Hypergraph Attention Networks for Multimodal Learning

Eun-Sol Kim[1*†] Woo Young Kang[1*] Kyoung-Woon On[2] Yu-Jung Heo[2‡] Byoung-Tak Zhang[2,3]

[1]Kakao Brain
[2]Department of Computer Science and Engineering, Seoul National University
[3]AI Institute (AIIS), Seoul National University

Abstract

One of the fundamental problems that arise in multimodal learning tasks is the disparity of information levels between different modalities. To resolve this problem, we propose Hypergraph Attention Networks (HANs), which define a common semantic space among the modalities with symbolic graphs and extract a joint representation of the modalities based on a co-attention map constructed in the semantic space. HANs follow the process: constructing the common semantic space with symbolic graphs of each modality, matching the semantics between sub-structures of the symbolic graphs, constructing co-attention maps between the graphs in the semantic space, and integrating the multimodal inputs using the co-attention maps to get the final joint representation. From the qualitative analysis with two Visual Question and Answering datasets, we discover that 1) the alignment of the information levels between the modalities is important, and 2) the symbolic graphs are very powerful ways to represent the information of the low-level signals in alignment. Moreover, HANs dramatically improve the state-of-the-art accuracy on the GQA dataset from 54.6% to 61.88% only using the symbolic information in quantitatively.

1. Introduction

In this work, we address multimodal learning tasks, which deal with relating information from multiple sources, such as Visual Question and Answering tasks (with image and text), visual captioning (with image and text), and video understanding (with image, text, and sound). As neural network-based methods have been successively used to deal with large-scale unimodal data, such as images, natural languages, and audio signal inputs, those methods have been applied to multimodal learning. However, there is a severe lack of consideration regarding the adequate form of the input representations of the multimodal data to learn by using the neural network-based methods.

Most of the previous researches on learning multimodal inputs commonly take the following steps: to make input features of each modality as vector forms after applying pre-trained pre-processing methods, to integrate the multiple input features into a common vector space, and to apply problem-specific modules usually implemented with fully connected neural networks. Specifically, in the integration step, the feature vectors from different modalities are considered as abstracted information on the equivalent level, even though those are obtained from totally different pre-processing steps. In this conventional process, we argue that aligning the information level of heterogeneous modalities is a fundamental problem of multimodal learning and suggest a novel method to bind the modalities in a common semantic level.

To tackle this problem, we suggest using the symbolic graphs as the common semantic representation for multimodal learning. We define the symbolic graphs as directed graphs which contain nodes and edges, the nodes present semantic units with textual form and edges present the relationship between them. For example, scene graphs [19] can be used as the symbolic graphs for the image modality and dependency trees in natural sentences for the text modality. By extracting the symbolic graphs from each low-level inputs, we can compare the semantics between modalities in the same abstraction level.

Based on the symbolic graphs which are on the same semantic space, multimodal inputs can be effectively integrated. Here, we suggest a new graph neural net-based algorithm, called Hypergraph Attention Networks (HANs), which exploit the sub-structure of the graph to integrate symbolic information. The main idea of HANs is to construct the co-attention maps between multimodal inputs and to

*These authors contributed equally to this study.
†Corresponds to eunsol.kim@kakaobrain.com
‡Work done during internship at Kakao Brain.

integrate the inputs with the co-attention maps. While conventional attention methods usually compare node values independently to make attention maps, HANs consider structural similarity to consider high-level semantic similarity.

We show the effectiveness of the suggested method with the most popular application in a multimodal learning task, i.e., Visual Question Answering. We demonstrate the performance of HANs on two recent Visual Question Answering (VQA) datasets: VQA2.0 [39] and GQA [11] which focus on real-world visual reasoning and multi-step question answering. From the qualitative analysis of the suggested method with two datasets, we argue that 1) the symbolic graphs are a very powerful way to represent the information of the low-level signals, and 2) to align the information level between modalities is the fundamental problem. Quantitatively, also, the suggested method dramatically improves the state-of-the-art on the GQA dataset from 54.6% to 61.88% only using the symbolic information.

2. Related Work

In this section, previous works related to structural learning with neural networks and Visual Question Answering (VQA) tasks are summarized.

2.1. Graph Matching Algorithms

In our knowledge, there are a few studies exactly related to the suggested method, which deal with the problem of integrating multimodal inputs in graph forms. For this reason, instead, we review the studies of learning similarity of graphs and connect it to attention mechanism, which partially related to the suggested method.

The similarity between the two graphs can be defined by graph Weisfeiler-Lehman isomorphism test [4]. Recently, Xu et al. [34] showed that the representations learned by Graph Neural Networks (GNNs) could be at most as powerful as the Weisfeiler-Lehman graph isomorphism test. That is, the representations with sufficient message passing can be used to determine whether two graphs are isomorphic or not. Based on [34], Li et al. [21] proposed Graph Matching Networks (GMNs) to learn the similarity between two graphs. In GMNs, node representations are updated not only with message passing in each graph but also cross-graph attention mechanism to learn the similarity between two graphs. Because the message passing can capture the dependence of the graph, the cross-graph attention used in GMNs can grasp structural similarity in two graphs.

