Conceptual and Syntactical Cross-modal Alignment with Cross-level Consistency for Image-Text Matching

Pengpeng Zeng
Center for Future Media
University of Electronic Science and Technology of China
Chengdu, China
is.pengpengzeng@gmail.com

Lianli Gao
Center for Future Media
University of Electronic Science and Technology of China
Chengdu, China
lianli.gao@hotmail.com

Xinyu Lyu
Center for Future Media
University of Electronic Science and Technology of China
Chengdu, China
xinyulyu68@gmail.com

Shuaiqi Jing
Center for Future Media
University of Electronic Science and Technology of China
Chengdu, China
jingshuaiqi@uestc.edu.cn

Jingkuan Song*
Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China
Chengdu, China
jingkuan.song@gmail.com

ABSTRACT

Image-Text Matching (ITM) is a fundamental and emerging task, which plays a key role in cross-modal understanding. It remains a challenge because prior works mainly focus on learning fine-grained (*i.e.*, coarse and/or phrase) correspondence, without considering the syntactical correspondence. In theory, a sentence is not only a set of words or phrases but also a syntactic structure, consisting of a set of basic syntactic tuples (*i.e.*, *<(attribute) object - predicate - (attribute) subject>*). Inspired by this, we propose a **C**onceptual and **S**yntactical Cross-modal Alignment with **C**ross-level **C**onsistency (CSCC) for Image-text Matching by simultaneously exploring the multiple-level cross-modal alignments across the concept and syntactic with a consistency constraint. Specifically, a conceptual-level cross-modal alignment is introduced for exploring the fine-grained correspondence, while a syntactical-level cross-modal alignment is proposed to explicitly learn a high-level syntactic similarity function. Moreover, an empirical cross-level consistent attention loss is introduced to maintain the consistency between cross-modal attentions obtained from the above two cross-modal alignments. To justify our method, comprehensive experiments are conducted on two public benchmark datasets, *i.e.*, MS-COCO (1K and 5K) and Flickr30K, which show that our CSCC outperforms state-of-the-art methods with fairly competitive improvements.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**.

KEYWORDS

Image-Text Matching, Graph Neural Network, Hypergraph Neural Network, Attention, Alignment

ACM Reference Format:
Pengpeng Zeng, Lianli Gao, Xinyu Lyu, Shuaiqi Jing, and Jingkuan Song. 2021. Conceptual and Syntactical Cross-modal Alignment with Cross-level Consistency for Image-Text Matching. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475380

*Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475380

1 INTRODUCTION

Cross-modal understanding of visual and language is of great importance to the development of artificial intelligence, which has attracted much interest from the multimedia community. Towards this goal, one core solution is to learn the latent correspondence between visual and its related language, which is commonly referred to as image-text matching. Image-text matching is usually regarded as the fundamental task of many visual-language tasks, including visual grounding [5, 39], image captioning [10, 46] and visual question answering [14, 21, 23].

One promising and scalable strategy for learning image-text matching is the fine-grained correspondence, which builds the granular alignment between the local regions in the image and words in the sentence. The main reason is that people tend to describe the salient objects, the corresponding attributes, and the relationship between the objects about the image. For instance, Karpathy *et al.* [15] proposed a dense alignment method to compute the similarity between the extracted region features in the image and word features in the sentences. Due to the dependencies between regions in the image or words in the sentence, graph neural networks are utilized to enhance visual or textural features by building the relationship between them. Li *et al.* [20] proposed Visual Semantic Reasoning to build up connections between image regions and generate visual representations with semantic relationships. Liu *et al.* [26] adopted off-the-shelf Stanford CoreNLP [28] to extract

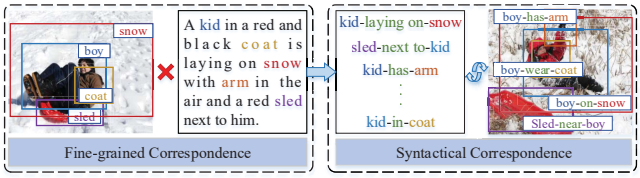


Figure 1: Image-text matching examples from the fine-grained and the syntactical correspondence, respectively. In the left figure, the concepts between the image and the text are relevant, but the result of matching is wrong. In the right figure, the syntactical correspondence is capable of building the better alignment between the concept-level and syntactic-level and obtaining the correct matching.

semantic dependency between concepts of the text and learned the fine-grained correspondence.

However, due to the large heterogeneity gap between images and texts, the above correspondence may not well consider the rich semantics information. Actually, the semantics of the sentence is complicated. Firstly, a sentence is composed of a set of words or phrases with different concepts, including objects (*e.g.,* nouns), attributes (*e.g.,* adjectives), and relations (*e.g.,* verbs). There are dependencies between different concept words. For instance, relational terms usually indicate relationships between objects, and attribute terms present the appearance information of an object. Moreover, the sentence has its specific structure, which can be decoupled into one or more basic syntactic tuples (<*(attribute) object-predicate-(attribute) subject*>). Thus, the process of matching between images and texts needs to consider the above two semantics information. For the left example of Figure 1, despite the relevant concepts of image and texts, the result of matching is wrong. For the left example of Figure 1, we parse the image and text to obtain the syntactic tuple, and aligning the syntactic tuples can obtain the correct result. In reality, when comparing the pair-wise image-text, we humans usually capture low-level information of concepts between inter-modalities (*e.g.,* objects, attributes, and relations) at the first glimpse. Then, higher-level semantics, *e.g.,* syntactic structure, are mined by examining the correlations of the structure of images and texts to obtain a better understanding. This indicates that the process of image-text matching should be gradual and progressive, which contains conceptual and syntactical cross-modal alignment.

Based on the above insight, we propose a novel **C**onceptual and **S**yntactical Cross-modal Alignment with **C**ross-level **C**onsistency, termed *CSCC*, for image-text matching. Specifically, the pipeline of CSCC mainly consists of three parts: 1) *Conceptual Cross-modal Alignment (CCA)*: we first construct the image scene graph SG^I and the text scene graph SG^T from the image and its correspondence text. Then, we infer the conceptual-level embedding by building the relationship between concept nodes in the scene graph. To compute the matching score at concept-level, a concept-level cross-modal alignment is proposed to build the fine-grained correspondence from SG^I to SG^T and SG^T to SG^I. 2) *Syntactical Cross-modal Alignment (SCA)*: based on the structure of the basic syntactic tuple, we produce multiple syntactic tuples from the scene graph. To learn the syntactic tuple-level embedding, we introduce the hypergraph to represent the syntactic tuple graph and adapt a hypergraph convolutional network to learn the interaction between the local feature and syntax. Similar to CCA, we also propose a syntactic tuple-level cross-modal alignment to establish the alignment between inter-modalities at the syntactical level and generate the syntactical matching score. 3) *Cross-level Consistency (CC)*: considering that humans have a consistent observation ability under different levels of matching conditions, we design a cross-level consistency loss. It encourages the consistency between co-correlation attention from two-level cross-modal alignment and further boosts the performance of image-text matching.

