

Dual Energy-Flow Enhanced Graph Neural Network for Visual Question Answering

Abstract

Scene Graphs (SG), as a structural abstraction of natural images, contain massive detailed information. Modeling visual reasoning through SG can significantly improve the ability and strengthen the interpretability of reasoning. However, existing models often fail to neither *jointly* exploit objects, relations, and attributes information in SG, nor balance the importance of objects and relations. In this paper, we introduce a novel Dual Energy-Flow Enhanced Graph Neural Network (DE-GNN), which learns a comprehensive representation by encoding full-scale scene graphs information from objects, attributes, and relations. Specifically, two types of SG structures are employed in the encoder: (i) *Object-significant graphs* which absorb attribute and relation information into nodes' representations. (ii) *Relation-significant graphs* which intensify the model's perception of relation features. In addition, we design an *energy-flow mechanism* to enhance the information transfer from edges and adjacent nodes to the updating nodes. We conduct extensive experiments on public GQA and Visual Genome datasets and achieve new state-of-the-art performances highlighting the benefits of our method.¹

Introduction

Recent developments in deep learning have accelerated the research progress in Computer Vision (CV) and Natural Language Processing (NLP) areas. Multimodal fusion tasks between image and text have attracted a growing attention, such as image captioning and visual question answering (VQA) tasks. In particular, the task of VQA requires a model to answer a free-form natural language question using visual information from an image. VQA has proven to be a crucial multimodal task with a large scope of applications such as AI assistants, multimodal customer service dialogue and image-based search to name a few.

Scene graph (SG) reasoning is an important branch of VQA tasks, see (Hildebrandt et al. 2020). To generate the scene graph, the model extracts objects' names, attributes and relationships from the images and constructs them into graph representation as illustrated in Figure 1. SG representation modeling displays several virtues over classical techniques leveraging object features extracted from images by, for e.g., convolutional neural network (CNN) since

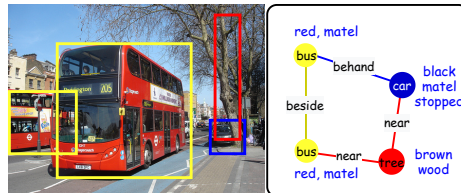


Figure 1: Visualization of a scene graph with objects, attributes, and relations information

in SG (a) the features are presented in plain and free text form (Damodaran et al. 2021), (b) it makes use of graph structures which have better interpretability (Zhang, Chao, and Xuan 2019). In this contribution, we propose two reasoning methods on scene graphs: (i) Consider scene graphs as probabilistic graphs and iteratively update nodes' probabilities using soft instructions extracted from questions such as Neural State Machine (NSM) (Hudson and Manning 2019b; Le et al. 2020); (ii) Apply Graph Neural Network (GNN) into scene graphs (Singh et al. 2019; Li et al. 2019) to learn a joint representation of the nodes and their relations, and then feed the representation into a predictor to generate the answer.

Scene graph reasoning frameworks have proven to be useful in VQA tasks, see for instance (Johnson et al. 2015; Hildebrandt et al. 2020; Yang et al. 2020). However, none of the existing methods fully utilize the scene graph information, including objects, attributes, and relations. Besides, they fail at generating comprehensive representations for objects using features from their neighbors and their attributes. Generally, information from objects and relations connected to them are reconstructed into object features in GNN-based methods. However, these encoding methods lack information from objects' attributes and objects on the other side of the edges. The NSM methods use attention mechanisms to update answer possibilities of objects, attributes, and relations, but they cannot learn the joint representation of all three types of information. Additionally, GNN-based models and NSM models are more focused towards object features, and considering relation features as references. Empirically, we demonstrate that a correct relation representation is crucial to the VQA task and enables to alleviate the bottlenecks of VQA implied by inefficient usages of scene graph