2.2. Visual Question Answering

Visual Question Answering (VQA) is one of the representative multimodal learning tasks to answer a textual question about an image scene. The conventional VQA models [1, 36, 16, 17, 23, 6, 38, 15] learn the joint embedding of a pair of the question and the image with two stages. First, it learns image features and question embeddings based on pre-trained models (e.g., pre-trained CNNs models for an image and Word2Vec models for a question). Second, it combines the learned visual features with question embeddings using a multimodal pooling and an attention mechanism. Kim et al. [17] proposed multi-modal low-rank bilinear pooling (MLB), which approximates bilinear pooling between two input embeddings with efficient computation, by enforcing the rank of the weight tensor to be 1. Yu et al. [38] generalized MLB to Multi-modal Factorized Bilinear Pooling (MFB), as the rank of the weight tensor larger than 1. Bilinear Attention Network (BAN) [15] extends MLB in the respect that it considers bilinear interactions between two input groups, such as multiple feature sets of the question and the image. Also, based on a powerful self-attention mechanism [30], Tan & Bansal proposed a cross-modal Transformer to learn vision-and-language interactions [27].

2.3. VQA with Graph Structure

The approaches modeling object interactions through graph representations have been getting a growing interest in the computer vision field. For the VQA task, Teney et al. [28] initially proposed a method combining graph representations of questions and abstract images with Graph Neural Networks (GNNs). Also, the methods to model interactions between objects through implicit and explicit graph structure are proposed for counting problem [40, 29]. High-level semantic information such as attribute and the visual relationship was also exploited with [20, 37, 32, 31] to make the model more powerful and interpretable. Norcliffe-Brown et al. [24] introduced a method to construct a semantic structure in image conditioned on a question. Later, Cadene et al. [5] extend this idea to modeling spatial semantic pairwise relations between all pairs of regions. Recently, a conditional iterative message passing algorithm for VQA and GQA datasets was proposed to learn context-aware node representations conditioned on a given question [9]. Also, Hudson et al. [12] suggested the Neural State Machine (NSM) to address vision and language information on a symbolic level. To solve the GQA task, NSM first predicts a probabilistic scene graph. Then, to answer a given question, they perform sequential reasoning over the graph based on an iterative node traversing algorithm.

3. Hypergraph Attention Networks

The main purpose of the suggested method is to align information levels between multimodal inputs and to integrate the inputs within the same information level. We define the common semantic space between the modalities with the symbolic graphs. After extracting symbolic graphs of each modality, the semantics between two graphs are compared, and then the co-attention maps are constructed based on the semantic similarities. Then, the joint representation of the

