Heterogeneous Attention Network for Effective and Efficient Cross-modal Retrieval

Tan Yu[1], Yi Yang[2], Yi Li[2], Lin Liu[2], Hongliang Fei[1], Ping Li[1]
1. Cognitive Computing Lab, Baidu Research
2. Baidu Search Ads (Phoenix Nest), Baidu Inc.
10900 NE 8th St. Bellevue, Washington 98004, USA
No. 10 Xibeiwang East Road, Beijing 10193, China
{tanyu01, yangyi15, liyi01, liulin03, hongliangfei, liping11}@baidu.com

ABSTRACT

Traditionally, the task of cross-modal retrieval is tackled through joint embedding. However, the global matching used in joint embedding methods often fails to effectively describe matchings between local regions of the image and words in the text. Hence they may not be effective in capturing the relevance between the text and the image. In this work, we propose a heterogeneous attention network (HAN) for effective and efficient cross-modal retrieval. The proposed HAN represents an image by a set of bounding box features and a sentence by a set of word features. The relevance between the image and the sentence is determined by the set-to-set matching between the set of word features and the set of bounding box features. To enhance the matching effectiveness, we exploit the proposed heterogeneous attention layer to provide the cross-modal context for word features as well as bounding box features. Meanwhile, to optimize the metric more effectively, we propose a new soft-max triplet loss, which adaptively gives more attention to harder negatives and thus trains the proposed HAN in a more effective manner compared with the original triplet loss. Meanwhile, the proposed HAN is efficient, and its lightweight architecture only needs a single GPU card for training. Extensive experiments conducted on two public benchmarks demonstrate the effectiveness and efficiency of our HAN. This work has been deployed in production Baidu Search Ads and is part of the "PaddleBox" platform.

CCS CONCEPTS

• **Information systems** → **Similarity measures**; **Image search**.

KEYWORDS

Cross-modal retrieval; Image retrieval

ACM Reference Format:
Tan Yu[1], Yi Yang[2], Yi Li[2], Lin Liu[2], Hongliang Fei[1], Ping Li[1]. 2021. Heterogeneous Attention Network for Effective and Efficient Cross-modal Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404835.3462924

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '21, July 11–15, 2021, Virtual Event, Canada
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462924

1 INTRODUCTION

Compared with texts, images and videos can capture users' attention more easily. They have grown to be the largest traffic on the internet. The emergence of image and video content brings the increasing demands for cross-modal understanding. One of the most widely used cross-modal understanding tasks is text-to-image retrieval, which builds a link between the text and the visual content. It considerably boosts search flexibility and provides a more convenient way for richer information retrieval.

Due to the modal gap, text-to-image retrieval is more challenging than traditional single-modal retrieval, such as text-to-text or image-to-image retrieval. Traditionally, text-to-image retrieval is tackled by joint embedding. It first extracts text features and image features individually and then maps text features and video features into the same feature space. After joint embedding, texts and images can be directly compared in the joint feature space. Early works of joint embedding mainly rely on linear projection [12, 21, 31]. Basically, they seek to learn a linear projection to maximize the pairwise correlation. Due to the limited capacity of the linear projection, some complex patterns might not be effectively captured through a linear operation. Thus, several methods exploit non-linear kernels besides linear projection to further enhance the discriminating power of features. Recently, inspired by great success achieved by deep learning in search [2, 17, 39, 50], some deep joint embedding methods [7, 16, 48] were proposed. They normally adopt a two-branch architecture. To be specific, one branch utilizes a convolutional neural network (CNN) [13, 19] to learn the image representation. The other branch is built on the sequence model such as recurrent neural network (RNN) [34] and its variants, LSTM [11] and GRU [5], to generate the text feature. The two-branch network is trained in an end-to-end manner through a pairwise and triplet loss [33]. Thanks to the strong capability of deep neural networks, the learned representation achieves better performance than the linear projection. Recently, ACMR [41] exploited adversarial learning to bridge the distribution gap between text features and image features. It trains a generator to confuse the discriminator so that the distribution of image features is closed to that of text features.

Although the joint embedding methods often achieve good performance in cross-modal retrieval, there is still room for improvement. One drawback of the joint embedding methods is that they represent an image as well as a sentence as a global feature, and the relevance between the image and the sentence is determined by the similarity between their global features. In many cases, the relevance between a sentence and an image depends on local matchings between several local regions in the image and a few words in

the sentence. Thus, the global matching between the image-level feature and the sentence-level feature might not capture their relevance effectively. To achieve a more effective matching, SCAN [20] represented an image by a set of detected bounding boxes, which are candidate local regions of interesting objects. Meanwhile, SCAN represents a sentence by a set of word features. After that, SCAN formulates the sentence-to-image retrieval task as matching between a set of word features and a set of bounding box features. In this case, relying on local matchings, SCAN can effectively capture the relevance between a sentence and an image, achieving a better performance than the methods based on global features.

Another drawback of joint embedding is that the text only interacts with the image in the matching phase, which might fail to provide the cross-modal context in generating image features and text features. Recently, inspired by the success of Transformer [40] and BERT [6] in natural language processing (NLP), some BERT-based retrieval models [9, 14, 22, 23, 26, 37, 47, 49] were proposed. They can achieve cross-modal attention by fusing the text features and bounding box features in the early stage. In this case, the bounding box features and word features are mutually attended. By exploiting attention, they have achieved state-of-the-art performance in retrieval. Nevertheless, existing BERT-based retrieval methods such as Unicoder-VL [22] and ViLBERT [26] stacked a large number of self-attention layers, which are computationally expensive for both training and inference. The inefficiency of cross-modal BERT limits their usefulness in large-scale cross-modal retrieval applications. Meanwhile, due to the massive number of parameters, it needs to be pre-trained on large-scale datasets to suppress the over-fitting. Moreover, when conducting the cross-modal retrieval, both Unicoder-VL and ViLBERT are based on the global feature matching, which is not effective in capturing the matching at the local level, as we mentioned above.

Observing the limitations of existing cross-modal BERT methods, we propose a Heterogeneous Attention Network (HAN) to achieve an effective and efficient cross-modal retrieval. Like SCAN, HAN also represents an image by a set of bounding box features and represents a text sentence by a set of word features. However, we inject the cross-modal context in generating word features and bounding box features by heterogeneous attention before cross-modal matching. Besides, compared with Unicoder-VL and ViLBERT based on matching global features, the proposed HAN is based on matching local features, which is more effective in capturing the relevance between an image and a sentence. Meanwhile, our HAN is much more efficient than Unicoder-VL and ViLBERT in both training and search phases. To be specific, our HAN only takes 3 layers of transformers, which only needs a single NVIDIA V100 GPU card for training. The efficiency advantage in training achieved by the proposed HAN is considered essential for industrial applications where the data is frequently updated. Meanwhile, the faster retrieval speed can bring the user a better experience. Moreover, we propose a soft-hard negative mining strategy to train the proposed HAN more effectively. Our experiments show the proposed soft-hard negative mining consistently outperforms the existing hard negative mining methods used in SCAN [20] and Unicoder-VL [22]. Systematic experiments conducted on two public benchmark datasets demonstrate the effectiveness of the proposed HAN.

The **contributions** of this work are two-fold:

- An efficient and effective model, HAN, is proposed. Compared with existing state-of-the-art methods relying on expensive Transformer, our method achieves a comparable accuracy but is much more efficient in cross-modal retrieval.
- We propose a novel soft-max hard negative mining loss for training the proposed HAN. It adaptively gives higher weights to the harder negative samples, achieving better performance than using the existing triplet loss.