The main contributions can be summarized as follows:

- We propose a *CSCC* that explicitly constructs scene graphs and syntactical tuple graphs for images and sentences, and simultaneously learns fine-grained and syntactical correspondences with our novel conceptual and syntactical cross-modal alignments, respectively.
- We design a cross-level consistency loss to further improve the effectiveness of our *CSCC* by setting an attention-based constraint to make sure that two-level cross-modal alignments perform consistency learning.
- We conduct extensive experiments on two public benchmarks: MS-COCO (1K and 5K) and Flickr30K and achieve the state-of-the-art performance.

2 RELATED WORKS

2.1 Image-Text Matching

Image-text matching task aims to explore the correspondence between the images and texts. The early methods [6, 18, 22, 27, 33, 35, 44] embed the feature vectors of image and text into the common embedding space optimized by a ranking loss. For example, Kiros et al. [18] attempted to embed the representation of image and sentence by using convolution neural network (CNN) and recurrent neural network (RNN) separately, and optimize the model by a triplet ranking loss. Although such methods have made great progress, these methods cannot capture the fine-grained details. Several works have recently been proposed exploring the fine-grained correspondence between regions in the image and words in the sentence for image-text matching [13, 19, 29, 30, 40]. For instance, Huang *et al.* [13] proposed a context-modulated attention scheme to selectively focus on salient pair-wise image-sentence instances. Then, a multi-modal LSTM was applied to propagate local similarities into a global one sequentially. Moreover, the attention mechanism has been applied to image-text matching, aiming to attend to the key part in the input sequence. Lee *et al.* [19] proposed a stacked cross attention network (SCAN) which aligns between fragments from the different modalities. However, the above methods neglect the syntactic structure of the sentence. Our model explores the multiple-level correspondence, including fine-grained correspondence and syntactical correspondence.

2.2 Graph Convolution Network

Graph convolution network (GCN) has gained popularity and been employed in various vision-and-language tasks, such as image captioning [4, 37, 45], VQA [8, 9, 14, 34] and visual grounding [5, 39].

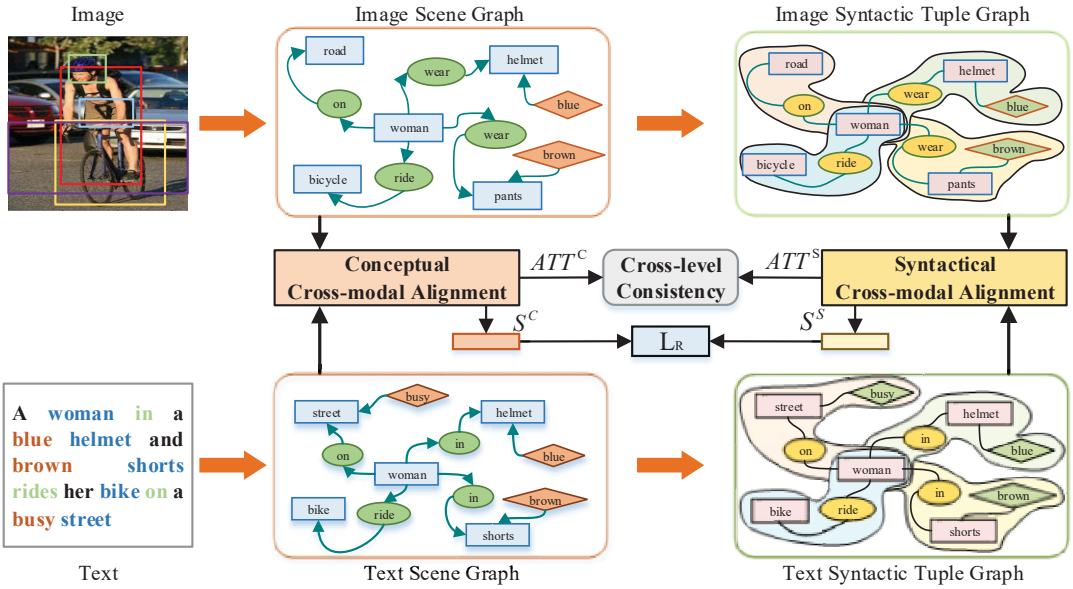


Figure 2: Framework of the proposed CSCC. We devise the conceptual cross-modal alignment and syntactical cross-modal alignment to learn fine-grained and syntactical correspondence, respectively. The cross-level consistency is applied to build the relationship between two-level alignments.

Thanks to its great ability to enhance the embedding feature by building the relationship between the fragments, GCN has been introduced into image-text matching. Li *et al.* [20] built a region relationship model to enhance the visual representation by considering the semantic correlation between image regions. Liu *et al.* [26] proposed a Graph Structured Matching Network that constructs the graph structure for image and text and performs the fine-grained phrase correspondence. Despite the graph convolution networks capture the pair-wise relationship, the GCN can not learn the high-order information. Thus, sub-graph networks are proposed to learn the structure information between sub-graphs in the graph, such as Subgraph Neural Networks (SubGNN) [1] and Hypergraph Convolutional Networks (HGCN) [7], which are applied in many tasks [16, 43]. In our work, based on the syntactical structure of the language, we regard a basic syntactic tuple as a hyperedge and obtain a syntactic tuple-level embedding by HGCN.

3 APPROACH

In this work, we present a novel **C**onceptual and **S**yntactical Cross-modal Alignment with **C**ross-level **C**onsistency (CSCC) for image-text matching. The overview of the network is depicted in Figure 2.

