Memorize, Associate and Match: Embedding Enhancement via Fine-Grained Alignment for Image-Text Retrieval

Jiangtong Li, Liu Liu, Li Niu, and Liqing Zhang, *Member, IEEE*

***Abstract*— Image-text retrieval aims to capture the semantic correlation between images and texts. Existing image-text retrieval methods can be roughly categorized into embedding learning paradigm and pair-wise learning paradigm. The former paradigm fails to capture the fine-grained correspondence between images and texts. The latter paradigm achieves fine-grained alignment between regions and words, but the high cost of pair-wise computation leads to slow retrieval speed. In this paper, we propose a novel method named MEMBER by using Memory-based EMBedding Enhancement for image-text Retrieval (MEMBER), which introduces global memory banks to enable fine-grained alignment and fusion in embedding learning paradigm. Specifically, we enrich image (*resp.*, text) features with relevant text (*resp.*, image) features stored in the text (*resp.*, image) memory bank. In this way, our model not only accomplishes mutual embedding enhancement across two modalities, but also maintains the retrieval efficiency. Extensive experiments demonstrate that our MEMBER remarkably outperforms state-of-the-art approaches on two large-scale benchmark datasets.**

***Index Terms*— Image-text retrieval, memory network, attention mechanism, transformer.**

I. Introduction

RECENTLY, with the rapid growth of multimedia data on the internet, vision and natural language have become the main aspects for artificial intelligence to recognize our world. To bridge the gap between these two modalities, cross-modal modeling, including image-text retrieval [1]–[4], image captioning [5], visual question answering [6], and visual commonsense reasoning [7], has drawn more attention from both academia and industry. Image-text retrieval is one of the fundamental tasks, aiming to capture correspondence between images and texts. Researchers have proposed lots of works and made great progress in this task. Existing works can be roughly categorized into two groups: embedding learning methods and pair-wise learning methods.

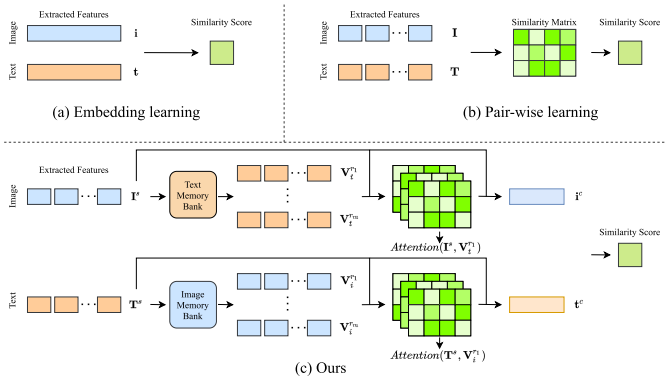

Fig. 1. Illustrative Figure of (a) embedding learning paradigm, (b) pair-wise learning paradigm and (c) our method.

As a straightforward solution, early works attempted to directly map images and texts from different modalities to a shared embedding space by enforcing constraints such as triplet ranking loss [2] or correlation maximization [8], which belong to embedding learning paradigm and are illustrated in Figure 1 (a). This type of works learned global representations within each modality and used different techniques like attention mechanism [9] or graph convolution networks [10] to filter out irrelevant information, and then calculated the similarity matrix through euclidean distance or cosine distance. However, such methods can only capture coarse correspondence between images and texts. Therefore, they work well in simple retrieval scenarios, but are not suitable in more realistic cases that involve multiple objects.

To learn fine-grained correspondence, recent research works further explored to perform fine-grained alignment between regions and words, which belong to pair-wise learning paradigm and are illustrated in Figure 1 (b). Lee *et al.* [3] built object-level correspondence and adopted stacked cross attention to align regions and words. To further enhance the cross-modal interaction, Chen *et al.* [4] proposed IMRAM to iteratively match each image-text pair with recurrent attention mechanism. Chen and Luo [11] proposed to aggregate the affinity between regions and words in each image-text pair. These methods are capable of capturing region-word alignment between images and texts by a complex fine-grained alignment

Manuscript received August 17, 2020; revised July 7, 2021; accepted October 6, 2021. Date of publication November 5, 2021; date of current version November 11, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100704, in part by the NSF of China under Grant 62076162, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102, and in part by the Shanghai Municipal Science and Technology Key Project under Grant 20511100300. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julian Fierrez. *(Corresponding authors: Li Niu; Liqing Zhang.)*

The authors are with the MOE Key Laboratory of Artificial Intelligence, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: keep_moving-lee@sjtu.edu.cn; shirlley@sjtu.edu.cn; ustcnewly@sjtu.edu.cn; zhang-lq@cs.sjtu.edu.cn).

Digital Object Identifier 10.1109/TIP.2021.3123553

scorer, and have achieved state-of-the-art performance on several benchmark datasets. Unfortunately, the retrieval speed of these methods has been slowed down largely. One of the reasons is that the fine-grained alignment requires calculating all the matching relations between every region and every word in each image-text pair. Besides, the number of all image-text pairs is huge, which makes this process quite time-consuming.

Comparing the embedding learning paradigm and the pair-wise learning paradigm, we conclude that the fine-grained alignment across modalities is essential for the performance of image-text retrieval, however, inefficient pair-wise scoring in pair-wise learning paradigm leads to slow retrieval speed. Besides, scoring the images and texts in a shared embedding space is the key to accelerate retrieval speed, whereas the coarse correspondence captured by embedding learning methods limits its application to complex scenarios with multiply objects. To combine the advantages from both learning paradigms, we attempt to enhance the embedding represenation via fine-grained alignment. For this purpose, as illustrated in Figure 1 (c), we propose Memory-based EMBedding Enhancement for image-text Retrieval (MEMBER) method, which integrates key-value memory banks to help our model perform fine-grained alignment and fusion in the embedding learning paradigm. Unlike previous embedding learning methods, our model learns two types of embeddings: self-embedding and cross-embedding, where self-embedding is generated within each modality and cross-embedding is generated by interacting with the cross-modal memory bank.

Corresponding to two types of embeddings, our proposed MEMBER method has two stages: self-learning stage and memory-based cross-learning stage, as illustrated in Figure 2. In self-learning stage, we first extract region (*resp.*, token) features from each image (*resp.*, text), and then adopt a siamese transformer to encode region and token features into corresponding self-features. After that, they are compacted to self-embeddings to facilitate retrieval. We design a key-value image (*resp.*, text) memory bank, where each key-value pair is self-embedding and self-features for images (*resp.*, texts) in the whole training set. Then, we associate each image (*resp.*, text) with relevant cross-modal information in the memory bank to enhance the image (*resp.*, text) embedding. Specifically, given an image (*resp.*, a text), we use its compact self-embedding to search the text (resp., image) memory bank for relevant text (resp., image) self-features. Then, we perform fine-grained alignment and fusion between its self-features with the relevant cross-modal self-features. Finally, we perform retrieval in both self-embedding space and cross-embedding space. Although fine-grained alignment and fusion is required in this stage, it is only performed between each image (*resp.*, text) and a few relevant texts (*resp.*,, images), which is different from and faster than pair-wise learning methods. (see Sec.IV-E)

Our cross-learning stage is inspired by cognitive science [12]. When seeing a new sentence, people may first focus on a related topic or experience in the memory, associate with some related scene fragments from the topic or experience, link some noun/verb groups with these scene fragments, and understand this sentence better by combining all the information together. This process coincides with the "memorize and associate" behavior of our model, that is, utilizing the key-value memory banks to recall relevant fine-grained features and adopting the fine-grained alignment and fusion for embedding enhancement. Note that some works [13], [14] also adopted memory bank to help image-text retrieval. Song *et al.* [13] applied category-based memory in image-text retrieval, where the category information is unavailable in our situation. Ji *et al.* [14] only used several memory slots to restore and forget the batch information, which only utilized the representation of previous few batches and the memory slots lack explicit meanings. In contrast, our memory banks hold global memory with explicit meanings, *i.e.*, pairs of compact self-embedding and self-features. Besides, the function of our memory banks is to help images (*resp.*, texts) extract useful cross-modal information from relevant texts (*resp.*, images), which is also different from previous works [13], [14].

The effectiveness of our proposed MEMBER method is verified by comprehensive experimental results on two benchmark datasets. Our main contributions are summarized as follows:

- To combine the advantages of embedding learning and pair-wise learning paradigms, we integrate fine-grained alignment into embedding learning paradigm.
- We propose a novel MEMBER method, which utilizes global memory to accomplish fine-grained alignment and fusion for mutual embedding enhancement.
- Comprehensive experiments on two large-scale benchmark datasets reveal that our method significantly outperforms the state-of-the-art methods.

II. Related Work

A. Image-Text Retrieval

The key issue of image-text retrieval is to measure the semantic similarity between a text and an image. For this purpose, existing works can be categorized into two groups, embedding learning methods [2], [15], [16] and pair-wise learning methods, [11], [17]–[19].

The embedding learning methods aim to learn a modal-invariant and representative embedding for each image and text. Rasiwasia *et al.* [8] proposed Canonical Correlation Analysis (CCA) to optimize the statistical values to learn linear projection matrices, which motivates many follow-up works [20]–[22] to learn more accurate projection matrices for better correlation performance. Kiros *et al.* [23] adopted the hinge-based triplet loss to learn the image and text embeddings in a shared space. Faghri *et al.* [2] paid attention to the hardest negative with the triplet ranking loss. He *et al.* [16] combined classification loss, clustering loss and ranking loss together, along with a new proposed benchmark, which performed retrieval among three different modalities. Wu *et al.* [15] applied self-attention layers to discover the relationships among regions (*resp.*, words) in images (*resp.*, texts). Li *et al.* [10] performed reasoning with Graph Convolutional Networks [24] to generate features with semantic relationships.

The pair-wise learning methods aim to calculate the similarity between each image-text pair more accurately with fine-grained alignment. Karpathy *et al.* [1] extracted objects from

images, and matched them with words in texts to explore the fine-grained image-text correspondence. Huang *et al.* [25] proposed a cross-modal attention to selectively attend to several pairs of instances of images and texts, by predicting pairwise instance-aware saliency maps. Wu *et al.* [26] proposed an online learning method to learn the similarity function across modalities. Peng *et al.* [27] paid much attention on unsupervised image-text retrieval, which combined the image-to-text translation and fine-grained alignment together to capture the image-text correspondence under unsupervised manner. He and Peng [28] proposed a fine-grained visual-textual representation method, where the text attention was used to discover discriminative visual-textual pairwise information for boosting categorization performance and the intra-modality and inter-modality information was also preserved to generate complementary fine-grained representation To capture structure information in images and texts, Wang *et al.* [17] designed two particular scene graph encoders and explored the graph matching from both object-level and relationship-level. Liu *et al.* [29] utilized extra information (*i.e.*, the text semantic parsing labels) to parse images and texts into graphs, and adopted the graph structured network to match them.

