Visual-Semantic Matching by Exploring High-Order Attention and Distraction

Yongzhi Li¹, Duo Zhang², Yadong Mu³*
¹Center for Data Science, ²EECS, ³Wangxuan Institute of Computer Technology, Peking University
{yongzhili,zhduodyx,myd}@pku.edu.cn

Abstract

Cross-modality semantic matching is a vital task in computer vision and has attracted increasing attention in recent years. Existing methods mainly explore object-based alignment between image objects and text words. In this work, we address this task from two previously-ignored aspects: high-order semantic information (e.g., object-predicate-subject triplet, object-attribute pair) and visual distraction (i.e., despite the high relevance to textual query, images may also contain many prominent distracting objects or visual relations). Specifically, we build scene graphs for both visual and textual modalities. Our technical contributions are two-folds: firstly, we formulate the visual-semantic matching task as an attention-driven cross-modality scene graph matching problem. Graph convolutional networks (GCNs) are used to extract high-order information from two scene graphs. A novel cross-graph attention mechanism is proposed to contextually reweigh graph elements and calculate the inter-graph similarity; Secondly, some top-ranked samples are indeed false matching due to the co-occurrence of both highly-relevant and distracting information. We devise an information-theoretic measure for estimating semantic distraction and re-ranking the initial retrieval results. Comprehensive experiments and ablation studies on two large public datasets (MS-COCO and Flickr30K) demonstrate the superiority of the proposed method and the effectiveness of both high-order attention and distraction.

1. Introduction

The rapid growth of various multimedia data such as text, images, and videos has brought great difficulties to users in accurate and effective search nowadays. Among them, cross-modal search between vision and language is of great importance in practical applications. Therefore, cross-modal retrieval between images and text has attracted plenty of attention from researchers in recent years [43]. This paper focuses on cross-modal retrieval of text and images with complex semantics. For this task, how to effectively eliminate the huge gap between data of two different modalities is the key to solve the problem.

Thanks to advances in computer vision and natural language processing, some early developments [17, 5] have proposed to use pre-trained neural networks that encode multi-modality data into feature vectors, then project them into a common feature space and measure the similarity by computing the distance between their representations. Although such methods are capable of capturing global semantic information, they lack accurate modeling of high-level semantic information and are not clearly interpretable. Some modern methods have attempted to perform some more granular semantic learning to enhance the feature representation of data. For example, Karpathy et al. [14] aligned textual words and image regions for image captioning. The work in [10, 35] proposed using CNN to extract semantic concepts from pictures to enhance the representation of global features. Kuang et al. [20] introduced an attention mechanism to identify important parts of text and images.

A majority of these methods are based on first-order

![Figure 1. Illustration of the importance of predicate (top row), attributes (middle row), and semantic distraction (bottom row) in visual-semantic matching. From left to right, three columns represent the querying caption, ground-truth images, and the highly-ranked false matching returned by conventional methods, respectively. Key words in queries are highlighted in bold.](image-url)
information like semantic concepts or objects. However, high-order information such as the relationships between objects and object attributes, are rarely explored in current literatures. When facing structural queries, these methods are often frustrated by ambiguous false matching. Some examples are provided in Figure 1. In the first example, even all key objects and scene concepts (man, horse and street) relevant to the querying sentence (“a man riding on a horse on a street”) are precisely detected, one would still be unable to distinguish some confusing samples without considering high-order information in the query (e.g., triplet relation person-ride-horse in this example). We regard object-attribute pair as another kind of informative high-order information. An example is found in the second row of Figure 1, which illustrates the importance of the color attribute.

To address this issue, we adopt scene graphs [13, 12] for representing highly-structural visual or textual semantics, and formulate visual-semantic matching as a heterogeneous graph matching problem. Figure 2 shows two exemplar scene graphs that encapsulate various pairwise or triple relationships, respectively. We propose to use Graph Convolutional Networks (GCNs) [16, 46] to contextualize each individual graph node. Both cross-graph attention and intra-graph self-attention mechanisms are developed to reweight each graph element and calculate the similarity between querying and reference scene graphs.

