Learning to Select Question-Relevant Relations for Visual Question Answering

Jaewoong Lee[1]*, **Heejoon Lee**[2]*, **Hwanhee Lee**[1] and **Kyomin Jung**[1]
[1]Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2]SK Hynix, Sungnam, Korea
{hello3196, wanted1007, kjung}@snu.ac.kr
{heejoon1.lee@sk.com}

Abstract

Previous existing visual question answering (VQA) systems commonly use graph neural networks(GNNs) to extract visual relationships such as semantic relations or spatial relations. However, studies that use GNNs typically ignore the importance of each relation and simply concatenate outputs from multiple relation encoders. In this paper, we propose a novel layer architecture that fuses multiple visual relations through an attention mechanism to address this issue. Specifically, we develop a model that uses question embedding and joint embedding of the encoders to obtain dynamic attention weights with regard to the type of questions. Using the learnable attention weights, the proposed model can efficiently use the necessary visual relation features for a given question. Experimental results on the VQA 2.0 dataset demonstrate that the proposed model outperforms existing graph attention network-based architectures. Additionally, we visualize the attention weight and show that the proposed model assigns a higher weight to relations that are more relevant to the question.

1 Introduction

VQA (visual question answering) is a task that aims to output an answer for a given question related to a given image. VQA is a multimodal task that requires an understanding of multiple modalities. Therefore, VQA has received much attention in both computer vision and natural language processing research.

Most related works on VQA focus on the problem of image understanding and various attention mechanisms to fuse textual and image inputs. For example, Bottom-up Top Down Attention (Anderson et al., 2018) uses the features from detection models instead of CNN outputs and demonstrates their effectiveness with VQA tasks. Additionally, variable attention networks such as a stacked attention network (Yang et al., 2016) show that an attention mechanism between visual and text modalities is necessary for solving VQA tasks to find the objects to be focused on to answer a given question.

Q: Is the man wearing a tie?

Implicit	Semantic	Spatial	Avg
no	yes	no	no

Q: How many riders are on the motorcycle?

Implicit	Semantic	Spatial	Avg
1	1	0	1

Figure 1: Two examples showing simple weighted sum results in wrong predictions. In both cases, although one relation encoder has given the correct answer, the final model's answer is incorrect, and averaging them result in wrong answer.

However, to solve higher-level VQA problems that require multi-hop reasoning, the model must consider various relations, such as geometric relationships between objects in the image. For this reason, researchers try to extract higher-level visual information using a graph neural network(GNN) based relation encoder to aggregate the relational information between the objects in an image.

For example, ReGAT (Li et al., 2019), an existing VQA architecture that uses GNNs, utilizes various relations between objects using graph attention networks (Veličković et al., 2018). Specifically, ReGAT uses three predefined relations: implicit, semantic, and spatial. To capture visual information, ReGAT constructs GNN-based relation encoders for each relation and combines the output probability distributions from the encoders using fixed weights to make the final prediction. However, this process can be problematic because the importance of each relationship for the given question cannot be considered. Figure 1 shows two examples where ReGAT does not make the correct prediction due to using fixed weights. In all cases, even though the

* Equal Contribution

Proceedings of the Third Workshop on Multimodal Artificial Intelligence, pages 87–96

correct answer is given by one of the relation encoders, ReGAT finally predicts the incorrect answer due to the incorrect answers in the other encoders. For example, in the first example in Figure 1, a semantic relationship is particularly important compared to the other relationships because the model must consider the relation defined as *wearing* between the man and the tie. And the prediction from this semantic relation encoder is correct. However, the other encoders, including implicit and spatial, outputs incorrect answers because these relations are less related to the given question. Therefore, while ReGAT uses various attention mechanisms on objects to obtain relation-aware features, simply using the average or weighted summation with fixed weights to combine the relation-aware features can lead to incorrect predictions when averaging is insufficient to smooth out the noise from the less important relation features.

