Exploring images with deep learning for classification, retrieval and synthesis

Liu, Y.

Author: Liu, Y.
Title: Exploring images with deep learning for classification, retrieval and synthesis
Issue Date: 2018-10-24

Chapter 6

Cycle-consistent Embeddings for Cross-modal Retrieval

In the previous chapter, we have exploited an image-text matching network to correlate visual-textual features in a latent embedding space. In this chapter, we further address how we can preserve inter-modal correlations and intra-modal consistency while matching visual and textual representations (RQ5).

To narrow the modality gap between vision and language, prior approaches attempt to discover their correlated semantics in a common feature space. However, these approaches omit the intra-modal semantic consistency when learning the inter-modal correlations. To address this problem, we propose cycle-consistent embeddings in a deep neural network for matching visual and textual representations. Our approach named as CycleMatch can maintain both inter-modal correlations and intra-modal consistency by cascading dual mappings and reconstructed mappings in a cyclic fashion. Moreover, in order to achieve a robust inference, we propose to employ two late-fusion approaches: average fusion and adaptive fusion. Both of them can effectively integrate the matching scores of different embedding features, without increasing the network complexity and training time. In the experiments on cross-modal retrieval, we demonstrate comprehensive results to verify the effectiveness of the proposed approach. Our approach achieves state-of-the-art performance on two well-known multi-modal datasets, Flickr30K and MSCOCO.

Keywords

Cross-modal retrieval, Embedding, Deep neural networks, Late fusion

6.1 Introduction

Nowadays, the explosive growth of multimedia data in social networks (*e.g.* image, video, text and audio) have triggered a massive amount of research activities in multi-modal understanding and reasoning. In this chapter, we focus on the task of image-text matching, which aims to incorporate heterogeneous representations from visual and textual modalities. In practice, this task plays an essential role in a wide variety of vision-and-language applications, for examples, cross-modal retrieval [192, 193], visual question answering [58, 194], zero-shot recognition [195, 196] and visual grounding [197, 198].

The core issue with image-text matching is searching for an appropriate embedding space where related images and texts can be matched correctly. Driven by the great strides made by deep learning [4, 7, 10], recent research has been dedicated to exploring deep neural networks for learning powerful embedding features, in order to narrow the modality gap between visual and textual domains. These networks are typically composed of two branches for generating visual and textual embedding features in a common latent space, respectively [53, 64, 65, 67, 68]. Then, a similarity-based ranking loss is used to measure the latent embedding features. Latent embeddings can distill common semantic information about both the visual content and textual description. To directly match the similarities between vision and language, researchers further exploit dual embeddings by translating an input feature in the source space to be the feature in the target space [71, 72, 76, 77]. Both the latent and dual embeddings can capture inter-modal semantic correlations, however, they are limited in preserving intra-modal semantic consistency. Our motivation for this work is that: *A robust embedding method should be able to learn representations of both the source and target modalities.* Inspired by this motivation, in this chapter we focus on solving the fifth research question **RQ 5: How can we preserve both inter-modal correlations and intra-modal consistency for learning robust visual and textual embeddings?**

Inspired by the idea of cycle-consistent learning [94, 199], we propose cycle-consistent embeddings in an image-text matching network, which can incorporate both *inter-modal correlations* and *intra-modal consistency* for learning robust visual and textual embeddings. Figure 6.1 illustrates our embedding method by integrating three feature embeddings, including dual, reconstructed and latent embeddings. Specifically, it has two cycle branches, one starting from an image feature in the visual space and the other from a text feature in the textual space. For each branch, it first accomplishes a dual mapping by translating an input feature in the source space to be a dual embedding in the target space. Inverse to the dual mapping, we then exploit a reconstructed mapping, with the aim of translating the dual embedding back to the source space. Moreover, we learn a latent space during the dual and reconstructed mappings and correlate the latent embeddings. In the three feature

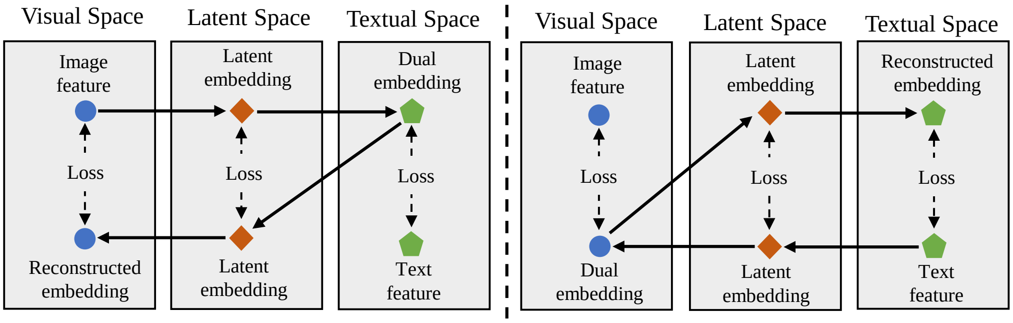


Figure 6.1: Schematic pipeline of our proposed cycle-consistent embedding method. It is composed of two cycle branches starting from (Left) visual space and (Right) textual space, respectively. We first perform a dual mapping by transforming the input feature into the target feature space. Then the dual embedding is used to generate a reconstructed embedding in a reconstructed mapping. In addition, we construct a latent space to correlate latent embeddings of the two mappings. The two branches share the mapping functions for transformations between three feature spaces, and can be trained jointly by optimizing the matching losses in the three feature spaces.

spaces, we compute their ranking losses to jointly optimize the whole embedding learning. Consequently, our visual-textual embedding method can learn not only *inter-modal mappings* (*i.e.* image-to-text and text-to-image), but also *intra-modal mappings* (*i.e.* image-to-image and text-to-text).

The contributions of this work are as follows:

- We propose a novel deep cycle-consistent embedding network for image-text matching. Our approach called CycleMatch can cascade dual and reconstructed mappings together to maintain inter-modal correlations and intra-modal consistency. To our best knowledge, this is the first work to explore the usage of cycle consistency for solving the task of image-text matching.

- To improve the inference at the test stage, we present two late-fusion approaches to efficiently integrate the matching scores of multiple embedding features without increasing the training complexity.

- In the experiments, our cycle-consistency embedding outperforms traditional embeddings with considerable improvements for cross-modal retrieval on two multi-modal datasets, *i.e.* Flickr30K and MSCOCO. In addition, our results are competitive with the state-of-the-art approaches.

The rest of this chapter is structured as follows. Related works are introduced in Section 6.2. Section 6.3 details the proposed CycleMatch. The experimental results are reported in Section 6.4. Finally, Section 6.5 summarizes the conclusions.

6.2 Related Work

Our work is related to the image-text matching methods based on deep neural networks, and other works about cycle-consistent learning.

Deep visual-textual embeddings

With the increasing progress of deep learning, research efforts have been made to CCA into deep neural networks [49, 50, 51, 62]. However, most deep CCA models rely on expensive decorrelation computations, which limit their generalization abilities at large-scale data. Alternatively, a number of recent approaches [52, 55, 64, 65, 66] address the task by designing two-branch networks to embed visual and textual features into a common latent space, and then learn latent embeddings by optimizing a ranking loss between matched and unmatched image-text pairs. For instance, Wang *et al.* [53] built a simple and efficient matching network to preserve the structure relations between images and texts in the latent space. To associate image regions with words, the attention mechanism was integrated into visual-textual embedding models [67, 68]. In addition to the pairwise ranking loss, recent approaches [69, 70] leveraged extra loss functions to enhance the discrimination of the learned embedding features.

Another line of research [71, 72, 73, 74, 75] focuses on learning dual embeddings between two modalities, *e.g.* projecting visual features into the textual feature space and vice versa. Essentially, the dual embedding models are motivated by autoencoders. For instance, Feng *et al.* [71] proposed a correspondence cross-modal autoencoder model. 2WayNet [76] built the projections between two modalities and regularized them with Euclidean loss. Recently, the work of Gu *et al.* [77] utilized two generative models to synthesize grounded visual and textual representations. Also, Huang *et al.* [200] jointly modeled image-sentence matching and sentence generation. Note that, latent embeddings can be additionally used in the dual embedding models to enhance cross-modal relations.

In contrast to the above studies, our approach builds a reconstructed mapping upon the dual mapping, and generates cycle-consistent embeddings that are beneficial to the process of matching visual-textual representations. In Figure 6.2, we show the differences of our model from previous works.