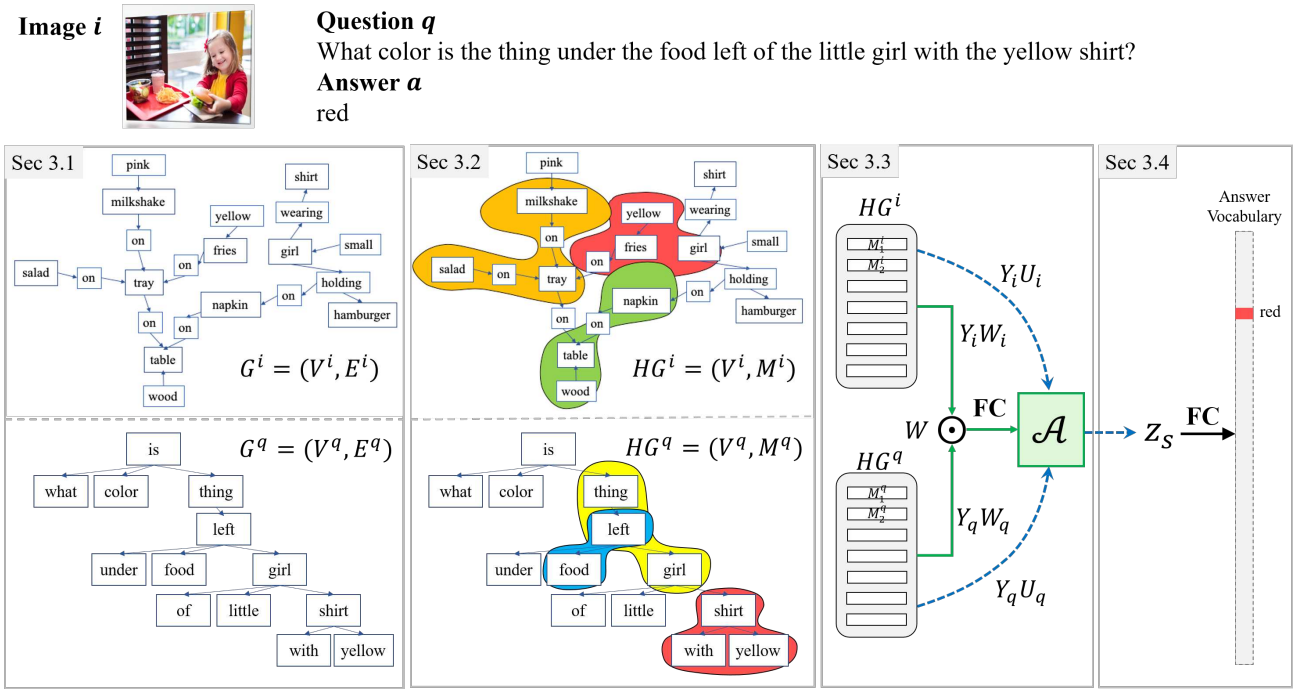


Figure 1. The overall architecture of the suggested model. For a given pair of image and question, two symbolic graphs are constructed. After constructing the symbolic graphs G^i and G^q, two hypergraphs HG^i and HG^q with random-walk based hyperedge are constructed. By comparing the semantics of each hyperedges, a co-attention map $\mathcal{A}$ is constructed. The two hypergraphs are combined by the co-attention map $\mathcal{A}$, and the final representation z_s is used to predict an answer for the given question.

multimodal inputs is constructed based on the co-attention maps.

The suggested method, called Hypergraph Attention Networks (HANs), consists of four components: (1) constructing symbolic graphs, (2) sampling random-walk paths on the symbolic graphs to construct the hypergraphs, (3) matching semantics between hyperedges to construct co-attention maps, and (4) integrating the hypergraphs to get the final representation of the multimodal inputs. The overall architecture of the proposed approach is shown in Figure 1.

To make clear the further discussion, HANs is explained with a specific multimodal learning task, Visual Question and Answering that has a different level of information in vision modality (image) and language modality (text question).

3.1. Constructing Symbolic Graphs

The symbolic representations of the two modalities are defined with graph forms.

For the image modality, symbolic graphs of the images $G^i = \{V^i, E^i\}$ are constructed based on the scene graph information [14]. V^i is the set of nodes that correspond to words of object labels, attributes, and the relations between the objects. The object labels and attributes represent the name of the object and color, the shape of the object, respectively. In addition, the relationships between two objects are described with predicate phrases, *e.g.* `to the left of`. From that information, the symbolic graph $G^i = (V^i, E^i)$ of an image is defined with a set of nodes $V^i = \{v_1^i, v_2^i, ..., v_S^i\}$ correspond to the set of words for labels, attributes, and predicates. Furthermore, the set of edges E^i are defined as following rules: (1) if a object node v_j^i has an attribute v_k^i, then $(j,k) \in E^i$, (2) if two objects v_j^i and v_k^i have a relationship v_l^i, then $(j,l) \in E^i$ and $(l,k) \in E^i$. The reason to make edge-labeled scene graphs flat is to align the structure between G^q and G^i.

For the text modality, we obtain the dependency tree of the question sentence by using the Spacy library[1]. The symbolic representation of the question $G^q = \{V^q, E^q\}$ consists of the set of tokens (V^q) and the dependency between the tokens (E^q). In detail, $(i,j) \in E^q$ if v_i^q and v_j^q has the dependency.

As both V^i and V^q correspond to *word representations*, we consider two symbolic graphs are in the common (same) information level.

3.2. Constructing the Hypergraphs

After building two symbolic graphs G^q and G^i, the co-attention map $\mathcal{A}$ is constructed by matching semantics of their sub-graphs. As the sub-graph matching problem is one of the NP-hard problems, we suggest a simple but very

[1]https://spacy.io/

powerful approximate algorithm, HANs. We consider each hyperedge (a sequence of nodes sampled by random-walk algorithm along with directed edges) as a sub-graph, so $\mathcal{A}$ is constructed by calculating the similarity between the hyperedges from the G^i and G^q.

From G^q and G^i, two probability distributions are defined to construct the hypergraphs. The initial probability that a node v_i will be selected is defined with,

$$P^0_{v_i} = \frac{deg^+(v_i) + \epsilon}{\sum_{j=1}^{N} deg^+(v_j)}$$

where N and $deg^+(v_i)$ represent the number of total nodes and out-going edges from node v_i, respectively. In addition, the transition probability for both P^q and P^i is defined with,

$$P_{v,u} = \begin{cases} \frac{1-\epsilon}{deg^+(v)}, & \text{if } (v,u) \in E \\ \epsilon, & \text{if } (v,u) \notin E \end{cases}$$

where v and u are arbitrary nodes of a graph.

Along with P^q and P^i, S^q and S^i random work steps for G^q and G^i are conducted. In other words, a random-walk path is defined by a transition sequence $v_0 \rightarrow v_1 \rightarrow ... \rightarrow v_k$, which starts from a random node $v_0 \in sample(P^0)$ and samples k node to transition to next node as $v_{i+1} := sample(P_{v_i})$.

Now, the nodes in a random-walk path are connected in a hyperedge, and then two hypergraphs $HG^i = (V^i, M^i)$ and $HG^q = (V^q, M^q)$ can be obtained, where a $m^i \in M^i$ corresponds to $v^i_0 \rightarrow v^i_1 \rightarrow ... \rightarrow v^i_k$.

3.3. Building Co-attention Maps between Hypergraphs

Now, the sub-graph matching problem to get the co-attention map is approximated with the method which matches the semantics between the hyperedges. In this section, we define the semantics of each hyperedge M and explain the method to compare the semantics between the hyperedges.

