TRAR: Routing the Attention Spans in Transformer for Visual Question Answering

Yiyi Zhou[12], Tianhe Ren[12], Chaoyang Zhu[12], Xiaoshuai Sun[12*], Jianzhuang Liu[3],
Xinghao Ding[2], Mingliang Xu[4], Rongrong Ji[12]
[1]Media Analytics and Computing Lab, School of Informatics, Xiamen University, China
[2]School of Informatics, Xiamen University, China
[3]Noah's Ark Lab, Huawei Technologies [4]Zhengzhou University, China
{zhouyiyi,xssun,dxh,rrji}@xmu.edu.cn, {rentianhe,cyzhu}@stu.xmu.edu.cn
liu.jianzhuang@huawei.com, iexumingliang@zzu.edu.cn

Abstract

Due to the superior ability of global dependency modeling, Transformer and its variants have become the primary choice of many vision-and-language tasks. However, in tasks like Visual Question Answering (VQA) and Referring Expression Comprehension (REC), the multi-modal prediction often requires visual information from macro- to micro-views. Therefore, how to dynamically schedule the global and local dependency modeling in Transformer has become an emerging issue. In this paper, we propose an example-dependent routing scheme called TRAnsformer Routing (TRAR) to address this issue[1]. Specifically, in TRAR, each visual Transformer layer is equipped with a routing module with different attention spans. The model can dynamically select the corresponding attentions based on the output of the previous inference step, so as to formulate the optimal routing path for each example. Notably, with careful designs, TRAR can reduce the additional computation and memory overhead to almost negligible. To validate TRAR, we conduct extensive experiments on five benchmark datasets of VQA and REC, and achieve superior performance gains than the standard Transformers and a bunch of state-of-the-art methods.

1. Introduction

After gaining dominance in the field of natural language processing [60, 10, 74, 9, 27], Transformer [60] also becomes the prime choice of many vision-and-language (V&L) tasks [14, 7, 30]. More and more researchers [39, 71, 79, 58, 22] follow the design paradigm of Transformer to propose various multi-modal networks, achieving new state-of-the-art performance on various benchmarks [14, 30, 28, 7]. Their great success largely attributes to the superior global dependency modeling of *self-attention* (SA), which can not only capture the relationships within modalities, but also facilitate the vision and language alignments.

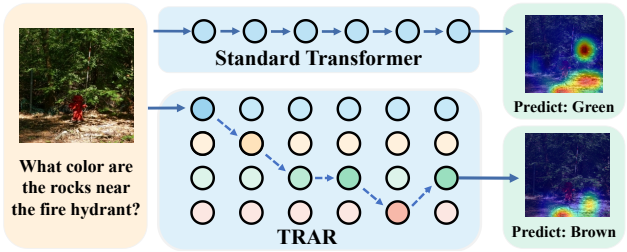


Figure 1: Illustration of our Transformer Routing (TRAR) and the traditional static Transformer. Circles denotes the self-attention modules, and their colors represent different attention spans (receptive fields). TRAR can dynamically schedule the attention spans for each example.

However, in some V&L tasks, such as *visual question answering* (VQA) [14] and *referring expression comprehension* (REC) [30], the multi-modal inference often requires visual attentions from different receptive fields. As shown in Fig. 1, to answer the question, the model should not only understand the overall semantic, but more importantly, it also requires to capture the local relationships. In this case, only relying on the global dependency modeling in SA is still insufficient to meet such a requirement. This finding is also supported by the recent development of image Transformers [38, 62]

Such an issue becomes more prominent in the end-to-end multi-modal inference. After Jiang *et al.* [25] revealed that well pre-trained grid features can also have expressive de-

*Corresponding author.
[1]Source code: https://github.com/rentainhe/TRAR-VQA/

scription power, recent endeavors [22, 41, 76] have begun to re-pursue the design of one-stage V&L models. However, compared with the widely used detection features [2], the semantic information of grid features are more fragmented. Therefore, the global dependency modeling of SA is more likely to introduce noise during attention, disturbing the model inference, *e.g.*, associating unrelated regions.

To deal with this problem, helping Transformer networks to explore different attention spans has become an emerging demand. An intuitive solution is to build a dynamic and hybrid network, where each layer has a set of attention modules of different receptive fields. Then, the model can select the suitable ones according to the given example. However, the direct application of this solution might be counterproductive, because the additional parameters and computations will further exacerbate the model costs, which is already the main criticism of Transformer [27].

In this paper, we propose a novel yet lightweight routing scheme called *Transformer Routing* (TRAR), which achieves the automatic selection of attentions with negligible additional computation and memory overhead. Specifically, TRAR equips each visual SA layer with a path controller to predict the next attention span (or receptive field) [45] based on the output of the previous step. To address the issue of redundant parameters and computation, TRAR regards SA as a feature update function for the densely connected graph [79], and constructs different adjacency masks for the defined attention spans. Afterwards, the task definition of module selection can be converted to the one of mask selection, reducing the additional cost to a large degree.

To validate TRAR, we apply it to the single-stage Transformer networks for two multi-modal tasks, VQA and REC, and conduct extensive experiments on five benchmark datasets, VQA2.0 [14], CLVER [28], RefCOCO [68], RefCOCO+ [68] and RefCOCOg [43]. The experimental results not only confirm the merits of TRAR over the default Transformer network, but also show new SOTA performance on multiple benchmarks[2], *e.g.,* 72.7 on VQA2.0 [14] and 68.9 on RefCOCOg [43].