Cycle-consistent learning

There are a few papers exploring cycle consistency for diverse applications [94, 199, 201, 202, 203]. They are mainly motivated by the fact that, cycle-consistent learning is encouraged to produce additional feedback signals to improve the bi-directional translations. Specifically, He *et al.* [199] proposed a dual-learning mechanism based

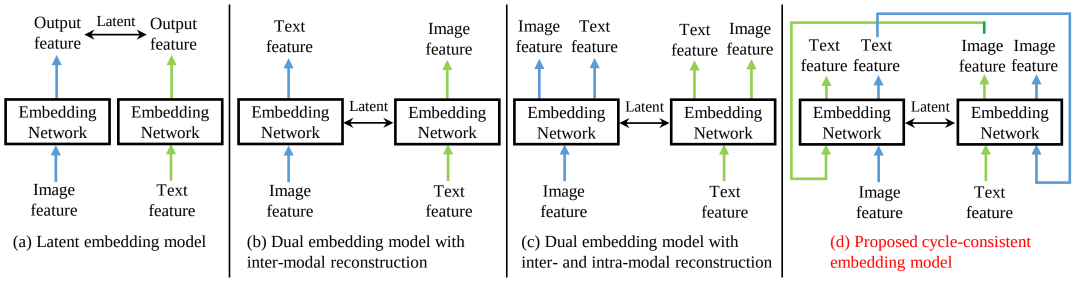


Figure 6.2: Conceptual illustration of variants of image-text matching models. (a) Latent embedding model. (b) Dual embedding model with inter-modal reconstruction. (c) Dual embedding model with inter-modal and intra-modal reconstruction. (d) Our cycle-consistent embedding model. Notice that the models in (b)(c)(d) also impose latent embeddings on hidden layers. Our model cascades the two embedding networks in a cyclic fashion, which can enhance interactions between two embedding networks.

on deep reinforcement learning, where one agent was used to learn the primal task, *e.g.* English-to-French translation, and the other agent for the dual task, *e.g.* French-to-English translation. More recently, Zhu *et al.* [94] exploited cycle-consistent adversarial networks (CycleGAN), which combined a cycle-consistency loss with an adversarial loss [79] to perform unpaired image-to-image translations between two different visual domains. A similar idea was also presented in [204, 205].

Although prior works have shown the effectiveness of using cycle-consistent constraints for intra-modal domain mappings, yet in the context of cross-modal representation learning, its effectiveness has not been well investigated. In contrast to prior approaches that utilize cycle-consistent constraints within one modality (*e.g.* neural machine translation and image-to-image translation), our work is the first to extend the usage of cycle consistency for learning visual-textual embeddings. The work of Chen and Zitnick [206] is relevant to ours, as their model can both generate textual captions and reconstruct visual features given an image representation. However, their model lacks the inverse cycle mapping, *i.e.* text-to-image-to-text, which can be jointly learned in our model. Last but not least, these existing works did not consider matching latent embeddings during the cycle-consistent scheme.

6.3 Cycle-consistent Embeddings

In this section, we present the proposed network (CycleMatch) with cycle-consistent embeddings for matching visual and textual representations. For a robust inference, we exploit two late-fusion approaches by taking advantage of multiple embedding features learned in the network.

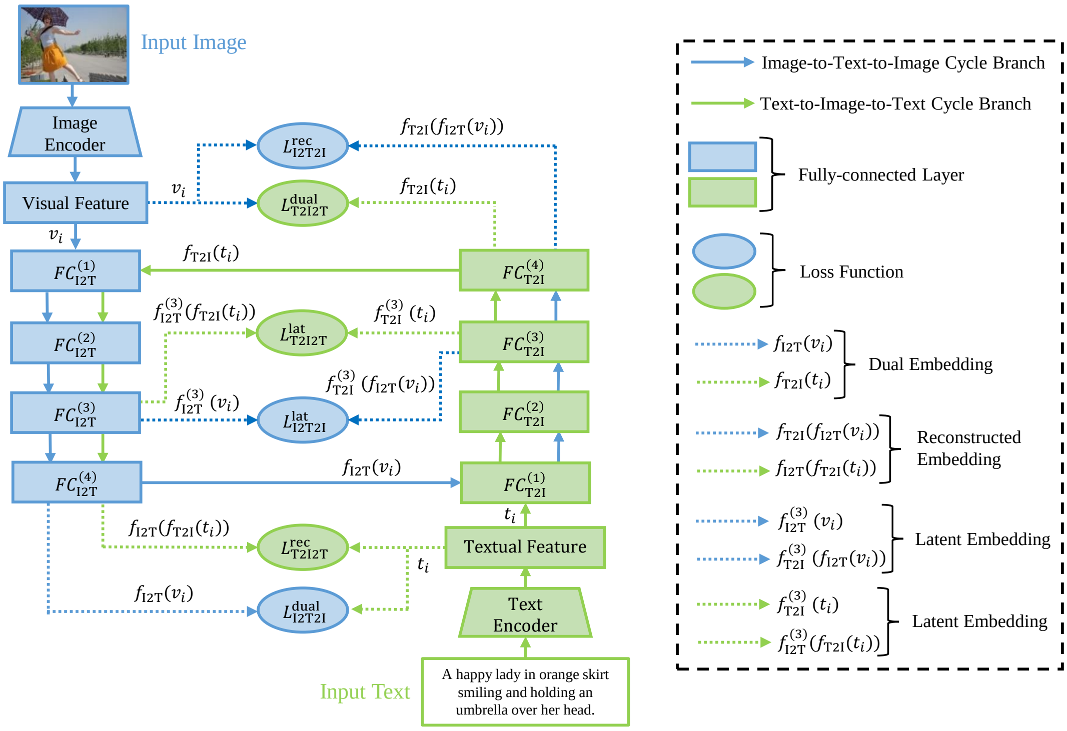

Input Image
Image Encoder
Visual Feature
v_i
$FC_{\text{I2T}}^{(1)}$
$FC_{\text{I2T}}^{(2)}$
$FC_{\text{I2T}}^{(3)}$
$FC_{\text{I2T}}^{(4)}$
$L_{\text{I2T2I}}^{\text{rec}}$
$L_{\text{T2I2T}}^{\text{dual}}$
$L_{\text{T2I2T}}^{\text{lat}}$
$L_{\text{I2T2I}}^{\text{lat}}$
$L_{\text{T2I2T}}^{\text{rec}}$
$L_{\text{I2T2I}}^{\text{dual}}$
$f_{\text{T2I}}(f_{\text{I2T}}(v_i))$
$f_{\text{T2I}}(t_i)$
$f_{\text{I2T}}^{(3)}(f_{\text{T2I}}(t_i))$
$f_{\text{T2I}}^{(3)}(t_i)$
$f_{\text{T2I}}^{(3)}(f_{\text{I2T}}(v_i))$
$f_{\text{I2T}}^{(3)}(v_i)$
$f_{\text{I2T}}(v_i)$
$f_{\text{I2T}}(f_{\text{T2I}}(t_i))$
$FC_{\text{T2I}}^{(4)}$
$FC_{\text{T2I}}^{(3)}$
$FC_{\text{T2I}}^{(2)}$
$FC_{\text{T2I}}^{(1)}$
t_i
Textual Feature
Text Encoder
Input Text
A happy lady in orange skirt smiling and holding an umbrella over her head.
Image-to-Text-to-Image Cycle Branch
Text-to-Image-to-Text Cycle Branch
Fully-connected Layer
Loss Function
$f_{\text{I2T}}(v_i)$
$f_{\text{T2I}}(t_i)$
Dual Embedding
$f_{\text{T2I}}(f_{\text{I2T}}(v_i))$
$f_{\text{I2T}}(f_{\text{T2I}}(t_i))$
Reconstructed Embedding
$f_{\text{I2T}}^{(3)}(v_i)$
$f_{\text{T2I}}^{(3)}(f_{\text{I2T}}(v_i))$
Latent Embedding
$f_{\text{T2I}}^{(3)}(t_i)$
$f_{\text{I2T}}^{(3)}(f_{\text{T2I}}(t_i))$
Latent Embedding

Figure 6.3: The proposed CycleMatch exploits two cycle branches for image-text matching. For each branch, it is divided into two sub-branches from the *fourth* FC layer (*i.e.* $FC_{\text{IT}}^{(4)}$ and $FC_{\text{TI}}^{(4)}$). One sub-branch continues accomplishing the dual mapping to the target feature space, while the other sub-branch is used to perform the reconstructed mapping back to the source feature space. Consequently, the cycle branches allow to jointly learn dual, reconstructed and latent embedding features. We can train the network end-to-end by optimizing several loss functions simultaneously.

