Wasserstein Coupled Graph Learning for Cross-Modal Retrieval

Yun Wang[1]*, Tong Zhang[1]*, Xueya Zhang[1], Zhen Cui[1]†, Yuge Huang[2],
Pengcheng Shen[2], Shaoxin Li[2], Jian Yang[1]
[1]PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional
Information of Ministry of Education, School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China.
[2] Youtu Lab, Tencent.
{yun.wang, tong.zhang}@njust.edu.cn, xyzhang678@163.com, zhen.cui@njust.edu.cn,
{yugehuang, quantshen, darwinli}@tencent.com, csjyang@njust.edu.cn

Abstract

Graphs play an important role in cross-modal image-text understanding as they characterize the intrinsic structure which is robust and crucial for the measurement of cross-modal similarity. In this work, we propose a Wasserstein Coupled Graph Learning (WCGL) method to deal with the cross-modal retrieval task. First, graphs are constructed according to two input cross-modal samples separately, and passed through the corresponding graph encoders to extract robust features. Then, a Wasserstein coupled dictionary, containing multiple pairs of counterpart graph keys with each key corresponding to one modality, is constructed for further feature learning. Based on this dictionary, the input graphs can be transformed into the dictionary space to facilitate the similarity measurement through a Wasserstein Graph Embedding (WGE) process. The WGE could capture the graph correlation between the input and each corresponding key through optimal transport, and hence well characterize the inter-graph structural relationship. To further achieve discriminant graph learning, we specifically define a Wasserstein discriminant loss on the coupled graph keys to make the intra-class (counterpart) keys more compact and inter-class (non-counterpart) keys more dispersed, which further promotes the final cross-modal retrieval task. Experimental results demonstrate the effectiveness and state-of-the-art performance.

1. Introduction

Cross-modal retrieval is a long-standing challenge in multimedia, and has drawn increasing attention due to its wide applications which bring great convenience to people's daily life. For instance, someone may need to search for some images that match a piece of text. However, it would be rather labor-consuming and tedious to manually browse those massive images on the Internet. For this issue, the cross-modal (e.g. across vision and text) retrieval technique would effectively accomplish the search of those specific images meeting the criteria, and hence significantly improve the work efficiency. Therefore, it's quite necessary and meaningful to investigate the cross-modal retrieval.

This work mainly focuses on the retrieval across vision (videos or images) and text. Currently, some related tasks have been investigated such as video description [3] and video/image query and answer (Q & A) [44]. For these tasks, some methods focused on either learning the appearance features from images or capturing dynamics among sequential frames/texts first, then bridging images/videos and texts by measuring their representation similarity. In this process, sophisticated models with powerful feature learning ability were employed, including ResNet [13], gated recurrent unit (GRU) [4], and long-short term memory (LSTM) [15]. Although notable progress is achieved, however, the intrinsic structure of each modal, such as the interaction between entities in images, is not well exploited.

To facilitate the structure modeling in cross-modal tasks, datasets such as the Real-world Scene Graph dataset [20] and Moviegraphs [46] were built with annotated graphs of both vision (videos/images) and text provided. Specifically, a Graph Wasserstein Correlation analysis (GWCA) [53] method was proposed on Moviegraphs dataset by integrating the graph signal filtering with metric learning in Wasserstein space (W-space). Although considerable success was achieved based on the Wasserstein metric (W-metric), there were still some critical issues to be tackled. On one hand, GWCA didn't achieve discriminant learning on graphs as it only considered the intra-class compactness between the

*Yun Wang and Tong Zhang have equal contribution.

†The corresponding author.

matched pair of cross-modal graphs. However, for the cross-modal heterogeneous graphs, considering their inter-class dispersion is also crucial for the similarity measurement; on the other hand, GWCA had a shallow architecture that limited the feature learning ability to handle the graph diversity of different modalities. And it could be hardly extended to a deep architecture due to the involved computation-consuming singular value decomposition (SVD). Besides, to achieve discriminant deep learning on graphs in W-space is rather challenging, which quite differs from the conventional Wasserstein analysis method [9]. Generally, two critical problems need to be solved: i) High complexity exists in computing inter-class and intra-class scatters, e.g. the explicit calculation on mean and covariance of graphs are rather difficult in W-metric space; ii) The minibatch-based processing of deep learning architectures makes it difficult to access the global covariance of all samples within one class or across different ones.

To tackle all the issues aforementioned, we propose a deep graph neural network framework named Wasserstein Coupled Graph Learning (WCGL) for the cross-modal retrieval task. For the input graphs from two modalities, a coupled graph dictionary is constructed as the reference set to learn discriminative representation while avoiding explicit statistic computation on the mean and covariance of graph samples. In the coupled dictionary, two parts of keys are contained where each key in one part is a graph corresponding to one modality while has a counterpart key in the other part. With the coupled graph dictionary, the input graphs can be transformed into the dictionary space through a proposed Wasserstein graph embedding (WGE) process by calculating the graph correlation with respect to the corresponding graph keys. During this process, the coupled dictionary serves as a bridge to transform the cross-modal graphs into the succinct Euclidean subspace. Specifically, the graph correlation is measured in W-space through the regularized W-metric with optimal transport (OT) matrices. Based on the W-metric, the Wasserstein graph embedding would be advantageous in characterizing the graph structural information, and hence achieves better graph feature learning. To learn more discriminative features, the coupled graph dictionary is designed to be dynamically updated during training. Specifically, in the constrain of a maximum Wasserstein discriminant loss (WD-loss), i.e. ratio of inter-class (non-counterpart keys) versus intra-class (counterpart keys) W-distance, the encoded dictionary keys are optimized to preserve intra-class compactness and inter-class dispersion, which would better facilitate the cross-modal similarity measurement. Finally, we construct a fully end-to-end training network which contains the graph encoding, Wasserstein graph embedding, and discriminant graph learning. We test the WCGL on four cross-modal retrieval datasets, including Real-world Scene Graphs [20], Flickr30K [39], MSCOCO [28], Moviegraphs [46], and the experimental results demonstrate its effectiveness.

To summarize, the contributions are three-fold:

- We propose a novel Wasserstein Coupled Graph Learning framework to alleviate the diversity of cross-modal data. In this framework, a coupled graph dictionary is introduced to project cross-modal data into a common dictionary space, where the dictionary learning is performed through W-metric. To the best of our knowledge, this is the first work that performs the coupled graph dictionary learning on graphs in W-space, and uses it to deal with the cross-modal retrieval task.
- We propose the discriminant learning on both graphs and the coupled dictionary to learn better representation ability, where the WD-loss in Wasserstein space is specifically imposed on counterpart/non-counterpart keys in the coupled dictionary.
- We verify the effectiveness of our method and report the state-of-the-art results on the Real-world Scene Graph [20], Flickr30K [39], MSCOCO [28] and Moviegraphs [46] datasets.