Cross attention is also widely used in pair-wise learning methods to boost the fine-grained alignment between images and texts. Peng *et al.* [30] proposed recurrent cross-attention network to capture modality-specific cross-modal similarity. Huang *et al.* [31] designed bi-directional cross-attention network to explore the spatial-semantic relation for image-text retrieval. Lee *et al.* [3] obtained the image (*resp.*, text) features by attending each region (*resp.*, word) feature to all word (*resp.*, region) features. To utilize multi-level visual-textual alignment, Peng *et al.* [32] proposed MAVA to incorporate local-level, global-level, and relation-level information together. Besides, with the help of the cross attention networks, Chen *et al.* [4] proposed IMRAM to match fragments across different modalities iteratively. DP-RNN [11] utilized the similarity scores to enhance the final features. Note that, these models can capture the image-text correspondences well by fine-grained alignment, but their retrieval speed will be very slow when the retrieval space is large, which makes them unsuitable for real-world application. Unlike these models, our method provides a new perspective: fine-grained alignment and fusion for mutual embedding enhancement, which can maintain relatively fast retrieval speed.

B. Memory-Enhanced Network

Memory-enhanced network was first proposed by Weston *et al.* [33] to enhance the network's long-term memory capability by augmenting it with a series of extra memory components, where the memory components can be read and written to store input facts and to retrieve supporting facts given an input query. Sukhbaatar *et al.* [34] extended the idea and developed the first end-to-end memory network (MemN2N) with a recurrent attention model over a large external memory. Graves *et al.* [35] proposed Neural Turing Machine, which adopted a key-value structure to tackle the problem of sorting and recalling during memory writing and memory reading. A similar key-value memory mechanism was also adopted by Miller *et al.* [36] to utilize different encoding schemes for memory reading. To get a soft-selection over memory slots, Kim *et al.* [37] proposed a new structured attention network, which used a conditional random field to capture structural dependencies in memory slots.

Memory-enhanced networks have become popular in the fields of computer vision [38] and natural language processing [39], [40]. For example, Zhu *et al.* [41] proposed Iterative Querying Model (IQM) to encode human knowledge into an extra memory bank for more accurate reasoning. Park *et al.* [38] adopted Long-short Term Memory (LSTM) to capture the personalized feature during sequence modeling. Wang *et al.* [39] proposed to enhance the RNN decoder in neural machine translator with a pre-defined external memory, which aimed to capture relevant information during the sequence decoding. And Cheng *et al.* [40] extended the neural machine reader with an external memory network to store contextual information of input document.

Memory-enhanced network is also widely used in multi-modal modeling. For example, the stacked attention networks (SANs) [42] regarded the whole image as a memory bank and then used the text semantic representation to search for all the regions in a given image to infer the corresponding answers. Song *et al.* [13] proposed a category-based modal-shared memory bank for cross-modal retrieval. Ji *et al.* [14] restored the inter-modal and intra-modal information in the memory bank to narrow the modality gap between images and texts. Note that the memory banks used in these works either only captured category information or utilized local information. In contrast, our work not only utilizes global memory with explicit meanings, but also facilitates fine-grained alignment and fusion between two modalities.

III. Methodology

In this section, we will introduce our MEMBER method, which is short for Memory-based EMBedding Enhancement for image-text Retrieval. In Sec. III-A, we will present the problem definition and notation. In Sec. III-B, we will introduce the background knowledge of transformer, which is used multiple times in our model. In Sec. III-C, we will detail our MEMBER method, revealing how to utilize global cross-modal information efficiently.

A. Problem Definition

Suppose we have a set of training images $\{\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{N_i}\}$ and a set of training texts $\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{N_t}\}$ with provided matching correspondence (each image has several matched texts), where N_i and N_t are the number of images and texts, respectively. Our method builds global memory banks based on the training images and texts, and learns self-embeddings and cross-embeddings of images/texts. In the test stage, given images and texts, we can obtain their self-embeddings and cross-embeddings, based on which the similarity of each image-text pair is calculated to perform retrieval. For clarity, in the rest part of this paper, we will omit the index number of images/texts, and all the similarity is measured under cosine

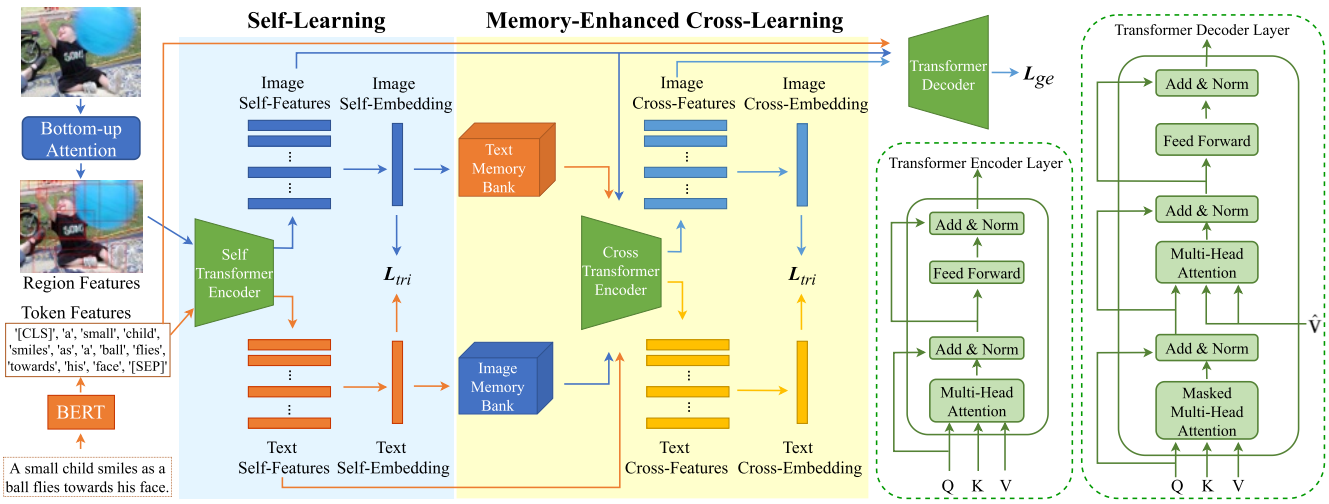

Fig. 2. The flowchart of our MEMBER method. We first adopt BERT [43] and Faster R-CNN [44] (bottom-up attention [45]) to extract token (*resp.*, region) features from texts (*resp.*, images), based on which we perform self-learning to obtain the self-embeddings. Then, memory-enhanced cross-learning is followed to obtain the improved cross-embeddings. The structures of transformer encoder and decoder layers are shown on the right.

similarity. We use $\mathbf{X}^T$ to denote the transpose of $\mathbf{X}$ and $\mathbf{1}$ to denote an all-one column vector.

The overall structure of our proposed MEMBER method is illustrated in Figure 2. We first represent each image as a sequence of region features $\mathbf{I}^e$ and each text as a sequence of token features $\mathbf{T}^e$. Through a self-transformer encoder E^s, we can obtain the image self-features $\mathbf{I}^s$ and text self-features $\mathbf{T}^s$. Then, a pooling layer is adopted to compact $\mathbf{I}^s$ (*resp.*, $\mathbf{T}^s$) into an image (*resp.*, a text) self-embedding $\mathbf{i}^s$ (*resp.*, $\mathbf{t}^s$), which is used to perform retrieval in self-embedding space and memory search. We set up an image (*resp.*, a text) memory bank which stores the image (*resp.*, text) self-embeddings and their corresponding self-features of all training images (*resp.*, texts). Although all training texts and images are stored in the memory bank by default, actually, only using around 12,000 texts and 2,400 images can achieve comparable results (see Sec. IV-D). By using text (*resp.*, image) memory bank and cross-transformer encoder E^c, we can obtain enhanced image (*resp.*, text) cross-features $\mathbf{I}^c$ (*resp.*, $\mathbf{T}^c$). Another pooling layer is adopted to compact $\mathbf{I}^c$ (*resp.*, $\mathbf{T}^c$) into an image (*resp.*, a text) cross-embedding $\mathbf{i}^c$ (*resp.*, $\mathbf{t}^c$). Finally, we perform retrieval in both self-embedding and cross-embedding spaces.

B. Background on Transformer

We use transformer encoder and decoder [46] to encode and decode sequences of features in our method, which are widely used and have achieved great success in many areas, such as language modelling [43] and cross-modal retrieval [15]. Given two sequences of features, the transformer encoder can align these two sequences and accomplish information fusion. A transformer encoder (*resp.*, decoder) contains multiple transformer encoder (*resp.*, decoder) layers, with the structure of each layer shown in Figure 2.

Each transformer encoder layer is constructed by a multi-head attention sub-layer and a feed-forward sub-layer. The multi-head attention sub-layer takes queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ as input. $\mathbf{Q} \in \mathbb{R}^{n_q \times d}$, $\mathbf{K} \in \mathbb{R}^{n_k \times d}$, and $\mathbf{V} \in \mathbb{R}^{n_v \times d}$ are all sequences of features, where d is the feature dimension, n_q, n_k, and n_v are the length of queries, keys, and values, respectively. In practice, $\mathbf{K}$ is usually identical with $\mathbf{V}$, *i.e.*, $\mathbf{K} = \mathbf{V}$. According to whether $\mathbf{Q}$ is identical with $\mathbf{V}$, we can divide transformer encoder layer into self-transformer encoder layer ($\mathbf{Q} = \mathbf{V}$) and cross-transformer encoder layer ($\mathbf{Q} \neq \mathbf{V}$). For each query in $\mathbf{Q}$, the attention sub-layer calculates its similarities with all keys in $\mathbf{K}$, and obtains the weighted average of corresponding values in $\mathbf{V}$ as the attended value. Besides, transformer encoder layer employs the multi-head attention mechanism, which calculates the attended values based on multiple projections. Specifically, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are projected to lower dimension h times respectively using h projection matrices. Then h attention weight $A_j \ \forall j \in [1, h]$ is calculated to produce the attended values $H_j \ \forall j \in [1, h]$:

$$\begin{aligned} \mathbf{A}_j &= Softmax(\frac{(\mathbf{Q}\mathbf{W}_j^Q)(\mathbf{K}\mathbf{W}_j^K)^T}{\sqrt{\hat{d}}}), \\ \mathbf{H}_j &= \mathbf{A}_j \mathbf{V}\mathbf{W}_j^V, \end{aligned} \tag{1}$$

in which $\mathbf{W}_j^Q$, $\mathbf{W}_j^K$, $\mathbf{W}_j^V \in \mathbb{R}^{d \times \hat{d}}$ are projection matrices with $\hat{d} = \frac{d}{h}$. Then, h attended values can be obtained by

$$MH(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{H}_1, \ldots, \mathbf{H}_h]\mathbf{W}^O, \tag{2}$$

where $\mathbf{W}^O \in \mathbb{R}^{h\hat{d} \times d}$ is a projection matrix, $[\mathbf{H}_1, \ldots, \mathbf{H}_h]$ means concatenation, and $MH(\cdot, \cdot, \cdot)$ is short for Multi-Head.

