TransVG++: End-to-End Visual Grounding With Language Conditioned Vision Transformer

Jiajun Deng, Zhengyuan Yang, Daqing Liu, Tianlang Chen, Wengang Zhou, Senior Member, IEEE, Yanyong Zhang, Fellow, IEEE, Houqiang Li, Fellow, IEEE, and Wanli Ouyang, Senior Member, IEEE

Abstract—In this work, we explore neat yet effective Transformer-based frameworks for visual grounding. The previous methods generally address the core problem of visual grounding, i.e., multi-modal fusion and reasoning, with manually-designed mechanisms. Such heuristic designs are not only complicated but also make models easily overfit specific data distributions. To avoid this, we first propose TransVG, which establishes multi-modal correspondences by Transformers and localizes referred regions by directly regressing box coordinates. We empirically show that complicated fusion modules can be replaced by a simple stack of Transformer encoder layers with higher performance. However, the core fusion Transformer in TransVG is stand-alone against uni-modal encoders, and thus should be trained from scratch on limited visual grounding data, which makes it hard to be optimized and leads to sub-optimal performance. To this end, we further introduce TransVG++ to make two-fold improvements. For one thing, we upgrade our framework to a purely Transformer-based one by leveraging Vision Transformer (ViT) for vision feature encoding. For another, we devise Language Conditioned Vision Transformer that removes external fusion modules and reuses the uni-modal ViT for vision-language fusion at the intermediate layers. We conduct extensive experiments on five prevalent datasets, and report a series of state-of-the-art records.

Index Terms—Deep learning, transformer network, vision and language, visual grounding.

I. INTRODUCTION

VISUAL grounding, which aims to localize a region referred to by a language expression in an image, is a core technology to bridge the natural language expression delivered by human beings and visual contents in the physical world. The evolution of this technique is of great potential to promote vision-language understanding, and to provide an intelligent interface for human-machine interaction. Existing methods addressing this task generally follow two-stage or one-stage pipelines shown in Fig. 1. Specifically, two-stage approaches [1], [2] first generate a set of region proposals, and then take visual grounding as a natural language object retrieval problem [3], [4] to find the best matching region given language expressions. Differently, one-stage approaches [5], [6], [7] perform vision-language fusion at the output of the vision backbone network and the language model. Then, they make dense predictions with a sliding window over pre-defined anchor boxes, and keep the box with the maximum score as the final prediction.

Multi-modal fusion and reasoning is the primary problem in visual grounding. The early two-stage and one-stage methods address multi-modal fusion in a simple way. Concretely, the pioneer two-stage method, Embedding Net [9], measures the similarity between region embedding and expression embedding by cosine distance. The early one-stage approach, FAOA [7], encodes the language feature vector to each spatial position of vision feature maps by directly concatenating them. In general, these attempts are efficient, but lead to sub-optimal results, especially when it comes to complicated language expressions [10], [11]. The following studies have proposed diverse architectures to ameliorate their performance. Among two-stage methods, modular attention network [12], various scene graphs [13], [14], [15], and multi-modal tree [10], [16] are designed to improve multi-modal relationships modeling. The one-stage method [11] also explores query decomposition and proposes a multi-round fusion mechanism.

Despite their effectiveness, these sophisticated fusion or matching modules are built on pre-assumed dependencies of language expressions and visual scenes, making the models easily overfit specific scenarios, such as certain query lengths or object relationships. Meanwhile, these mechanisms limit the plenitudinous interaction between vision and language contexts, which also hurts the performance of visual grounding algorithms. Besides, even though the target is to localize the referred region, most of the previous methods achieve this target in an indirect way. They generally define surrogate problems of language-guided candidate matching, selection, and refinement, following the common practice of image-text retrieval and object detection. Therefore, extra efforts have to be devoted to...
obtaining candidates, including region proposals [2], [17], [18] and pre-defined anchor boxes [7], [11]. Since these methods’ predictions are made out of candidates, their performance is easily influenced by the step to generate such candidates and by the heuristics to assign targets to candidates.

In this work, we explore an alternative approach in fusion module design, and re-formulate the prediction processing back to a simple regression problem. We first present TransVG, the preliminary version framework to address the problem of visual grounding with Transformers. We empirically show that the structured fusion modules can be replaced by a simple stack of Transformer encoder layers. The insight behind our design is that the basic component of Transformer, i.e., attention module, is ready to establish intra- and inter-modality correspondence for vision and language inputs, despite we neither suppose any language expression structure nor visual layout. The pipeline of TransVG is illustrated in Fig. 1(c). We first feed the image and language expression into two sibling branches. The Transformer built on the top of a convolutional neural network and language Transformer built on word embeddings are applied in these two branches to model the global cues in vision and language domains, respectively. Then, the abstracted vision tokens and language tokens are fused together, and a vision-language Transformer is leveraged to perform cross-modal relation reasoning. Particularly, at the prediction step, TransVG directly outputs 4-dim coordinates of a bounding box to localize the referred region, dim coordinates of a bounding box to localize the referred region.

Although Transformers have been leveraged to establish multi-modal dependencies, TransVG shares the same meta-architecture with previous two-stage and one-stage methods, which include two independent uni-modal feature encoders, a stand-alone multi-modal fusion or matching module, and a prediction module. Within this meta-architecture, the parameters of vision encoder and language encoder can be initialized with well pre-trained models, while the primary fusion module, i.e., vision-language Transformer, is left to be trained from scratch with limited visual grounding data. As a result, the stand-alone fusion Transformer is hard to be optimized and leads to sub-optimal performance.

To this end, we further introduce an advanced version framework, namely TransVG++, to remove the stand-alone fusion Transformer, and instead performs vision-language fusion by re-using the vision feature encoder. As shown in Fig. 1(d), we devise Language Conditioned Vision Transformer (LViT) to play the role of both vision feature encoding and vision-language reasoning. Technically, LViT is obtained with minimal adaptation and extra parameters, by integrating information of language expressions into intermediate layers of the plain Vision Transformer (ViT) [8]. Two novel alternative strategies, i.e., namely language prompter and language adapter, are proposed to convert uni-modal visual encoder layers of ViT into linguistic conditioned visual encoder layers of LViT. Built on a fully Transformer architecture and waiving the external fusion Transformer, TransVG++ achieves consistently better performance with even smaller model size and less computation cost when compared with the preliminary version. Furthermore, piggybacking on the successful ViT series allows TransVG++ to be scaling up with minimal efforts. We empirically show that larger LViT models consistently lead to better TransVG++ variants, and expect the trend to hold for stronger Transformer based vision backbones.

We benchmark the proposed frameworks on five prevalent datasets, including RefCOCO [17], RefCOCO+ [17], RefCOCOg [18], ReferItGame [19], Flickr30 K Entities [20]. Our preliminary framework respectively achieves 83.38%, 59.24%, 68.71%, 70.73% and 79.10% accuracy on the test set of these five datasets, consistently outperforming the previous two-stage and one-stage approaches. Remarkably, TransVG++ further achieves 4.99%, 8.04%, 8.32%, 3.97% and 2.39% absolute

Fig. 1. A comparison of (a) two-stage pipeline, (b) one-stage pipeline, (c) our preliminary TransVG framework, and (d) our advanced TransVG++ framework. TransVG performs intra- and inter-modality relation reasoning with a stack of Transformer layers in a homogeneous way, and grounds the referred region by directly regressing the box coordinates. TransVG++ takes further steps that upgrade the hybrid architecture composed of convolutional neural networks and Transformer networks to purely Transformer-based one and remove the stand-alone fusion Transformer by introducing Language Conditioned Vision Transformer to enable fusion at the intermediate layers of a vision feature encoder. Particularly, we use Transformer (vision) to represent a stack of Transformer encoder layers built on the top of a convolutional neural network (CNN), distinguished from Vision Transformer [8], a specific model in the literature.
improvements over the preliminary one, setting a series of state-of-the-art records.