2 RELATED WORK

This paper reports our recent progress in developing heterogeneous attention networks for cross-modal retrieval of Baidu's advertising systems [8, 44]. In particular, the work has been deployed for production, based on the "PaddleBox" training platform [51, 52].

This section reviews some closely related existing works, which are coarsely divided into two categories: 1) global-feature methods, 2) local-feature methods. Below we review methods in these two categories, respectively.

2.1 Global-feature methods

Early cross-modal retrieval methods [12, 31] relied on canonical correlation analysis (CCA) to map text and image features to a joint space. Inspired by the success of deep learning, recent works [7, 16] exploited deep neural network to learn the image and text features in an end-to-end manner. They normally utilize a pairwise or triplet loss function to bridge the modal gap and achieve better performance than CCA-based methods. To further boost the training effectiveness, VSE++ [7] proposed to pay more attention to the hard negatives within each mini-batch, achieving better performance. In parallel, ACMR [41] exploited adversarial learning and trained the feature extractor to confuse the modal classifier. It aims to make the distribution of the image features closed to that of text features.

2.2 Local-feature methods

Observing the limitations of methods based on global features, researchers proposed several local-feature methods. DAN [29] used local features of a CNN to replace the original global image features. It attends to specific local features of an image and words in text and estimates the similarity between the image and the text by focusing on their shared semantics. SCAN [20] further replaced the local convolutional features used in DAN [29] by features of bounding boxes detected from faster-RCNN. The detected bounding boxes are candidate locations of interesting objects, which are more suitable for modeling the semantics of images. Meanwhile, SCAN proposes a cross attention module, which attends to items from one modal to the items from the other modal. Then SCAN formulates the cross-modal retrieval as a matching between the attended features of items from one modal with the original features from items to the other modal. By exploiting finer-level matching, SCAN achieves a considerably better performance than methods using global features.

Recently, inspired by the great success of BERT in natural language processing (NLP), some cross-modal BERT methods [4, 10, 22–24, 26–28, 37, 38, 46] were proposed. Like SCAN, they represent the image by a set of bounding box features and the text sentence

by a set of word features. The bounding box features and word features are mutually attended. Cross-modal BERT methods can be coarsely grouped into two categories. The first category including ViLBERT [26] and LXMERT [38] adopt a two-stream architecture. One stream is designed for the text modal, in which word features are attended by bounding box features. The other stream is designed for the image modal in which the bounding box features are attended by word features of the text. Meanwhile, they design several tasks such as masked language modeling, masked object feature regression, text-image matching, and visual question answering for pre-training the proposed model on a large-scale dataset. In parallel, the other category including VisualBERT [23], Unicoder-VL [22] and VL-BERT [37] adopt the one-stream architecture. They encode the word features and visual features through a single stack of transformer blocks. Similarly, they also design several tasks for pre-training on a large-scale dataset. Nevertheless, one-stream and two-stream vision BERT methods such as ViLBERT and Unicoder-VL stack a large number of transformer blocks, which are computationally expensive and impractical in some online applications. Meanwhile, in the matching phase, ViLBERT and Unicoder-VL use global features, which might be ineffective for capturing the relevance between the text and the image.

3 METHOD

As shown in Figure 1, the proposed heterogeneous attention network (HAN) consists of four modules: 1) text and image encoding module, 2) heterogeneous attention module, 3) matching module and 4) loss module . In particular, the image is represented by a set of bounding box features generated from the faster R-CNN [32].

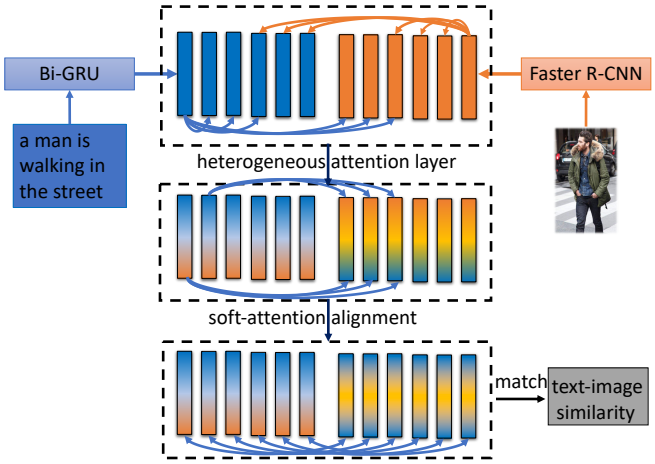

Figure 1: The architecture of the proposed heterogeneous attention network (HAN) model. An image is represented by a set of bounding box features and a sentence is represented by a sequence of word features. The heterogeneous attention layer processes the word features and bounding box features to provide the cross-modal context. The relevance between the image and the sentence is attained through set-to-set matching between the set of attended word features and the set of attended bounding box features.

The text is represented by a sequence of word features from bidirectional GRU (Bi-GRU). Then the word features and bounding box features are merged in a set and pass through the devised heterogeneous attention layer to obtain the attended word features and bounding box features. After that, we align the attended bounding box features with respect to attended word features through soft attention. Finally, the relevance between the image and the text is obtained by matching the aligned bounding box features and the attended word features. Below we introduce them in detail.

3.1 Image and Text Encoding Module

For each image, we extract m bounding boxes through an object detector. We denote the bounding boxes features by $\{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_m\}$. Meanwhile, for a sentence of n words, we represent word features from word embedding, $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n\}$. To further enhance the discriminating power of bounding box features, we use a fully-connected layer to process each bounding box feature $\mathbf{o}_i$:

$$\bar{\mathbf{o}}_i = \mathbf{W}\mathbf{o}_i + \mathbf{b},$$

where $\mathbf{W}$ and $\mathbf{b}$ are parameters of the fully-connected layer.

As for the text features, we use bi-directional GRU (biGRU) to encode the sequential information of words. To be specific, the forward GRU reads word features from $\mathbf{w}_1$ to $\mathbf{w}_n$ and attain the hidden state at each word:

$$[\overrightarrow{\mathbf{w}}_1, \cdots, \overrightarrow{\mathbf{w}}_n] = \overrightarrow{\mathrm{GRU}}([\mathbf{w}_1, \cdots, \mathbf{w}_n]).$$

Meanwhile, the backward GRU reads word features from $\mathbf{w}_n$ to $\mathbf{w}_1$ and generates the hidden state at each word:

$$[\overleftarrow{\mathbf{w}}_1, \cdots, \overleftarrow{\mathbf{w}}_n] = \overleftarrow{\mathrm{GRU}}([\mathbf{w}_1, \cdots, \mathbf{w}_n]).$$

Then the word features are obtained by

$$\bar{\mathbf{w}}_i = (\overrightarrow{\mathbf{w}}_i + \overleftarrow{\mathbf{w}}_i)/2, \ , i \in [1, n].$$

The bounding box features $\{\bar{\mathbf{o}}_1, \bar{\mathbf{o}}_2, \cdots, \bar{\mathbf{o}}_m\}$ and the word features $\{\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \cdots, \bar{\mathbf{w}}_m\}$ are attained based on the information from a single modal and do not exploit the cross-modal context. They are the input of the proposed heterogeneous attention module.