Then a feed-forward sub-layer is applied on the top of the multi-head attention sub-layer, which consists of two linear transformations with a ReLU activation between them. For different positions, they use the same linear transformations, while the parameters from layer to layer are different. Moreover, residual connections are employed around both multi-head attention sub-layer and feed-forward sub-layer, followed by layer normalization [47]. Based on $MH(\mathbf{Q}, \mathbf{K}, \mathbf{V})$,

the entire transformer encoder layer can be formulated as

$$\mathbf{X} = LN(\mathbf{Q} + MH(\mathbf{Q}, \mathbf{K}, \mathbf{V})), \quad (3)$$

$$FFN(\mathbf{X}) = \max(0, \mathbf{X}\mathbf{W}_1 + \mathbf{1}\mathbf{b}_1^T)\mathbf{W}_2 + \mathbf{1}\mathbf{b}_2^T, \quad (4)$$

$$TFE(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = LN(\mathbf{X} + FFN(\mathbf{X})), \quad (5)$$

where $\mathbf{X} \in \mathbb{R}^{n_q \times d}$, $\mathbf{W}_1 \in \mathbb{R}^{d \times \bar{d}}$, $\mathbf{W}_2 \in \mathbb{R}^{\bar{d} \times d}$, $\mathbf{b}_1 \in \mathbb{R}^{\bar{d} \times 1}$, $\mathbf{b}_2 \in \mathbb{R}^{d \times 1}$. $LN(\cdot)$ is short for Layer Normalization. And $TFE(\cdot, \cdot, \cdot)$ represents a TransFormer Encoder layer, with the output size being $n_q \times d$, completing information fusion between $\mathbf{Q}$ and $\mathbf{V}$. Intuitively, for self-transformer encoder layer, we represent query $\mathbf{Q}$ by itself; for the cross-transformer encoder layer, we represent query $\mathbf{Q}$ by a different value $\mathbf{V}$.

Transformer encoder is formed by stacking multiple transformer encoder layers $TFE(\cdot, \cdot, \cdot)$, in which the output of previous layer is replicated as the input queries, keys, and values for next layer. We use transformer encoder in both self-learning (Sec. III-C.1) and cross-learning stage (Sec. III-C.2).

The transformer decoder shares a similar structure as the transformer encoder, except an extra multi-head attention sub-layer with side input $\hat{\mathbf{V}}$ shown in Figure 2. We use transformer decoder to generate texts from image features (Sec. III-C.3). For more details of transformer, refer to Vaswani *et al.* [46].

C. Our Method

In this section, we will introduce our self-learning stage in Sec. III-C.1 and memory-enhanced cross-learning stage in Sec. III-C.2. Then, we will describe our loss function in Sec. III-C.3 and discuss our retrieval strategy in Sec. III-C.4

1) Self-Learning Stage: Given an image $\mathbf{x}_i$ and a text $\mathbf{x}_t$ as a pair of inputs, we use different feature extractors to represent each of them as a sequence of feature vectors.

a) Region features extraction: To capture the fine-grained region information in each image, we employ bottom-up attention [45] to extract convolutional feature for each image region. Specifically, we follow Lee *et al.* [3] and use the Faster R-CNN model [44] to extract the region features. Therefore, an image is represented as a sequence of image region features $\mathbf{I}^e \in \mathbb{R}^{n_i \times d_i} = [\mathbf{i}_1^e, \ldots, \mathbf{i}_{n_i}^e]$ ordered by confidence score, where n_i is the number of regions and d_i is region feature dimension.

b) Token features extraction: Motivated by the improvement achieved in the natural language processing, we apply the transformer encoder to extract word features of each text, which are rich in semantics. In particular, a pre-trained BERT [43] is employed to generate context-sensitive token features. Through this model, we can represent each text as a sequence of features $\mathbf{T}^e \in \mathbb{R}^{n_t \times d_t} = [\mathbf{t}_1^e, \ldots, \mathbf{t}_{n_t}^e]$, where n_t is the number of tokens and d_t is token feature dimension.

c) Self encoding: To encourage the information sharing among regions (*resp.*, words) within each sequence of image region (*resp.*, text token) features, we apply a self-transformer encoder introduced in Sec. III-B to learn better image (*resp.*, text) features. First, we project region features and token features into the same dimension d, which is formulated as

$$\mathbf{I}^d = \mathbf{I}^e\mathbf{W}_i + \mathbf{1}\mathbf{b}_i^T, \quad \mathbf{T}^d = \mathbf{T}^e\mathbf{W}_t + \mathbf{1}\mathbf{b}_t^T, \quad (6)$$

where $\mathbf{W}_i \in \mathbb{R}^{d_i \times d}$, $\mathbf{b}_i \in \mathbb{R}^{d \times 1}$, $\mathbf{W}_t \in \mathbb{R}^{d_t \times d}$, $\mathbf{b}_t \in \mathbb{R}^{d \times 1}$.

Then, we employ a self-transformer encoder E^s, where the input query, key, and value are all $\mathbf{I}^d$ (*resp.*, $\mathbf{T}^d$), and obtain the image (*resp.*, text) self-features $\mathbf{I}^s$ (*resp.*, $\mathbf{T}^s$):

$$\mathbf{I}^s = E^s(\mathbf{I}^d, \mathbf{I}^d, \mathbf{I}^d), \quad \mathbf{T}^s = E^s(\mathbf{T}^d, \mathbf{T}^d, \mathbf{T}^d). \quad (7)$$

In detail, given the projected region features $\mathbf{I}^d$ (*resp.*, projected token features $\mathbf{T}^d$), for each item in $\mathbf{I}^d$ (*resp.*, $\mathbf{T}^d$), we calculate its attention weights with all the items in $\mathbf{I}^d$ (*resp.*, $\mathbf{T}^d$), and obtain the attended values (see Equ. (1) and (2)). Then, we fuse the attended values with $\mathbf{I}^d$ (*resp.*, $\mathbf{T}^d$) to enhance itself (see Equ. (3), (4), and (5)). In this way, E^s encourages information propagation among region (*resp.*, token) features within each image (*resp.*, text) to learn better image (*resp.*, text) self-features $\mathbf{I}^s$ (*resp.*, $\mathbf{T}^s$) [46]. Regularly, we employ different transformer encoders for images and texts separately. To save parameters, motivated by Chopra *et al.* [48], we utilize a siamese self-transformer encoder, where images and texts features share a same transformer encoder.

To obtain compact representation, a pooling layer is employed to compact image (*resp.*, text) self-features into an image (*resp.*, text) self-embedding $\mathbf{i}^s$ (*resp.*, $\mathbf{t}^s$). We design three strategies for this pooling layer:

- first: select the first vector in image (*resp.*, text) self-features as the self-embedding [43].
- mean: average all vectors in image (*resp.*, text) self-features as the image (*resp.*, text) self-embedding.
- max: aggregate the image (*resp.*, text) self-features by choosing the max value in each dimension as the image (*resp.*, text) self-embedding.

We employ triplet loss to pull close the self-embeddings of matched images and texts (see Sec. III-C.3), so that self-embeddings can be used for memory search in Sec. III-C.2 and the final retrieval.

2) Memory-Enhanced Cross-Learning Stage: Given image (*resp.*, text) self-embeddings and self-features, we attempt to enhance them with cross-modal information. For this purpose, we construct an image (*resp.*, text) memory bank to store the global information of all training images (*resp.*, texts). In the following, we will take the text memory bank as an example. For the j-th slot in text memory bank, the key $\mathbf{k}_t^j$ is the text self-embedding of the j-th training text and the value $\mathbf{V}_t^j$ is the corresponding text self-features. Given an input image, we intend to learn its cross-embedding enhanced by relevant fine-grained text information in the text memory bank. In particular, we first search relevant texts in the text memory bank, and then use the self-features of relevant texts to enrich the self-features of this image via fine-grained alignment and fusion.

a) Memory update: The text self-embeddings and self-features are consistently learned during training. In order to update the memory bank in time and avoid re-calculating self-embeddings/features, we choose to continuously replace the key-value pairs in the memory bank with the self-embeddings and self-features after each training step.

b) Memory query and response: Given image self-embedding $\mathbf{i}^s$ of input image $\mathbf{x}_i$, we first calculate the cosine

similarity between $\mathbf{i}^s$ and each key $\mathbf{k}_t^j$ in this memory bank, where $\cos(\mathbf{i}^s, \mathbf{k}_t^j) = \frac{\mathbf{i}^s \cdot \mathbf{k}_t^j}{||\mathbf{i}^s||_2 ||\mathbf{k}_t^j||_2}$. Then we select the top n_m most relevant texts (the key-value pairs) in the memory bank as the memory responses $\{(\mathbf{k}_t^{r_i}, \mathbf{V}_t^{r_i})|_{i=1}^{n_m}\}$. Note that in the test stage, a test image does not have matched texts in the text memory bank. To enhance the generalization ability to test set, we filter out the matching pairs before getting the top n_m memory responses during training. After that, each memory response weight is calculated by

$$\alpha_j^m = \frac{\exp(\cos(\mathbf{i}^s, \mathbf{k}_t^j))}{\sum_{i=1}^{n_m} \exp(\cos(\mathbf{i}^s, \mathbf{k}_t^{r_i}))}. \tag{8}$$

Note that although we use global memory, the query and response are based on self-embeddings, which makes this step very efficient even for potentially large training set. We also try storing partial training set in memory banks, which our method is still effective with small memory bank (see Sec. IV-D).

c) Cross encoding: Now for input image $\mathbf{x}_i$, we have its image self-features $\mathbf{I}^s$, memory response values $\mathbf{T}_r^m = \{\mathbf{V}_t^{r_1}, \ldots, \mathbf{V}_t^{r_{n_m}}\}$, and associated memory response weights $\{\alpha_1^m, \ldots, \alpha_{n_m}^m\}$. We design two fusion strategies, *i.e.*, early-fusion and late-fusion, to fuse the information of image self-features and text memory responses. The difference lies in whether we perform cross-modal information fusion before or after weighted average. For early-fusion strategy, we first calculate the weighted average of memory response values $\mathbf{T}_r^m$ based on α^m. Then we obtain image cross-features by

$$\mathbf{T}^m = \sum_{j=1}^{n_m} \mathbf{V}_t^{r_j} * \alpha_j^m, \quad \mathbf{I}^c = E^c(\mathbf{I}^s, \mathbf{T}^m, \mathbf{T}^m), \tag{9}$$

where E^c is a cross-transformer encoder. As explained in Sec. III-B, E^c performs fine-grained alignment and fusion between $\mathbf{I}^s$ and $\mathbf{T}^m$, producing enhanced features $\mathbf{I}^c$.