In summary, we make three-fold contributions:

- **We propose the first purely Transformer-based framework to address the problem of visual grounding and reformulate the prediction process to directly regress the box coordinates of referred regions.**
- **We present an elegant view of how to capture intra- and inter-modality context homogeneously with Transformers, and further investigate how to remove stand-alone fusion modules by integrating language information into intermediate layers of a vision encoder.**
- **We conduct extensive experiments to validate the merits of our method, and show significantly improved results on several prevalent benchmarks.**

The preliminary version of this work is published in [21]. We have made significant improvements and extensions to our preliminary work. The major technical improvements can be concluded in two aspects. For the most important point, we remove the stand-alone fusion module and enable the vision encoder to be re-used for multi-modal fusion at the intermediate layers. Besides, we upgrade the hybrid architecture composed of convolutional neural networks and Transformer networks to a purely Transformer-based one. To the best of our knowledge, this is the first fully Transformer-based framework in this field, without introducing any inductive bias. As demonstrated by experimental results, the advanced framework outperforms the preliminary one by a large margin and meanwhile achieves better efficiency on model size and computation cost.

To facilitate the further investigation, we build a benchmark of Transformer-based visual grounding frameworks at: //github.com/djiajunustc/TransVG, and will make our codes and models available.

II. RELATED WORK

A. Visual Grounding

Visual grounding aims to ground a natural language description onto the referred region in an image. The previous visual grounding methods can be categorized into two-stage methods [2], [10], [12], [13], [14], [16], [22], [23], [24] and one-stage methods [5], [6], [7], [11], [25]. Our preliminary TransVG [21] gets rid of generated region proposals and heuristically designed anchors, directly localizing the referred region, which is the first Transformer based method.

In the following, we first deliver a literature review on the two-stage and one-stage methods. After that, we discuss some following Transformer based approaches and give a glance at fusion in vision encoder, which is one of the core insights of our advanced approach.

**Two-Stage Methods:** Inspired by the success of region based object detectors [26], [27], [28], two-stage visual grounding methods fuse the language feature to the visual content at region level, thus extracting a set of region features at first. At the first stage, region proposals can be either generated with external modules based on super-pixels grouping [2], [29], [30], or predicted with pre-trained object detectors [10], [12], [23], [31]. The main efforts of approaches in this direction are devoted to the second stage, addressing visual grounding as text-region matching. The pioneer studies [1], [9], [18] obtain good results by optimizing the feature embedding networks with maximum-margin ranking loss to maximize the similarity between the positive object-query pairs. The following work DBNet [32] and Similarity Net [2] show that the similarity and dissimilarity of text-region pairs can be predicted by directly performing binary classification. The comprehensive region proposals provided by the vision encoder make it easy for two-stage methods to reason the relationships among objects.

RCCF [17] first proposes to incorporate visual comparison-based context into referring expression models, which demonstrates the significance of involving out-of-object information for visual grounding. MattNet [12] introduces the modular design and improves the grounding accuracy by better modeling the subject, location, and relation-related language description. The following studies further improve the two-stage methods by better modeling the object relationships [10], [13], [14], [15], [33], enforcing correspondence learning [34], or making use of phrase co-occurrences [35], [36], [37]. A more recent work M-DGT [38] first constructs a multi-modal graph based on the language embedding and region feature of initial boxes, and then exploits a node Transformer and a graph Transformer to gradually transform the initial boxes to approach the target location.

**One-Stage Methods:** One-stage approaches [5], [6], [7], [11], [25] get rid of the region proposals and fuse the linguistic context with visual features densely at each spatial position of feature maps. The fused image-text feature maps are then leveraged to perform bounding box prediction in a sliding-window manner. The early work FAOA [7] encodes the language expression into a language vector, and fuses the language vector into the YOLOv3 detector [39] to ground the referred region. RCCF [6] formulates the visual grounding problem as a correlation filtering process [40], [41], and picks the peak value of the correlation heatmap as the center of target objects. The recent work ReSC [11] devises a recursive sub-query construction module to address the limitations of FAOA [7] on grounding complex queries by multi-round fusion. LSPN [42] further improves the reasoning ability of the one-stage grounding methods.

**Transformer-Based Methods:** Our preliminary method, i.e., TransVG, for the first time, introduces the Transformer-based pipeline. It shows that intra- and inter-modality correspondence can be easily established by a stack of Transformer encoder layers, and proposes an alternative prediction paradigm to directly regress the target coordinates. VLTVG [43] introduces a verification mechanism to first establish text-conditioned discriminative visual feature, and then exploits multi-stage cross-modal decoder to iteratively infer the referred location. QRNet [44] introduces query-aware dynamic attention to extract query-refined visual features with a hierarchical structure, and then combines the multi-scale visual features by query-aware fusion before feeding them to the multi-modal fusion Transformer.

**Fusion in the Vision Encoder:** Abstracting away the distinctions in the fusion module, one- and two-stage visual grounding approaches share the same meta-framework, i.e., single-modal
feature encoders followed by an external multi-modal feature fusion or matching module. In this study, we alternatively explore how to remove the stand-alone fusion module, and re-use the vision feature encoder (i.e., a plain ViT [8] in our TransVG++) to perform the vision-language fusion. We vision two major advantages of the proposed novel paradigm. First, removing the explicit fusion layers makes the framework efficient and light-weighted, thus also performing better with the same amount of computations. Second, the unified vision encoder design facilitates the model to benefit from the advances in computer vision studies, such as stronger image classification and object detection models pretrained on large-scale datasets.

B. Transformer

Transformer is first introduced in [45] to tackle the problem of neural machine translation (NMT). Compared with recurrent units in RNNs [46], [47] and LSTMs [48], the core component of Transformer, i.e., attention modules, show remarkable superiority in long-term sequence modeling. Transformer attracts increasing interests in the computer vision community, and has made unnegligible impacts on vision and vision-language tasks.

Transformers in Vision Tasks: Inspired by the great success in NMT, a series of Transformers [8], [49], [50], [51], [52], [53], [54], [55] applied to vision tasks have been proposed. The intrusive work DETR [49] formulates object detection as a set prediction problem. It introduces a set of learnable object queries, reasons global context and object relations with attention mechanism, and outputs the final set of instance predictions. ViT [8] and DeiT [53] show that a pure Transformer can achieve excellent accuracy for image classification. The following work Swin Transformer [54] re-introduces inductive bias into vision Transformers by devising shifted-window attention. The recent works UViT [56] and ViTDet [57] demonstrate the non-hierarchical ViT can be directly applied as the backbone network of object detectors. Our TransVG++ also explores the usage of plain ViT in downstream tasks. Distinguishingly, we investigate how to adapt the uni-modal ViT for the multimodal visual grounding task, while best preserving its power and architecture for vision feature extraction.

Transformer in Vision-Language Tasks: Motivated by the powerful pre-trained model of BERT [58], some researchers start to investigate visual-linguistic pre-training (VLP) [59], [60], [61], [62], [63] to jointly represent images and texts. In general, the early works take region 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. Some recent works [64], [65], [66], [67] further improve the model to make use of raw image patches. Besides following the pretext tasks of BERT, MDETR [68] explores using object detection as the pretext task to help improve the performance of some downstream vision-language tasks.

Despite with similar base units, the goal of VLP is to learn a generalizable vision-language representation with large-scale data. In contrast, we focus on exploring novel Transformer-based visual grounding frameworks. Besides, ViT has also shown its power in other vision-language tasks, while mainly as the backbone of vision encoder [66], [67]. They still build stand-alone blocks for multi-modal fusion. We alternatively explore having the plain ViT do more work and removing the stand-alone fusion layers.

III. OUR APPROACH

In this work, we present two versions of novel architecture for visual grounding by leveraging the glamorous Transformer neural networks. An overview of our proposed frameworks are illustrated in Fig. 2. The preliminary version framework, namely TransVG, performs vision-language feature fusion with a simply stack of Transformer encoder layers, getting rid of manually designed mechanisms. The advanced version framework, i.e., TransVG++, is featured by removing the stand-alone fusion module and introducing Language Conditioned Vision Transformer (LVIT), which plays the roles of both vision feature encoding and multi-modal fusion. Particular, LVIT is converted from the uni-modal ViT [8] model with minimal adaptation and extra costs. Moreover, different from previous methods that rely on candidate boxes selection or refinement, both of our proposed frameworks introduce a learnable [REG] token, and localize the referred region by directly regressing the box coordinates with the output state of [REG] token.

In the Section III-A, we first review the background knowledge of Transformer networks. Then, we elaborate our model design of the preliminary version (Section III-B) and advanced version (Section III-C) frameworks. Finally, in Section III-D, we introduce the training objectives of our frameworks.