As each node v represents a symbol in word level, the semantic of each hyperedge M can be defined by combining the word representations within the same hyperedge. We suggest a simple but powerful way to define the semantics by using pre-defined word vectors, such as GloVe [25].

$$y(m) := f(g_0, g_1, ..., g_k) \tag{1}$$

where $g \in \mathbb{R}^{300}$ represents a 300 dimensional GloVe vector [25] of a node v. A simple mean function is used for f, so $y(m)$ can be represented with a real-valued vector in $\mathbb{R}^{300}$.

Now, the co-attention map $\mathcal{A}$ is built by measuring similarities between semantics of two hyperedges $y(m^i)$ and $y(m^q)$. For the similarity measure, the low-rank bilinear pooling method is used as follows.

$$\mathcal{A} = \text{softmax}(W \circ (Y_q W_q)(Y_i W_i)^\top) \tag{2}$$

where $Y_q \in \mathbb{R}^{N^q \times 300}$, $Y_i \in \mathbb{R}^{N^i \times 300}$ represent k-step hyperedges sampled from a dependency tree and a scene graph. $W_q, W_i \in \mathbb{R}^{300 \times h}$ and $W \in \mathbb{R}^{N^q \times N^i}$ represent linear mappings which are all learnable parameters.

Here, the co-attention map has two interesting characteristics. First, the co-attention map $\mathcal{A}$ is based on comparing the semantics with the symbolic representations, while previous works on the neural representations having different information levels. Second, the suggested method considers not only unitary relationships between two nodes, but also the inherent structures by constructing the hypergraphs, while most of the previous researches on the graph matching compare the (neural) representations between two nodes.

Furthermore, in terms of the semantics of the hyperedges $y(m)$, we can consider utilizing the structural information of the symbolic graphs. To get the informative node representations by considering the information of neighboring nodes, message passing based Graph Neural Network (GNN) [7] is designed[2].

For the node feature matrix with GloVe vector $X \in \mathbb{R}^{S \times d}$, where S is the number of nodes and d is the dimension of GloVe vector, the new node feature matrix $X_{new} \in \mathbb{R}^{S \times d}$ can be obtained as follows:

$$\begin{aligned} Z_{\text{in}} &= \sigma(D_{in}^{-1} A X W_{\text{in}} + X W_{\text{in}}) \\ Z_{\text{out}} &= \sigma(D_{out}^{-1} A^\top X W_{\text{out}} + X W_{\text{out}}) \\ X_{new} &= \sigma((Z_{\text{in}} \circ Z_{\text{out}}) W_{\text{mrg}}) \end{aligned} \tag{3}$$

where $A \in \{0,1\}^{S \times S}$ is an adjacency matrix corresponds to E, *i.e.,* $A_{i,j} = 1$ if $(i,j) \in E$ and $A_{i,j} = 0$ otherwise. $D_{in}, D_{out} \in \mathbb{R}^{N \times N}$ are indegree, outdegree (diagonal) matrix of A, respectively. All W_{in}, W_{out}, W_{msg} are learnable parameters. Also, $\circ$ is the element-wise multiplication. We also employ a residual connection [8] followed by layer normalization [3]. Now, $y(m)$ can be newly defined with X_{new}. In Tabel 1, the effectiveness of using X_{new} will be analysed.

3.4. Getting Final Representations

As the equation (2) provides the co-attention matrix $\mathcal{A} \in \mathbb{R}^{N^q \times N^i}$, we can integrate two hypergraphs $HG^i = (V^i, M^i)$ and $HG^q = (V^q, M^q)$ using any bilinear operator B, such as BAN [15] or MFB [38].

Formally, a final representation $\mathbf{z}_s$ for integrating G^q and G^i is inferred by applying a bilinear operator B to $Y_q \in \mathbb{R}^{N^q \times 300}$, $Y_i \in \mathbb{R}^{N^i \times 300}$ and $\mathcal{A} \in \mathbb{R}^{N^q \times N^i}$. If we choose BAN as $U_q, U_i \in \mathbb{R}^{300 \times h}$, $\mathbf{z}_s$ can be represented as follows:

$$\mathbf{z}_s = (Y_q U_q)^\top \mathcal{A} (Y_i U_i) \tag{4}$$

[2]In this work, as the symbolic graph is a directed graph, both outgoing and incoming message passing procedures are considered.

Then, $\mathbf{z}_s$ is used to predict an answer word with a fully connected layer.

One thing that should be noted is that the integration of the image and the question sentence is only through indirect means, soft co-attention maps. Consequently, the interaction between these two modalities is mediated through probability distributions only.