3.2 Heterogeneous attention module

The heterogeneous attention module treats word features and bounding box features equally. We do not differentiate word features from bounding box features through additional type embedding feature like existing cross-modal BERT methods [22]. To make them unbiased, we conduct the ℓ_2 normalization on both word features and bounding box features. To be specific, the input of heterogeneous attention module $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_{m+n}]$ are obtained by concatenating the normalized word features and bounding box features:

$$[\mathbf{h}_1, \cdots, \mathbf{h}_{m+n}] = [\frac{\bar{\mathbf{o}}_1}{\|\bar{\mathbf{o}}_1\|_2}, \cdots, \frac{\bar{\mathbf{o}}_m}{\|\bar{\mathbf{o}}_m\|_2}, \frac{\bar{\mathbf{w}}_1}{\|\bar{\mathbf{w}}_1\|_2}, \cdots, \frac{\bar{\mathbf{w}}_m}{\|\bar{\mathbf{w}}_n\|_2}].$$

The heterogeneous attention module is a stack of several self-attention blocks. To be specific, the input of the first self-attention block is $\mathbf{H}$ defined above, and the input of an upper self-attention block is the output of the lower self-attention block. Next, we introduce the details of the first self-attention block and the configurations of other blocks are similar. The transformer block takes $\mathbf{H} =$

$[\mathbf{h}_1, \cdots, \mathbf{h}_{m+n}]$ as input and generates the query items $\{\mathbf{q}_i\}_{i=1}^{m+n}$, value items $\{\mathbf{v}_i\}_{i=1}^{m+n}$ and key items $\{\mathbf{k}_i\}_{i=1}^{m+n}$ by

$$\mathbf{q}_i = \mathbf{W}_q\mathbf{h}_i + \mathbf{b}_q, \qquad \mathbf{k}_i = \mathbf{W}_k\mathbf{h}_i + \mathbf{b}_k, \qquad \mathbf{v}_i = \mathbf{W}_v\mathbf{h}_i + \mathbf{b}_v,$$

where $\{\mathbf{W}_q, \mathbf{b}_q\}$ are parameters of the fully-connected layer to generate query items, $\{\mathbf{W}_k, \mathbf{b}_k\}$ are parameters for key items and $\{\mathbf{W}_v, \mathbf{b}_v\}$ are parameters for value items. For each query item $\mathbf{q}_i$, its attention with respect to each key $\mathbf{k}_j$ is computed by

$$a_{i,j} = \mathbf{q}_i^\top \mathbf{k}_j, \;\; i \in [1, m+n], \;\; j \in [1, m+n]$$

and then is normalized by a soft-max operation by

$$\hat{a}_{i,j} = \frac{e^{\beta a_{i,j}}}{\sum_{k=1}^{n+m} e^{\beta a_{i,k}}}, \;\; i \in [1, m+n], \;\; j \in [1, m+n]$$

where β is a positive constant controlling the softness. We further define the attention vector of the i-th heterogeneous item as $\mathbf{a}_i = [\hat{a}_{i,1}, \hat{a}_{i,2}, \cdots, \hat{a}_{i,m+n}]$. By exploiting the attentions from other items, the attended i-th heterogeneous item is attained by

$$\hat{\mathbf{h}}_i = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n+m}]\mathbf{a}_i, \;\; i \in [1, m+n].$$

Using the residual structure, i-th heterogeneous item is refined by

$$\bar{\mathbf{h}}_i = \text{fc}(\hat{\mathbf{h}}_i) + \mathbf{q}_i, \;\; i \in [1, m+n],$$

where $\text{fc}(\cdot)$ denotes a fully-connected layer. After that, the i-th heterogeneous item is further processed by a layer normalization. The new heterogeneous items $[\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \cdots, \tilde{\mathbf{h}}_{n+m}]$ are used as the input of the next transformer block. We denote the output of the last transformer block by $[\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \cdots, \tilde{\mathbf{h}}_{n+m}]$.

3.3 Alignment and matching module

The final cross-modal attended word features correspond to the first n heterogeneous items from heterogeneous attention module:

$$\hat{\mathbf{w}}_i = \tilde{\mathbf{h}}_i, \;\; i \in [1, n].$$

The final cross-modal attended bounding box features are the rest:

$$\hat{\mathbf{o}}_i = \tilde{\mathbf{h}}_{n+i}, \;\; i \in [1, m].$$

The similarity between the image and the text is obtained by matching the set of word features $\{\hat{\mathbf{w}}_i\}_{i=1}^n$ and the set of bounding box features $\{\hat{\mathbf{o}}_i\}_{i=1}^m$. Following SCAN [20], we conduct the alignment and matching between text features and bounding box features.

Alignment. It first computes the cosine similarity between each possible pair of bounding box feature and word feature:

$$s_{i,j} = \frac{\hat{\mathbf{w}}_i^\top \hat{\mathbf{o}}_j}{\|\hat{\mathbf{w}}_i\|_2 \|\hat{\mathbf{o}}_j\|_2}, \;\; i \in [1, n], \;\; j \in [1, m].$$

Through soft-max operation, we compute the normalized similarity:

$$a_{i,j} = \frac{e^{\beta s_{i,j}}}{\sum_{k=1}^{m} e^{\beta s_{i,k}}}, \;\; i \in [1, n], \;\; j \in [1, m],$$

where β is a positive constant for controlling the softness of attentions. Then, for each word feature $\hat{\mathbf{w}}_i$, it generates an aligned word feature by looking through all bounding box features by

$$\tilde{\mathbf{o}}_i = \sum_{j=1}^{m} a_{i,j}\hat{\mathbf{o}}_j, \;\; i \in [1, n].$$

Intuitively, $\tilde{\mathbf{o}}_i$ pays more attention to the features of bounding boxes with high relevance score with respect to the word feature $\hat{\mathbf{w}}_i$.

Matching. The relevance between each word feature $\hat{\mathbf{w}}_i$ and each aligned word feature $\mathbf{a}_i$ is obtained by

$$r_i = \frac{\hat{\mathbf{w}}_i^\top \tilde{\mathbf{o}}_i}{\|\hat{\mathbf{w}}_i\|_2 \|\tilde{\mathbf{o}}_i\|_2}, \;\; i \in [1, n].$$

The relevance between the image and the text is obtained by computing the average value of $\{r_i\}_{i=1}^n$:

$$S_{\text{AVG}} = \frac{1}{n}\sum_{i=1}^{n} r_i. \tag{1}$$

Note that the above alignment and matching is conducted on the text side. That is, it aligns the bounding box features according to the text words features. This setting is due to the fact we are more interested in text-to-image retrieval, which has more practical application compared with image-to-text retrieval. If the image-to-text retrieval is of more importance in some cases, the alignment

Algorithm 1 The pipeline of the proposed heterogeneous attention network (HAN).

Input: a sentence $S = [w_1, \cdots, w_n]$ and an image I. The number of heterogeneous attention layers L.

Output: $s(S, I)$, the relevance score of the text-image pair (S, I).