A. Background: Transformer

Before detailing our proposed frameworks, we first deliver a background overview of attention modules and encoder layers of Transformer neural networks [45].

Attention Module: The core component of Transformer encoder layer is the attention module. Here, we take single-head attention as an example for explanation. The input of single-head attention module is a sequence of feature tokens. Given the query tokens $x^q$ and support tokens $x^s$, three independent fully connected (FC) layers are applied on them to generate the query embedding $f^Q$, key embedding $f^K$ and value embedding $f^V$ as follows:

$$ f^Q = FC(x^q), \quad f^K = FC(x^s), \quad f^V = FC(x^s). $$

(1)

When support token set is the same as query token set, the attention module is named as a self-attention module. Otherwise, it is named as a cross-attention module. After obtaining query, key and value embedding, the output of a single-head attention layer is computed as

$$ \text{Attn}(f^Q, f^K, f^V) = FC(\text{Softmax}(\frac{f^Q f^K}{\sqrt{d^K}}) \cdot f^V), $$

(2)

where $d^K$ is the channel dimension of $f^K$, $\text{Softmax}(\cdot)$ is the softmax function applied across the key/value embedding.
**Fig. 2.** An overview of our proposed TransVG and TransVG++ frameworks. TransVG includes four components: a vision branch, a language branch, a vision-language fusion module and a prediction head. Vision tokens, language tokens and a learnable \([\text{REG}]\) token are put together as the inputs of the vision-language Transformer for multi-modal reasoning. The output state of \([\text{REG}]\) token is fed into the prediction head for box coordinates regression. To extend the preliminary framework, TransVG++ removes the stand-alone fusion module, and introduce Language Conditioned Vision Transformer, which enables vision-language fusion at the intermediate layers of a vision feature encoder.

**Fig. 3.** An illustration of two varieties of Transformer encoder layers, including (a) post-normalization encoder layer and (b) pre-normalization encoder layer.

**Encoders Layer:** In Fig. 3, we illustrate two kinds of Transformer encoder layers, distinguished by the position to perform layer normalization (LN) [69]. Concretely, a Transformer encoder layer has two main sub-layers, i.e., a multi-head self-attention (MHSA) layer and a feed-forward network (FFN). Multi-head attention is a variant of single-head attention by splitting embedding channels into multiple groups. FFN is an multi-layer perceptron (MLP) composed of FC layers and non-linear activation layers.

In Transformer encoder layers, each sub-layer is put into a residual structure. Let us denote the input as \(x_n\), a post-normalization Transformer encoder layer layer computes

\[
x'_n = \text{LN}(x_n + \mathcal{F}_{\text{MHSA}}(x_n)),
\]

while the computation procedure in a pre-normalization Transformer encoder layer is

\[
x'_n = x_n + \mathcal{F}_{\text{MHSA}}(\text{LN}(x_n)),
\]

\[
x_{n+1} = x'_n + \mathcal{F}_{\text{FFN}}(\text{LN}(x'_n)).
\]

**B. Preliminary Version: TransVG Framework**

In this subsection, we present TransVG, the preliminary version framework based on a stack of Transformer encoder layers with direct box coordinates prediction. As shown in Fig. 2(a), given an image and a language expression as inputs, we first separate them into two sibling branches, i.e., a vision branch and a language branch, to generate corresponding feature embedding. Then, we construct the inputs of vision-language fusion module by putting the vision and language feature embedding together, and append a learnable token (i.e., \([\text{REG}]\) token). The vision-language Transformer homogeneously embeds the input tokens from different modalities into a common semantic space by modeling intra- and inter-modality context with the self-attention layers. Finally, the output state of \([\text{REG}]\) token is leveraged to directly predict 4-dim coordinates of a referred region in the prediction head.

**Vision Branch:** The vision branch includes a convolutional network and a following Transformer encoder. We exploit the commonly used ResNet [70] as the convolutional backbone...
network, and build the Transformer encoder with a stack of 6
decoder transformer encoder layers. In each encoder layer, there are 8 heads in the MHSA module, and 2 FC layers
followed by ReLU activation layers in the FFN. The output channel dimensions of these 2 FC layers in the FFN are 2048
and 256, respectively.

Given an image \( z_0 \in \mathbb{R}^{256} \times H_0 \times W_0 \) as the input, we exploit the
backbone network to generate a 2D feature map \( z \in \mathbb{R}^{C_v} \times H \times W \). Typically, the channel dimension \( C \) is 2048, and the width
and height of \( z \) are \( \frac{H_0}{H}, \frac{W_0}{W} \) of the original image size (\( H = \frac{H_0}{8}, W = \frac{W_0}{8} \)). Then, we leverage a \( 1 \times 1 \) convolutional layer to
reduce the channel dimension of \( z \) to \( C_v = 256 \) and obtain \( z' \in \mathbb{R}^{C_v} \times H \times W_0 \). Since the input of a Transformer encoder layer is
supposed to be a sequence of 1D vectors, we further flatten \( z' \) into \( z_v \in \mathbb{R}^{C_v \times N_v} \), where \( N_v = H \times W \) is the number of
input vision tokens. To make the visual Transformer sensitive to the original 2D positions of input tokens, we follow [49],
[71] to utilize sine position embedding 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 context reasoning
in parallel, and outputs visual embedding \( f_v \), which shares the same shape as input \( z_v \).

**Language Branch:** The language branch is a sibling to the
vision branch, and it includes a language embedding layer and a
language Transformer. To make the best of pre-trained
language models, the architecture of this branch follows BERT [58].
Typically, there are 12 pre-normalization Transformer encoder layers. The output channel dimension of language Transformer is
\( C_l = 768 \).

Given a language expression, we first represent each word ID
as a one-hot vector. Then, in the language embedding layer, we
convert each one-hot vector into a language token by looking up
the token table. We follow the common practice in NMT [45],
[58], [72], [73] to append a [CLS] token and a [SEP] token at the beginning and end positions of language tokens. Different
from the sine position embedding leveraged in the vision branch, we make use of learnable position embedding in the language branch, and directly add them to language tokens. After that, we
take these language tokens as inputs of linguistic Transformer to obtain the output language feature embedding \( f_l \in \mathbb{R}^{C_l \times N_l} \),
where \( N_l \) is the number of language tokens.

**Vision-Language Fusion Module:** As the core component in
our model to fuse multi-modal information, the architecture of the
vision-language fusion module (abbreviated as V-L module) is extremely simple and elegant. Specifically, the V-L module includes two linear projection layers (one for each modality) and a
vision-language Transformer (also with a stack of 6 post-
normalization Transformer encoder layers).

Given vision tokens \( f_v \in \mathbb{R}^{256 \times N_v} \) out of the vision branch and language tokens \( f_l \in \mathbb{R}^{768 \times N_l} \) out of the language branch, we apply a linear projection layer on each of them to project
them into embedding with the same channel dimension. We
denote the projected visual embedding and linguistic embedding
as \( g_v \in \mathbb{R}^{C_p \times N_v} \) and \( g_l \in \mathbb{R}^{C_p \times N_l} \), where the projected feature dimension \( C_p = 256 \). Then, we pre-append a learnable embedding (namely a [REG] token) to \( g_v \) and \( g_l \), and formulate

the joint input tokens of the visual-linguistic Transformer as

\[
x_0 = [g_r, g_1, g_2, \ldots, g_N, g_1^P, g_1^P, g_1^P, \ldots, g_1^P],
\]

where \( g_r \in \mathbb{R}^{C_p \times 1} \) 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 \( x_0 \in \mathbb{R}^{C_p \times (1+N_v+N_l)} \) in the joint embedding space as described above, we apply the vision-language Transformer to embed \( x_0 \) 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 embedding 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
techniques, regardless of their modality. For example, a vision token can attend to a vision token, and it can also freely attend to
a language token. Typically, the output state of the [REG] token develops a consolidated representation enriched by both
vision and language 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 apply a regression block to the [REG] token. The regression block is implemented by an MLP with two ReLU activated hidden layers and a linear
output layer. The output of the prediction head is the 4-dim box
coordinates.