3.1 Overview

Given an image I and a text T, the task is to compute the similarity between them. Our way contains two-level cross-modal alignments: Conceptual Cross-modal Alignment (CCA) and Syntactic Cross-modal Alignment (SCA), and a Cross-level Consistency (CC) which is to perform the consistency learning between two-level alignments. Technically, for the conceptual cross-modal alignment, two scene graphs (SG^I and SG^T) composed of a set of concept nodes (*e.g.,* objects, relations, and attributes) are obtained by applying scene graph algorithms. Due to the dependencies between concept nodes, a Graph Convolutional Network (GCN) is adopted to learn the concept-level embedding by propagating the information of neighbor nodes into the local node. Concept-level cross-modal alignment performs the fine-grained correspondence of concept nodes between two modalities, which obtains the concept-level co-correlation matrix ATT^C and the matching score S^C, shown in Figure 3. As for the syntactical cross-modal alignment, we first construct the syntactic tuple graph from the scene graph according to the basic structure of the syntactic tuple. Then, a HyperGraph Convolutional Network (HGCN) is used to generate the syntactic-level embedding. Given the visual and textual syntactic-level embedding, the syntactic-level cross-modal alignment is utilized to establish syntactical correspondence between inter-modalities, which also generates the syntactic-level co-correlation matrix ATT^S and matching score S^S. Thus, the final matching score is the sum of the above two-level matching scores (S^C and S^S). Besides, a cross-level consistency is designed to encourage a stronger correlation between ATT^C and ATT^S, which aims at reducing the difference between the two matrices. More details will be elaborated on in the following sections.

3.2 Conceptual Cross-modal Alignment

The proposed Conceptual Cross-modal Alignment (CCA) aims to learn the fine-grained correspondence, which contains scene graph construction, concept-level encoding, and conceptual-level cross-modal alignment.

Scene Graph Construction. Benefiting from its powerful representation ability, scene graph [11, 12] has been widely used in many cross-modal tasks to represent the visual concepts and textual concepts about images and texts. Generally, the scene graph is defined as $SG = (N, E)$, where N and E denotes a set of nodes and edges, respectively. In our task, we extract the image scene graph SG^I and the text scene graph SG^T:

1) Image Scene Graph. $SG^I = (N^I, E^I)$ is obtained by feeding the image I into the image scene graph algorithm [48]. In practice, we first identify the objects within the image using an external object detector (*e.g.,* Faster R-CNN pre-trained on MS-COCO datasets) to produce a set of object proposals and the corresponding object features o. After that, a visual scene graph detector is employed to predict the relationship between object proposals. Furthermore, another attribute classifier is used to get the attribute label of each object within the proposal. Thus, nodes in N^I has three kinds of concepts, including relation nodes r, object nodes o, attribute nodes a. Each node in N^I is represented by a d-dimensional vector, denoted as n_r^I, n_o^I and n_a^I. Based on the above operation, there are dependencies between concept nodes: 1) if an object o_i^I has an attribute a_{il}^I, there exist a directed edge from a_{il}^I to o_i^I; and 2) if a relationship triplet $< o_i^I, r_{ij}^I, o_j^I >$ exists, we assign two directed edges from o_i^I to r_{ij}^I and o_j^I to r_{ij}^I, respectively.

2) Text Scene Graph: Given a text T, we employ a fixed rule-based language parsing toolkit [2] to produce a sentence scene graph $\mathrm{SG}^T = (N^T, E^T)$. Specifically, the captions of an image are fed into the parser to obtain the label information of each word (*e.g.,* object, attribute and relation) and the dependency between each word. Similar to SG^I, there are also three kinds of nodes in the N^T. For these three types of nodes, we adopt the same word-embedding layer to represent the feature of each node.

Concept-level Encoding. Since the relationship of the concept is found to be effective, we utilize Graph Convolutional Networks (GCN) to encode the scene graph SG and obtain the concept-level embedding U by merging the surrounding information into the local node. The concept-level embedding U includes relation embedding, attribute embedding, and object embedding. Following [45], we adapt multi-spatial graph convolutions to learn different embedding for each type:

1) Relation Embedding. For a relation node r_{ij} in SG, it connects two surrounding objects o_i and o_j, denoted as $\langle o_i, r_{ij}, o_j \rangle$. Thus, the relation embedding $u_{r_{ij}}$ can be represented by merging the information of the object node and its surrounding ones:

$$u_{r_{ij}} = g_r\left(concat\left(\mathrm{n}_{o_i}, \mathrm{n}_{o_j}, \mathrm{n}_{r_{ij}}\right)\right), \tag{1}$$

where g_r means the graph convolutional operation for relation.

2) Attribute Embedding. For an object node o_i in SG, it usually owns several attributes $\{a_1^i, a_2^i, ..., a_{N_{a_i}}^i\}$, where N_{a_i} denotes the total number of its attribute nodes. We fuse the information of object o_i and all related attribute nodes to obtain attribute embedding u_{a_i}:

$$u_{a_i} = \frac{1}{N_{a_i}} \sum_{l=1}^{N_{a_i}} g_a\left(concat\left(\mathrm{n}_{o_i}, \mathrm{n}_{a_{i,l}}\right)\right), \tag{2}$$

where $a_{i,l}$ denotes the l-th attribute node of o_i and g_a is the graph convolutional operation for the attributes.

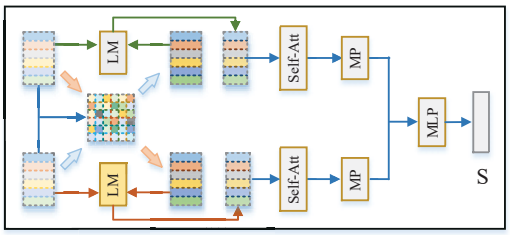


Figure 3: The illustrations of Cross-modal Alignment Module. The cross-modal alignment module can be both used for concept-level and syntactic-level cross-modal alignment.

3) Object Embedding. In the SG, an object o_i plays the role of "subject" or "object" depended on the different edge direction. Thus, we use different functions to incorporate all such knowledge, based on the role of the object o_i, into the object o_i:

$$u_{o_i} = \frac{1}{N_{r_i}}[\sum_{o_j \in sbj(o_i)} g_s(concat(\mathrm{n}_{o_i}, \mathrm{n}_{o_j}, \mathrm{n}_{r_{ij}}))+ \sum_{o_k \in obj(o_i)} g_o(concat(\mathrm{n}_{o_k}, \mathrm{n}_{o_i}, \mathrm{n}_{r_{ki}}))], \tag{3}$$

where $o_j \in sbj(o_i)$ and $o_k \in obj(o_i)$ mean (o_i, o_j) and (o_k, o_i) form subject-object pairs, respectively. And $N_{r_i} = |sbj(o_i)| + |obj(o_i)|$. g_s and g_o are the graph convolutional operation for objects as a"subject" or an "object".

Thus, we adopt the above embedding operation of GCN to obtain concept-level embedding U^I and U^T for SG^I and SG^T, respectively.