To resolve this shortcoming of previous models, we propose a novel model that can dynamically select a proper graph representation by considering the input question. We use attention mechanisms to make full use of relation encoders by giving them question-adaptive weights. Specifically, we train all relation encoders concurrently and learn adaptive weights to form a combined joint representation. Using these attention weights, the proposed model assigns higher weights to the relations that are meaningful for a given question. Experimental results demonstrate that the proposed model outperforms the previous existing model. Our model has an accuracy of 64.27%, compared to the existing model with 62.65% in VQA v2.0 dataset. Additionally, the proposed attention module can be easily visualized and has a natural form of interpretability. Thus, we analyze examples through the visualization of attention weights and verify that our model properly assigns attention weights to the relevant relationships for the question. Our contributions can be summarized as follows:

- We propose a novel attention-based VQA model that can dynamically select an essential relation for the given question.
- Experimental results show that the proposed model with adaptive attention weights for each relation outperforms the existing model.
- We also visualize the attention weights given to each relation and show that the proposed model can properly assign higher weights to question-related relations.

2 Related Work

2.1 Visual Question Answering

Models that are designed to solve VQA (Antol et al., 2015) are typically composed of four parts: an image encoder, a question encoder, multimodal fusion, and an answer predictor. In many studies, such as (Yang et al., 2016; Fan and Zhou, 2018; Patro and Namboodiri, 2018; Lu et al., 2016; Teney et al., 2018; Nam et al., 2017; Zhu et al., 2017; Malinowski et al., 2018), CNN-based attention mechanisms are frequently used in image encoders, which use the attention mechanism with images to concentrate on useful objects based on the input questions. Conversely, (Lu et al., 2016; Nam et al., 2017; Fan and Zhou, 2018; Yang et al., 2020) also uses attention mechanisms in question encoders to produce image-adaptive question embeddings.

Many previous works on VQA (Yao et al., 2018; Kipf and Welling, 2016; Santoro et al., 2017; Hu et al., 2018; Cadene et al., 2019; Yang et al., 2018; Teney et al., 2017; Norcliffe-Brown et al., 2018; Wang et al., 2019) use graph attention networks to extract visual features from images. Graph attention networks can more accurately identify various relations, such as semantic relations or spatial relations, between important objects with regard to questions, making the model more accurate and more interpretable. Among those studies, (Li et al., 2019) adds another encoder called an implicit relation encoder and applies each relation encoder directly to images to produce a graph representation for each relation. Then the model uses those representations equally to predict the answer.

Our model also uses relation encoders and graph representations but learns how much from each encoder's output will be used based on each question.

2.2 Relation-aware Graph Attention Network

The relation-aware graph attention network(ReGAT) uses a graph attention network to solve visual question answering tasks. Using a graph network to tackle such tasks was also previously explored in (Yao et al., 2018), where a pretrained semantic relation classifier was used to learn semantic relationships between objects. Using this information, a graph network was created, and graph convolution was used to finally obtain the relation-aware representation of each object. This method has been shown to be successful in image captioning. ReGAT improves this