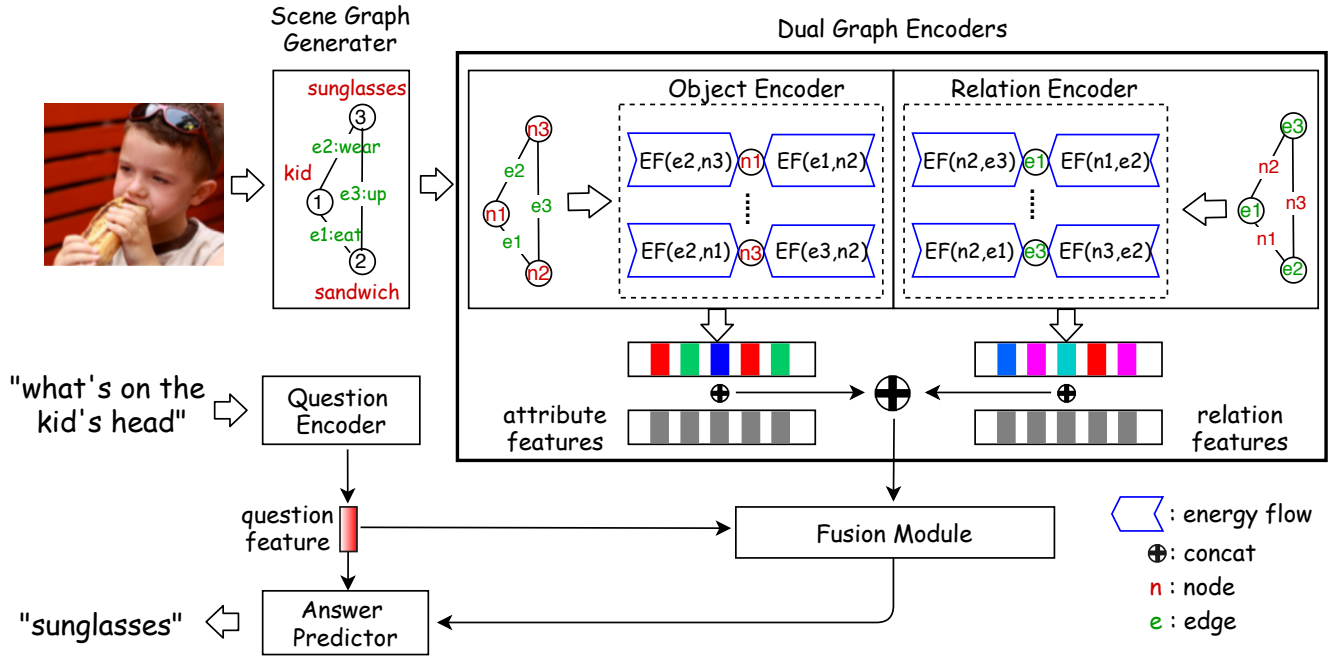


Figure 2: Model structure of the proposed Dual Energy-Flow enhanced Graph Neural Networks. EF stands for the energy-flow module. Images are transformed into scene graphs by the scene graph generator. The object-significant form and relation-significant form of the scene graph are injected into the object encoder and the relation encoder. Nodes' representations are generated from the sum of energy-flow modules. The representations are then be fused with question representation to predict an answer.

information described above.

Therefore, as a fix to current ineffective strategies, we propose the Dual Energy-Flow Enhanced Graph Neural Network (DE-GNN) for VQA, introducing a novel scene graph reasoning model that extracts full-scale feature maps from objects, attributes, and relations information in scene graphs. Concretely, as shown in Figure 2, our DE-GNN model contains a scene graph generator, a question encoder, dual graph encoders, and a fusion module. Essentially, the scene graph generator extracts graphs out of images. Besides, to preserve integrated information in the encoding process, we transform scene graphs into a relation-significant modality, in which nodes represent relations and edges represent objects, and an object-significant modality, in which nodes represent objects and edges represent relations. Lastly, after receiving scene graphs in two modalities, dual graph encoders can produce feature maps focusing on both relations and objects.

Furthermore, to learn a node's joint representation from its attributes, edges, and adjacent nodes, we modify the gated graph neural network (GGNN) structure in our proposed DE-GNN by adding the energy-flow module. It is a bidirectional GRU that guides the internal information flow. The encoder can capture information from nodes, edges, and adjacent nodes that connect to them. The outputs of the encoder pass through multi-head attention layers using question features extracted from the question encoder, see Figure 2. Hence, the model can dynamically focus on the critical parts of the questions and use the most similar part of the

scene graph as the most adequate answer.

In summary, our main contributions are as follows:

- We propose a novel DE-GNN model to learn a comprehensive representation of scene graphs by encoding graphs' object-significant modality and relation-significant modality.
- Our energy-flow module is more suitable for processing graphs with meaningful edges and nodes with internal attributes.
- We conduct experiments on GQA and Visual Genome and experimental results demonstrate the effectiveness of DE-GNN which can improve the reasoning accuracy on semantically complicated questions.

Related Works

Visual Question Answering. Most VQA approaches utilize a question encoder architecture that can learn complex temporal dynamics using a sequence of hidden states. To encode the image, most VQA approaches employ CNN-based pre-trained models like Mask-RCNN or Faster-RCNN (Fan and Zhou 2018; Patro and Nambodiri 2018; Nam, Ha, and Kim 2017). The image encoder and question encoder then pass through a multimodal fusion part and the output fusion vector pass through an answer predictor.

To learn image representations that more focused on questions, many attention-based models are proposed such as BUTD (Anderson et al. 2018), SAT (Yang et al. 2016),

question-guided spatial attention (Xu and Saenko 2016). In Lu et al. (2016), the authors proposed a hierarchical co-attention model that jointly implements both image-guided question attention and question-guided visual attention. MacNet (Hudson and Manning 2018) uses Mac-cells to combine attention and encoding function. However, there still exists a significant semantic gap between image and natural language. Transformer-based models such as Unicoder-VL (Li et al. 2020) can achieve outstanding performance on VQA tasks, but these models need complicated pretrain strategies and extra datasets. The pretraining tasks are time-consuming and hard to update under the changeable environment. To solve the existing problems in attention-based and transformer-based VQA models, we apply scene graphs as our reasoning model base.

