# Graph Structured Network for Image-Text Matching

Chunxiao Liu[1,2], Zhendong Mao[3,*], Tianzhu Zhang[3], Hongtao Xie[3], Bin Wang[4], Yongdong Zhang[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]University of Science and Technology of China, Hefei, China [4]Xiaomi AI Lab, Beijing, China
liuchunxiao@iie.ac.cn, maozhendong2008@gmail.com,
{tzzhang, htxie, zhyd73}@ustc.edu.cn, wangbin11@xiaomi.com

## Abstract

*Image-text matching has received growing interest since it bridges vision and language. The key challenge lies in how to learn correspondence between image and text. Existing works learn coarse correspondence based on object co-occurrence statistics, while failing to learn fine-grained phrase correspondence. In this paper, we present a novel Graph Structured Matching Network (GSMN) to learn fine-grained correspondence. The GSMN explicitly models object, relation and attribute as a structured phrase, which not only allows to learn correspondence of object, relation and attribute separately, but also benefits to learn fine-grained correspondence of structured phrase. This is achieved by node-level matching and structure-level matching. The node-level matching associates each node with its relevant nodes from another modality, where the node can be object, relation or attribute. The associated nodes then jointly infer fine-grained correspondence by fusing neighborhood associations at structure-level matching. Comprehensive experiments show that GSMN outperforms state-of-the-art methods on benchmarks, with relative Recall@1 improvements of nearly 7% and 2% on Flickr30K and MSCOCO, respectively. Code will be released at:* https://github.com/CrossmodalGroup/GSMN.

## 1. Introduction

Image-text matching is an emerging task that matches instance from one modality with instance from another modality. This enables to bridge vision and language, which has potential to improve the performance of other multi-modal applications. The key challenge in image-text matching lies in learning correspondence of image and text, such that can reflect similarity of image-text pairs accurately.

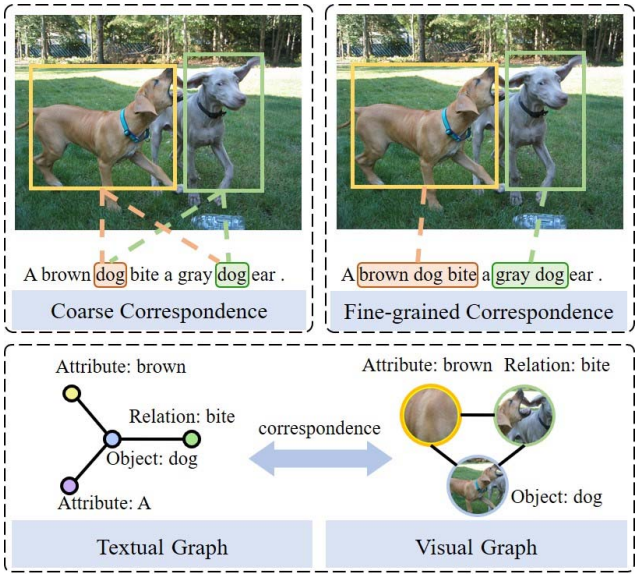


Figure 1: Illustration of coarse and fine-grained correspondence. In the left figure, the two dogs are coarsely correlated with the word *"dog"*, while neglecting their relation and attribute (bite or being bitten? gray or brown?). In the right figure, the gray and brown dogs are fine-grained correlated with finer textual details, which is achieved by learning phrase correspondence using a graph-based method.

Existing approaches either focus on learning global correspondence or local region-word correspondence. The general framework of global correspondence learning methods is to jointly project the whole image and text into a common latent space, where corresponding image and text can be unified into similar representations. Techniques to common space projection range from designing specific networks [23] to adding constraints, such as triplet loss [29], adversarial loss [27] and classification loss [15]. Another branch of image-text matching learns local region-word correspondence, which is used to infer the global similarity of

*Zhendong Mao is the corresponding author.

image-text pairs. Some researchers focus on learning local correspondence between salient regions and keywords. For example, Ji et al. [10] present to correlate words with partial salient regions detected by a lightweight saliency model, which demands external saliency dataset as a supervision. Recent works discover all possible region-word correspondences. For instance, Lee et al. [14] propose to correlate each word with all the regions with different weights, and vice versa. Following this work, wang et al. [30] integrate positional embedding to guide the correspondence learning and Liu et al. [18] present to eliminate partial irrelevant words and regions in correspondence learning.

However, existing works only learn coarse correspondence based on object co-occurrence statistics, while failing to learn fine-grained correspondence of structured object, relation and attribute. As a result, they suffer from two limitations: (1) it is hard to learn correspondences of the relation and attribute as they are overwhelmed by object correspondence. (2) objects are prone to correspond to wrong categories without the guidance of descriptive relation and attribute. As shown in Figure 1, the coarse correspondence will incorrectly correlate the word *"dog"* with all the dogs in the image, while neglecting dogs are with finer details, i.e. brown or gray. By contrast, the fine-grained correspondence explicitly models the object *"dog"*, relation *"bite"* and attribute *"brown"* as a phrase. Therefore, the relation *"bite"* and attribute *"brown"* can also correlate to a specific region, and they will further promote identifying fine-grained phrase *"brown dog bite"*.