**C. Advanced Version: TransVG++ Framework**

To further drive the evolution of Transformer-based visual
grounding, we introduce TransVG++, the advanced version
framework with fully Transformer-based architecture. We depict
the overview of TransVG++ framework in Fig. 2(b). As illustrated,
the main difference between TransVG++ and TransVG is that we remove the stand-alone vision-language fusion module,
and capitalize on a novel vision feature encoder, namely Lan-
guage Contionted Vision Transformer (LViT). On the one hand,
LViT upgrades the hybrid vision feature encoder composed of
ResNet and 6 following Transformer encoder layers, and on the
other hand LViT enable multi-modal fusion at its intermediate
layers. Besides, the vision input of our TransVG++ is no longer
the whole images, but equally divided raw image patches.

Particularly, LViT is modified from a uni-modal ViT [8] model
with few efforts and negligible extra computation cost. Specif-
cally, a standard ViT model is composed of 12 pre-normalization
Transformer encoder layers. We denote them as vision encoder
layers, since they are only leveraged to extract vision feature
embedding, and equally divide them into 4 groups. Our core
modification is that we convert the last vision encoder layer of
each group into a language conditioned vision encoder layer,
which integrates the language feature of referring expressions
into vision tokens. Motivated by the success of prompt tuning
[74] and network adapter [75], [76] in transfer pre-trained
models to downstream tasks, we propose two simple yet effective strategies for language feature integration: (a) language prompter, and (b) language adapter.

**Language Prompter:** We illustrates the architecture of a language conditioned vision encoder layer with language prompter in Fig. 4(a). In this strategy, we first feed the language tokens into a feed forward network (FFN) to generate language prompt tokens. After that, language prompt tokens are concatenated with the [REG] token and vision tokens to construct the input of a multi-head self-attention (MHSA) module. The tokens out of MHSA module is split into two groups, i.e. one group for vision tokens and [REG] token, and another group for prompt tokens. The prompt tokens are dropped, while the other group is fed into the following FFN to obtain the input of next layers. Here, the parameters of the FFN to generate prompt tokens are randomly initialized, and those of other components make use of parameters from the corresponding original vision encoder layer of a pre-trained ViT backbone network.

**Language Adapter:** The architecture of a language conditioned vision encoder layer with language adapter is illustrated in Fig. 4(b). In this strategy, we inject the language expression information by adding an extra multi-head cross-attention (MHCA) module between the MHSA module and FFN of the original vision encoder layer. The query embedding of this MHCA module is the output of MHSA module, and both the key and value embedding are from the language tokens. By this way, the outputs of MHCA module, namely language adapt tokens, are the aggregation of language features with adaptive weights against each vision token. The language adapt tokens are then elementwisely added to the vision tokens for vision-language feature fusion. Similar to the language prompter, the parameters of the MHCA module are randomly initialized, and those of other parts are initialized with corresponding parameters from a pre-trained ViT backbone network.

By capitalizing on either a language prompter or a language adapter, the vision encoder layer can be converted to language conditioned vision encoder layers, and thus enables it to be re-used for absorbing text expression information from language tokens. The proposed LViT maintains the original architecture of a ViT model, so that we can naturally take advantage of pre-trained ViT model to ease the optimization for the whole framework. Particularly, both of these two strategies introduce negligible model parameters and computation cost compared with the whole model, while achieving inspiring performance improvements against our preliminary version with stand-alone fusion Transformers. In our experiments, we conduct extensive ablation studies and analysis on these two strategies, and empirically show that using language adapters can achieve a slightly better accuracy.

**D. 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), we directly predict 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 [77] (GIoU loss), which is not affected by the scales. Let us denote the prediction as $\hat{b} = (x, y, w, h)$, and the normalized ground-truth box as $\hat{b} = (\hat{x}, \hat{y}, \hat{w}, \hat{h})$. The training objective of our method is

$$\mathcal{L} = \mathcal{L}_{\text{smooth-l1}}(b, \hat{b}) + \mathcal{L}_{\text{giou}}(b, \hat{b}),$$

where $\mathcal{L}_{\text{smooth-l1}}(\cdot)$ and $\mathcal{L}_{\text{giou}}(\cdot)$ are the smooth L1 loss and GIoU loss, respectively.

**IV. EXPERIMENTS**

In this section, we present extensive experiments to validate the merits of our proposed TransVG/TransVG++ framework. In Sections IV-A and IV-B, we first give a brief introduction to the datasets and experimental setup. In Section IV-D, we conduct extensive ablative experiments to investigate the effectiveness of each component in our preliminary and advanced frameworks. Then, in Section IV-C, we present the main results of our models,
and compare them with other state-of-the-art methods. After that, in Section IV-E, we discuss the model size and computation cost of TransVG++, and point out future directions to make it more applicable on resource-limited devices. Finally, in Section IV-G, we show some qualitative results to help validation and analysis.

A. Datasets and Evaluation Metric

We conduct experiments on five prevalent datasets and follow the standard protocol to evaluate our framework in terms of accuracy. Specifically, only when the Jaccard overlap between the predicted region and the ground-truth region is above 0.5, the prediction is regarded as a correct one. These five datasets are detailed as follows:

RefCOCO [17] includes 19,994 images with 50,000 referred objects. In each image, there are multiple instances belonging to the same categories. Each referred instance has more than one referring expression, and there are 142,210 referring expressions in total. 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.

RefCOCO+ [17] contains 19,992 images with 49,856 referred objects and 141,564 referring expressions. Compared with RefCOCO, the words indicate the absolute position, like ‘left’ and ‘right’, are not included in the language expressions from this dataset. 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 [18] has 25,799 images with 49,856 referred objects and expressions. In this dataset, each image contains 2 to 4 instances for the referred categories, and each instance is with an area of more than 5% of the whole image. There are two commonly used split protocols for this dataset. One is RefCOCOg-GoogLe (RefCOCOg-g) [18], and the other is RefCOCOg-umd [1]. We report our performance on both RefCOCOg-g (val-g) and RefCOCOg-umd (val-u and test-u) to make a comprehensive comparison.

ReferItGame [19] includes 20,000 images collected from the SAIAPR-12 dataset [78], and each image has one or a few regions with corresponding referring expressions. This dataset is divided into three subsets, i.e., a train set with 54,127 referring expressions, a validation set with 5,726 referring expressions and a test set with 60,103 referring expressions. We use the validation set for ablation studies and compare our method with others on the test set.

Flickr30K Entities [20] is built on the original Flickr30K [79], through introducing short region phrase correspondence annotations. It contains 31,783 images with 427 K referred entities. We follow the previous works [2, 11, 20, 30] to separate the these images into 29,783 for training, 1000 for validation, and 1000 for testing.

B. Experimental Setup

Inputs: The input image size is set as 640 × 640. 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. The maximum length of language expressions is set as 40 for RefCOCOg, and set as 20 for other datasets. We cut off the language query if its length is longer than maximum length (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 consistent in each batch. The padded tokens are recorded with a mask, and will not influence the output.

Network Architecture: In our preliminary framework, we use ResNet-50 or ResNet-101 [70] as the convolutional backbone network. The output of ResNet is 32 times downsampling against the input image. Transformer encoder layers in the preliminary framework follow the post-normalization structure. The embedding dimension is set as 256, the head number of multi-head attention modules is set as 8, and the hidden dimension in FFN is set as 2048.

In our advanced framework, the patch embedding layer divides a 640 × 640 image into patches of equal size, each with 16 × 16 pixels. These patches are flattened into vectors and fed into a fully connected layer to generate the initial state of visual tokens. To encode position information into visual tokens, learnable position embedding is added to each visual token. Particularly, the resolution of position embedding in the standard ViT model for image classification is 14 × 14, as the input size is only 224 × 224. To make the position embedding match the input size of our method, we follow [56] to resize it to 40 × 40 with bicubic interpolation. We experiment with three representative models of ViT series, i.e., ViT-tiny, ViT-small and ViT-base, which is for vision feature encoding. The configuration of LViT varies according to the ViT variety. The feature dimension of our language adapter and language prompter changes according to the ViT model. Specifically, for ViT-tiny, the embedding dimension is 192, the head number of multi-head attention module is 3, the hidden dimension in FFN is 768. For ViT-small, the embedding dimension is 384, the head number of multi-head attention module is 6, the hidden dimension in FFN is 1536. For ViT-small, the embedding dimension is 768, the head number of multi-head attention module is 12, the hidden dimension in FFN is 3072. By default, 4 evenly distributed visual encoder layers (i.e., the 3 rd, 6th, 9th, and 12th encoder layers) are converted into language conditioned visual encoder layers via our proposed language prompters or language adapters. No matter what ViT model is leveraged, the hidden dimension in regression MLP is kept unchanged (i.e., 256 as that in preliminary version) to eliminate the influence of feature dimension in prediction head.