Considering that $\mathbf{V}_t^{r_j}$ in Equ. (9) have different length and simply averaging the memory response values may be ineffective, we introduce the late-fusion strategy. Particularly, we fuse each response value with the image self-features through the cross-transformer encoder, and then calculate the weighted average of outputs as the image cross-features $\mathbf{I}^c$:

$$\mathbf{I}_j^c = E^c(\mathbf{I}^s, \mathbf{V}_t^{r_j}, \mathbf{V}_t^{r_j}), \quad \mathbf{I}^c = \sum_{j=1}^{n_m} \mathbf{I}_j^c * \alpha_j^m. \tag{10}$$

As mentioned before, to enhance the image feature with the text memory responses, in both Equ. (9) and (10), we use the image self-feature ($\mathbf{I}^s$) as the query and use the weighted text response ($\mathbf{T}^m$) or original response value ($\mathbf{V}_t^{r_j}$) as the key and value. At length, for each item in $\mathbf{I}^s$, we first calculate the attention weight with all the items in $\mathbf{T}^m$ or $\mathbf{V}_t^{r_j}$, and obtain the attended values (see Equ. (1) and (2)). Then, we fuse the cross-modal attended values with $\mathbf{I}^s$ to enhance the image self-feature with cross-modal fine-grained alignment and fusion (see Equ. (3), (4), and (5)).

Finally, the same pooling layer in Sec. III-C.1 is employed to compact the image cross-features $\mathbf{I}^c$ into an image cross-embedding $\mathbf{i}^c$. Note that we use transformer encoder in cross-learning stage for fine-grained alignment and fusion between image self-features $\mathbf{I}^s$ and memory response values $\mathbf{T}_r^m$.

Analogous to text memory bank, we also have an image memory bank. Given an input text $\mathbf{x}_t$, we can obtain its cross-features $\mathbf{T}^c$ and cross-embedding $\mathbf{t}^c$ via memory-based cross-modal enhancement in a similar way.

3) Loss Function: To enforce the distance of matched image-text pairs closer than unmatched ones, we use triplet ranking loss [4], [10] in both self-embedding space and cross-embedding space. Following Faghri *et al.* [2], we put emphasis on the hardest negatives, *i.e.*, the negatives closest to each training query. For a positive pair $(\mathbf{i}, \mathbf{t})$, a hardest negative image embedding $\hat{\mathbf{i}}$, and a hardest negative text embedding $\hat{\mathbf{t}}$, we define the triplet loss as

$$\mathcal{L}_{tri}(\mathbf{i}, \mathbf{t}) = [\beta - S(\mathbf{i}, \mathbf{t}) + S(\mathbf{i}, \hat{\mathbf{t}})]_+ + [\beta - S(\mathbf{i}, \mathbf{t}) + S(\hat{\mathbf{i}}, \mathbf{t})]_+, \tag{11}$$

where β serves as a margin parameter and $[x]_+ = max(x, 0)$. We adopt the cosine similarity as $S(\cdot, \cdot)$. For computational efficiency, rather than selecting the hardest negatives in the entire training set, we use the hardest one in each mini-batch.

In addition, the learned image self-features and cross-features ($\mathbf{I}^s$ and $\mathbf{I}^c$) of an input image are supposed to have the ability to generate its matched text. Following Vaswani *et al.* [46], we use the transformer decoder (see Figure 2) to generate text from image features, where the input queries, keys, and values are all masked original token features $\mathbf{T}^e = [\mathbf{t}_1^e, \ldots, \mathbf{t}_{n_t}^e]$, and the side input $\hat{\mathbf{V}}$ is image self/cross-features. We introduce a generation loss by maximizing the log-likelihood of predicting matched text:

$$\mathcal{L}_{ge}(\mathbf{I}, \mathbf{T}^e) = -\sum_{k=1}^{n_t} \log p(\mathbf{t}_k^e | \mathbf{t}_1^e, \ldots, \mathbf{t}_{k-1}^e, \mathbf{I}, \theta), \tag{12}$$

where θ is the parameters of transformer decoder and $\mathbf{I}$ indicates image self/ross-features. We substitute self/cross-embeddings into Equ. (11), and substitute self/cross-features and token features into Equ. (12), leading to final loss function:

$$\mathcal{L} = \mathcal{L}_{tri}(\mathbf{i}^s, \mathbf{t}^s) + \mathcal{L}_{ge}(\mathbf{I}^s, \mathbf{T}^e) + \mathcal{L}_{tri}(\mathbf{i}^c, \mathbf{t}^c) + \mathcal{L}_{ge}(\mathbf{I}^c, \mathbf{T}^e). \tag{13}$$

4) Retrieval: We propose two strategies to integrate both self-embedding and cross-embedding space: embedding combination and similarity combination. The former strategy first adds self-embeddings and cross-embeddings to get combined-embeddings, and then calculates the similarities using the combined-embeddings. The latter strategy first calculates the similarities based on self-embeddings and cross-embeddings separately, and then averages two types of similarities.

IV. Experiment

A. Experiment Setup

1) Dataset: We evaluate our MEMBER method and all the other baselines on two large-scale benchmark image-text retrieval datasets: Microsoft COCO [50] and Flickr30K [51].

Microsoft COCO [50]: originally consists of 82,783 training images and 40,504 validation images, and each image is

TABLE I

COMPARISON WITH EXISTING MODELS ON FLICKR30K. SELF, CROSS, AND COMB REPRESENT THE RETRIEVAL PERFORMANCE ON SELF-EMBEDDING SPACE, CROSS-EMBEDDING SPACE, AND THE COMBINATION OF THESE TWO SPACES

Learning Paradigm	Method	Text Retrieval			Image Retrieval			R@sum
		R@1	R@5	R@10	R@1	R@5	R@10	
Pair-wise Learning	SCAN [3]	67.4	90.3	95.8	48.6	77.7	85.2	465.0
	CAMP [49]	68.1	89.7	95.2	51.5	77.1	85.3	466.9
	PFAN [19]	70.0	91.8	95.0	50.4	78.7	86.1	472.0
	DP-RNN [11]	70.2	91.6	95.8	**55.5**	**81.3**	**88.3**	482.7
	MAVA [32]	71.5	91.6	95.9	52.4	79.6	86.6	477.6
	IMRAM-full [4]	**74.1**	**93.0**	**96.6**	53.9	79.4	87.2	**484.2**
Embedding Learning	M3A-Net [14]	58.1	82.8	90.1	44.7	72.4	81.1	429.2
	VSRN [10]	71.3	90.6	96.0	54.7	81.8	88.2	482.6
	SAEM [15]	69.1	91.0	95.1	52.4	81.1	88.1	476.8
	MEMBER (Self)	72.1	91.8	96.2	57.1	82.1	90.1	489.4
	MEMBER (Cross)	75.3	93.1	**97.4**	58.1	83.8	90.4	498.1
	MEMBER (Comb)	**77.5**	**94.7**	97.3	**59.5**	**84.8**	**91.0**	**504.8**
	MEMBER (Comb) w/o BERT	76.1	93.9	96.7	57.5	83.3	89.9	497.4

annotated with five text descriptions. Following the split in Lee *et al.* [3], we select 5,000 validation images and 5,000 test images from the original validation set and then add the rest 30,504 images from the validation set into the training set. The testing results are reported for both averaging over 5 folds of 1K test images and directly testing on the full 5K test images as Li *et al.* [10] and Chen *et al.* [4].

Flickr30K [51]: consists of 31,000 images collected from the Flickr website. Each image corresponds to five human annotated texts. We follow the split in Lee *et al.* [3], by using 1,000 images for validation, 1,000 images for testing, and 29,000 images for training.

2) Evaluation Metrics: To compare our proposed method with state-of-the-art methods, we adopt the same evaluation metrics on both datasets as Chen *et al.* [4]. Specifically, we adopt Recall at K (R@K) to measure the performance of bi-directional retrieval tasks, *i.e.*, retrieving texts given an image (Text Retrieval) and retrieving images given a text (Image Retrieval). We report R@1, R@5, and R@10 on both datasets. To further demonstrate the effectiveness of our proposed method, we also report an extra metric "R@sum", which is the summation of all evaluation metrics as Chen *et al.* [4].

3) Implementation Details: We implement our method using PyTorch [52], which is trained on one GTX 1080 Ti GPU. We use Adam [53] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train the model for 30 epochs. For the learning rate, we start with learning rate 0.0002 and decay the learning rate by 0.1 after every 10 epochs. We use a mini-batch size of 128.

On both datasets, for token feature extraction, we follow Wu *et al.* [15] and use the pre-trained weights of BERT model [43] which has 12 transformer layers, 12 heads, 768 hidden units for each token, and 110M parameters in total; for region feature extraction, we follow Chen *et al.* [4] and use the Faster R-CNN model [44] with ResNet-101 [54] pre-trained by Anderson *et al.* [45] on Visual Genomes [55]. For transformer encoder and transformer decoder, the hidden dimension d and number of heads h are set to 512 and 4, respectively. We stack two transformer encoder (*resp.*, decoder) layers for our transformer encoders (*resp.*, decoder). Memory banks are updated during training and the memory response size n_m for both memory banks is set as 5. The pooling strategy for both self-learning and cross-learning stages is set as "max". We choose the late-fusion (see Sec. III-C.2) and similarity combination (see Sec. III-C.4) as fusion strategy and retrieval strategy. For loss function, the margin β is set as 0.05.

B. Comparison With Existing Methods

In this section, we compare our method with eight prior methods, which include three embedding learning models and five pair-wise learning models. For embedding learning models, we compare with M3A-Net [14], VSRN [10], and SAEM [15], where M3A-Net adopted a local memory bank to enhance the embedding and SAEM adopted pre-trained BERT to extract token representation. For pair-wise learning models, we compare with SCAN [3], CAMP [49], PFAN [19], DP-RNN [11], MAVA [32], and IMRAM [4], all of which use attention mechanism to enhance the pair-wise matching. Except for M3A-Net [14], all the other methods applied Faster-RCNN [44] to extract image region features. The state-of-the-art results for both learning paradigms are highlighted in bold.