To learn fine-grained correspondence, we propose a Graph Structured Matching Network (GSMN) that explicitly models object, relation and attribute as a phrase, and jointly infer fine-grained correspondence by performing matching on these localized phrases. This unions the correspondence learning of object, relation and attribute in a mutually enforced way. On the one hand, relation correspondence and attribute correspondence can guide the fine-grained object correspondence learning. On the other hand, the fine-grained object correspondence forces the network to learn relation correspondence and attribute correspondence explicitly. Concretely, the proposed network constructs graph for image and text, respectively. The graph node consists of the object, relation and attribute, the graph edge exists if any two nodes interact with each other (e.g. the node of an object will connect with the node of its relations or attributes). Then we perform node-level and structure-level matching on both visual and textual graphs. The node-level matching associates each node with nodes from another modality differentially, which are then propagated to neighborhoods at structure-level matching. The phrase correspondence can be inferred with the guidance of node correspondence. Moreover, the correspondence of object node can be updated as long as its neighboring relation and attribute point to a same object. At last, the updated correspondence is used for predicting the global similarity of image-text pairs, which jointly considers correspondence of all the individual phrases.

The main contributions of this paper are summarized as: (1) We propose a Graph Structured Matching Network that explicitly constructs the graph structure for image and text, and performs matching by learning fine-grained phrase correspondence. To the best of our knowledge, this is the first framework that performs image-text matching on heterogeneous visual and textual graphs. (2) To the best of our knowledge, this is the first work that uses graph convolutional layer to propagate node correspondence, and uses it to infer fine-grained phrase correspondence. (3) We conduct extensive experiments on Flickr30K and MSCOCO, showing our superiority over state-of-the-arts.

## 2. Related Work

Existing works learn correspondence of image and text based on object co-occurence, which is roughly categorized into two types: global correspondence and local correspondence learning methods, where the former learns the correspondence between the whole image and sentence, and the latter learns that between local region and word.

The main goal of global correspondence learning methods [19, 4, 29, 29, 31, 22, 34, 1] is to maximize similarity of matched image-text pairs. A main line of research on this field is to first represent image and text as feature vectors, and then project them into a common space optimized by a ranking loss. Some works focus on designing specific networks. For instance, Liu et al. [19] propose to densely correlate image and text exploiting residual blocks. Gu et al. [4] imagine what the matched instance should look like, and improve the correspondence of target instance to this imagined instance. Some works focus on optimization, Wang et al. [29] point out that the correspondence within the same modality should also be preserved while learning correspondence in different modalities. Based on this observation, Wu et al. [31] preserve graph structure among neighborhood images or texts. Such global correspondence learning methods cannot learn correspondence of image and text accurately, because primary objects play the dominant role in the global representation of image-text pairs while secondary objects are mostly ignored.

The local correspondence learning methods learn region-word correspondence. Some works focus on learning correspondence of salient objects. Karparthy et al. [12] make the first attempt by optimizing correspondence of the most similar region-word pairs. Huang et al. [9] present to order semantic concepts and composite them to infer correspondence. Similarly, Huang et al. [8] propose to recurrently select corresponding region-word pairs. Ji et al. [10] exploit saliency model to localize salient regions,

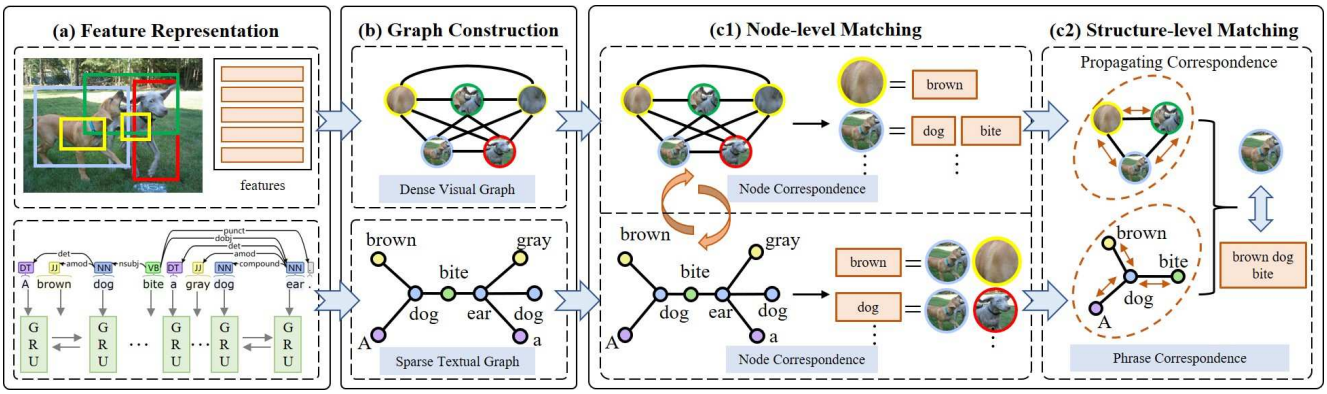


Figure 2: An overview of our approach, which consists of three modules: (a) Feature Extraction: Faster-RCNN [26] and Stanford CoreNLP [21] are employed to detect salient regions, and parse the semantic dependency, respectively. (b) Graph Construction: The node of graph is object, relation or attribute, the edge exists if any two nodes are semantically dependent. (c1) Node-level Matching: learn correspondence of object, relation and attribute separately. (c2) Structure-level Matching: Propagating the learned correspondence to neighbors to jointly infer fine-grained phrase correspondence.