Additionally, we argue that the issue of semantic distraction still remains unexplored in the previous literatures. In specific, most of existing methods primarily concern the relevance between query and reference samples. However, whether matched semantics dominate a reference image or text is not considered. An example is provided in the third row of Figure 1. The rightmost image is highly-ranked since it contains all key words in the query, yet shall be actually classified as false matching given the co-occurrence of vast distracting contents (e.g., the children, the pedestrians etc.). We are thus inspired to propose an information-theoretic metric to explicitly quantify visual distraction, which is used to re-rank the initial retrieved top matchings.

Our contributions can be summarized as below:

1) We aim to effectively explore high-order information in the visual-semantic matching task, particularly the object-predicate-subject and object-attribute types. Technically, GCNs are incorporated into our model for encoding above-mentioned high-order information. Multiple attention mechanisms are tailored for computing the similarity between querying and reference scene graphs.

2) To our best knowledge, we are the first to explicitly explore the visual distraction problem in structural visual-semantic matching. Informational entropy is novelty adapted to gauge the dominance of distracting factors in a reference image or text.

---

Figure 2. An image-text pair and their corresponding scene graphs. Graphical shapes represent different types of graph nodes.

2. Related Work

**Embedding based methods.** A widely used framework is to map the semantic embedding vectors of image and text into a common space and calculate the similarities according to the cosine or euclidean distance [37, 40, 18, 21, 4, 30, 3, 17, 7]. Kiros et al. [17] first used convolutional neural network (CNN) and recurrent neural network (RNN) to encode image and sentence features, and learned a cross-modal representation from triplet ranking loss. Gu et al. [7] proposed using generative object to enhance the fine-grained feature representations. Zheng et al. [51] suggested using a dual-task to embed the semantic features more discriminatively in the shared space. [39] introduced two-branch embeddings and proposed novel neighborhood constraints.

**Semantic knowledge based methods.** Several modern works explored the alignment of visual objects and textual words, as found in [31, 11, 29, 13]. The pioneering work [14] adopted an R-CNN [6] model to detect local regions from an image and aligned them with words in the sentence. Huang et al. [10] proposed learning semantics and orders to improve the image representations. A context-modulated attention scheme was developed in [9] to selectively attend instances appearing in both the image and sentence. Furthermore, Lee et al. proposed a method in [20], which used stacked cross attention to match two modalities in a finer-grained model. Some other research [38, 35] adopted external knowledge to further enhance the model capability.

**Graph matching.** Similarity-based graph search or matching has been a long-standing research task in a number of communities including data mining [45, 2, 28, 23] and natural language processing [44, 27]. Regarding the domain of computer vision, graph matching has been used for video re-identification [47] and 3D model retrieval [26] etc. With the development of graph convolutional networks (GCNs) [16] in recent years, the authors of [49] proposed a GMN network to align key-points in different images. [41] furthermore suggested an embedding based cross-graph affinity method to model the graph structure.
3. Approach

The image-text (or referred to as visual-semantic) matching problem is defined as follows: given an image-sentence pair, our model aims to calculate a similarity score between them, such that accurate cross-modality search is feasible.\(^1\) As shown in Figure 3, our model utilizes scene graphs to represent visual or textual semantic modalities. The extracted scene graphs first go through two graph convolutional networks (GCNs) to obtain contextualized embeddings. Intuitively, the importance of an object or relation in the reference heavily hinges on its relevance to the query. Inspired by this intuition, a cross-graph attention module is proposed for weighing graph nodes. Then, a multi-perspective local matching layer produces a matching vector for each node in the graphs, and a node attention mechanism is included to get global matching representation. Finally, the fully-connected layers predict the similarity score by taking the global representation as input.

3.1. Text Encoding and Sentence Scene Graph

To acquire the representation of a given sentence \(S\), a bi-directional LSTM [8] is adopted to model the long-range context information. For each word in \(S\), its index first passes an embedding layer to get a word embedding vector, which is then fed into the bi-LSTM to get a temporally-contextual representation \(h_i\). The output of the last step \(h_n\) is used to represent the whole sentence.

\(^1\)There are two settings: image-to-text or text-to-image matching. In each setting, one modality serves as the query and the other plays the role of reference data. Such asymmetry significantly affects the design of optimization objective, as later described in Section 3.7 and experiments.

