TransVG: End-to-End Visual Grounding with Transformers

Jiajun Deng[†], Zhengyuan Yang[‡], Tianlang Chen[‡], Wengang Zhou[†,§], Houqiang Li[†,§]

[†] CAS Key Laboratory of GIPAS, University of Science and Technology of China, Hefei, China

[‡] University of Rochester

[§] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

dengjj@mail.ustc.edu.cn

Abstract

In this paper, we present a neat yet effective transformer-based framework for visual grounding, namely TransVG, to address the task of grounding a language query to the corresponding region onto an image. The state-of-the-art methods, including two-stage or one-stage ones, rely on a complex module with manually-designed mechanisms to perform the query reasoning and multi-modal fusion. However, the involvement of certain mechanisms in fusion module design, such as query decomposition and image scene graph, makes the models easily overfit to datasets with specific scenarios, and limits the plenitudinous interaction between the visual-linguistic context. To avoid this caveat, we propose to establish the multi-modal correspondence by leveraging transformers, and empirically show that the complex fusion modules (e.g., *modular attention network, dynamic graph, and multi-modal tree) can be replaced by a simple stack of transformer encoder layers with higher performance. Moreover, we re-formulate the visual grounding as a direct coordinates regression problem and avoid making predictions out of a set of candidates* (i.e., *region proposals or anchor boxes). Extensive experiments are conducted on five widely used datasets, and a series of state-of-the-art records are set by our TransVG. We build the benchmark of transformer-based visual grounding framework and make the code available at* https://github.com/djiajunustc/TransVG.

1. Introduction

Visual grounding (also known as referring expression comprehension [31, 60], phrase localization [23, 38], and natural language object retrieval [21, 25]) aims to predict the location of a region referred by the language expression onto an image. The evolution of this technique is of great potential to provide an intelligent interface for the natural language expression of human beings and the visual components of the physical world. Existing methods addressing this task can be broadly grouped into the two-stage and one-stage pipelines shown in Figure 1. In specific, the two-stage approaches [31, 34, 46, 60] first generate a set of sparse region proposals and then exploit region-expression matching to find the best one. The one-stage approaches [9, 27, 56] perform visual-linguistic fusion at intermediate layers of an object detector and output the box with the maximal score over pre-defined dense anchors.

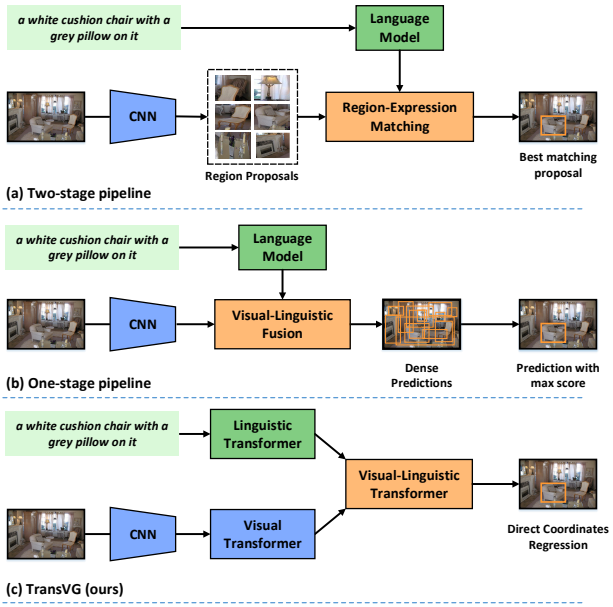

Figure 1. A comparison of (a) two-stage pipeline, (b) one-stage pipeline, and (c) our proposed TransVG framework. TransVG performs intra-modality and inter-modality relation reasoning with a stack of transformer layers in a homogeneous way, and grounds the object by directly regressing the box coordinates.

Multi-modal fusion and reasoning is widely studied in the literature [1, 35, 49, 54, 65], and it is the core problem in visual grounding. In general, the early two-stage and one-stage methods address multi-modal fusion in a simple way. Concretely, the two-stage Similarity Net [46] measures the

similarity between region and expression embedding with an MLP, and the one-stage FAOA [56] encodes the language vector to visual feature by direct concatenation. These simple designs are efficient but lead to sub-optimal results, especially on long and complex language expressions. Following studies have proposed diverse architectures to improve the performance. Among two-stage methods, modular attention network [59], various graphs [48, 52, 53], and multi-modal tree [28] are designed to better model the multi-modal relationships. The one-stage method [55] has also explored better query modeling by proposing a multi-round fusion module.

Despite the effectiveness, these complicated fusion modules are built on certain pre-defined structures of language queries or image scenes, inspired by the human prior. Typically, the involvement of manually-designed mechanisms in fusion module makes the models overfit to specific scenarios, such as certain query lengths and query relationships, and limits the plenitudinous interaction between visual-linguistic contexts. Moreover, even though the ultimate goal of visual grounding is to localize the referred object, most of the previous methods ground the queried object in an indirect fashion. They generally define surrogate problems of language-guided candidates prediction, selection, and refinement. Typically, the candidates are sparse region proposals [60, 31, 46] or dense anchors [56], from which the best region is selected and refined to get the final grounding box. Since these methods' predictions are made out of candidates, the performance is easily influenced by the prior knowledge to generate proposals (or pre-defined anchors) and by the heuristics to assign targets to candidates.

In this study, we explore an alternative approach to avoid the aforementioned problems. Formally, we introduce a neat and novel transformer-based framework, namely TransVG, to effectively address the task of visual grounding. We empirically show that the structurized fusion modules can be replaced by a simple stack of transformer encoder layers. Particularly, the core component of transformers (*i.e.*, attention layer) is ready to establish intra-modality and inter-modality correspondence across visual and linguistic inputs, despite that we do not pre-define any specific fusion mechanism. Besides, we find that directly regressing the box coordinates works better than previous methods to ground the queried object indirectly. Our TransVG directly outputs 4-dim coordinates to ground the object instead of making predictions based on a set of candidate boxes.