and hence the region-word can be correlated more accurately. A lightweight saliency model is employed using an external saliency dataset as a supervision. The local correspondence policy has also been widely used in other fields [32, 17], like [17] that learns distinction and connection among multi-tasks. Another branch of researches [18, 14, 7, 20, 30] present to discover all possible region-word correspondence. Ma et al. [20] present to jointly map global image and text, local regions and words into a common space, which can implicitly learn region-word correspondence. A recent approach SCAN [14] greatly improves the matching performance, which is most relevant to our work. They learn region-word correspondence using attention mechanism, where each region corresponds to multiple words and vice versa. These works learn correspondence based on object co-occurrence, and have achieved much progress in image-text matching. Nonetheless, these only learn coarse correspondence since they mostly rely on correspondence of salient objects, while neglecting the correspondence of relation and attribute is as important as object correspondence. Moreover, the correspondence of relation and attribute can benefit object to correspond to a specific type with a finer detail. By contrast, we explicitly model the image and text as graph structures, and learn fine-grained phrase correspondence. Instead of transforming the image and text as scene graphs using rule-based [33, 11] or classifier-based [28, 6] methods, we only need to identify whether nodes are interact with each other, which avoids the loss of information caused by scene graph generation.

## 3. Method

The overview of our proposed network is illustrated in Figure 2. We first extract features of image and text, and then construct visual and textual graph. Next, the node-level matching learns node correspondence, and propagate to neighbors in structure-level matching, in which the correspondences of object, relation and attribute are fused to infer the fine-grained phrase correspondence.

### 3.1. Graph Construction

**Textual Graph.** Formally, we seek to construct an undirected sparse graph $G_1 = (V_1, E_1)$ for each text, we use matrix $A$ to represent the adjacent matrix of each node, and add self-loops. The edge weight is denoted as a matrix $W_e$, which shows the semantic dependency of nodes.

To construct the textual graph, we first identify the semantic dependency within the text using off-the-shelf Stanford CoreNLP [21]. This can not only parse the object (nouns), relation (verbs) and attribute (adjectives or quantifiers) in a sentence, but also parse their semantic dependencies. For example, given a text *"A brown dog bite a gray dog ear"*, *"A"*, *"brown"* are attributes for the first object *"dog"*, and the *"bite"* is its relation. They are semantically dependent since all of them describe the same object. Based on this observation, we set each word as the graph node, and there exists graph edge between nodes if they are semantically dependent. Then we compute the similarity matrix $S$ of word representations $u$ as

$$s_{ij} = \frac{\exp(\lambda u_i^T u_j)}{\sum_{j=0}^{m} \exp(\lambda u_i^T u_j)}. \tag{1}$$

where the $s_{ij}$ indicates the similarity between $i$-th and $j$-th node. $\lambda$ is a scaling factor. The weight matrix $W_e$ can be obtained by a Hadamard product between similarity matrix and adjacent matrix, followed by $L_2$ normalization, i.e.

$$W_e = \|S \circ A\|_2 . \tag{2}$$

Additionally, we also implement the textual graph as a fully-connected graph. In contrast to sparse graph that employs semantic dependency of words, it can exploit implicit dependencies. We find the sparse and dense graphs are complementary to each other, and can greatly improve the performance, see section 4.2.1.

**Visual Graph.** To construct the visual graph $G_2 = (V_2, E_2)$, we represent each image as an undirected fully-connected graph, where the node is set as salient regions detected by Faster-RCNN [26], and each node is associated with all the other nodes. Inspired by [24] in visual question answering, we use the polar coordinate to model the spatial relation of each image, which disentangles the orientation and distance of pair-wise regions. This can capture both semantic and spatial relationships among different regions, since the relation and attribute are expected to close to object, and the direction information allows to estimate the type of relations. For example, the relations *"on"* and *"under"* show opposite relative position to the object *"desk"*. To get edge weight for this fully-connected graph, we compute polar coordinate $(\rho, \theta)$ based on the centres of the bounding boxes of pair-wise regions, and set the edge weight matrix $W_e$ as pair-wise polar coordinates.

## 3.2. Multimodal Graph Matching

Given a textual graph $G_1 = (V_1, E_1)$ of a text, and a visual graph $G_2 = (V_2, E_2)$ of an image, our goal is to match two graphs to learn fine-grained correspondence, producing similarity $g(G_1, G_2)$ as global similarity of an image-text pair. We define the node representation of textual graph as $U_\alpha \in \mathbb{R}^{m\times d}$, and the node representation of visual graph as $V_\beta \in \mathbb{R}^{n\times d}$. Here, $m$ and $n$ denotes the node number of textual and visual graph, $d$ is the representation dimension. To compute the similarity of these heterogeneous graphs, we first perform node-level matching to associate each node with nodes from another modality graph, i.e. learning node correspondence, and then perform structure-level matching i.e. learning phrase correspondence, by propagating associated nodes to neighbors, which jointly infer fine-grained correspondence of structured object, relation and attribute.

