Chop Chop BERT: Visual Question Answering by Chopping VisualBERT's Heads

Chenyu Gao[1,2,3] , **Qi Zhu**[1] , **Peng Wang**[1,3] * and **Qi Wu**[4]
[1]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]School of Software, Northwestern Polytechnical University, Xi'an, China
[3]National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, China
[4]University of Adelaide, Australia
{chenyugao, zhu_qi_happy_}@mail.nwpu.edu.cn, peng.wang@nwpu.edu.cn, qi.wu01@adelaide.edu.au

Abstract

Vision-and-Language (VL) pre-training has shown great potential on many related downstream tasks, such as Visual Question Answering (VQA), one of the most popular problems in the VL field. All of these pre-trained models (such as VisualBERT, ViLBERT, LXMERT and UNITER) are built with Transformer, which extends the classical attention mechanism to multiple layers and heads. To investigate why and how these models work on VQA so well, in this paper we explore the roles of individual heads and layers in Transformer models when handling 12 different types of questions. Specifically, we manually remove (chop) heads (or layers) from a pre-trained VisualBERT model at a time, and test it on different levels of questions to record its performance. As shown in the interesting echelon shape of the result matrices, experiments reveal different heads and layers are responsible for different question types, with higher-level layers activated by higher-level visual reasoning questions. Based on this observation, we design a dynamic chopping module that can automatically remove heads and layers of the VisualBERT at an instance level when dealing with different questions. Our dynamic chopping module can effectively reduce the parameters of the original model by 50%, while only damaging the accuracy by less than 1% on the VQA task.

1 Introduction

Transformer [Vaswani *et al.*, 2017] architecture was designed for translation tasks but has shown good performance across many other tasks, ranging from text summarization [Egonmwan and Chali, 2019; Xu *et al.*, 2020], language modeling [Dai *et al.*, 2019; Ma *et al.*, 2019] to question answering [Shao *et al.*, 2019; Sanh *et al.*,]. The BERT (Bidirectional Encoder Representations from Transformers) [Devlin *et al.*, 2018] model is then proposed for general pretraining. By fine-tuning the pre-trained model on downstream tasks, BERT further advances the state-of-the-art for multiple NLP tasks. Inspired by it, many variants have been proposed towards the solution of vision-and-language tasks, *e.g.,* Visual Question Answering (VQA [Kafle and Kanan, 2017b]), including VisualBERT [Li *et al.*, 2020], VL-BERT [Su *et al.*, 2020], ViLBERT [Lu *et al.*, 2019], LXMERT [Tan and Bansal, 2019], and UNITER [Chen *et al.*, 2020]*, etc.*

These vision-language BERT models are all Transformer-based, where the most innovative part is the multi-layer and multi-head settings, compared to the classic attention. However, how the heads and layers contribute to those reasoning-required tasks such as Visual Question Answering, remains a mystery. Here we want to explore the role of heads and layers and focus on the VQA task, hoping to find their patterns that human can 'see' and 'understand'. When faced with high-level tasks, there are different areas activated in our brain to fulfill different tasks. Our findings reveal that different heads and layers are indeed activated by different question types.

For the first time, we come up with a statistical analysis of Transformer's heads and layers triggered off by 12 types of questions on the Task Driven Image Understanding Challenge (TDIUC) [Kafle and Kanan, 2017a]. The TDIUC dataset is a large VQA dataset with 12 more fine-grained categories proposed to compensate for the bias in distribution of different question types of VQA 2.0 [Goyal *et al.*, 2017], which provide convenience for our analysis. Our experiments based on the VisualBERT, as for it's general Transformer style architecture without more extra designs. The experiments can be divided into two parts: Manual Chopping and Dynamic Chopping. For manual chopping in Section 4.2, we explored the effect of a particular head by removing it, or removing all the other heads in a layer but that one. As the most obvious pattern is found at a layer level, we further experiment by chopping one layer at a time. As there are 12 kinds of questions in TDIUC and 12 layers in VisualBERT, we get a result matrix with the size 12×12. The echelon shape in this matrix reveal different layers play different roles across 12 types of question, with higher-level reasoning questions demanding higher-level Transformer layers gradually. For Dynamic Chopping, we chop the layers simultaneously (Section 4.3) under the control of instance-conditioned layer scores, which are generated by a dynamic chopping module. We further

[*] Peng Wang is corresponding author. This work is supported by the National Key RD Program of China (No. 2020AAA0106900), and the National Natural Science Foundation of China (No.61876152, No.U19B2037).

find that removing some layers can decrease parameters but increase the accuracy of models.

Overall, the major contribution of this work is to delve into the roles of Transformer heads and layers in different types of visual demanding questions for the first time. We hope our interesting finding, especially the presented echelon shape of experiment results matrices, which shows a clear tendency of Transformer to gradually rely on higher-level layers when reaching higher-level reasoning questions, can inspire further investigation of the inner mechanism of self-attention layers.