Concept-level Cross-modal Alignment. Considering that the co-correlation between inter-modalities is important, the concept-level cross-modal alignment is introduced to compute the similarity of concept nodes between two scene graphs. Specifically, we first perform a cross-attention mechanism to obtain the cross-modal attention matrix ATT^C, sized $\left|N^I\right| \times \left|N^S\right|$. Each attention weight $\alpha_{ij} \in ATT^C$ means how the visual node u_i^I corresponds to each textual node u_j^T by calculating the cosine similarity between them:

$$\alpha_{ij} = cosine(u_i^I, u_j^S), \tag{4}$$

where $cosine(a, b)$ returns the cosine value in [-1,1] which measures the similarity of its two inputting vectors a and b.

Then, we multiply the attention matrix ATT^C with the textual concept-level embedding U^T to produce the attended visual embedding embedding $\tilde{U}^I$:

$$\tilde{U}^I = [ATT^C]_+ U^I, \tag{5}$$

where $[x]_+ \equiv \max(x, 0)$. By switching the role of SG^I, SG^T in the above procedure, the attended textual embedding $\tilde{U}^T$ is obtained for U^T in G_S.

Based on the attended embedding, we apply the local similarity function to calculate local-matching representation M^T and M^I between each original embedding and its attended embedding for SG^I and SG^T, respectively. In particular, local-matching representation M^T and M^I are computed as below:

$$\begin{aligned} M^T &= norm(W_T \tilde{U}^T) - norm(W_T U^T), \\ M^I &= norm(W_I \tilde{U}^I) - norm(W_I U^I), \end{aligned} \tag{6}$$

where W_* is the learnable parameter and $norm()$ means the l2-normalized operation on the input matrix.

To achieve more comprehensive similarity reasoning, we utilize the self-attention module to propagate the similarity messages based on the matching representation and adopt a mean-pooling operation to obtain the global matching representation on each modality. The specific operation of self-attention refers to [36].

$$\begin{aligned} \tilde{M}^I &= Self\_Att(M^I) + M^I, \\ \tilde{M}^T &= Self\_Att(M^{\mathrm{T}}) + M^{\mathrm{T}}, \\ \bar{m}^I &= Mean\_Pooling(\tilde{M}^I), \\ \bar{m}^T &= Mean\_Pooling(\tilde{M}^{\mathrm{T}}). \end{aligned} \tag{7}$$

Finally, we fuse the two global matching representations and feed them to Multi-Layer Perception (MLP) to get the final matching score S^C at concept-level.

We simplify formulate the above operation as follow:

$$S^C, ATT^C = CCA(SG^I, SG^T). \tag{8}$$

3.3 Syntactical Cross-modal Alignment

Different from the above *CCA*, the Syntactical Cross-modal Alignment (*SCA*) aims to learn the syntactical correspondence between inter-modalities, which contains syntactic tuple graph construction, syntactical tuple-level encoding and syntactical-level cross-modal alignment.

Syntactic Tuple Graph Construction. In this section, we introduce the theory of hypergraph [7] to construct the syntactic tuple graph. Different from the scene graph, the hypergraph is defined as HG = {X, Y, Z}, where X denotes a set of vertexes, Y denotes a set of hyperedges and Z is diagonal matrix of hyperedge weights. Z can be initialized with an identity matrix. It is worth noting that a hyperedge connects more than two vertexes, unlike the edge in a scene graph that connects only two vertexes. Based on the scene graph, we treat the vertexes that satisfy the condition of the basic syntactic tuple (*i.e.*, *<(attribute) object-relation-(attribute) subject>*) as a hyperedge. The basic syntactic tuple is centered on the relation node. Thus, the number of hyperedge is equal to the number of relation nodes in the scene graph. For example in the Figure 2, the sentence contains four hyperedges, which are represented by polygons of different colors. A hyperedge around the relation "on" connects the following nodes: "on", "street", "woman", and "busy", which is marked by a light orange polygon. Let ST_j denotes the vertex set of j-th syntactic tuple, then the element of hypergraph incidence matrix H can be formulated as:

$$[H]_{ij} = \begin{cases} 1, x_i \in ST_j \\ 0, x_i \notin ST_j. \end{cases} \tag{9}$$

According to the above theory, we can represent two syntactic tuple graphs using hypergraph, namely HG^I and HG^T, and obtain the correspondence hypergraph incidence matrix H^I and H^T.

Syntactic Tuple-level Encoding. To obtain the syntactic tuple-level embedding, we adopt HyperGraph Convolutional Network (*HGCN*) to capture the dependencies about syntactic and explore the interaction between local features and syntactic. The operation of *HGCN* performs the transformation of "vertex-hyperedge-vertex". At the l-th layer of *HGNN*, F_l is the vertex features of hypergraph and ϕ_l is the learnable filter matrix of hypergraph neural network. The convolutional layer of *HGCN* can be formulated as:

$$F_{l+1} = \delta(D_x^{-1/2} H Z D_y^{-1} H' D_x^{-1/2} F_l \phi_l), \tag{10}$$

where $\delta(.)$ is the nonlinear activation function and H' represents the transpose operation of H. D_y and D_v denote the degrees of the hyperedge and vertex, respectively. They are used for normalization. For a vertex $x \in X$ and a hyperedge $y \in Y$, their degrees can be calculated by:

$$d(x) = \sum_{y \in Y} z(y) a(x, y), d(y) = \sum_{x \in Y} h(x, y), \tag{11}$$

where $z(y)$ is an element on the diagonal of matrix Z and indicates the weight of hyperedge and $h(x, y)$ is the element of H. According to Equation 10, the process of the convolutional layer of HGCN is shown as follows. Specifically, at the j-th layer of HGNN, the transformation of "vertex-hyperedge" firstly projects the vertex feature F_l into the new vertex feature by a linear operation and multiplies the new vertex feature with the H' to produce the hyperedge feature. The hyperedge feature gathers the information of vertexes in the hyperedge. Note that the F_1 is the output of the concept-level embedding U. The transformation of "hyperedge-vertex" means that the final vertex feature F_{l+1} is obtained by implementing the multiplication of H to associate the related hyperedge feature. Finally, the final syntactic tuple-level embedding is calculated by:

$$F = H' F_{l+1}. \tag{12}$$

Thus, we adopt the above embedding operation of HGCN to obtain syntactic tuple-level embedding F^I and F^T for HG^I and HG^T, respectively.