6.3.1 System architecture

Figure 6.3 depicts an overview of the CycleMatch architecture. The entire network consists of three components: feature encoder, feature embedding and feature matching. First of all, given an input image I_i and text T_i, we employ individual feature encoders to extract the visual feature $\boldsymbol{v}_i = En_{\text{img}}(I_i)$ and textual feature $\boldsymbol{t}_i = En_{\text{text}}(T_i)$. Then, we develop several fully-connected (FC) layers (*i.e.* $FC_{\text{I2T}}^{(j)}$) to perform the *Image-to-Text* (I2T) mapping and several other FC layers (*i.e.* $FC_{\text{T2I}}^{(j)}$) for the *Text-to-Image* (T2I) mapping. Let $f_{\text{I2T}}(\cdot)$ and $f_{\text{T2I}}(\cdot)$ represent the mapping functions for I2T and T2I, respectively. In addition, connecting FC_{I2T} and FC_{T2I} can form two cycle mappings between the visual and textual feature spaces. Specifically, given $\boldsymbol{v}_i$, we first transform it to be $f_{\text{I2T}}(\boldsymbol{v}_i)$ in the textual feature space and then learn its reconstructed feature $f_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}_i))$ in the visual feature space. Moreover, we also correlate intermediate features derived from $FC_{\text{I2T}}^{(3)}$ and $FC_{\text{T2I}}^{(3)}$, so as to learn a latent feature space. Similarly, $\boldsymbol{t}_i$ is used to start another cycle mapping. In a nutshell, each cycle mapping can learn dual, reconstructed and latent embeddings in a cyclic fashion.

6.3.2 Formulation

Next, we will detail the above three embeddings and formulate their loss functions separately. The entire network contains two cycle-consistent embedding branches: one for *image-to-text-to-image* (I2T2I) mapping and the other for *text-to-image-to-text* (T2I2T) mapping. Here, we take the I2T2I mapping for an example.

Dual embedding

In a dataset collection with N image-text pairs, we take as input $\boldsymbol{v}_i$ into $FC^{(j)}_{\text{I2T}}$, where $i = 1, \ldots, N$ and $j = 1, \ldots, 4$, and generate the dual embedding $f_{\text{I2T}}(\boldsymbol{v}_i)$ in the textual space, which should have the same dimension as the ground-truth textual feature $\boldsymbol{t_i}$. Then, we need to normalize the two features and compute their similarity using the cosine distance

$$s(f_{\text{I2T}}(\boldsymbol{v}_i), \boldsymbol{t}_i) = \frac{f_{\text{I2T}}(\boldsymbol{v}_i) \cdot \boldsymbol{t}_i}{||f_{\text{I2T}}(\boldsymbol{v}_i)|| \cdot ||\boldsymbol{t}_i||}. \tag{6.1}$$

During training, it is important to construct a number of negative pairs, in addition to the positive pair. Thereby, we search for the top K negative samples in a mini-batch for both $f_{\text{I2T}}(\boldsymbol{v}_i)$ and $\boldsymbol{t}_i$, which are denoted with $f_{\text{I2T}}(\boldsymbol{v}^-_{i,k})$ and $\boldsymbol{t}^-_{i,k}$, respectively, where $k = 1, \ldots, K$. To learn dual mappings, we need to employ a pairwise ranking loss function with respect to positive and negative pairs:

$$\begin{aligned} &\mathcal{L}^{\text{dual}}_{\text{I2T2I}} = \\ &\sum_{i=1}^{N}\sum_{k=1}^{K}\Big\{\max\big[0, m - s(f_{\text{I2T}}(\boldsymbol{v}_i), \boldsymbol{t}_i) + s(f_{\text{I2T}}(\boldsymbol{v}_i), \boldsymbol{t}^-_{i,k})\big] \\ &+ \alpha \max\big[0, m - s(f_{\text{I2T}}(\boldsymbol{v}_i), \boldsymbol{t}_i) + s(f_{\text{I2T}}(\boldsymbol{v}^-_{i,k}), \boldsymbol{t}_i)\big]\Big\}, \end{aligned} \tag{6.2}$$

where m is a margin parameter and α adjusts the weights of the two loss terms. Ideally, the matched distance $s(f_{\text{I2T}}(\boldsymbol{v}_i), \boldsymbol{t}_i)$ should be smaller than any of the unmatched distances $s(f_{\text{I2T}}(\boldsymbol{v}_i), \boldsymbol{t}^-_{i,k})$ and $s(f_{\text{I2T}}(\boldsymbol{v}^-_{i,k}), \boldsymbol{t}_i)$.

Reconstructed embedding

In addition to learning inter-modal correlations from dual mappings, we further explore reconstructed mappings to maintain the intra-modal semantic consistency. We cascade the dual and reconstructed mappings to form an intra-modal autoencoder and minimize the reconstruction error based on the ranking loss instead of the traditional Euclidean loss. Specifically, we feed $f_{\text{I2T}}(\boldsymbol{v}_i)$ into $FC^{(j)}_{\text{T2I}}$, to produce a reconstructed embedding feature $\tilde{\boldsymbol{v}}_i$ in the visual feature space with

$$\tilde{\boldsymbol{v}}_i = f_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}_i)) = f_{\text{T2I}} \circ f_{\text{I2T}}(\boldsymbol{v}_i). \tag{6.3}$$

The ranking loss for making the reconstructed embedding feature $\tilde{\boldsymbol{v}}_i$ match with the original visual feature $\boldsymbol{v}_i$ can be written as follows

$$\begin{aligned} \mathcal{L}^{\text{rec}}_{\text{I2T2I}} =& \sum_{i=1}^{N}\sum_{k=1}^{K}\Big\{\max\big[0, m - s(\tilde{\boldsymbol{v}}_i, \boldsymbol{v}_i) + s(\tilde{\boldsymbol{v}}_i, \boldsymbol{v}^-_{i,k})\big] \\ &+ \alpha \max\big[0, m - s(\tilde{\boldsymbol{v}}_i, \boldsymbol{v}_i) + s(\tilde{\boldsymbol{v}}^-_{i,k}, \boldsymbol{v}_i)\big]\Big\}. \end{aligned} \tag{6.4}$$

Since $\mathcal{L}^{\text{rec}}_{\text{I2T2I}}$ also has an effect on the parameters of $FC^{(j)}_{\text{I2T}}$, the reconstructed mappings can help to improve the learning of dual mappings as well.

Latent embedding

Furthermore, we exploit a latent feature space to enhance the correlations between the dual and reconstructed mappings. Latent embeddings are able to distill common semantic information from visual and textual representations. Specifically, we make use of the intermediate representations from the third FC layers, *i.e.* $FC^{(3)}_{\text{I2T}}$ and $FC^{(3)}_{\text{T2I}}$. When $\boldsymbol{v_i}$ passes through $FC^{(3)}_{\text{I2T}}$, we can extract an intermediate feature $f^{(3)}_{\text{I2T}}(\boldsymbol{v}_i)$. Also, the dual embedding $f_{\text{I2T}}(\boldsymbol{v}_i)$ passes through $FC^{(3)}_{\text{T2I}}$ to generate another intermediate feature $f^{(3)}_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}_i))$. The ranking loss for matching latent embeddings thereby becomes

$$\begin{aligned} \mathcal{L}^{\text{lat}}_{\text{I2T2I}} = \sum_{i=1}^{N}\sum_{k=1}^{K}\Big\{&\max\Big[0, m - s\big(f^{(3)}_{\text{I2T}}(\boldsymbol{v}_i), f^{(3)}_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}_i))\big) \\ &+ s\big(f^{(3)}_{\text{I2T}}(\boldsymbol{v}_i), f^{(3)}_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}^-_{i,k}))\big)\Big] \\ &+ \alpha \max\Big[0, m - s\big(f^{(3)}_{\text{I2T}}(\boldsymbol{v}_i), f^{(3)}_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}_i))\big) \\ &+ s\big(f^{(3)}_{\text{I2T}}(\boldsymbol{v}^-_{i,k}), f^{(3)}_{\text{T2I}}(f_{\text{I2T}}(\boldsymbol{v}_i))\big)\Big]\Big\}. \end{aligned} \tag{6.5}$$