2 Related Work & Background

2.1 Transformer in VQA

Transformer-based models, such as VisualBERT [Li *et al.*, 2020] and ViLBERT [Lu *et al.*, 2019], have gained popularity most recently, as they are well suited to pre-training and can be easily transferred to other similar tasks. VisualBERT consists of a stack of Transformer layers that implicitly align elements of an input text and regions in an associated image with self-attention. ViLBERT proposes to learn joint representation of images and text using a BERT-like architecture with separate Transformers for vision and language that can attend to each other. LXMERT [Tan and Bansal, 2019] is a large-scale Transformer model that consists of three encoders: an object relationship encoder, a language encoder, and a cross-modality encoder. UNITER [Chen *et al.*, 2020] adopts Transformer to leverage its self-attention mechanism designed for learning contextualized representations.

2.2 Multi-head & Multi-layer Transformer

Transformer [Vaswani *et al.*, 2017] architecture relies entirely on the self-attention mechanism. The self-attention **Attn** is calculated with a set of queries $\{q_i\}_{i=1}^N$, a set of keys $\{k_i\}_{i=1}^N$ and a set of values $\{v_i\}_{i=1}^N$:

$$\begin{aligned}\mathbf{attn_i} &= \sum_{j=1}^{N} \mathbf{softmax}(q_i k_j/\sqrt{d})v_j, \\ \mathbf{Attn} &= [attn_1, attn_2, ..., attn_N]^T,\end{aligned} \tag{1}$$

where N is the number of key-value pairs, d is the dimension of keys and values, and $1/\sqrt{d}$ is a scaling factor.

With a total of H heads, multi-head attention can be regarded as an ensemble of each head h's self-attention:

$$\mathbf{MH\_Attn} = \mathbf{Concat}(\mathbf{Attn}_1, ..., \mathbf{Attn}_h, ..., \mathbf{Attn}_H), \tag{2}$$

where $H = 12$ in our paper.

A Transformer encoder contains a total of L layers, with each one including two sub-layers: a multi-head self-attention layer and a fully connected feed forward layer. All of the queries and key-value pairs in a same self-attention layer come from the output of the previous layer in the encoder. Therefore, each position in a layer of encoder can attend to all the positions in the previous layer. In our paper, we set $L = 12$.

For each layer, there is a multi-head attention layer. The multi-head attention in layer l can be written as:

$$\mathbf{MH\_Attn}^l = \mathbf{Concat}(\mathbf{Attn}_1^l, ..., \mathbf{Attn}_h^l, ..., \mathbf{Attn}_H^l), \tag{3}$$

where l ranges from 1 to L.

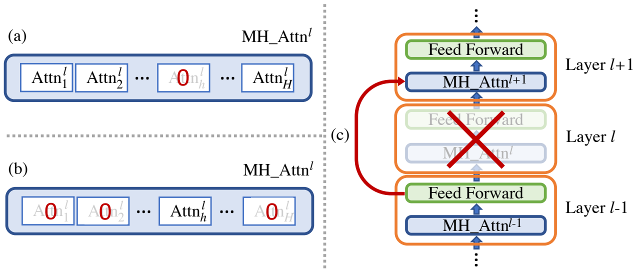

Figure 1: (a) to remove one head h in layer l, the self-attention weights $\mathbf{Attn}_h^l$ are multiplied with zero-valued binary mask. (b) to study whether one head h in layer l can replace the whole layer, all of the self-attention weights in layer l are multiplied with zero-valued binary mask except $\mathbf{Attn}_h^l$. (c) A part of Transformer encoder layers. After chopping Layer l, the output of Layer $l-1$ becomes the input of Layer $l+1$.

3 Chopping Method

We perform chopping upon our base model – VisualBERT, which is a general Transformer style architecture without extra designs. Viewing one head as an element, we get an $L \times H = 12 \times 12$ grid-like structure for the model. The purpose of chopping is to disable a part of the model and make it unable to participate in the forward process. For heads and layers, we have separate strategies to reach this end, as they have different characteristics in cooperating. Heads work in parallel while layers work in a sequential manner.

3.1 Manual Chopping

In manual chopping, whether to chop a component or not is controlled by our signal manually. The signals are given at two levels: heads and layers.

Manual Chopping of Heads

The importance of each head is investigated thoroughly. Every time, We simply modified the multi-head attention in a specific layer l by multiplying one head attention by a binary mask variable α_h^l $(h = 1, 2, ..., H)$:

$$\mathbf{MH\_Attn}^l = \mathbf{Concat}(\alpha_1^l\mathbf{Attn}_1^l, ..., \alpha_2^l\mathbf{Attn}_h^l, ..., \alpha_H^l\mathbf{Attn}_H^l), \tag{4}$$

where α_h^l can be either 0 or 1. The Transformer model we used have 12 layers with 12 heads in each layer, there are totally 144 α_h^l. For each time of removing one head, only one α is set to 0 and all the others are set to 1 (in Figure 1 (a)). As for removing all heads but one, only one α is set to 1 and all the others are set to 0 in a specific layer (in Figure 1 (b)). To study the contribution of each head, we exhaust each α_h^l in the above two experiments.