To obtain a sentence scene graph (SSG), we use a fixed rule-based language parser [1] to translate the inputting sentences into scene graphs. Following the prior practice [46], we feed all the captions of an image into the parser and get a tuple \(G_s = (N,E)\), where \(N\) and \(E\) are the sets of nodes and edges, respectively. We define three kinds of nodes in \(N\): object node \(o\), attribute node \(a\), and relation node \(r\) that are represented as rectangles, diamonds, and ellipses respectively in Figure 2. \(o_i\) represents the \(i_{th}\) object. The relationship between object \(o_i\) and \(o_j\) is denoted as \(r_{ij}\), and \(a_{i,l}\) is the \(l_{th}\) attribute of object \(o_i\). Each node in \(N\) is represented by a \(d\)-dimensional vector, denoted as \(n^o_i, n^a_i,\) or \(n^e_i\), according to the node type. In particular, we use a trainable embedding layer to get the node features. As shown in Figure 2, the edges in \(E\) are formulated as follows:

- If an object \(o_i\) owns an attribute \(a_{i,l}\), we assign a directed edge from \(a_{i,l}\) to \(o_i\);
- A relationship triplet \(< o_i, r_{ij}, o_j >\) exists, two directed edges will be assigned from \(o_i\) to \(r_{ij}\) and from \(r_{ij}\) to \(o_j\), respectively.

3.2. Image Encoding and Image Scene Graph

Given an input image, we first use a Faster-RCNN model [32] pretrained on MSCOCO [24] to get a number of object proposals and corresponding ROI features \(f_{roi}\). Global average pooling is applied on the feature map to get a global image feature \(f_{global}\).

To generate the image scene graph (ISG), we borrow the visual scene graph detector from [50] to predict the relationships between the object proposals. Another classifier pre-trained on Visual Genome [19] is used for predicting
the attributes of each object proposal. Similar to $G_S$, there are also three kinds of nodes in the ISG $G_I$. However, the object nodes $n^o_i$ in $G_I$ take the ROI features as the node features, while the relation nodes $n^r_i$ and attribute nodes $n^a_i$ take the class-label embeddings [46] as the original features.

3.3. Graph Convolutional Networks (GCNs)

To effectively explore the high-order information in $G_S$ and $G_I$, GCNs are adopted to merge the local information from each node and its neighbors into a new set of context-aware embeddings at some $d$-dimensional feature space. We hereafter use variable $x$ (with proper index) to denote such embedding vectors. There are three types of embeddings: relation embedding $x_{r_{ij}}$, object embedding $x_{o_{i}}$ for object node $o_i$, and attribute embedding $x_{a_{i}}$ for attribute node $a_i$, respectively. Inspired by the previous work [46], we propose four spatial graph convolutions: $g_r$, $g_o$, $g_s$, and $g_o$ to generate above-mentioned embeddings. In practice, these four functions are implemented as multilayer perception (MLP) with same network architecture but independent parameters. For brevity, we only elaborate on SSG. The derivation of ISG is similar.

For each $r_{ij}$ in SSG, the relation embedding $x_{r_{ij}}$ is processed according to Eqn. (1) as below to jointly represent a relation from all nodes that have a relationship with $o_i$: $x_{r_{ij}} = g_r \left( \text{concat}(n^o_{a_{i}}, n^a_{o_{i}}, n^o_{r_{ij}}) \right)$. (1)

For the attribute nodes, the spatial convolution operation is depicted in blue color in Figure 2. The information of all attribute nodes connected to each object node $o_i$ is merged to get a single contextual feature vector:

$$ x_{a_{i}} = \frac{1}{N_{a_{i}}} \sum_{l=1}^{N_{a_{i}}} g_o \left( \text{concat}(n^o_{a_{i}}, n^a_{a_{i}}) \right) , $$ (2)

where $\{a_{i,l}\}$ forms $o_i$’s attribute-related neighbor set with a cardinality of $N_{a_{i}}$.