6.3.3 Full objective

Similar to the above I2T2I branch, it is straightforward to express the matching losses in the T2I2T branch, including $\mathcal{L}^{\text{dual}}_{\text{T2I2T}}$, $\mathcal{L}^{\text{rec}}_{\text{T2I2T}}$ and $\mathcal{L}^{\text{lat}}_{\text{T2I2T}}$. During training, we need to incorporate all the loss functions jointly. Finally, the full objective is to minimize the total loss:

$$\begin{aligned}&\underset{W_{\mathrm{I2T}},W_{\mathrm{T2I}}}{\arg\min}\ \mathcal{L}_{\mathrm{total}} = \\ &\mathcal{L}^{\mathrm{dual}}_{\mathrm{I2T2I}} + \mathcal{L}^{\mathrm{rec}}_{\mathrm{I2T2I}} + \mathcal{L}^{\mathrm{lat}}_{\mathrm{I2T2I}} + \mathcal{L}^{\mathrm{dual}}_{\mathrm{T2I2T}} + \mathcal{L}^{\mathrm{rec}}_{\mathrm{T2I2T}} + \mathcal{L}^{\mathrm{lat}}_{\mathrm{T2I2T}},\end{aligned} \tag{6.6}$$

where W_{I2T} and W_{T2I} indicate the parameters in $FC^{(j)}_{\mathrm{I2T}}$ and $FC^{(j)}_{\mathrm{T2I}}$, respectively. They are unshared due to the specialization of two different modalities. To demonstrate the effectiveness of our CycleMatch, we utilize the t-SNE [207] algorithm to visualize the embedding features learned in the visual, textual and latent feature spaces, separately. As shown in Figure 6.4, we randomly select 100 image-text pairs from the Flickr30K dataset [189]. From all the feature maps, we can visibly observe high similarities between two matched samples.

6.3.4 Late-fusion inference

By performing cycle-consistent embeddings, we can represent one sample with a set of three different features, for instance, $\{\boldsymbol{v}_i, f_{\mathrm{I2T}}(\boldsymbol{v}_i), f^{(3)}_{\mathrm{I2T}}(\boldsymbol{v}_i)\}$ for an image. Since the reconstructed embedding $\tilde{\boldsymbol{v}}_i$ and the other latent embedding $f^{(3)}_{\mathrm{T2I}}(f_{\mathrm{I2T}}(\boldsymbol{v}_i))$ are related to $\boldsymbol{v}_i$ and $f^{(3)}_{\mathrm{I2T}}(\boldsymbol{v}_i)$, we do not consider them for simplicity. Each of the three features can be used to measure an image-text matching score. Instead of using only one score, it is encouraged to leverage different scores together to achieve a more robust inference. This is driven by the late-fusion technique [208] in multimedia retrieval, which is a simple and efficient approach to combine the prediction scores of individual features. In this work, we present two effective late-fusion approaches, namely average fusion and adaptive fusion.

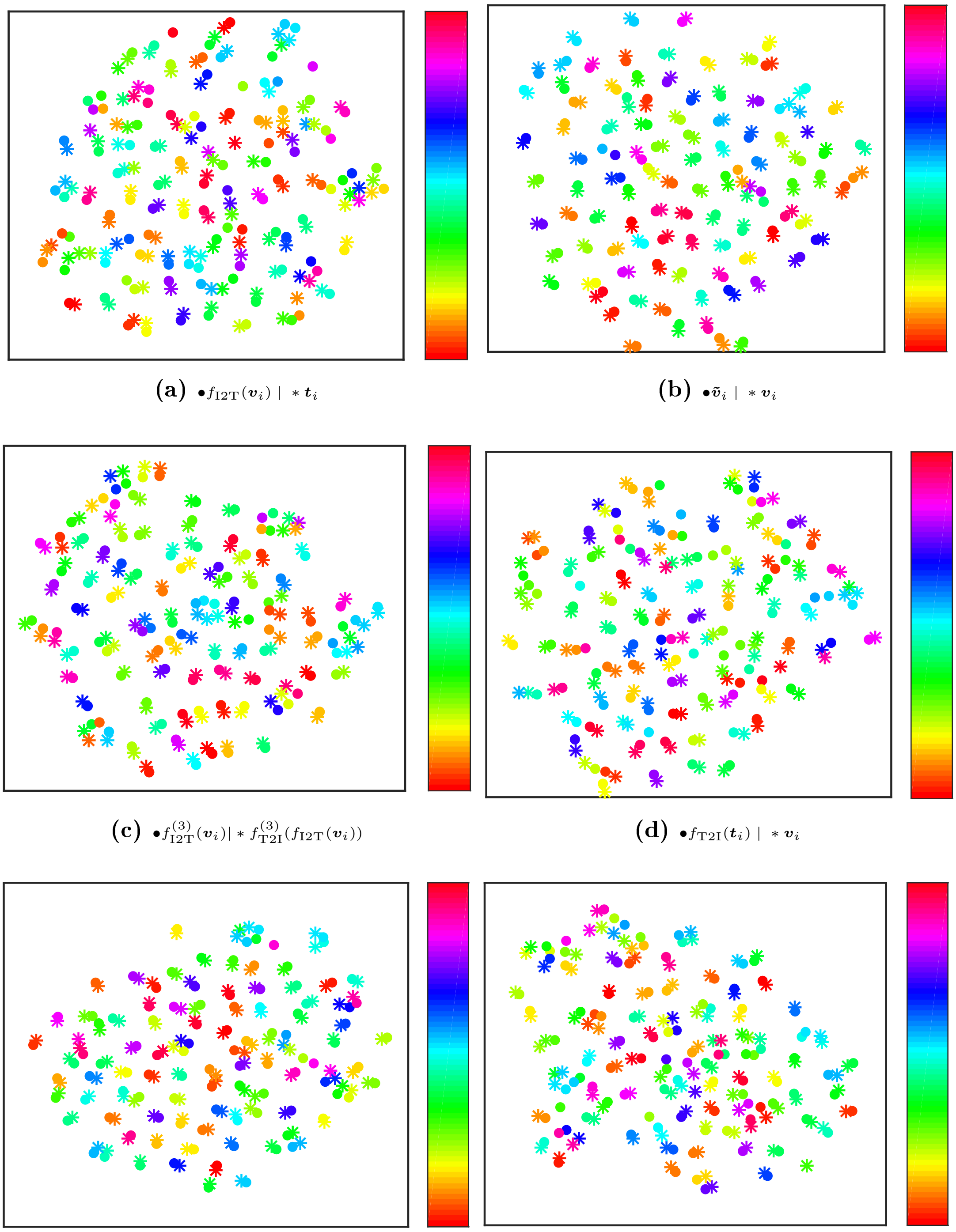
Average fusion

Given a query image I_q, we extract three features $\{\boldsymbol{v}_q, f_{\mathrm{I2T}}(\boldsymbol{v}_q), f^{(3)}_{\mathrm{I2T}}(\boldsymbol{v}_q)\}$. Similarly, an arbitrary text T_i in the dataset can be described with $\{\boldsymbol{t}_i, f_{\mathrm{T2I}}(\boldsymbol{t}_i), f^{(3)}_{\mathrm{T2I}}(\boldsymbol{t}_i)\}$. We can compute three similarity scores between I_q and T_i:

$$\begin{cases}\text{visual score} : s^{(1)}(\boldsymbol{v}_q, \boldsymbol{t}_i) = s(\boldsymbol{v}_q, f_{\mathrm{T2I}}(\boldsymbol{t}_i)), \\ \text{textual score} : s^{(2)}(\boldsymbol{v}_q, \boldsymbol{t}_i) = s(f_{\mathrm{I2T}}(\boldsymbol{v}_q), \boldsymbol{t}_i), \\ \text{latent score} : s^{(3)}(\boldsymbol{v}_q, \boldsymbol{t}_i) = s(f^{(3)}_{\mathrm{I2T}}(\boldsymbol{v}_q), f^{(3)}_{\mathrm{T2I}}(\boldsymbol{t}_i)).\end{cases} \tag{6.7}$$

Then we combine the three scores to obtain an average fusion score as follows

$$s^{avg}(\boldsymbol{v}_q, \boldsymbol{t}_i) = \frac{\sum_{j=1}^{3} s^{(j)}(\boldsymbol{v}_q, \boldsymbol{t}_i)}{3}. \tag{6.8}$$