Manual Chopping of Layers

As the layers are in a serial manner, to chop off one specific Layer l, we directly skip it and connect the output of Layer $l-1$ to the input of Layer $l+1$. The detail is shown in Figure 1 (c). We chop each layer in the Transformer one time.

#	Question types	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
1	Absurd	−0.18%	−0.75%	−0.56%	**-5.73%**	0.23%	−0.13%	0.05%	0.02%	−0.03%	0.04%	−0.05%	−0.03%
2	Obj Pres	−0.12%	−0.79%	−0.05%	−0.33%	−0.18%	0.08%	0.01%	0.02%	0.03%	0.03%	0.02%	−0.02%
3	Scene Rec	−0.22%	0.19%	−0.36%	−0.19%	−0.19%	−0.12%	0.10%	−0.06%	0.11%	−0.07%	0.04%	−0.10%
4	Sub-Obj Rec	0.07%	−0.61%	−0.16%	−0.30%	−0.39%	0.06%	−0.05%	−0.05%	0.06%	0.05%	0.06%	0.09%
5	Sport Rec	−0.13%	−0.21%	0.09%	−0.36%	−0.20%	−0.03%	−0.10%	−0.06%	−0.06%	−0.09%	−0.05%	−0.05%
6	Color Attr	−0.25%	**-3.09%**	-0.40%	**-1.26%**	−0.60%	−0.22%	−0.18%	−0.06%	0.06%	−0.07%	−0.07%	−0.13%
7	Sentiment	−0.48%	−0.73%	**-4.36%**	−0.97%	0.24%	**-1.45%**	0.24%	0.24%	0.24%	0.24%	0.24%	0.24%
8	Count	**-4.96%**	**-2.03%**	**-2.45%**	**-7.09%**	**-1.08%**	−0.34%	−0.28%	−0.27%	−0.41%	−0.25%	−0.13%	−0.08%
9	Positional Rec	0.33%	**-1.87%**	−0.54%	**-1.78%**	**-1.54%**	−0.57%	0.45%	0.57%	0.54%	0.63%	0.60%	**1.03%**
10	Other Attr	0.42%	**-3.58%**	0.63%	**-1.52%**	**-1.46%**	**-2.54%**	0.42%	0.17%	−0.66%	0.30%	0.36%	−0.36%
11	Activity Rec	**-1.12%**	**-3.13%**	0.88%	**-4.49%**	**1.04%**	**-5.46%**	**1.52%**	**2.01%**	**-1.04%**	0.56%	−0.80%	**-1.85%**
12	Util & Aff	**-3.85%**	**-11.54%**	**1.92%**	**-5.77%**	**-9.62%**	**-5.77%**	**-5.77%**	**-5.77%**	**-1.92%**	**-1.92%**	**-1.92%**	**1.92%**

Table 1: The relative difference of accuracy on each question type when only one head is removed, which is calculated by $(Acc_{new} - Acc_{org})/Acc_{org}$. In each layer, we only show the one with the **largest absolute value** out of 12 heads. For the complete result of all 12 heads, please see Appendix A. We conduct this head removal on TDIUC's 12 question types. Underlined numbers indicate that its absolute value ($|(Acc_{new} - Acc_{org})/Acc_{org}|$) is above 1%.

#	Question types	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
1	Absurd	0.19%	0.60%	0.25%	−0.54%	0.88%	−0.01%	0.26%	0.09%	0.07%	0.17%	0.06%	0.07%
2	Obj Pres	−0.51%	−0.49%	−0.21%	**-1.14%**	0.06%	0.10%	0.03%	0.10%	0.12%	0.08%	0.00%	−0.02%
3	Scene Rec	−0.17%	0.09%	−0.17%	−0.38%	−0.14%	0.03%	0.27%	0.02%	0.14%	0.19%	0.19%	0.25%
4	Sub-Obj Rec	−0.10%	−0.34%	−0.49%	−0.39%	**-1.11%**	0.06%	−0.17%	0.00%	−0.12%	0.12%	0.06%	0.10%
5	Sport Rec	−0.35%	−0.30%	0.00%	−0.52%	−0.61%	−0.27%	−0.34%	0.01%	−0.02%	−0.12%	0.02%	0.07%
6	Color Attr	**-1.32%**	**-1.64%**	**-2.24%**	**-1.64%**	**-5.68%**	−0.68%	−0.61%	−0.15%	−0.01%	−0.20%	−0.05%	−0.09%
7	Sentiment	0.73%	0.48%	0.24%	0.00%	0.48%	**1.21%**	0.00%	0.97%	0.97%	0.97%	0.24%	0.24%
8	Count	**-1.97%**	**-3.78%**	**-1.53%**	**-4.17%**	**-5.86%**	**-1.02%**	−0.66%	−0.40%	−0.59%	−0.30%	0.20%	−0.08%
9	Positional Rec	−0.99%	**-2.89%**	**-3.65%**	**-8.65%**	**-7.75%**	**-1.09%**	**-1.15%**	−0.27%	−0.03%	0.51%	0.93%	−0.81%
10	Other Attr	−0.49%	−0.97%	**-1.33%**	0.32%	**-4.68%**	−0.89%	0.21%	−0.30%	−0.99%	−0.40%	0.21%	−0.13%
11	Activity Rec	−0.08%	0.08%	−0.32%	−0.24%	**-2.65%**	**-2.25%**	**-2.41%**	**1.12%**	**-3.13%**	**-1.12%**	−0.40%	−0.56%
12	Util & Aff	0.00%	**-9.62%**	0.00%	**-1.92%**	**-15.38%**	**-5.77%**	**-5.77%**	**-3.85%**	0.00%	0.00%	0.00%	0.00%