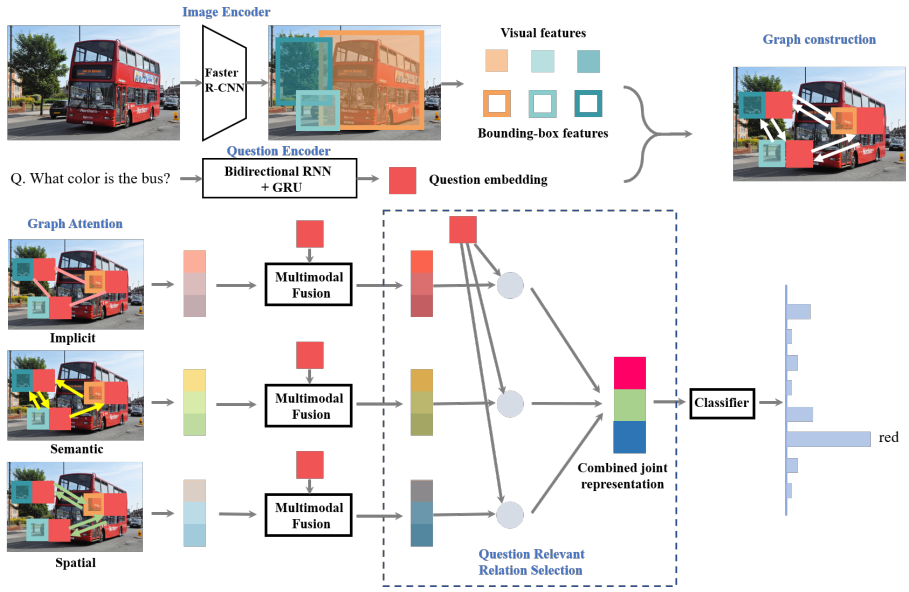

Figure 2: Overall architecture of the proposed model. After the encoders and graph attention layer, the question-relevant relation selection layer efficiently aggregates several visual relationships through an attention mechanism.

graph network using two additional relation types, spatial relations, implicit relations, and graph attention, instead of graph convolution. Spatial relation graphs are similar to semantic relation graphs but use geometric information between the objects to construct the graph. An implicit relation graph, conversely, uses no preexisting relationships between objects. A fully-connected graph is formed with the detected objects as nodes, and the interaction between objects is captured using attention over this graph. Graph attention allows each node in the neighborhood to have different importance and can capture more dynamic information between objects.

3 Model

Our model consists of four major components: a question encoder, an image encoder, a graph attention layer, and a question-relevant relation selection layer. The overall architecture of the proposed model is shown in Figure 2. In this section, we also describe the multimodal fusion method, which is a technique for fusing questions and image information.

3.1 Encoder

Our model uses a bidirectional RNN with a gated recurrent unit(GRU) (Cho et al., 2014) as the question encoder. Bidirectional RNN uses two hidden layers to process the input sequence in both directions. GRU is a simplified variant of LSTM (Hochreiter and Schmidhuber, 1997) that uses fewer parameters. The question encoder is built using these two well-known architectures. The output question embedding is later used as an input to the graph attention layer, multimodal fusion, and question relevant relation selection layer.

For the image, we use Faster R-CNN (Ren et al., 2015), which can identify the image as a set of objects. Each object has a visual feature vector v_i and a bounding-box feature b_i that contains its location information. These objects are forged into a graph that is used as inputs to the graph attention layer.

3.2 Graph Attention Layer

The graph attention (Wang et al., 2019) layer injects visual relationship information between objects into their corresponding visual features. To facilitate this process, we construct a fully-connected graph where each node represents each object from an image. Then, we aggregate the information from each node with the following procedure:
Each node in the neighborhood of v_i including itself, is projected by matrix W; then the edge weight α_{ij} is multiplied. All results are then summed and passed through a nonlinearity function to produce v_i^*, the relation-aware visual feature for object i. To make these relation graphs question-adaptive, question embedding is concatenated to each object's visual feature before applying graph attention. The way the edge weights are calculated differs depend-

ing on the type of relation graph.

Algorithm 1: Graph attention

Data: G initial graph, q question embedding, W projection matrix
Result: G^* relation-aware graph
Let G^* be an empty graph;
for *each* $v \in G.V$ **do**
  Let n be a new node;
  $S = 0$;
  for *each* $w \in G.Adj[v]$ **do**
    $w_q = [w.\text{visual}||q]$;
    $w_q = w_q W$;
    $\alpha =$ EDGE-WEIGHT$(v, w_q, w.\text{bbox}, \text{relation type})$;
    $S = S + \alpha * w_q$;
  end
  $S = \text{RELU}(S)$;
  $n.\text{visual} = S$;
  ADD-VERTEX(G^*, n);
end
return G^*;

Implicit Relation Graph If the model is trained with no predefined edge weights, an implicit relation graph is created, where the model learns the relationship between objects on its own.

The edge weights for the implicit relation graph are learned using both the visual feature v and bounding-box feature b of each object. The detailed equation is as follows (Hu et al., 2018):

$$\alpha_{ij} = \frac{\alpha^b_{ij} \cdot \exp(\alpha^v_{ij})}{\sum_{j=1}^{K} \alpha^b_{ij} \cdot \exp(\alpha^v_{ij})} \quad (1)$$

α^v_{ij} is calculated by the scaled dot-product of the two visual features:

$$\alpha^v_{ij} = (Uv_i')^T \cdot Vv_j' \quad (2)$$

α^b_{ij} is computed by the following equation:

$$\alpha^b_{ij} = \max(0, w \cdot f_b(b_i, b_j)) \quad (3)$$

,where f_b represents the geometric relationship between objects i and j. Further details are available in (Hu et al., 2018).