The performance on Flickr30K dataset is shown in Table I, where our proposed MEMBER method outperforms all the existing methods on all evaluation metrics by a large margin. Compared with the state-of-the-art embedding learning method VSRN [10], the performance gains of our method are 6.2% on text retrieval (R@1), 4.8% on image retrieval (R@1), and 22.2% on R@sum. Besides, our method can also outperform the state-of-the-art pair-wise learning model IMRAM-full [4] by 3.4% on text retrieval (R@1), 5.6% on image retrieval (R@1), and 20.6% on R@sum.

To further reveal the effect of pre-trained BERT features, we conduct extra experiment without using the pre-trained BERT feature for text representation and the experiment results are shown in the Table I and Table II. Based on the experiment results, without the pre-trained BERT feature, our method still outperform all the methods in both embedding learning

TABLE II

COMPARISON WITH EXISTING METHODS ON MICROSOFT COCO. SELF, CROSS, AND COMB REPRESENT THE RETRIEVAL PERFORMANCE IN SELF-EMBEDDING SPACE, CROSS-EMBEDDING SPACE, AND THE COMBINATION OF THESE TWO SPACES

Learning Paradigm	Method	Text Retrieval			Image Retrieval			R@sum
		R@1	R@5	R@10	R@1	R@5	R@10	
1K								
Pair-wise Learning	SCAN [3]	72.7	94.8	98.4	58.8	88.4	94.8	507.9
	CAMP [49]	72.3	94.8	98.3	58.5	87.6	95.0	506.5
	PFAN [19]	76.5	**96.3**	**99.0**	61.6	89.6	**95.2**	**518.2**
	DP-RNN [11]	75.3	95.8	98.6	**62.5**	**89.7**	95.1	517.0
	MAVA [32]	76.4	**96.3**	98.5	60.7	89.0	95.0	515.9
	IMRAM-full [4]	**76.7**	95.6	98.5	61.7	89.1	95.0	516.6
Embedding Learning	M3A-Net [14]	70.4	91.7	96.8	58.4	87.1	94.0	498.4
	VSRN [10]	76.2	94.8	98.2	62.8	89.7	95.1	516.8
	SAEM [15]	71.2	94.1	97.7	57.8	88.6	94.9	504.3
	MEMBER (Self)	75.2	96.1	97.8	60.7	89.2	94.8	513.8
	MEMBER (Cross)	76.6	95.4	98.0	63.0	90.6	**95.8**	519.4
	MEMBER (Comb)	**78.5**	**96.8**	**98.5**	**63.7**	**90.7**	95.6	**523.8**
	MEMBER (Comb) w/o BERT	77.8	96.3	97.9	62.7	89.9	95.1	519.7
5K								
Pair-wise Learning	SCAN [3]	50.4	82.2	90.0	38.6	69.3	80.4	410.9
	CAMP [49]	50.1	82.1	89.7	39.0	68.9	80.2	410.0
	MAVA [32]	50.7	82.4	90.7	**40.2**	**70.0**	**80.6**	414.6
	IMRAM-full [4]	**53.7**	**83.2**	**91.0**	39.7	69.1	79.8	**416.5**
Embedding Learning	M3A-Net [14]	48.9	75.2	84.4	38.3	65.7	76.9	389.4
	VSRN [10]	53.0	81.1	89.4	40.5	70.6	81.1	415.7
	MEMBER (Self)	53.0	81.7	90.2	39.1	69.7	80.6	414.3
	MEMBER (Cross)	53.6	81.8	**90.3**	39.8	70.4	81.3	417.2
	MEMBER (Comb)	**54.5**	**82.3**	90.1	**40.9**	**71.0**	**81.8**	**420.6**
	MEMBER (Comb) w/o BERT	54.0	81.7	89.6	40.2	70.5	81.1	417.1

paradigm and pair-wise learning paradigm in terms of both R@1. Besides, we can find that the pre-trained BERT features have larger impact on the Flickr30K, since the data size of the Flickr30K are much less than the Microsoft COCO dataset. Furthermore, we also find that the impact of the pre-trained BERT feature is not quite large (about 1.0-2.0 % in Flickr30K and 0.5-1.0 % in Microsoft COCO), which mainly due to the following three reasons, 1) we integrate the global memory bank and the cross-model information to enhance the embedding representation, which make the information across the whole dataset well shared; 2) we follow the structure of BERT and also utilize the transformer structure in both self-learning stage and cross-learning stage, which make the information can be shared between both modalities; 3) the BERT is training in only text modality, therefore, the pre-trained BERT feature may not fit in cross-model situation.

The performance on Microsoft COCO dataset is shown in Table II. In both 1K and 5K setting, our method can outperform existing methods with a large gap on R@1. Compared with the best baselines in both learning paradigms, *i.e.* VSRN [10] and IMRAM-full [4] in 1K test set, our method achieves an improvement of 2.3% and 1.8% on text retrieval (R@1), and 0.9% and 2.0% on image retrieval (R@1), respectively.

By comparing the improvement of our method on both datasets, we find the improvement on Flickr30K is more significant. We conjecture that images in Microsoft COCO have fewer objects and simpler relations, which compromises the enhancement from fine-grained alignment and fusion. Besides, the performance of cross-embedding is always better than that of self-embedding, which indicates the effect of our memory-based cross-modal enhancement.

C. Ablation Study

By taking Flickr30K as an example, we perform a series of experiments to verify the effect of different modules in our model. In Sec. IV-C.1, we analyze the impact of different loss terms. In Sec. IV-C.2, we experiment some alternative choices on pooling strategy, fusion strategy, retrieval strategy, and token features. In Sec. IV-C.3, we study the impact of different memory response sizes. In Sec. IV-C.4, we study the effect of different memory bank arrangements. In Sec. IV-C.5, we study the effect of different fusion strategy arrangements. In Sec. IV-C.6, we study the effect on different hyper-parameters, including the hidden dimension d, the number of heads h, and the margin β. The text retrieval R@1, image retrieval R@1, and R@sum of MEMBER(Comb) are reported in this section.

1) Effect of Loss Term: As shown in Table III, the impact of $\mathcal{L}_{ge}^{c}$ is larger than that of $\mathcal{L}_{ge}^{s}$, because the quality of cross-embedding is more important for the retrieval performance. Besides, when we remove all the losses (*i.e.*, $\mathcal{L}_{tri}^{c}$ and $\mathcal{L}_{ge}^{c}$) used in cross-learning stage, we find that the performance of our self-embedding also drops about 1% and 3% in text and image retrieval respectively, because our model is trained end-to-end and the gradient from cross-learning stage can also boost the performance of self-embedding.

TABLE III

THE ABLATION STUDY OF LOSS TERMS AND TOKEN EMBEDDINGS. $\mathcal{L}^c_{tri}$, $\mathcal{L}^s_{ge}$, AND $\mathcal{L}^c_{ge}$ ARE SHORT FOR $\mathcal{L}_{tri}(\mathbf{i}^c, \mathbf{t}^c)$, $\mathcal{L}_{ge}(\mathbf{I}^s, T^e)$, AND $\mathcal{L}_{ge}(\mathbf{I}^c, T^e)$ IN EQU. (13). ✓ (*Resp.*, ×) MEANS ADDING (*Resp.*, REMOVING) THIS LOSS TERM DURING TRAINING. T IS SHORT FOR TEXT RETRIEVAL, AND I IS SHORT FOR IMAGE RETRIEVAL

$\mathcal{L}^c_{tri}$	$\mathcal{L}^s_{ge}$	$\mathcal{L}^c_{ge}$	R@1(T)	R@1(I)	R@sum
✓	×	✓	75.6	58.4	498.9
✓	✓	×	74.9	57.7	495.3
✓	×	×	74.3	56.1	493.8
×	✓	×	71.2	53.8	475.4

TABLE IV

THE ABLATION STUDY ON SOME ALTERNATIVE CHOICES OF "POOLING", "FUSION", AND "RETRIEVAL". FOR "TOKEN", BERT MEANS WE USE PRE-TRAINED BERT MODEL TO EXTRACT TOKEN FEATURES AND RANDOM MEANS WE RANDOMLY INITIALIZE TOKEN EMBEDDINGS AND UPDATE THEM DURING TRAINING

Pooling	Fusion	Retrieval	Token	R@1(T)	R@1(I)	R@sum
max	late	similarity	BERT	77.5	59.5	504.8
mean	late	similarity	BERT	75.7	58.2	498.1
first	late	similarity	BERT	73.3	55.9	486.9
max	early	similarity	BERT	75.2	58.1	496.3
max	late	embedding	BERT	76.8	58.9	501.2
max	late	similarity	Random	76.4	57.5	497.7

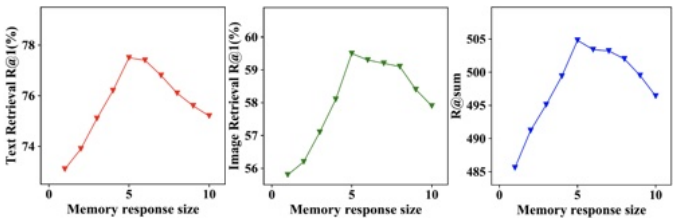

Fig. 3. The text retrieval R@1, image retrieval R@1, and R@sum variance of our method when using different memory response sizes.

2) Alternative Choices: As shown in Table IV, the "max" pooling strategy outperforms the other two, because the "max" pooling strategy can capture the most striking characteristics of each dimension. For fusion strategy, the late-fusion can handle fine-grained alignment for texts of different lengths separately, resulting in better performance than early-fusion. The difference between two retrieval strategies is minor and "similarity combination" strategy is slightly better than "embedding combination" strategy.

Following Wu *et al.* [15], we adopt pre-trained BERT to extract the token features. To further verify the effectiveness of our method, we also try to randomly initialize token embeddings and update them along with training. Comparing the last row of Table IV with other baselines in Table I, our method still outperforms all of them.

3) Effect of Memory Response Size: As shown in Figure 3, the best memory response size for text retrieval R@1, image retrieval R@1, and R@sum is 5. Besides, as the memory response size increases, the model performance on all three metrics first increases and then decreases. This might be because that when the memory response size is large, the noise in the memory responses degrades the quality of the cross-embeddings. We also find that the text retrieval performance drops faster than image retrieval as the memory response size increases. We conjecture that the image retrieval ability is weaker than the text retrieval ability (see Table I), therefore, extracting cross-modal information from the image memory bank may introduce more noise to the text cross-embedding.