Table 2: The relative difference of accuracy on each question type when only one head is kept in each layer, which is calculated by $(Acc_{new} - Acc_{org})/Acc_{org}$. In each layer, we only show the one with the **largest value** out of 12 heads. For the result of all 12 heads, please see Appendix B. Underlined numbers indicate that its absolute value is above 1%.

3.2 Automatic Dynamic Chopping

In contrast to manual chopping, we hope the model can automatically learn how important and relevant a specific layer is. Therefore, we design a dynamic chopping module to explore the effect of each layer l by learning a score S_l for each layer, then remove layers with scores below a given threshold. An instance level layer score is calculated by:

$$\begin{aligned} [\mathbf{q}_1^l, ..., \mathbf{q}_h^l, ..., \mathbf{q}_H^l] &= \mathbf{W}(hidden\_state^l[0]), \\ S_l &= \mathbf{Sigmoid}(\frac{\sum_{i=1}^{H} \mathbf{q}_i^l}{H}), \end{aligned} \tag{5}$$

where $\mathbf{W}$ is a linear layer, $hidden\_state^l$ is the hidden_state output by layer l and we only take out the first element of this feature, which is also called **[CLS]** feature. 12 head scores $\mathbf{q}_h^l$ are averaged and put into a $\mathbf{Sigmoid}$ function to yield a value between 0 and 1, measuring the degree of importance for a specific layer. The score matrices will change when dealing with different instances.

Then we use this layer score S_l to determine whether to chop a layer or not. If S_l is below a threshold, then the Layer l will be chopped at the forward time. To chop the Layer l, we directly skip it and connect the output of Layer $l - 1$ to the input of the Layer $l + 1$.

The dynamic chopping module is trained independently without modifying any parameters in the Transformer model. The maximal learning rate is $1e - 3$ and the batch size is 480. This module is trained with a binary cross-entropy loss (for answer prediction) and an additional loss ($L1$ norm of head scores).

4 Experiments

4.1 Experimental Settings

We thoroughly study the effect of Transformer's heads and layers on the VQA task when questions are in different types. All of our experiments are conducted on the Task Driven Image Understanding Challenge (TDIUC) [Kafle and Kanan, 2017a] dataset, a large VQA dataset. This dataset was proposed to compensate for the bias in distribution of different question types of VQA 2.0 [Goyal *et al.*, 2017]. It is larger than VQA 2.0 and divides questions into 12 more fine-grained categories based on the task they solve, which provide us convenience to do analysis by question types.

TDIUC has 12 question types, including both classical computer vision tasks and novel high-level vision tasks which

Layer Q types	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5	Layer-6	Layer-7	Layer-8	Layer-9	Layer-10	Layer-11	Layer-12	Simple Acc
Absurd	-0.96%	-1.14%	-5.17%	-41.89%	1.19%	0.08%	0.24%	0.09%	0.11%	0.33%	0.14%	-0.01%	98.06%
Obj Pres	-2.58%	-7.02%	-1.25%	-9.98%	-3.30%	0.00%	0.01%	0.07%	0.08%	-0.06%	0.03%	-0.09%	95.18%
Scene Rec	-2.47%	-2.41%	-2.15%	-2.53%	-2.22%	-0.23%	0.19%	-0.16%	-0.37%	-0.28%	0.07%	0.00%	93.15%
Sub-Obj Rec	-1.32%	-2.75%	-2.09%	-3.82%	-3.65%	-0.58%	-0.51%	-0.24%	-0.38%	-0.12%	-0.11%	-0.05%	86.59%
Sport Rec	-0.53%	-1.03%	-0.32%	-5.88%	-3.36%	-0.92%	-1.28%	-0.14%	-0.35%	-0.41%	-0.37%	0.02%	95.12%
Color Attr	-9.78%	-10.83%	-5.40%	-7.86%	-14.53%	-1.47%	-1.00%	-0.25%	-0.32%	-0.48%	-0.38%	-0.57%	72.66%
Sentiment	-0.48%	0.24%	-4.60%	-0.73%	-3.15%	-1.21%	0.24%	0.00%	0.00%	-1.69%	0.00%	-1.69%	65.14%
Count	-7.57%	-7.12%	-9.96%	-29.04%	-11.33%	-2.11%	-1.42%	-0.55%	-1.04%	-1.00%	-0.25%	-0.63%	55.49%
Positional Rea	-8.98%	-7.87%	-8.89%	-16.88%	-15.04%	-5.37%	-4.34%	-2.89%	-2.53%	-0.78%	1.42%	-0.51%	27.00%
Other Attr	-8.64%	-8.72%	-1.42%	-3.68%	-13.63%	-22.87%	-1.44%	-1.48%	-5.61%	-4.08%	-1.50%	-1.04%	51.37%
Activity Rec	-1.85%	-4.65%	-2.25%	-7.70%	-20.39%	-13.80%	-15.17%	-0.96%	-16.45%	-2.65%	-0.16%	-3.69%	46.46%
Util & Aff	-3.85%	-19.23%	-13.46%	-23.08%	-40.38%	-7.69%	-15.38%	-11.54%	-3.85%	-11.54%	-9.62%	-5.77%	30.41%