Scene Graph Generation and Reasoning. Most scene graph generation (SGG) methods use object detection methods like mask-rcnn or faster-rcnn to extract region proposals from images (Xu et al. 2017; Yang et al. 2018; Zellers et al. 2018; Woo et al. 2018; Dai, Zhang, and Lin 2017; Li et al. 2017; Yin et al. 2018). Methods to reduce the SGG training bias has been put forward (Tang et al. 2020). Due to the graph hierarchy extracted from images, SG can promote explainable reasoning for downstream multimodal tasks such as VQA (Zhang, Chao, and Xuan 2019). In our work, scene graph generation methods are used to transform VQA datasets into scene graph datasets. Our model is then tested in datasets generated by different SGG methods.

In typical scene graph reasoning models, neural state machine (Hudson and Manning 2019b) first predicts a scene graph that represents its underlying semantics and serves as a structured model of the world. Then it performs sequential reasoning over the graph, iteratively traversing its nodes to answer a given question or draw a new inference. But, as we describe below, state machine-based models can not effectively capture complicated scene graph features. For instance, FSTT (Singh et al. 2019) uses GGNN based model to encode scene graphs but it neglects vital information from edges and attributes. Relation-aware Graph Attention Network (Li et al. 2019) encodes each image into a graph and models multitype inter-object relations via a graph attention mechanism, to learn question-adaptive relation representations. However, neither can fully use attribute information nor can it learn the comprehensive representation of scene graphs using graph attention network. Our model uses GGNN structure to learn more comprehensive scene graph representations.

Graph Neural Network. GNN (Scarselli et al. 2009) are a class of traditional neural networks methods designed to infer on data described by graphs models. A group of graph-based models (Morris et al. 2019; Liu et al. 2019) were proposed for different graph tasks including graph representation learning. Inspired by convolution neural network, graph convolutional network (GCN) (Kipf and Welling 2017) improves GNNs efficiency with fast approximated spectral operations. GAT (Velickovic et al. 2018) introduces the attention mechanism to GNN, leveraging masked self-attentional layers to address the shortcomings of prior methods based

on graph convolutions or their approximations. GGNN (Li et al. 2016) uses gated recurrent units (GRU) to accelerate the training speed and gain favorable inductive biases on large-scaled graphs. Similar to scene graphs, (Wang et al. 2016, 2018; Sun and Li 2019) apply GNN-based models on knowledge graphs. However, existing GNN-based models cannot effectively process graphs with node attributes and complicated labels. Our DE-GNN model can learn a comprehensive representation using full-scale scene graph information from objects, attributes, and relations to overcome these problems.

DE-GNN Methodology

Beforehand, we define the VQA task. It is a classification task that, given a text question about an image, output an answer. Formally, given question q and image m , the model aims to maximizing a conditional distribution over candidate answers a :

$$\hat{a} = \arg \max_{a \in A} p_{\theta}(a|q, m) \quad (1)$$

where A is the set of all possible answers, p_{θ} represents the VQA model with the trainable vector of parameters θ and \hat{a} denotes the final answer.

Our proposed architecture designed for the VQA task is illustrated in Figure 2. Our model contains a scene graph generator, a question encoder, dual graph encoders and a fusion module. For the scene graph generator, we follow the method detailed in (Tang 2020) and other three baselines referred in this work, which we will describe in the experiment section. For the question encoder, semantic questions are first projected into an embedding space using GLOVE pretrained word embedding model (Pennington, Socher, and Manning 2014). After adding a positional encoding matrix into questions, we use long short-term memory (LSTM) networks to generate questions embedding $q \in R^{dim}$. We introduce our dual GGNN encoders in the following subsection.

Object/Relation-Significant Graph

We organize scene graphs into object-significant and relation-significant modalities.

Object-Significant Graph. We define the object-significant modality as G_{obj} , every nodes represent objects in the image and every edges represent relations between two objects. Define N as the node set and E as the edge set. For $n_i, n_j \in N, e_k \in E$, $\langle n_i - e_k - n_j \rangle$ denotes the relation tuple that represents the relation e_k from object n_i to object n_j . Noted that relation tuples are not symmetrical: if $\langle n_i - e_k - n_j \rangle$ is a valid relation tuple, $\langle n_j - e_k - n_i \rangle$ may not exist. Also, n_i and n_j may have several relations.

Relation-Significant Graph. We define relation-significant modality as G_{rel} , every nodes represent relations appear between objects in the image and every edges represent objects, which is completely opposed to the object-significant modality. For $e_i, e_j \in E, n_k \in N$, $\langle e_i - n_k - e_j \rangle$ denotes the relation tuple that represents the relation e_i and e_j have a shared object n_k . Noted that relation tuples are also not symmetrical.