$[\mathbf{w}_1, \cdots, \mathbf{w}_n] \leftarrow \text{word2vec}([w_1, \cdots, w_n])$.
$[\bar{\mathbf{w}}_1, \cdots, \bar{\mathbf{w}}_n] \leftarrow \text{BiGRU}([\mathbf{w}_1, \cdots, \mathbf{w}_n])$.
$[\mathbf{o}_1, \cdots, \mathbf{o}_m] \leftarrow \text{Faster\_RCNN}(I)$.
$\mathbf{h}_i \leftarrow \bar{\mathbf{w}}_i / \|\bar{\mathbf{w}}_i\|_2, \; i \in [1, n]$
$\bar{\mathbf{o}}_i \leftarrow \mathbf{W}\mathbf{o}_i + \mathbf{b}, \; i \in [1, m]$
$\mathbf{h}_{i+n} \leftarrow \bar{\mathbf{o}}_i / \|\bar{\mathbf{o}}_i\|_2, \; i \in [1, m]$
$\bar{\mathbf{W}}^{(0)} \leftarrow \hat{\mathcal{W}}^{(2)}, \; \bar{\mathbf{B}}^{(0)} \leftarrow \hat{\mathcal{B}}^{(2)}$
for $l \in [1, L]$ **do**
  for $i \in [1, m+n]$ **do**
    $\mathbf{q}_i \leftarrow \mathbf{W}_q\mathbf{h}_i + \mathbf{b}_q, \; \mathbf{k}_i \leftarrow \mathbf{W}_k\mathbf{h}_i + \mathbf{b}_k, \; \mathbf{v}_i \leftarrow \mathbf{W}_v\mathbf{h}_i + \mathbf{b}_v$
  end for
  for $i \in [1, m+n]$ **do**
    $\mathbf{a}_i \leftarrow [\mathbf{k}_1, \cdots, \mathbf{k}_{m+n}]^\top \mathbf{q}_i$
    $\mathbf{a}_i \leftarrow \text{softmax}(\mathbf{a}_i)$
    $\mathbf{h}_i \leftarrow [\mathbf{v}_1, \cdots, \mathbf{v}_{n+m}]\mathbf{a}_i$
    $\mathbf{h}_i \leftarrow \text{fc}(\mathbf{h}_i) + \mathbf{q}_i$
    $\mathbf{h}_i \leftarrow \text{layer\_norm}(\mathbf{h}_i)$
  end for
end for
$\hat{\mathbf{w}}_i \leftarrow \mathbf{h}_i, \; i \in [1, n]$
$\hat{\mathbf{o}}_i \leftarrow \mathbf{h}_{n+i}, \; i \in [1, m]$
for $i \in [1, n]$ **do**
  $s_{i,j} \leftarrow \frac{\hat{\mathbf{w}}_i^\top \hat{\mathbf{o}}_j}{\|\hat{\mathbf{w}}_i\|_2 \|\hat{\mathbf{o}}_j\|_2}, \; j \in [1, m]$
  $\mathbf{s}_i \leftarrow [s_{i,1}, \cdots, s_{i,m}]$
  $\mathbf{s}_i \leftarrow \text{softmax}(\mathbf{s}_i)$
  $\mathbf{a}_i \leftarrow [\hat{\mathbf{o}}_1, \cdots, \hat{\mathbf{o}}_m](\mathbf{s}_i)$
end for
$\text{s}(S, I) \leftarrow \sum_{i=1}^{n} \frac{\hat{\mathbf{w}}_i^\top \mathbf{a}_i}{\|\hat{\mathbf{w}}_i\|_2 \|\mathbf{a}_i\|_2}$
return $\text{s}(S, I)$

and matching should be conducted on the image side. To achieve excellent performance in both text-to-image and image-to-text retrieval, SCAN [20] ensembles two models which are trained by image-side alignment and text-side alignment, respectively. We can also improve the model performance by ensembling models trained on different settings, but that is not the focus of this paper.

In Algorithm 1, we summarize the pipeline of the proposed HAN to obtain the final relevance between an image and a text. In the training phase, the relevance score between the image and the text is used to compute the loss. In the testing phase, the relevance score is used for ranking.

3.4 Loss module

The loss function is computed by a mini-batch of text-image pairs $\{I_i, T_i\}_{i=1}^K$, where T_i is the ground truth text corresponding to the image I_i. Meanwhile, we assume that, in the mini-batch, the images except I_i in the mini-batch is irrelevant to the text T_i. We define S_{ij} as the relevance score between i-th image and the j-th computed by S_{AVG} defined in Eq. (1). Thus, the training targets to increase the relevance score between an image and its corresponding text, and meanwhile decrease the relevance score between an image and its irrelevant texts. Straightforwardly, we can use the hinge-based triplet loss [3] to achieve this goal:

$$\mathcal{L}_{\text{sum}} = \sum_{i=1}^{K} \Big\{ \sum_{j \neq i} [S_{ij} - S_{ii} + \mu]_+ + \sum_{i \neq j} [S_{ij} - S_{jj} + \mu]_+ \Big\}, \quad (2)$$

where $[x]_+ = \max(x, 0)$ is a clip function and μ is a positive constant which we term as margin. The above triplet loss consists of two parts. The first part penalizes all negative texts given the query image, and the second part penalizes the negative images given the query text. To further emphasize hard negatives, VSE++ [7] defines the max triplet loss defined by

$$\mathcal{L}_{\text{max}} = \sum_{i=1}^{K} \Big\{ \max_{j \neq i} [S_{ij} - S_{ii} + \mu]_+ + \max_{i \neq j} [S_{ij} - S_{jj} + \mu]_+ \Big\}. \quad (3)$$

By emphasizing the hard negative pairs, the max triplet loss has achieved better performance than original triplet loss [7]. Nevertheless, the max triplet loss only considers the hardest negative pairs and thus it might suffer from mode collapse [43]. To emphasize the hard negatives and meanwhile consider all the negative pairs, we propose a soft-max triplet loss:

$$\mathcal{L}_{\text{soft}} = \sum_{i=1}^{K} \Big\{ [\sum_{j \neq i} [S_{ij} - S_{ii} + \mu]_+^p]^{1/p} + [\sum_{i \neq j} [S_{ij} - S_{jj} + \mu]_+^p]^{1/p} \Big\}, \quad (4)$$

where $p > 1$ is a constant to control the softness of the loss function. Intuitively, in Eq. (4) we can rewrite

$$[S_{ij} - S_{ii} + \mu]_+^p = w_{ij} [S_{ij} - S_{ii} + \mu]_+,$$

where

$$w_{ij} = [S_{ij} - S_{ii} + \mu]_+^{p-1}.$$

Since $p > 1$, the larger $[S_{ij} - S_{ii} + \mu]$ leads to a larger w_{ij}. That is, it gives a higher weight to a harder example. Meanwhile, it takes all negative examples into consideration. In fact, when $p = 1$, the proposed soft-max hinge loss $\mathcal{L}_{\text{soft}}$ in Eq. (4) degenerates to the original hinge-based triplet loss $\mathcal{L}_{\text{sum}}$ defined in Eq. (2). When $p \to +\infty$, $\mathcal{L}_{\text{soft}}$ degenerates to $\mathcal{L}_{\text{max}}$ defined in Eq. (3).

4 EXPERIMENTS

We first introduce the datasets and settings. Then we compare with existing methods to demonstrate the effectiveness of the proposed HAN. After that, systematic ablation studies are demonstrated.

4.1 Datasets and settings

Datasets. We conduct experiments on two public benchmark datasets, MS-COCO [25] and Flickr30K [45]. Flickr30K dataset contains 31,000 images collected from Flickr website. Each image is paired with five captions. Following the split in [16], we select 1,000 images for validation and another 1,000 images for testing and use the rest for training. MS-COCO dataset consists of 123,287 images, and each image is paired with five text descriptions. We follow [16] to divide the dataset into 82,783 training samples, 5,000 validation samples and 5,000 test samples. We follow [7] which adds 30,504 samples in the validation set of MS-COCO into the training set. All the results are reported by averaging over 5 folds of 1K test images. We evaluate the cross-modal retrieval performance through image-to-text and text-to-image recall@K. Though recall@1 is the most widely used metric in real-world applications, we also report recall@{5, 10}, and the average value of recall@{1, 5, 10}.