2. Related Work

In this section, we first review those previous works about cross-modal understanding, then introduce works related to graph learning and coupled dictionary learning.

Cross-modal Understanding. Various relevant works have been proposed to deal with cross-modal learning tasks, and we mainly review them across videos [7, 31, 36] or images [26, 27, 17] and text. For those works about video-text retrieval, Xu [50] obtained a sentence-level vector and a video vector by aggregating vectorized subject-verb-object triplets and mean pooling over frame-level features, respectively, and then projected them into a joint space. Yu [52] utilized LSTM to encode video and text representation, and then applied a bilinear layer to explore their interactions. Recently, Vicol et al [46] proposed a new Moviegraphs dataset for retrieving videos and texts with graphs, which also shows the effectiveness of graphs to describe the important structural information in videos and texts. For the image-text understanding, Chen [3] performed the sentence generation and image retrieval to find the bi-directional mapping between images and their textual descriptions. Li [27] introduced a reasoning model to generate a visual representation which captures main objects and semantic of a scene.

Graph Learning. In recent years, graph neural networks (GNNs) are given more attention in the field of artificial intelligence [34, 16, 19]. Specifically, graph convolutional neural networks (GCNNs) aim to model non-gridded graph data that are flexible in structures. Inspired

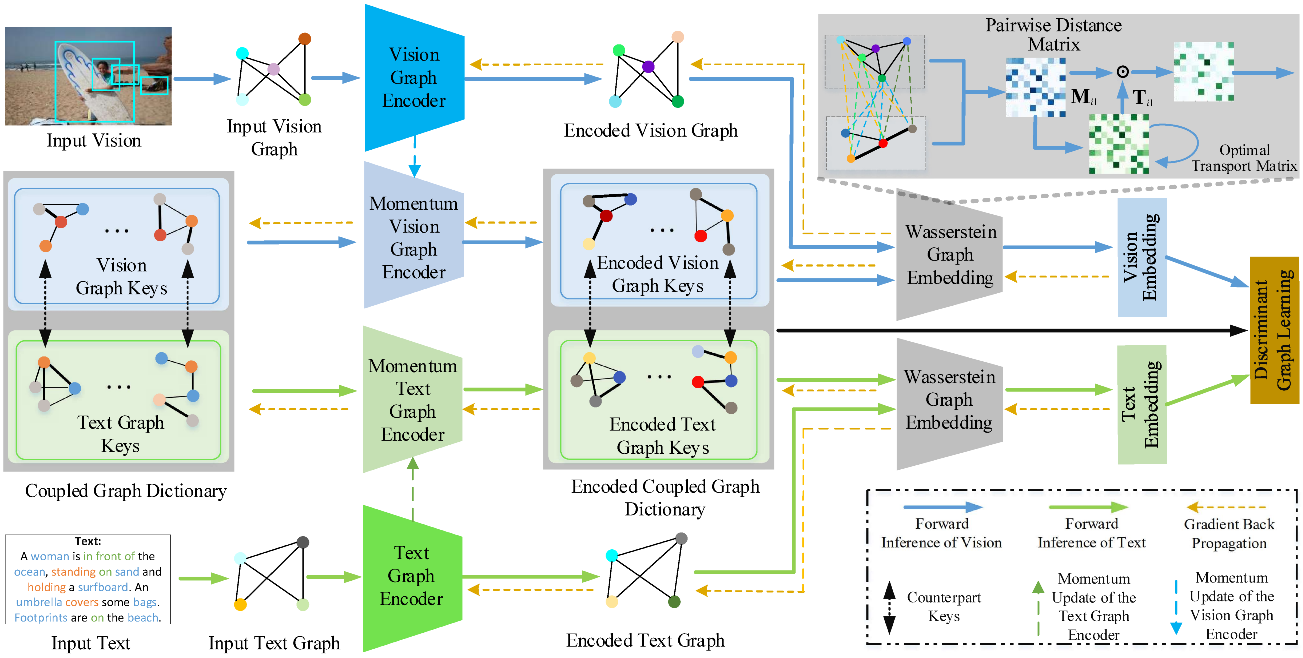

Input Vision
Input Vision Graph
Vision Graph Encoder
Encoded Vision Graph
Pairwise Distance Matrix
$\mathbf{M}_{il}$
$\odot$
$\mathbf{T}_{il}$
Optimal Transport Matrix
Vision Graph Keys
Text Graph Keys
Coupled Graph Dictionary
Momentum Vision Graph Encoder
Momentum Text Graph Encoder
Encoded Vision Graph Keys
Encoded Text Graph Keys
Encoded Coupled Graph Dictionary
Wasserstein Graph Embedding
Wasserstein Graph Embedding
Vision Embedding
Text Embedding
Discriminant Graph Learning
Text:
A woman is in front of the ocean, standing on sand and holding a surfboard. An umbrella covers some bags. Footprints are on the beach.
Input Text
Input Text Graph
Text Graph Encoder
Encoded Text Graph
Forward Inference of Vision
Forward Inference of Text
Gradient Back Propagation
Counterpart Keys
Momentum Update of the Text Graph Encoder
Momentum Update of the Vision Graph Encoder

Figure 1. The architecture of our WCGL. We take the retrieval task across image and text as an example. Three main learning processes are involved in the WCGL framework: graph encoding, Wasserstein graph embedding, and discriminant graph learning. Given two cross-modal input samples, i.e. an image and a text sample, two corresponding graphs are first generated. Also, a coupled dictionary containing two parts of counterpart graph keys is constructed. For these input graphs and graph keys, corresponding graph encoders are constructed to extract robust feature representation. More details could be found in Section 3.2. Based on the coupled dictionary, Wasserstein graph embedding is performed to transform the input graphs into the dictionary space (Section 3.3), where W-metric is employed to measure the correlation between the input graphs and the corresponding graph keys. Moreover, the discriminant graph learning is performed to preserve the intra-class (counterpart keys) compactness and inter-class (non-counterpart keys) dispersion of the coupled dictionary during training. Finally, the coupled graph dictionary is dynamically optimized together with graph encoders. More details can be found in the main body.

by the success of the standard CNN [24], various graph convolution operations have been explored, yielding multiple graph CNN variants [37, 6], including graph convolutional network (GCN) [21], PATCHY-SAN (referred to as PSCN) [37], Diffusion-convolutional neural networks (DCNN) [1], NgramCNN [33], and GWCA [53].