Figure 2: The relative difference of accuracy on each type when one layer (all 12 heads) is removed, which is calculated by $(Acc_{new} - Acc_{org})/Acc_{org}$. The matrix shows an interesting echelon form, which reveals the tendency of Transformer to rely on higher-level layers for higher-level questions.

require varying degrees of image understanding and reasoning. The total number of unique answers is $1,618$, which is then set as the size of our answer vocabulary. Experiments are conducted on 4 NVIDIA GeForce 2080Ti GPUs with a batch size of 480. We choose a general and popular pretraining model VisualBERT [Li *et al.*, 2020], which is built on top of PyTorch. This model consists of visual embeddings & text embeddings, a one-stream Transformer and a classifier specific for VQA tasks. There are $L = 12$ layers of encoder, and each layer has $H = 12$ heads. The hidden state dimension is 768. We load the model pre-trained on COCO Caption [Chen *et al.*, 2015] dataset, then finetune it with a leaning rate of $5e-5$ on the TDIUC dataset. The image features are extracted from a ResNet-101 [He *et al.*, 2016] based Faster R-CNN [Ren *et al.*, 2015] model pre-trained on Visual Genome [Krishna *et al.*, 2017].

4.2 Effect of Manual Chopping

Removing One Head at One Time

First, we follow the previous work [Michel *et al.*, 2019] to study if there is a specific role for a particular attention head h in different question types. To study the contribution of a particular head, we mask a single head (*i.e.,* multiplying $\mathbf{Att}_h^l$ with zero-valued binary mask) in Transformer at a time, then evaluate the model's performance. If one head is important in a certain question type, the accuracy after the removal of this head will decrease dramatically. On the other hand, the increase in accuracy after removal can tell that this head is insignificant or even plays a negative role in this question type.

In Table 1, we show the relative difference of accuracy $((Acc_{new} - Acc_{org})/Acc_{org})$ on each type when only one head is removed. In each layer, there are 12 heads and we show the one who has the **largest absolute value** of relative difference to get a clue about the maximal influence of one layer. In most of the question types, the largest absolute value of relative difference out of 12 heads is still a small number, meaning some heads in this layer can be removed to get a more efficient model. In question types such as "`Activity Recognition`" and "`Utilities & Affordances`", removal of a specific head in more layers can bring a dramatic decrease in accuracy, which can illustrate that more layers (both low-level and high-level) are needed here. Observing the percentage of change in accuracy for each head (can be found in Appendix A), we found that in the majority of question types, removing one head does not hurt the accuracy a lot. Moreover, removal of some heads can even bring the improvement of accuracy, which means heads are not equally important in answering a specific question type.

Removing All Heads but One

It is also a question whether more than one head h is needed in a specific layer. We mask all heads in a layer except for a single one and show the relative difference of accuracy on each question type in Table 2. In each layer, we only show the **largest value** of relative difference out of 12 heads $(\mathbf{max}((Acc_{new}^i - Acc_{org}^i)/Acc_{org}^i), i \in [1, 12])$. If there exists one head that can replace or even surpass all heads in a layer, this largest value will be a positive value. For some layers in a specific question type, one head is indeed sufficient at test time.

On the other hand, in some layers even the largest value of relative difference is a negative number, which means the high accuracy comes from the cooperative work of multiple heads. In question types `Color Attributes` and `Activity Recognition`, nearly none of a single head can replace a whole layer in all 12 layers. Besides, observing the relative difference of accuracy for each head (in Appendix B), we find that most of the heads in one layer tend to have the same level of influence. To further investigate whether there are some specific roles for different layers, we conduct experiments by removing a whole layer at a time below.

Removing One Layer

To investigate the effect of layers, we remove one layer in Transformer at a time. The dramatically decrease of accuracy illustrates that a layer plays a vital role in a question type. The more the accuracy decrease, the more important a layer is. While a positive relative difference tells that a layer is useless and can be removed. From the results in Figure 2, removing most of the layers cause a more or less decrease of accuracy, while other layers bring a slight increase of accuracy.