Syntactic-level Cross-modal Alignment. This module is similar to concept-level cross-modal alignment, which firstly computes the similarities of syntactic-level embeddings between inter-modalities and obtains the co-correlation matrix ATT^S. Then, based on the ATT^S, we produce the attended embedding of hyperedges, use the local similarity function to compute the difference between each syntactic-level embedding and its attended embedding, and obtain the matching representation. Next, the Self-Att module is adopted to aggregate the matching representation. Finally, we integrate the matching information of the two modalities through Mean-Pooling, Sum, and MLP operations to get the final matching score S^S at syntactic-level.

This above process can be simply formulated as:

$$S^S, ATT^S = SCA(HG^I, HG^S). \tag{13}$$

3.4 Cross-level Consistency

The Cross-level Consistency module aims to build up the correlation between the conceptual level and syntactical level. We argue that the weights of the cross-modal attention matrix should be consistent at different level conditions, which means that the difference between the weights of ATT^C and ATT^S should be slight. Since the dimensions of the two attention matrices are inconsistent, we maintain the same dimensions by multiplying the matrices. The loss function of MSE is then used to measure the difference between

Table 1: Comparison with the state-of-the-art methods on MS-COCO 1K and Flickr30K test set. "* " denotes an ensemble model and "-" means the result is not provided.

Methods	MS-COCO 1K						Flickr30K					
	Sentence Retrieval			Image Retrieval			Sentence Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
$DVSA_{cvpr15}$	38.4	69.9	80.5	27.4	60.2	74.8	22.2	48.2	61.4	15.2	37.7	50.5
$DSPE_{tpami18}$	50.1	79.7	89.2	39.6	75.2	86.9	40.3	68.9	79.9	29.7	60.1	72.1
$VSE++_{bmvc18}$	64.6	90.0	95.7	52.0	84.3	92.0	52.9	80.5	87.2	39.3	70.1	79.5
$SCAN^{*}_{eccv18}$	72.7	94.8	98.4	58.8	88.4	94.8	67.4	90.3	95.8	48.6	77.7	85.2
$VSRN_{iccv19}$	74.0	94.3	97.8	60.8	88.4	94.1	70.4	89.2	93.7	53.0	77.9	85.7
$VSRN^{*}_{iccv19}$	76.2	94.8	98.2	62.8	89.7	95.1	71.3	90.6	96.0	54.7	81.8	88.2
$CAMP_{iccv19}$	72.3	94.8	98.3	58.5	87.9	95.0	68.1	89.7	95.2	51.5	77.1	85.3
$BFAN^{*}_{mm19}$	74.9	95.2	-	59.4	88.4	-	68.1	91.4	-	50.8	78.4	-
$PFAN_{ijcai19}$	76.5	96.3	**99.0**	61.6	89.6	95.2	70.0	91.8	95.0	50.4	78.7	86.1
$CAMERA_{mm20}$	75.9	95.5	98.8	63.4	90.9	95.8	**76.5**	**95.1**	97.2	58.9	84.7	90.2
$MMCA_{cvpr20}$	74.8	95.6	97.7	61.6	89.8	95.2	74.2	92.8	96.4	54.8	81.4	87.8
$CAAN_{cvpr20}$	75.5	95.4	98.5	61.3	89.7	95.2	70.1	91.6	97.2	52.8	79.0	87.9
$IMRAM^{*}_{cvpr20}$	76.7	95.6	98.5	61.7	89.1	95.0	74.1	93.0	96.6	53.9	79.4	87.2
$GSMN_{cvpr20}$	76.1	95.6	98.3	60.0	88.7	95.0	71.4	92.0	96.1	53.9	79.7	87.1
$CSMN^{*}_{cvpr20}$	78.4	**96.4**	98.6	63.3	90.1	95.7	76.4	94.3	**97.3**	57.4	82.3	89.0
CSCC(our)	**78.8**	96.1	**99.0**	**66.6**	**92.5**	**96.4**	72.7	93.4	96.5	**61.2**	**86.7**	**91.5**

the attention matrices.

$$L_{CC} = MSE(H^{I\prime} ATT^{C} H^{T}, ATT^{S}). \tag{14}$$

3.5 Learning Objectives

Based on the above operation, we obtain two-level similarity scores (S^C and S^S). The final matching score S^F is the sum of these scores. To optimize the total model *CSCC* in the training, the triplet ranking loss is widely used to measure the matching degree of the matched image-text pairs.

$$\begin{aligned} L_R = &[\gamma - S^F(I,T) + S^F(I,T^-)]_+ \\ &+[\gamma - S^F(I,T) + S^F(I^-,T)]_+ \end{aligned}, \tag{15}$$

where γ is the margin parameter. (I,T) denotes a matched image-text pair and (I^-,T) and (I,T^-) denote the corresponding hardest negative pair.

Thus, the final learning objective is sum of two losses (L_R and L_{CA}):

$$L = L_R + \lambda L_{CC}, \tag{16}$$

where λ is a trade-off of the effects of two loss functions on the final model.

4 EXPERIMENTS

4.1 Experimental Setting

4.1.1 Datasets. We evaluate our method on two public benchmarks: MS-COCO dataset [24] and Flickr30K dataset [47] in the experiments.

MS-COCO is a large-scale benchmark for image understanding, which contains 123,287 images with five corresponding text descriptions for each image. Similar to the prior work [19], we separate the dataset into 113,287 images for training, 5,000 for validation and 5,000 for testing. For the evaluation on MS-COCO, we report the results of two settings, in which we test our model on 1K and all 5K testing images, respectively.

Flickr30K is also an image-text matching dataset, which contains 31,783 images collected from the Flickr website. Each image is annotated with five descriptions generated by web users. Images cover a broad spectrum of human activities, events, and scenes. For fair comparison, we adopt the split provided by [47] that uses 28,783 images for training, 1,000 images for validation, and 1,000 images for testing.

4.1.2 Evaluation Metrics. The image-text matching task includes two sub-tasks, namely sentence retrieval (image query) and image retrieval (sentence query). Recall@K (R@K) is usually used to measure the performance of the retrieval task, which is defined as the probability of ground-truth matching appearing in the top K-ranked candidates. K is set to 1,5, and 10 in all experiments.

4.1.3 Implementation Details. For the node embedding of the scene graph, except for the visual object node, we set the word embedding size as 300, which is initialized with pretrained Glove embeddings [31]. For the visual object embedding, we use Faster-RCNN pretrained on MS-COCO dataset to obtain the object features and the dimensions of it is 2,048. The dimensions of GCN, HGCN and Self-Att are set to 512. The number of layers of GCN, HGCN and Self-Att are set to two, two and one, respectively.