To compute object embedding $x_{o_{i}}$, we collect information from all nodes that have a relationship with $o_i$, according to:

$$ x_{o_{i}} = \frac{1}{N_{r_{i}}} \sum_{o_j \in s b j(o_{i})} g_a(\text{concat}(n^o_{o_{i}}, n^a_{o_{i}}, n^o_{r_{ij}})) + \sum_{o_k \in o b j(o_{i})} g_o(\text{concat}(n^o_{a_{k}}, n^a_{a_{k}}, n^o_{r_{kj}})), $$ (3)

where $o_j \in s b j(o_{i})$ implies that $o_j$ acts as the subject in some relation with the object $o_i$, $o_k \in o b j(o_{i})$ implies a role of object. And $N_{r_{i}} = |s b j(o_{i})| + |o b j(o_{i})|$. This operation is shown in yellow color in Figure 2.

3.4. Cross-Graph Attention

As stated before, it is crucial to calculate the asymmetric correlation between any two cross-graph nodes, either from $G_S$ to $G_I$ or from $G_I$ to $G_S$. We design a cross heterogeneous graph attention mechanism, shown in Figure 3. For each node $x^S_i$ in $G_S$, we calculate the cosine similarities with all nodes $x^I_j$ in $G_I$ to get an attention matrix $M_{att}$ sized $|G_S| \times |G_I|$. Specifically, $M_{att} = [u_{i,j}]$ is calculated as below:

$$ u_{i,j} = \text{cosine}(x^S_i, x^I_j), \quad i \in 1, \ldots, |G_S|, j \in 1, \ldots, |G_I|, $$ (4)

where cosine$(\cdot, \cdot)$ returns the cosine value in $[-1, 1]$ for its two inputing vectors.

These similarities are then used as weights to compute the attentive embedding for $x^S_i$ by a weighted sum on all the node embeddings of $G_I$, namely

$$ \hat{x}^S_i = \sum_{j=1}^{|G_I|} \frac{u_{i,j} \cdot x^I_j}{\sum_{j=1}^{|G_I|} u_{i,j}}, $$ (5)

where $[x]+ = \max(x, 0)$. We can switch the role of $G_I, G_S$ in above procedure, obtaining attentive embedding $\hat{x}^I_i$ for each $x^I_i$ in $G_I$.

3.5. Local Graph Matching

Based on the attentive embedding of each node, we proceed to calculate local-matching vectors for all nodes in both $G_S, G_I$. The goal is to obtain intra-graph contextually-enhanced features from multiple perspective (i.e., semantics and graph structure). For $G_S$, a multi-perspective cosine matching function is applied on each node $x^S_i$ and its attentive embedding $\hat{x}^S_i$. Similar treatment for $G_I$. In particular, local-matching vectors $m^S_i$ and $m^I_j$ are computed as below:

$$ m^S_i = f_m(x^S_i, \hat{x}^S_i; W), \quad m^I_j = f_m(x^I_j, \hat{x}^I_j; W), $$ (6)

where $f_m$ denotes the multi-perspective matching function. Let $W \in \mathbb{R}^{l \times d}$ be a learnable parameter matrix and $W_k$ be the $k$-th row of $W$. $l$ is pre-defined to specify how many perspectives are required. Given two $d$-dimensional vectors $v_1$ and $v_2$, the matching vector $m$ is rendered as below:

$$ m_k = \text{cosine}(W_k \circ v_1, W_k \circ v_2), \quad k \in [1, 2, \ldots, l], $$

$$ m = |m_1, m_2, \ldots, m_l|, $$ (7)

where $\circ$ denotes element-wise product. This implicitly defines $f_m$.

3.6. Node Attention and Global Matching

Intuitively, different nodes in graphs shall not be equally weighed. Some nodes are more important according to common sense (e.g., human-related objects are often more
node attention mechanism to attach a weight likely to be relevant to queries. We thus further design a node attention mechanism to attach a weight \( \beta_i \) to each node \( x_i \) in graphs. Take SSG for instance,

\[
\beta_i^S = \frac{\exp(\phi(h_n, x_i^S))}{\sum_{i=1}^{\left|G_s\right|} \exp(\phi(h_n, x_i^S))},
\]

where \( x_i^S \) is the context embedding in \( G_S \) as described in Equations (1)(2)(3). \( \phi \) is a learnable sub-network that reads \( h_n, x_i^S \). Recall that \( h_n \) is a global feature vector for the entire sentence. Replacing \( h_n \) by \( f_{global} \), \( x_i^S \) by \( x_i^I \), and \( \left|G_s\right| \) by \( \left|G_I\right| \) in Eqn. (8) leads to a new formula for ISG.