TABLE V

THE EFFECT OF DIFFERENT MEMORY BANK ARRANGEMENTS. IMAGE CE (*Resp.*, TEXT CE) MEANS THE TYPE OF MEMORY BANKS USED TO ENHANCE THE IMAGE(RESP., TEXT) CROSS-EMBEDDING. TEXT BANK, IMAGE BANK, BOTH, AND / REPRESENT TEXT MEMORY BANK, IMAGE MEMORY BANK, BOTH OF THESE TWO MEMORY BANKS, AND NONE OF THESE MEMORY BANKS

Image CE	Text CE	R@1(T)	R@1(I)	R@sum
/	/	71.2	53.8	475.4
Text Bank	Image Bank	77.5	59.5	504.8
Image Bank	Text Bank	75.9	58.1	497.8
Image Bank	Image Bank	72.7	56.5	484.8
Text Bank	Text Bank	73.0	55.3	488.7
Both	Text Bank	74.5	57.6	493.3
Both	Image Bank	74.9	57.5	494.1
Text Bank	Both	74.9	57.9	493.9
Image Bank	Both	74.3	57.6	491.7
Both	Both	75.5	58.0	495.9

4) Effect of Memory Banks Arrangement: To further study the effect of our text and image memory banks, we allow the image (*resp.*, text) self-embedding to search on image memory bank, text memory bank, and both of these two memory banks. The memory response size for each memory bank is set as 5.

As shown in Table V, we can find that using text (*resp.*, image) memory bank to enhance the image (*resp.*, text) cross-embedding is the best strategy for our model. Whereas, if we get the memory response values from the memory bank of the same modality, the performance of our model in combine-embedding space is better than the performance in self-embedding (See Tables I and IV.), but the improvement is limited. Besides, when we compare the first, the second and the last row in Table V, we can find that the cross-embedding enhanced by both memory banks underperforms the cross-embedding enhanced by a single same/cross modality memory bank. We suspect that modality gap between memory response values from both memory banks would degrade the quality of cross-embedding, because cross-encoder needs to fuse both text and image information into the text (*resp.*, image) self-features for text (*resp.*, image) cross-features.

5) Effect of Fusion Strategy Arrangement: In Table IV, we perform experiments on early- and late-fusion strategies, but we keep the strategies for generating image and text cross-embedding the same. Here, we study the cases of using different fusion strategies to generate cross-embeddings.

As shown in Table VI, if we generate image cross-embedding and text cross-embedding with different fusion strategies, the performance of our model drops sharply. We conjecture that the shared cross-transformer encoder leads to the bad compatibility of different fusion strategies in a model.

TABLE VI

THE EFFECT OF DIFFERENT FUSION STRATEGY ARRANGEMENTS. IMAGE CE (*Resp.,* TEXT CE) MEANS THE FUSION STRATEGY USED TO GENERATE IMAGE (*Resp.*, TEXT) CROSS-EMBEDDING. EARLY AND LATE REPRESENT EARLY-FUSION AND LATE-FUSION STRATEGIES RESPECTIVELY

Image CE	Text CE	R@1(T)	R@1(I)	R@sum
Late	Late	77.5	59.5	504.8
Early	Early	75.2	58.1	496.3
Late	Early	71.9	55.9	476.8
Early	Late	72.8	56.3	484.7

TABLE VII

THE EFFECT OF DIFFERENT HYPER-PARAMETERS. d, h AND β REPRESENT THE HIDDEN DIMENSION, THE NUMBER OF HEADS AND THE MARGIN RESPECTIVELY

d	h	β	R@1(T)	R@1(I)	R@sum
512	4	0.050	77.5	59.5	504.8
256	4	0.050	75.9	58.1	498.5
1024	4	0.050	76.2	58.3	499.1
512	2	0.050	76.4	58.3	499.4
512	8	0.050	76.9	58.9	502.3
512	16	0.050	76.7	58.4	499.8
512	4	0.025	75.7	57.8	495.9
512	4	0.100	76.9	58.7	502.3
512	4	0.200	76.6	58.9	501.7

6) Effect of Hyper-Parameters: To verify the robustness of our model, we report the results using different combinations of hidden dimension d, number of heads h, and margin β.

From Table VII, we can find that the performance of our model remains stable as the hyper-parameters change in a reasonable range. For margin β, if it is too small, the distance between positive and negative image-text pairs would be too small to be distinguished, whereas, if it is large, the generalization ability of our model will also be affected negatively. For hidden size d, if it is too small, the model might lose the information in self-embedding and cross-embedding space when projecting token features (*resp.*, region features) from token feature size d_t (*resp.*, region feature size d_i) to the hidden size d. Otherwise, the increasing amount of parameters in our model can also lead to over-fitting and poor generalization from training set to test set.

D. Analysis on Memory Bank Size

For the experiments in Table I and II, the image (*resp.*, text) memory bank consists of all the training images (*resp.*, texts). To analyze the effect of memory searching space, we randomly select a subset from all training images (*resp.*, texts) as the new image (*resp.,* text) memory bank. Besides, the training and retrieving processes share the same usage ratio, which is the ratio of the selected images (*resp.,* text) to the all training images (*resp.,* texts). Then we conduct experiments on both Flickr30K and Microsoft COCO datasets, and report the text retrieval R@1, image retrieval R@1, and R@sum of MEMBER(Comb) in Table VIII.

TABLE VIII

THE EFFECT OF MEMORY BANK USAGE RATIO ON FLICKR30 AND MICROSOFT COCO (1K). USAGE RATIO, BANK SIZE(T), AND BANK SIZE(I) REPRESENT TRAINING INSTANCE USAGE RATIO, TEXT MEMORY BANK SIZE, AND IMAGE MEMORY BANK SIZE

Usage Ratio	Bank Size(T)	Bank Size(I)	R@1(T)	R@1(I)	R@sum
Flickr30K					
0%	-	-	72.1	57.1	489.4
20%	29,000	5,800	72.9	57.6	490.5
40%	58,000	11,600	74.4	58.1	493.4
60%	87,000	17,400	76.8	58.8	498.1
80%	116,000	23,200	77.2	59.0	501.6
100%	145,000	29,000	77.5	59.5	504.8
Microsoft COCO (1K)					
0%	-	-	75.2	60.7	513.8
20%	113,287	22,657	78.1	63.3	522.5
40%	226,574	45,315	78.4	63.5	523.1
60%	339,861	67,972	78.3	63.1	522.5
80%	453,148	90,630	78.4	63.6	523.7
100%	566,435	113,287	78.5	63.7	523.8

For Flickr30K dataset, we can find that using about 60% training images and texts for memory search can achieve comparable results, but more training images and texts continue to bring improvements. What's more, for Microsoft COCO, using only about 20% of training instances can achieve the comparable performance with the state-of-the-art performance. Comparing the size of text (*resp.* image) memory bank between Flickr30K and Microsoft COCO, we can conclude that about 120,000 texts and 24,000 images are enough to build the memory bank for our proposed MEMBER, which saves lots of time when the training set is large.

E. Time Complexity Analysis

Given N images and M texts, the time complexity of retrieval for embedding learning methods is $O(k_1NM + k_2(M + N))$, and that for pair-wise learning methods is $O(k_3NM + k_4(M + N))$, where the k_1, k_2, k_3, and k_4 are time complexity of inner product of two vectors, embedding generation in embedding learning methods, calculating pair-wise similarity with fine-grained alignment, and feature preparation in pair-wise learning methods. Usually, these four time complexities satisfy $k_2 \approx k_3 \approx k_4 \gg k_1$, so embedding learning methods are much faster than pair-wise learning methods.

The process of generating self/cross-embedding is relatively complex due to fine-grained alignment and fusion, but they are only performed based on top-n_m memory responses for each image and text. So our model is still more efficient than pair-wise learning methods.

To verify that our MEMBER strikes a good balance between efficiency and effectiveness, we compare the retrieval time with two embedding-learning methods (*i.e.*, VSRN [10] and SAEM [15]) and three pair-wise learning methods (*i.e.*, SCAN [3], PFAN [19], and IMRAM-full [4]) based on their released codes and hyper-parameters on Flickr30k test set, which contains 1,000 images and 5,000 texts to perform bi-directional retrieval. For fair comparison, all methods are run

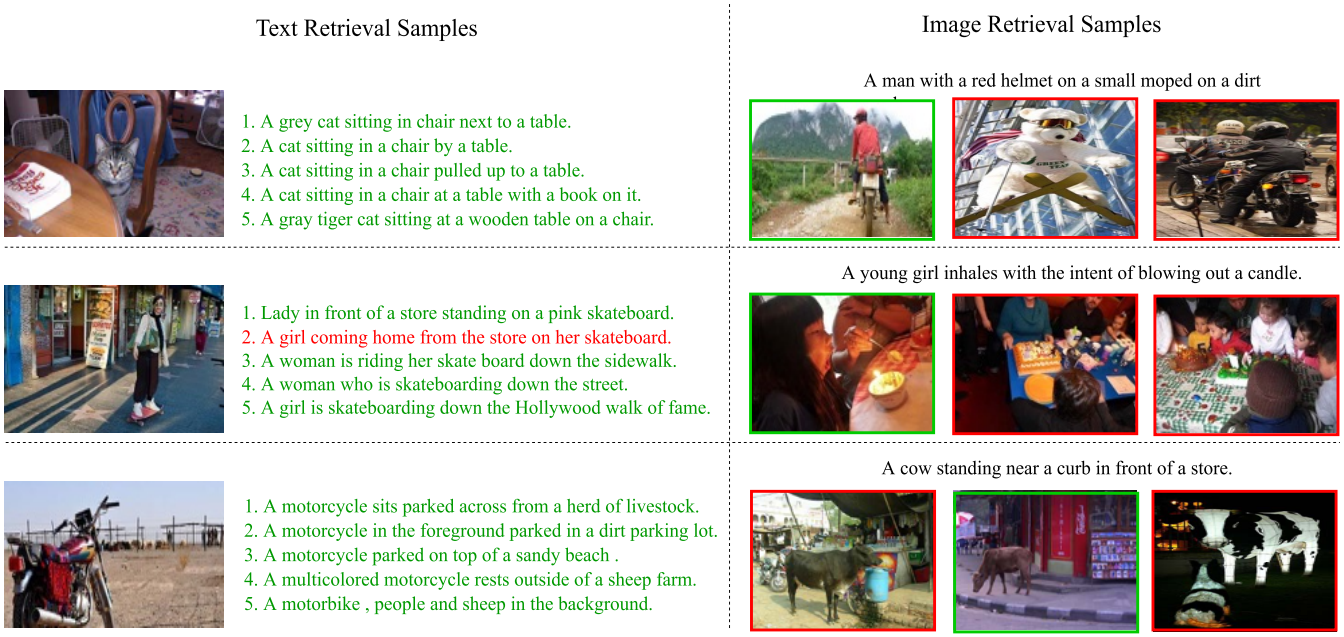

Fig. 4. Examples of image retrieval and text retrieval results. We show the top 5 retrieved texts for each image query and top 3 retrieved images for each text query. The correct text (*resp.*, image) retrieval results are in green color (*resp.*, boxes) and the incorrect ones are in red color (*resp.*, boxes).