The implementation of our proposed method is based on the Pytorch framework with a NVIDIA 3090 GPU. We utilize Adam [17] to optimize our model. For training, we set the margin γ = 0.2, and train the model with 20 epochs and 30 epochs with mini-batch size of 128 for MS-COCO and Flickr30K, respectively. We set λ = 0.5 in Eq. 16 for trading-off the effect between the ranking loss and consistence attention loss.

Table 2: Comparison with the state-of-the-art methods on MS-COCO 5K test set.

Methods	Sentence Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
$DVSA_{cvpr15}$	11.8	32.5	45.4	8.9	24.9	36.3
$VSE++_{bmvc18}$	41.3	71.1	81.2	30.3	59.4	72.4
$CAMP_{iccv19}$	50.1	82.1	89.7	39.0	68.9	80.2
$SCAN_{eccv18}$	50.4	82.2	90.0	38.6	69.3	80.4
$CAAN_{cvpr20}$	52.5	83.3	90.9	41.2	70.3	82.9
$CAMERA_{mm20}$	53.1	81.3	89.8	39.0	70.5	81.5
$VSRN_{iccv19}$	53.0	81.1	89.4	40.5	70.6	81.1
$IMRAM_{cvpr20}$	53.7	83.2	91.0	39.7	69.1	79.8
$MMCA_{cvpr2020}$	54.0	82.5	90.7	38.7	69.7	80.8
CSCC(our)	**55.6**	**83.6**	**91.2**	**40.8**	**73.2**	**84.3**

4.2 Performance Comparisons

Comparing Methods. We make a comparison with the several state-of-the-art methods in image-text matching: DVSA [15], DSPE [38], VSE++ [6], VSRN [20], SCAN [19], CAMP [41], BFAN [25], PFAN [40], CAMERA [42], MMCA [32], CAAN [49], IMRAM [3], and GSMN [26]. We directly quote the results from their original paper. Some of them are ensemble models, which average similarity scores of multiple single models trained independently (* denotes an ensemble model). Note that we only report the results of our method based on single model which achieves the superior results. It can be easily applied to ensemble model.

Comparisons on MS-COCO. Table 1 and 2 report the experimental results on MS-COCO dataset with 1K and 5K test images, respectively. For 1K test images dataset in Table 1, we can see that our proposed *CSCC* model exceeds all previous single model based methods with the best R@1=78.8% for sentence retrieval and R@1=66.6% for image retrieval, as well as achieves comparable results compared with ensemble models. In particular, compared with the GSMN which owns the same backbone with our model, our method obtains 4.1% and 6.6% relative gains on R@1 for sentence and image retrieval, respectively. It is noting that our method still achieves better performance on most of the evaluation metrics compared to the ensemble model of GSMN, and in particularly increases R@1 by 3.3% for image retrieval. For 5K test images in Table 2, our proposed method maintains the superiority with an improvement of 1.6% and 2.1% on the R@1 results for sentence and image retrieval, respectively. It clearly demonstrates the effectiveness of our method.

Comparisons on Flickr30K. To demonstrate the robustness of our method, we further provide quantitative results on Flickr30K datasets in Table 1. We can find that our CSCC model maintains relatively comparable performance compared to the existing methods. It indicates that it is beneficial to combine the conceptual cross-modal alignment with syntactical one for image-text matching. Specifically, although the improvement in Flickr30K is not so obvious than in MS-COCO dataset, our method gets the third place conditioned on the single model for sentence retrieval and achieves the state-of-the-art performance with R@1 of 61.2% for image retrieval.

Table 3: Ablation studies in the MS-COCO 1K test set. *CCA*, *CSTA*, and *CC* denote conceptual cross-modal alignment, syntactical cross-modal alignment and cross-level consistency, respectively. "w/o" means to remove a sub-module from the basic model.

Ablation Models	MS-COCO 1K					
	Sentence Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
CCA	77.5	95.2	97.8	65.0	91.4	94.9
SCA (w/o GCN)	77.4	95.0	98.0	65.2	91.6	95.0
SCA (w/o HGCN)	77.6	95.2	98.1	65.3	91.4	95.6
SCA	78.0	95.4	98.2	65.7	91.8	95.8
CCA+SCA (w/o GCN)	77.6	95.1	98.2	65.4	91.5	95.5
CCA+SCA (w/o HGCN)	77.8	95.2	98.1	65.3	91.7	95.2
CCA+SCA	78.6	95.8	98.6	66.2	92.2	96.1
CCA+SCA+CC	78.8	96.1	99.0	66.6	92.5	96.4

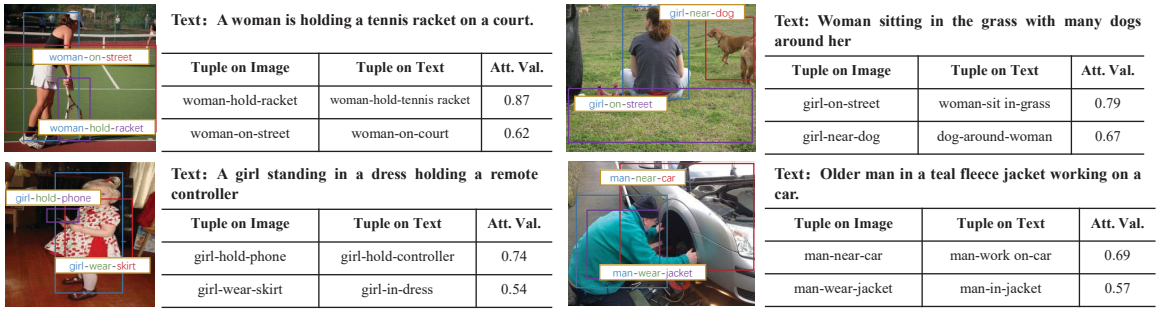
4.3 Ablation Study

The core of our proposed CSCC is to simultaneously explore the multiple-level cross-modal alignments across concept and syntactic with a consistency constraint. To verify the effectiveness of each component (CCA, SCA, and CC), we conduct the related ablation experiments based on the below settings individually shown in Table 3:

(1) *CCA*. This model is our baseline model, which only performs the conceptual cross-modal alignment (*CCA*).
(2) *SCA*. This model only performs the syntactical cross-modal alignment (*SCA*). On the basis of this, SCA (w/o GCN) and SCA (w/o HGCN) mean that we replace the GCN and HGCN with different fully connected layers to obtain the concept-level embedding and syntactic tuple-level embedding, respectively.
(3) *CCA + SCA*. This model is trained by combining the conceptual cross-modal alignment with the syntactical one, which aims to validate the effect of multi-level cross-modal alignments. Besides, we also verify the validity of GCN and HGCN condition on *CCA + SCA*,.
(4) *CCA+SCA+CC*. Here, *CC* denotes the cross-level consistency. We add the *CC* to the *CCA + SCA*, which aims at building the correlation between the cross-attentions obtained from *CCA* and *SCA*. The model is our final model, denoted as *CSCC*.