However, if certain relationships between objects are known beforehand and the edges are labeled based on this information, the model creates an explicit relation graph (Yao et al., 2018). We use two types of explicit relation graphs in this study.

Semantic Relation Graph The first explicit relation graph used is the semantic relation graph. Semantic relationships between objects are learned beforehand using a semantic relation classifier on a visual relationship dataset. Then, if objects i and j have relationship $p_{i,j}$, the edge between node i and j is labeled $p_{i,j}$. Objects with no semantic relationships have their edges pruned. A total of 15 such semantic relationships are used. The list of relationships used is shown in Table 1.

Relation type	Predicate list
Semantic	wearing, holding, sitting on, standing on, riding, eating, hanging from, carrying, attached to, walking on, playing, covering, lying on, watching, looking at
Spatial	1(inside), 2(covering), 3(overlap with IoU above 0.5), 4-11(overlap with IoU below 0.5)

Table 1: List of predicates used in the construction of semantic and spatial relation graphs.

Edge weights are calculated similarly to the implicit relation case but using only the visual features of each object. However, the direction and label of each edge must be considered. Further details are available in (Li et al., 2019).

Spatial Relation Graph The next explicit relation graph used is the spatial relation graph which encodes positional information between objects. Similar to the semantic relation graph, if two nodes i and j have a semantic relationship p, their edges are labeled $p_{i,j}$. Spatial relations are classified into 11 categories, and the category number and its meaning are shown in Table 1.

Attention weights are calculated in the same as with the semantic relation graph.

3.3 Multimodal Fusion

The graph attention layer produces relation-aware visual features for each object in the image. These features must be fused with question embedding to form a joint representation. The general form of multimodal fusion is computed as follows:

$$J = f(v, q) \quad (4)$$

,where v is the collection of relation-aware visual features of each object, q is the question embedding, and f is the multimodal fusion type. Popular multimodal fusion methods for VQA include bottom-up top-down (Anderson et al., 2018), multimodal Tucker fusion (Ben-Younes et al., 2017) and bilinear attention networks (Kim et al., 2018). We use BUTD and BAN fusion in the proposed model.

Bottom-up Top-down Fusion In the bottom-up top-down (BUTD) fusion method (Anderson et al., 2018), question embedding and visual features are fed into nonlinear layers and joint representation is obtained by elementwise multiplication of the results. However, because there are k object features and just one question embedding for each image-question pair, an attention mechanism is used for each image feature with the question as a query to obtain one overall summary v^* of k objects in the image:

$$v' = \sigma(vW_v + b_v) \tag{5}$$
$$q' = \sigma(qW_q + b_q) \tag{6}$$
$$p = \text{softmax}(\sigma((v' * q')W_h + b_h)) \tag{7}$$
$$v^* = v \cdot p \tag{8}$$

,where $v \in \mathbb{R}^{k \times v}$, $q \in \mathbb{R}^{1 \times q}$, $W_v \in \mathbb{R}^{v \times h}$, $W_q \in \mathbb{R}^{q \times h}$, $W_h \in \mathbb{R}^{h \times 1}$, $b_v \in \mathbb{R}^h$, $b_q \in \mathbb{R}^h$, $b_h \in \mathbb{R}^1$, and σ denote the ReLU nonlinearity function. When calculating the product of $v^{'}$ and $q^{'}$, $q^{'}$ is repeated k times, so that the same question embedding is multiplied to each of the visual features.