TABLE IX

RETRIEVAL TIME COMPARISON WITH EXISTING MODELS ON FLICKR30K TEST SET. FOR EMBEDDING LEARNING METHODS, THEY ONLY CONTAIN k_1 AND k_2 PART; FOR PAIR-WISE LEARNING METHODS, THEY ONLY CONTAIN k_3 AND k_4 PART. MEMBER MEANS OUR MEMBER IN COMBINED SPACE WITH FULL MEMORY BANK AND MEMBER(60%) MEANS THAT WE ONLY USE 60% TRAINING IMAGES AND TEXTS TO BUILD THE MEMORY BANKS (SEE IV-D)

Learning Paradigm	Model	k_1(s)	k_2(s)	k_3(s)	k_4(s)	Total(s)
Pair-wise Learning	SCAN [3]	-	-	541	90	631
	PFAN [19]	-	-	549	101	650
	IMRAM-full [4]	-	-	1561	90	1651
Embedding Learning	VSRN [10]	1	91	-	-	92
	SAEM [15]	1	74	-	-	75
	MEMBER(60%)	1	107	-	-	108
	MEMBER	1	110	-	-	111

on a computer with RYZEN 3700x CPU@3.60GHz, 32GB memory and one GPU of GTX 1080TI with 11GB memory.

From Table IX, we find that the k_2 part for embedding learning methods and the k_4 part for pair-wise learning methods are quite similar. Even though our model has a relatively complex feature encoder, we only spend about 15% more time in k_2 part than most of existing methods. Moreover, the k_1 part for embedding learning methods is much faster than the k_3 part for pair-wise learning methods, which also verifies our theoretical analysis above. Therefore, our model is much more efficient than the pair-wise learning methods. Besides, the reduction in the memory bank size cannot save much time, which reveals that the memory search process does not cost much time. Totally, our retrieval time is slightly longer than SAEM [15] and VSRN [10], but our performance is significantly better than SAEM [15] and VSRN [10] (see Tables I and II).

TABLE X

THE EFFECT OF MEMORY BANK SIZE ON THE INFERENCE TIME OF FLICKR30. BANK SIZE(T), BANK SIZE(I), SELF, SEARCH, CROSS REPRESENT THE TEXT MEMORY BANK SIZE, THE IMAGE MEMORY BANK SIZE, THE INFERENCE TIME OF SELF TRANSFORMER ENCODER, THE INFERENCE TIME OF MEMORY SEARCH, AND THE INFERENCE TIME OF CROSS TRANSFORMER ENCODER

Bank Size(T)	Bank Size(I)	Self	Search	Cross	Total
29,000	5,800	27.4s	5.4s	72.8s	105.6s
58,000	11,600	27.9s	6.7s	71.9s	106.5s
87,000	17,400	28.2s	7.8s	72.5s	108.5s
116,000	23,200	27.1s	8.9s	73.8s	109.8s
145,000	29,000	26.8s	10.8s	72.7s	110.3s

Further, to provide more detailed analysis towards the inference speed of our model, in Table X, we provide the inference time of all three parts (*i.e.* self-transformer encoder, memory search and cross transformer encoder) of our model with different memory bank sizes. We can find that as the memory bank size changes, the time spent by self-transformer encoder and cross transformer encoder is relatively stable, and the time spent by memory search changes from 5.4s to 10.8s as the memory banks size changes from 29,000 texts and 5,800 images to 145,000 texts and 29,000 images. Note that, the memory search is finished on CUDA device, therefore, the time variation is not proportional to $O(nm)$, where n is the size of the memory bank and m is the size of inference dataset. Besides, the cross encoding between the image or text self-features and all the memory response values can also be done on CUDA device in parallel, which make inference time ratio between cross transformer encoder and self-transformer encoder less than the memory response size, *i.e.* 5. In fact, if we set the memory bank size around 120,000

Text Retrieval Samples

1. Horse stand and drink from pond water near the road .
2. Horse near a body of water with a sky background .
3. Horse behind a fence near a body of water .
4. Horse grazing in a muddy portion of a flood field .
5. Five horse next to a body of water behind a fence .

MEMBER

1. Five horse next to a body of water behind a fence .
2. Two horse walk through the wood together .
3. Horse grazing in a muddy portion of a flood field .
4. A line of horse be lead by one white horse .
5. Horse stand and drink from pond water near the road .

VSRN

1. Horse grazing in a muddy portion of a flood field .
2. Horse behind a fence near a body of water .
3. Five horse next to a body of water behind a fence .
4. Horse be race next to motorcycle on a dirt track .
5. Horse near a body of water with a sky background .

IMRAM

1. A lot of people walk down a cover walkway .
2. People walk on tile in an arch hallway .
3. Many people be walk along through the hallway .
4. Several people milling about in a lobby area .
5. A group of people walk down a street near a river .

MEMBER

1. Many people be walk along through the hallway .
2. People walk on tile in an arch hallway .
3. Several people milling about in a lobby area .
4. People be walk along the sidewalk next to a river .
5. A lot of people walk down a cover walkway .

VSRN

1. Several people milling about in a lobby area .
2. Horse behind a fence near a body of water .
3. Many people be walk along through the hallway .
4. People walk on tile in an arch hallway .
5. Horse near a body of water with a sky background .

IMRAM

1. Two glaze donut sit in a white bag .
2. Sugar donut sit in a white paper bag .
3. Two glaze donut sit in a paper bag .
4. Two sugary donut be in the bottom of a bag .
5. Two glaze doughnut in a see through pastry bag .

MEMBER

1. Two glaze donut sit in a paper bag .
2. Some donut be on a round white plate .
3. Two sugary donut be in the bottom of a bag .
4. Two glaze doughnut in a see through pastry bag .
5. Two glaze donut sit in a white bag .

VSRN

1. Two glaze donut sit in a white bag .
2. Two glaze doughnut in a see through pastry bag .
3. Sugar donut sit in a white paper bag .
4. Two sugary donut be in the bottom of a bag .
5. Two glaze donut sit in a paper bag .

IMRAM

Image Retrieval Samples

A trainer lead a girl on horseback to a field .

MEMBER

VSRN

IMRAM

A pair of elephant line up next each other in an enclosure .

MEMBER

VSRN

IMRAM

A half eat burger with a plate of fry next to mason jar .

MEMBER

VSRN

IMRAM

Fig. 5. Examples of image retrieval and text retrieval results. Follow a certain selection rule, we select three images and three texts to compare with VSRN [10] and IMRAM-full [4]. We show the top 5 retrieved texts for each image query and top 3 retrieved images for each text query. The correct text (*resp.*, image) retrieval results are in green color (*resp.*, boxes) and the incorrect ones are in red color (*resp.*, boxes).

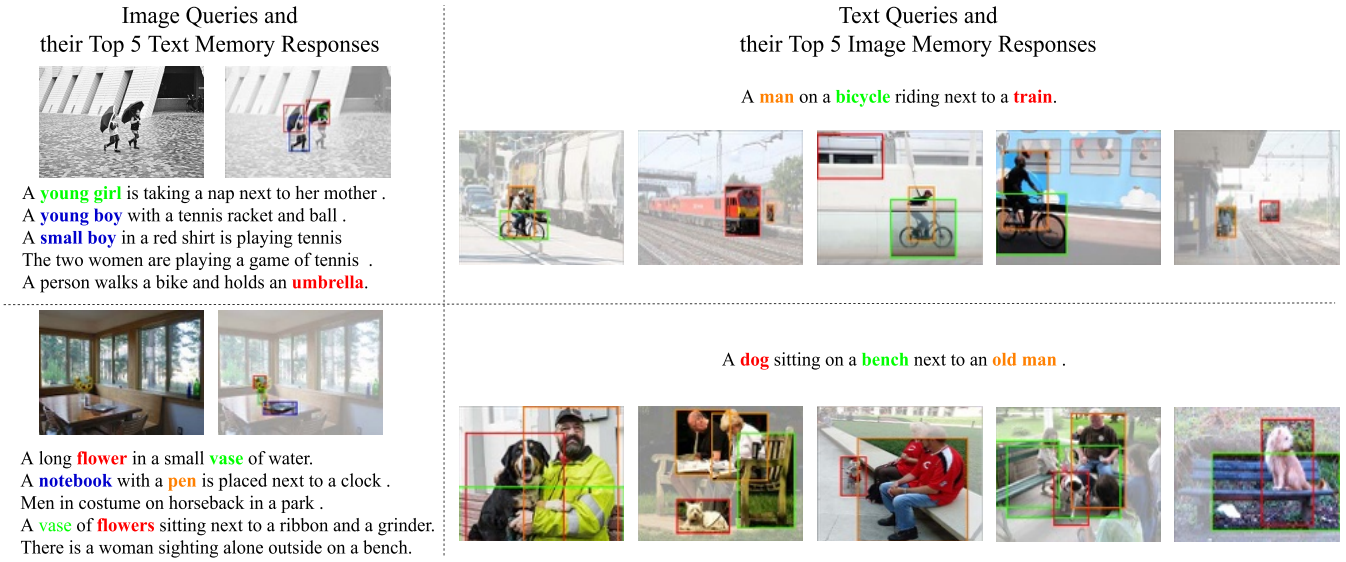

Fig. 6. The returned images (*resp.*, texts) from the memory bank with the input text (*resp.*, image). The local correspondences between images and texts are highlighted with different colors.

texts and 24,000 images, the memory search spends the shortest time in these three parts. Based on these experiment results and aforementioned theoretical analysis, our model has the speed advantage of embedding learning paradigm and is suitable for online test as other models in embedding learning paradigm.

F. Case Study

In Figure 4, we provide some text retrieval and image retrieval results. For each image (*resp.*, text) query, we show top-5 (*resp.*, top-3) ranked texts (*resp.,* images), where mismatched texts (*resp.,* images) are marked as red (*resp.,* enclosed by red boxes). These results show that our MEMBER method can retrieve the matching images or texts with a relatively high rank. Besides, our method can also capture local correspondences. For example, in the first image retrieval samples, the "red helmet" and "dirt road" can be well matched. However, for some confusing cases, our model still cannot distinguish some subtle changes in background, like the "curb" in the third image retrieval samples.