Results show that lower-level layers are enough for simple questions, such as questions belong to `Absurd`,

#	Question types	NMN [Andreas *et al.*, 2016]	RUA [Noh and Han, 2016]	QTA [Shi *et al.*, 2018]	VisualBERT (random 50%)	VisualBERT (>0.05)	VisualBERT (>0.1)	VisualBERT (>0.3)	VisualBERT (>0.5)	VisualBERT (>0.7)	VisualBERT (full)
1	Absurd	87.51	96.08	100.0	4.02	98.07	98.07	98.10	**98.21**	84.40	98.06
2	Obj Pres	92.50	94.38	94.55	52.01	95.19	95.25	**95.32**	94.52	72.46	95.20
3	Scene Rec	91.88	93.96	93.80	34.68	93.15	93.18	**93.36**	93.05	73.73	93.15
4	Sub-Obj Rec	82.02	86.11	86.98	4.07	86.59	**86.62**	86.61	86.38	70.14	86.59
5	Sport Rec	89.99	93.47	95.55	14.91	95.12	**95.13**	**95.13**	94.22	67.66	95.12
6	Color Attr	54.91	66.68	60.16	21.83	72.62	**72.66**	72.61	71.05	64.20	**72.66**
7	Sentiment	58.04	60.09	64.38	39.27	65.14	64.98	**65.46**	64.67	50.16	65.14
8	Count	49.21	48.43	53.25	20.52	55.40	55.39	55.46	53.82	42.05	**55.49**
9	Positional Rec	27.92	35.26	34.71	0.90	27.02	**27.04**	26.95	26.03	19.36	27.00
10	Other Attr	47.66	56.49	54.36	1.59	**51.37**	**51.37**	51.28	50.38	47.13	**51.37**
11	Activity Rec	44.26	51.60	60.10	0.00	46.56	46.46	**46.61**	43.73	25.09	46.46
12	Util & Aff	25.15	31.58	31.48	10.53	30.41	**30.99**	**30.99**	27.49	22.22	30.41
13	A-MPT	62.59	67.81	69.11	42.54	68.04	68.10	**68.16**	66.96	53.22	68.05
14	H-MPT	51.87	59.00	60.08	33.61	56.72	56.89	**56.91**	54.71	42.07	56.73
15	Simple Accuracy	79.56	84.26	87.52	28.25	86.13	86.16	**86.21**	85.47	69.10	86.51

Table 3: The accuracy on TDIUC dataset after removing Transformer layers whose scores are below the specific thresholds. The highest accuracy values on the right part of the table are in bold. When we remove layers whose scores are below 0.3, the accuracy, arithmetic mean-per-type (A-MPT) accuracy and harmonic mean-per-type accuracy (H-MPT) all surpass those of the full VisualBERT model, and over half of the question types also increase in accuracy.

Q types \ Layer	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5	Layer-6	Layer-7	Layer-8	Layer-9	Layer-10	Layer-11	Layer-12
Absurd	0.4465	0.8462	0.6381	0.9168	0.5291	0.6471	0.3905	0.0695	0.5387	0.1057	0.1468	0.0397
Obj Pres	0.4460	0.9102	0.5348	0.9586	0.8449	0.4393	0.0403	0.3445	0.5527	0.6726	0.0032	0.0013
Scene Rec	0.4472	0.8760	0.7806	0.9590	0.8332	0.5602	0.3896	0.3176	0.5801	0.6446	0.2239	0.2531
Sub-Obj Rec	0.4472	0.8682	0.6657	0.9258	0.8543	0.6747	0.8609	0.5088	0.7740	0.5695	0.4791	0.4243
Sport Rec	0.4471	0.8525	0.6401	0.8543	0.7044	0.6027	0.7553	0.4025	0.6119	0.5385	0.4243	0.4462
Color Attr	0.4472	0.8696	0.7778	0.9170	0.9462	0.8174	0.8186	0.2538	0.3080	0.7168	0.4778	0.6841
Sentiment	0.4471	0.8747	0.7471	0.9224	0.8334	0.6660	0.2612	0.5554	0.7857	0.7064	0.2001	0.1102
Count	0.4472	0.9002	0.7579	0.9433	0.9782	0.7943	0.5905	0.6691	0.6253	0.4441	0.6150	0.0308
Positional Rea	0.4471	0.8210	0.7154	0.9380	0.8929	0.8291	0.8983	0.7470	0.8068	0.7124	0.7919	0.6279
Other Attr	0.4471	0.8506	0.7614	0.9160	0.9081	0.8614	0.7901	0.6582	0.8794	0.7871	0.6045	0.4163
Activity Rec	0.4471	0.8467	0.6767	0.8462	0.7754	0.8269	0.8597	0.6269	0.8038	0.4993	0.5494	0.6383
Util & Aff	0.4471	0.8421	0.6468	0.9658	0.8579	0.7943	0.7886	0.6336	0.8766	0.7587	0.6405	0.5568

Figure 3: Instance level layer score (an average of all instances) in each question type learned by attention on **[CLS]** feature. The result matrix here is also in an echelon form, similar to that of Figure 2.