Training and Inference Details: Our model is end-to-end optimized with AdamW optimizer, with batch size set as 32. We apply several widely adopted data augmentation techniques [6], [7], [11], [21], [80] at the training stage. In TransVG, we set the initial learning rate of fusion module and prediction head to $10^{-4}$ and that of the vision branch and language branch to $10^{-5}$. The weight decay is $10^{-4}$. On RefCOCO, RefCOCOg and ReferItGame datasets, TransVG is trained for 90 epochs with a learning rate dropped by a factor of 10 after 60 epochs. On Flickr30K Entities, TransVG is trained for 60 epochs,
with a learning rate dropping after 40 epochs. On RefCOCO+, TransVG is trained for 180 epochs, with a learning rate dropping after 120 epochs.

For TransVG++, the learning rate of the language branch is set as $10^{-5}$ and that of the language prompter/adapter and prediction head is set as $10^{-4}$. Different from freezing the pre-trained backbone when adopting prompting tuning or adapters in NLP techniques, our ViT backbone is also end-to-end finetuned with the whole model with an initial learning rate set as $10^{-5}$. Thanks to removing the stand-alone fusion Transformer, the total training epochs of TransVG++ are reduced to 60 epochs. The learning rate drops by 10 times since the 45-th epoch. We follow the common practice [7, 11, 21, 80, 82] to initialize the parameters of the language branch with a pre-trained BERT\textsc{base} model [58], and to initialize the parameters of vision branch with models trained on MSCOCO [83] (the overlapping images between the training set of MSCOCO and validation/test sets of RefCOCO/RefCOCO+/RefCOCOg datasets are removed when performing pre-training). We use DETR model [49] for vision branch initialization in TransVG, and instead use Mask R-CNN [84] in TransVG++, since these two initialization options achieve comparable performance, while the training epochs of Mask R-CNN is remarkably less than DETR.

Since our proposed frameworks directly output the box coordinates, there is no post-processing during inference.

C. Comparison With State-of-the-Art Methods

1) RefCOCO/RefCOCO+/RefCOCOg. Comparison with Two-stage and One-stage Methods: Table I reports the performance of our proposed TransVG (preliminary version) and TransVG++ (advanced version), together with other competitive two-stage and one-stage methods on the widely adopted RefCOCO, RefCOCO+, and RefCOCOg datasets. To make a comprehensive comparison, we present the results of TransVG with ResNet-50 and ResNet-101 as the backbone network, and evaluate the performance of TransVG++ with ViT-Tiny, ViT-Small and ViT-Base as the backbone network. We add "*" to reffermer [80] to indicate this model is trained with extra annotations.

For a comprehensive comparison, we evaluate the performance of TransVG with ResNet-50 and ResNet-101 as the backbone network, and evaluate the performance of TransVG++ with ViT-Tiny, ViT-Small and ViT-Base as the backbone network. We add "*" to reffermer [80] to indicate this model is trained with extra annotations.

Among all the methods, our advanced TransVG++ series achieve the best grounding accuracy on the validation set and testA set of RefCOCO and RefCOCO+. Even with ViT-tiny backbone, which has only about 5.53 M parameters, TransVG++ outperforms most of the competitors by a remarkable margin. Such a result validates the effectiveness of our proposed strategy to preserve the power of ViT and fuse the language
information into vision features by injecting the language expression tokens into the intermediate vision Transformer encoder layers. When upgrading ViT-tiny to models with larger capacity, i.e., ViT-small and ViT-base, we observe consistent performance improvements, which shows the unified fully Transformer design facilitates TransVG++ to benefit from the future advances in vision Transformers. Besides, the training process of TransVG++ convergences much faster than TransVG. In TransVG, the core vision-language Transformer devised for multi-modal fusion is trained from scratch, making it hard to be optimized on the limited visual grounding data. On the contrary, TransVG++ removes the stand-alone fusion Transformer, avoiding this problem. Specifically, to achieve the reported performance on the challenging RefCOCO+ dataset, the preliminary version model, i.e., TransVG, is trained for 180 epochs, while we only optimize TransVG++ for 60 epochs on the same dataset.

Comparison Among Transformer-based Methods: In Table I, we also present a comparison between our framework with other Transformer-based methods, i.e., Refformer [80], VGTR [81], VLTGV [43] and QRNet [44]. QRNet devises query-aware dynamic attention to refine visual features before multi-modal fusion, while Refformer and VLTGV introduce improvements to the stand-alone multi-modal fusion Transformer. Specifically, Refformer adds a query encoder and a visual context decoder to the multi-modal fusion module, and jointly optimizes the model with both annotations of referring expression comprehension (visual grounding) and that of referring expression segmentation. VLTGV performs iteratively verification and reasoning to improve the multi-modal fusion Transformer. Alternatively, our TransVG++ waives the requirements of the stand-alone fusion module, not only introducing fewer parameters but also addressing the aforementioned problem in a simpler and more effective way. As shown in the table, our TransVG++ outperforms other Transformer-based approaches on all the subsets of the evaluated datasets, except for lagging behind QRNet on testB set of RefCOCO dataset.

2) ReferItGame: To further validate the merits of our proposed framework, we also conduct experiments on ReferItGame dataset, and report the performance on the test set. As shown in Table II, both the preliminary TransVG and the advanced TransVG++ largely outperform previous two-stage and one-stage methods. Specifically, with ResNet-50 backbone, TransVG achieves 69.76% top-1 accuracy and outperforms ZSGNet [25] 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. As the first fully-Transformer model, TransVG++ capitalizing on ViT-base further outperforms TransVG with ResNet-101 by 3.97%, showing the effectiveness of our proposed framework to integrate language information into the vision encoders.

Among the competitors, MAttNet [12] is the most representative method that devises multi-modal fusion modules with redefined structures (i.e., modular attention networks to separately model subject, location and relationship). When we compare our model with MAttNet in Tables II and I, we can find that MAttNet shows comparable results to our preliminary version on RefCOCO/RefCOCO+/RefCOCOg, but lags behind ours on ReferItGame. 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 proposed framework effectively avoids this problem by establishing intra-modality and inter-modality correspondence with the flexible and adaptive attention mechanism.

3) Flickr30 K Entities: Table II also reports the performance of our framework on the Flickr30 K Entities test set. On this dataset, our TransVG achieves 79.10% top-1 accuracy with a ResNet-101 backbone network. Our TransVG++ (base) further boosts the accuracy to 81.49%, surpassing the strongest two-stage and one-stage competitors by a remarkable margin (i.e., 1.52% absolute improvement over the state-of-the-art two-stage method M-DGT [38] and 4.75% absolute improvement over the state-of-the-art one-stage method ReSC [11]).

D. Ablative Experiments

In this section, we conduct ablative experiments to investigate the effectiveness of each component in our proposed TransVG and TransVG++ frameworks. These models are optimized according to our reported training details.

1) Design of [REG] Token’s Initial State in TransVG: In Table III, we report the ablation study on how to obtain the initial state of [REG] token in our proposed TransVG. Specifically, we compare our learnable embedding with five other options to generate the initial state of [REG] token (i.e., the embedding

---

Tables and figures are not fully transcribed as the text is presented in a natural format.
We implement a learnable embedding as in (7). We detail these designs and analysis as follows:

- **Average pooled vision tokens**: We perform average pooling over the tokens out of vision branch, and exploit the average-pooled embedding as the initial state.

- **Max pooled vision tokens**: We perform max pooling over the tokens out of vision branch, and exploit the max-pooled embedding as the initial state.

- **Average pooled language tokens**: Similar to the first option, but using the language tokens.

- **Max pooled language tokens**: Similar to the second choice, but using the language tokens.

- **Sharing with [CLS] token**: We use the [CLS] token of language embedding to play the role of [REG] token. In this setting, 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.

Our proposed design to exploit a learnable embedding achieves 72.50% accuracy and 66.35% accuracy on the validation test set of ReferItGame and RefCOCOg-g datasets, respectively, which is the best performance among all the designs. Typically, the initial [REG] token of other designs is either generated from vision or language 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 vision-language Transformer. In TransVG++, we follow the design of TransVG by taking the initial state of the [REG] token as learnable embedding.