The pipeline of our proposed TransVG is illustrated in Figure 1(c). We first feed the RGB image and language expression into two sibling branches. The visual transformer and linguistic transformer are applied in these two branches to model the global cues in vision and language domains, respectively. Then, the abstracted visual tokens and linguistic tokens are fused, and the visual-linguistic transformer is exploited to perform cross-modal relation reasoning. Finally, the box coordinates of a referred object are directly regressed to make grounding. We benchmark our framework on five prevalent visual grounding datasets, including ReferItGame [23], Flickr30K Entities [38], RefCOCO [60], RefCOCO+ [60], RefCOCOg [31], and our method sets a series of state-of-the-art records. Remarkably, our proposed TransVG achieves 70.73%, 79.10% and 78.35% on the test set of ReferItGame, Flickr30K and RefCOCO datasets, with 6.13%, 5.80%, 6.05% absolute improvements over the strongest competitors.

In summary, we make three-fold contributions:

- We propose the first transformer-based framework for visual grounding, which holds neater architecture yet achieves better performance than the prevalent one-stage and two-stage frameworks.
- We present an elegant view of capturing intra- and inter-modality context homogeneously by transformers, and formulating visual grounding as a direct coordinates regression problem.
- We conduct extensive experiments to validate the merits of our method, and show significantly improved results on several prevalent benchmarks.

2. Related Work

2.1. Visual Grounding

Recent advances in visual grounding can be broadly categorized into two directions, *i.e.*, two-stage methods [19, 20, 28, 46, 48, 52, 59, 63, 68] and one-stage methods [9, 27, 42, 55, 56]. We briefly review them in the following.

Two-stage Methods. Two-stage approaches are characterized by generating region proposals in the first stage and then leveraging the language expression to select the best matching region in the second stage. Generally, the region proposals are generated using either unsupervised methods [37, 46] or a pre-trained object detector [59, 63]. The training loss of either binary classification [46, 64] or maximum-margin ranking [31, 34, 47] is applied in the second stage to maximize the similarity between the positive object-query pair. Pioneer studies [31, 47, 60] obtain good results with the two-stage framework. The early work MattNet [59] introduces the modular design and improves the grounding accuracy by better modeling the subject, location, and relation-related language description. Some recent studies further improve the two-stage methods by better modeling the object relationships [28, 48, 52], enforcing correspondence learning [29], or making use of phrase co-occurrences [3, 7, 13].

One-stage Methods. One-stage approaches get rid of the computation-intensive object proposal generation and region feature extraction in the two-stage paradigm. Instead, the linguistic context is densely fused with the visual

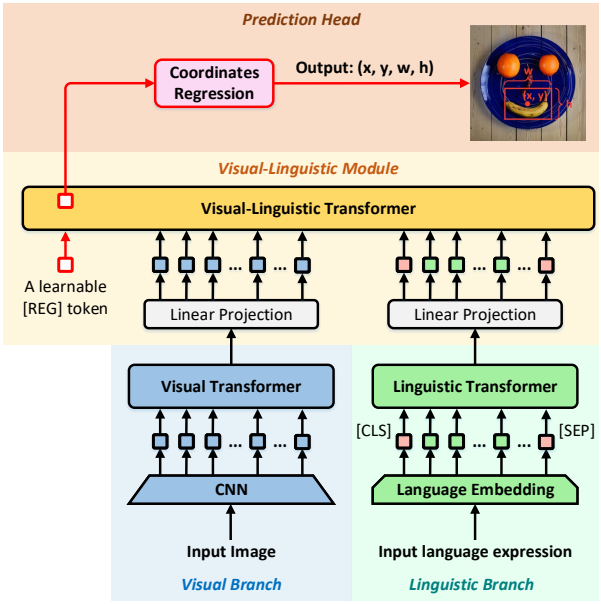

Figure 2. An overview of our proposed TransVG framework. It consists of four main components: (1) a visual branch, (2) a linguistic branch, (3) a visual-linguistic fusion module, and (4) a prediction head to regress the box coordinates.

features, and the language-attended feature maps are further leveraged to perform bounding box prediction in a sliding-window manner. The pioneering work FAOA [56] encodes the text expression into a language vector, and fuses the language vector into the YOLOv3 detector [40] to ground the referred instance. RCCF [27] formulates the visual grounding problem as a correlation filtering process [4, 17], and picks the peak value of the correlation heatmap as the center of target objects. The recent work ReSC [55] devises a recursive sub-query construction module to address the limitations of FAOA [56] on grounding complex queries.

2.2. Transformer

Transformer is first proposed in [45] to tackle the neural machine translation (NMT). The primary component of a transformer layer is the attention module, which scans through the input sequence in parallel and aggregates the information of the whole sequence with adaptive weights. Compared to the recurrent units in RNNs [18, 32, 44], the attention mechanism exhibits better performance in processing long sequences. This superiority has attracted a surge of research interest in applications of transformers in NLP tasks [11, 12, 39, 66] and speech recognition [33, 50].

Transformer in Vision Tasks. Inspired by the great success of transformers in neural machine translation, a series of transformers [5, 6, 8, 14, 22, 51, 62, 67] applied to vision tasks have been proposed. The infusive work DETR [5] formulates object detection as a set prediction problem. It introduces a small set of learnable object queries, reasons global context and object relations with attention mechanism, and outputs the final set of predictions in parallel. ViT [14] shows that a pure transformer can achieve excellent performance on image classification tasks. More recently, a pre-trained image processing transformer (IPT) is introduced in [6] to address the low-level vision problems, *e.g.*, denoising, super-resolution and deraining.