### 3.2.1 Node-level Matching

Each node in the textual and visual graphs will match with nodes from another modality graph to learn node correspondence. We first depict the node-level matching on textual graph in details, and then roughly describe that on visual graph since this operation is symmetric on two kinds of graphs. Concretely, we first compute similarities between visual and textual nodes, denoted as $U_\alpha V_\beta^T$, followed by a softmax function along the visual axis. The similarity value measures how the visual node corresponds to each textual node. Then, we aggregate all the visual nodes as a weighted combination of their feature vectors, where the weight is the computed similarities. This process can be formulated as:

$$C_{t\to i} = \text{softmax}_\beta(\lambda U_\alpha V_\beta^T)V_\beta. \tag{3}$$

where $\lambda$ is a scaling factor to focus on matched nodes.

Unlike previous approaches [3, 7, 14] that uses the learned correspondence to compute the global similarity, we present a multi-block module that computes block-wise similarity of the textual node and the aggregated visual node $C_{t\to i}$. This is computational efficiency and converts the similarity from a scalar into a vector for subsequent operations. Also, this allows different blocks to play different roles in matching. Concretely, we split the $i$-th feature of the textual node and the its corresponding aggregated visual nodes into $t$ blocks, represented as $[u_{i1}, u_{i2}, \cdots, u_{it}]$ and $[c_{i1}, c_{i2}, \cdots, c_{it}]$, respectively. The multi-block similarity is computed within pair-wise blocks. For instance, the similarity in $j$-th blocks is calculated as $x_{ij} = cos(u_{ij}, c_{ij})$. Here, $x_{ij}$ is a scalar value, $cos(\cdot)$ denotes cosine similarity. The matching vector of $i$-th textual node can be obtained by concatenating the similarity of all the blocks, that is

$$x_i = x_{i1} \,||\, x_{i2} \,||\, \cdots \,||\, x_{it}. \tag{4}$$

where "||" indicates concatenation. In this way, each textual node is associated with its matched visual nodes, which will be propagated to its neighbors at structure-level matching to guide neighbors learn fine-grained phrase correspondence.

Symmetrically, when given a visual graph, the node-level matching is proceeded on each visual node. The corresponding textual nodes will be associated differentially

$$C_{i\to t} = \text{softmax}_\alpha(\lambda V_\beta U_\alpha^T)U_\alpha \tag{5}$$

Then each visual node, together with its associated textual nodes, will be processed by the multi-block module, producing the matching vector $x$.

### 3.2.2 Structure-level Matching

The structure-level matching takes the node-level matching vectors as input, and propagates these vectors to neighbors along with the graph edge. Such a design benefits to learn fine-grained phrase correspondence as neighboring nodes guide that. For example, a sentence *"A brown dog bite a gray dog ear"*, the first *"dog"* will correspond to the visual brown dog in a finer level, because its neighbors *"bite"* and *"brown"* point to the brown dog, and hence the *"dog"* prefer to correlate with the correct dog in the image. To be specific, the matching vector of each node is updated by integrating neighborhood matching vectors using GCN. The GCN layer will apply $K$ kernels that learn how to integrate neighborhood matching vectors, formulated as

$$\hat{x}_i = ||_{k=1}^{K} \sigma\left(\sum_{j\in N_i} W_e W_k x_j + b\right). \tag{6}$$

where $N_i$ denotes the neighborhood of $i$-th node, $W_e$ indicates the edge weight depicted in section 3.1, $W_k$ and $b$ are the parameters to be learned of $k$-th kernel. Note that $k$ kernels are applied, the output of the spatial convolution is defined as a concatenation over the output of $k$ kernels, producing convolved vector that reflects the correspondence of connected nodes. These nodes form the localized phrase.

The phrase correspondence can be inferred by propagating neighboring node correspondence, which can be used to reason the overall matching score of image-text pair. Here, we feed the convolved vectors into a multi-layer perceptron (MLP) to jointly consider the learned correspondence of all the phrases, and infer the global matching score. This represents how much one structured graph matches another structured graph. This process is formulated as

$$s_{t\to i} = \frac{1}{n}\sum_i W_s^u(\sigma(W_h^u\hat{x}_i + b_h^u)) + b_s^u, \tag{7}$$

$$s_{i\to t} = \frac{1}{m}\sum_j W_s^v(\sigma(W_h^v\hat{x}_j + b_h^v)) + b_s^v. \tag{8}$$

where $Ws, bs$ denote parameters of MLP, which includes two fully-connected layers, the function $\sigma(\cdot)$ indicates the tanh activation. Note that we perform structure-level matching on both visual and textual graphs, which can learn phrase correspondence complement to each other. The overall matching score of an image-text pair is computed as the sum of matching score at two directions

$$g(G_1, G_2) = s_{t\to i} + s_{i\to t}. \tag{9}$$

#### 3.2.3 Objective Function

Following previous approaches [18, 14, 2, 30], we employ the triplet loss as the objective function. When using the text $T$ as query, we sample its matched images and mismatched images at each mini-batch, which form positive pairs and negative pairs. The similarity in positive pairs should be higher than that in negative pairs by a margin $\gamma$. Analogously, when using the image $I$ as query, the negative sample should be a text that mismatches the given query, their similarity relative to positive pairs should also satisfy the above constraints. We focus on optimizing hard negative samples that produce the highest loss, that is

$$L = \sum_{(I,T)} [\gamma - g(I,T) + g(I,T^{'})]_+ + [\gamma - g(I,T) + g(I^{'},T)]_+. \tag{10}$$

where $I^{'}, T^{'}$ are hard negatives, the function $[\cdot]_+$ is equivalent to $max[\cdot, 0]$, and $g(\cdot)$ is the global similarity of an image-text pair computed by equation 9.