Settings. Following [20], we detect the bounding boxes by Faster-RCNN [32] built on ResNet101 pre-trained by Anderson *et al.* [1] on Visual Genomes [18]. The feature dimension of the detected bounding box is 2048. After the fully-connected layer, the feature dimension of the bounding box is 512. Meanwhile, the word feature dimension after word embedding is 300. The dimension of the hidden state of the Bi-GRU is set as 512. We use k-means clustering to group bounding box features into 32 clusters and set the number of attention heads to 4. We set $p = 4, 8$, the number heterogeneous attention layers to 3, 1 for MS-COCO data and Flickr30K data respectively. We use Adam as the optimizer. The initial learning rate (LR) is set as 0.0001. We decrease LR to 0.00001 in the 10th epoch, and the whole training process finished in 20th epoch. All experiments are conducted on a single NVIDIA V100 GPU card and we set batch size as 64 for all experiments by default. The metric for evaluation is recall@K. We report recall@{1, 5, 10}, and their average value. Among them, recall@1 is the most important metric since it directly influences the users' experience. Meanwhile, we conduct two types of cross-modal retrieval tasks, text-to-image retrieval and image-to-text retrieval. Since text-to-image retrieval is more useful in real applications, the performance of text-to-image retrieval is of more importance in evaluating the cross-modal retrieval model.

4.2 Performance comparison

First, we compare with methods based on global features, DSPE [42] and VSE++ [7]. As shown in Table 1, the methods using global features are not as competitive as the methods based on local features such as SCO [15], SCAN [20] and PVSE [36] and our HAN. For instance, VSE++ only achieves a 52.0 recall@1 in the text-to-image retrieval. In contrast, SCAN achieves a 56.4 recall@1. By fusing the outputs of two SCAN models, SCAN ensemble achieves a better performance than the single-model SCAN. Compared with ensembled

Table 1: Comparisons with other methods. The best is in bold, while the best within each group is underlined.

Method	External data	MS-COCO						Flickr30K					
		text2image recall@			image2text recall@			text2image recall@			image2text recall@		
		1	5	10	1	5	10	1	5	10	1	5	10
DSPE [42]	No	39.6	75.2	86.9	50.1	79.7	89.2	29.7	60.1	72.1	40.3	68.9	79.9
VSE++ [7]	No	52.0	–	92.0	64.6	–	95.7	39.6	–	79.5	52.9	–	87.2
SCO [15]	No	56.7	87.5	94.8	69.9	92.9	97.5						
SCAN [20]	No	56.4	87.0	93.9	70.9	94.5	97.5	45.8	74.4	83.0	61.8	87.5	93.7
SCAN ensemble [20]	No	58.8	88.4	94.8	72.7	94.8	98.4	48.6	77.7	85.2	67.4	90.3	95.8
PVSE [36]	No	55.2	86.5	93.7	69.2	91.6	96.6	–	–	–	–	–	–
ViLBERT [26]	No	–	–	–	–	–	–	45.5	76.8	85.0	–	–	–
HAN (ours)	No	65.4	90.5	95.3	78.7	96.4	98.8	54.8	81.1	87.4	74.1	92.4	96.4
Unicoder-VL [22]	text only	63.9	91.6	96.5	75.1	94.3	97.8	57.8	82.2	88.9	73.0	89.0	94.1
ViLBERT [26]	text+image	–	–	–	–	–	–	58.2	84.9	91.5	–	–	–
Unicoder-VL [22]	text+image	**69.7**	**93.5**	**97.2**	**84.3**	**97.3**	**99.3**	**71.5**	**90.9**	**94.9**	**86.2**	**96.3**	**99.0**

Table 2: Influence of the number of bounding boxes. The experiments are conducted on two cases: 1) without heterogeneous layer, 2) with 1 heterogeneous layer.

	without heterogeneous layer								with 1 heterogeneous layer							
	text2image recall@				image2text recall@				text2image recall@				image2text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
36	58.4	87.4	93.3	79.7	73.8	95.2	98.2	89.1	61.6	89.0	94.4	81.7	77.7	95.6	98.5	90.6
64	60.7	87.4	93.9	80.7	74.4	95.8	98.7	89.6	63.0	90.4	95.7	83.0	77.5	96.2	98.5	90.7
100	61.0	88.0	94.0	81.0	75.5	96.1	98.4	90.0	63.4	90.0	95.3	83.0	79.7	96.5	98.3	91.5

SCAN, our single-model HAN achieves a higher retrieval accuracy. For example, in the text-to-image retrieval task, HAN achieves a 65.3 recall@1, whereas the SCAN ensemble only achieves a 58.8 recall@1. The better performance of ours is owing to the proposed soft-max triplet loss and the heterogeneous attention layers.

We further compare with cross-modal BERT methods including ViLBERT [26] and Unicoder-VL [22]. As shown in Table 1, without being pre-trained on external data, VILBERT is even worse than SCAN. The worse performance might be caused by over-fitting since ViLBERT contains a large number of parameters. In parallel, Unicoder-VL initializes the weights by a pre-trained standard BERT model and fine-tune the model on the target task. Thanks to the knowledge transferred from the pre-trained standard BERT, it effectively alleviates the over-fitting and achieves much better performance than VILBERT without pre-training. But Unicoder-VL, when only exploiting weight initialization, is still not as good as ours on the MS-COCO dataset. Meanwhile, Unicoder-VL stacks 12 transformer blocks, which is much more computationally costly than ours. Finally, by using more text-image pairs from a large-scale external dataset, both Unicoder-VL and ViLBERT achieve a much better performance as expected. To be specific, Unicoder-VL is pre-trained by a merge of SBU Captions [30] and Conceptual Captions [35], consisting of 3.8 million image-caption pairs. ViLBERT is pre-trained by Conceptual Captions dataset with 3.3 million image-caption pairs. We believe that the proposed HAN can also be improved through being pre-trained on a large-scale text-image dataset. But due to limited computing resources, pre-training on such large-scale datasets is impractical for us. Improving the performance through pre-training is not the focus of this paper, either. Note that, by exploiting attention, the proposed HAN and other local-feature methods such as ViLBERT and Unicoder-VL take significantly more computational cost than methods based on global-feature methods such as DSPE and VSE++. To be specific, in the text-to-image retrieval scenario, the global-feature methods can pre-compute the image features in the corpus in advanced and only the query's feature is extracted online. In contrast, HAN and other local-feature methods need extract features for all images online.

4.3 Analysis on MS-COCO dataset

Influence of the number of detected bounding boxes. We conduct the ablation study on the number of detected bounding boxes. A larger number of bounding boxes has a stronger power in modeling the objects in the image. But it brings more computation and memory cost for training and inference. We vary the number of detected bounding boxes among $\{36, 64, 100\}$. To make the experiments more systematic, we study two cases: 1) without heterogeneous attention layer 2) with one heterogeneous attention layer. As shown in Table 2, the retrieval accuracy improves as the number of bounding boxes increases. Without heterogeneous attention layer, using 36 bounding boxes, the text-to-image recall@1 is only 58.4. Using 100 bounding boxes, the recall@1 achieves 61.0. Using a single heterogeneous attention layer, with 36 bounding boxes, the text-to-image recall@1 is 61.6 whereas 100 bounding boxes achieves a 63.4 recall@1. Note that, the increase of the number o bounding boxes also leads to an increase in memory and time cost. Thus, to maintain the high accuracy achieved by a huge number of bounding box features and meanwhile takes low memory and time cost, we conduct k-means cluster on 100 detected bounding boxes.