Transformer in Vision-Language Tasks. Motivated by the powerful pre-trained model of BERT [12], some researchers start to investigate visual-linguistic pre-training (VLP) [10, 26, 30, 43, 57] to jointly represent images and texts. In general, these models take the object proposals and text as inputs, and devise several transformer encoder layers for joint representation learning. Plenty of pre-training tasks are introduced, including image-text matching (ITM), word-region alignment (WRA), masked language modeling (MLM), masked region modeling (MRM), *etc.*

Although with similar base units (*i.e.* transformer encoder layers), the goal of VLP is to learn a generalizable vision-language representation with large-scale data to facilitate down-stream tasks. In contrast, we focus on developing a novel transformer-based visual grounding framework, and learning to perform homogeneous multi-modal reasoning with a small amount of visual grounding data.

3. Transformers for Visual Grounding

In this work, we present Transformers for Visual Grounding (TransVG), a novel framework for the visual grounding task based on a stack of transformer encoders with direct box coordinates prediction. As shown in Figure 2, given an image and a language expression as inputs, we first separate them into two sibling branches, *i.e.*, a visual branch and a linguistic branch, to generate visual and linguistic feature embedding. Then, we put the multi-modal feature embedding together and append a learnable token (named [REG] token) to construct the inputs of visual-linguistic fusion modules. The visual-linguistic transformer homogeneously embeds the input tokens from different modalities into a common semantic space by modeling intra-modality and inter-modality context with the self-attention mechanism. Finally, the output state of the [REG] token is leveraged to directly predict the 4-dim coordinates of a referred object in the prediction head.

In the following subsections, we first review the preliminary for transformer and then elaborate our designs of transformers for visual grounding.

3.1. Preliminary

Before detailing the architecture of TransVG, we briefly review the conventional transformer proposed in [45] for machine translation. The core component in a transformer is the attention mechanism. Given the query embedding $\boldsymbol{f}^q$,

key embedding $\boldsymbol{f}^k$ and value embedding $\boldsymbol{f}^v$, the output of a single-head attention layer is computed as:

$$\text{Attn}(\boldsymbol{f}^q, \boldsymbol{f}^k, \boldsymbol{f}^v) = \text{softmax}(\frac{\boldsymbol{f}^q \boldsymbol{f}^k}{\sqrt{d^k}}) \cdot \boldsymbol{f}^v, \quad (1)$$

where d^k is the channel dimension of $\boldsymbol{f}^k$. Similar to classic neural sequence transduction models, the conventional transformer has an encoder-decoder structure. However, in our approach, we only use transformer encoder layers.

Concretely, each transformer encoder layer has two sub-layers, *i.e.*, a multi-head self-attention layer and a simple feed forward network (FFN). The multi-head attention is a variant of single-head attention (as in Function 1), and self-attention indicates the query, key and value are from the same embedding set. FFN is an MLP composed of fully connected layers and ReLU activation layers.

In the transformer encoder layer, each sub-layer is put into a residual structure, where layer normalization [2] is performed after the residual connection. Let us denote the input as $\boldsymbol{x}_n$, the procedure in a transformer encoder layer is:

$$\boldsymbol{x}'_n = \text{LN}(\boldsymbol{x}_n + \mathcal{F}_{\text{MSA}}(\boldsymbol{x}_n)), \quad (2)$$

$$\boldsymbol{x}_{n+1} = \text{LN}(\boldsymbol{x}'_n + \mathcal{F}_{\text{FFN}}(\boldsymbol{x}'_n)), \quad (3)$$

where $\text{LN}(\cdot)$ indicates layer normalization, $\mathcal{F}_{\text{MSA}}(\cdot)$ is the multi-head self-attention layer, and $\mathcal{F}_{\text{FFN}}(\cdot)$ represents the feed forward network.

3.2. TransVG Architecture

As depicted in Figure 2, there are four main components in TransVG: (1) a visual branch, (2) a linguistic branch, (3) a visual-linguistic fusion module, and (4) a prediction head.

Visual Branch. The visual branch starts with a convolutional backbone network, followed by the visual transformer. We exploit the commonly used ResNet [16] as the backbone network. The visual transformer is composed of a stack of 6 transformer encoder layers. Each transformer encoder layer includes a multi-head self-attention layer and an FFN. There are 8 heads in the multi-head attention layer, and 2 FC layers followed by ReLU activation layers in the FFN. The output channel dimensions of these 2 FC layers are 2048 and 256, respectively.

Given an image $\boldsymbol{z}_0 \in \mathbb{R}^{3\times H_0 \times W_0}$ as the input of this branch, we exploit the backbone network to generate a 2D feature map $\boldsymbol{z} \in \mathbb{R}^{C\times H\times W}$. Typically, the channel dimension C is 2048, and the width and height of the 2D feature map are $\frac{1}{32}$ of the original image size ($H = \frac{H_0}{32}$, $W = \frac{W_0}{32}$). Then, we leverage a 1×1 convolutional layer to reduce the channel dimension of $\boldsymbol{z}$ to $C_v = 256$ and obtain $\boldsymbol{z}' \in \mathbb{R}^{C_v \times H \times W}$. Since the input of a transformer encoder layer is expected to be a sequence of 1D vectors, we further flatten $\boldsymbol{z}'$ into $\boldsymbol{z}_v \in \mathbb{R}^{C_v \times N_v}$, where $N_v = H \times W$ is the number of input tokens. To make the visual transformer sensitive to the original 2D positions of input tokens, we follow [5, 36] to utilize sine spatial position encodings as the supplementary of visual feature. Concretely, the position encodings are added with the query and key embedding at each transformer encoder layer. The visual transformer conducts global vision context reasoning in parallel, and outputs the advanced visual embedding $\boldsymbol{f}_v$, which shares the same shape as $\boldsymbol{z}_v$.