In summary, the main contributions of this paper are threefold:

- We reveal the issue of attention span, which is critical for the development of end-to-end Transformers.
- We propose the first example-dependent routing scheme for Transformer to dynamically schedule global and local attentions, which requires negligible additional computation and memory overhead.
- The proposed TRAR helps the one-stage Transformers achieve new SOTA performances on multiple benchmark datasets of VQA and REC.

[2]Single model without the large-scale BERT style pre-training.

2. Related Work

2.1. Visual Question Answering

Visual question answering (VQA) is a task of answering human questions based on given images. It is often considered as a classification task with fixed categories [3, 14, 28]. The rapid development of VQA has been supported by the emergence of various benchmark datasets [1, 3, 14, 28, 29, 32, 52] and methods [13, 59, 66, 72, 54, 46, 77, 78, 79, 80]. With the prevalence of Transformer networks [60], recent advances in VQA also resort to stacking multiple attention layers, *e.g.,* Bilinear Attention layers [31] or Self-Attention Layers [71], for capturing relationships within and across modalities. The success of large-scale unsupervised pretraining in NLP [11] has further promoted the popularization of Transformer and its variants in VQA, leading to a new trend of large-scale V&L pretraining [39, 40, 58, 55, 22].

2.2. Referring Expression Comprehension

Referring expression comprehension (REC) is a task of grounding (locating) target regions in an image based on the given natural-language expressions [30], which is also known as *visual grounding*. Recent years have witnessed its fast advancements with a bunch of methods [6, 49, 20, 19, 69, 67, 76, 41], which can be roughly divided into two main categories. The first one is multi-stage modeling [6, 49, 20, 19, 69, 67], which typically regards REC as a metric learning problem, *i.e.*, selecting the best region from a set of proposals/objects based on the given expression. In these methods, a joint embedding network for two modalities is built for computing the matching degree of each region-expression pair [67]. The other one is single-stage modeling, [65, 76, 41, 64], which considers REC as a language-guided detection task. These methods typically embed a language encoder into a detection network like YOLOv3 [51], and perform multi-modal regressions to obtain the referent. In this paper, we mainly focus on the single-stage modeling of REC.

2.3. Dynamic Neural Networks

Dynamic neural networks are an emerging research topic in deep learning. Differing from the traditional static network structures, dynamic networks can adapt their structures or parameters to the given example during inference, yielding appealing properties like better representation power, adaptiveness, compatibility and generality [15]. According to architecture design, the research of dynamic networks can be categorized into three main directions,

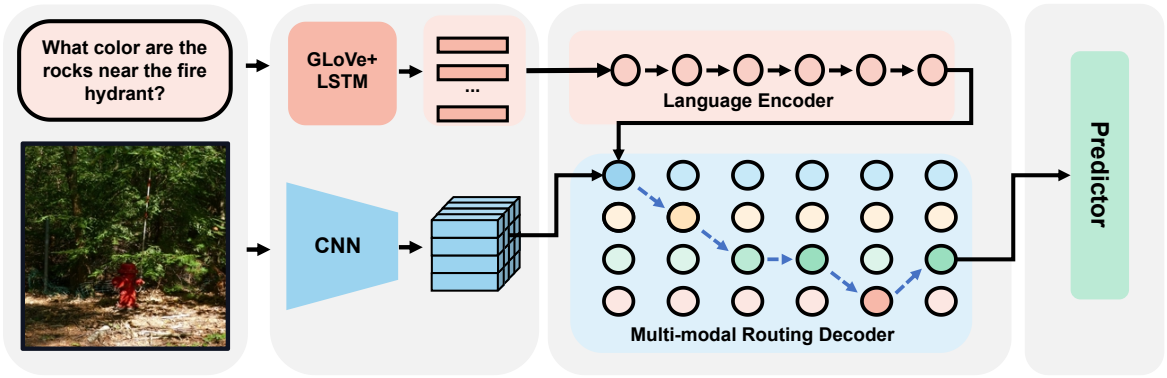


Figure 2: The framework of the proposed Transformer Routing (TRAR) scheme for the one-stage Transformer. The different colors of the decoding layers denote visual attentions of different spans. With TRAR, the Transformer can dynamically select the visual attention span of each step, so as to form the optimal reasoning path for each example.

which are *dynamic depth* for network early exiting [5] or layer skipping [35, 61], *dynamic width* for skipping neurons [4] or channels [37], and *dynamic routing* for multi-branch or tree structure networks [21, 34, 63]. Our method belongs to the last direction. Most of existing dynamic routing models [34, 63] are built on super networks, where each routing option is an independent module. Although the inference efficiency can be maintained via categorical choice or budget constraints [34], the network parameters and computations are very redundant during training, which leads to a huge demand for experimental resource.

3. Transformer Routing

The framework of the proposed Transformer Routing (TRAR) is given in Fig. 2. In the following sub-sections, we introduce its routing process, path controller, attention spans, optimization, and network structure.

3.1. Routing Process

To achieve the goal of dynamic routing (selection) for each example, an intuitive solution is to create a multi-branch network structure, where each layer is equipped with modules of different settings. Specifically, given the features of the last inference step, $\mathbf{X} \in \mathbb{R}^{n \times d}$, and the routing space, $\mathbf{F} = [\mathbf{F}_0, .., \mathbf{F}_n]$, where n denotes the number of features and d is the feature dimension, the output of the next inference step $\mathbf{X}'$ is obtained by:

$$\mathbf{X}' = \sum_{i=0}^{n} \alpha_i \mathbf{F}_i \left(\mathbf{X}\right), \tag{1}$$

where α is the path (selection) probability predicted by the path controller (described in Sec.3.2) and $\mathbf{F}$ is a set of modules. During testing, α can be either binarized for a *hard* selection, or remain continuous to obtain a *soft* routing [34].