**2) Design of Fusion Strategies in TransVG++**: In Table IV, we present ablation studies on the effectiveness of different vision-language fusion strategies. Our preliminary fusion strategy is to concatenate vision tokens and language tokens together and fed them into 6 stand-alone Transformer encoder layers. By leveraging this preliminary strategy, the model achieves 71.00% accuracy and 67.24% accuracy on the validation set of ReferItGame and RefCOCOg-g, respectively.

In general, speaking, models capitalizing on language adapters with MHCA for language adapt token generation, which achieves 73.17% on ReferItGame and 70.60% on RefCOCOg-g, outperforming the preliminary strategy that exploits 6 stand-alone Transformer encoder layers by 2.17% and 3.36%, respectively. We leverage this setting as our default one and report the performance of models with this setting in our main results.

**3) Improvements From TransVG to TransVG++**: In this section, we verify the improvements from TransVG to TransVG++,
i.e., upgrading the vision branch to a fully Transformer-based architecture and removing the stand-alone vision-language fusion module. The language branch of the evaluated models is with the same architecture and is initialized with BERT parameters so that we omit the language branch in these ablative experiments. The involved experiments are conducted on the validation set of RefCOCOg-G [18] dataset. Table V details step-by-step improvements, together with the model size and accuracy. Since the language branch of all the models follows the same configuration, it is omitted in this table. For a clearer comparison, we split LViT into ViT and language adapters in this table. These models are evaluated on the validation set of RefCOCOg-g dataset. It exploits ResNet-50 and the following Transformer encoder layers with ViT-tiny, and the number of parameters and computation cost introduced by our proposed fusion module are reduced from 7.52 M to 0.57 M and the computation cost of performing vision-language fusion are reduced from 7.38 G FLOPs to 0.92 G FLOPs, while the accuracy boosts from 67.24% to 70.60%. This comparison shows the advantages of our proposed fusion-in-backbone mechanism in the aspects of model size, computation overhead, and accuracy. Overall, by making the best of a fully Transformer-based structure and the fusion-in-backbone strategy, model (e) boosts the performance of baseline (a) from 66.35% to 70.60% accuracy.

### E. Discussion on Model Size and Computation Cost

**Comparison:** Table VI presents the model size and computation cost of each component in our method and the representative two-stage/one-stage methods (i.e., MAttNet [12] and ReSC [11]). Here we present the statistical results of TransVG++ method TransVG++. It upgrades the fusion strategy by exploiting our proposed language adapters, which fuses language expression information into the vision tokens at intermediate layers of a ViT backbone network. Compared to model (c), the model parameters of the fusion module are reduced from 7.52 M to 0.57 M and the FLOPs of the prediction head in ReSC are extremely large compared to other methods. The huge computation cost of the vision branch in MAttNet is due to its dependence on region feature extraction and the modular attention networks for multi-modal fusion, which heuristically decompose the subject, object, and relationships of the referred object. The relatively large computation cost of the prediction head in ReSC is also due to its stand-alone multi-modal fusion module, which recurrently performs sub-query reasoning. In contrast to these competitors that exploit complicated and costing fusion modules, the number of parameters and computation cost introduced by our proposed language adapter are negligible. Specifically, in TransVG++ (tiny), the language adapter only introduces 0.57 M parameters.

| Models       | Model Size | Computation Cost |
|--------------|------------|------------------|
|               | V. | L. | Head | V. | L. | Head |
| MAttNet [12] | 57.01M | 4.52M | 4.21M | 447.32G | 443.64M | 16.80M |
| ReSC [11]    | 61.93M | 109.48M | 8.42M | 25.04G | 6.80G | 11.9G |
| TransVG++ (tiny) | 6.39M | 109.63M | 0.12M | 18.46G | 6.80G | 232.46K |

"V." represents the vision feature encoder, "L." indicates the language feature encoder, and "Head" is the prediction head. The FLOPs of each method are obtained according to its default setting.

---

**TABLE V**

**ABLATIVE EXPERIMENTS ON STEP-BY-STEP IMPROVEMENTS FROM TRANSVG TO TRANSVG++**

| Vision Feature | V-L Fusion | Model Size (M) | Acc. (%) |
|----------------|------------|----------------|----------|
| (a) R-50 + 6 enc layers | 6 enc layers | 30.18 | 7.52 | 66.35 |
| (b) R-50-DC5+ 6 enc layers | 6 enc layers | 30.18 | 7.52 | 65.96 |
| (c) ViT-tiny | 6 enc layers | 5.53 | 7.52 | 67.24 |
| (d) ViT-tiny language adapter | 5.53 | 0.57 | 70.60 |

Both the accuracy on the validation set of RefCOCOg-G and the model size are reported. Our default setting is marked in gray.

---

**TABLE VI**

**COMPARISON OF THE MODEL SIZE AND COMPUTATION COST (FLOPS) OF OUR TRANSVG++ AND THE REPRESENTATIVE TWO-STAGE AND ONE-STAGE METHODS (I.E., MATTNET [12] AND RESC [11])**

---

**Baseline (TransVG):** Model (a) is our preliminary baseline, i.e., TransVG [21], which achieves 66.35% accuracy on the validation set of RefCOCOg-g dataset. It exploits ResNet-50 and 6 uni-modal Transformer encoder layers for vision feature extraction, and a stand-alone multi-modal Transformer composed of 6 encoder layers for vision-language fusion. In this model, the vision branch has 30.18 M parameters and the vision-language fusion module has 7.52 M parameters.

**Feature Resolution Increasing:** Model (b) leverages dilation convolution [87] in the last stage of ResNet-50 backbone. By setting the dilation ratio as 2, the downsampling rate of ResNet-50 reduces from 32 to 16. Therefore, the input vision tokens to the following Transformer encoder layers are the same number as that of patches for a ViT backbone. As observed, the performance drops to 65.96%, demonstrating that naively increasing the feature resolution of the fusion module cannot improve the visual grounding accuracy.

**Vision Backbone Upgrading:** Model (c) upgrades the vision feature extraction module of model (a) by replacing ResNet-50 and the following Transformer encoder layers with ViT-tiny, and the model size of (c) is more than 5 times smaller than that of (a), while the accuracy of (a) achieves 67.24%, outperforming (a) by 0.89%. Note that ViT-tiny’s capability for image classification and object detection lags behind ResNet-50, not to mention that followed by 6 Transformer encoder layers, which shows the inconsistency between visual perception and vision-language understanding. Besides, the better performance of (c) demonstrates the advantage of the fully Transformer-based framework.

**Fusion Strategy Upgrading (TransVG++):** Model (d) is our advanced version method TransVG++. It upgrades the fusion strategy by exploiting our proposed language adapters, which fuses language expression information into the vision tokens at intermediate layers of a ViT backbone network. Compared to model (c), the model parameters of the fusion module are reduced from 7.52 M to 0.57 M and the computation cost of performing vision-language fusion are reduced from 7.38 G FLOPs to 0.92 G FLOPs, while the accuracy boosts from 67.24% to 70.60%. This comparison shows the advantages of our proposed fusion-in-backbone mechanism in the aspects of model size, computation overhead, and accuracy. Overall, by making the best of a fully Transformer-based structure and the fusion-in-backbone strategy, model (e) boosts the performance of baseline (a) from 66.35% to 70.60% accuracy.
TABLE VII
ANALYSIS OF THE MODEL SIZE AND COMPUTATION COST (FLOPS) OF OUR TRANSVG++ SERIES

| Models       | Model Size | Computation Cost |
|--------------|------------|------------------|
|              | V. L. Head | V. L. Head       |
| TransVG++ (tiny) | 6.39M 109.63M 0.12M | 18.4G 6.80G 232.46K |
| TransVG++ (small) | 24.57M 109.78M 0.17M | 72.8G 6.80G 330.76K |
| TransVG++ (base) | 96.33M 110.07M 0.26M | 289.4G 6.80G 527.36K |

*“V.” represents the vision feature encoder, “L.” indicates the language feature encoder, and “Head” is the prediction head.

and 0.92 G FLOPs. This demonstrates the parameter efficiency and the computation cost efficiency of our proposed fusion-in-backbone paradigm.