(a) $\bullet f_{\mathrm{I2T}}(\boldsymbol{v}_i) \mid * \boldsymbol{t}_i$

(b) $\bullet \tilde{\boldsymbol{v}}_i \mid * \boldsymbol{v}_i$

(c) $\bullet f^{(3)}_{\mathrm{I2T}}(\boldsymbol{v}_i) | * f^{(3)}_{\mathrm{T2I}}(f_{\mathrm{I2T}}(\boldsymbol{v}_i))$

(d) $\bullet f_{\mathrm{T2I}}(\boldsymbol{t}_i) \mid * \boldsymbol{v}_i$

(e) $\bullet \tilde{\boldsymbol{t}}_i \mid * \boldsymbol{t}_i$

(f) $\bullet f^{(3)}_{\mathrm{T2I}}(\boldsymbol{t}_i) | * f^{(3)}_{\mathrm{I2T}}(f_{\mathrm{T2I}}(\boldsymbol{t}_i))$

Figure 6.4: Visualization of our embedding features by using 100 image-text pairs in Flickr30K [189]. The first and second rows represent the embedding features learned in the I2T2I and T2I2T branches respectively. In each feature map, matched samples are shown with the same color. In (a)(d), the dual embedding features ('•') can match with the corresponding target features ('∗'); In (b)(e), the reconstructed embedding features ('•') look closely similar to the source features ('∗'). In (c)(f), the two latent embedding features ('•' and '∗') can learn to correlate with each other as well.

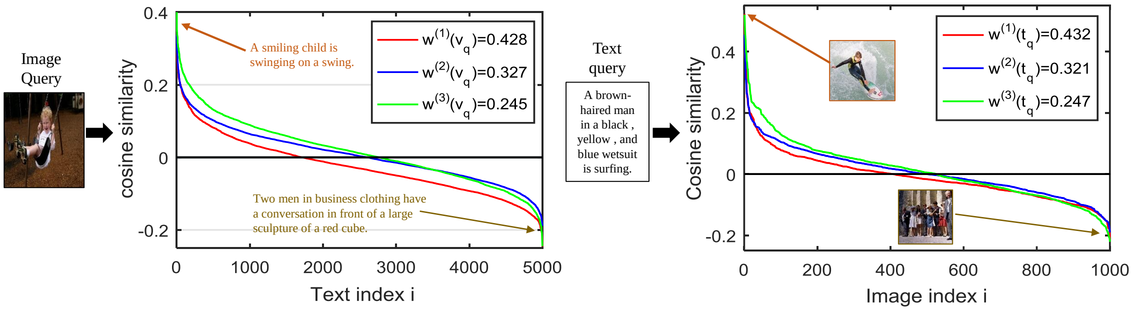


Figure 6.5: Illustration of the sorted score curves based on three different features. For the query image in left, the first curve (in red) forms the smallest area above the X axis, so the corresponding feature (*i.e.* visual embedding feature) can have the largest weight (0.428). We show a matched text at the beginning of the curves and an unmatched text at the end of the curves. Similarly, we demonstrate a text query example in right.

It is similar to compute the fusion score $s^{avg}(\boldsymbol{t}_q, \boldsymbol{v}_i)$ in terms of a query text T_q.

Adaptive fusion

To study the importance of different features, we further learn adaptive weights when combining the three scores. As suggested in [209], the score curve by using a superior feature can be sorted in an "L" shape, while the curve by using an inferior feature tends to gradually descend. In addition, the area under the curve can be used as an indicator to measure the weight of the corresponding feature. Driven by this observation, we can use the sorted score curves of the above three features to decide their weights. Specifically, we utilize each of the three features to compute the score curve of a query image I_q to all the text samples. Then, we sort the score curves and compute their areas with respect to the horizonal axis. In Figure 6.5, we show three sorted score curves for either a query image (Left) or text (Right). In contrast to [209] where the scores are in $[0, 1]$, our scores are based on the cosine distance, ranging from -1 to 1. Accordingly, we can obtain both a positive area and a negative area, locating on the two sides of the $X = 0$ axis. To alleviate the effects of long tails in the curves, we utilize only the positive area to compute the weight and omit the negative one. The positive area associated with the j-th feature can be approximated by

$$area_{+}^{(j)}(\boldsymbol{v}_q) = \sum_{i=1}^{N} \max\left[0, s^{(j)}(\boldsymbol{v}_q, \boldsymbol{t}_i)\right]. \tag{6.9}$$

Smaller positive area means that the corresponding feature should have greater weights. Hence, the adaptive weights of I_q *w.r.t.* the three features can be expressed with

$$w^{(j)}(\boldsymbol{v}_q) = \frac{1}{area_{+}^{(j)}(\boldsymbol{v}_q)}. \tag{6.10}$$

In addition, we normalize the three weights to make sure $\sum_{j=1}^{3} w^{(j)}(\boldsymbol{v}_q) = 1$. Finally, the adaptive fusion score for matching I_q and T_i becomes

$$s^{adt}(\boldsymbol{v}_q, \boldsymbol{t}_i) = \sum_{j} w^{(j)}(\boldsymbol{v}_q) \cdot s^{(j)}(\boldsymbol{v}_q, \boldsymbol{t}_i). \tag{6.11}$$

Likewise, we demonstrate a text query T_q in the right of Figure 6.5, and show its adaptive weights, $w^{(j)}(\boldsymbol{t}_q)$. Notice that our adaptive fusion approach can achieve specific weights for different query samples. It is an unsupervised and efficient manner without adding extra parameters and manual tuning. In the experiments, we analyze the effects of these two late-fusion approaches on the inference of cross-modal retrieval.

6.4 Experiments

First, we compare CycleMatch with various baseline models to verify its effectiveness. In addition, we present in-depth analysis on the two late-fusion approaches. Moreover, our results can be competitive with the state-of-the-art performance for cross-modal retrieval on two well-known datasets. Finally, we present additional ablation study on the effect of feature encoders and variance of test splits.

6.4.1 Experimental setup

We present the Dataset protocols, evaluation metrics, Network details, training details and training time, used in our experimental setup.

Dataset protocols

The experiments are performed on two well-known datasets: Flickr30K [189] and MSCOCO [117]. 1) Flickr30K [189] consists of 31,783 images and each image is associated with five different sentences. We use the dataset split of [190], namely 29,783 training images, 1,000 validation images and 1,000 test images. 2) MSCOCO [117] is one of the largest multi-modal datasets, which includes 82,783 training images and 40,504 validation images. We pick five ground-truth sentences for each image. 1,000 test images are selected from the validation set [190]. Notice that some works [53, 69, 77] merge the remaining validation images into the training set, to further increase the performance. However, we keep only using the original training set for fairness.

Evaluation metrics

For evaluating the performance of cross-modal retrieval, we adopt the common metric R@K, which measures the recall rate of a correctly retrieved ground-truth at top K retrieved candidates. Generally, K is set to 1, 5 and 10 for both image-to-text and text-to-image retrieval.

Network details

In terms of the image encoder, we employed the powerful ResNet-152 [10] pre-trained on the ImageNet dataset [5]. Besides, we recast the CNN model to its fully convolutional network (FCN) counterpart, which can capture rich region representations. The last layer of the FCN model is spatially averaged to generate a 2,048 dimensional visual representation. To extract the textual representation, we utilized the pre-trained RNN encoder proposed in [210]. It can represent one sentence with a 4,096 dimensional feature vector. Currently, we did not fine-tune the feature encoders during the training.

As for the two groups of four FC layers in CycleMatch (*i.e.* $FC_{\mathrm{I2T}}^{(j)}$ and $FC_{\mathrm{T2I}}^{(j)}$), the channels of the first three layers are fixed as [2048,512,512]. Note that, $FC_{\mathrm{I2T}}^{(4)}$ should have the same dimension as the textual feature and $FC_{\mathrm{T2I}}^{(4)}$ should be equal to the size of the visual feature.

Training details

We implemented the proposed approach based on the Caffe library [130]. It is important to shuffle the training samples randomly during the data preparation stage. The hyper-parameters are evaluated on the validation set of each dataset. We trained the model using SGD with a mini-batch size of 500, a weight decay of 0.0005, a momentum of 0.9 and an initialized learning rate of 0.1. The learning rate is divided by 10 when the decrease in loss stabilizes. We set $\alpha = 2$ and $m = 0.1$ in all the experiments. The number of negative samples in each min-batch is 50. The whole training procedure terminates after 60 epochs for both datasets.