3.5. Merging Visual Features

In addition to the integration of the symbolic level information discussed at Section 3.4, here we show a simple way to utilize given visual features with the integrated symbolic features. Firstly, we define the visual feature for each object in an image as $V_i \in \mathbb{R}^{N^v \times d}$. In this work, the visual features for each object are extracted from the pre-trained BUTD model [1]. Then, we project the V_i and Y_q onto same dimensional space using two one-layered fully connect layers. Now, we get $\hat{Y}_q \in \mathbb{R}^{N^q \times \hat{d}}$ and $\hat{V}_i \in \mathbb{R}^{N^v \times \hat{d}}$. Next, co-attention map $\mathcal{A}^*$ for $\hat{Y}_q$ and $\hat{V}_i$ can be predicted by using equation (2) and the visual-semantic feature $\mathbf{z}_v$ can be represented as follows:

$$\mathbf{z}_v = (\hat{Y}_q \hat{U}_q)^\top \mathcal{A}^* (\hat{V}_i \hat{U}_i) \tag{5}$$

where $\hat{U}_q, \hat{U}_i \in \mathbb{R}^{\hat{d} \times h}$. Finally, we combine the $\mathbf{z}_s$ and $\mathbf{z}_v$ by using two blocks of MRN [16] and the final output is used to predict an answer word with a fully connected layer.

4. Experimental Results

4.1. Two Visual Question Answering Datasets

In this work, two kinds of VQA dataset are used for experiments, which are Graph Question Answering (GQA, [11]) and VQA v2 [2, 39].

GQA dataset [11] is a new question and answering dataset featuring compositional questions over real-world images, with more than 110K images and 22M questions. Each question is associated with a structured representation of its semantics and a functional program that specifies the reasoning steps has to be taken to answer it. Each image is associated with a scene graph of the image's objects, attributes, and predicates. 1,740 objects, 620 attributes, and 330 predicate labels are defined as a semantic ontology for GQA. Each image contains 16.4 distinct objects, and each object has 0.54 attributes and 3.08 relationships on average. The dataset is split up roughly into proportions of 87%, 12%, 1% for train, validation, and test-dev sets, respectively. All scene graph annotations on the training and validation sets are publicly available.

The VQA v2 [2, 39] contains 204,721 natural images from COCO and 1,105,904 free-form questions obtained by crowdsourcing. Each question in the dataset is associated with 10 different answers. Accuracy on this dataset (VQA score) is computed so as to be robust to inter-human variability as $acc(a) = min\{\frac{\text{the number of times a is chosen}}{3}, 1\}$. The dataset is split up roughly into proportions of 40%, 20%, 40% for train, validation, and test sets, respectively, and we report the VQA score on the validation split as the experimental results in Section 4.4.

4.1.1 Data Preprocessing

Question and Image features We consider pairs of a question sentence and an image as inputs and the pairs are transformed into symbolic representations as preprocessing steps. As the symbolic representation of each question, a dependency tree is constructed by using the Spacy library. Each token from dependency parsing is mapped into 300-dimensional pre-trained GloVe word embeddings [25] and the dependencies between the tokens are represented by a directed adjacency matrix.

For image modality, scene graphs are used as a symbolic representation. Originally, the scene graph [19] consists of three components, which are the objects (names), their attributes, and the relations between the objects. In terms of graph notations, object names and the attributes are represented by nodes, and relations are annotated at edges between the corresponding nodes. In this paper, to make graph structures of two modalities be equal, all three components are represented by nodes, and the edges have only binary value.

Scene Graph Generation (SGG) The scene graph annotations for images are partially provided for the train and validation split of GQA. For the images of GQA test-dev and all splits of VQA, we generated scene graphs as follows.

Following the works [1], bounding boxes of objects in images are detected by the Faster R-CNN method, and the name and attributes of the objects are predicted based on the ResNet-101 features from the detected bounding boxes. We keep up to 100 objects with a confidence threshold of 0.3 and predict the relations between the objects from the frequency prior knowledge which is constructed from the GQA scene graphs[3].

Answer Vocabulary For the GQA dataset, we extract 1,853 possible answers vocabulary words from the train and validation sets. GQA dataset tightly controls the answer distribution by generating questions using question program. For the VQA task, following previous studies in VQA, we consider the 2,000 most common answers in the training dataset as possible answer vocabulary for our network to predict.

[3]We have been tried to generate a scene graph by using recently suggested SGG algorithms, such as [35, 33, 22]. However, we could not achieve any improvement in the GQA/VQA accuracies. The reasons might be that 1) very small size of vocabularies for object and relation labels are used for the conventional SGG problem setting, 2) the methods do not predict the attributes, and 3) the annotated scene graphs used for training the methods are very sparse.