Linguistic Branch. The linguistic branch is a sibling to the visual branch. Our linguistic branch includes a token embedding layer and a linguistic transformer. To make the best of the pre-trained BERT model [12], the architecture of this branch follows the design of the basic model of BERT series. Typically, there are 12 transformer encoder layers in the linguistic transformer. The output channel dimension of the linguistic transformer is $C_l = 768$.

Given a language expression as the input of this branch, We first convert each word ID into a one-hot vector. Then, in the token embedding layer, we tokenize each one-hot vector into a language token by looking up the token table. We follow the common practice in machine translation [11, 12, 39, 45] to append a [CLS] token and a [SEP] token at the beginning and end positions of the tokenized language expression. After that, we take the language tokens as inputs of the linguistic transformer, and generate the advanced language embedding $\boldsymbol{f}_l \in \mathbb{R}^{C_l \times N_l}$, where N_l is the number of language tokens.

Visual-linguistic Fusion Module. As the core component in our model to fuse the multi-modal context, the architecture of the visual-linguistic fusion module (abbreviated as V-L module) is simple and elegant. Specifically, the V-L module includes two linear projection layers (one for each modality) and a visual-linguistic transformer (with a stack of 6 transformer encoder layers).

Given advanced visual tokens $\boldsymbol{f}_v \in \mathbb{R}^{256\times N_v}$ out of the visual branch and advanced linguistic tokens $\boldsymbol{f}_l \in \mathbb{R}^{768\times N_l}$ out of the linguistic branch, we apply a linear projection layer to project them into embedding with same channel dimension. We denote the projected visual embedding and linguistic embedding as $\boldsymbol{p}_v \in \mathbb{R}^{C_p \times N_v}$ and $\boldsymbol{p}_l \in \mathbb{R}^{C_p \times N_l}$, where $C_p = 256$. Then, we pre-append a learnable embedding (namely a [REG] token) to $\boldsymbol{p}_v$ and $\boldsymbol{p}_l$, and formulate the joint input tokens of the visual-linguistic transformer as:

$$\boldsymbol{x}_0 = [\underbrace{p_v^1, p_v^2, \cdots, p_v^{N_v}}_{\text{visual tokens } \boldsymbol{p}_v}, \overbrace{p_l^1, p_l^2, \cdots, p_l^{N_l}}^{\text{linguistic tokens } \boldsymbol{p}_l}, p_r], \quad (4)$$

where $p_r \in \mathbb{R}^{C_p \times 1}$ represents the [REG] token. The [REG] token is randomly initialized at the beginning of the training stage and optimized with the whole model.

After obtaining the input $\boldsymbol{x}_0 \in \mathbb{R}^{C_p \times (N_v + N_l + 1)}$ in the joint embedding space as described above, we apply the

visual-linguistic transformer to embed $\boldsymbol{x}_0$ into a common semantic space by performing intra- and inter-modality relation reasoning in a homogeneous way. To retain the positional and modal information, we add learnable position encodings to the input of each transformer encoder layer.

Thanks to the attention mechanism, the correspondence can be freely established between each pair of tokens from the joint entities, regardless of their modality. For example, a visual token can attend to a visual token, and it can also freely attend to a linguistic token. Typically, the output state of the [REG] token develops a consolidated representation enriched by both visual and linguistic context, and is further leveraged for box coordinates prediction.

Prediction Head. We leverage the output state of [REG] token from the V-L module as the input of our prediction head. To perform box coordinates prediction, we attach a regression block to the [REG] token. The regression block is implemented by an MLP with two ReLU activated 256-dim hidden layers and a linear output layer. The output of the prediction head is the 4-dim box coordinates.

3.3. Training Objective

Unlike many previous methods that ground referred objects based on a set of candidates (*i.e.*, region proposals in two-stage methods and anchor boxes in one-stage methods), TransVG directly infers a 4-dim vector as the coordinates of the box to be grounded. This simplifies the process of target assignment and positive/negative examples mining at the training stage, but it also involves the scale problem. Specifically, the widely used smooth L1 loss tends to be a large number when we try to predict a large box, while tends to be small when we try to predict a small one, even if their predictions have similar relative errors.

To address this problem, we normalize the coordinates of the ground-truth box by the scale of the image, and involve the generalized IoU loss [41] (GIoU loss), which is not affected by the scales.

Let us denote the prediction as $\boldsymbol{b} = (x, y, w, h)$, and the normalized ground-truth box as $\hat{\boldsymbol{b}} = (\hat{x}, \hat{y}, \hat{w}, \hat{h})$. The training objective of our TransVG is:

$$\mathcal{L} = \mathcal{L}_{\text{smooth-l1}}(\boldsymbol{b}, \hat{\boldsymbol{b}}) + \lambda \cdot \mathcal{L}_{\text{giou}}(\boldsymbol{b}, \hat{\boldsymbol{b}}), \tag{5}$$

where $\mathcal{L}_{\text{smooth-l1}}(\cdot)$ and $\mathcal{L}_{\text{giou}}(\cdot)$ are the smooth L1 loss and GIoU loss, respectively. λ is the weight coefficient of GIoU loss to balance these two losses.

4. Experiments

4.1. Datasets

ReferItGame. ReferItGame [23] includes 20,000 images collected from the SAIAPR-12 dataset [15], and each image has one or a few regions with corresponding referring expressions. We follow the common practice to divide this dataset into three subsets, *i.e.*, a train set with 54,127 referring expressions, a validation set with 5,842 referring expressions and a test set with 60,103 referring expressions. We use the validation set to conduct experimental analysis and compare our method with others on the test set.