Coupled Dictionary Learning. In 2012, Yang et al. [51] proposed a coupled dictionary training method for single-image super-resolution based on patchwise sparse recovery. Then, various coupled dictionary learning (CDL) methods were proposed [18] in many fields, including the semi-CDL method [49] to conduct photo-sketch synthesis, the Semi-Supervised CDL method for Person Re-identification [32], and the CDL learning with Large-Margin Structure Inference [54] for Search-Based Depth Estimation. Besides, a generalized CDL method [35] was proposed and applied to the cross-modal matching task.

Our method differs from all these methods above from two main aspects: (1) Our WCGL framework conducts graph learning with a coupled graph dictionary in W-space, where each key in the dictionary is a graph rather than a point vector; (2) Our WCGL conducts discriminant learning based on the coupled graph keys by defining a WD-loss.

3. The Proposed Method

In this section, we will first give an overview of the proposed WCGL model, then describe the learning processes in the proposed framework in detail.

3.1. Overview

Fig. 1 shows the whole architecture of our WCGL framework. Generally, the proposed model consists of three main learning processes, i.e. graph encoding, Wasserstein graph embedding, and discriminant graph learning. For the cross-modal retrieval task, the inputs of the WCGL model are two graphs constructed from samples of different modalities. Then, the WCGL model aims to predict the similarity between these two input heterogeneous graphs. Considering the inter-modal diversity, a coupled graph dictionary is constructed to encode those input graphs into more succinct vectors through the WGE. This dictionary contains two parts of coupled graph keys where each part is used to model the input samples of one certain modality. Based on the WGE, the input graphs of different modalities are both transformed into the dictionary space to facilitate the similarity measurement. Specifically, in this process, the graph correlation between one input graph and the corresponding graph keys is calculated through W-metric, which well captures the graph structural information. In the training process, the coupled graph keys will be dynamically updated. To further learn discriminative features, the discriminant graph learning is proposed by specifically defining a maximum WD-loss which maximizes the ratio of inter-class versus intra-class W-distance. Finally, the whole architecture could be optimized in an end-to-end training model.

3.2. Graph Encoding

For given graphs, we use the graph encoder to learn robust features from them. To construct the graph encoder, we employ the effective and widely used GCNs [21], and stack them to obtain the powerful ability of learning graph topology. Formally, taking $f_{\mathrm{g}}(\cdot)$ of two-layer GCNs as an example, the graph encoding for the input graph denoted as $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$ can be written as:

$$f_{\mathrm{g}}(\mathbf{X}, \mathbf{A}, \Phi) = \sigma(\widehat{\mathbf{A}}ReLU(\widehat{\mathbf{A}}\mathbf{X}\mathbf{W}_0)\mathbf{W}_1). \tag{1}$$

In this equation, $\mathcal{V} \in \{1 \cdots, N\}$, $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote the node set, the adjacent matrix and the feature matrix, respectively. $\sigma(\cdot)$ is a non-linear activation function such as the softmax. N, d are the node number and the feature dimensionality, respectively. Φ denotes the parameter set and $\mathbf{W}_0, \mathbf{W}_1 \in \Phi$ are two weighting matrices for feature projection. Specifically, a little different from the standard GCN, here $\widehat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. According to [42], from the view of spectral filtering, similar learning effects can be achieved no matter whether Laplacian normalization is performed because the adjacent matrix and its Laplacian norm have the same eigenvectors. Besides, for simplification, we omit the bias variables in Eqn. (1).

In WCGL, four different kinds of graphs need to be encoded, including the input vision graph $\mathcal{G}_i^{\mathrm{v}} = (\mathcal{V}_i^{\mathrm{v}}, \mathbf{A}_i^{\mathrm{v}}, \mathbf{X}_i^{\mathrm{v}})$, the input text graph $\mathcal{G}_i^{\mathrm{t}} = (\mathcal{V}_i^{\mathrm{t}}, \mathbf{A}_i^{\mathrm{t}}, \mathbf{X}_i^{\mathrm{t}})$, the vision graph key $\mathcal{G}_j^{\mathrm{D,v}} = (\mathcal{V}_j^{\mathrm{D,v}}, \mathbf{A}_j^{\mathrm{D,v}}, \mathbf{X}_j^{\mathrm{D,v}})$, and the text graph key $\mathcal{G}_j^{\mathrm{D,t}} = (\mathcal{V}_j^{\mathrm{D,t}}, \mathbf{A}_j^{\mathrm{D,t}}, \mathbf{X}_j^{\mathrm{D,t}})$, respectively, and i, j denote the indexes of the graphs. The corresponding graph encoders of all graphs share the same structure defined in Eqn. (1), but use different parameters correspondingly. Taking the input vision graph $\mathcal{G}_i^{\mathrm{v}}$ as an example, with the parameter Φ^{v}, the encoded feature denoted as $\mathbf{F}_i^{\mathrm{v}}$ takes the following form:

$$\mathbf{F}_i^{\mathrm{v}} = f_{\mathrm{g}}(\mathbf{X}_i^{\mathrm{v}}, \mathbf{A}_i^{\mathrm{v}}, \Phi^{\mathrm{v}}) = \sigma(\widehat{\mathbf{A}}_i^{\mathrm{v}}ReLU(\widehat{\mathbf{A}}_i^{\mathrm{v}}\mathbf{X}_i^{\mathrm{v}}\mathbf{W}_0^{\mathrm{v}})\mathbf{W}_1^{\mathrm{v}}). \tag{2}$$

After graph encoding, the corresponding graph representation can be obtained, such as $\mathbf{F}_i^{\mathrm{v}} \in \mathbb{R}^{N^{\mathrm{v}} \times d'}, \mathbf{F}_i^{\mathrm{t}} \in \mathbb{R}^{N^{\mathrm{t}} \times d'}$ corresponding to the input graphs $\mathcal{G}_i^{\mathrm{v}}, \mathcal{G}_i^{\mathrm{t}}$. Here, $N^{\mathrm{v}}, N^{\mathrm{t}}$ are the node numbers of the input vision and text graphs. The feature dimensionality after encoding is denoted as d' and set to the same for all encoded graphs for simplification. For the coupled dictionary containing two parts of keys, assuming each part has K keys, then two corresponding feature sets of encoded keys can be obtained, denoted as $\mathcal{S}_{\mathrm{v}} = \{\mathbf{F}_1^{\mathrm{D,v}}, \mathbf{F}_2^{\mathrm{D,v}}, \cdots, \mathbf{F}_K^{\mathrm{D,v}}\}$ ($\mathbf{F}_j^{\mathrm{D,v}} \in \mathbb{R}^{N^{\mathrm{D,v}} \times d'}$) for the vision and $\mathcal{S}_{\mathrm{t}} = \{\mathbf{F}_1^{\mathrm{D,t}}, \mathbf{F}_2^{\mathrm{D,t}}, \cdots, \mathbf{F}_K^{\mathrm{D,t}}\}$ ($\mathbf{F}_j^{\mathrm{D,t}} \in \mathbb{R}^{N^{\mathrm{D,t}} \times d'}$) for the text. $N^{\mathrm{D,v}}, N^{\mathrm{D,t}}$ are the node numbers.