TransVG++ Series Analysis: As shown in Table VII, the parameters of the language branch make up the major part of TransVG++, i.e., 94.4% in TransVG++ (tiny), 81.6% in TransVG++ (small) and 53.3% in TransVG++ (base). Even with the largest ViT-base backbone, the parameters of the vision branch (i.e., LViT) are fewer than that of the language branch. Note that a language Transformer with the same configuration is used and the regression MLPs are with the same hidden dimension in the prediction head, no matter what vision backbone is leveraged. The trivial differences in the model size of these two components are due to the different embedding dimensions for matching different ViT models. We also present the computation cost (FLOPs) of each component in our TransVG++ and other competitors in Table VII. Although the language branch accounts for the majority of parameters in TransVG++, the computation cost of the vision branch (i.e., LViT) is much higher than that of the language branch. Remarkably, in TransVG++ (base), the computation of LViT makes up 97.7% of the total costs, which is 42.47 times higher than that of the language branch. Compared to the whole model, our proposed fusion-in-backbone mechanism with the language adapter only introduces trivial parameters and computation cost. Specifically, in TransVG++ (tiny), it only accounts for 0.49% parameter numbers and 3.64% FLOPs of the whole model.

Future Works: Although our proposed fusion-in-backbone strategy is efficient, there is still much room to reduce the redundancy in vision and language encoders. On the one hand, as the language branch accounts for the major part of parameters, one potential direction to slim our TransVG++ is transferring the knowledge of BERTBASE to a smaller variety of BERT model (e.g., TinyBERT [88]) with knowledge distillation techniques. On the other hand, to reduce the computation cost, the online token pruning technique [89] and sub-window attention strategy [56], [57] can be applied to the vision backbone. The above-mentioned approaches are devoted to improving model efficiency in each uni-modality. How to take multi-modal inputs and devise techniques to address the problem of model efficiency, including parameters efficiency and computation efficiency, is still not fully explored in the literature. We raise this potential direction as an open problem and leave it for future investigation.

F. Discussion on Modality Alignment Paradigms

In this section, we discuss the modality alignment paradigms in visual grounding algorithms and the justification of our LViT. We follow [65] to summarize the modality alignment paradigms into 4 typical categories. These categories are illustrated in Fig. 5, where the height of each block represents its relative computation cost.

Paradigm (a): This paradigm pays the main attention to the visual part while exploiting a comparable simpler language feature encoder and vision-language fusion module. Most of the early works follow this paradigm. One representative work is MAttNet [12], which manually designs three kinds of attention modules that match text to regional features.

Paradigm (b): With the introduction of large-scale pretraining (i.e., BERT) in the NLP community, the models following paradigm (b) pay their attention to improving the power of the language encoder. The representative works FAOA [7] and ZSGNet [25] densely combine language embedding out of a BERT model and vision features out of an object detector for multi-modal interaction. However, the over simple fusion module (e.g., concatenation) in paradigm (b) limits the interaction of vision and language features.

Paradigm (c): This paradigm shifts more attention to the vision-language fusion module. Our preliminary version TransVG [21], ReSC [11], VLTVG [90], etc., belong to this category. These methods rely on a recurrent network or Transformer to improve the power of vision-language fusion modules. These models achieve remarkable improvements compared to paradigm (a) and paradigm (b). However, the fusion modules have to be trained from scratch on limited visual grounding data, while the complicated architecture is hard to be optimized. Besides, the heavy fusion module inevitably leads to large extra computation cost.

Paradigm (d): Compared to other methods, our TransVG++ is with a totally different paradigm, which is summarized as paradigm (d) in Fig. 5. In TransVG++, the stand-alone fusion module is thoroughly removed, while the vision encoder works as both a vision feature extractor and a multi-modal fusion module by exploiting our proposed language prompter/adapter. Our proposed paradigm (d) has three advantages: 1) By re-using the vision feature encoder for multi-modal fusion, the core fusion module is no longer trained from scratch, easing the training phase. 2) The whole model can easily benefit from the ever-growing vision foundation models and scale up with minimal effort. 3) The computation cost is largely reduced compared to leveraging an extra fusion module.

G. Qualitative Result

We showcase the qualitative results of five examples from the RefCOCOg [18] test set in Fig. 6. We observe that both our preliminary and advanced version framework can successfully model queries with complicated relationships, e.g., “A woman see the sea water in back position” in language expression (b). We depict both the predicted bounding box (with rectangles with blue lines) and the [REG] token’s attention over visual
Fig. 5. Four typical modality alignment paradigms of visual grounding methods. The height of each block represents its relative computation cost. Our TransVG++ with Language Conditioned Vision Transformer belongs to the last category.

Fig. 6. Qualitative results of our proposed TransVG and TransVG++ frameworks on the RefCOCOg test set (better viewed in color). We show both the predicted bounding box and the [REG] token’s attention over vision tokens from the last fusion block of each framework.

V. CONCLUSION

In this paper, we present our TransVG (preliminary version) and TransVG++ (advanced version) frameworks to address the problem of visual grounding with Transformers. 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. The advanced version TransVG++ takes a step further, upgrading to a purely Transformer-based architecture and removing the stand-alone fusion modules by integrating language referring information to the ViT backbone. Our TransVG++ serves as a simple, efficient, and accurate framework for visual grounding, and exhibits huge potential for future investigation.

Limitations and Future Directions: Although our proposed TransVG++ shows its advantages compared to previous approaches by embracing the purely Transformer-based architecture and performing multi-modal fusion in the vision backbone, it still has several limitations. For one thing, the language prompter and language adapter are preliminary attempts to develop language conditioned vision Transformer. There are many different possible ways for modality alignment [65], and different finetuning strategies [91]. It is also of great potential to explore the one-tower model [92] and mixture-of-experts techniques [93]. For another thing, TransVG++ can only ground one language expression at a time. To move further in developing a general-purpose language-expression-based detector, the problem formulation and evaluation metrics of visual grounding can be extended, and our TransVG++ can be developed as an auto-regressive model [94]. Besides, to-date vision-language models, including TransVG++, are still too heavy to be deployed on mobile devices, how to make these models slim and run faster is also a valid problem.
[6] Y. Liao et al., “A real-time cross-modality correlation filtering method for referring expression comprehension,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10880–10889.

[7] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, “A fast and accurate object-to-object matching for visual grounding,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4683–4693.

[8] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Representations, 2021.

[9] L. Wang, Y. Li, and S. Lazebnik, “Learning deep structure-preserving image-text embeddings,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 5005–5013.

[10] D. Liu, H. Zhang, F. Wu, and Z.-J. Zha, “Learning to assemble neural module tree networks for visual grounding,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4673–4682.

[11] Z. Yang, T. Chen, L. Wang, and J. Luo, “Improving one-stage visual grounding by recursive sub-query construction,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 387–404.

[12] L. Yu et al., “MattNet: Modular attention network for referring expression comprehension,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1307–1315.

[13] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, “Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1960–1968.

[14] S. Yang, G. Li, and Y. Yu, “Dynamic graph attention for referring expression comprehension,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4644–4653.

[15] S. Yang, G. Li, and Y. Yu, “Graph-structured referring expression reasoning in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9952–9961.

[16] R. Hong, D. Liu, X. Mo, X. He, and H. Zhang, “Learning to compose and reason with language tree structures for visual grounding,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 8, pp. 2765–2779, Aug. 2021.

[17] J. Xiu, Z. Wang, J. Shao, X. Wang, and H. Li, “Improving referring expression grounding with cross-modal attention-guided erasing,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1950–1959.

[18] M. Bajaj, L. Wang, and L. Sigal, “GGraphground: Graph-based language grounding,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4281–4290.

[19] K. Chen, R. Kovvuri, and R. Nevatia, “Query-guided regression network with context policy for phrase grounding,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 824–832.

[20] P. Dogan, L. Sigal, and M. Gross, “Neural sequential phrase grounding (seqground),” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4175–4184.

[21] S. Chen and B. Li, “Multi-modal dynamic graph transformer for visual grounding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 15534–15543.

[22] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018, arXiv:1804.02767.

[23] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2010, pp. 2544–2550.

[24] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Mar. 2015.

[25] S. Yang, G. Li, and Y. Yu, “Propagating over phrase relations for one-stage visual grounding,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 589–605.

[26] L. Yang, Y. Xu, C. Yuan, W. Liu, B. Li, and W. Hu, “Improving visual grounding with visual-linguistic verification and iterative reasoning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 9499–9508.