### 3.3. Feature Representation

**Visual Representation.** Given an image $I$, we represent its feature as a combination of its $n$ salient regions, which are detected by Faster-RCNN pretrained on Visual Genome [13]. The detected regions are feed into pretrained ResNet-101 [5] to extract features, and then transformed into a $d$-dimensional feature space using a fully connected layer:

$$v_i = W_m[CNN(I_i)] + b_m. \tag{11}$$

where $CNN(\cdot)$ encodes each region within bounding box as a region feature, $W_m, b_m$ are parameters of the fully connected layer that transforms the feature into the common space. These region features form the image representation, denoted as $[v_1, v_2, \cdots, v_n]$.

**Textual Representation.** Given a text $T$ that contains $m$ words, we represent its feature as $[u_1, u_2, \cdots, u_m]$, where each word is associated with a feature vector. We first represent each word as a one-hot vector, and then embed it into $d$-dimensional feature space using a Bidirectional Gated Recurrent Unit (BiGRU), which enables to integrated forward and backward contextual information into text embeddings. The representation of $i$-th word is obtained by averaging the hidden state of forward and backward GRU at $i$-th time step.

## 4. Experiment

### 4.1. Dataset and Implementation Details

To validate the effectiveness of our proposed method, we evaluate it on two most widely used benchmarks, Flickr30K [25] and MSCOCO [16]. Each benchmark contains multiple image-text pairs, where each image is described by five corresponding sentences. Flickr30K collects 31,000 images and 31,000 $\times$ 5 = 155,000 sentences in total. Following the settings in previous works [12], this benchmark is split into 29,000 training images, 1,000 validation images, and 1,000 testing images. A large-scale benchmark MSCOCO contains 123,287 images and 123,287 $\times$ 5 = 616,435 sentences, we use 113,287 images for training, both the validation and testing sets contain 5,000 instances. The evaluation result is calculated on 5-folds of testing images.

The commonly used evaluation metrics for image-text matching are Recall@K (K=1,5,10), denoted as R@1, R@5, and R@10, which depict the percentage of ground truth being retrieved at top 1, 5, 10 results, respectively. The higher Recall@K indicates better performance. Additionally, to show the overall matching performance, we also compute the sum of all the Recall values (rSum) at image-to-text and text-to-image directions, that is

$$rSum = \underbrace{R@1 + R@5 + R@10}_{Image\ as\ query} + \underbrace{R@1 + R@5 + R@10}_{Text\ as\ query}. \tag{12}$$

As for implementation details, we train the proposed network on training set and validate it at each epoch on validation set, selecting the model with the highest rSum to be test. We train the proposed method on 1 Titan Xp GPU

Table 1: Image-text matching results on Flickr30K, $'ft'$ and $'fixed'$ are fine-tuning and no fine-tuning. The bests are in bold.

| Method | Image Backbone | Text Backbone | Image-to-Text | | | Text-to-Image | | | rSum |
|---|---|---|---|---|---|---|---|---|---|
| | | | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | |
| m-CNN [20] | fixed VGG-19 | ft CNN | 33.6 | 64.1 | 74.9 | 26.2 | 56.3 | 69.6 | 324.7 |
| DSPE [29] | fixed VGG-19 | w2v+HGLMM | 40.3 | 68.9 | 79.9 | 29.7 | 60.1 | 72.1 | 351.0 |
| VSE++ [2] | ft ResNet-152 | ft GRU | 52.9 | 79.1 | 87.2 | 39.6 | 69.6 | 79.5 | 407.9 |
| TIMAM [27] | fixed ResNet-152 | Bert | 53.1 | 78.8 | 87.6 | 42.6 | 71.6 | 81.9 | 415.6 |
| DANs [23] | ft ResNet-152 | ft LSTM | 55.0 | 81.8 | 89.0 | 39.4 | 69.2 | 79.1 | 413.5 |
| SCO [9] | fixed ResNet-152 | ft LSTM | 55.5 | 82.0 | 89.3 | 41.1 | 70.5 | 80.1 | 418.5 |
| GXN [4] | ft ResNet-152 | ft GRU | 56.8 | - | 89.6 | 41.5 | - | 80.1 | 268.0 |
| SCAN [14] | Faster R-CNN | ft Bi-GRU | 67.4 | 90.3 | 95.8 | 48.6 | 77.7 | 85.2 | 465.0 |
| BFAN [18] | Faster R-CNN | ft Bi-GRU | 68.1 | 91.4 | - | 50.8 | 78.4 | - | 288.7 |
| PFAN [30] | Faster R-CNN | ft Bi-GRU | 70.0 | 91.8 | 95.0 | 50.4 | 78.7 | 86.1 | 472.0 |
| GSMN (sparse) | Faster R-CNN | ft Bi-GRU | 71.4 | 92.0 | 96.1 | 53.9 | 79.7 | 87.1 | 480.1 |
| GSMN (dense) | Faster R-CNN | ft Bi-GRU | 72.6 | 93.5 | 96.8 | 53.7 | 80.0 | 87.0 | 483.6 |
| GSMN (sparse+dense) | Faster R-CNN | ft Bi-GRU | **76.4** | **94.3** | **97.3** | **57.4** | **82.3** | **89.0** | **496.8** |