Table 3: Influence of the number of clusters. The experiments are conducted on two cases: 1) without heterogeneous layer, 2) with 1 heterogeneous layer.

	without heterogeneous layer								with 1 heterogeneous layer							
	text2image recall@				image2text recall@				text2image recall@				image2text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
16	60.0	87.5	93.9	80.5	73.0	95.5	98.2	88.9	61.3	89.5	94.9	81.9	75.9	95.5	98.3	89.9
32	60.5	88.0	94.0	80.8	73.3	95.8	98.6	89.2	63.3	90.2	95.4	83.0	77.2	96.4	98.7	90.8
100	61.0	88.0	94.0	81.0	75.5	96.1	98.4	90.0	63.4	90.0	95.3	83.0	79.7	96.5	98.3	91.5

Table 4: Influence of p in the proposed softmax hinge loss. The experiments are conducted on two cases: 1) without heterogeneous layer, 2) with 1 heterogeneous layer.

	without heterogeneous layer								with 1 heterogeneous layer							
	text2image recall@				image2text recall@				text2image recall@				image2text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
1	52.6	85.3	93.4	77.1	63.5	92.8	97.5	84.6	57.6	87.6	94.0	79.7	67.6	94.8	98.1	86.8
2	55.0	86.6	93.8	78.5	66.7	93.6	97.5	85.9	59.0	88.3	94.7	80.7	71.5	95.1	98.4	88.3
4	56.7	87.2	93.8	79.2	71.1	95.0	98.1	88.1	61.2	89.0	94.7	81.6	72.9	95.6	98.7	89.1
8	60.5	88.0	94.0	80.8	73.3	95.8	98.6	89.2	63.3	90.2	95.4	83.0	77.2	96.4	98.7	90.8
$+\infty$	56.5	87.0	93.6	79.0	70.0	94.7	97.7	87.5	60.7	88.5	94.3	81.2	73.1	95.6	97.8	88.8

Table 5: Influence of the number of heads.

	text-to-image recall@				image-to-text recall@			
	1	5	10	avg	1	5	10	avg
1	62.7	90.0	95.4	82.7	78.3	96.3	98.6	91.1
2	63.1	90.4	95.2	82.9	79.3	96.1	98.3	91.2
4	63.3	90.2	95.4	83.0	77.2	96.4	98.7	90.8
8	62.8	89.5	95.5	82.6	77.2	95.9	98.5	90.5
16	62.7	89.7	95.3	82.6	77.4	96.1	98.9	90.8

Influence of the number of clusters. We group 100 bounding box features into tens of clusters using k-means clustering, and use the cluster centers to replace the original bounding box features as the initial representation of an image. Intuitively, the similar bounding boxes will be grouped into the same cluster and thus the cluster centers serve as representative objects. By utilizing the clustering, the number of object features is significantly reduced, and meanwhile the main content is preserved. We evaluate the influence of the number of clusters. We vary the number of clusters between $\{16, 32\}$ and compare with the baseline using original 100 bounding boxes. As shown in Table 3, after clustering, the retrieval recall@1 increases as the number of clusters increases. This is expected, since more clusters can encode richer information. Meanwhile, using 32 clusters, the performance is comparable with the method using the original 100 bounding boxes. For instance, without heterogeneous attention layer, the text-to-image recall@1 of 100 bounding boxes is 60.5. Using a heterogeneous attention layer, 100 bounding boxes achieve a 63.4 text-to-image recall@1 and 32 cluster centers achieve 63.3 recall@1. In contrast, 32 cluster centers achieve a 61.0 recall@1. By default, we use 32 cluster centers on all experiments.

Influence of the number of heads in the heterogeneous attention layer. Standard transformer adopts a multi-head setting to encode richer information. We evaluate the influence of the number of heads in the cross-modal retrieval accuracy. We vary the number of heads among $\{1, 2, 4, 8, 16\}$ and use a single heterogeneous attention layer. As shown in Table 5, when the number of heads increases from 1 to 4, the text-to-image recall@1 increases. But when the head number increases from 4 to $\{8, 16\}$, the text-to-image recall@1 slightly drops. The worse performance might be caused by that too many heads diminish the capability of each head. By default, we set the head number as 4.

Influence of p in the proposed softmax hinge loss. As we mentioned, when $p = 1$, the proposed soft-max triplet loss degenerates the original triplet loss. Meanwhile, when $p \to +\infty$, the proposed soft-max triplet loss degenerates to the max triplet loss. We further evaluate the influence of p in the proposed soft-max triplet loss. We test it on two cases: 1) without heterogeneous attention layer and 2) using a heterogeneous attention layer. As shown in Table 4, $p = 8$ achieves the best performance, and it is considerably better than original triplet loss ($p = 1$) and max triplet loss ($p \to +\infty$). For instance, without heterogeneous attention layer, when $p = 8$, the text-to-image recall@1 is 60.5, whereas the original triplet loss ($p = 1$) only attains a 52.6 recall@1 and the max triplet loss ($p \to +\infty$) only obtains a 56.5 recall@1. With a heterogeneous attention layer, when $p = 8$, the text-to-image recall@1 is 63.3. The original triplet loss ($p = 1$) only achieves a 57.6 recall@1 and the max triplet loss ($p \to +\infty$) only attains a 60.7 recall@1, both of which are considerably worse than that achieved by $p = 8$. The better performance when $p = 8$ demonstrates the advantage of the proposed softmax margin loss in negative mining.

Influence of the number of heterogeneous attention layers. Stacking more heterogeneous attention layers can exploit the cross-modal attention more effectively. But it will increase the memory and computation cost. To demonstrate the effectiveness of stacking heterogeneous attention layers, we conduct experiments by

Table 6: Influence of the number of heterogeneous attention layers. Experiments are conducted on two cases: 1) $p = 4$, 2) $p = 8$.

	$p = 4$								$p = 8$							
	text2image recall@				image2text recall@				text2image recall@				image2text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
0	56.7	87.2	93.8	79.2	71.1	95.0	98.1	88.1	60.5	88.0	94.0	80.8	73.3	95.8	98.6	89.2
1	61.2	89.0	94.7	81.6	72.9	95.6	98.7	89.1	63.3	90.2	95.4	83.0	77.2	96.4	98.7	90.8
2	62.9	89.4	94.7	82.3	76.7	96.0	98.2	90.3	64.4	90.0	95.3	83.2	77.9	96.5	98.9	91.1
3	65.4	90.5	95.3	83.7	78.7	96.4	98.8	91.3	64.4	90.6	95.5	83.5	80.9	96.1	99.0	92.0
4	64.9	90.5	95.6	83.7	78.5	96.2	98.6	91.1	64.5	90.3	95.5	83.4	78.8	97.0	98.6	91.5

Table 7: Ablation study on the configurations of text encoder, we compare the Bi-GRU with Transformer. We compare with two settings of Transfomer. The Transforer row shows the performance without pre-training. The Transformer+ row shows the performance based on pre-training on the text dataset.

	without heterogeneous layer								with 1 heterogeneous layer							
	text2image recall@				image2text recall@				text2image recall@				image2text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
Bi-GRU	60.5	88.0	94.0	80.8	73.3	95.8	98.6	89.2	63.3	90.2	95.4	83.0	77.2	96.4	98.7	90.8
Transformer	54.0	85.7	93.8	77.8	63.2	92.3	97.7	84.4	57.3	88.1	95.0	80.1	70.7	95.2	98.3	88.1
Transformer+	59.2	88.2	93.6	80.3	71.2	93.8	97.7	87.6	62.4	90.1	95.6	82.7	75.4	94.8	97.8	89.3

increases the number of heterogeneous attention layers from 0 to 4. We conduct experiments on two cases, $p = 4$ and $p = 8$. As shown in Table 6, when the number of heterogeneous attention layers increases from 0 to 2, the retrieval performance improves consistently, which demonstrates the effectiveness of heterogeneous layers in cross-modal retrieval. Meanwhile, when the number of heterogeneous attention layers increases from 2 to 3, the text-to-image recall@1 saturates when $p = 8$. In contrast, when $p = 4$, the text-to-image recall@1 continues to increase as the number of heterogeneous attention layers increases from 2 to 3. By default, we set $p = 8$ when the number of heterogeneous attention layers is equal or less than 2 and set $p = 4$ otherwise. Meanwhile, we see the text-to-image recall@1 does not become better when the number of heterogeneous attention layer increases from 3 to 4. This might be caused by over-fitting caused by a huge number of parameters when the number of heterogeneous attention layers increases.