3.3. Wasserstein Graph Embedding

The WGE aims to project one input graph into the dictionary space by exploring the correlation between the input graph and its corresponding keys. As the input and keys are all graphs, which do not lie in the Euclidean space, the common metric such as cosine similarity cannot be used to measure their correlation directly. For this issue, we employ the W-metric to learn the graph correlation in the W-space [19]. For one input graph, e.g. $\mathbf{F}_i^{\mathrm{v}}$, the correlation is calculated between it and all the keys in $\mathcal{S}_{\mathrm{v}} = \{\mathbf{F}_1^{\mathrm{D,v}}, \mathbf{F}_2^{\mathrm{D,v}}, \cdots, \mathbf{F}_K^{\mathrm{D,v}}\}$. Then, a K-dimensional feature vector $\mathbf{z}_i^{\mathrm{v}} = [z_{i1}^{\mathrm{v}}, \cdots, z_{iK}^{\mathrm{v}}]$ can be obtained for $\mathbf{F}_i^{\mathrm{v}}$. Specifically, the W-distance between two graphs, e.g. $\mathbf{F}_i^{\mathrm{v}}$ and $\mathbf{F}_j^{\mathrm{D,v}}$, can be written as:

$$z_{ij}^{\mathrm{v}} = W_\lambda(\mathbf{F}_i^{\mathrm{v}}, \mathbf{F}_j^{\mathrm{D,v}}) = \langle \mathbf{T}_{ij}^{\lambda}, \mathbf{M}_{ij} \rangle, \tag{3}$$

$$\text{s.t.}\ , \mathbf{T}_{ij}\mathbf{1}_{N^{\mathrm{D,v}}} = \mathbf{1}_{N^{\mathrm{v}}}/N^{\mathrm{v}}, \mathbf{T}_{ij}^T\mathbf{1}_{N^{\mathrm{v}}} = \mathbf{1}_{N^{\mathrm{D,v}}}/N^{\mathrm{D,v}}, \tag{4}$$

where $\mathbf{T}_{ij} \in \mathbb{R}_+^{N^{\mathrm{v}} \times N^{\mathrm{D,v}}}$ and $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}^T\mathbf{B})$. In Eqn. (3), $\mathbf{M}_{ij}$ is the pairwise distance matrix in Euclidean space, and the element $M_{ij}(r, l)$ in the r-th row and l-th column calculates the squared Euclidean distance between the r-th node of $\mathbf{F}_i^{\mathrm{v}}$ and l-th node of $\mathbf{F}_j^{\mathrm{D,v}}$. $\mathbf{T}_{ij}^{\lambda}$ is the solution of an entropy-smoothed optimal transport problem:

$$\mathbf{T}_{ij}^{\lambda} = \underset{\mathbf{T}_{ij}}{\arg\min} \quad \lambda\langle \mathbf{T}_{ij}, \mathbf{M}_{\mathbf{F}_i^{\mathrm{v}}, \mathbf{F}_j^{\mathrm{D,v}}} \rangle - \Omega(\mathbf{T}_{ij}). \tag{5}$$

Here, $\Omega(\mathbf{T}_{ij}) = -\sum_{rl} T_{ij}(r, l)\log(T_{ij}(r, l))$ where $T_{ij}(r, l)$ is the element in the r-th row and l-th column of $\mathbf{T}_{ij}$. $\Omega(\mathbf{T}_{ij})$ can be seen as a discrete joint probability distribution calculating the entropy of $\mathbf{T}_{ij}$. λ is a regularization coefficient controlling the local information involved in distance between cross-graph points. Specifically, the optimization in Eqn. (5) can be efficiently solved by Sinkhorn's fixed point iterations [5], and the solution can be written as:

$$\begin{aligned} \mathbf{T}_{ij} &= \mathrm{diag}(\mathbf{u}_{ij})\mathbf{K}_{ij}\mathrm{diag}(\mathbf{v}_{ij}) \\ &= \mathbf{u}_{ij}\mathbf{1}_{N^{\mathrm{D,v}}}^T \odot \mathbf{K}_{ij} \odot \mathbf{1}_{N^{\mathrm{v}}}\mathbf{v}_{ij}^T, \end{aligned} \tag{6}$$

where $\odot$ represents element-wise production, and $\mathbf{K}_{ij}$ is calculated based on the distance matrix $\mathbf{M}_{ij}$ with $\mathbf{K}_{ij} = e^{-\lambda\mathbf{M}_{ij}}$. diag$(\cdot)$ transforms a vector to a diagonal matrix. In Sinkhorn iterations, $\mathbf{u}_{ij}$ and $\mathbf{v}_{ij}$ are updated. Taking the k-th iteration as an example, it can be formulated as:

$$\mathbf{v}_{ij}^k = \frac{\mathbf{1}_{N^{\mathrm{D,v}}}/N^{\mathrm{D,v}}}{\mathbf{K}_{ij}^T\mathbf{u}_{ij}^{k-1}}, \mathbf{u}_{ij}^k = \frac{\mathbf{1}_{N^{\mathrm{v}}}/N^{\mathrm{v}}}{\mathbf{K}_{ij}\mathbf{v}_{ij}^k}. \tag{7}$$

For the initialization of the update process above, $\mathbf{u}_{ij}^{0}$ is assigned as an all-1 vector $\mathbf{1}_{N^{\mathrm{v}}}$.

Similar with the calculation of $\mathbf{z}_i^{\mathrm{v}}$, the feature $\mathbf{z}_i^{\mathrm{t}}$ for the i-th input text graph $\mathbf{F}_i^{\mathrm{t}}$ can be obtained by following the corresponding WGE process above.

