Graph Wasserstein Correlation Analysis for Movie Retrieval

Xueya Zhang, Tong Zhang, Xiaobin Hong, Zhen Cui, and Jian Yang

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
{zhangxueya,tong.zhang,xbhong,zhen.cui,csjyang}@njust.edu.cn

Abstract. Movie graphs play an important role to bridge heterogenous modalities of videos and texts in human-centric retrieval. In this work, we propose Graph Wasserstein Correlation Analysis (GWCA) to deal with the core issue therein, i.e, cross heterogeneous graph comparison. Spectral graph filtering is introduced to encode graph signals, which are then embedded as probability distributions in a Wasserstein space, called graph Wasserstein metric learning. Such a seamless integration of graph signal filtering together with metric learning results in a surprise consistency on both learning processes, in which the goal of metric learning is just to optimize signal filters or vice versa. Further, we derive the solution of the graph comparison model as a classic generalized eigenvalue decomposition problem, which has an exactly closed-form solution. Finally, GWCA together with movie/text graphs generation are unified into the framework of movie retrieval to evaluate our proposed method. Extensive experiments on MovieGrpahs dataset demonstrate the effectiveness of our GWCA as well as the entire framework.

Keywords: graph Wasserstein metric · graph correlation analysis · movie retrieval

1 Introduction

Nowadays, people show growing enthusiasm in searching desired movie clips, which contain either attractive plots or funny dialogue with vivid performance of actors, for multiple purposes including materials accumulation for presentation and entertainment. However, in many cases, they can just describe their understanding/impression of plots or dialogue content of those target clips, but are hardly accessible to the exact movie names or frame locations. This makes it time/energy-consuming and tedious to search the desired clips by manually browsing those movies one by one. Consequently, automatic movie-text retrieval become quite necessary and meaningful.

Xueya Zhang and Tong Zhang have equal contributions.
Corresponding author: zhen.cui@njust.edu.cn.

Among movie retrieval, the elements mainly consist of visual videos and descriptive texts, which have been investigated in some cross tasks such as video description [6] and video/image query and answer (Q & A) [28]. Most methods take some sophisticated dynamic models, e.g.,gated recurrent unit (GRU) [8] and long-short term memory (LSTM) [32], to capture the dynamics within both videos and texts, and then bridge them based on those obtained representation. However, these do not cater to flexible movie contour search, where some actors might be only posed by one searcher. Just to address this case, recently MovieGraphs dataset [30] is successfully initiated with annotated graphs to describe the interactions of entities in movie clips, and provides rather appropriate evaluations on more flexible movie-description retrieval for boosting machine understanding on movie clips.

Motivated by this case, in this work, we follow the technique line of graph modeling, which is more versatile to describe structured information in human-centric situation of movie graph retrieval. As a universal tool, graph can represent various data in the real world by defining nodes and edges that reveal multiple relationships between objects. For one given movie clip, those actors or other entities could be understood as nodes, their interactions may be defined as the edge connections. Accordingly, the text description can also be modeled with graph structure. Hence, the task of movie retrieval can be converted into the problem of graph searching, whose core issue is the comparisons between graph structured data. The inter-graph comparison contains two crucial problems: graph signal processing and graph distance metric. The former focuses on how to mine useful information from graph structure data, while the latter concerns the measurement of two graphs. On one hand, the obstacle to encode graph signals is not only to process graph signals as discrete time signal but also need model dependencies arising from irregular data. On the other hand, for graph structured data, Euclidean metrics fundamentally limit the ability to capture latent semantic structures, which however need not conform to Euclidean spatial assumptions. Further, could graph signal processing be seamlessly integrated with graph distance metric learning for more effective comparisons between graph data?

In this paper, we propose a Graph Wasserstein Correlation Analysis (GWCA) method to deal with the comparisons of pairwise movie graphs. The proposed GWCA elegantly formulates graph signal encoding together with graph distance metric learning into a unified model. Inspired by the recent spectral graph theory, we encode graph structure data with spectral graph filtering, which generalizes the previous classic signal processing. Instead of direct frequency domain, we take an approximation strategy, i.e., the polynomial of graph Laplacian, to efficiently encode graph data. The encoded signals of graph are embedded as probability distributions in a Wasserstein space, which is much larger and more flexible than Euclidean space. Accordingly, the distance between graph data is defined in Wasserstein space, which is called Wasserstein metric. Such a metric can not only captures the similarity of the distributions of graph signals, but also be able to preserve the transitivity in embedding space. In this way, graph signal

filtering and Wasserstein metric learning are jointly encapsulated into a unified mathematic model, which efficiently preserves the first-order and second-order proximity of the nodes of graph, empowering the learned node representations to reflect both graph topology structure. Surprisingly, we derive this model as a classic eigenvalue decomposition problem with closed-form solution, where the solution is just associated with graph encoding. Finally, our GWCA is used to movie graph retrieval, where multiple heterogeneous graphs are built and crossly-compared, e.g., annotation graph versus description graph, video graph versus annotation graph, etc. Extensive experiments on MovieGraphs dataset demonstrate the effectiveness of our proposed method, and new state-of-the-art results are also achieved.

In summary, our contribution are three folds:

- We propose an elegant inter-graph comparison model by seamlessly integrating graph signals filtering together with graph Wasserstein metric learning, where the latter is just the optimization of the former.
- We derive the solution of model as a classic generalized eigenvalue decomposition problem, which has an exact closed-form solution.
- We design an entire framework for movie retrieval including graph generation and GWCA, and finally validate the effectiveness of our proposed method.