Attribute types. Define L as attribute types (such as material, color, etc). For each node $n_i \in N$ that corresponds to an object in the image, we define a set of $L + 1$ property variables $\{n_i^j\}_{j=0}^L$, where n_i represents the object's name embedding and n_i^l represents the embedding of node's l^{th} attribute.

Dual Encoders

In our DE-GNN model, every input scene graph is transformed into an information tuple (N, E, A_{in}, A_{out}) where:

- N is a collection of node embeddings.
- E is a collection of directed edges that specify valid relation between nodes.
- A_{in} is the adjacency matrix of incident edges.
- A_{out} is the adjacency matrix of output edges.

Let h_i^t is the hidden state of node n_i in GGNN at timestep t , then at $t = 0$, we initialize h_i^0 as the GLOVE embedding of n_i with appropriate zero padding:

$$h_i^0 = [n_i^T, 0]^T. \quad (2)$$

The incident and output edges are retrieved in the respective adjacency matrices A_{in} and A_{out} .

Energy-Flow Module To enhance the information transfer from edges and adjacent nodes to the updating nodes, we use the Energy-Flow module (EF). EF module comes as a replacement of the fully-connected layers from the original GGNN model. Take a tuple $\langle n_i, e_k, n_j \rangle$ as the processing sample of the energy-flow module. The embedding state, noted e_k , of the edge e_k and neighbor node n_j 's hidden state h_j are injected into a bidirectional GRU network as input sequence while the node n_i 's hidden state h_i is injected as the GRU's initial hidden state. The output of the GRU represents the updating information for hidden state h_i , which corresponds to the key information from edge e_k and node n_j that is related to node n_i . The sum of every GRU output is n_i 's total information gain from n_i 's adjacent nodes and edges. We detail the complete energy-flow module formula as follows:

$$EF_i(A_{in}) = \sum_{\substack{\langle n_i, e_k, n_j \rangle \in A_{in} \\ k, j}} \text{GRU}([e_k, h_j], h_i),$$

$$EF_i(A_{out}) = \sum_{\substack{\langle n_j, e_k, n_i \rangle \in A_{out} \\ k, j}} \text{GRU}([e_k, h_j], h_i),$$

where $EF_i(A_{in})$ is n_i 's incident information gain, and $EF_i(A_{out})$ is n_i 's output information gain.

Propagation Model At timestep t , the hidden states of all nodes are updated by the following gated propagator module:

$$k_i^t = [EF_i^t(A_{in}), EF_i^t(A_{out})], \quad (3)$$

where k_i^t represents the node n_i 's representation from all its incident edges, output edges and adjacent nodes.

Then, we adopt GRU-like updates to incorporate information from adjacent nodes and from the previous timestep leading to an update of each node's hidden state:

$$c_i^t = [h_i^{(t-1)}, k_i^{(t-1)}]W + b,$$

$$z_i^t = \sigma(U^z c_i^t), \quad (4)$$

$$r_i^t = \sigma(U^r c_i^t),$$

where W, U^z and U^r are referred to as the trainable weight matrices and b as a bias term. At timestep t , we denote by z_i^t and r_i^t the update and reset gates, respectively.

$$\tilde{h}_i^t = \tanh(U_1 k_i^{(t-1)} + U_2 (r_i^t \odot h_i^{(t-1)})), \quad (5)$$

$$h_i^t = (1 - z_i^t) \odot h_i^{(t-1)} + z_i^t \odot \tilde{h}_i^t. \quad (6)$$

Here, U_1 and U_2 denote the trainable parameters of the linear layers, the operator \odot is the element-wise multiplication. After T steps, the GGNN encoder generates the final hidden state map G of the graph. Finally, we compute the graph embedding $g_i \in G$ for node n_i as follows:

$$g_i = \sigma(f(h_i^T, n_i)), \quad (7)$$

where $f(\cdot, n_i)$ is the multi-layer perceptron (MLP) layer which receives the concatenation of h_i^T and n_i , then generates the final representation of node n_i .

Fusion Module and Answer Predictor

Once the dual encoders embedded in our model output the node and relation features, we first fuse the attributes into feature maps. For node feature map G^N and relation feature map G^E , the fusion feature map F^N and F^E are defined as

$$F_i^N = \begin{cases} [g_i^N, n_i^0] \\ \dots \\ [g_i^N, n_i^L] \end{cases}, F_j^E = [g_j^E, e_j], F = [F^N, F^E], \quad (8)$$

where F_i^N indicates the fusion features of node i and g_i^N is node i 's representation from the GGNN encoder. We denote by the vector (n_i^0, \dots, n_i^L) the embeddings attributes of node i . F_j^E corresponds to the fusion feature of edge j . g_j^E is edge j 's representation from the GGNN encoder. e_j is edge j 's original embedding. The full-scale feature map, noted F , is the concatenation of F^N and F^E .