3.4. Discriminant Graph Learning

To effectively promote the cross-modal retrieval task, our WCGL model aims to possess two expected properties, i.e. the discriminability of the learned features and the discriminant learning on encoded dictionary keys. To learn discriminative features, a supervised learning objective is adopted to guide the network optimization. For the discriminant learning on encoded dictionary keys, the distribution of encoded keys is expected to be compact within each pair of counterpart keys while dispersed across non-counterpart pairs.

For the purposes above, we propose two parts of losses to guide the whole architecture's optimization cooperatively, including the triplet loss E_s and WD-loss E_w. Then, the total loss of the whole architecture can be formulated as:

$$E = E_s - \beta E_w, \tag{8}$$

where β is a trade-off parameter between E_s and E_w.

Graph Discriminant Learning. For the input pairs of cross-modal samples, both positive (matched) and negative (mis-matched) pairs are sampled at each mini-batch. Then, E_s is formulated as following:

$$\begin{aligned} E_s(\mathbf{z}_i^{\mathrm{v}}, \mathbf{z}_i^{\mathrm{t}}) = &\sum_{\hat{\mathbf{z}}_i^{\mathrm{t}}} [\gamma - s(\mathbf{z}_i^{\mathrm{v}}, \mathbf{z}_i^{\mathrm{t}}) + s(\mathbf{z}_i^{\mathrm{v}}, \hat{\mathbf{z}}_i^{\mathrm{t}})]_+ \\ &+ \sum_{\hat{\mathbf{z}}_i^{\mathrm{v}}} [\gamma - s(\mathbf{z}_i^{\mathrm{v}}, \mathbf{z}_i^{\mathrm{t}}) + s(\hat{\mathbf{z}}_i^{\mathrm{v}}, \mathbf{z}_i^{\mathrm{t}})]_+, \end{aligned} \tag{9}$$

where $\hat{\mathbf{z}}_i^{\mathrm{v}}, \hat{\mathbf{z}}_i^{\mathrm{t}}$ are negatives, $[x]_+ \equiv \max(x, 0)$, and $s(\cdot)$ is the cosine similarity measurement of vision and text. The similarity in positive pairs should higher than in negative pairs by a margin γ, otherwise the loss may be created.

Dictionary Discriminant Learning. Based on the two encoded parts of the dictionary denoted as $\mathcal{S}_{\mathrm{v}} = \{\mathbf{F}_1^{\mathrm{D,v}}, \mathbf{F}_2^{\mathrm{D,v}}, \cdots, \mathbf{F}_K^{\mathrm{D,v}}\}$ and $\mathcal{S}_{\mathrm{t}} = \{\mathbf{F}_1^{\mathrm{D,t}}, \mathbf{F}_2^{\mathrm{D,t}}, \cdots, \mathbf{F}_K^{\mathrm{D,t}}\}$, the WD-loss E_w takes the following form:

$$E_w = \frac{\sum_{i=1}^{K} \sum_{j \in [1,K], j > i} T_{\mathrm{v,t}}^{\lambda}(i,j) W_{\lambda}(\mathbf{F}_i^{\mathrm{D,v}}, \mathbf{F}_j^{\mathrm{D,t}})}{\sum_{k=1}^{K} T_{\mathrm{v,t}}^{\lambda}(k,k) W_{\lambda}(\mathbf{F}_k^{\mathrm{D,v}}, \mathbf{F}_k^{\mathrm{D,t}})}, \tag{10}$$

$$\text{s.t. } \mathbf{T}_{\mathrm{v,t}}^{\lambda} = \arg\min \lambda \langle \mathbf{T}_{\mathrm{v,t}}, \mathbf{M}_{\mathrm{v,t}} \rangle - \Omega(\mathbf{T}_{\mathrm{v,t}}). \tag{11}$$

In above equations, $\mathbf{T}_{\mathrm{v,t}}^{\lambda}$ is the transport matrix imposing weights on the distance of the two keys from $\mathcal{S}_{\mathrm{v}}$ and $\mathcal{S}_{\mathrm{t}}$, respectively, $\mathbf{M}_{\mathrm{v,t}}$ is a W-distance matrix whose each element measures the W-distance between the corresponding pair of graph keys from $\mathcal{S}_{\mathrm{v}}$ and $\mathcal{S}_{\mathrm{t}}$, respectively.

To optimize the whole architecture, we use the back-propagation to adjust the parameters involved in both the graph encoders corresponding to input graphs and WGE. However, for the parameters of graph encoders for the dictionary keys, i.e. $\Phi^{\mathrm{D,v}}$ and $\Phi^{\mathrm{D,t}}$, the momentum update strategy, which is widely used in previous works [12, 41, 40], is employed. The reason is explained in [12] that rapidly changing encoders may reduce the dictionary representations' consistency and lead to poor result. For this issue, the momentum update mechanism [12] could make the dictionary encoders to evolve more smoothly to obtain better encoded features. Formally, taking the parameters of two encoders corresponding to the vision, i.e. Φ^{v} and $\Phi^{\mathrm{D,v}}$, as an example, the update process has the following form:

$$\Phi^{\mathrm{D,v}} \leftarrow m\Phi^{\mathrm{D,v}} + (1-m)\Phi^{\mathrm{v}}, \tag{12}$$

where m means a momentum coefficient. Similarly, we can also update $\Phi^{\mathrm{D,t}}$ with the momentum mechanism above.

4. Experiments

In this section, we will first introduce the four used datasets [20, 39, 28, 46] and implementation details, then show the comparison results on the datasets, and finally analyze our model by conducting ablation experiments.

4.1. Datasets and Protocols

The Image-text Datasets. The Real-world Scene Graphs datasets [20] contains 5,000 images with 5,000 human-annotated scene graphs that describe these images in detail. Each image can be described by three elements named object, attribute, and relationship. For the cross-modal task, graphs are first constructed from images and texts, then one modality is alternately used to retrieve the other based on these graphs. For the performance evaluation, we follow the same protocol in [20] that 4,000 images are used for training and 1,000 for testing.

Flickr30K [39] contains 31,000 images and 155,000 captions. Following the same protocol [29, 48, 30], for the Flickr30K, we use 29,000 images, 1,000 images and 1,000 images for training, validation and testing, respectively.

For the benchmark MSCOCO [28] which contains 123,287 images and 616,435 captions, it is split into 113,287 training, 5,000 validation and 5,000 testing images. The performance is calculated by 5-folds of testing images.