After obtaining v^*, it is then fed into nonlinear layers along with q, and the results are multiplied elementwise to compute the joint representation J finally as follows:

$$J = \sigma(v^* W_v^{'} + b_v^{'}) * \sigma(qW_q^{'} + b_q^{'}) \tag{9}$$

Bilinear Attention Networks The bilinear attention network(BAN) (Kim et al., 2018) fusion method takes a single-channel input and a multichannel input as inputs and combines them to form a single-channel joint representation. In the proposed model, the question vector q is the single-channel input that will be used across the multi-channel input relation-aware visual features v to produce the joint representation J. The detailed equations are as follows:

$$a = ((qU) * (vV))P \tag{10}$$
$$p = \text{softmax}(a) \tag{11}$$
$$v^* = v \cdot p \tag{12}$$

,where $v \in \mathbb{R}^{k \times v}$, $q \in \mathbb{R}^{1 \times q}$, $U \in \mathbb{R}^{q \times h}$, $V \in \mathbb{R}^{v \times h}$, and $P \in \mathbb{R}^{h \times m}$, where m denotes the number of attention heads. These equations indicate that the vector on the left side is repeated k times and multiplied elementwise to the right matrix. When using multiple attention heads($m > 1$), v^* is the concatenation of all the attended outputs.

Once v^* is obtained, the final joint representation J is calculated as follows:

$$J = ((qU^{'}) * (v^* V^{'}))P^{'} \tag{13}$$

3.4 Question Relevant Relation Selection

The QRR (question-relevant relation) layer calculates the combined joint representation J^* given the joint representation of each relation, J_{imp}, J_{sem}, J_{spa} and the question embedding q.

Most questions do not use all relations with an equal amount of importance to predict the answer. For example, in the right example of Figure 1, the question requires understanding the spatial relationship between the riders and the motorcycle. Spatial information between objects is primarily encoded in the spatial joint representation, J_{spa}. However, the semantic joint representation J_{sem}, which encodes interactive dynamics between objects, plays nearly no part in answering this question. Thus, using fixed weights for each relation(e.g., 0.3, 0.4, 0.3, respectively in the original model) to predict the answer will not produce the best result due to noise from unnecessary attention given to irrelevant relation encodings like this example. The QRR layer gradually determines which of these relations is the most essential in deriving the correct answer to the given question by feeding the three representations and the given question to an attention network.

More specifically, the QRR layer computes the combined joint representation J^* through the following attention mechanism:

$$h = \tanh((J^{'} W_v + b_v) \oplus (qW_q + b_q)) \tag{14}$$
$$p = \text{softmax}(hW_p + b_p) \tag{15}$$

,where $J^{'} \in \mathbb{R}^{3 \times d}$ is the concatenation of the three joint representations, and $q \in \mathbb{R}^q$ is the question embedding. $J^{'}$ and q are first passed through a linear layer with $W_v \in \mathbb{R}^{d \times k}$, $b_v \in \mathbb{R}^{3 \times k}$, $W_q \in \mathbb{R}^{q \times k}$, and $b_q \in \mathbb{R}^{1 \times q}$, where k is a hyperparameter denoting the dimension of the hidden layer. The operator $\oplus$ indicates that the row of the second operand is to be added to each row of the first operand. The resulting matrix is passed through tanh nonlinearity which yields $h \in \mathbb{R}^{3 \times k}$. The attention distribution over the different relations $p \in \mathbb{R}^3$ are finally obtained by passing h through a linear layer with $W_p \in \mathbb{R}^{k \times 1}$, $b_p \in \mathbb{R}^{3 \times 1}$ and the result is passed through softmax. Each element of $p = [p_{imp} \quad p_{sem} \quad p_{spa}]^T$ represents the

optimal weight of each relation given question q. The combined joint representation J^* can then be computed by the inner product of $J^{'}$ and p:

$$J^* = p_{imp}J_{imp} + p_{sem}J_{sem} + p_{spa}J_{spa} \quad (16)$$

This question-adaptive combined joint representation is then fed into the classifier to make a prediction. The combined joint representation J^* is a selective summary of the three relations tailored to the input question q. Compared to the ReGAT, which simply uses fixed weights regardless of the question, the proposed model determines weights dynamically and produces an exclusive representation of the image for the given question. The attention mechanism used in this study is similar to that used in bottom-up top-down multimodal fusion. However, in multimodal fusion, attention values are calculated among the different objects in an image. In the QRR layer, the attention distribution is computed over the three different relations, which allows the model to make more informative predictions and achieve higher accuracy. The QRR layer also adds interpretability to the original model by allowing us to examine the weight of each relation type directly.