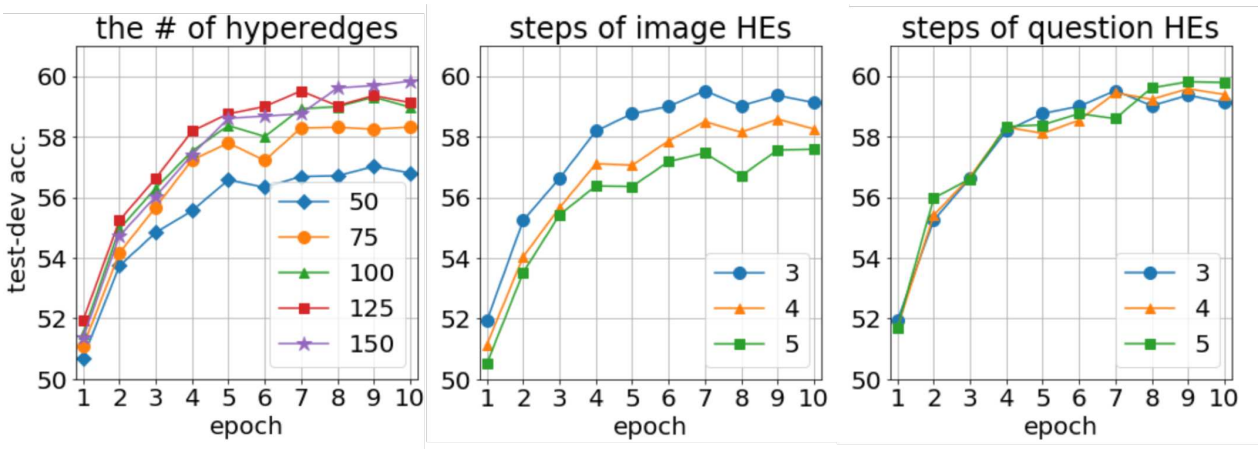


Figure 2. Test-dev accuracies with various hyper-parameter combinations. **Left:** Test-dev accuracy with various number of image hyperedges when the number of hyperedges for questions are fixed to 50. **Middle:** Accuracy with various number of steps (k) of image hyperedges with three-step question hyperedges. **Right:** Accuracy with various number of steps (k) of question hyperedges with three step image hyperedges.

Table 1. As a plug-in module for the attention (Att.), HANs are combined with state-of-the-arts VQA algorithms, BAN [15] and MFB [38]. Those are used as bilinear module B. For the most of the metrics, HANs improves the GQA performance. For the distribution (Dist.) metric, the lower score, the better.

Method					Performance Measures with Test-dev Split					
No.	Feature	HE	Att.	B	Binary	Open	Plaus.	Valid.	Dist.	**Overall Acc.**
1	Symbol	No	No	MFB [38]	60.02	47.24	81.86	95.09	0.74	53.22
2	Symbol	Yes	HAN	MFB [38]	61.70	47.49	81.83	95.02	0.68	54.14
3	Symbol	No	No	BAN [15]	60.27	50.06	82.80	95.94	0.86	54.84
4	Symbol	Yes	HAN	BAN [15]	65.89	58.36	83.39	96.50	0.49	61.88
5	Image	No	No	MAC [10]	71.23	38.91	84.48	96.16	5.34	54.06
6	Image	No	No	BAN [15]	76.00	40.41	85.58	96.16	10.52	57.10
7	Image	No	No	NSM [12]	78.94	49.25	84.28	96.41	3.71	63.17
8	Symbol+Image	Yes+GNN	HAN	BAN [15]	71.87	63.03	82.95	95.79	2.49	69.46

4.2. Implementation details

For the GQA, we firstly project the Y_i, Y_q, and V_i onto 256-dimensional space with a single fully-connected layer, respectively. Then, we use BAN based on a concatenated 8 glimpses setting to get 2048 dimensional feature vectors for $\mathbf{z}_s$ and $\mathbf{z}_v$. After that, for the Symbol+Image experiment described at Table 1, we stack 2 MRN blocks. Each block has two fully connected layers with Batch Normalization [13] and hyperbolic tangent activation functions. After each block, we apply Dropout [26] with 0.2 and 0.5 probabilities, respectively. Finally, one fully-connected layer is used for classification. For training, we use Adam [18] optimizer with initial learning rate 3e-4 and the exponential learning rate scheduler with gamma 0.9. Using these settings, we totally run 30 epochs and report the best result.

For the VQA 2.0, we firstly project the Y_i, Y_q, and V_i onto 1024 dimensional space with a single fully-connected layer, respectively. Then, we use 8 glimpses BAN with residual summations of the glimpses. Thus, we get 1024 dimensional feature vectors $\mathbf{z}_s$ and $\mathbf{z}_v$. Instead of using MRN for fusing the $\mathbf{z}_s$ and $\mathbf{z}_v$, it was enough to concatenate them. Finally, one fully-connected layer is used for classification. For training, we use the Adamax [18] optimizer with initial learning rate 1e-3. We decrease the learning rate by a factor of 4 per 2 epochs after the initial 10 epochs. Using these settings, we totally run 30 epochs and report the best result.

4.3. Quantitative Results on GQA

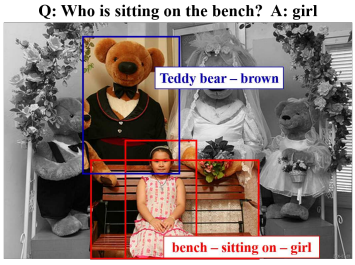
For the GQA evaluation metric, we report top-1 accuracy on the test-dev split. Furthermore, new metrics proposed in [11] such as plausibility, validity, and distribution measures are also applied to complement the accuracy metric[4].

We compare HANs against state-of-the-art methods to evaluate the effectiveness. HANs show consistent improvement over all state-of-the-arts methods previously suggested.