After getting the importance of each node in the graph, a global weighted sum is adopted to fuse all the matching vectors into a global representation:

\[
\bar{m}_S = \sum_{i=1}^{\left|G_S\right|} \beta_i^S m_i^S, \quad \bar{m}_I = \sum_{j=1}^{\left|G_I\right|} \beta_j^I m_j^I.
\]

Finally, \( \bar{m}_S, \bar{m}_I \) are concatenated together and fed into a MLP followed by a sigmoid function to predict the matching similarity, denoted by \( \text{score}_m \).

### 3.7. Distraction Based Re-Ranking

We regard that a good match should simultaneously satisfy two conditions: 1) the matched reference contains all key semantics of the query (i.e., maximal relevance); 2) the contents in the reference irrelevant to the query should not be dominating, since they distract viewers (i.e., minimal distraction). We argue that the second condition is insufficiently explored in previous studies. An example is presented in Figure 4.

This work adopts information entropy to quantify distraction and uses distraction scores to re-rank initial retrieved results. Take text-to-image matching for instance. The computations is purely based on the attention matrix \( M_{att} \) described in Section 3.4. For text-to-image matching, we estimate the distraction by asking each node in \( G_S \) to vote for each node in \( G_I \). To ensure votes from each node in \( G_S \) are equal, we first perform \( L_1 \) normalization on the row corresponding to specific node in \( G_S \). Namely \( M_{att}(k, \cdot) \leftarrow M_{att}(k, \cdot) / \sum_j M_{att}(k, j) \). Next, by column sum we obtain how many votes each node in \( G_I \) receives from all nodes in \( G_S \), termed as distraction vector \( v_{dist} \in \mathbb{R}^{\left|G_I\right|} \). It is formally computed via \( v_{dist}(j) = \sum_i M_{att}(i, j) \). We conduct \( L_1 \) normalization to ensure \( v_{dist} \) forms a valid probabilistic distribution. Finally, the distraction score, denoted by \( \text{score}_d \), is computed by information entropy:

\[
\text{score}_d = - \sum_{i=1}^{\left|G_I\right|} (v_{dist,i} + \epsilon) \cdot \log (v_{dist,i} + \epsilon),
\]

where \( \epsilon \) is a tiny constant introduced for numeric stability.

For false matches, \( v_{dist} \) tends to be sparse, with most zeros attribute to the distracting objects or relations. It is known that sparse distribution leads to smaller information entropy.

In practice, we let \( \text{score}_f = \text{score}_m + \gamma \cdot \text{score}_d \) be a fused score for re-ranking initial results. \( \gamma \) is set to \( 4 \times 10^{-3} \) in our experiments. For efficiency, we only calculate the distraction scores for top-10 results in the inference time. Similar derivation for image-to-text matching is omitted due to space limit.

### 3.8. Loss Function of Joint Learning

For image-to-text or text-to-image matching, we use a composite loss function with two components. One is the triplet loss \( L_t \) [34] to maximize the margin of positive sample and negative samples. The other binary cross entropy loss \( L_{ce} \) aims to effectively decrease the negative samples’ scores. Definitions are:

\[
L_t = \sum_i [\text{score}_{m,i}' - \text{score}_{m,i} + \delta]_+, \quad L_{ce} = \sum_i y_i \log (\text{score}_{m,i}) + (1 - y_i) \log (1 - \text{score}_{m,i}),
\]

where the \( \delta \) is a margin hyper-parameter. \( \text{score}_{m,i}' \) denotes the \( i \)-th negative sample’s score and it’s label \( y_i \) is set to 0, while \( \text{score}_{m,i} \) is the positive sample score and it’s label is set to 1. The entire loss is \( \lambda_1 L_t + \lambda_2 L_{ce} \), where \( \lambda_1 \) and \( \lambda_2 \) are two hyper-parameters.