Text Encoder. We adopt Bi-GRU as the text encoder, an alternative is the transformer encoder used in ViLBERT [26] and LXMERT [38]. As we know, despite transformer has achieved excellent performance in many NLP tasks, it needs stack a large number of layers and need a large-scale training dataset to suppress over-fitting. To demonstrate the effectiveness of the Bi-GRU, we compare it with the baseline which replaces the Bi-GRU with a transformer consisting of 12 standard self-attention blocks. We test the performance of transformer on two settings: 1) the first settings randomly initialize the weights of the transformer; 2) the second setting initialize the weights of transformer by loading the weights pre-trained by BERT. To make a fair comparison, we make the other parts except the text encoder fixed. As shown in Table 7, the performance of Bi-GRU is considerably better than the randomly initialized transformer . For instance, without heterogeneous attention layer, using Bi-GRU, for the text-to-image retrieval, it achieves a 60.5 recall@1. In contrast, using the randomly initialized transformer, it only achieves a 54.0 recall@1 for the text-to-image retrieval. Meanwhile, using one heterogeneous attention layer, Bi-GRU achieves a 63.3 recall@1 in the text-to-image retrieval task, whereas the transformer only achieves 57.3 recall@1. The considerably worse performance of randomly initialized transformer validates our claim that transformer is more prone to over-fitting. By loading the pre-trained weights, transformer achieves a comparable retrieval accuracy with our method, but it takes significantly more time than the Bi-GRU encoder used in our method. As shown in Table 8, our Bi-GRU encoder takes 0.1s for 64-sample batch. In contrast, transformer takes 1.6s. The significant efficiency improvement can considerably enhance the user's experience in search.

Table 8: Time cost of our text encoder and transformer.

method	ours	transformer
time	0.1s	1.6s

Global versus local. As we mentioned, the existing vision BERT methods such as ViLBERT [26] and Unicoder-VL [22] are based on global matching between the global text feature and the global image features. In contrast, ours is based on the local matching between word features and bounding boxes features. To demonstrate the advantage of the local matching over the global matching, we compare ours with the baseline which replaces the local matching with the global matching in Table 9. Following Unicoder-VL [22], we

Table 9: Comparisons between global and local matching.

	text-to-image recall@				image-to-text recall@			
	1	5	10	avg	1	5	10	avg
Global	49.7	82.8	91.6	74.7	66.3	91.8	96.7	84.9
Local	63.3	90.2	95.4	83.0	77.2	96.4	98.7	90.8

Table 10: Influence of p in the proposed softmax hinge loss. Experiments are conducted on two cases: 1) without heterogeneous attention layer, 2) with 1 heterogeneous attention layer.

	without heterogeneous attention layer								with 1 heterogeneous layer							
	text-to-image recall@				image-to-text recall@				text-to-image recall@				image-to-text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
1	35.8	63.3	71.3	56.8	47.2	79.9	88.9	72.0	47.4	75.2	83.3	68.6	60.5	87.7	93.2	80.5
2	37.8	64.7	74.0	58.8	52.3	80.7	89.0	74.0	49.0	77.0	84.3	70.1	62.5	89.4	94.3	82.1
4	48.2	76.6	84.8	69.9	66.1	88.9	95.2	83.4	51.4	77.4	85.1	71.3	67.2	90.8	94.8	84.3
8	50.3	77.5	84.9	70.9	67.8	90.6	94.6	84.3	54.8	81.1	87.4	74.4	74.1	92.4	96.4	87.6
$+\infty$	47.1	76.0	84.6	69.2	63.9	88.9	94.3	82.4	50.0	77.2	84.0	70.4	66.5	88.2	94.2	83.0

Table 11: Influence of the number of heterogeneous attention layers with two cases: 1) $p = 4$, 2) $p = 8$.

	$p = 4$								$p = 8$							
	text-to-image recall@				image-to-text recall@				text-to-image recall@				image-to-text recall@			
	1	5	10	avg	1	5	10	avg	1	5	10	avg	1	5	10	avg
0	48.2	76.6	84.8	69.9	66.1	88.9	95.2	83.4	50.3	77.5	84.9	70.9	67.8	90.6	94.6	84.3
1	51.4	77.4	85.1	71.3	67.2	90.8	94.8	84.3	54.8	81.1	87.4	74.4	74.1	92.4	96.4	87.6
2	51.3	77.0	84.6	71.0	66.9	90.1	94.5	83.8	50.6	79.1	86.4	72.0	65.1	90.4	95.3	83.6

add a CLS token to generate the global text feature and a SEP token to obtain the global image feature. The relevance between the image and the text is determined by the cosine similarity score between the features of CLS and SEP generated by the last heterogeneous attention layer. To make a fair comparison, we compare the local matching and the global matching with other parts fixed, and use a heterogeneous attention layer for both local matching and global matching. As shown in Table 9, the local matching considerably outperforms the global matching. To be specific, on the text-to-image retrieval task, the global matching only achieves a 49.7 recall@1, which is considerably lower than 63.3 recall@1 achieved by the local matching. The better performance achieved by local features than global features proves the effectiveness of finer-level matching achieved by local features.

4.4 Analysis on Flick30K dataset

Influence of p in the proposed softmax hinge loss. We test the influence of p in the proposed softmax hinge loss on two cases: 1) without heterogeneous attention layer and 2) using one heterogeneous attention layer. As shown in Table 10, $p = 8$ achieves the best performance on both cases, and it is considerably better than sum hinge loss ($p = 1$) and max hinge loss ($p \to +\infty$). For instance, without heterogeneous attention layer, when $p = 8$, the text-to-image recall@1 is 50.3, whereas the sum hinge loss ($p = 1$) only achieves a 35.8 recall@1 and the max hinge loss ($p \to +\infty$) only achieves a 47.1 recall@1. With a heterogeneous attention layer, when $p = 8$, the text-to-image recall@1 is 54.8, whereas the sum hinge loss ($p = 1$) only achieves a 47.4 recall@1 and the max hinge loss ($p \to +\infty$) only attains a 50.0 recall@1. The better performance when $p = 8$ over the case when $p = 1$ or $p \to +\infty$ demonstrates the effectiveness of the proposed softmax hinge loss.

Influence of the number of heterogeneous attention layers. We conduct experiments by increasing the number of heterogeneous attention layers, from 0 to 2. We conduct experiments on two cases, $p = 4$ and $p = 8$. As shown in Table 11, when the number of heterogeneous attention layers increases from 0 to 1, the retrieval performance increases considerably, which demonstrates the effectiveness of the heterogeneous attention layers. For instance, when $p = 8$, without heterogeneous attention layers, it only achieves a 50.3 recall@1 in the text-to-image retrieval. In contrast, using one heterogeneous attention layer, our HAN achieves a 54.8 recall@1 in the text-to-image retrieval. Nevertheless, when the number of heterogeneous attention layers increases from 1 to 2, the retrieval accuracy becomes worse, especially on the case when $p = 8$. The worse performance might be caused by over-fitting. Thus, by default, we only use a single heterogeneous attention layer for small-scale datasets to suppress over-fitting. When training samples are enough, more heterogeneous attention layers are encouraged.