For the bilinear operator B explained in Section 3.4, two

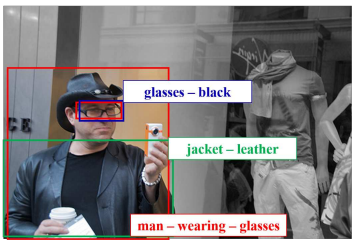
[4]The consistency metric is not used for the metric because the metric highly depends on whether to use all or balanced dataset.

Q: Who is sitting on the bench? A: girl



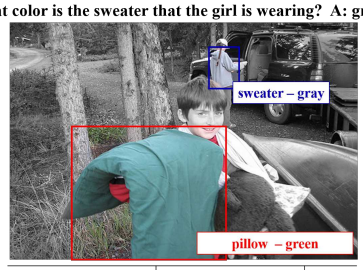
Hyperedge on Question	Hyperedge on Image	Att. Val.
Who – sitting	bench – sitting on – girl	0.9929
sitting on – bench	bench – sitting on – girl	0.7127
on – the bench	Teddy bear – brown	0.4580

Q: Is he wearing glasses? A: Yes



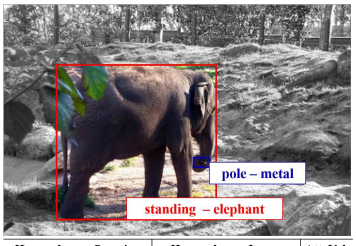
Hyperedge on Question	Hyperedge on Image	Att. Val.
he – wearing	man – wearing – glasses	0.9177
wearing – glasses	glasses – black	0.8668
he – wearing	jacket – leather	0.8422

Q: what color is the sweater that the girl is wearing? A: gray



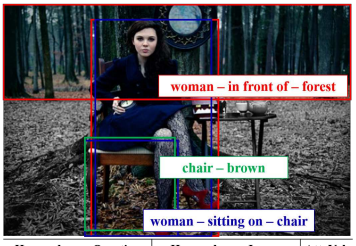
Hyperedge on Question	Hyperedge on Image	Att. Val.
what – color	pillow – green	0.7478
sweater – that – wearing	sweater – gray	0.6568
what – color	sweater – gray	0.6444

Q: What kind of animal holds the stick? A: elephant



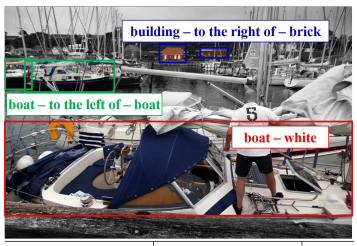
Hyperedge on Question	Hyperedge on Image	Att. Val.
of – animal	standing – elephant	0.9725
hold – the – stick	pole – metal	0.8017
what – kind – holds	pole – metal	0.6573

Q: Who is sitting in the forest? A: woman



Hyperedge on Question	Hyperedge on Image	Att. Val.
who – sitting	woman – in front of – forest	0.8248
who – sitting	woman – sitting in – chair	0.5531
the - forest	chair – brown	0.5224

Q: Do you see either buses or vehicles in this photograph? A: No



Hyperedge on Question	Hyperedge on Image	Att. Val.
buses – vehicles	boat – white	0.7447
Buses - vehiecles	building–to the right of–brick	0.7431
the - forest	chair – brown	0.4954

Figure 3. The visualization for co-attention maps $\mathcal{A}$ of HANs with six examples. Among the all pairs of images and question hyperedges, three hyperedge pairs with top-3 attention-value are presented. The question is shown on the top of the image and the hyperedge pairs are on the bottom. Corresponding regions attended by HANs are represented on the image.

state-of-the-arts methods on VQA tasks are used, which are BAN[15] and MFB [38]. From the comparison between experiments from No.1 to 4 in Table 1, we can argue that the co-attention maps $\mathcal{A}$ learned by HANs are very effective in this task. The reason might be that BAN considers the co-attention for every pair of two modalities (random-walk-based hyperedges in our case) while, MFB fuses the two modalities based on the Hadamard product of two compressed feature vectors followed by projection with low-rank matrices. As the GQA dataset requires a model to capture multiple facts for answering a given question, BAN architecture was more effective than MFB in this problem. For a reason, we think that any pair-wise bilinear attention method will show significant improvement when combined with HANs.

In experiments 5, 6, and 7, we summarized the state-of-the-art accuracy of the GQA dataset. Those methods utilized image features, not using symbolic representations. MAC network [10] is the baseline method, which is suggested by the authors of the GQA dataset. As the benchmark performance, accuracy with BAN [15] is also provided in the leaderboard[5]. One thing that should be noted is that the results in the leaderboard are based on all datasets, but our results in Table 1 only use the balanced set.

[5]https://evalai.cloudcv.org/web/challenges/challenge-page/225/leaderboard/733

Effects of the Symbolic Representations We compare HANs' performance (just use the symbolic graphs) with the conventional VQA methods which use the image features. From the experimental results Table 1, we confirm that symbolic representations are very crucial.

Hyper-parameter Search First of all, we analyze the characteristics of HANs with various hyper-parameter combinations. The test-dev accuracy with varying three parameters, which are the number of hyperedges, the number of random-walk steps (k) of question graphs, and the number of random-walk steps (k) of image graphs, are summarized in Figure 2.