The Video-text Dataset. The Moviegraphs dataset [46] contains 51 movies in total. Each movie is attached with the corresponding textual description. These movies and texts are split into 7,637 corresponding video and text samples. We use the graphs generated from video and text samples to retrieve each other, which is more challenging compared with the protocol in Moviegraphs [46] that uses the annotated graph. We divide the 7,637 clips into 5,050 training clips, 1,060 validation clips, and 1,527 testing clips.

Table 1. The comparison results on the Flickr30K and MSCOCO datasets.

Dataset	**Flickr30K**						**MSCOCO**					
Method	**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
DSPE [47]	40.3	68.9	79.9	29.7	60.1	72.1	50.1	79.7	89.2	39.6	75.2	86.9
VSE++ [8]	52.9	79.1	87.2	39.6	69.6	79.5	64.7	-	95.9	52.0	-	92.0
GXN [11]	56.8	-	89.6	41.5	-	80.1	68.5	-	97.9	56.6	-	94.5
SCAN [26]	67.4	90.3	95.8	48.6	77.7	85.2	72.7	94.8	98.4	58.8	88.4	94.8
BFAN [29]	68.1	91.4	-	50.8	78.4	-	74.9	95.2	-	59.4	88.4	-
GOT [2]	70.9	92.8	95.5	50.7	78.7	86.2	-	-	-	-	-	-
SGM [48]	71.8	91.7	95.5	53.5	79.6	86.5	73.4	93.8	97.8	57.5	87.3	94.3
GSMN(dense) [30]	72.6	**93.5**	96.8	53.7	80.0	87.0	74.7	95.3	98.2	60.3	88.5	94.6
WCGL	**74.8**	93.3	**96.8**	**54.8**	**80.6**	**87.5**	**75.4**	**95.5**	**98.6**	**60.8**	**89.3**	**95.3**

Table 2. The comparison results on the Real-word Scene Graph and Moviegraphs datasets.

Dataset	**Real-word Scene Graph**						**Moviegraphs**					
Method	**Image-to-Text**			**Text-to-Image**			**Video-to-Text**			**Text-to-Video**		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
PCA-cos	0.20	1.30	3.40	0.70	2.10	3.80	0.00	0.50	0.80	0.10	3.39	5.43
CCA-cos [45]	26.9	44.5	60.9	16.2	42.1	61.5	6.20	20.7	27.0	7.90	21.2	30.1
SG-obj [25]	-	-	-	11.3	26.0	34.7	-	-	-	-	-	-
GCN [21]	17.4	47.2	64.0	15.6	44.1	61.4	7.30	19.1	26.5	7.90	17.6	24.0
GWCA [53]	29.6	46.0	61.9	20.9	46.6	61.7	7.40	20.2	27.1	8.40	**23.6**	28.8
WCGL	**33.3**	**48.7**	**64.1**	**21.5**	**46.8**	**63.3**	**9.20**	**22.5**	**29.5**	**10.7**	22.2	**30.8**

4.2. Graph Generation

To achieve comprehensive and fair comparison, we strictly follow the baselines to construct nodes and edges for vision/text graphs on four datasets. Below, we introduce them formally. For more details, please refer to the baselines, SG-obj [20], GSMN(dense) [30] and GWCA [53].

Real-world Scene Graphs. We construct image and text graphs following the setup of "SG-obj" method [20]. For an image, we use R-CNN [10] to detect related objects. Each detected object is regarded as a node, and the corresponding object potentials are used as the feature description. For the corresponding text, we search the words related to the object nodes and extract their feature through the GloVe [38]. For both image and text graphs, we construct edges based on the information provided by this dataset.

Flickr30K and MSCOCO. The graph generation manners of the Flickr30K and MSCOCO datasets are the same. Following the methods [26, 29, 30], given an image I, N salient regions are detected by Faster-RCNN which is pretrained on Visual Genome [23], and then are feed into the pretrained ResNet-101 [14] to extract features. For each word of a context T, we employ a bi-directional GRU [43] to integrate forward and backward contextual information. Finally, we can obtain word representation by averaging both directional hidden states. For edges of an image graph, we use the polar coordinate to model the spatial relation of regions. Meanwhile, the textual graph is a fully-connected graph which is consistent with GSMN(dense) model [30].

Moviegraphs. According to the method in [46, 53], the video and text graphs contain the character and attribute nodes, where the attribute includes the age, gender and emotion states. For an video graph, we first detect the faces of actors in each clip, and assign their names based on the IMDb gallery images. Also based on the detected faces, we train different classifiers to predict the attributes of age, gender and emotion states. Then, all these kinds of nodes, e.g. names and emotion states with discrete states, are encoded into one-hot vectors and are further adjusted to the same length to form the video graph. For the text description, we first search the character and attribute related words, and extract their feature through GloVe [38]. For both types of graphs, we take the cosine similarity as the weight of edges.

4.3. Implementation Details

Parameter Setting. The parameter setting is the same on all the used datasets. For the detailed architecture of our WCGL framework, we use the one-layer GCN as the graph encoders, which transform the feature dimension of the input graphs to 128. For the graph dictionary, the number of keys corresponding to one modality is set to 12. Besides, the regularization coefficient λ for calculating the OT matrix in

Eqn. (3) and the trade-off parameter β in Eqn. (8) are set to 1 and 0.01, respectively. The batch size is 64. During training, most parameters are optimized with Adam optimizer for 60 epochs with the weight decay of 10^{-4} and the learning rate of 0.001, except those ones in the momentum graph encoders for graph keys. The graph encoders for graph keys are updated according to Eqn. (12) through the momentum strategy with the coefficient m of 0.999 as used in [12].

Dictionary Construction. For the coupled dictionary containing two parts of graph keys, each part is initialized by randomly selecting a fixed number of constructed graph samples of the corresponding modality. Specifically, to guarantee the counterpart relationship of the keys in the coupled dictionary, the pair of graph keys from different parts should come from the matched video/image and text samples. Moreover, the counterpart relationship can be preserved during training with the constraint of the WD-loss.