Flickr30K Entities. Flickr30K Entities [38] augments the original Flickr30K [58] with short region phrase correspondence annotations. It contains 31,783 images with 427K referred entities. We follow the previous works [38, 37, 46, 55] to separate the these images into 29,783 for training, 1000 for validation, and 1000 for testing.

RefCOCO/ RefCOCO+/ RefCOCOg. RefCOCO [60] includes 19,994 images with 50,000 referred objects. Each object has more than one referring expression, and there are 142,210 referring expressions in this dataset. The samples in RefCOCO are officially split into a train set with 120,624 expressions, a validation set with 10,834 expressions, a testA set with 5,657 expressions and a testB set with 5,095 expressions. Similarly, RefCOCO+ [60] contains 19,992 images with 49,856 referred objects and 141,564 referring expressions. It is also officially split into a train set with 120,191 expressions, a validation set with 10,758 expressions, a testA set with 5,726 expressions and a testB set with 4,889 expressions. RefCOCOg [31] has 25,799 images with 49,856 referred objects and expressions. There are two commonly used split protocols for this dataset. One is RefCOCOg-google [31], and the other is RefCOCOg-umd [34]. We report our performance on both RefCOCOg-google (val-g) and RefCOCOg-umd (val-u and test-u) to make comprehensive comparisons.

4.2. Implementation Details

Inputs. We set the input image size as 640×640 and the max expression length as 40. When performing image resizing, we keep the original aspect ratio of each image. The longer edge of an image is resized to 640, while the shorter one is padded to 640 with the mean value of RGB channels. Meanwhile, We cut off the language query if its length is longer than 38 (leaving one position for the [CLS] token and one position for the [SEP] token). Otherwise, we pad empty tokens after [SEP] token to make the input length equal to 40. For both the input image and language expression, the padded pixel/word is recorded with a mask and will not be involved in the computation of transformers.

Training Details. The whole architecture of our TransVG is end-to-end optimized with AdamW optimizer. We set the initial learning rate of the V-L module and prediction head to 10^{-4}, the visual branch and linguistic branch to 10^{-5}, and set weight decay to 10^{-4}. Our visual branch is initialized with the backbone and encoder of DETR model [5], and our linguistic branch is initialized with the basic BERT model [12]. For the other components, the parameters are randomly initialized with Xavier init. On all the datasets ex-

cept Flickr30K Entities, our model is trained for 90 epochs with a learning rate dropped by a factor of 10 after 60 epochs. As for the Flickr30K Entities, our model is trained for 60 epochs, with a learning rate drops after 40 epochs. We set the batch size to 64. The weight coefficient λ is set to 1. To avoid overfitting, we exploit dropout operation after the multi-head self-attention layer and the FFN of each transformer encoder layer. The dropout ratio is set to 0.1 by default. We follow the common practice in [27, 55, 56] to perform data augmentation at the training stage.

Inference. Since our TransVG directly outputs the box coordinates, there is no extra operation at the inference stage.

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

To validate the merits of our proposed TransVG, we report our performance and compare it with other state-of-the-art methods on five visual grounding benchmarks, including ReferItGame [23], Flickr30K Entities [38], RefCOCO [60], RefCOCO+ [60], and RefCOCOg [31]. We follow the standard protocol to report the performance in terms of top-1 accuracy (%). Specifically, once the Jaccard overlap between the predicted region and the ground-truth box is above 0.5, the prediction is regarded as a correct one.

ReferItGame. Table 1 shows the result comparison between state-of-the-art methods on the ReferItGame test set. We group the methods into two-stage methods, one-stage methods, and transformer-based methods. Among all the methods, TransVG achieves the best performance as the first transformer-based approach. With ResNet-50 backbone, TransVG achieves 69.76% top-1 accuracy and outperforms ZSGNet [42] with the same backbone network by 11.13%. By replacing ResNet-50 with a stronger ResNet-101, the performance boosts to 70.73%, which is 6.13% higher than the strongest competitor ReSC-Large for one-stage methods and 7.73% higher than the strongest competitor DDPN for two-stage methods, respectively.

In particular, we find the recurrent architecture in ReSC shares the same spirit with our stacking architecture in the visual-linguistic transformer that fuses the multi-modal context in multiple rounds. However, in ReSC, recurrent learning is only performed to construct the language sub-query, and this procedure is isolated from the sub-query attended visual feature modulation. In contrast, our TransVG embeds the visual and linguistic embedding into a common semantic space by homogeneously performing intra- and inter-modality context reasoning. The superiority of our performance empirically demonstrates the effectiveness of our unified visual-linguistic encoder and fusion module designs. It also validates that the complicated multi-modality fusion module can be replaced by a simple stack of transformer encoder layers.

Flickr30K Entities. Table 1 also reports the performance of our TransVG on the Flickr30K Entities test set. On this dataset, our TransVG achieves 79.10% top-1 accuracy with a ResNet-101 backbone network, surpassing the recently proposed Similarity Net [46], CITE [37], DDPN [61], ZSGNet [42], FAOA [56], and ReSC-Large [55] by a remarkable margin (*i.e.*, 5.80% absolute improvement over the previous state-of-the-art record).

Table 1. Comparisons with state-of-the-art methods on the test set of ReferItGame [23] and Flickr30K Entities [38] in terms of top-1 accuracy (%). The previous methods follow the two-stage or one-stage directions, while ours is transformer-based. We highlight the best and second best performance in the red and blue colors.