The hyperedges can be thought of as sub-structures of the given graphs. Therefore, HANs with a large number of hyperedges are closer to the exact sub-graph matching problem. From Figure 2, we could check this point with the fact that the more sampling hyperedges, the better test-dev accuracy.

The critical characteristic of HANs is to integrate multimodal symbolic graphs by comparing the semantics of their sub-structures, while most of the existing approaches use the similarity of only node features. We use the random-walk algorithm to approximate sub-structures of the graph. We define a single random-walk path as a hyperedge and a hypergraph as the set of the random-walk paths. By comparing the semantics between two hypergraphs, the sub-graph matching

problem can be efficiently resolved. Here, the semantics of a hyperedge is defined by a simple average function of node feature vectors. It is worthwhile to note that because the random-walk paths are relatively short, the simple average function was enough, and we empirically found that it performs well compared with other options (such as summation, maximization, or more complicated functions).

4.4. Quantitative Results on VQA

In this section, we show the comparative results on VQA v2 dataset. Similar to section 4.2, we show the effectiveness of HANs with BAN. For this experiment, VQA scores on the validation set are reported as the accuracies.

Table 2. VQA scores on validation set for VQA v2 dataset are summarized. Likewise the GQA task, HANs combined with BAN improves VQA performance. Here, we report both reported and reproduced results of BAN to validate the effectiveness of HANs without any additional module.

No.	Use HE	Method	Acc.
9	No	Bottom-Up [1]	63.37
10	No	MFH [38]	64.31
11	No	BAN [15] (reported)	66.04
12	No	BAN [15] (reproduced)	64.85
13	Yes	HAN (ours)	65.05

Here, we note that we reproduced the validation score of BAN based on the official implementation[6] for a fair comparison with our model; data sampling strategy and same initial word embedding vectors. From Table 2, we observed that using hyperedges information extracted from a symbolic level scene graph can improve the VQA performance compared to the reproduced BAN. Importantly, the improvement was achieved without heavy engineering such as data augmentation with Visual Genome dataset [19] and enhancing word embedding [15].

4.5. Qualitative Results

We now visualize some co-attention maps generated by HANs with the GQA dataset in Figure 3. Among the all pairs of M^i and M^q, three pairs of hyperedges with top 3 attention values out of eight-glimpse are presented.

Figure 3 shows the promising results achieved by the proposed method. We highlight the regions according to the image hyperedges with high attention weights. Since questions of GQA dataset are generated by a rule-based question generation program based on a scene graph, it is important for a model to focus on not only objects but also their relationship. For example, at the top-left example of Figure 3, our model successfully focuses on the triplet *bench - sitting on -girl*. Thus, our model can predict the correct answer instead of the *Teddy bear* which is being behind of the *bench*.

[6]https://github.com/jnhwkim/ban-vqa

5. Discussion and Conclusion

We have shown an interesting approach to multimodal learning, which transforms the low-level multimodal inputs into symbolic graph forms and integrates the multiple symbolic graphs with the co-attention maps. To construct the co-attention maps, a novel sub-structure matching method based on the hypergraph structure is suggested.

From the experimental results with the GQA and VQA v2 dataset, we showed that the symbolic graph is a very powerful way to represent the information of the low-level signals. The method to integrate two graphs by matching semantics between sub-structures works well. Also, HANs show a new state-of-the-art performance on the GQA task. Furthermore, we observe that our model can focus on both objects and relations between them by using the trained co-attention map.

Compared to get a dependency tree as the inherent structure of the sentences, to get scene graphs of images might hard. Interestingly, we showed that approximated scene graphs using question sets are powerful.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Stefano Berretti, Alberto Del Bimbo, and Enrico Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1089–1105, 2001.

[5] Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. *arXiv preprint arXiv:1902.09487*, 2019.

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

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[10] Drew Arad Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations*, 2018.

[11] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] Drew A Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *arXiv preprint arXiv:1907.03950*, 2019.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[14] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

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

[16] Jin-Hwa Kim, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Multimodal residual learning for visual qa. In *Advances in neural information processing systems*, pages 361–369, 2016.

[17] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. 2017.

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

[19] 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. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[20] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. *arXiv preprint arXiv:1903.12314*, 2019.

[21] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. *arXiv preprint arXiv:1904.12787*, 2019.

[22] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1261–1270, 2017.

[23] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6087–6096, 2018.

[24] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems*, pages 8334–8343, 2018.

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

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

[27] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[28] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[29] Alexander Trott, Caiming Xiong, and Richard Socher. Interpretable counting for visual question answering. 2017.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[31] Peng Wang, Qi Wu, Chunhua Shen, and Anton van den Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1173–1182, 2017.

[32] Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1367–1381, 2017.

[33] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.

[34] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[35] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

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

[37] Dongfei Yu, Jianlong Fu, Tao Mei, and Yong Rui. Multi-level attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4709–4717, 2017.

[38] Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE transactions on neural networks and learning systems*, (99):1–13, 2018.

[39] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[40] Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. 2018.