Then, the question embedding q generated from the LSTM encoder and the full-scale feature map F are fed into a multi-head attention layer, where the query is stored in F and the key and values are stored in q . The reasoning vector, noted r , and which stems from the graph and the question, is computed using a weighted sum of the feature map using the scores output from the attention layer; i.e.:

$$r = \text{Attention}(F, q). \quad (9)$$

Regarding the answer predictor module, we adopt a two-layer MLP noted by $f(\cdot)$. This MLP can be viewed as a classifier over the set of candidate answers. The input of the answer predictor is the concatenation vector (q, r) . This type of classifier has been applied in many VQA models such as NSM (Hudson and Manning 2019b) and MacNet (Lu et al. 2016). Formally, the output answer reads:

$$\hat{a} = \arg \max(\text{softmax}(f((q, r)))) \quad (10)$$

Numerical Experiments

Datasets

The **Visual Genome** dataset contains 108 077 images with comprehensively annotated objects, attributes, and relations. To enrich the scene graph annotation in Visual Genome, we use a scene graph generation method and motifs (Zellers et al. 2018) to generate a new scene graph dataset. Compared with the Visual Genome dataset, the new one has the same images and questions-answers tuples, but has scene graph annotations with different qualities and biases. We split both datasets into train, valid, and test sets using a 7 : 1 : 2 ratio.

The **GQA** dataset (Hudson and Manning 2019a) focuses on real-world reasoning, scene understanding and compositional question answering. It consists of 113k images and 22M questions of assorted types and varying compositionality degrees, measuring performance on an array of reasoning skills such as object and attribute recognition, transitive relation tracking, spatial reasoning, logical inference and comparisons.

Implementation Details

We use 50-dimensional GLOVE word embeddings model (Pennington, Socher, and Manning 2014) to embed words in the scene graph and questions. In order to record the questions' position information, we set up the positional encoding matrix PE:

$$\begin{aligned} \text{PE}_{\text{pos}=2i} &= \sin(\text{pos}/10000^{2i/d_m}), \\ \text{PE}_{\text{pos}=2i+1} &= \cos(\text{pos}/10000^{2i/d_m}), \end{aligned}$$

where pos is the position of the word in the question sequence. If pos is odd, the position information is generated by a \sin function, else, it is generated by a \cos function. We also let model dimension equal to $d_m = 50$. After adding position information, the question embeddings are injected into a single-directional GRU network. The dimension of the hidden layers of the GRU is 100, and the dropout rate is 0.2.

In our energy-flow enhanced GGNN encoder, the propagator time step is 5, and we use a bidirectional GRU as our energy-flow module. Here, we set the dimension of the single GRU hidden layer to 50.

In the fusion module, we apply a multi-head attention layer with 5 heads and no dropout. Regarding the answer predictor, we select the top-2000 answer candidates and use a 2-layer MLP as the output classifier.

We use Adam (Kingma and Ba 2015) as the optimizer, and Cross Entropy Loss as the loss function during the training of our model. For motif dataset, we set the batch size to 512. For Visual Genome ground truth dataset and GQA dataset, we set the batch size to 16 due to their abundant scene graph annotations.

The learning rate is decaying depending on the epoch number. We initialize the learning rate to be $1e^{-3}$, and when 30% epochs finish, the learning rate drops to $2e^{-4}$. When 60% epochs finish, the learning rate drops to $4e^{-5}$ and it becomes $8e^{-6}$ after 80% epochs finish. We train our model and other baselines on a single V100 GPU.

Empirical Results

In this subsection, we provide the experimental results on various datasets mentioned above. The different baselines compared in our experiments all use various methods to generate the scene graphs for images. In order to ensure general fairness across the methods, we implement them from scratch, removing their scene graph generation parts to eliminate the interference of different generation methods.

Table 1 reports the results on the test sets of the VG ground truth datasets and the motif dataset generated from the VG dataset. Compared to the baseline models, we can observe that our DE-GNN model outperforms the others at 3%-4%.

In addition, we provide detailed results on the VG dataset with different question types. Compared to the other scene graph based VQA models, our model perform well in "what", "where", "who" and "why" type. Specially, our model have 6% accuracy improvement in "why" type questions, which highly require VQA models' reasoning ability.

We report on Table 3 the detailed results on the test sets of the GQA dataset. Compared to the baseline models, our DE-GNN model achieves state-of-the-art accuracy performance.

We also evaluate our model and other baselines across GQA dataset's various metrics, where "Binary" represents binary-answer questions, "Open" represents open domain questions and "Distribution" represents the distance between prediction distribution and standard answer distribution. In open domain questions which are difficult for reasoning, our model outperforms the others at 10%. In distribution metric, our model also achieves 2nd lowest score compared to other baselines.

To demonstrate that our dual encoders structure can intensify the model's perception of relation features and learn a comprehensive representation from nodes, attributes, and relations information, we make badcase analysis for baselines and our model on Motif dataset.

To classify the badcase answer category, we generate three dictionaries to record objects, attributes and relations that appear in scene graphs.