Time complexity

We use the total loss in Eq. (6.6) to perform the training procedure. Each loss term is a simple and efficient ranking loss that is widely used in retrieval tasks. We used a Titan X card with 12 GB to train all models in the experiments. For the full CycleMatch model, training required about 19 hours on the Flickr30K dataset and 47 hours on the MSCOCO dataset, respectively.

Table 6.1: Summary of various embedding methods for image-text matching.

Embedding methods	Main description
LatentMatch	It is a latent embedding model by matching $f_{\mathrm{I2T}}^{(3)}(\boldsymbol{v}_i)$ and $f_{\mathrm{T2I}}^{(3)}(\boldsymbol{t}_i)$.
DualMatch	It is a dual embedding model with two dual mappings: I→T and T→I.
CycleMatch(w/o latent)	It is an ablation model without latent embeddings.
CycleMatch(I2T2I)	It consists of an I2T2I cycle branch and an I→T dual mapping.
CycleMatch(T2I2T)	It is composed of a T2I2T cycle branch and a T→I dual mapping.
CycleMatch	It is the fully implemented model by integrating two cycle branches.

Table 6.2: Comparison of the cross-modal retrieval results on Flickr30k and MSCOCO. Higher R@K numbers are better, where $K = 1, 5, 10$. The full CycleMatch model outperforms other baseline models on both datasets.

Method	Flickr30K dataset						MSCOCO dataset					
	Image to Text			Text to Image			Image to Text			Text to Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
LatentMatch	49.7	77.4	85.0	37.8	69.8	80.6	53.9	82.9	90.8	43.0	75.8	85.9
DualMatch	53.4	80.5	87.1	40.1	70.9	81.0	56.3	83.5	91.5	45.5	76.7	87.5
CycleMatch(w/o latent)	56.8	81.7	90.3	41.1	72.5	81.3	58.5	84.0	92.4	46.9	78.3	88.7
CycleMatch(I2T2I)	57.0	82.4	**91.0**	42.4	73.6	82.0	**61.1**	85.5	93.1	46.3	79.3	89.0
CycleMatch(T2I2T)	56.4	81.9	90.6	**43.2**	74.3	82.6	59.7	84.7	92.6	**47.6**	79.7	89.6
CycleMatch	**57.8**	**83.3**	90.9	**43.2**	**74.8**	**83.8**	60.5	**86.3**	**93.7**	47.2	**80.3**	**90.4**

6.4.2 Comparisons with baseline methods

To demonstrate the superiority of our approach, we implemented several other embedding approaches based on the same network settings and training hyper-parameters as CycleMatch. Table 6.1 describes the details regarding these methods. In terms of inference, LatentMatch is evaluated with only the latent score. However, all the other models have both visual and textual scores, therefore we utilize the average fusion approach to accomplish their inference for a fair comparison. Table 6.2 reports results of these models on both Flickr30K and MSCOCO for both image-to-text retrieval and text-to-image retrieval. It can be seen that, CycleMatch surpasses LatentMatch and DualMatch with significant improvements, and achieves overall superior performance over other variants of CycleMatch. Furthermore, we can observe the following findings:

Impact of reconstructed embeddings. The main difference between DualMatch and CycleMatch(w/o latent) is that the latter model uses a reconstructed mapping upon the traditional dual mapping. The performance improvement from CycleMatch(w/o latent) shows the benefit of learning reconstructed embeddings in a cyclic fashion.

Impact of latent embeddings. By comparing the results of CycleMatch and CycleMatch(w/o latent), we find that integrating the latent embeddings into CycleMatch brings further improvements over all R@K measurements. For example, R@5 shows about 2% gains for both I→T and T→I. Although using only latent embeddings (*i.e.* LatentMatch) is inferior to other models, it is beneficial to adopt them to improve other embedding methods like CycleMatch.

Impact of cycle branches. Both CycleMatch(I2T2I) and CycleMatch(T2I2T) can outperform LatentMatch and DualMatch, even though only one cycle-consistent embedding branch is used. By comparing these two models, CycleMatch(I2T2I) performs better for I→T retrieval, while CycleMatch(T2I2T) yields better results for T→I retrieval. When we incorporate the two cycle branches jointly to construct a full CycleMatch, it achieves overall superior performance over any single cycle branch on both datasets. It is consistent with our motivation that it is beneficial to model image-text co-translation simultaneously.

In addition to the R@K performance, we further analyze the matching scores by using our embedding features. To be specific, we randomly select 100 image-text pairs from the test set, and compute the similarity between an image and a text. As shown in Figure 6.6, matched image-text pairs (with the same index) have greater similarity scores than unmatched ones. This means that our embedding features are able to learn the correlations between visual and textual representations.

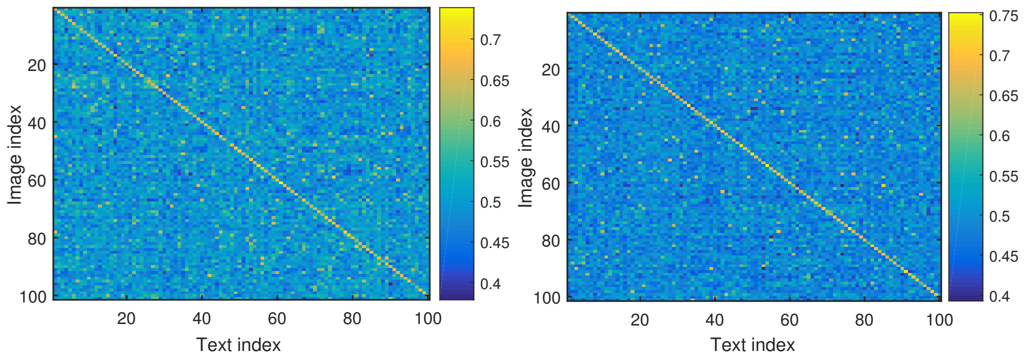
6.4.3 Analysis of late-fusion inference

Recall that CycleMatch contains visual, textual and latent scores for inference (Section 6.3.4). In this experiment, we compare three strategies to study the effect of two late-fusion inference approaches on the retrieval performance of CycleMatch. Specifically, the one-score strategy uses only a single visual score; the two-score strategy integrates visual and textual scores together; the three-score strategy combines all three scores by further adding the latent score. Table 6.3 reports the results of the three strategies. For the two-score and three-score strategies, we present the results of using the average and adaptive fusion, respectively. From the results, we can make the following observations:

1) The two-score strategy improves the one-score counterpart with 1%-3% gains. As the visual and textual scores match the samples in two different feature spaces, their complementary scores are able to improve the inference quality.

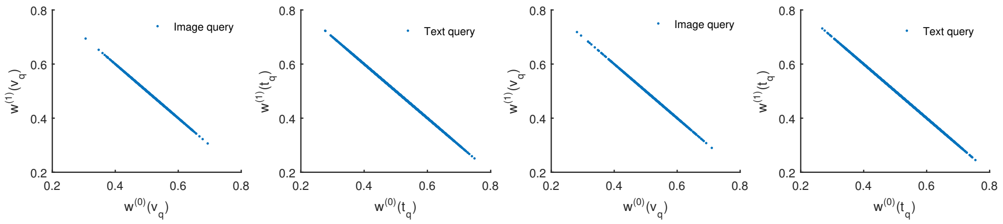
2) The adaptive fusion outperforms the average one in terms of both two-score and three-score strategies. Although their performance gap over the R@K measurements is not significant, the adaptive fusion is an efficient method without imposing extra parameters and manual tuning. In addition, the inference time of the adaptive fusion is close to that of the average fusion.

3) The three-score strategy fails to achieve further improvements over the two-score one. We attribute this to the fact that, the latent score measures the similarity between $f^{(3)}_{\mathrm{I2T}}(\boldsymbol{v}_i)$ and $f^{(3)}_{\mathrm{T2I}}(\boldsymbol{t}_i)$. However, we do not use a direct matching loss between them during training CycleMatch. Although adding this latent score for



(a) Flickr30K **(b)** MSCOCO

Figure 6.6: Similarity matrix of 100 image-text pairs from the test set. The related images and texts have the same index numbers. The diagonal line demonstrates high inter-modal correlations for matched image-text pairs. The original cosine scores are re-scaled to be [0,1].