Models	Backbone	ReferItGame test	Flickr30K test
Two-stage:			
CMN [20]	VGG16	28.33	-
VC [63]	VGG16	31.13	-
MAttNet [59]	ResNet-101	29.04	-
Similarity Net [46]	ResNet-101	34.54	60.89
CITE [37]	ResNet-101	35.07	61.33
PIRC [24]	ResNet-101	59.13	72.83
DDPN [61]	ResNet-101	63.00	73.30
One-stage:			
SSG [9]	DarkNet-53	54.24	-
ZSGNet [42]	ResNet-50	58.63	63.39
FAOA [56]	DarkNet-53	60.67	68.71
RCCF [27]	DLA-34	63.79	-
ReSC-Large [55]	DarkNet-53	64.60	69.28
Transformer-based:			
TransVG (ours)	ResNet-50	**69.76**	**78.47**
TransVG (ours)	ResNet-101	**70.73**	**79.10**

RefCOCO/RefCOCO+/RefCOCOg. To further validate the effectiveness of our proposed TransVG, we also conduct experiments to report our performance on the RefCOCO, RefCOCO+ and RefCOCOg datasets. The top-1 accuracy (%) of our method, together with other state-of-the-art methods, is reported in Table 2. Our TransVG consistently achieves the best performance on the RefCOCO and RefCOCOg for all the subsets and splits. Remarkably, we achieve 78.35% on the RefCOCO testB set, 6.05% absolute improvement over the previous state-of-the-art result. When performing grounding on longer expressions (on the RefCOCOg dataset), our method also works well, which further validates our neat architecture's effectiveness to process complicated queries. On RefCOCO+, TransVG also achieves comparable performance to that with the best records. We study the failure cases and find some extreme examples whose expressions are not suitable for generating embedding with transformers. For example, a query that just tells a number "32" in the annotation degenerates our linguistic transformer to an MLP in this situation.

Among the competitors, MAttNet [59] is the most representative method that devises multi-modal fusion modules with re-defined structures (*i.e*, modular attention networks to separately model subject, location and relationship).

Table 2. Comparisons with state-of-the-art methods on RefCOCO [60], RefCOCO+ [60] and RefCOCOg [31] in terms of top-1 accuracy (%). We highlight the best and second best performance in the red and blue colors.

Models	Venue	Backbone	RefCOCO			RefCOCO+			RefCOCOg		
			val	testA	testB	val	testA	testB	val-g	val-u	test-u
Two-stage:											
CMN [20]	*CVPR'17*	VGG16	-	71.03	65.77	-	54.32	47.76	57.47	-	-
VC [63]	*CVPR'18*	VGG16	-	73.33	67.44	-	58.40	53.18	62.30	-	-
ParalAttn [68]	*CVPR'18*	VGG16	-	75.31	65.52	-	61.34	50.86	58.03	-	-
MAttNet [59]	*CVPR'18*	ResNet-101	76.65	81.14	69.99	**65.33**	**71.62**	56.02	-	66.58	67.27
LGRANs [48]	*CVPR'19*	VGG16	-	76.60	66.40	-	64.00	53.40	61.78	-	-
DGA [52]	*ICCV'19*	VGG16	-	78.42	65.53	-	69.07	51.99	-	-	63.28
RvG-Tree [19]	*TPAMI'19*	ResNet-101	75.06	78.61	69.85	63.51	67.45	56.66	-	66.95	66.51
NMTree [28]	*ICCV'19*	ResNet-101	76.41	81.21	70.09	**66.46**	**72.02**	**57.52**	64.62	65.87	66.44
One-stage:											
SSG [9]	*arXiv'18*	DarkNet-53	-	76.51	67.50	-	62.14	49.27	47.47	58.80	-
FAOA [56]	*ICCV'19*	DarkNet-53	72.54	74.35	68.50	56.81	60.23	49.60	56.12	61.33	60.36
RCCF [27]	*CVPR'20*	DLA-34	-	81.06	71.85	-	70.35	56.32	-	-	65.73
ReSC-Large [55]	*ECCV'20*	DarkNet-53	77.63	80.45	72.30	63.59	68.36	56.81	63.12	67.30	67.20
Transformer-based:											
TransVG (ours)	-	ResNet-50	**80.32**	**82.67**	**78.12**	63.50	68.15	55.63	**66.56**	**67.66**	**67.44**
TransVG (ours)	-	ResNet-101	**81.02**	**82.72**	**78.35**	64.82	70.70	**56.94**	**67.02**	**68.67**	**67.73**

When we compare our model with MAttNet in Table 1 and Table 2, we can find that MAttNet shows comparable results to our TransVG on RefCOCO/RefCOCO+/RefCOCOg, but lags behind our TransVG on RefeItGame. The reason is that the pre-defined relationship in multi-modal fusion modules makes it easy to overfit to specific datasets (*e.g.*, with specific scenarios, query lengths, and relationships). Our TransVG theoretically avoids this problem by establishing intra-modality and inter-modality correspondence with the flexible and adaptive attention mechanism.

4.4. Ablation Study

In this section, we conduct ablative experiments to verify the effectiveness of each component in our proposed framework. We exploit ResNet-50 as the backbone network of the visual branch, and all of the compared models are trained for 90 epochs as described in the implementation details.

Design of the [REG] Token. We study the design of the [REG] token on RefCOCO dataset, and report the results in Table 3. There are several choices to construct the initial state of the [REG] token. We detail these designs and analysis them as follows:

— *Average pooled visual tokens.* We perform average pooling over the visual tokens and exploit the average-pooled embedding as the initial state of [REG] token.

— *Max pooled visual tokens.* We take the max-pooled visual token embedding as the initial [REG] token.

— *Average pooled linguistic tokens.* Similar to the first choice, but using the linguistic tokens.

— *Average pooled linguistic tokens.* Similar to the second choice, but using the linguistic tokens.