Table 2: Image-text matching results on MSCOCO, $'ft'$ and $'fixed'$ are fine-tuning and no fine-tuning. The bests are in bold.

| Method | Image Backbone | Text Backbone | Image-to-Text | | | Text-to-Image | | | rSum |
|---|---|---|---|---|---|---|---|---|---|
| | | | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | |
| m-CNN [20] | fixed VGG-19 | ft CNN | 42.8 | 73.1 | 84.1 | 32.6 | 68.6 | 82.8 | 384.0 |
| DSPE [29] | fixed VGG-19 | w2v+HGLMM | 50.1 | 79.7 | 89.2 | 39.6 | 75.2 | 86.9 | 420.7 |
| VSE++ [2] | ft ResNet-152 | ft GRU | 64.7 | - | 95.9 | 52.0 | - | 92.0 | 304.6 |
| DPC [35] | ft ResNet-152 | ft ResNet-152 | 65.5 | 89.8 | 95.5 | 47.1 | 79.9 | 90.0 | 467.8 |
| GXN [4] | ft ResNet-152 | ft GRU | 68.5 | - | 97.9 | 56.6 | - | 94.5 | 317.5 |
| SCO [9] | fixed ResNet-152 | ft LSTM | 69.9 | 92.9 | 97.5 | 56.7 | 87.5 | 94.8 | 499.3 |
| SCAN [14] | Faster R-CNN | ft Bi-GRU | 72.7 | 94.8 | 98.4 | 58.8 | 88.4 | 94.8 | 507.9 |
| BFAN [18] | Faster R-CNN | ft Bi-GRU | 74.9 | 95.2 | - | 59.4 | 88.4 | - | 317.9 |
| PFAN [30] | Faster R-CNN | ft Bi-GRU | 76.5 | 96.3 | **99.0** | 61.6 | 89.6 | 95.2 | 518.2 |
| GSMN (sparse) | Faster R-CNN | ft Bi-GRU | 76.1 | 95.6 | 98.3 | 60.4 | 88.7 | 95.0 | 514.0 |
| GSMN (dense) | Faster R-CNN | ft Bi-GRU | 74.7 | 95.3 | 98.2 | 60.3 | 88.5 | 94.6 | 511.6 |
| GSMN (sparse+dense) | Faster R-CNN | ft Bi-GRU | **78.4** | **96.4** | 98.6 | **63.3** | **90.1** | **95.7** | **522.5** |

with 30 and 20 epochs for Flickr30K and MSCOCO, respectively. The Adam optimizer is employed with mini batch size 64. The initial learning rate is set as 0.0002 with decaying 10% every 15 epochs on Flickr30K, and 0.0005 with decaying 10% every 5 epochs on MSCOCO. We set the dimension of word embeddings as 300, which are then feed into Bi-GRU to get 1024-diemensioanl word representation. As for image feature, each image contains 36 regions that are most salient, and extract 2048-dimensional features for each region. The region feature is then transformed into a 1024-dimensional visual representation by a fully-connected layer. At the structure-level matching, we use one spatial graph convolution layer with 8 kernels, each of which are 32-dimensional. After that, we feed each node in the graph into two fully-connected layers followed by a tanh activation to reason the matching score. The scaling factor $\lambda$ setting is investigated at section 4.2.3. As for optimization, the margin $\gamma$ is empirically set as 0.2.

## 4.2. Experimental Results

### 4.2.1 Comparisons with state-of-the-arts

**Baselines.** we make a comparison with several networks in image-text matching, including (1) typical works m-CNN [20], DSPE [29] and DANs [23] that learn global image-text correspondence by designing different network blocks. (2) VSE++ [2], DPC [35] and TIMAM [27] that learn correspondence using different optimization. (3) SCO [9], GXN [4] that learn region-word correspondence by designing specific networks. (4) state-of-the-art methods SCAN [14], BFAN [18], PFAN [30].

**Quantitative Analysis.** We provide two versions of our approach, one models the text as a sparse graph and another one models it as a dense graph. We ensemble them by averaging their similarity of image-text pairs, and find that can greatly improve the performance. Note that state-of-the-art

Table 3: The ablation study on Flickr30K to investigate the effect of different network structures.

| Model | Image-to-Text | | Text-to-Image | |
|---|---|---|---|---|
| | R@1 | R@10 | R@1 | R@10 |
| GSMN-w/o graph | 63.2 | 94.5 | 48.7 | 84.5 |
| GSMN-w/o t2i | 64.6 | 93.5 | 45.8 | 82.6 |
| GSMN-w/o i2t | 67.0 | 95.5 | 52.3 | 86.3 |
| GSMN-2GCN | 68.4 | 94.8 | 51.5 | 86.0 |
| GSMN-GRU | 71.1 | 95.3 | 50.9 | 85.6 |
| GSMN-full (sparse) | 71.4 | 96.1 | 53.9 | 87.1 |
| GSMN-full (dense) | 72.6 | 96.8 | 53.7 | 87.0 |