5 CONCLUSION

In this paper, we propose a Heterogeneous Attention Network (HAN) for effective and efficient cross-modal retrieval. It represents an image by a set of bounding box features and represents a sentence by a set of word features. The relevance between the image and the sentence is determined by the set-to-set matching between bounding box features and word features. To provide the cross-modal context in bounding boxes and word features, we propose a heterogeneous attention layer. The global context enhances the bounding box and word features and improves the effectiveness of matching. Meanwhile, we propose a soft-max hinge loss, which emphasizes hard samples but does not discard the relatively easy negative pairs, effectively improving the training effectiveness. Meanwhile, HAN relies on Bi-GRU, which is considerably more efficient than existing mainstream cross-modal BERT methods using a stack of many transformer layers. The lightweight HAN only needs a single V100 GPU card for training. Systematic experiments conducted on two public benchmark datasets, MSCOCO and Flickr30K, demonstrate the effectiveness and efficiency of the proposed HAN.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, 6077–6086.

[2] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. 2020. A Stochastic Treatment of Learning to Rank Scoring Functions. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining (WSDM)*. Houston, TX, 61–69.

[3] Olivier Chapelle, Quoc Le, and Alex Smola. 2007. Large margin optimization of ranking measures. In *NIPS workshop: Machine learning for Web search*.

[4] Jaemin Cho, Jiasen Lu, Dustin Schwenk, Hannaneh Hajishirzi, and Aniruddha Kembhavi. 2020. X-LXMERT: Paint, Caption and Answer Questions with Multi-Modal Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online, 8785–8805.

[5] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation* (2014), 103.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Minneapolis, MN, 4171–4186.

[7] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*. Newcastle, UK.

[8] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. MOBIUS: Towards the Next Generation of Query-Ad Matching in Baidu's Sponsored Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. Anchorage, AK, 2509–2517.

[9] Hongliang Fei, Tan Yu, and Ping Li. 2021. Cross-lingual Cross-modal Pretraining for Multimodal Retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Online.

[10] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-Scale Adversarial Training for Vision-and-Language Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. virtual.

[11] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* 12, 10 (2000), 2451–2471.

[12] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A Multi-View Embedding Space for Modeling Internet Images, Tags, and Their Semantics. *Int. J. Comput. Vis.* 106, 2 (2014), 210–233.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, 770–778.

[14] Weixiang Hong, Qingpei Guo, Wei Zhang, Jingdong Chen, and Wei Chu. 2021. Neural Networks for Information Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Online.

[15] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. 2018. Learning Semantic Concepts and Order for Image and Sentence Matching. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, 6163–6171.

[16] Andrej Karpathy and Li Fei-Fei. 2017. Deep Visual-Semantic Alignments for Generating Image Descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (2017), 664–676.

[17] Tom Kenter, Alexey Borisov, Christophe Van Gysel, Mostafa Dehghani, Maarten de Rijke, and Bhaskar Mitra. 2017. Neural Networks for Information Retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Shinjuku, Tokyo, 1403–1406.

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* 123, 1 (2017), 32–73.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*. Lake Tahoe, NV, 1106–1114.

[20] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked Cross Attention for Image-Text Matching. In *Proceedings of the 15th European Conference on Computer Vision (ECCV), Part IV*. Munich, Germany, 212–228.

[21] Dongge Li, Nevenka Dimitrova, Mingkun Li, and Ishwar K. Sethi. 2003. Multimedia content processing through cross-modal association. In *Proceedings of the Eleventh ACM International Conference on Multimedia (ACMMM)*. Berkeley, CA, 604–611.

[22] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. New York, NY, 11336–11344.

[23] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019).

[24] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *Proceedings of the 16th European Conference on Computer Vision (ECCV), Part XXX*. Glasgow, UK, 121–137.

[25] 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 *Proceedings of the 13th European Conference on Computer Vision (ECCV), Part V*. Zurich, Switzerland, 740–755.

[26] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 13–23.

[27] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-Task Vision and Language Representation Learning. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, 10434–10443.

[28] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353* (2020).

[29] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 2156–2164.

[30] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Advances in Neural Information Processing Systems (NIPS)*. Granada, Spain, 1143–1151.

[31] Nikhil Rasiwasia, José Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R. G. Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th International Conference on Multimedia (ACMMM)*. Firenze, Italy, 251–260.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (2015), 1137–1149.

[33] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, 815–823.

[34] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[35] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Melbourne, Australia, 2556–2565.

[36] Yale Song and Mohammad Soleymani. 2019. Polysemous Visual-Semantic Embedding for Cross-Modal Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, 1979–1988.

[37] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia.

[38] Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 5099–5110.

[39] Shulong Tan, Zhixin Zhou, Zhaozhuo Xu, and Ping Li. 2020. Fast Item Ranking under Neural Network based Measures. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining (WSDM)*. Houston, TX, 591–599.

[40] 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 *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, 5998–6008.

[41] Bokun Wang, Yang Yang, Xing Xu, Alan Hanjalic, and Heng Tao Shen. 2017. Adversarial Cross-Modal Retrieval. In *Proceedings of the 2017 ACM on Multimedia Conference (ACMMM)*. Mountain View, CA, 154–162.

[42] Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning Deep Structure-Preserving Image-Text Embeddings. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, 5005–5013.

[43] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. 2017. Sampling Matters in Deep Embedding Learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy, 2859–2867.

[44] Zhiqiang Xu, Dong Li, Weijie Zhao, Xing Shen, Tianbo Huang, Xiaoyun Li, and Ping Li. 2021. Agile and Accurate CTR Prediction Model Training for Massive-Scale Online Advertising Systems. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*. Virtual Event, Xi'an, Shaanxi, China.

[45] 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. *Trans. Assoc. Comput. Linguistics* 2 (2014), 67–78.

[46] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-ViL: Knowledge Enhanced Vision-Language Representations Through Scene Graph. *arXiv preprint arXiv:2006.16934* (2020).

[47] Puxuan Yu, Hongliang Fei, and Ping Li. 2021. Cross-lingual Language Model Pretraining for Retrieval. In *Proceedings of the Web Conference (WWW)*. Ljubljana, Slovenia.

[48] Tan Yu, Xuemeng Yang, Yan Jiang, Hongfang Zhang, Weijie Zhao, and Ping Li. 2021. TIRA in Baidu Image Advertising. In *Proceedings of the 37th International Conference on Data Engineering (ICDE)*.

[49] Tan Yu, Yi Yang, Yi Li, Xiaodong Chen, Mingming Sun, and Ping Li. 2020. Combo-Attention Network for Baidu Video Advertising. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. Virtual Event, CA, USA, 2474–2482.

[50] Hamed Zamani, Bhaskar Mitra, Xia Song, Nick Craswell, and Saurabh Tiwary. 2018. Neural Ranking Models with Multiple Document Fields. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*. Marina Del Rey, CA, 700–708.

[51] Weijie Zhao, Deping Xie, Ronglai Jia, Yulei Qian, Ruiquan Ding, Mingming Sun, and Ping Li. 2020. Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems. In *Proceedings of the 3rd Conference on Machine Learning and Systems (MLSys)*. Austin, TX.

[52] Weijie Zhao, Jingyuan Zhang, Deping Xie, Yulei Qian, Ronglai Jia, and Ping Li. 2019. AIBox: CTR Prediction Model Training on a Single Node. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. Beijing, China, 319–328.