### 4. Experiments

#### 4.1. Datasets

For experiments, following the previous work [20], we selected two large widely-used datasets:

**MS-COCO** [24] is a large-scale dataset which contains 123,287 images, and each image in it is annotated with five text descriptions. We follow [14] to prepare the training,
### Table 1. Results on MS-COCO 1K and 5K. The sentence retrieval and image retrieval utilize images and sentences as queries respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MS-COCO 1K</th>
<th>MS-COCO 5K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sentence Retrieval</td>
<td>Image Retrieval</td>
</tr>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
</tr>
<tr>
<td>DVSA [14]</td>
<td>38.4</td>
<td>69.9</td>
</tr>
<tr>
<td>VQA-ICR [25]</td>
<td>50.5</td>
<td>80.1</td>
</tr>
<tr>
<td>DSPE [40]</td>
<td>50.1</td>
<td>79.7</td>
</tr>
<tr>
<td>VSE++ [4]</td>
<td>64.6</td>
<td>90.0</td>
</tr>
<tr>
<td>TBNN [39]</td>
<td>54.0</td>
<td>84.0</td>
</tr>
<tr>
<td>DPC [51]</td>
<td>65.6</td>
<td>89.8</td>
</tr>
<tr>
<td>DXN [7]</td>
<td>68.5</td>
<td>-</td>
</tr>
<tr>
<td>SCo [10]</td>
<td>69.9</td>
<td>92.9</td>
</tr>
<tr>
<td>SCAN [20]</td>
<td>72.7</td>
<td>94.8</td>
</tr>
<tr>
<td>SAEM [42]</td>
<td>71.2</td>
<td>94.1</td>
</tr>
<tr>
<td>VSRN [22]</td>
<td>76.2</td>
<td>94.8</td>
</tr>
<tr>
<td>Ours</td>
<td>77.0</td>
<td>96.1</td>
</tr>
<tr>
<td>+Dist</td>
<td>77.8</td>
<td>96.1</td>
</tr>
</tbody>
</table>

Validation and testing set. All images are split into three parts which contain 113,287, 5,000, 5,000 samples respectively. For the evaluation of MS-COCO 5K setting, we used all these 5K testing images. We also used 1/5 of these testing samples as MS-COCO 1K setting for ablation study and compare with some algorithms which report their results only on MS-COCO 1K dataset.

### 4.2. Evaluation Metrics

As a common practice in information retrieval, we measure the performance of sentence retrieval (image query) and image retrieval (sentence query) by recall at top-K (R@K), which is defined as the fraction of queries for which the correct item is retrieved in the closest K points to the query. Take the image retrieval as an example, given a query sentence, we rank the similarity scores of all the images in the test set and select top-K candidates. We regard the query as a "successful" one if the ground-truth image is in these candidates. And the R@K is the proportion of these successful queries in the whole test set. K is set to 1, 5 and 10 in all experiments.

### 4.3. Implementation Details

In all of our experiments, the hidden units of the LSTM are set to 1024 to encode the text semantic information. VGG16 [36] pre-trained on ImageNet [33] is used as the image feature extractor for the Faster-RCNN [32] and to get the global image feature. The dimension of all node embeddings in the scene graphs is set to 256. The perspective number l in Eqn. (7) is set to 64.

Since the visual scene graph detector may introduce some noise in the high-order information, we further add a first-order branch. Specifically, we remove the GCN module, and pool the cross attention matrix produced by the context vectors $h_i$ and ROI features $f_{ roi}$ in a LogSum-Exp [20] manner to obtain a first-order part $score_{1, m}^1$. This is fused with the high-order $score_{2, m}$ which is detailed in previous section to get the final matching similarity $score_{m} = score_{2, m} + 0.1 score_{1, m}^1$.

We implement all models using the Pytorch framework. For the loss function, we set the hyper-parameters $\lambda_1 = 1.0$, $\lambda_2 = 0.5$, and $\delta = 1.0$. For each positive sample in the triplet loss, ten randomly selected samples are used to form the negative pairs. Adam optimizer [15] with default setting (learning rate=$10^{-3}$, momentum=0.9, weight-decay=$10^{-4}$) is applied to tune the model parameters. The learning rate attenuates by 1/10 for every 25 epochs. We adopt early stop strategy to avoid over-fitting.