4 Experiments

4.1 Datasets

We evaluate the proposed model using the VQA 2.0 (Goyal et al., 2017) dataset. VQA 2.0 dataset contains real images from MSCOCO (Lin et al., 2014) with questions in 3 categories: *Yes/No*, *Numbers* and *Others*. VQA 2.0 dataset was proposed to counter language priors present in the previous VQA dataset by providing complementary images that are similar but have different answers for the same question. There are 256,016 images and an average of 5.4 questions per image in the dataset. And the dataset has ten answers collected from human annotators for each image.

4.2 Implementation Details

For the question encoder, we set the question embedding dimension and GRU hidden dimension as 1024. We also set 1024 as the dimension of the relation-aware visual features and the QRR hidden layer. We use bottom-up and top-down fusion to fuse the visual features and question embedding.

We pretrain the semantic relation classifier using the Visual Genome dataset (Krishna et al., 2017), which contains 108,000 images with labels for objects, attributes, and relationships. The classifier is trained over the 14 semantic relations that we have defined in Table 1.

In the experiments, we use the PyTorch 1.3.1 (Paszke et al., 2017) framework to implement the proposed model. A batch size of 64 per GPU is used and we train the model for 20 epochs. We use a gradual warm-up learning rate, with the learning rate set initially to 0.0005 and increase linearly to 0.002 in the first 4 epochs. The learning rate is reduced by half every 2 epochs after the 15th epoch. We use the Adamax optimizer (Kingma and Ba, 2014) with weight normalization and dropout (Srivastava et al., 2014). We then train the model using a binary cross-entropy loss.

We measure the accuracy using the following metric:

$$\text{acc}(p) = \min\left(1, \frac{\sum_{i=1}^{10} 1(a_i = p)}{3}\right) \quad (17)$$

,where p is the model's prediction and a_i is the answer provided by human annotators.

Model	Yes/No	Others	Numbers	Overall
BUTD	80.30	55.80	42.80	63.20
MUTAN	**81.45**	47.17	37.32	60.17
Implicit	77.50	52.44	44.21	61.39
Semantic	76.85	51.35	44.19	60.61
Spatial	77.49	52.54	43.79	61.38
ReGAT+BUTD	78.80	45.82	53.57	62.65
ReGAT+BAN	81.22	49.87	55.45	65.02
Ours+BUTD	79.71	46.62	56.01	64.27
Ours+BAN	80.84	49.36	**56.65**	**65.37**

Table 2: Performance on VQA 2.0 dev split with different models.

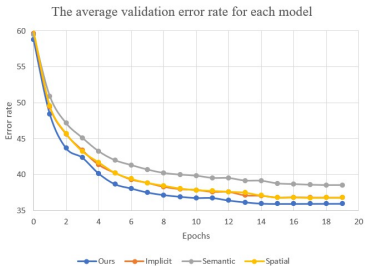

Figure 3: Average validation error rate of four models (the proposed model, implicit only, semantic only, spatial only).

4.3 Performance Comparison

Table 2 summarizes the results of the proposed experiment. We compare the results with the results

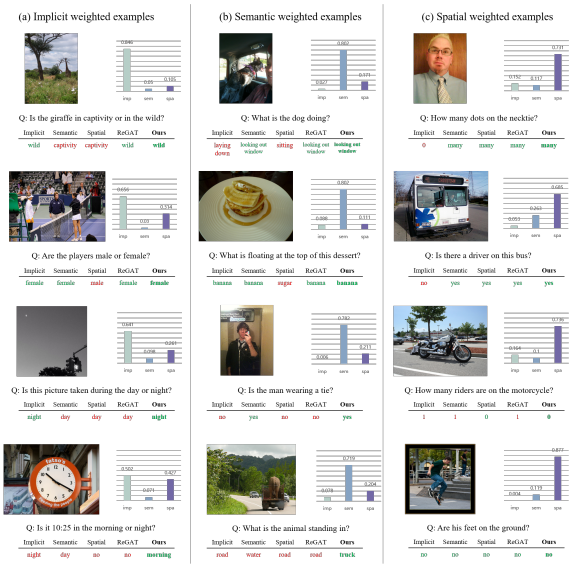

Figure 4: Model output examples and visualization of the attention weights for the QRR layer. Each column represents the cases in which each relation is given the highest weight.