However, from Eq. 1 we can see that such a routing scheme can inevitability make the network become very cumbersome, and exacerbates the training overhead greatly. Besides, due to the *dot-product* operations in SA, Transformer has been long criticized for its expensive computation and memory footprint [27].

In this case, it is critical to optimize the definition of path routing to alleviate the burden of experiments. By revisiting the definition of the standard self-attention, defined as:

$$\begin{aligned} \mathbf{X}' &= SA\left(\mathbf{X}\right) = \mathbf{A}\mathbf{X}\mathbf{W}_v, \\ \mathbf{A} &= \text{Softmax}\big(\frac{\left(\mathbf{X}\mathbf{W}_q\right)^T \mathbf{X}\mathbf{W}_k}{\sqrt{d}}\big), \end{aligned} \tag{2}$$

we can see that SA can be regarded as a feature update function for a fully connected graph, when $\mathbf{A} \in \mathbb{R}^{n \times n}$ is regarded as a weighted adjacency matrix [79].

In this case, to obtain the features of different attention spans, we just need to limit the graph connections of each input element. It can be accomplished by placing an adjacency mask $\mathbf{D} \in \mathbb{R}^{n \times n}$ after the *dot-product* operation, which is used to calculate the coupling coefficents between all elements. Its formulation is:

$$\mathbf{A} = \text{Softmax}\big(\frac{\left(\mathbf{X}\mathbf{W}_q\right)^T \mathbf{X}\mathbf{W}_k}{\sqrt{d}}\mathbf{D}\big), \tag{3}$$

where the values of $\mathbf{D}$ are binary and set to 1 if they are within the attention span of the target element[3]. Therefore, attention is only performed within the defined spans.

Based on Eq. 3, a routing layer for SA is then defined as:

$$\text{SA}_R\left(\mathbf{X}\right) = \sum_{i=0}^{n} \alpha_i \text{Softmax}\big(\frac{\left(\mathbf{X}\mathbf{W}_q\right)^T \mathbf{X}\mathbf{W}_k}{\sqrt{d}}\mathbf{D}_i\big)\mathbf{X}\mathbf{W}_v, \tag{4}$$

[3]During deployment, the zero values will be replaced by a large negative value.

where $\mathbf{W}_q, \mathbf{W}_k$ and $\mathbf{W}_v$ can be shared between different SA layers, thereby reducing the parameter size.

However, Eq. 4 is still computationally expensive. So we further simplify the problem of module selection to the choice of adjacency masks, $\mathbf{D}$, defined as:

$$\text{SA}_R(\mathbf{X}) = \text{Softmax}\big(\frac{(\mathbf{X}\mathbf{W}_q)^T \mathbf{X}\mathbf{W}_k}{\sqrt{d}} \sum_{i=0}^{n} \alpha_i \mathbf{D}_i\big)\mathbf{X}\mathbf{W}_v. \tag{5}$$

From Eq. 5 we can see that the additional computation and memory footprint can be reduced to almost 0, and the goal of selecting features from different attention spans still can be accomplished.

3.2. Path Controller

In TRAR, each visual SA layer is equipped with a path controller to predict the probabilities of routing options, *i.e.,* the module selection. Specifically, given the input features $\mathbf{X} \in \mathbb{R}^{n \times d}$, the path probabilities $\alpha \in \mathbb{R}^n$ is defined as:

$$\begin{aligned} \alpha &= \text{Softmax}\left(\text{MLP}(f_{att})\right), \\ f_{att} &= \text{AttentionPool}\left(\mathbf{X}\right). \end{aligned} \tag{6}$$

Here, MLP refers to the multi-layer predictor, and AttentionPool is the attention based pooling method [71].

Eq. 6 can be replaced by other controller designs like *Gating Function* [34]. But during experiments, we found this *softmax*-based predictor is more efficient.

3.3. Attention Span

Attention span denotes the receptive field of attention features, and its definition is not new in both natural language processing (NLP) and computer vision (CV) communities [57, 45, 50]. In compute vision, Parmar *et al.* [45] borrowed the *sliding-window* design of convolution to make each visual region interact with its neighbors within a constrained receptive field, *e.g.,* 3×3 or 5×5. Our definition is similar to it, except that we use the concept of *order neighborhood* from graph topology [12] to denote different degrees of attention spans, *e.g.,* $1st$-order neighborhood equal to 3×3, and $2nd$-order is 5×5, and so on. This definition allows most elements to be located at the center of the attention span, which is also theoretically consistent with the routing process defined in Sec. 3.1.

3.4. Optimization

In this paper, we provide two types of inference methods for TRAR, namely, *soft routing* and *hard routing*.

Soft routing. As shown in Eq. 5, by applying the *softmax* function, we relax the categorical choice of the routing path to a continuous and differentiable operation. Then, the controller weights can be jointly optimized with the Transformer weights according to the task objective, *i.e.*, $\arg min_{w,z} \mathcal{L}_{train}(w, z)$, where w and z are the weights of the Transformer and controllers, respectively. During testing, the features of different attention spans are dynamically combined, which is similar to most soft routing schemes [34]. Since *soft routing* requires no additional hyper-parameter tunning, it is relatively easy to train. The efficiency is also not affected by the dynamic feature aggregations, as analyzed in Sec. 3.1.