### 4.4. Comparison with State-of-the-Art Models

In this section, we first present our quantitative results and comparison with other state-of-the-art methods on the MS-COCO dataset in Table 1. To make a more comprehensive comparison, we used two scales test sets (1K and 5K) on MS-COCO. It can be seen that our model exceeds the current best method VSRN [22] by most of the metrics on both image retrieval and sentence retrieval tasks. In particular, in the 1K setting, compared to the previous method our model has a significant improvement in the image retrieval scenario (by 2.3 on R@1 and 3.4 on R@5).

We also conducted the same experiments on the Flickr30K dataset, Table 2 shows the comparison results on the testing set. Obviously, we can find that our method still dominates other methods under most of the evaluation metrics, which strongly indicates the superiority of the proposed method.
Figure 5. Visualization the cross graph attention mechanism. Each row is an example, and the column (A) shows the query sentences. Part of object proposals are shown in column (B). And columns (C), (D), and (E) show the attention results for the object, relation, and attribute nodes, respectively. The color of the mask reflects the attention value of the area corresponding to the node. The warmer red color represents greater attentive response. Best viewed in color.

Table 2. Experimental results on Flickr30K.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Sentence Retrieval</th>
<th>Image Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
</tr>
<tr>
<td>DVSA</td>
<td>22.2</td>
<td>48.2</td>
</tr>
<tr>
<td>VQA-ICR</td>
<td>33.9</td>
<td>62.5</td>
</tr>
<tr>
<td>DSPE</td>
<td>40.3</td>
<td>68.9</td>
</tr>
<tr>
<td>VSE++</td>
<td>41.3</td>
<td>69.0</td>
</tr>
<tr>
<td>TBNN</td>
<td>37.5</td>
<td>64.7</td>
</tr>
<tr>
<td>DPC</td>
<td>55.6</td>
<td>81.9</td>
</tr>
<tr>
<td>DXN</td>
<td>56.8</td>
<td>-</td>
</tr>
<tr>
<td>SCO</td>
<td>55.5</td>
<td>82.0</td>
</tr>
<tr>
<td>SCAN</td>
<td>67.4</td>
<td>90.3</td>
</tr>
<tr>
<td>SAEM</td>
<td>69.1</td>
<td>91.0</td>
</tr>
<tr>
<td>VSRN</td>
<td>71.3</td>
<td>90.6</td>
</tr>
<tr>
<td>Ours</td>
<td>70.8</td>
<td>92.7</td>
</tr>
<tr>
<td>+Dist</td>
<td>70.8</td>
<td>92.7</td>
</tr>
</tbody>
</table>

Table 3. Ablation studies on the MS-COCO 1K test set.

4.5. Ablation Studies

To explore the effect of high-order information (relationships and attributes) on visual semantic matching performance, we performed related ablation experiments. The quantitative results on MS-COCO 1K test set are shown in Table 3. First, we removed the relation and attribute information in the data, only used the object information when training and testing. This is denoted by ‘o’ in Table 3. On the basis of this, we added the relation information to the data, and the result is recorded as ‘or’. Finally, we contained the complete object, relation and attribute information and showed the result on the row noted as ‘ora’. It can be clearly noticed that with the addition of more information, the performance of the model has steadily improved in each indicator, which confirms the validity of the relation and attribute information in the visual semantic matching task.

We also explored the impact of the two attention mechanisms proposed above. We removed the node attention (noted as \(-\text{nodeatt}\)) and the cross graph attention (noted as \(-\text{crossatt}\)) separately and used mean pooling instead. The bottom part of Table 3 shows the results. It can be seen that the performance of the model has dropped signifi-
small cars move to pass around a London bus

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a motorbike sitting in front of a wine display case</strong></td>
<td><strong>a motorbike sitting in front of a wine display case</strong></td>
<td><strong>a dirt bike sitting on a counter by some wine bottles</strong></td>
<td><strong>a black and white parked motorcycle and some bicycles</strong></td>
<td><strong>a bike sits among bottles of wine on a shelf</strong></td>
</tr>
</tbody>
</table>