— *Sharing with [CLS] token.* We use the [CLS] token of linguistic embedding to pl the [REG] token. Concretely, the [CLS] token out of the V-L module is fed into the prediction head.

— *Learnable embedding*.* This is our default setting by randomly initializing the [REG] token embedding at the beginning of the training stage. And the parameters of this embedding are optimized with the whole model.

Table 3. Ablative experiments on RefCOCO to study the [REG] token design in our framework. The initial state of the [REG] token is either obtained from visual/linguistic tokens out of the corresponding branch or by exploiting a learnable embedding.

Initial State of [REG] Token	RefCOCO@val
Average pooled visual tokens	79.12
Max pooled visual tokens	78.37
Average pooled linguistic tokens	78.51
Max pooled linguistic tokens	78.74
Sharing with [CLS] token	77.90
Learnable embedding*	**80.32**

Our proposed design to exploit a learnable embedding achieves 80.32% top-1 accuracy on the validation set of RefCOCO, which is the best performance among all the designs. Typically, the initial [REG] token of other designs is either generated from visual or linguistic tokens, which involves biases to the specific prior context of the corresponding modality. In contrast, the learnable embedding tends to be more equitable and flexible when performing relation reasoning in the visual-linguistic transformer.

Transformers in Visual and Linguistic Branches. We study the role of the transformers in the visual branch and the linguistic branch (*i.e.*, visual transformer and linguistic transformer). Table 4 summarizes the results of several

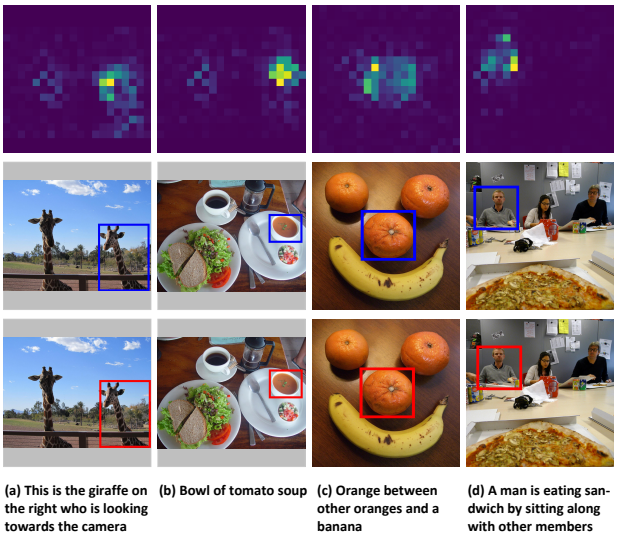

Figure 3. Qualitative results of our TransVG on the RefCOCOg test set (better viewed in color). We show the [REG] token's attention to visual tokens in the top row. Blue and red boxes are the predicted regions and the ground truths, respectively.

models with or without the visual transformer and the linguistic transformer. The baseline model without both visual transformer and linguistic transformer reports an accuracy of 64.24%. When we only attach either the visual transformer or the linguistic transformer, an improvement of 68.48% and 66.78% are achieved, respectively. With the complete architecture, the performance is further boosted to 69.76% on the ReferIt test set. This result demonstrates the essential of transformers in the visual branch and linguistic branch to capture intra-modality global context before performing multi-modal fusion.

4.5. Qualitative Results

We showcase the qualitative results of four examples from the RefCOCOg [31] test set in Figure 3. We observe that our approach can successfully model queries with complicated relationships, *e.g.*, "orange between other oranges and a banana" in Figure 3 (c). The first row of Figure 3 visualizes the [REG] token's attention to the visual tokens in the visual-linguistic transformer. TransVG generates interpretable attentions on the referred object that corresponds to the overall object shape and location.

Motivated by the correspondence between visual attention and predicted regions, we visualize the [REG] token's attention score on the visual tokens in the visual-linguistic transformer's intermediate layers to better understand TransVG. Figure 4 shows the [REG] token's attention score on the visual tokens from the second, forth and sixth transformer encoder layers. In the early layer (layer 2), we observe that the [REG] token captures the global context by attending to multiple regions in the whole image. In the

Table 4. Ablative experiments of the visual transformer and linguistic transformer in our framework. The performance is evaluated on the test set of ReferItGame [23] in terms of top-1 accuracy (%). "Tr." represents transformer.

Visual Branch w/o Tr.	Visual Branch w/ Tr.	Linguistic Branch w/o Tr.	Linguistic Branch w/ Tr.	Accuracy (%)	Runtime (ms)
✓		✓		64.24	33.67
✓			✓	66.78$_{\uparrow 3.54}$	47.57
	✓	✓		68.48$_{\uparrow 4.24}$	40.14
	✓		✓	**69.76**$_{\uparrow 5.52}$	61.77

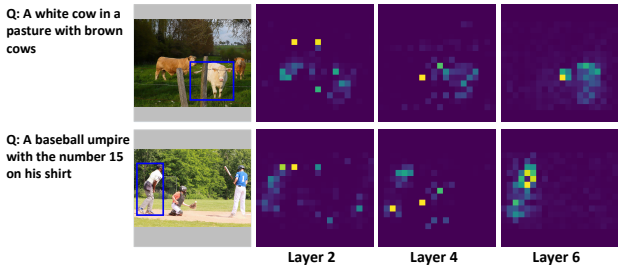

Figure 4. Visualization of the [REG] token's attention score on visual tokens from the second (layer 2), forth (layer 4) and sixth (layer 6) encoder layer of the visual-linguistic transformer.

middle layer (layer 4), the [REG] token tends to attend the discriminative regions which are closely related to the referred object (*e.g.*, the bus behind the man in the first example, which indicates the scene is on the road). In the final layer (layer 6), TransVG attends to the referred object and generates a more accurate attention prediction for the object's shape, which enables the model to regress the target's coordinates correctly.