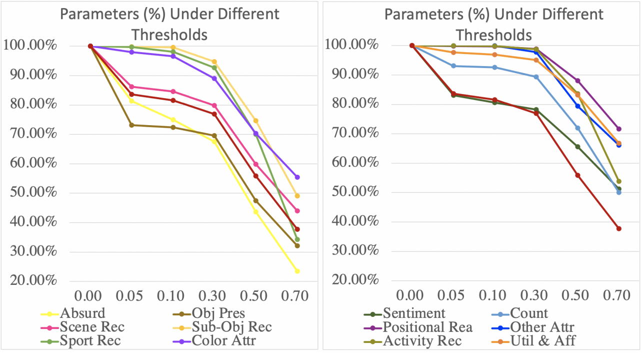

Figure 4: The chopping curve, showing percentage of Transformer parameters after chopping layers under the control of several different thresholds.

`Object Presence`, *etc.*; higher-level layers are important in questions where more reasoning abilities are needed such as questions belong to `Sentiment Understanding`, `Activity Recognition`, *etc.*The 12 × 12 result matrix in Figure 2 also shows an interesting echelon shape that gradually changes according the level of reasoning. Questions belonging to `Other Attributes` (*e.g.,* 'What shape is the clock?') and `Object Utilities and Affordances` (*e.g.,* 'What object can be used to break glass?') may ask the materials or shapes of objects, which needs strong reasoning ability and common sense knowledge to answer them; `Activity Recognition` (*e.g.,* 'What is the girl doing?') requires scene and activity understanding abilities; `Positional Reasoning` (*e.g.,* 'What is to the left of the man on the sofa?') and `Sentiment Understanding` (*e.g.,* 'How is she feeling?') are two kinds of well known complicated questions requiring multiple reasoning and understanding abilities. Removing high-level layers will dramatically hurt accuracy in the above-mentioned question types.

We also observe some other interesting phenomena. In question type `Scene Recognition` (*e.g.,* 'What room is this?') removal of the first 5 layers causes decrease of accuracy in roughly a same manner, reflecting that they are all important for scene understanding. In question type `Object Presence` (*e.g.,* 'Is there a cat in the image?') Layer 4 plays the most important role. Layer 4 is also important in other questions classes related to objects such as `Subordinate Object Recognition`, `Counting` (where the model needs to count the number of objects) and `Color Attributes` (locate the object firstly then recog-

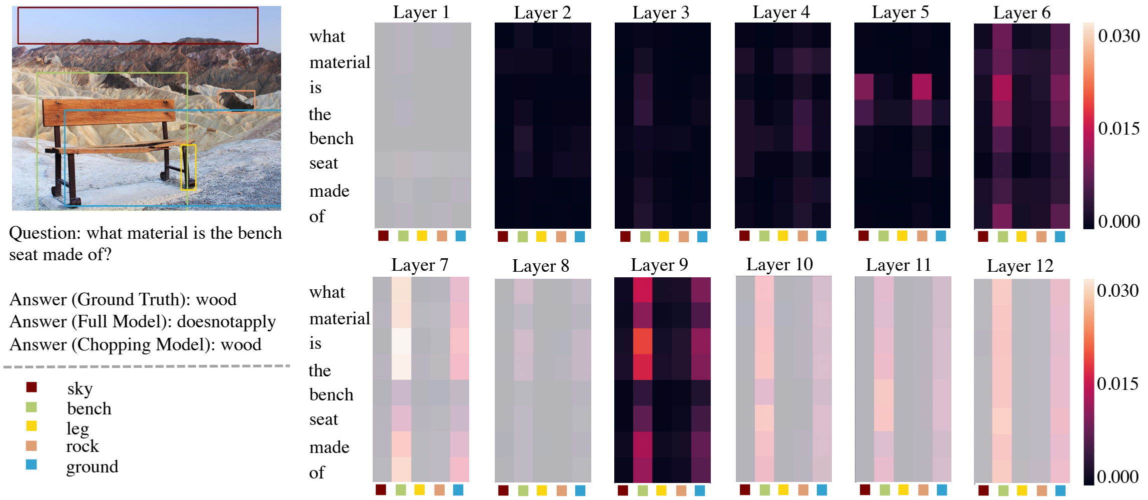

Figure 5: Attention weights of each layers in VisualBERT. Here we only show the attention weights matrices of the question and top-5 objects. The weights shown are the average of a layer (all 12 heads). Layers covered by semi-transparent rectangular are chopped under the threshold (0.5).

nize its color). The question type `Absurd` (*i.e.,* Nonsensical queries about the image) is the easiest class, and it can be answered easily by determining whether the object mentioned in the question appears in the image, which also depends on objects and the removal of Layer 4 impacts on the accuracy greatly. Layer 3 and Layer 5 play a critical role in question type `Counting`; Layer 2 and Layer 5 are important in question type `Color Attributes`, which illustrates that some layers cooperate to work out special reasoning problems.