In Figure 5, we visualize another three text retrieval samples and three image retrieval samples, and compare with the state-of-the-art models in embedding learning paradigm, *i.e.* VSRN [10] and pair-wise learning paradigm, *i.e.* IMRAM [4]. To follow a certain principle and avoid deliberate sampling, we first sort all images along with their corresponding texts in validation set of Microsoft COCO by increasing the COCO id to get the sorted id for images and texts. Since each image corresponds to five texts, the sorted id of images ranges from [1,1000], and the sorted id of texts ranges from [1, 5000]. For text retrieval samples, we select the image every one hundreds images based on the sorted indices from 100. For image retrieval samples, we select the text every five hundreds texts based on the sorted indices from 501 to prevent overlap between selected images and texts.

From these samples, we can find that, under relatively simple situation (*i.e.* the donut sample and the elephant sample), all these three methods are capable of retrieving right images or texts. However, when the situation gets complex, the VSRN [10] tends to ignore some key information, like the "A trainer lead" in the first image retrieval sample and the "half eat burger" in the third image retrieval sample. On the contrary, our MEMBER can not only handle both simple and complex situation, but also can retrieve the matching images or texts with a higher rank than IMRAM [4].

G. Visualization and Qualitative Analysis

As mentioned in Sec. III-C.2, when using a text (*resp.*, image) to search the image (*resp.*, text) memory bank, the returned image (*resp.*, text) features from the memory bank do not strictly match this query text (*resp.*, image) because we have filtered out its matched images (*resp.*, texts) in the memory bank. So our goal of memory-based enhancement is to extract useful cross-modal information from loosely matched pairs. To validate this point, we visualize the input image (*resp.*, text) and the returned texts (*resp.*, images) in Figure 6. We utilize the attention weight from the cross-learning stage to generate the cross-modal local correspondence in Figure 6. By taking image-to-text retrieval as an example, given an image self-feature and a memory response value, we first use multi-head attention mechanism to calculate h attention weights $\mathbf{A}_j \ \forall j \in [1, h]$ (see Equ. (1)). Then we average these attention weights to get the final attention weight $\bar{\mathbf{A}}$. Finally, We select the local correspondence according to the entries larger than 0.9 in $\bar{\mathbf{A}}$. Color correspondence in Figure 6 indicates the local correspondence between regions in images and words in texts. From these examples, we find that most of the returned images (*resp.*, texts) have local correspondence with the input text (*resp.*, image). In the example "A man on a bicycle riding next to a train", returned images have the corresponding objects, "bicycle", "man", and "train", where the related region and word are highlighted in the same color. Since each of these returned images (*resp.*, texts) can only provide part of the local correspondences, it is important

to aggregate top-5 returned images (*resp.*, texts) from the memory bank.

V. Conclusion

In this paper, we have studied image-text retrieval from a new viewpoint, *i.e.*, enhancing embedding via fine-grained alignment and fusion. We have proposed a novel method for memory-based mutual embedding enhancement, with the retrieval performed in both self-embedding space and cross-embedding space. Besides, our method maintains a relatively fast speed. Comprehensive experiments on two benchmark datasets have demonstrated that our method remarkably outperforms the state-of-the-art approaches.

Acknowledgment

The authors thank Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.

References

[1] A. Karpathy, A. Joulin, and L. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Proc. 28th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 1889–1897.

[2] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: Improving visual-semantic embeddings with hard negatives," in *Proc. 29th Brit. Mach. Vis. Conf.*, Surrey, U.K., Sep. 2018, p. 12.

[3] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. 15th Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 212–228.

[4] H. Chen, G. Ding, X. Liu, Z. Lin, J. Liu, and J. Han, "IMRAM: Iterative matching with recurrent attention memory for cross-modal image-text retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12652–12660.

[5] X. Chen and C. L. Zitnick, "Mind's eye: A recurrent visual representation for image caption generation," in *Proc. CVPR*, Jun. 2015, pp. 2422–2431.

[6] S. Antol *et al.*, "VQA: Visual question answering," in *Proc. IEEE ICCV*, Jun. 2015, pp. 2425–2433.

[7] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6720–6731.

[8] N. Rasiwasia *et al.*, "A new approach to cross-modal multimedia retrieval," in *Proc. 18th ACM Int. Conf. Multimedia*, Oct. 2010, pp. 251–260.

[9] H. Nam, J.-W. Ha, and J. Kim, "Dual attention networks for multimodal reasoning and matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2156–2164.

[10] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Nov. 2019, pp. 4653–4661.

[11] T. Chen and J. Luo, "Expressing objects just like words: Recurrent visual embedding for image-text matching," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Apr. 2020, pp. 10583–10590.

[12] C. L. Grady, A. R. McIntosh, M. N. Rajah, and F. I. Craik, "Neural correlates of the episodic encoding of pictures and words," *Proc. Nat. Acad. Sci. USA*, vol. 95, no. 5, pp. 2703–2708, 1998.

[13] G. Song, D. Wang, and X. Tan, "Deep memory network for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1261–1275, May 2019.

[14] Z. Ji, Z. Lin, H. Wang, and Y. He, "Multi-modal memory enhancement attention network for image-text matching," *IEEE Access*, vol. 8, pp. 38438–38447, 2020.

[15] Y. Wu, S. Wang, G. Song, and Q. Huang, "Learning fragment self-attention embeddings for image-text matching," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2088–2096.

[16] X. He, Y. Peng, and L. Xie, "A new benchmark and approach for fine-grained cross-media retrieval," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1740–1748.

[17] S. Wang, R. Wang, Z. Yao, S. Shan, and X. Chen, "Cross-modal scene graph matching for relationship-aware image-text retrieval," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1497–1506.

[18] S. Wang, Y. Chen, J. Zhuo, Q. Huang, and Q. Tian, "Joint global and co-attentive representation learning for image-sentence retrieval," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, Seoul South Korea, Oct. 2018, pp. 1398–1406.

[19] Y. Wang *et al.*, "Position focused attention network for image-text matching," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3792–3798.

[20] R. He, M. Zhang, L. Wang, Y. Ji, and Q. Yin, "Cross-modal subspace learning via pairwise constraints," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5543–5556, Dec. 2015.

[21] D. Wang, X. Gao, X. Wang, L. He, and B. Yuan, "Multimodal discriminative binary embedding for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4540–4554, Oct. 2016.

[22] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2494–2507, May 2017.

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," in *Proc. 28th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 827–838.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–14.

[25] Y. Huang, W. Wang, and L. Wang, "Instance-aware image and sentence matching with selective multimodal LSTM," in *Proc. CVPR*, Jul. 2017, pp. 7254–7262.

[26] Y. Wu, S. Wang, and Q. Huang, "Online asymmetric similarity learning for cross-modal retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3984–3993.

[27] Y. Peng, Z. Ye, J. Qi, and Y. Zhuo, "Unsupervised visual-textual correlation learning with fine-grained semantic alignment," *IEEE Trans. Cybern.*, early access, Sep. 15, 2020, doi: 10.1109/TCYB.2020.3015084.

[28] X. He and Y. Peng, "Fine-grained visual-textual representation learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 520–531, Feb. 2020.

[29] C. Liu, Z. Mao, T. Zhang, H. Xie, B. Wang, and Y. Zhang, "Graph structured network for image-text matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10918–10927.

[30] Y. Peng, J. Qi, and Y. Yuan, "Modality-specific cross-modal similarity measurement with recurrent attention network," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5585–5599, Nov. 2018.

[31] F. Huang, X. Zhang, Z. Zhao, and Z. Li, "Bi-directional spatial-semantic attention networks for image-text matching," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2008–2020, Apr. 2019.

[32] Y. Peng, J. Qi, and Y. Zhuo, "MAVA: Multi-level adaptive visual-textual alignment by cross-media bi-attention mechanism," *IEEE Trans. Image Process.*, vol. 29, pp. 2728–2741, Nov. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8910611

[33] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2014, *arXiv:1410.3916*. [Online]. Available: http://arxiv.org/abs/1410.3916

[34] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 2440–2448.

[35] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," 2014, *arXiv:1410.5401*.

[36] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1400–1409.

[37] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–110.

[38] C. C. Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," in *Proc. CVPR*, Jul. 2017, pp. 6432–6440.

[39] M. Wang, Z. Lu, H. Li, and Q. Liu, "Memory-enhanced decoder for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 278–286.

[40] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. EMNLP*, Austin, TX, USA, Nov. 2016, pp. 551–561.

[41] Y. Zhu, J. J. Lim, and L. Fei-Fei, "Knowledge acquisition for visual question answering via iterative querying," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6146–6155.

[42] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. CVPR*, Jun. 2016, pp. 21–29.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2018, pp. 4171–4186.
[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. CVPR*, Sep. 2015, pp. 91–99.
[45] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6077–6086.
[46] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 32th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 5998–6008.
[47] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[48] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 539–546.
[49] Z. Wang *et al.*, "CAMP: Cross-modal adaptive message passing for text-image retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5763–5772.
[50] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, Sep. 2014, pp. 740–755.
[51] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 67–78, Feb. 2014.
[52] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. 31th Annu. Conf. Neural Inf. Process. Syst. (Autodiff Workshop)*, Montreal, QC, Canada, Dec. 2017, pp. 1–4.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 2nd Int. Conf. Learn. Represent.*, Apr. 2014, pp. 1–15.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[55] R. Krishna *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

Jiangtong Li received the B.E. degree from Shanghai Jiao Tong University, Shanghai, China, in 2019, where he is currently pursuing the Ph.D. degree with the Computer Science and Engineering Department. His current research interests cover cross-modal retrieval, visual-question answering, and natural language processing.

Liu Liu received the B.E. degree from the Harbin Institute of Technology in 2019. She is currently pursuing the master's degree with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai, China. Her current research interests cover cross-modal retrieval, image composition, and deep learning.

Li Niu received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2017. He is currently an Associate Professor with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai, China. Before joining Shanghai Jiao Tong University, he was a Postdoctoral Associate at Rice University, Houston, TX, USA. His current research interests include machine learning, deep learning, and computer vision.

Liqing Zhang (Member, IEEE) received the Ph.D. degree from Zhongshan University, Guangzhou, China, in 1988. He was promoted to a Full Professor position in 1995 at the South China University of Technology. He worked as a Research Scientist at the RIKEN Brain Science Institute, Japan, from 1997 to 2002. Since September 2002, he has been a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He has published more than 250 papers in journals and international conferences. His current research interests cover computational theory for cortical networks, visual cognitive representation and inference, and statistical learning.