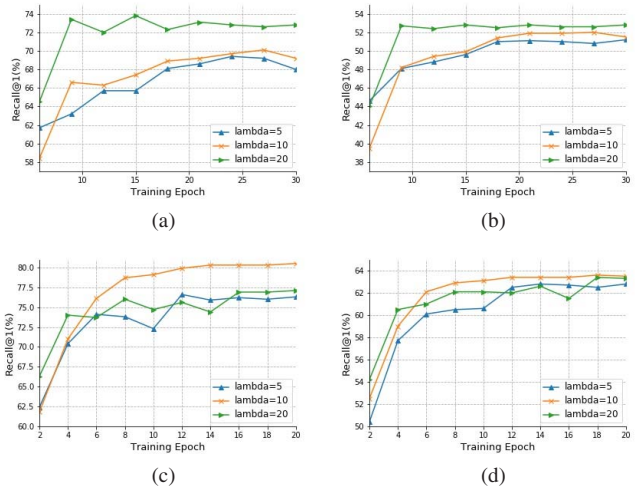


Figure 3: Comparison of Recall@1 results on Flickr30K and MSCOCO with different $\lambda$ settings. (a) Image-to-text on Flickr30K. (b) Text-to-image on Flickr30K. (c) Image-to-text on MSCOCO. (d) Text-to-image on MSCOCO.

methods [30, 14, 18] also use ensemble model. As shown in Table 1, we can observe that the proposed network outperforms state-of-the-arts with respect to all the evaluation metrics on Flickr30K. Compared with the state-of-the-art method PFAN [30] that also utilizing the position information of salient regions, our approach obtains relative R@1 gains with 6.4% and 7% at image-to-text and text-to-image matching. Differs from PFAN [30] that embeds position information into visual representation, our approach employs it as the weight of graph edge. The improvement indicates that structured models object, relation and attribute can greatly improve the matching performance. Although a previous approach SCAN [14] uses similar method to learn object correspondence, our approach achieves more improvement, with nearly 10% R@1 gain, since it ignores to explicitly learn correspondence of the relation and attribute. In addition, our single model also outperforms their ensemble model by a large margin, and the dense model is better than sparse one as it can discover latent dependencies.

The quantitative results on a larger and more complicated dataset MSCOCO is shown at Table 2. We can observe that our approach can outperform state-of-the-art methods with nearly 2% improvement in terms of Recall@1, which is more concerned by users in real applications. Our Recall@10 in image-to-text matching is slightly lower than PFAN since noise exists. Compared with SCAN that is most relevant to our work, we suppress it in terms of all the evaluation metrics, getting over 5.5% and 4.5% relative Recall@1 improvements on two directions. Note that the sparse model performs better than the dense model, it mainly arises from the sentence in this dataset is more complicated, and thus might incorrectly correlate totally irrelevant words if a fully-connected graph is built.

#### 4.2.2 Impact of different network structures

To validate the impact of different network structures, we conduct ablation studies incrementally on Flickr30K. We compare the full dense model and full sparse model with five models: (1) GSMN-w/o graph, which only performs node-level matching. (2) GSMN-w/o i2t, which only applies the node-level matching and structure-level matching on image-to-text direction. (3) GSMN-w/o t2i, which only applies the node-level matching and structure-level matching on text-to-image direction. (4) GSMN-2GCN, its depth of GCN layer is set as 2. (5) GSMN-GRU, a network that only uses GRU instead of Bi-GRU as the text encoder. As shown in Table 3, The two full models outperform all these types of networks, and they largely exceed the network that only performs matching on single direction. Note that GSMN-2GCN requires more computational cost and GPU memory, results show that a deeper network will drop the performance as it additionally considers indirectly connected nodes, which will disturb the learned correspondence. Compared with GSMN-GRU, our approach achieves more improvement on text-to-image graph, it derives from the Bi-GRU can better model the semantic dependency among object, relation and attribute than GRU, and hence the edge weight of textual graph can be accurately reflected. Note that GSMN-w/o i2t gets better performance than GSMN-w/o t2i, because the implicit relation among regions is difficult to be discovered.

#### 4.2.3 Impact of different parameters

To validate the impact of different parameters, we conduct extensive experiments on two benchmarks. In this work, the most sensitive parameter is the scaling factor $\lambda$ that determines the relative weight of different nodes in node-level matching, and the edge weight of textual graph. A large $\lambda$ will filter out extensive nodes, and only preserve little nodes that are highly relevant to the specific node. A


Figure 4: Visualization of node correspondence and phrase correspondence with score inside the box. Best viewed in color.


Figure 5: Visualization of text-to-image matching on Flickr30K. For each text query, we show top 3 ranked images from left to right, where mismatched images are with red boxes and matched images are with green boxes.

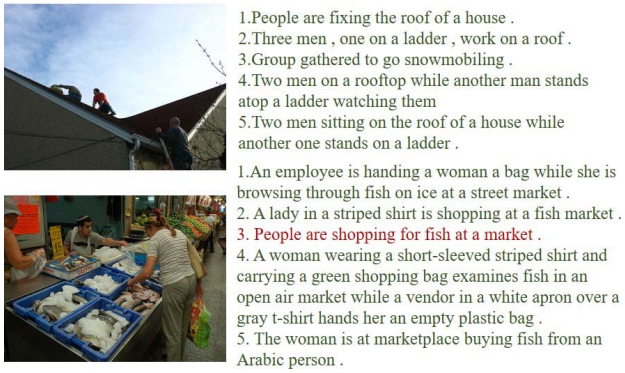


Figure 6: Visualization of image-to-text matching on Flickr30K. For each image query, we show top 5 ranked texts, where mismatched texts are marked as red.

small $\lambda$ is unable to distinguish relevant nodes from irrelevant ones. Hence, an appropriate parameter is important in our proposed network. Here, we investigate the matching performance with setting the $\lambda$ as 5, 10 and 20, see figure 3. We observe the Recall@1 on validation set at each training epoch. The top two subfigures are on Flickr30K, it is obvious that when $\lambda = 20$, the proposed network yields better Recall@1 on two matching directions, and there is just little difference when the parameter is set as 5 and 10. The bottom two subfigures are on MSCOCO, showing that $\lambda = 10$ is much better. The different parameter setting on two datasets might be caused by different data distribution.