5. Conclusion

In this paper, we present TransVG, a transformer-based framework for visual grounding. Instead of leveraging complex manually designed fusion modules, TransVG uses a simple stack of transformer encoders to perform the multi-modal fusion and reasoning for the visual grounding task. Extensive experiments indicate that TransVG's multi-modal transformer layers effectively perform the step-by-step fusion and reasoning, which enable TransVG to set a series of new state-of-the-art records on multiple datasets. Our TransVG serves as a new framework and exhibits huge potential for future investigation.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Contract 61836011, 61632019, and 62021001, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015. 1

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 4

[3] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. G3raphground: Graph-based language grounding. In *ICCV*, pages 4281–4290, 2019. 2

[4] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550, 2010. 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3, 4, 5

[6] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv:2012.00364*, 2020. 3

[7] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, pages 824–832, 2017. 2

[8] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pages 1691–1703, 2020. 3

[9] Xinpeng Chen, Lin Ma, Jingyuan Chen, Zequn Jie, Wei Liu, and Jiebo Luo. Real-time referring expression comprehension by single-stage grounding network. *arXiv preprint arXiv:1812.03426*, 2018. 1, 2, 6, 7

[10] 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. 3

[11] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *ICLR*, 2018. 3, 4

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. 3, 4, 5

[13] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (seqground). In *CVPR*, pages 4175–4184, 2019. 2

[14] 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 *ICLR*, 2021. 3

[15] Hugo Jair Escalante, Carlos A Hernández, Jesus A Gonzalez, Aurelio López-López, Manuel Montes, Eduardo F Morales, L Enrique Sucar, Luis Villaseñor, and Michael Grubinger. The segmented and annotated iapr tc-12 benchmark. *CVIU*, 114:419–428, 2010. 5

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[17] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37:583–596, 2014. 3

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997. 3

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

[20] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, pages 1115–1124, 2017. 2, 6, 7

[21] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, pages 4555–4564, 2016. 1

[22] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *ECCV*, pages 17–33, 2020. 3

[23] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 1, 2, 5, 6, 8

[24] Rama Kovvuri and Ram Nevatia. Pirc net: Using proposal indexing, relationships and context for phrase grounding. In *ACCV*, pages 451–467, 2018. 6

[25] Jianan Li, Yunchao Wei, Xiaodan Liang, Fang Zhao, Jianshu Li, Tingfa Xu, and Jiashi Feng. Deep attribute-preserving metric learning for natural language object retrieval. In *MM*, pages 181–189. ACM, 2017. 1

[26] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137, 2020. 3

[27] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *CVPR*, pages 10880–10889, 2020. 1, 2, 3, 6, 7

[28] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, pages 4673–4682, 2019. 2, 7

[29] 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

[30] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 3

[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. 1, 2, 5, 6, 7, 8

[32] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *InterSpeech*, 2010. 3

[33] Niko Moritz, Takaaki Hori, and Jonathan Le. Streaming automatic speech recognition with the transformer model. In *ICASSP*, pages 6074–6078, 2020. 3

[34] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, pages 792–807, 2016. 1, 2, 5

[35] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *CVPR*, pages 10971–10980, 2020. 1

[36] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, pages 4055–4064, 2018. 4

[37] Bryan A. Plummer, Paige Kordas, M. Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *ECCV*, pages 249–264, 2018. 2, 5, 6

[38] 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. *IJCV*, 123(1):74, 2017. 1, 2, 5, 6

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*, 2019. 3, 4

[40] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018. 3

[41] 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 *CVPR*, pages 658–666, 2019. 5

[42] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *ICCV*, pages 4694–4703, 2019. 2, 6

[43] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. 2020. 3

[44] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv:1503.00075*, 2015. 3

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4

[46] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *TPAMI*, 41:394–407, 2018. 1, 2, 5, 6

[47] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013, 2016. 2

[48] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *CVPR*, pages 1960–1968, 2019. 2, 7

[49] Yingjie Wang, Qiuyu Mao, Hanqi Zhu, Yu Zhang, Jianmin Ji, and Yanyong Zhang. Multi-modal 3d object detection in autonomous driving: a survey. *arXiv:2106.12735*, 2021. 1

[50] Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP*, pages 6874–6878, 2020. 3

[51] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *CVPR*, pages 5791–5800, 2020. 3

[52] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *ICCV*, pages 4644–4653, 2019. 2, 7

[53] Sibei Yang, Guanbin Li, and Yizhou Yu. Graph-structured referring expression reasoning in the wild. In *CVPR*, pages 9952–9961, 2020. 2

[54] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. In *CVPR*, pages 9847–9857, 2021. 1

[55] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *ECCV*, 2020. 2, 3, 5, 6, 7

[56] 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, 3, 6, 7

[57] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. Tap: Text-aware pre-training for text-vqa and text-caption. *arXiv:2012.04638*, 2020. 3

[58] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *ACL*, 2:67–78, 2014. 5

[59] 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

[60] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 1, 2, 5, 6, 7

[61] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. In *IJCAI*, 2018. 6

[62] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, pages 528–543, 2020. 3

[63] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, pages 4158–4166, 2018. 2, 6, 7

[64] Yuting Zhang, Luyao Yuan, Yijie Guo, Zhiyuan He, I-An Huang, and Honglak Lee. Discriminative bimodal networks for visual localization and detection with natural language queries. In *CVPR*, pages 557–566, 2017. 2

[65] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, pages 13278–13288, 2020. 1

[66] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. In *ICLR*, 2020. 3

[67] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 3

[68] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton van den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*, pages 4252–4261, 2018. 2, 7