of single relation encoder models and several existing VQA models, including ReGAT, BUTD and MUTAN. We also present a graph that shows the average validation error rate of each relation and our model for each epoch in Figure 3.

When using the BUTD fusion method, the proposed model outperforms ReGAT by 1.62%p in accuracy. We also observe consistent improvement in accuracy when looking at the results for each question type. Our model surpasses ReGAT by 0.91%p in *Yes/No* questions, 0.80%p in *Others* questions, and 2.44%p in *Number* questions. The table also shows the results when using BAN as the multimodal fusion method. The proposed model outperforms ReGAT by 0.35%p in accuracy overall. However, results are somewhat mixed if we consider the accuracy based on each question type. For the *Others* questions, the proposed model yields better accuracy than any other model and outperforms ReGAT by 1.20%p. For *Yes/No* and *Numbers* questions, however, the proposed model fails to achieve the accuracy produced by ReGAT by 0.38%p and 0.51%p, respectively, even though the proposed model surpasses all single relation models. Compared with other existing models, the proposed model with any fusion method outperforms BUTD and MUTAN by more than 1%p. The accuracy of each type of question shows us that the proposed model performs better with *Number* questions, even surpassing the models that outperform

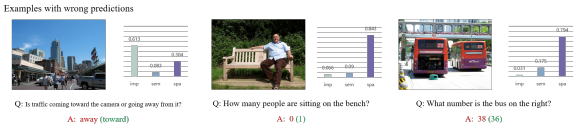

Figure 5: Cases where the model predicts incorrect answers.

the proposed model in overall accuracy.

To interpret the relative importance of each relation type, we analyze the weights used with each relation encoder on the VQA v2.0 validation dataset. On average, the implicit encoder has a weight of 5.78%, the semantic encoder has a weight of 73.70%, and the spatial encoder has a weight of 20.52%. These results show that the three relations are not equally important in answering each question, highlighting our claim that assigning fixed weights of 0.3, 0.4, and 0.3 to each encoder is not optimal. In fact, semantic relation has a much larger weight than the other two relations in most cases. We also define weights above 0.05 as the meaningful usage of that relation. Based on this criterion, 29.88% of the examples show meaningful usage of all three relations, which further highlights that in the remaining 70.12% of the dataset, using two types or one type of relation encoder is sufficient for predicting the correct answer.

4.4 Qualitative Analysis

We visualize the amount of attention given to each relation encoder depending on the input question, as shown in Figure 4. We present certain image-question pairs from the dataset that best demonstrate the usefulness of the QRR layer and visualize the attention given to each relation using a bar graph with each number representing the relative weight. We also show the predictions of the single-relation models and ReGAT below the question along with the proposed model's prediction for comparison.

In Figure 4, we present 12 image-question pairs along with predictions from each model. We organize them into three columns where each column contains examples with the most attention in implicit, semantic, and spatial relations, respectively. Across all examples, we see that the QRR layer has correctly captured the most relevant relation in answering the given question.

The examples for the implicit weighted examples in Figure 4 (a) contain questions that require a thorough understanding of the image to answer.

For example, the first entry asks whether the animal in the picture is in a given state or not. This question cannot be easily answered with only a superficial description of the image. The implicit relation graph has learned this relationship correctly, and the proposed model identified this relation as most important. The third example shows why the proposed model yields higher accuracy than ReGAT. Only the implicit-relation model yields the correct answer, possibly by connecting the small cluster of white pixels in the top left corner to an object seen at night. The other two relations provide incorrect answers; however, ReGAT cannot filter out such misleading information. The proposed model accurately selects the implicit relation as the most critical relation by giving it a weight of 0.641.