4.4. Experiment Results

We compare the WCGL framework with multiple kinds of methods. For Flickr30K and MSCOCO datasets, the baselines include: (1) DSPE [47] and VSE++ [8] learning global correspondence between image-text; (2) region-word correspondence-learning methods, including GXN [11], BFAN [26], GOT [2], SGM [48] and GSMN [30]. For the Real-word Scene Graph and Moviegraphs datasets, the baseline methods can be divided into: (1) the learning methods in Euclidean space, including the Principal Component Analysis (PCA)-cos (cosine similarity), Canonical Correlation Analysis (CCA) [45]-cos, and the SG-obj using conditional random field (CRF) [25]; (2) the graph learning methods including GCN [21] and GWCA [53]. The results using the metric of Recall@1, 5, 10 (R@1, 5, 10) are shown in Table 1 and Table 2. We have the following observations:

(1) For the Flickr30K and MSCOCO datasets, methods based on global image-text correspondence generally perform worse than the other baseline methods. The reason may be that the former ignores detailed structral information. Noteworthy, GOT [2] only uses W-metric directly for graph matching, which may not be a good solution for the diversity of cross-modal data.

(2) For the Real-word Scene Graph and Moviegraphs datasets, in general, the learning methods in Euclidean space achieve relatively low performance while CCA achieves relatively high performance among them, which may be because it additionally considers the correlation between those cross-modal samples. Moreover, GWCA achieves the better performance than the others. It would be advantageous in jointly exploiting the graph topological structure and measuring the 2^{th} W-distance (based on the calculation of the mean and covariance of cross-graph nodes) between graphs.

(3) Our WCGL achieves the best performance in almost all the cases comparing to the other methods. The performance gain of the proposed WCGL over GWCA and GSMN(dense) verifies the effectiveness of the proposed joint graph and dictionary discriminant learning.

4.5. Ablation Study

It is meaningful to make clear how the modules or parameter setting influence the performance of cross-modal the retrieval task. For this purpose, we conduct several additional experiments to dissect our framework on the Realword Scene Graph dataset by Text-to-Image.

Table 3. The results of the ablation study by Text-to-Image on the Real-world Sence Graph.

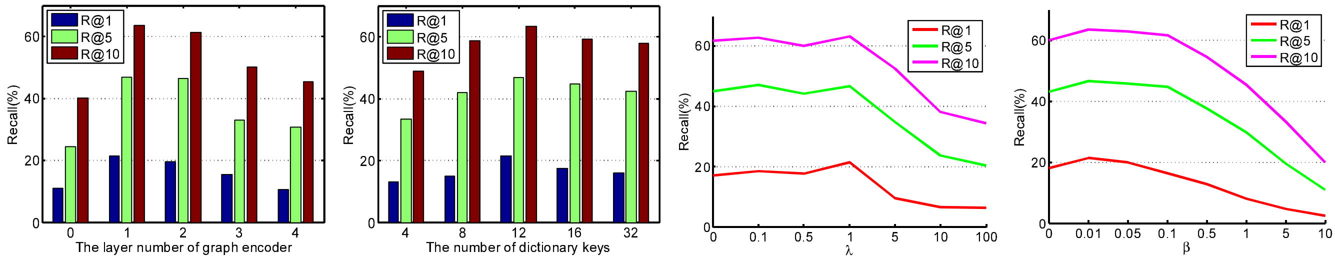
Method	R@1	R@5	R@10
One-layer GCN	15.6	44.1	61.4
WCGL_No_WD	18.0	43.2	60.0
WCGL	**21.5**	**46.8**	**63.3**

Table 4. The results of different dictionary strategies on Flickr30K.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
WCGL_single	70.0	91.2	94.7	52.7	79.4	86.6
WCGL_coupled	71.1	92.6	95.8	53.0	79.7	87.0
WCGL	**74.8**	**93.3**	**96.8**	**54.8**	**80.6**	**87.5**

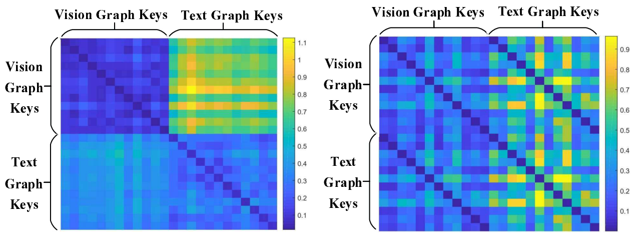
The effectiveness of the coupled dictionary. We simply remove the coupled dictionary module to evaluate its effectiveness. This operation actually results in a one-layer GCN and the performance could be found in Table 3. The performance gap may come from the fact that the coupled dictionary captures the structural correlation between graphs, which can not be well modeled by GCNs. In addition, we also compare different dictionary strategies [22] in Table 4. As it is shown, our WCGL achieves the best performance. Specifically, WCGL_single means to use a single and fixed common dictionary while WCGL_coupled means to use the fixed coupled dictionary to learn graph embeddings.

The influence of the layer number of GCNs in the graph encoders. To quantify the influence of the layer number, we vary it in the range {0,1,2,3,4}. Specifically, the layer number of 0 means that we use two projection layers instead of GCNs to learn the features of nodes. As it is shown in Fig. 2(a), one-layer GCN achieves the best results, while stacking more layers of GCNs actually degrades the performance. The reason may be that the repeating node aggregation of GCN in more layers causes the over-smooth problem and makes the nodes less distinguished.



(a) (b) (c) (d)

Figure 2. Evaluating the influence of the number of GCN layers (a), the number of keys in each part of the dictionary (b), the regularization coefficient λ (c), and the trade-off coefficient β for the WD-loss (d) on the Real-word Scene Graph dataset by Text-to-Image.



(a) (b)

Figure 3. The visualization of cross-key W-distances of the dictionary. (a) and (b) show the cross-key W-distances of the dictionary before and after optimization, respectively.

The influence of the key number of each part in the dictionary. We tune the key number in the range of $\{4, 8, 12, 16, 32\}$ to see how the performance varies accordingly. The result is shown in Fig. 2(b). Generally, in the range of [0, 12], more keys lead to higher performance, which is reasonable as more keys bring better feature representation ability. However, excessive graph keys may on the contrary degrade the performance of the WCGL model, as too much redundancy would be led into the embedding in the dictionary space, and then influence the similarity measurement.

The evaluation of regularization coefficient λ in Eqn. (5). The value of the regularization coefficient λ is tuned in the range of $\{0, 0.1, 0.5, 1, 5, 10, 100\}$, and the according results are shown in Fig. 2(c). It can be seen that relatively small performance fluctuation exists when λ is less than 1. However, the performance degrades obviously when λ is further increased to 10. This is because λ controls local information between the points across two graphs. For too large values of λ, the OT would become little sensitive to the local correlation between the input graph and the corresponding keys, which would decrease the performance.