From Table 3, we have the following observations:

- Among all the above settings, CCA performs the worst, but it also exceeds most of comparing methods shown in Table 1.
- Compared with *CCA*, the result shows that *SCA* performs better on both sentence and image retrieval tasks, with an increase of 0.5% on R@1 and 0.7% on R@1, respectively. It reveals the importance of syntactical cross-modal alignment, which captures the high-level semantics information.
- *CCA + SCA* further improves the performance on the MS-COCO dataset. In terms of R@1, it surpasses the *SCA* by 0.6% and 0.5% in R@1 for sentence retrieval and image retrieval, respectively. The result shows that it is effective to integrate



Text: A woman is holding a tennis racket on a court.

Tuple on Image	Tuple on Text	Att. Val.
woman-hold-racket	woman-hold-tennis racket	0.87
woman-on-street	woman-on-court	0.62

Text: Woman sitting in the grass with many dogs around her

Tuple on Image	Tuple on Text	Att. Val.
girl-on-street	woman-sit in-grass	0.79
girl-near-dog	dog-around-woman	0.67

Text: A girl standing in a dress holding a remote controller

Tuple on Image	Tuple on Text	Att. Val.
girl-hold-phone	girl-hold-controller	0.74
girl-wear-skirt	girl-in-dress	0.54

Text: Older man in a teal fleece jacket working on a car.

Tuple on Image	Tuple on Text	Att. Val.
man-near-car	man-work on-car	0.69
man-wear-jacket	man-in-jacket	0.57

Figure 4: Visualization for co-attention matrix ATT^S of *CSCC* with four examples. Each example consists of two syntactical tuples of image, two syntactic tuples of text, and the top-2 attention values between the matched tuples.

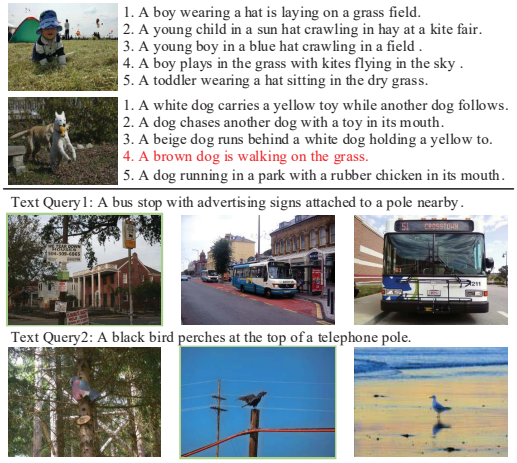


Figure 5: Visualization of sentence retrieval and image retrieval. The upper part shows the sentence retrieval results while the bottom part shows the image retrieval results. Red sentence represents the negative sample and green boxes indicate correct images.

the conceptual cross-modal alignment and syntactical cross-modal alignment.

- For the result of *CCA* + *SCA* + *CC*, the performance improvement validates the advantage of cross-level consistency *CC* to build the correlation between the cross-attentions obtained from *CCA* and *SCA*.
- In addition, we verify the influence of GCN and HGCN on concept-level and syntactic tuple-level embedding, respectively. Compared with *SCA*, *SCA* (w/o GCN) and *SCA* (w/o HGCN) both obtain a slight reduction in performance. The result of *SCA* (w/o GCN) indicates that the GCN can obtain better conceptual-level embedding by learning the relationship between the concept nodes on the scene graph. The result of *SCA* (w/o HGCN) verifies that the HGCN can better explore the interaction between the local feature and syntactic compared with the simple fully connected network. Similar experimental results also appear in the *CCA* + *SCA* condition.

4.4 Qualitative Results

To better understand the proposed syntactic-level cross-modal alignment, we visualize some syntactical-level cross-attention matrices ATT^S generated by *CSCC*, shown in Figure 4. Among all pairs of syntactic tuples between image and sentence, two pairs of tuples with top-2 attention values are presented. We can find that the generated tuples reflect the information of images or texts well. By observing the top-2 attention values, we can see that tuples with the same meaning between images and texts have high similarities. For example, at the top-left example, our model gets the attention value of 0.87 between the tuple "woman-hold-racket" on the image and the tuple "woman-hold-tennis racket" on the text.

Also, we display some results of the sentence retrieval and image retrieval shown in Figure 5. The results show that our approach always retrieves correct samples with a high rank. Although some negative samples are ranked very high, they are difficult for humans to distinguish.

5 CONCLUSION

Most successful cross-modal image-text retrieval systems are based on fine-grained correspondence. However, only fine-grained correspondence is insufficient to capture the rich semantic information of language, including concept and syntactic. Therefore, in this work, we propose a **C**onceptual and **S**yntactical Cross-modal Alignment with **C**ross-level **C**onsistency (CSCC) for image-text matching, which simultaneously explores the semantic information between the concept and syntactic. To learn the syntactic-level semantic, we devise a syntactical cross-modal alignment that computes the similarities between inter-modalities at the syntactic level. To establish the relationship of cross-level cross-modal alignment, we design a cross-level consistency to constraint the consistency between cross-modal attentions obtained from two cross-modal alignments. Superior experimental results on two image-text matching datasets demonstrate the advantages of our final model (CSCC).

ACKNOWLEDGEMENTS

This work is supported by National Key Research and Development Program of China (No. 2018AAA0102200), the National Natural Science Foundation of China (Grant No. 61772116, No. 61872064, No. 62020106008), Sichuan Science and Technology Program (Grant No. 2019JDTD0005), and Kuaishou.

REFERENCES

[1] Emily Alsentzer, Samuel G. Finlayson, Michelle M. Li, and Marinka Zitnik. 2020. Subgraph Neural Networks. In *NeurIPS*.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *ECCV*, Vol. 9909. 382–398.

[3] Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. 2020. IMRAM: Iterative Matching With Recurrent Attention Memory for Cross-Modal Image-Text Retrieval. In *CVPR*. 12652–12660.

[4] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. 2020. Say As You Wish: Fine-Grained Control of Image Caption Generation With Abstract Scene Graphs. In *CVPR*. 9959–9968.