[27] J. Ye et al., “Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 15502–15512.

[28] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[29] T. Mikolov, M. Karafiát, L. Burget, J. Černỳ, and S. Khudanpur, “Recurrent neural network based language model,” in Proc. 11th Annu. Conf. Int. Speech Commun. Assoc., 2010, pp. 1045–1048.

[30] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured long-term memory networks,” 2015, arXiv:1503.00075.

[31] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, pp. 1735–1780, 1997.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 213–229.

[33] M. Chen et al., “Generative pretraining from pixels,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1691–1703.

[34] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, “Learning texture transformer for image super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5791–5800.

[35] Y. Zeng, J. Fu, and H. Chao, “Learning joint spatial-temporal transformations for video inpainting,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 580–587.

[36] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 10347–10357.

[37] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 10012–10022.

[38] D. Meng et al., “Conditional DETR for fast training convergence,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 3651–3660.
[56] W. Chen et al., “A simple single-scale vision transformer for object localization and instance segmentation,” 2021, arXiv:2112.09747.

[57] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring plain vision transformer backbones for object detection,” 2022, arXiv:2203.16577.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[59] Y.-C. Chen et al., “Uniter: Universal image-text representation learning,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 104–120.

[60] X. Li et al., “Oscar: Object-semantics aligned pre-training for vision-language tasks,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 121–137.

[61] J. Lin, D. Batra, D. Parikh, and S. Lee, “ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 13–23.

[62] W. Su et al., “VL-BERT: Pre-training of generic visual-linguistic representations,” 2019, arXiv:1908.08350.

[63] Z. Yang et al., “TAP: Text-aware pre-training for text-VQA and text-caption,” 2020, arXiv:2012.04638.

[64] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, “Pixel-BERT: Aligning image pixels with text by deep multi-modal transformers,” 2020, arXiv:2004.00849.

[65] W. Kim, B. Son, and I. Kim, “ViLT: Vision-and-language transformer without convolution or region supervision,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 5583–5594.

[66] J. Li, R. R. Selvaraju, A. D. Gomes, S. Joty, C. Xiong, and S. Hoi, “Align before fuse: Vision and language representation learning with momentum distillation,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 9694–9705.

[67] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, “SimVL: Simple visual language model pretraining with weak supervision,” in Proc. Int. Conf. Learn. Representations, 2022.

[68] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, “MDETR-modulated detection for end-to-end multi-modal understanding,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 1780–1790.

[69] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[70] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[71] N. Parmar et al., “Image transformer,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 4055–4064.

[72] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, “Universal transformers,” in Proc. Int. Conf. Learn. Representations, 2018.

[73] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2019, arXiv:1910.10683.

[74] X. Liu et al., “GPT understands, too,” 2021, arXiv:2103.10385.

[75] N. Houlsby et al., “Parameter-efficient transfer learning for NLP,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 2790–2799.

[76] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, “AdapterFusion: Non-destructive task composition for transfer learning,” 2020, arXiv:2005.00247.

[77] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 655–666.

[78] H. J. Escalante et al., “The segmented and annotated IAPR TC-12 benchmark,” Comput. Vis. Image Understanding, vol. 114, pp. 419–428, 2010.

[79] Y. Yu, H. Mao, R. Girshick, and K. He, “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2019, arXiv:1910.10683.

[80] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual annotations: New similarity metrics for semantic inference over event descriptions,” Anna. Meeting Assoc. Comput. Linguistics, vol. 2, pp. 67–70, 2014.

[81] L. Deng et al., “TRANSVG++: END-TO-END VISUAL GROUNDING WITH LANGUAGE CONDITIONED VISION TRANSFORMER” 13651

[82] Y.-L. Sung, J. Cho, and M. Bansal, “LST: Ladder side-tuning for parameter and memory efficient transfer learning,” Proc. Adv. Neural Inf. Process. Syst., vol. 35, pp. 12991–13005, 2022.

[83] W. Wang et al., “Image as a foreign language: BEIT pretraining for all vision and vision-language tasks,” 2022, arXiv:2208.10442.

[84] Z. Riquelme et al., “Scaling vision with sparse mixture of experts,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 8583–8595.

[85] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton, “Pix2Seq: A language modeling framework for object detection,” in Proc. Int. Conf. Learn. Representations, 2021.

Jiajun Deng received the BE and PhD degrees from the department of Electrical Engineering and Information Science, University of Science and Technology of China (USTC), in 2016 and 2021, respectively. He is a postdoctoral researcher with the University of Science and Technology of China (USTC). His research interests include computer vision and multi-modal learning.

Zhengyuan Yang received the BE degree in electrical engineering from the University of Science and Technology of China (USTC), in 2016, and the PhD degree in computer science from the University of Rochester (UR), in 2021. He is a researcher with Microsoft. His research interests are vision+language and multi-modal learning.

Daqing Liu received the PhD degree from the Department of Automation, University of Science and Technology of China (USTC), in 2021. He is an algorithm scientist with JD Explore Academy of JD.com Inc China. He used to be a research assistant at Nanyang Technological University (NTU) from 2018 to 2019. His current research interest is multi-modality learning, including multi-modality understanding, retrieval, and generation.

Tianlang Chen received the BE degree in electrical engineering from the University of Science and Technology of China (USTC), in 2016, and the master’s and PhD degrees from the Computer Science Department, University of Rochester (UR), in 2019 and 2021, respectively. He is an applied scientist with Amazon. His research interests mainly include tasks related to joint visual-textual learning, such as image captioning, image-text matching, visual question answering, and visual grounding.
Wengang Zhou (Senior Member, IEEE) received the BE degree in electronic information engineering from Wuhan University, China, in 2006, and the PhD degree in electronic engineering and information science from the University of Science and Technology of China (USTC), China, in 2011. From 2011 to 2013, he worked as a post-doc researcher with the Computer Science Department, University of Texas at San Antonio. He is currently a professor with the EEIS Department, USTC. His research interests include multimedia information retrieval, computer vision, and computer game. In those fields, he has published more than 100 papers in ACM/IEEE Transactions and CCF Tier-A International Conferences. He is the recipient of the Best Paper Award for ICIMCS 2012. He served as the publication chair of IEEE ICME 2021.

Yanyong Zhang (Fellow, IEEE) received the BS from the University of Science and Technology of China (USTC), in 1997, and the PhD from Penn State University, in 2002. From 2002 and 2018, she was on the faculty of the Electrical and Computer Engineering Department, Rutgers University. She was also a member of the Wireless Information Networks Laboratory (Winlab). Since July 2018, she joined the School of Computer Science and Technology with USTC. She has 21 years of research experience in the areas of sensor networks, ubiquitous computing, and high-performance computing, and has published more than 140 technical papers in these fields. She received the NSF CAREER award in 2006. She has served/currently serves as the associate editor for several journals, including IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, IEEE Transactions on Dependable and Secure Computing, and Elsevier Smart Health. She has served on various conference TPCs including DSN, Sensys, Infocom, etc. She is the TPC co-chair for IPSN’21.

Houqiang Li (Fellow, IEEE) received the BS, the MEng, and the PhD degrees in electronic engineering from the University of Science and Technology of China, Hefei, China, in 1992, 1997 and 2000, respectively. He is a professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. His research interests include multimedia search, image/video analysis, video coding and communication. He has authored and co-authored more than 200 papers in journals and conferences. He is the winner of National Science Funds (NSFC) for Distinguished Young Scientists, the distinguished professor of Changjiang Scholars Program of China, and the leading scientist of Ten Thousand Talent Program of China. He served as an associate editor of IEEE Transactions on Circuits and Systems for Video Technology from 2010 to 2013. He served as the TPC co-chair of VCIP 2010, and he served as the general co-chair of ICME 2021. He is the recipient of National Technological Invention Award of China (second class), in 2019 and the recipient of National Natural Science Award of China (second class), in 2015. He was the recipient of the Best Paper Award for VCIP 2012, the recipient of the Best Paper Award for ICIMCS 2012, and the recipient of the Best Paper Award for ACM MUM in 2011.

Wanli Ouyang (Senior Member, IEEE) received the PhD degree from the Department of Electronic Engineering, The Chinese University of Hong Kong (CUHK). He is currently an associate professor with the School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia. His research interests include image processing, computer vision, and pattern recognition.