(a) I→T **(b)** T→I **(c)** I→T **(d)** T→I

Figure 6.7: Visualization of adaptive weights for 1000 image queries and 5000 text queries on Flickr30K(a, b) and MSCOCO (c, d). Each dot in the maps is a query sample, having two weights for the adaptive fusion. Note that $w^{(0)} + w^{(1)} = 1$. The weights of query samples are mostly gathered between 0.4 and 0.6. It suggests that both visual and textual scores play an important role in the inference results.

inference will not bring further performance gains, learning the latent embeddings in CycleMatch is still important for improving the entire embedding procedure. As we discussed earlier, CycleMatch performs better than the variant without latent embeddings, namely CycleMatch(w/o latent).

As we can see, the two-score adaptive fusion achieves the best results. In Figure 6.7, we further present and analyze the two adaptive weights (*i.e.* $w^{(1)}(\cdot)$ and $w^{(2)}(\cdot)$), which are learned in the two-score adaptive fusion for visual and textual scores. Figure 6.7(a,b) and (c,d) shows the weights for Flickr30K and MSCOCO, respectively. For I2T retrieval, we illustrate the adaptive weights of 1000 image queries, namely $w^{(1)}(v_q)$ and $w^{(2)}(v_q)$; for T2I retrieval, we show all the weights of 5000 text queries, denoted as $w^{(1)}(t_q)$ and $w^{(2)}(t_q)$. Notice that, each dot in Figure 6.7 represents a query sample that learns individual weights based on its score curves. It can be seen that most samples have weights ranging from 0.4 to 0.6, which suggests that both visual and textual scores have an important impact on the inference results.

Table 6.3: Evaluation on the effect of different inference strategies on the R@K measurements. The two-score strategy based on the adaptive fusion achieves the best results (in bold face).

Inference method	Flickr30K dataset						MSCOCO dataset					
	Image to Text			Text to Image			Image to Text			Text to Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
One-score, without fusion	54.8	82.6	90.1	40.1	70.9	81.0	58.6	85.5	92.6	45.5	78.3	88.7
Two-score, average fusion	57.8	83.3	90.9	43.2	74.8	83.8	60.5	86.3	93.7	47.2	80.3	90.4
Two-score, adaptive fusion	**58.6**	83.6	**91.6**	**43.6**	**75.3**	**84.2**	**61.1**	**86.8**	94.2	**47.9**	80.9	**90.9**
Three-score, average fusion	57.4	83.5	91.0	43.2	74.7	83.9	59.7	86.0	94.0	46.9	80.6	89.8
Three-score, adaptive fusion	57.8	**83.8**	91.2	43.5	74.7	84.0	61.0	86.4	**94.5**	47.8	**81.0**	90.7

6.4.4 Comparisons with state-of-the-art approaches

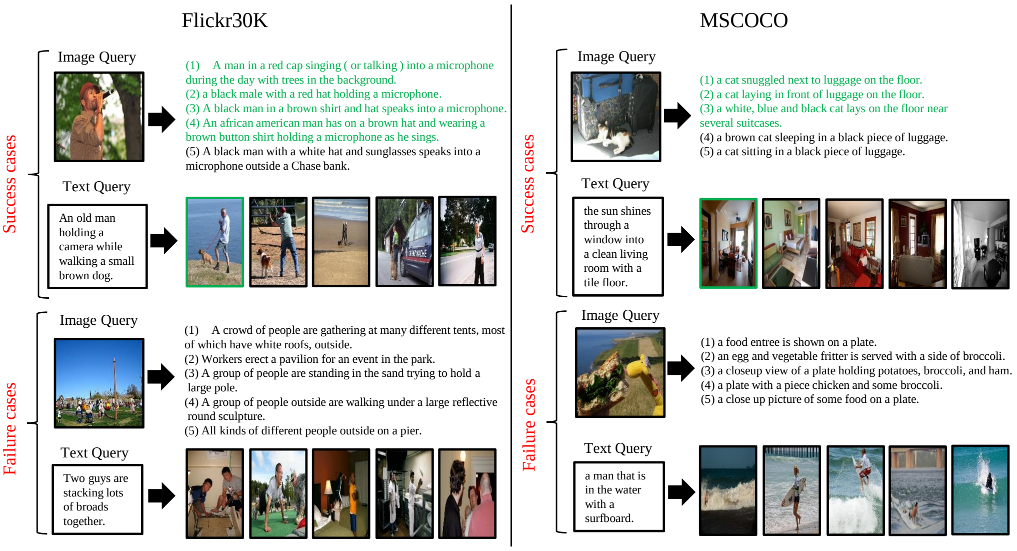
In Table 6.4 and Table 6.5, we present a comprehensive comparison with previous papers where they reported the cross-modal retrieval performance on Flickr30K and MSCOCO. It can be seen that our CycleMatch (the two-score adaptive fusion) outperforms recent state-of-the-art approaches [64, 67, 211] with promising improvements on both datasets. It is worth noting that these approaches employ different feature encoders that have a significant influence on the performance. For a clear comparison, we further list the image and text encoders used in these approaches. In the following experiments, we will study the effect of different feature encoders on the performance of CycleMatch.

To boost the performance, recent several approaches [69, 77, 200, 211] further fine-tune the image encoders during training their models. Their results with fine-tuning the image encoders achieve better performance on MSCOCO than Flickr30K. We should know that it is feasible to fine-tune the image encoders while training our CycleMatch, which can help to further improve our results. In addition, the fine-tuning process will maintain the findings we mentioned as above. More importantly, our results on the Flickr30K dataset can even compete with the fine-tuned results in [69, 77, 200, 211]. On the MSCOCO dataset, the fine-tuned approaches [69, 77, 200, 211] further merge the validation images into the training set, in order to largely increase the performance. However, we still use the original training set for a fair comparison with other prior approaches.

In addition to the quantitative evaluation, we present our image-to-text and text-to-image retrieval examples in Figure 6.8, which includes both success and failure cases. For each query sample, the top-5 candidates are retrieved, of which the ground-truth samples are highlighted in green. We notice that, the retrieved candidates are semantically related to the query sample in some extent, even for the failure cases.

Table 6.4: Comparison with the state-of-the-art approaches on Flickr30K for image-text retrieval. For the approaches without fine-tuning, we show the best results in blue color; For the ones with fine-tuning, the best results are highlighted with red color. Overall, our results with ResNet show state-of-the-art performance on this dataset.

Method	Image encoder	Text encoder	Image to Text			Text to Image		
			R@1	R@5	R@10	R@1	R@5	R@10
Without fine-tuning image encoders								
DCCA [49]	AlexNet	TF-IDF	16.7	39.3	52.9	12.6	31.0	43
DVSA [55]	AlexNet	RNN	22.2	48.2	61.4	15.2	37.7	50.5
UVSE [66]	VGG-19	RNN	23.0	50.7	62.9	16.8	42.0	56.5
mCNN [52]	VGG-19	CNN	33.6	64.1	74.9	26.2	56.3	69.6
VQA-aware [191]	VGG-19	RNN	33.9	62.5	74.5	24.9	52.6	64.8
GMM-FV [51]	VGG-16	GMM+HGLMM	35.0	62.0	73.8	25.0	52.7	66.0
m-RNN [190]	VGG-16	RNN	35.4	63.8	73.7	22.8	50.7	63.1
RNN-FV [185]	VGG-19	RNN	35.6	62.5	74.2	27.4	55.9	70.0
HM-LSTM [65]	AlexNet	RNN	38.1	-	76.5	27.7	-	68.8
DSPE [53]	VGG-19	HGLMM	40.3	68.9	79.9	29.7	60.1	72.1
sm-LSTM [68]	VGG-19	RNN	42.5	71.9	81.5	30.2	60.4	72.3
VSE++ [211]	ResNet-152	RNN	43.7	-	82.1	32.2	-	72.1
DualCNN [69]	ResNet-152	ResNet-152	44.2	70.2	79.7	30.7	59.2	70.8
RRF-Net [64]	ResNet-152	HGLMM	47.6	77.4	87.1	35.4	68.3	79.9
2WayNet [76]	VGG-16	GMM+HGLMM	49.8	67.5	-	36.0	55.6	-
DAN [67]	ResNet-152	RNN	55.0	81.8	89.0	39.4	69.2	79.1
CycleMatch (Ours)	VGG-19	RNN	51.4	80.6	88.1	38.5	71.0	81.3
CycleMatch (Ours)	ResNet-152	RNN	58.6	83.6	91.6	43.6	75.3	84.2
With fine-tuning image encoders								
DualCNN [69]	ft ResNet-152	ResNet-152	55.6	81.9	89.5	39.1	69.2	80.9
VSE++ [211]	ft ResNet-152	RNN	52.9	-	87.2	39.6	-	79.5
cnp + ctx + gen [200]	ResNet-152, ft VGG-19	RNN	55.5	82.0	89.3	41.1	70.5	80.1



(1) A man in a red cap singing (or talking) into a microphone during the day with trees in the background.
(2) a black male with a red hat holding a microphone.
(3) A black man in a brown shirt and hat speaks into a microphone.
(4) An african american man has on a brown hat and wearing a brown button shirt holding a microphone as he sings.
(5) A black man with a white hat and sunglasses speaks into a microphone outside a Chase bank.