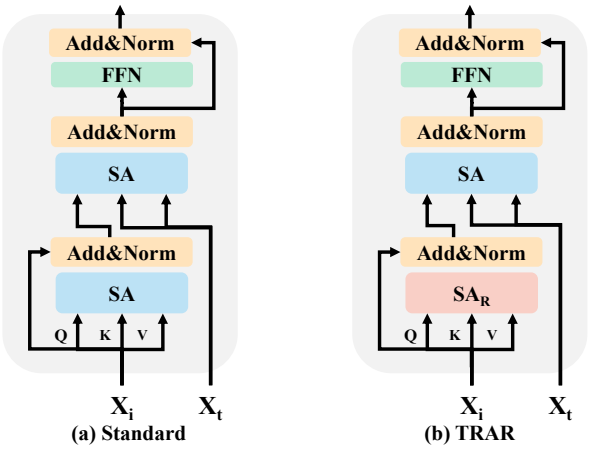


Figure 3: Illustration of the implementation of our routing module on the multi-modal decoding layer [71]. $\mathbf{X}_i$ and $\mathbf{X}_t$ denote the visual and textual features, respectively. SA_R is the SA routing layer. *Add&Norm* denotes addition and layer normalization, respectively, and *FFN* is the feed-forward network [60].

Hard routing. Hard routing is to achieve binary path selection, which can further introduce specific CUDA kernels [45, 50] to accelerate the model inference. However, the categorical routing makes the weights of the controllers non-differentiable, and directly binarizing the results of the soft routing might lead to the feature gap between training and testing. To handle this problem, we introduce the *Gumbel-max* trick [24] to achieve differential path routing, *i.e.,* replacing *softmax* in Eq. 5 with *Gumbel softmax*:

$$\alpha_i = \frac{\exp\left((\log(\pi_i) + g_i)/\tau\right)}{\sum_j \exp\left((\log(\pi_j) + g_j)/\tau\right)}, \tag{7}$$

where g_i are *i.i.d* samples drawn from Gumbel(0,1) [24], τ is the *softmax* temperature, π is the *log softmax* probability. In the initial training phase, τ is set to a larger value, *e.g.,* 10, and will be reduced as the training progresses. When τ approaches 0, the *Gumbel softmax* become one-hot, which is identical to the categorical distribution. In terms of optimization, we can use the chain rule to compute path-wise gradients from the Transformer network to the controllers.

3.5. Network Structure

We build the routing network based on the representative multi-modal Transformer proposed by [71], also termed MCAN. Concretely, similar to the standard Transformer [60], MCAN has six encoding layers used to modeling the language features extracted by an LSTM [17], and six decoding layers for processing visual features and cross-modal alignments simultaneously. During deployment, we replace the visual SA module with the proposed routing module, *i.e.,* Eq. 5, as shown in Fig. 3.

In VQA, the routing network use a convolution neural network (CNN), *e.g.,* ResNetx-101 [25], as the visual backbone. Two *AttentionPooling* layers [71] are added after the language and visual outputs of the Transformer, where the attention feature vectors of the two modalities are combined as the joint representation, which are followed by a multi-layer predictor for multi-label classification. The network structure for REC is similar to that for VQA. The difference is that we apply an additional multi-scale fusion scheme [41] to enhance the description power of the grid features following the setting in [41]. For REC, we use the regression layer from YOLOv3 [51] as the predictor.

4. Experiments

To validate the proposed TRAR, we apply it to two highly competitive V&L tasks, namely *visual question answering* (VQA) [14] and *referring expression comprehension* [30] (REC, also known as *visual grounding*), and conduct extensive experiments on five benchmark datasets, VQA2.0 [14], CLVER [28], RefCOCO [68], RefCOCO+ [68] and RefCOCOg [43].

4.1. Datasets

VQA2.0 [14] is a widely-used benchmark dataset for VQA developed based on VQA1.0 [3]. It also uses images from MS-COCO [52] and has about 1,105,904 VQA examples, of which 443,757, 214,254 and 447,793 examples for training, validation and testing, respectively. Compared to VQA 1.0, the dataset distribution is more balanced.

RefCOCO (UNC RefExp) [68] has 142,210 referring expressions for 50,000 bounding boxes in 19,994 images from MS-COCO [36], which is split into *train*, *validation*, *Test A* and *Test B* with 120,624, 10,834, 5,657 and 5,095 samples, respectively. The expressions are typically short sentences with an average length of 3.5 words. Test A are about people while Test B are objects.

RefCOCO+ [68] has 141,564 expressions for 49,856 boxes in 19,992 images from MS-COCO. It is also divided into splits of *train* (120,191), *val* (10,758), Test A (5726) and Test B (4,889). Compared to RefCOCO, its expressions include more appearances (attributes) than absolute locations [30] to describe the target box.

RefCOCOg (Google RefExp) [43] has 95,010 expressions for 49,822 boxes in 25,799 images from MS-COCO. The split has 85,474 and 9,536 samples for training and validation, respectively. Since the *test* set is not released, we use the *UNC* partitions [68, 67] of the *val* split for validation and testing. Compared to the above two datasets, the expressions in RefCOCOg are collected in a non-interactive way, and the lengths are longer (8.4 words on average).

CLEVR [28] is a synthetic VQA dataset introduced by Johnson *et al.* [28], which aims to examine various reasoning skills, *e.g.*, relation and counting. It contains 70K images and 700k questions.