The semantically weighted examples in Figure 4 4 contain questions and answers that are heavily related to the 14 semantic relations that we have defined. The first example asks for the action of the dog. In this example, only the semantic-encoder that is most relevant to the question yields the correct answer. Unlike ReGAT that fails to answer correctly, our model gives higher weights to the most important relation to deliver the correct answer. The third example shows that ReGAT is unable to guess correctly due to suboptimal weight distribution. The proposed model blocks out all unnecessary noise by assigning the semantic relation the largest weight for this image-question pair.

The examples in Figure 4 (c) show questions that involve understanding the geometric relationship between objects. The third example demonstrates the effectiveness of the proposed model, which has correctly determined that objects that have an 'on' spatial relation with the motorcycle are the most important in giving the right answer, of which there are none. Other single-relation models and ReGAT possibly suffer from question bias and provide an incorrect answer of 1, which may be correct in many different cases.

The examples in the first and second columns of the last row are interesting in that the proposed model is the only network that has correctly predicted the answer, which shows that the proposed method can derive new answers using optimized weights for each relation type.

4.5 Error Analysis

We explore frequently observed error cases where the proposed model fails to produce the correct answer and present examples in Figure 5. For each example, the prediction of the proposed model is shown in red, and the true label is shown in green. From the examples, we observe the typical reasons for these errors. Most error cases are due to the incorrect prediction of the relation encoder itself, even though our model correctly predicts the type of visual relation. In the first image, the question asks for the direction of motion of the traffic. The bar graph on the right shows that the proposed model determines the implicit relation as the most important relation. However, the implicit relation encoder itself fails to encode such information in the visual features correctly, and our model propagates the incorrect answer to the final output. In the second image, the most important relationship is the semantic relation, where the relation "sitting on" is explicitly encoded between the objects "person" and "bench". However, the proposed model fails to yield the correct result in this case by assigning near-zero weight to the semantic relation. The final prediction then deviates from the correct answer by considering to irrelevant relations. In the last image, the question asks for the number written on the bus on the right. It is clear that the spatial relationship should be used, and indeed, the proposed model assigned the highest weight to the spatial relation. However, the predicted answer '38' is incorrect, which may occur because the quality of the picture is low, and '36' may even be interpreted as '38' by humans. Thus, even if the proposed model correctly identifies the best relation for the given question, it still predicts incorrect results if the optimal encoder itself cannot answer the question correctly.

5 Conclusion

In this paper, we propose a novel stacked attention model that assigns dynamic attention weights for various visual relations with the VQA model. We show that the proposed model yields higher accuracy than existing graph attention network models that equally consider each relation. Additionally, the proposed model, which uses an attention mechanism, has a natural form of interpretability through the visualization of learnable weights multiplied by each encoder's output. By analyzing attention weights, we show that the proposed method provides higher attention to the desired relation encoder.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.

Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620.

Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. 2019. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1989–1998.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Haoqi Fan and Jiatong Zhou. 2018. Stacked latent attention for multimodal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

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

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10313–10322.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297.

Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. 2018. Learning visual question answering by bootstrapping hard attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 299–307.

Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. 2018. Learning conditioned graph structures for interpretable visual question answering. In *Advances in neural information processing systems*, pages 8334–8343.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Badri Patro and Vinay P Namboodiri. 2018. Differential attention for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7680–7688.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia,

and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Damien Teney, Peter Anderson, Xiaodong He, and Anton Van Den Hengel. 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4223–4232.

Damien Teney, Lingqiao Liu, and Anton van Den Hengel. 2017. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. 2019. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1960–1968.

Cheng Yang, Weijia Wu, Yuxing Wang, and Hong Zhou. 2020. Multi-modality global fusion attention network for visual question answering. *Electronics*, 9(11):1882.

Zhuoqian Yang, Jing Yu, Chenghao Yang, Zengchang Qin, and Yue Hu. 2018. Multi-modal learning with prior visual relation reasoning. *arXiv preprint arXiv:1812.09681*, 3(7).

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29.

Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699.

Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. 2017. Structured attentions for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1291–1300.

A Example Appendix

This is an appendix.