An old man holding a camera while walking a small brown dog.

(1) A crowd of people are gathering at many different tents, most of which have white roofs, outside.
(2) Workers erect a pavilion for an event in the park.
(3) A group of people are standing in the sand trying to hold a large pole.
(4) A group of people outside are walking under a large reflective round sculpture.
(5) All kinds of different people outside on a pier.

Two guys are stacking lots of broads together.

(1) a cat snuggled next to luggage on the floor.
(2) a cat laying in front of luggage on the floor.
(3) a white, blue and black cat lays on the floor near several suitcases.
(4) a brown cat sleeping in a black piece of luggage.
(5) a cat sitting in a black piece of luggage.

the sun shines through a window into a clean living room with a tile floor.

(1) a food entree is shown on a plate.
(2) an egg and vegetable fritter is served with a side of broccoli.
(3) a closeup view of a plate holding potatoes, broccoli, and ham.
(4) a plate with a piece chicken and some broccoli.
(5) a close up picture of some food on a plate.

a man that is in the water with a surfboard.

Figure 6.8: Qualitative results of our CycleMatch on Flickr30K and MSCOCO. Given one query, the top-5 candidates are retrieved. In the success cases, the correct matches are highlighted with green. In the failure cases, our method can still retrieve some reasonable false candidates related to the query.

Table 6.5: Comparison with the state-of-the-art approaches on MSCOCO for image-text retrieval. For the approaches without fine-tuning, the best results are highlighted with blue color; For the ones with fine-tuning, we show the best results in red color. Among the approaches without fine-tuning the image encoders, our approach with ResNet can achieve the state-of-the-art performance.

Method	Image encoder	Text encoder	Image to Text			Text to Image		
			R@1	R@5	R@10	R@1	R@5	R@10
Without fine-tuning image encoders								
STV [212]	VGG-19	RNN	33.8	67.7	82.1	25.9	60.0	74.6
DVSA [55]	AlexNet	RNN	38.4	69.9	80.5	27.4	60.2	74.8
GMM-FV [51]	VGG-16	GMM+HGLMM	39.4	67.9	80.9	25.1	59.8	76.6
m-RNN [190]	VGG-16	RNN	41.0	73.0	83.5	29.0	42.2	77.0
RNN-FV [185]	VGG-19	RNN	41.5	72.0	82.9	29.2	64.7	80.4
BiLSTM-Max [210]	ResNet-101	RNN	42.6	75.3	87.3	33.9	69.7	83.8
mCNN [52]	VGG-19	CNN	42.8	73.1	84.1	32.6	68.6	82.8
UVSE [66]	VGG-19	RNN	43.4	75.7	85.8	31.0	66.7	79.9
HM-LSTM [65]	AlexNet	RNN	43.9	-	87.8	36.1	-	86.7
order-embeddings [213]	VGG-19	RNN	46.7	-	88.9	37.9	-	85.9
DSPE [53]	VGG-19	HGLMM	50.1	79.7	89.2	39.6	75.2	86.9
VQA-aware [191]	VGG-19	RNN	50.5	80.1	89.7	37.0	70.9	82.9
DualCNN [69]	ResNet-50	ResNet-50	52.2	80.4	88.7	37.2	69.5	80.6
sm-LSTM [68]	VGG-19	RNN	53.2	83.1	91.5	40.7	75.8	87.4
2WayNet [76]	VGG-16	GMM+HGLMM	55.8	75.2	-	39.7	63.3	-
RRF-Net [64]	ResNet-152	HGLMM	56.4	85.3	91.5	43.9	78.1	88.6
VSE++ [211]	ResNet-152	RNN	58.3	-	93.3	43.6	-	87.8
CycleMatch (Ours)	VGG-19	RNN	55.1	83.5	91.3	43.7	76.7	88.4
CycleMatch (Ours)	ResNet-152	RNN	61.1	86.8	94.2	47.9	80.9	90.9
With fine-tuning image encoders								
DualCNN [69]	ft ResNet-50	ResNet-50	65.6	89.8	95.5	47.1	79.9	90.0
VSE++ [211]	ft ResNet-152	RNN	64.6	-	95.7	52.0	-	92.0
Gen-XRN [77]	ft ResNet-152	RNN	68.5	-	97.9	56.6	-	94.5
cnp + ctx + gen [200]	ResNet-152, ft VGG-19	RNN	69.9	92.9	97.5	56.7	87.5	94.8

6.4.5 Effect of feature encoders

As shown in Figure 6.3, we extract visual and textual features from off-the-shelf feature encoders. The proposed CycleMatch can be compatible with diverse feature encoders, but it is still encouraged to study the effect of different feature encoders on the performance. We report the results in Table 6.6.

Considering the image encoders, we use the VGG-19 and ResNet-152 models to extract the visual features and compare their results. We can see that, ResNet-152 has a considerable improvements over VGG-19 on all measurements, especially for R@1 accuracies. This shows the benefit of using more powerful CNN models for improving the visual embeddings. In addition, the feature dimension with ResNet-152 (*i.e.* 2,048) is lower than that with VGG-19 (*i.e.* 4,096). Therefore, in this work we take the ResNet-152 model as the preferable image encoder.

In terms of the text encoders, we test another two encoders apart from the RNN encoder. The first one is word2vec [188], which describes each word in the sentence with a 300-dimensional feature vector. We then compute the average of all the word features to represent the sentence feature. The second one is an expensive representation based on the Hybrid Gaussian-Laplacian mixture model (HGLMM) [51].

Table 6.6: Evaluation on the effect of different feature encoders on the performance of CycleMatch. By comparison, ResNet-152 is a superior image encoder and RNN is a more powerful text encoder.

Image encoder	Text encoder	Flickr30K						MSCOCO					
		Image to Text			Text to Image			Image to Text			Text to Image		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Effect of image encoders													
VGG-19	RNN	51.4	80.6	88.1	38.5	71.0	81.3	55.1	83.5	91.3	43.7	76.7	88.4
ResNet-152	RNN	58.6	83.6	91.6	43.6	75.3	84.2	61.1	86.8	94.2	47.9	80.9	90.9
Effect of text encoders													
ResNet-152	word2vec	48.1	78.7	87.4	37.7	70.8	81.1	55.9	83.8	91.8	44.7	79.1	87.7
ResNet-152	HGLMM	54.5	81.6	90.9	41.3	73.1	82.8	58.4	85.5	93.4	46.2	80.3	89.4
ResNet-152	RNN	58.6	83.6	91.6	43.6	75.3	84.2	61.1	86.8	94.2	47.9	80.9	90.9

Specifically, HGLMM computes a 18,000-dimension feature vector with 30 centers (*i.e.* 300*30*2). Similar to [53], we further reduce it to a 6,000-dimension feature vector in order to decrease the training complexity. As shown in Table 6.6, the RNN encoder is more powerful than both word2vec and HGLMM. In addition, the feature dimension based RNN (*i.e.* 4,096) is feasible and practical during training CycleMatch.

6.5 Chapter Conclusions

In this chapter, we have developed a novel embedding method for the multi-modal task of matching visual and textual representations. We proposed cycle-consistent embeddings to learn both intra-modal correlations and intra-modal consistency. Our approach taking advantage of multiple embedding techniques is able to outperform any single embedding method. The experimental results have demonstrated the superiority of our method over other embedding methods. In addition, we have presented two simple and efficient late-fusion approaches to increase the inference quality. The late-fusion inference can integrate different matching scores together without increasing the training complexity. Finally, our approach has shown state-of-the-art performance for cross-modal retrieval on Flickr30K and MSCOCO.

Future work. we will take into account local relations when matching images and sentences, for example, semantic correlations between visual regions and phases. One potential solution is to exploit the attention mechanism to localize the objects corresponding to the phase description.