4.3 Effect of Dynamic Chopping

In our previous section, we analyze the effect of 'removing one head', 'removing all heads but one within a single layer' and 'directly removing one layer'. Based on the observations, we consider a single layer as a basic module in Transformer and want to further explore what would happen if we chop two or more layers at the same time. In the dynamic chopping module, a score is learned for each layer and we set several thresholds to determine whether to remove a layer or not, then we record the change in accuracy.

The mean of layer scores is collected in each type and the results can be found in Figure 3. The distribution of layer scores is similar to that of relative difference in manually removing one layer (Figure 2). When removing a layer hurts the accuracy a lot, the dynamic chopping module tends to learn a higher score for this layer.

We set several thresholds to find the best trade-off between accuracy and model size. Layers whose score is lower than the threshold value are removed. All the results can be found in Table 3 without fine-tuning. The percentage of parameters under the control of different thresholds are shown in Figure 4 and the detailed value of parameters can be found in Appendix D. For lower-level questions, the chopping curve is steeper, meaning it is easier to chop more layers. However, for higher-level questions, the chopping curve is smoother, as the chopping of layers needs stricter thresholds.

With larger thresholds, fewer parameters are kept by the VisualBERT model. When we set threshold to (0.3), the model achieves higher accuracy on half of the question types with roughly 76.92% of parameters reserved. When we reserve layers whose score is larger than 0.5, only 55.85% of parameters are reserved, with only 1% loss of accuracy. When we set threshold to (0.7), more parameters will be removed, with a significant impact on performance. However, to remove 50% layers randomly yields a much worse performance. The results show that our dynamic chopping module successfully predict scores to control which layer can be chopped efficiently. We also observe that one layer cannot replace the whole Transformer (Appendix C), and the result shows leaving only one layer in almost all question types cause over 50% of accuracy drop. All the results illustrate that not all layers can be reduced without significantly influencing performance. A qualitative example is also shown in Figure 5. After chopping insignificant layers under the threshold (0.5), the surviving layers can generate the true answer. Please refer to Appendix E and Appendix F for more examples and the analysis about failure cases.

5 Conclusion

In this paper, we conduct extensive experiments on a Visual Question Answering dataset categorized by question types and explore the roles of heads and layers in a Vision-and-Language pre-trained Transformer by ablating these heads and layers separately. Interesting results as shown in an echelon shape reveal higher-level layers are required for higher-level questions. A dynamic chopping module is further designed to automatically learn efficiently chopped model. We hope our findings to advance the understanding of Transformer structure under different level of visual reasoning question types and inspire the investigation of more accurate and more efficient visual reasoning models.

References

[Andreas *et al.*, 2016] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. pages 39–48, 2016.

[Chen *et al.*, 2015] 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.

[Chen *et al.*, 2020] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Learning universal image-text representations. In *Proc. Eur. Conf. Comp. Vis.*, 2020.

[Dai *et al.*, 2019] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proc. Conf. Association for Computational Linguistics*, 2019.

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

[Egonmwan and Chali, 2019] Elozino Egonmwan and Yllias Chali. Transformer-based model for single documents neural summarization. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 70–79, 2019.

[Goyal *et al.*, 2017] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6904–6913, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016.

[Kafle and Kanan, 2017a] Kushal Kafle and Christopher Kanan. An analysis of visual question answering algorithms. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1965–1973, 2017.

[Kafle and Kanan, 2017b] Kushal Kafle and Christopher Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, 2017.

[Krishna *et al.*, 2017] 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. *Int. J. Comput. Vision*, 123(1):32–73, 2017.

[Li *et al.*, 2020] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. In *Proc. Conf. Association for Computational Linguistics*, 2020.

[Lu *et al.*, 2019] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 13–23, 2019.

[Ma *et al.*, 2019] Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. A tensorized transformer for language modeling. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 2232–2242, 2019.

[Michel *et al.*, 2019] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Proc. Advances in Neural Inf. Process. Syst.*, pages 14014–14024, 2019.

[Noh and Han, 2016] Hyeonwoo Noh and Bohyung Han. Training recurrent answering units with joint loss minimization for vqa. *arXiv preprint arXiv:1606.03647*, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 91–99, 2015.

[Sanh *et al.*,] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*.

[Shao *et al.*, 2019] Taihua Shao, Yupu Guo, Honghui Chen, and Zepeng Hao. Transformer-based neural network for answer selection in question answering. *IEEE Access*, 7:26146–26156, 2019.

[Shi *et al.*, 2018] Yang Shi, Tommaso Furlanello, Sheng Zha, and Animashree Anandkumar. Question type guided attention in visual question answering. In *Proc. Eur. Conf. Comp. Vis.*, pages 151–166, 2018.

[Su *et al.*, 2020] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *Proc. Int. Conf. Learn. Representations*, 2020.

[Tan and Bansal, 2019] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proc. Conf. Empirical Methods in Natural Language Processing*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 5998–6008, 2017.

[Xu *et al.*, 2020] Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text summarization. In *Proc. Conf. Association for Computational Linguistics*, pages 5021–5031, 2020.