Metric. For VQA2.0, we use the VQA accuracy [3] as the evaluation metric, while on CLEVR, the classification accuracy is used. For REC benchmarks, we follow the setting in [30] to use the IoU accuracy as the metric, *i.e.,* the prediction is correct when the overlapping degree (IoU) between the predicted bounding box and the ground truth one is larger than 0.5

4.2. Implementation

Most of the deployments of TRAR for VQA and REC are similar. LSTM [17] is used as the language encoder, and its dimension is set to 512. The input text words are initialized by GLOVE [47] embeddings with a dimension of 256. All transformers have 6 encoding layers for language modeling, and 6 decoding layers for visual attention and multi-modal interaction [71]. The dimensions of the self-attention and FFN are 512 and 2,048, respectively, and the number of attention heads is 8. For the path controller, its hidden dimension is 1,024. For *Gumbel Softmax*, the maximal value of temperature τ is set to 10.0, while the minimal one is 0.1. τ updated by :

$$\tau_i = \tau_{max} - (\tau_{max} - \tau_{mini}) * i/(m-1), \qquad (8)$$

where i denotes the i-th epoch and m is the number of total training epochs.

On VQA2.0, the visual backbone is ResNext152 [25] pre-trained on *Visual Genome* [32]. Its grid features are first padded to the scale of 16×16, and then pooled by a kernel size of 2×2 with a stride of 2. So the resolution for Transformers is 8×8. We defined 3 adjacency masks with the neighborhood orders of 1*st*, 2*nd* ,3*rd*, corresponding to the sizes of 3×3, 5×5, and 7×7. On CLEVR, the backbone is ResNet-101 [16] provided by the dataset, and the resolution of its features is 13×13. We use the adjacency masks of 1*st* and 3*rd* on CLEVR. The numbers of training epochs for VQA and CLEVR are 13 and 16, respectively, where the first three epochs are for model warming. The batch size is set to 64. The learning rates are all set to $1e$-4, which are decayed by 0.2 on the 10*th*, 13*th* and 15*th* epochs.

On three REC datasets, we use DarkNet [51] as the backbone. For fair comparisons with SOTA methods [73, 67,

Table 1: Comparison with different baselines on VQA2.0 *val* set and RefCOCO. * denotes only counting the parameter size of the Transformer. MAdds [18] is the metric for computation efficiency.

VQA	Para.*	MAdds	All	Yes/No	Num.
Base	45M	2.8G	67.3	85.0	49.1
Routing	67M	8.0G	67.7	85.3	50.3
Routing$_{WS}$	45M	8.0G	67.5	85.1	49.4
TRAR$_S$	45M	2.8G	67.7	85.2	49.6
TRAR$_H$	45M	2.8G	67.6	85.2	49.9
REC	**Para.**	**MAdds**	**val**	**TestA**	**TestB**
Base	45M	2.8G	75.8	77.0	68.5
Routing	73M	9.2G	75.1	78.4	70.0
Routing$_{WS}$	45M	9.2G	75.5	78.7	69.8
TRAR$_S$	45M	2.8G	77.6	80.1	70.7
TRAR$_H$	45M	2.8G	77.3	79.6	70.4

Table 2: Effects of routing options *w.r.t.* neighborhood orders on the VQA2.0 *val* set and RefCOCO *val* set. 0* denotes that no mask is used.

	VQA		REC	
Orders	**TRAR$_S$**	**TRAR$_H$**	**TRAR$_S$**	**TRAR$_H$**
[0*]	67.3	67.3	75.8	75.8
[0,1]	67.6	67.5	76.2	76.5
[0,1,2]	67.7	67.6	76.7	76.9
[0,1,2,3]	67.6	67.6	77.6	76.1
[0,1,2,3,4]	-	-	77.5	77.3

64], we test two backbones, which are pre-trained on the complete and incomplete MSCOCO datasets [67], respectively. We use the outputs of *Layer26*, *Layer43* and the last layer as the input visual features, which are processed by a multi-scale fusion module [41], and the output scale is 13×13. The adjacency masks of 1*st*, 2*nd*, 3*rd* and 4*th* orders are used for TRAR. The batch size is set to 64, and the number of training epochs is 45, 3 of which are for model warming. The learning rates are all set to $1e$-4, which are decayed on the 20*th* and 30*th* epochs.

4.3. Experimental Analysis

Ablations. We first compare the proposed TRAR with a set of baselines on VQA and REC, the results of which are given in Tab. 1. In Tab. 1, *Base* denotes the default multi-modal Transformer, and *Routing* refers to the conventional routing scheme defined in Eq. 1. *Routing$_{WS}$* denotes the weight-sharing setting of *Routing*, as defined in Eq. 4. TRAR with *soft routing* and *hard routing* is denoted as *TRAR$_S$* and *TRAR$_H$*, respectively. From Tab. 1, we can observe that all routing schemes can bring performance improvements to Transformer, leading to up to +4% improvement on TestA of RefCOCO. Such a result validate our motivation of attention span routing. Notably, the additional costs of parameter size and computation of TRAR are very little. In contrast, the conventional routing scheme, *i.e.,* Routing, increases about 66% and 184%, respectively. More importantly, on most metrics, TRAR can outperform Routing and Routing$_{WS}$ slightly, showing its better efficiency in model training. These observations strongly confirm the merits of TRAR.

Table 3: Comparison with the state-of-the-arts on VQA2.0 with single model and no large-scale pre-training.