[5] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. 2018. Visual Grounding via Accumulated Attention. In *CVPR*. 7746–7755.

[6] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*. 12.

[7] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks. In *AAAI*. 3558–3565.

[8] Lianli Gao, Pengpeng Zeng, Jingkuan Song, Yuan-Fang Li, Wu Liu, Tao Mei, and Heng Tao Shen. 2019. Structured two-stream attention network for video question answering. In *AAAI*, Vol. 33. 6391–6398.

[9] Lianli Gao, Pengpeng Zeng, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. 2018. Examine before you answer: Multi-task learning with adaptive-attentions for multiple-choice VQA. In *ACM MM*. 1742–1750.

[10] Longteng Guo, Jing Liu, Xinxin Zhu, Peng Yao, Shichen Lu, and Hanqing Lu. 2020. Normalized and Geometry-Aware Self-Attention Network for Image Captioning. In *CVPR*. 10324–10333.

[11] Yuyu Guo, Lianli Gao, Jingkuan Song, Peng Wang, Nicu Sebe, Heng Tao Shen, and Xuelong Li. 2021. Relation Regularized Scene Graph Generation. *IEEE Transactions on Cybernetics* (2021).

[12] Yuyu Guo, Jingkuan Song, Lianli Gao, and Heng Tao Shen. 2020. One-shot Scene Graph Generation. In *ACM MM*. 3090–3098.

[13] Yan Huang, Wei Wang, and Liang Wang. 2017. Instance-Aware Image and Sentence Matching with Selective Multimodal LSTM. In *CVPR*. 7254–7262.

[14] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik G. Learned-Miller, and Xinlei Chen. 2020. In Defense of Grid Features for Visual Question Answering. In *CVPR*. 10264–10273.

[15] Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*. 3128–3137.

[16] Eun-Sol Kim, Woo-Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. 2020. Hypergraph Attention Networks for Multimodal Learning. In *CVPR*. 14569–14578.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[18] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *CoRR* abs/1411.2539 (2014).

[19] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked Cross Attention for Image-Text Matching. In *ECCV*, Vol. 11208. 212–228.

[20] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual Semantic Reasoning for Image-Text Matching. In *ICCV*. 4653–4661.

[21] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Relation-Aware Graph Attention Network for Visual Question Answering. In *ICCV*. 10312–10321.

[22] Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. 2017. Identity-Aware Textual-Visual Matching with Latent Co-attention. In *ICCV*. 1908–1917.

[23] Xiangpeng Li, Lianli Gao, Xuanhan Wang, Wu Liu, Xing Xu, Heng Tao Shen, and Jingkuan Song. 2019. Learnable aggregating net with diversity learning for video question answering. In *ACM MM*. 1166–1174.

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, Vol. 8693. 740–755.

[25] Chunxiao Liu, Zhendong Mao, An-An Liu, Tianzhu Zhang, Bin Wang, and Yongdong Zhang. 2019. Focus Your Attention: A Bidirectional Focal Attention Network for Image-Text Matching. In *ACM MM*. 3–11.

[26] Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. 2020. Graph Structured Network for Image-Text Matching. In *CVPR*. 10918–10927.

[27] Yu Liu, Yanming Guo, Erwin M. Bakker, and Michael S. Lew. 2017. Learning a Recurrent Residual Fusion Network for Multimodal Matching. In *ICCV*. 4127–4136.

[28] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL*. 55–60.

[29] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. In *CVPR*. 2156–2164.

[30] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. 2017. Hierarchical Multimodal LSTM for Dense Visual-Semantic Embedding. In *ICCV*. 1899–1907.

[31] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[32] Leigang Qu, Meng Liu, Da Cao, Liqiang Nie, and Qi Tian. 2020. Context-Aware Multi-View Summarization Network for Image-Text Matching. In *ACM MM*. 1047–1055.

[33] Fumin Shen, Xiang Zhou, Yang Yang, Jingkuan Song, Heng Tao Shen, and Dacheng Tao. 2016. A Fast Optimization Method for General Binary Code Learning. *IEEE Trans. Image Process.* 25, 12 (2016), 5610–5621.

[34] Jingkuan Song, Pengpeng Zeng, Lianli Gao, and Heng Tao Shen. 2018. From pixels to objects: cubic visual attention for visual question answering. In *IJCAI*. 906–912.

[35] Jingkuan Song, Hanwang Zhang, Xiangpeng Li, Lianli Gao, Meng Wang, and Richang Hong. 2018. Self-Supervised Video Hashing With Hierarchical Binary Auto-Encoder. *IEEE Trans. Image Process.* 27, 7 (2018), 3210–3221.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*. 5998–6008.

[37] Jing Wang, Jinhui Tang, and Jiebo Luo. 2020. Multimodal Attention with Image Text Spatial Relationship for OCR-Based Image Captioning. In *ACM MM*. 4337–4345.

[38] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. 2019. Learning Two-Branch Neural Networks for Image-Text Matching Tasks. *IEEE Trans. PAMI* 41, 2 (2019), 394–407.

[39] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. 2019. Neighbourhood Watch: Referring Expression Comprehension via Language-Guided Graph Attention Networks. In *CVPR*. 1960–1968.

[40] Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. 2019. Position Focused Attention Network for Image-Text Matching. In *IJCAI*. 3792–3798.

[41] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. 2019. CAMP: Cross-Modal Adaptive Message Passing for Text-Image Retrieval. In *ICCV*. 5763–5772.

[42] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. 2020. Multi-Modality Cross Attention Network for Image and Sentence Matching. In *CVPR*. 10938–10947.

[43] Xiangping Wu, Qingcai Chen, Wei Li, Yulun Xiao, and Baotian Hu. 2020. AdaHGNN: Adaptive Hypergraph Neural Networks for Multi-Label Image Classification. In *ACM MM*. 284–293.

[44] Xing Xu, Huimin Lu, Jingkuan Song, Yang Yang, Heng Tao Shen, and Xuelong Li. 2020. Ternary Adversarial Networks With Self-Supervision for Zero-Shot Cross-Modal Retrieval. *IEEE Trans. Cybern.* 50, 6 (2020), 2400–2413.

[45] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-Encoding Scene Graphs for Image Captioning. In *CVPR*. 10685–10694.

[46] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring Visual Relationship for Image Captioning. In *ECCV*, Vol. 11218. 711–727.

[47] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL* 2 (2014), 67–78.

[48] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *CVPR*. 5831–5840.

[49] Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z. Li. 2020. Context-Aware Attention Network for Image-Text Retrieval. In *CVPR*. 3533–3542.