The evaluation of the WD-loss. To evaluate the WD-loss, we first remove it from the WCGL framework (WCGL_No_WD in Table 3). As Table 3 shows, the performance degrades obviously after removing the WD-loss. Furthermore, we additionally set the range of β as $\{0, 0.01, 0.1, 0.5, 1, 5, 10\}$ (please see Fig. 2(d)). With an appropriate value (about the range of (0, 0.01]), the WD-loss could generally further promote the performance. However, a too large value of the trade-off coefficient β sharply degrades the performance. The reason may be that the large value of β leads to an imbalance and a large bias during the training, which reduces the discriminability of the graph and keys representation (please see Eqn. (8)). Additionally, we visualize the cross-key W-distances before and after optimization. As shown in Fig. 3, WD-loss can endow intra-class compactness and inter-class dispersion, where the distances between coupled keys are near zero.

5. Conclusion

In this paper, a deep WCGL framework was proposed for cross-modal retrieval task. Considering the difficulty of the discriminant analysis in W-space, as well as the diversity of cross-modal data, a coupled graph dictionary was constructed as the reference set to avoid calculating statistics on the mean and covariance of graphs. Based on the constructed dictionary, the cross-modal graphs can be transformed into the dictionary space through the proposed WGE, which better facilitates the cross-modal similarity calculation. To learn more discriminative features, the coupled dictionary was made to be dynamically updated in the training process. Moreover, the discriminant learning on the coupled dictionary was performed by introducing the WD-loss, which constrains the intra-class compactness and inter-class dispersion. We evaluated the proposed model on four public cross-modal retrieval datasets, and dissected the framework with ablation analysis. The experimental results verified the effectiveness of our framework.

6. Acknowledgements

This work was supported by the Natural Science Foundation of Jiangsu Province (BK20190019, BK20190452), the National Natural Science Foundation of China (62072244, 61906094), the Fundamental Research Funds for the Central Universities (No. 30921011104).

References

[1] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.

[2] Liqun Chen, Zhe Gan, Y. Cheng, Linjie Li, L. Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. *ArXiv*, abs/2006.14744, 2020.

[3] Xinlei Chen and C Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[7] J. Dong, Xirong Li, and Cees G. M. Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia*, 20:3377–3388, 2018.

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

[9] Rémi Flamary, Marco Cuturi, Nicolas Courty, and Alain Rakotomamonjy. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[11] Jiuxiang Gu, J. Cai, Shafiq R. Joty, Li Niu, and G. Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7181–7189, 2018.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[16] Xiaobin Hong, Tong Zhang, Zhen Cui, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang. Graph game embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7711–7720, 2021.

[17] Yan Huang, Qi Wu, and Liang Wang. Learning semantic concepts and order for image and sentence matching. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2018.

[18] K. Jia, Xiaogang Wang, and X. Tang. Image transformation based on learning dictionaries across image spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:367–380, 2013.

[19] Tong Zhang, Yun Wang, Zhen Cui, Chuanwei Zhou, Baoliang Cui, Haikuan Huang, and Jian Yang. Deep wasserstein graph discriminant learning for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10914–10922, 2021.

[20] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[22] S. Kolouri, Navid Naderializadeh, G. Rohde, and Heiko Hoffmann. Wasserstein embedding for graph learning. *ArXiv*, abs/2006.09430, 2021.

[23] R. Krishna, Yuke Zhu, O. Groth, J. Johnson, Kenji Hata, J. Kravitz, Stephanie Chen, Yannis Kalantidis, L. Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

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

[27] Kunpeng Li, Yulun Zhang, K. Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4653–4661, 2019.

[28] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[29] C. Liu, Zhendong Mao, Anan Liu, Tianzhu Zhang, Bo Wang, and Yongdong Zhang. Focus your attention: A bidirectional focal attention network for image-text matching. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[30] C. Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for

image-text matching. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10918–10927, 2020.

[31] Xinchen Liu, Wu Liu, Meng Zhang, Jingwen Chen, L. Gao, C. Yan, and T. Mei. Social relation recognition from videos via multi-scale spatial-temporal reasoning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3561–3569, 2019.

[32] Xiao Liu, Mingli Song, Dacheng Tao, Xingchen Zhou, Chun Chen, and Jiajun Bu. Semi-supervised coupled dictionary learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3557, 2014.

[33] Zhiling Luo, Ling Liu, Jianwei Yin, Ying Li, and Zhaohui Wu. Deep learning of graphs with ngram convolutional neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2125–2139, 2017.

[34] Tong Zhang, Baoliang Cui, Zhen Cui, Haikuan Huang, Jian Yang, Hongbo Deng, and Bo Zheng. Cross-graph convolution learning for large-scale text-picture shopping guide in e-commerce search. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1657–1666. IEEE, 2020.

[35] Devraj Mandal and Soma Biswas. Generalized coupled dictionary learning approach with applications to cross-modal matching. *IEEE Transactions on Image Processing*, 25(8):3826–3837, 2016.

[36] Niluthpol Chowdhury Mithun, Juncheng Billy Li, F. Metze, and A. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 2018.

[37] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[38] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[39] Bryan A. Plummer, Liwei Wang, C. Cervantes, Juan C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74–93, 2015.

[40] Senthil Purushwalkam Shiva Prakash and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *Advances in Neural Information Processing Systems*, 33, 2020.

[41] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160, 2020.

[42] Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs. *IEEE transactions on signal processing*, 61(7):1644–1656, 2013.

[43] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681, 1997.

[44] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.

[45] Cajo JF Ter Braak. Canonical correspondence analysis: a new eigenvector technique for multivariate direct gradient analysis. *Ecology*, 67(5):1167–1179, 1986.

[46] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8581–8590, 2018.

[47] L. Wang, Y. Li, J. Huang, and S. Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:394–407, 2019.

[48] S. Wang, R. Wang, Z. Yao, S. Shan, and X. Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1497–1506, 2020.

[49] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2216–2223. IEEE, 2012.

[50] R. Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[51] Jianchao Yang, Zhaowen Wang, Zhe L. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21:3467–3478, 2012.

[52] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3261–3269, 2017.

[53] Xueya Zhang, Tong Zhang, Xiaobin Hong, Zhen Cui, and Jian Yang. Graph wasserstein correlation analysis for movie retrieval. In *European Conference on Computer Vision*, pages 424–439. Springer, 2020.

[54] Yan Zhang, Rongrong Ji, Xiaopeng Fan, Yan Wang, Feng Guo, Yue Gao, and Debin Zhao. Search-based depth estimation via coupled dictionary learning with large-margin structure inference. In *European Conference on Computer Vision*, pages 858–874. Springer, 2016.