	Test-dev				Test
Method	**All**	**Yes/No**	**Num.**	**Others**	**All**
BoUp[59]	65.32	81.82	44.21	57.26	65.67
Pythia [26]	68.49	-	-	-	-
BAN [31]	70.04	85.42	54.04	60.52	70.35
DFAF [46]	70.22	86.09	53.32	60.49	70.34
ReGAT [33]	70.27	86.08	54.42	60.33	70.58
MCAN [71]	70.63	86.82	53.26	60.72	70.90
AGAN [79]	71.16	86.87	54.29	61.56	71.50
MMNAS [70]	71.24	87.27	55.68	61.05	71.46
Base	71.45	87.43	53.80	61.81	-
TRAR$_S$	72.00	87.43	54.69	62.72	-
TRAR$_H$	71.82	87.49	53.84	62.52	-
TRAR$_S$*	**72.62**	**88.11**	**55.33**	**63.31**	**72.93**

$*$ With grid features of resolution 16×16.

We also examine the effects of TRAR's routing space, *i.e.*, the selection of neighborhood orders, which is shown in Tab. 2. The first observation is that adding a local attention mask, *i.e.,* 1*st* order, can lead to a distinct performance improvement, which validates the importance of local dependency modeling in VQA and REC. We also find that the benefit of high-order masks is less obvious, the receptive fields of which are close to the standard SA. The other finding is that TRAR works better on high-resolution feature maps, *e.g.*, the ones of REC with a scale of 13×13. Such an advantage is also confirmed in the experiments of CLEVR as shown in Tab. 4.

Comparison with SOTAs. We further compare TRAR with SOTA methods on VQA2.0. The results are given in Tab. 3. It can be seen that TRAR not only outperforms these Transformers, but also achieves the new SOTA performance on this highly competitive benchmark. We also validate TRAR on the other widely used benchmark CLEVR [28], the results of which are given in Tab.4. CLEVR is a dataset that focuses on visual reasoning, where the questions are usually longer and more complex compared to VQA2.0. On CLEVR, the performance gains of TRAR becomes more obvious, which further confirms the effectiveness of the routing attention spans.

The comparison results with SOTA methods of REC are

Table 4: Comparison with the state-of-the-arts on CLEVR. * denotes that the program annotations are used.

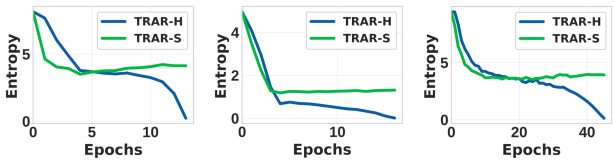
Method	Overall	Count	Exist	Comp_Num	Query_Attr	Comp_Attr
Human [28]	92.60	86.70	96.60	86.40	95.00	96.00
FILM [48]	97.60	94.50	99.20	93.80	99.20	99.0
XNM [53]*	97.70	96.00	98.70	98.00	98.40	97.60
DDRprog [56]*	98.30	96.50	98.80	98.40	99.10	99.0
TBD [44]*	98.70	96.80	98.90	99.10	99.40	99.60
MAC [23]	98.90	97.20	99.50	99.40	99.30	99.50
NS-CL [42]	98.90	**98.20**	98.80	99.00	99.30	99.10
Base	98.54	96.34	99.24	98.60	99.43	98.93
TRAR$_S$	99.00	97.53	**99.55**	99.10	**99.66**	99.12
TRAR$_H$	**99.10**	97.65	99.54	**99.42**	99.62	**99.40**

Table 5: Comparison with the state-of-the-arts on RefCOCO, RefCOCO+ and RefCOCOg. * denotes that the backbone is pre-trained on the complete MS-COCO dataset.

	RefCOCO		RefCOCO+		RefCOCOg	
Two-Step	**TestA**	**TestB**	**TestA**	**TestB**	**Val**	**Test**
Listener[69]	73.1	64.9	60.0	49.6	59.3	59.2
VarCN[75]	73.3	67.4	58.4	53.2	-	-
PAtt[81]*	75.3	65.5	61.3	50.8	-	-
DPPN[73]*	76.9	67.5	60.0	49.6	-	-
MattNet[67]	80.4	69.3	**70.3**	56.0	66.7	67.0
One-Step	**TestA**	**TestB**	**TestA**	**TestB**	**Val**	**Test**
FAOA [65]	74.9	66.3	61.9	49.5	59.4	58.9
SSG [8]*	76.5	67.5	62.1	49.3	-	58.8
RSC [64]*	80.5	72.3	68.4	56.8	67.3	67.2
GIN [76]	78.7	72.7	67.2	54.2	62.7	62.3
GIN [76]*	81.1	77.3	65.5	57.4	65.5	65.6
Base	77.0	68.4	62.3	51.9	62.9	62.3
TRAR$_S$	80.1	70.7	67.9	54.9	64.1	64.2
TRAR$_H$	79.6	71.3	65.1	53.5	63.3	62.5
TRAR$_S$*	81.4	**78.6**	69.1	56.1	**68.9**	**68.3**
TRAR$_H$*	**81.5**	77.3	66.9	**57.8**	66.1	65.8

shown in Tab. 5. From this table, we can first observe that the performance gains of TRAR over Transformer are more distinct on REC, *e.g.,* +9% on TestA of RefCOCO+. Meanwhile, compared with the existing one-stage SOTAs, TRAR also shows obvious merits in performance, suggesting its generalization ability for V&L tasks. Compared with the two-stage methods, which are much less efficient, the overall performance of TRAR is still superior.

In summary, these results greatly validate the effectiveness of TRAR. We believe its contribution to the V&L community is significant.



(a) VQA2.0 (b) CLVER (c) RefCOCO

Figure 4: The change of routing entropies of TRAR with hard routing (TRAR$_H$) and soft routing (TRAR$_S$).