For each answer, we first extract potential relations and match them up with the relation dictionary. Then we split the objects and attributes in the answer and search objects and attributes dictionaries. As for some evasive answers containing both objects and its' attributes, we add them in both object and attribute categories.

We present Table 4 the results for our badcase analysis. Our DE-GNN model surpasses all baselines in the terms of objects detection. Also, our model reduces nearly half of the wrong answers in FSTT, Re-GAT, and NSM in the attribute aspect. Finally, our model does well in relation retrieval, outperforming GNN based FSTT and Re-GAT.

The question fusion module, which concatenates the question vector with the reasoning vector before entering into the answer predictor module, is a common module in VQA models, including NSM, FSTT, and our DE-GNN model. This method has been shown to improve the accuracy of VQA models. However, from a cognitive point of view, the question fusion module lacks of interpretability since the question features are included in the reasoning vector of the

Table 1: Visual QA accuracy(%) on different question types. VG-GT: Visual Genome with ground truth scene graphs. Motif: Visual Genome with motif scene graphs.

Question type	What	Color	Where	How	Who	When	Why	Overall
Percentage	(54%)	(14%)	(17%)	(3%)	(5%)	(4%)	(3%)	(100%)
VG-GT								
NSM (Hudson and Manning 2019b)	33.1	52.4	51.0	52.9	49.8	77.9	12.3	45.1
MLP (Jabri, Joulin, and van der Maaten 2016)	-	-	-	-	-	-	-	58.5
F-GN (Zhang, Chao, and Xuan 2019)	60.9	53.6	62.0	46.2	63.3	83.7	50.9	60.1
U-GN (Zhang, Chao, and Xuan 2019)	61.6	54.0	62.4	45.9	63.9	83.2	50.3	60.5
SAN (Yang et al. 2016)	-	-	-	-	-	-	-	62.6
FSTT (Singh et al. 2019)	65.5	45.6	70.1	47.8	68.3	82.1	91.5	65.6
ReGAT (Li et al. 2019)	72.1	70.8	64.4	68.9	72.7	65.0	92.3	71.2
DEGNN (ours)	75.9	64.9	73.1	66.8	82.6	81.4	98.8	75.4
Motif								
NSM (Hudson and Manning 2019b)	31.8	62.4	53.1	51.4	47.6	83.3	10.9	43.1
FSTT (Singh et al. 2019)	48.8	40.4	49.2	40.1	40.6	54.5	70.3	48.1
F-GN (Zhang, Chao, and Xuan 2019)	58.7	60.8	60.4	47.2	61.8	84.8	49.0	60.0
U-GNN (Zhang, Chao, and Xuan 2019)	59.4	58.2	60.3	54.3	66.6	85.3	48.1	60.5
ReGATT (Li et al. 2019)	75.4	69.2	57.6	69.9	69.1	57.4	91.8	69.9
DEGNN (ours)	79.4	67.6	62.7	65.3	72.8	63.0	96.1	72.9

Table 2: Ablation study

Models	Acc.
Base	35.4%
+ <i>EF</i>	-%
+ <i>Oj</i>	-%
+ <i>Oj+EF</i>	39.3%
+ <i>Re</i>	-%
+ <i>Re+EF</i>	38.8%
+ <i>Oj+Re</i>	-%
+ (w/o <i>QF</i>)	54.9%
+ (w/o <i>attr</i>)	71.6%
DEGNN(ours)	75.2%

Table 3: Performance on the GQA dataset.

Models	Binary \uparrow	Open \uparrow	Validity \uparrow	Distribution \downarrow	Accuracy \uparrow
Human	91.20	87.40	98.90	-	89.30
BottomUp	66.64	34.83	96.18	5.98	49.74
MAC	71.23	38.91	96.16	5.34	54.06
SK T-Brain	77.42	43.10	96.26	7.54	59.19
PVR	77.69	43.01	96.45	5.80	59.27
GRN	77.53	43.35	96.18	6.06	59.37
Dream	77.84	43.72	96.38	8.40	59.72
LXRT	77.76	44.97	96.30	8.31	60.34
NSM	78.94	49.25	96.41	3.71	63.17
ReGAT	83.57	62.58	92.70	9.32	70.50
DEGNN(ours)	69.79	72.21	93.80	3.78	71.21

Table 4: Badcase Analysis on Motif dataset.

Models	NSM	FSTT	ReGAT	DE-GNN
Relation	2656	3638	3531	2605
Object	4177	2588	2089	1113
Attribute	8410	7687	3531	2605

model. Also, the addition of the question fusion module may lead the reasoning model to only *guess answers* from questions, which negatively influences the reasoning itself. We retrain our model and other baselines without the question fusion module to evaluate the reasoning ability without the influence of outer question information. Table 2 shows the results of models without question fusion module. Note that there is no question fusion module in the Re-GAT baseline, so the Re-GAT result is the same as Table ???. After reducing the concatenation of the question vector and reasoning vector, FSTT, NSM, and our DE-GNN model suffer accuracy recessions. Without question fusion, our model still outperforms other baselines.

