A real-time global inference network for one-stage referring expression comprehension. *TNNLS*, 2021. 2, 6, 7

[55] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *ECCV*, pages 598–615. Springer, 2022. 1, 2, 5, 6, 7

[56] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, pages 4995–5004, 2016. 1