4.4. Qualitative Analysis.

Fig. 4 depicts the changes of TRAR's routing entropies. From this figure, we can see that after a short period of training, the entropies of *soft routing* and *hard routing* have significantly declined, suggesting that the model has been able to select the attention spans. The difference is that as the training progresses, the entropy of soft routing will become steady, while the one of hard routing will continue to decrease until it approaches zero.

To obtain a deep insight into the reasoning process of TRAR, we also visualize the attentions during inference in Fig. 5. We compare its results with those of the standard Transformer in the first sub-figure. It can be seen that the attention of Transformer is often divergent and random, which can easily associate the target region with unrelated ones, as we analyze in Sec. 1. Take the first example for instance, given the target region of *bus*, Transformer's attention is constantly swaying between the bus and surrounding environment, and finally is related to the bus with the passengers around, which, however, is inconsistent with the question. In contrast, from these examples, we can see that TRAR can dynamically select the attention span based on the output of the previous step, which is consistent with its target. Meanwhile, such an attention routing scheme can also improve the error-tolerance of self-attention modeling. From the first example of the second sub-figure, it can be seen that TRAR can also easily attend to incorrect regions.

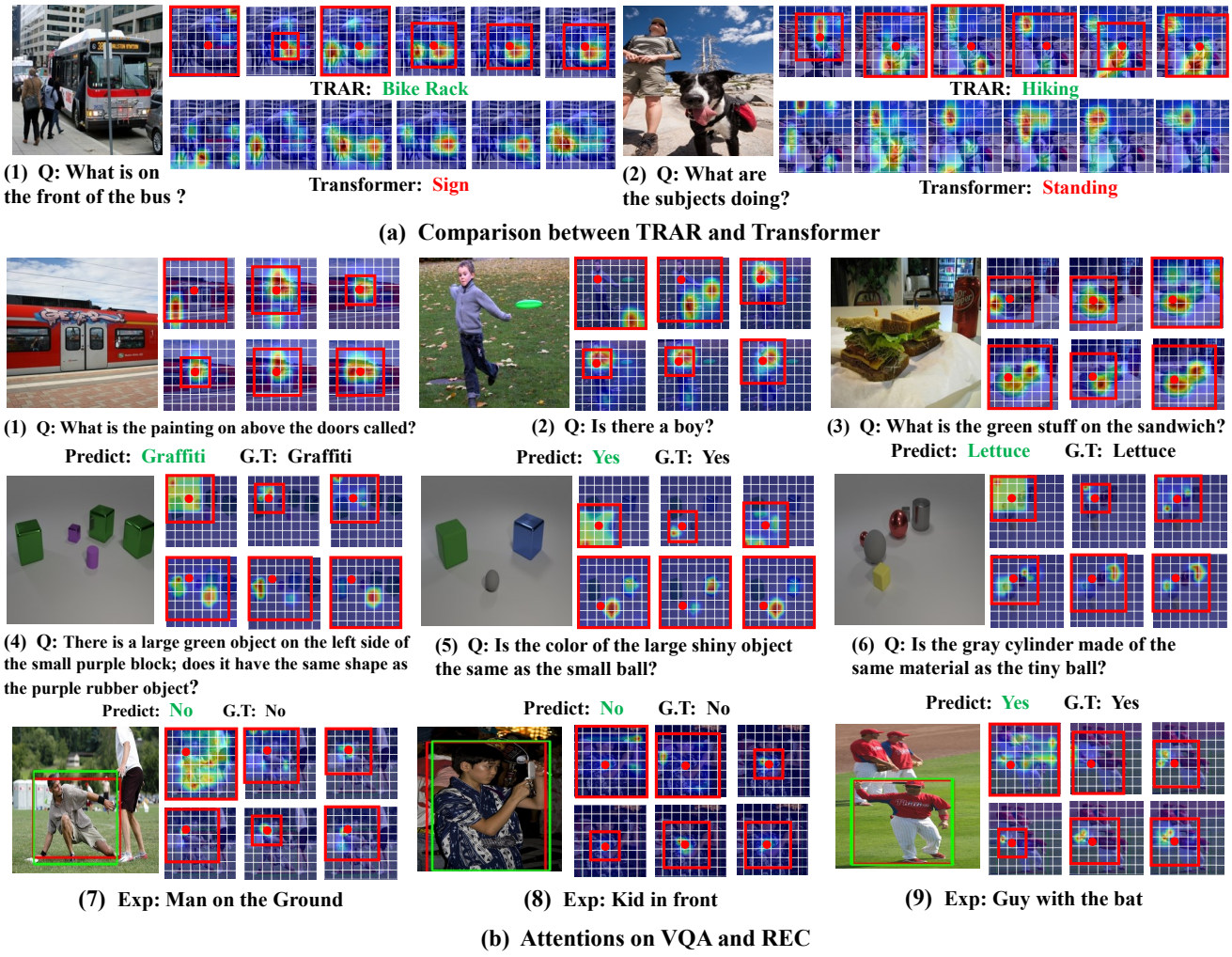


Figure 5: Visualizations of the attentions in TRAR. An area enclosed by a red box denotes the attention span of the chosen grid, *i.e.,* the one with red dot. TRAR can help the model to use different spans of attentions to schedule the global and local dependency modeling.

However, as the path routing progresses, its attention can be adjusted to the correct area, helping the model to answer the question. These appealing properties are also confirmed in the examples of REC.

5. Conclusion

In this paper, we investigates the dependency modeling of Transformer for two vision and language tasks, namely VQA and REC. These two tasks typically require the visual attentions from different receptive fields, which cannot be fully handled by the standard Transformer. To this end, we propose a lightweight and effective routing scheme called *Transformer Routing* (TRAR) to help the model dynamically select the attention spans for each example. In particular, TRAR transforms the module selection problem into the one of selecting attention masks, thereby making the additional computation and memory overhead negligible. To validate TRAR, we conduct extensive experiments on 5 benchmark datasets, and the experimental results greatly confirm the merits of TRAR.