Ablation Study

We compare three ablated forms of DE-GNN with our complete one. The accuracy results are reported in Table ?? and are obtained using the Motif dataset. The Object-EF model corresponds to the object encoder part of our DE-GNN model, which contains one energy-flow enhanced GGNN network to encode *object-significant* graphs. On the other hand, the relation-EF model is the other half of our DE-GNN model, which contains one energy-flow enhanced GGNN network to encode *relation-significant* graphs. We use the original GGNN network as the baseline model to encode object-significant graphs.

First, we validate the effectiveness of applying dual structure to balance the importance of relations and objects by splitting our DE-GNN into an object-single model and a relation-single model. Table ?? shows that both object-EF model and relation-EF model perform poorly, at about 39%. It also shows that both relations and objects are vital to VQA performance. Lack of any of them leads to severe accuracy recession. Combining the object-single model and the

relation-single model leads to an empirical gain of approximately 35% accuracy upward, which shows that the dual structure is significant in balancing relation and object information.

Then, we validate the effectiveness of applying energy-flow structure to learn a more comprehensive representation for scene graphs than the original GGNN structure, which represents the baseline in Table ?? . We compare the object-EF model and the baseline model, which both learn representations from object-significant graphs, and note that after adding the energy-flow structure, there is an accuracy improvement of around 3%. This latter point shows that energy-flow structure can successfully improve the representation quality of scene graphs.

Visualization

To better illustrate the effectiveness of the dual encoder structure and the energy-flow module in our DE-GNN model, we compare the attention scores learned by DE-GNN model with those learned by our baseline, object-EF, and relation-EF models. Figure 3 exhibits the detail of the visualization results. The top row corresponds to three typical input images along with their associated questions.

Comparing the first graph in row 2 with row 3 shows that energy-flow enhanced GGNN encoder can more correctly focus on crucial objects than the original GGNN encoder. For the objects mentioned in the questions, the attention scores of "bus" increase by 0.29 with the addition of the energy-flow module. As for objects unrelated to questions, the attention scores of "car" and "tree" decrease by 0.12 and 0.08. The original GGNN encoder can not fuse object and attribute information, which leads to the wrong answer. Our object-EF model learns a jointly representation from objects and attributes. The top-3 candidate attributes for the question are "red", "steel", and "driving" while the object-EF model correctly answers "red".

Comparing the first and third graphs in row 3 with row 4 shows that both object-EF and relation-EF can capture related information, but these models have a significant bias on objects and relations. Relation-EF shows the acute perception on relations, but the model can not capture object information or score objects correctly, which leads to relation-EF's failure on the first graph. The object-EF model can not capture "on" relation in graph 3, while relation-EF model can easily capture the correct answer.

Our DE-GNN model, presented row 5 of Figure 3, does not only balance the importance of objects and relations, but also learn jointly representations from objects, attributes, and relations using the energy-flow module. Comparing the second graph in row 3 and row 5, our model correctly capture "horse" object and "behind" relation. This leads to the correct attention score of "old man", which is 0.61 higher than the score from object-EF. Note from Figure 3 that our model answers all three questions correctly unlike baselines.

Conclusion

In this work, we study classical scene graph reasoning methods such as GNN-based models and Neural State Machine



Figure 3: Visualization of attention scores learned by baseline, object-EF, relation-EF, and DE-GNN.

for Visual Question Answering tasks. We observe that existing methods fail to *jointly* exploit nodes, relations, and attributes information in scene graphs. To address this problem, we propose the Dual Energy-Flow Enhanced Graph Neural Network (DE-GNN), which encodes each scene graph into feature representations via an object encoder and a relation encoder generating full-scale feature maps using nodes, attributes, and relations information. We demonstrate the effectiveness of our method on the various datasets achieving significant improvement and state-of-the-art performances.

In future work, we will test our DE-GNN model on more VQA datasets, such as VQA-CP2. We hope that our work can further enhance the effect of scene graph for reasoning modeling.