1 | 2 | 3 | 4 | 5 | 6
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>two small children standing at a sink brushing their teeth</strong></td>
<td><strong>two small children standing at a sink brushing their teeth</strong></td>
<td><strong>the wooden bow of a ship with an out of focus boat in the back ground</strong></td>
<td><strong>a group of people sitting on the backs of elephants</strong></td>
<td><strong>several elephants wearing decorations walking in a line</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a group of people sitting on the backs of elephants</strong></td>
<td><strong>three elephants with three women riding on top of them</strong></td>
<td><strong>three elephants wearing decorations walking in a line</strong></td>
<td><strong>elephants and their riders performing at a circus</strong></td>
<td><strong>three elephants standing on a stool with woman sitting on</strong></td>
</tr>
</tbody>
</table>

Figure 6. Demonstrate the effect of distraction based re-ranking. The upper part shows the image retrieval results while the bottom part shows the sentence retrieval results. The original ranking is shown in the upper right corner of each image and behind the sentences respectively. Green boxes indicate ground truth images and the red sentences represent negative samples.


ty without the attention mechanism. For example, without the cross graph attention, the R@1 result of the image retrieval is reduced by 17.6. After removing the node attention mechanism, the performance on the sentence retrieval is attenuated by 26.4% relatively on R@1. This illustrates the essentiality of the proposed attention mechanisms.

### 4.6. Visualization and Analysis

To further demonstrate the interpretability of our model, we selected several examples in Figure 5 to visualize the cross graph attention components learned in our model. Given an attention matrix, we first applied a binary mask to get three sub-attention matrices for object, relation, and attribute nodes. Similar to the operation in Section 3.7, we first normalized the contribution of each image node on row, and then got the attention value of each image node by column summation. We assigned this value to the region corresponding to this node to get a colorful mask. For the object and attribute node, their corresponding region is the area of object proposal. The corresponding area of the relation node is the union of two related object areas. For the overlapping part, we take the maximum value. The warmer red color in the mask reflects larger attention response value.

In column (A) of Figure 5, we first showed the three kinds of nodes in SSG in bold fonts of different colors. Column (B) shows some of the object proposals extracted by Faster-RCNN. The next three columns show the attention effects of the object, relation, and attribute nodes, respectively. From the results, we can see that our model accurately detects the areas corresponding to the nodes in image that aligned to the sentence nodes. Take the result in the first line as an example, the warm color in 1 (C) reflects the object information of tomatoes, olives, and plate. The red area in 4 (D) contains the information of “ride” and “near” mentioned in the sentence, which shows that our network can effectively extract relevant relation information. For the attribute nodes, the regions with highest attention value in 3 (E) and 4 (E) indicate the “black” attribute of dog and shirt. These examples strongly illustrate that our cross-attention module has learned interpretable alignments between sentence nodes in $G_S$ and image nodes in $G_I$.

### 4.7. Distraction Based Re-ranking

To demonstrate the effectiveness of our semantic distracting based re-ranking strategy, we showed several results corrected by the post-processing in Figure 6. Due to the limited space, only the top-3 images and top-5 sentences are presented, the original retrieval ranking is also provided. As shown, the distraction score effectively lowered the ranking of false matching samples. Take the first one for example, in the original top-1, the queried car only accounts for a tiny part of the picture, while the irrelevant bus station and buildings are dominant. In the second sample, the mentioned dolls appears in the top false matching but a storybook accounts for a larger part. As for the sentence retrieval examples, the “bicycles” and “decorations” do not appear in the query images, so the ranking of negative sentences are lowered.

### 5. Conclusion

We explored and confirmed the importance of higher-order information (relationships and attributes) and distraction-based re-ranking in the visual semantic matching task. Ablation and visualization experiments both confirmed the rationality and interpretability of our model design. **Acknowledgement:** This work is supported by National Key R&D Program of China (2018AAA0100702), Beijing Natural Science Foundation (Z190001) and National Natural Science Foundation of China (61772037).
References


[22] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In ICCV, 2019. 6


[27] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In EMNLP, 2017. 2


[38] Huan Wang, Song Liu, and Liang-Tien Chia. Does ontology help in image retrieval?: a comparison between keyword, text ontology and multi-modality ontology approaches. In ACM Multimedia, 2006. 2


[40] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In CVPR, 2016. 2, 6


[45] Xifeng Yan, Philip S Yu, and Jiawei Han. Substructure similarity search in graph databases. In SIGMOD, 2005. 2