Acknowledgement This work is supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No.U1705262, No. 62072386, No. 62072387, No. 62072389, No.62002305, No.61772443, No.61802324 and No.61702136), China Postdoctoral Science Foundation (2021T40397), Guangdong Basic and Applied Basic Research Foundation (No.2019B1515120049) and the Fundamental Research Funds for the central universities (No. 20720200077, No. 20720200090 and No. 20720200091).

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. *CVPR*, 2018.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Int. Conf. Comput. Vis.*, 2015.

[4] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

[5] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pages 527–536. PMLR, 2017.

[6] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. *international conference on computer vision*, pages 824–832, 2017.

[7] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[8] 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.

[9] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

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

[11] Jacob Devlin, Mingwei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, pages 4171–4186, 2019.

[12] Olivier Duchenne, Francis Bach, In-So Kweon, and Jean Ponce. A tensor-based algorithm for high-order graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2383–2395, 2011.

[13] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[14] Yash Goyal, Tejas Khot, Douglas Summersstay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[15] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *arXiv preprint arXiv:2102.04906*, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[17] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[18] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[19] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. *computer vision and pattern recognition*, pages 4418–4427, 2017.

[20] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. *computer vision and pattern recognition*, pages 4555–4564, 2016.

[21] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[22] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[23] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *ICLR*, 2018.

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[25] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020.

[26] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[27] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[28] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Feifei, C Lawrence Zitnick, and Ross B Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[29] Kushal Kafle and Christopher Kanan. An analysis of visual question answering algorithms. *ICCV*, 2017.

[30] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L Berg. Referitgame: Referring to objects in photographs of natural scenes. pages 787–798, 2014.

[31] Jinhwa Kim, Jaehyun Jun, and Byoungtak Zhang. Bilinear attention networks. *NIPS*, 2018.

[32] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[33] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. *Int. Conf. Comput. Vis.*, 2019.

[34] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8553–8562, 2020.

[35] Ji Lin, Yongming Rao, Jiwen Lu, and Jie Zhou. Runtime neural pruning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2178–2188, 2017.

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014.

[37] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 2017.

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[39] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[40] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. 2020.

[41] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10034–10043, 2020.

[42] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*, 2018.

[43] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Maria Camburu, Alan L Yuille, and Kevin P Murphy. Generation and comprehension of unambiguous object descriptions. *computer vision and pattern recognition*, pages 11–20, 2016.

[44] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4942–4950, 2018.

[45] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[46] Gao Peng, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven C H Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra- and inter- modality attention flow for visual question answering. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[47] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[48] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[49] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. *international conference on computer vision*, pages 1946–1955, 2017.

[50] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, pages 68–80, 2019.

[51] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[52] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *NeurIPS*, 2015.

[53] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8376–8384, 2019.

[54] Robik Shrestha, Kushal Kafle, and Christopher Kanan. Answer them all! toward universal visual question answering models. *CVPR*, 2019.

[55] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[56] Joseph Suarez, Justin Johnson, and Fei-Fei Li. Ddrprog: A clevr differentiable dynamic reasoning programmer. *arXiv preprint arXiv:1803.11361*, 2018.

[57] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335, 2019.

[58] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[59] Damien Teney, Peter Anderson, Xiaodong He, and Anton Van Den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *CVPR*, 2018.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 2017.

[61] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018.

[62] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[63] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2369–2378, 2020.

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

[65] 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*, 2019.

[66] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

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

[68] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. *european conference on computer vision*, pages 69–85, 2016.

[69] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. *CVPR*, pages 3521–3529, 2017.

[70] Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. Deep multimodal neural architecture search. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3743–3752, 2020.

[71] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[72] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Int. Conf. Comput. Vis.*, 2017.

[73] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. *international joint conference on artificial intelligence*, pages 1114–1120, 2018.

[74] Biao Zhang, Deyi Xiong, and Jinsong Su. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, 2018.

[75] Hanwang Zhang, Yulei Niu, and Shihfu Chang. Grounding referring expressions in images by variational context. *computer vision and pattern recognition*, pages 4158–4166, 2018.

[76] Yiyi Zhou, Rongrong Ji, Gen Luo, Xiaoshuai Sun, Jinsong Su, Xinghao Ding, Chia-Wen Lin, and Qi Tian. A real-time global inference network for one-stage referring expression comprehension. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[77] Yiyi Zhou, Rongrong Ji, Jinsong Su, Xiaoshuai Sun, and Weiqiu Chen. Dynamic capsule attention for visual question answering. *AAAI*, 33(1):9324–9331, 2019.

[78] Yiyi Zhou, Rongrong Ji, Jinsong Su, Xiaoshuai Sun, and Xiangming Li. Free vqa models from knowledge inertia by pairwise inconformity learning. *AAAI*, 2019.

[79] Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Gen Luo, Xiaopeng Hong, Jinsong Su, Xinghao Ding, and Ling Shao. K-armed bandit based multi-modal network architecture search for visual question answering. In *ACM Int. Conf. Multimedia*, pages 1245–1254, 2020.

[80] Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jinsong Su, Deyu Meng, Yue Gao, and Chunhua Shen. Plenty is plague: Fine-grained learning for visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[81] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian D Reid, and Anton Van Den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. *computer vision and pattern recognition*, pages 4252–4261, 2018.