References

- Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 6077–6086. IEEE Computer Society.
- Dai, B.; Zhang, Y.; and Lin, D. 2017. Detecting Visual Relationships with Deep Relational Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3298–3308. IEEE Computer Society.
- Damodaran, V.; Chakravarthy, S.; Kumar, A.; Umapathy, A.; Mitamura, T.; Nakashima, Y.; Garcia, N.; and Chu, C. 2021. Understanding the Role of Scene Graphs in Visual Question Answering. *CoRR*, abs/2101.05479.
- Fan, H.; and Zhou, J. 2018. Stacked Latent Attention for Multimodal Reasoning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 1072–1080. IEEE Computer Society.
- Hildebrandt, M.; Li, H.; Koner, R.; Tresp, V.; and Günnemann, S. 2020. Scene Graph Reasoning for Visual Question Answering. *CoRR*, abs/2007.01072.
- Hudson, D. A.; and Manning, C. D. 2018. Compositional Attention Networks for Machine Reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Hudson, D. A.; and Manning, C. D. 2019a. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 6700–6709. Computer Vision Foundation / IEEE.
- Hudson, D. A.; and Manning, C. D. 2019b. Learning by Abstraction: The Neural State Machine. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 5901–5914.
- Jabri, A.; Joulin, A.; and van der Maaten, L. 2016. Revisiting Visual Question Answering Baselines. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, 2016, Proceedings, Part VIII*, volume 9912 of *Lecture Notes in Computer Science*, 727–739. Springer.
- Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Fei-Fei, L. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3668–3678.
- Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Le, T. M.; Le, V.; Venkatesh, S.; and Tran, T. 2020. Neural Reasoning, Fast and Slow, for Video Question Answering. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, 1–8. IEEE.
- Li, G.; Duan, N.; Fang, Y.; Gong, M.; and Jiang, D. 2020. Unocoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 11336–11344. AAAI Press.
- Li, L.; Gan, Z.; Cheng, Y.; and Liu, J. 2019. Relation-Aware Graph Attention Network for Visual Question Answering. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 10312–10321. IEEE.
- Li, Y.; Ouyang, W.; Zhou, B.; Wang, K.; and Wang, X. 2017. Scene Graph Generation from Objects, Phrases and Region Captions. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 1270–1279. IEEE Computer Society.
- Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. S. 2016. Gated Graph Sequence Neural Networks. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Liu, Z.; Chen, C.; Li, L.; Zhou, J.; Li, X.; Song, L.; and Qi, Y. 2019. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4424–4431.
- Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 289–297.
- Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 4602–4609. AAAI Press.
- Nam, H.; Ha, J.; and Kim, J. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. In *2017*

- IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2156–2164. IEEE Computer Society.
- Patro, B. N.; and Namboodiri, V. P. 2018. Differential Attention for Visual Question Answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 7680–7688. IEEE Computer Society.
- Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Doha, Qatar: Association for Computational Linguistics.
- Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Trans. Neural Networks*, 20(1): 61–80.
- Singh, A.; Mishra, A.; Shekhar, S.; and Chakraborty, A. 2019. From Strings to Things: Knowledge-Enabled VQA Model That Can Read and Reason. 4601–4611.
- Sun, M.; and Li, P. 2019. Graph to Graph: a Topology Aware Approach for Graph Structures Learning and Generation. In Chaudhuri, K.; and Sugiyama, M., eds., *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, 2946–2955. PMLR.
- Tang, K. 2020. A Scene Graph Generation Codebase in PyTorch. <https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch>.
- Tang, K.; Niu, Y.; Huang, J.; Shi, J.; and Zhang, H. 2020. Unbiased Scene Graph Generation From Biased Training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 3713–3722. IEEE.
- Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Wang, Y.; Che, W.; Guo, J.; and Liu, T. 2018. A Neural Transition-Based Approach for Semantic Dependency Graph Parsing. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 5561–5568. AAAI Press.
- Wang, Y.; Guo, J.; Che, W.; and Liu, T. 2016. Transition-Based Chinese Semantic Dependency Graph Parsing. In Sun, M.; Huang, X.; Lin, H.; Liu, Z.; and Liu, Y., eds., *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data - 15th China National Conference, CCL 2016, and 4th International Symposium, NLP-NABD 2016, Yantai, China, October 15-16, 2016, Proceedings*, volume 10035 of *Lecture Notes in Computer Science*, 12–24.
- Woo, S.; Kim, D.; Cho, D.; and Kweon, I. S. 2018. LinkNet: Relational Embedding for Scene Graph. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 558–568.
- Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene Graph Generation by Iterative Message Passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3097–3106. IEEE Computer Society.
- Xu, H.; and Saenko, K. 2016. Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, 2016, Proceedings, Part VII*, volume 9911 of *Lecture Notes in Computer Science*, 451–466. Springer.
- Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph R-CNN for Scene Graph Generation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, volume 11205 of *Lecture Notes in Computer Science*, 690–706. Springer.
- Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A. J. 2016. Stacked Attention Networks for Image Question Answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 21–29. IEEE Computer Society.
- Yang, Z.; Qin, Z.; Yu, J.; and Wan, T. 2020. Prior Visual Relationship Reasoning For Visual Question Answering. In *2020 IEEE International Conference on Image Processing (ICIP)*, 1411–1415. IEEE.
- Yin, G.; Sheng, L.; Liu, B.; Yu, N.; Wang, X.; Shao, J.; and Loy, C. C. 2018. Zoom-Net: Mining Deep Feature Interactions for Visual Relationship Recognition. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, 330–347. Springer.
- Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural Motifs: Scene Graph Parsing With Global Context. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 5831–5840. IEEE Computer Society.
- Zhang, C.; Chao, W.; and Xuan, D. 2019. An Empirical Study on Leveraging Scene Graphs for Visual Question Answering. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, 288. BMVA Press.