#### 4.2.4 Case Study

We provide a visualization to show the learned node correspondence and phrase correspondence in Figure 4. Note that we only show the most relevant region for each textual node, it shows different kinds of nodes can associate with their corresponding regions with relatively higher scores. Moreover, we can infer phrase correspondence enclosed by multiple bounding boxes, and their scores are greatly improved. Also, we visualize the text-to-image and image-to-text matching results on Flickr30K, shown in Figure 5 and Figure 6. These show our approach always retrieves the ground truth with a high rank. In addition, our approach is able to learn fine-grained correspondence of the relation and attribute. For example, for the first text query in Figure 5, our network can distinguish different kinds of hats.

### 4.3. Conclusion

In this paper, we propose a graph structured matching network for image-text matching, which performs matching on heterogeneous visual and textual graphs. This is achieved by node-level matching and structure-level matching that infer fine-grained correspondence by propagating node correspondence along the graph edge. Moreover, such a design can learn correspondence of relation and attribute, which are mostly ignored by previous works. With the guidance of relation and attribute, the object correspondence can be greatly improved. Extensive experiments demonstrate the superiority of our network.

## 5. Acknowledgements

This work is supported by the National Natural Science Foundation of China, Grant No.U19A2057, the Fundamental Research Funds for the Central Universities, Grant No.WK3480000008, the National Key Research and Development Program of China, Grants No.2016QY03D0505, 2016QY03D0503, 2016YFB081304, Strategic Priority Research Program of Chinese Academy of Sciences, Grant No.XDC02040400.

## References


[1] A. Eisenschtat and L. Wolf. Linking image and text with 2-way nets. In *CVPR*, pages 1855–1865, 2017.

[2] Fartash Faghri, David J. Fleet, Jamie Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*, 2018.

[3] Haoqi Fan and Jiatong Zhou. Stacked latent attention for multimodal reasoning. *CVPR*, pages 1072–1080, 2018.

[4] Jiuxiang Gu, Jianfei Cai, Shafiq R. Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. *CoRR*, abs/1711.06420, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems*, pages 7211–7221, 2018.

[7] Feiran Huang, Xiaoming Zhang, Zhonghua Zhao, and Zhoujun Li. Bi-directional spatial-semantic attention networks for image-text matching. *IEEE Transactions on Image Processing*, 28(4):2008–2020, 2019.

[8] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2310–2318, 2017.

[9] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2018.

[10] Zhong Ji, Haoran Wang, Jungong Han, and Yanwei Pang. Saliency-guided attention network for image-sentence matching. *arXiv preprint arXiv:1904.09471*, 2019.

[11] 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.

[12] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. 3:1889–1897, 2014.

[13] 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.

[14] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

[15] Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. Identity-aware textual-visual matching with latent co-attention. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1908–1917, 2017.

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

[17] An-An Liu, Yu-Ting Su, Wei-Zhi Nie, and Mohan Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):102–114, 2016.

[18] Chunxiao Liu, Zhendong Mao, An-An Liu, Tianzhu Zhang, Bin Wang, and Yongdong Zhang. Focus your attention: A bidirectional focal attention network for image-text matching. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 3–11. ACM, 2019.

[19] Yu Liu, Yanming Guo, Erwin M. Bakker, and Michael S. Lew. Learning a recurrent residual fusion network for multimodal matching. In *ICCV*, pages 4127–4136, 2017.

[20] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, pages 2623–2631, 2015.

[21] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[22] Niluthpol Chowdhury Mithun, Rameswar Panda, Evangelos E. Papalexakis, and Amit K. Roy-Chowdhury. Webly supervised joint embedding for cross-modal image-text retrieval. 2018.

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

[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] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *ICCV*, pages 2641–2649, 2015.

[26] S. Ren, K. He, R Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[27] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Adversarial representation learning for text-to-image matching. *arXiv preprint arXiv:1908.10534*, 2019.

[28] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015.

[29] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2019.

[30] Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. Position focused attention network for image-text matching. *arXiv preprint arXiv:1907.09748*, 2019.

[31] Yiling Wu, Shuhui Wang, and Qingming Huang. Learning semantic structure-preserved embeddings for cross-modal retrieval. In *2018 ACM Multimedia Conference*, pages 825–833. ACM, 2018.

[32] Ning Xu, Hanwang Zhang, An-An Liu, Weizhi Nie, Yuting Su, Jie Nie, and Yongdong Zhang. Multi-level policy and reward-based deep reinforcement learning framework for image captioning. *IEEE Transactions on Multimedia*, 2019.

[33] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019.

[34] Ying Zhang and Huchuan Lu. Deep cross-modal projection learning for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–701, 2018.

[35] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, and Yi-Dong Shen. Dual-path convolutional image-text embedding with instance loss. *arXiv preprint arXiv:1711.05535*, 2017.