2 Related Work

Most relevant works are proposed to inference across vision and text, where multiple tasks are tackled including image-text modeling [13, 21, 33], video/image query and answer (Q & A) [28] and video-text retrieval [9, 26]. For image-text understanding, a majority of work generate descriptive sentences for vision, and especially, [6] including sentence generation and image retrieval to find the bi-directional mapping between images and their textual descriptions. For video based works, [3] focus on understanding action of characters with scripts and [11] learn the relations among actors. [18] proposed the method using the retrieved action samples for visual learning and achieving action classification based on texts. In [22], authors propose an LSTM with visual semantic embedding method. Recently, Vicol *et al.* [30] proposed a new dataset MovieGraphs for retrieving videos and text with graphs, which also shows graphs containing sufficient information help us to understand the video and text better.

Graph Signal Processing. Graphs are generic data representation forms, which describe the geometric structures of data domains effectively. From the perspective of graph signal processing, the data on these graphs can be regarded as a finite collection of samples, and the sample at each vertex in the graph is graph signal. [27] concluded that spectral graph theory is regarded as the tool for defining the frequency spectra , and as an extension of the Fourier transform of the graph. It benefits the construction of expander graphs [14], spectral clustering [31] and so on, including definitions and notations such as the Non-Normalized Graph Laplacian and Graph Fourier Transform. References [2, 12, 25, 29] generate low dimensional representations for high-dimensional data

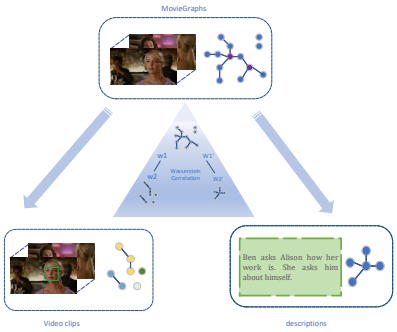

Fig. 1. Our proposed GWCA is used in two retrieval tasks. It jointly encapsulates graph signal filtering and Wasserstein metric learning into a unified mathematic model and W1 and W2 are learned in this process. Section 3 shows more details.

through spectral graph theory and the graph Laplacian [7], projecting the data on a low-dimensional subspace generated by a small subset of the Laplacian eigenbasis [2].
Generalized operators like filtering and translation then become the basis of developing the localized, multi-scale transforms. In [5], the basic graph spectral filtering enable discrete versions of continuous filtering, known as Gaussian smoothing, bilateral filtering, anisotropic diffusion, and non-local means filtering. Especially, Bruna *et al.* [4] consider possible generalizations of CNNs, which extends convolution networks to graph domains. Then Defferrard *et al.* [10] proposed a fast spectral filter, which use the Chebyshev polynomial approximation so that they are of the same linear computation complexity. Kipf *et al.* [16] motivate the convolutional architecture with a localized first-order approximation of spectral graph convolutions. In particular, [17, 20, 24] propose the literature of graph coarsening, downsampling and reduction. These graph modeling methods have also been applied to many tasks, such as node classification [15, 35], action recognition [19] and user recommendation [34].

3 Overview

In our task, we need to retrieval video clips and their descriptions using manually annotated graphs in [30] as queries. To better analyze the correspondence between annotated graphs, descriptions and video clips, we transform them into graph structured data and the task is converted into the problem of graph searching. For each pair of samples, we let $\mathbf{X}_1 \in \mathbb{R}^{n_1 \times d_1}, \mathbf{X}_2 \in \mathbb{R}^{n_2 \times d_2}$ denote the graph of samples in each pair respectively. To represent the annotated graph and constructed ourselves, following features are taken into consideration : 1) word embeddings for the annotated graph and the description; 2)features extracted by different neural networks for the video clip. The detail of the graph construction can be found in Section 5. In order to analyze graph correlation of

different magnitudes features, we project them into the same space and maximize the correlation between projections. We minimize the Wasserstein distance, that is, to learn weight parameters with regard of graphs. During training, we perform the metric training with pairwise samples and get weight parameters. In the process of testing, we search the most similar clip for the query over all the other clips with learned information.

4 Graph Correlation Analysis

Given a pair of (heterogeneous) graphs, e.g., annotation graph versus description graph, we denote them as $\mathcal{G}_1 = (\mathcal{V}_1, \mathbf{A}_1, \mathbf{X}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathbf{A}_2, \mathbf{X}_2)$, where $\mathcal{V}_1, \mathcal{V}_2$ are the node sets with the node numbers $|\mathcal{V}_1| = n_1$ and $|\mathcal{V}_2| = n_2$. The adjacent matrices $\mathbf{A}_1 \in \mathbb{R}^{n_1 \times n_1}, \mathbf{A}_2 \in \mathbb{R}^{n_2 \times n_2}$ record connections of edges, graph signals $\mathbf{X}_1 \in \mathbb{R}^{n_1 \times d_1}, \mathbf{X}_2 \in \mathbb{R}^{n_2 \times d_2}$ describe attributes of all nodes, where each row corresponds to the signal vector of one node therein and d_1, d_2 are the dimensions of signals. Our ultimate aim is to measure the distance of these two graphs $\mathcal{G}_1, \mathcal{G}_2$. Formally, we define the distance metric learning on these two graphs as

$$\mathcal{D}(\mathcal{G}_1, \mathcal{G}_2) = \mathcal{M}(\mathcal{F}(\mathcal{G}_1), \mathcal{F}(\mathcal{G}_2)), \tag{1}$$

where $\mathcal{F}(\cdot)$ is a function of graph signal processing, $\mathcal{M}(\cdot)$ is a distance metric function between two graphs.

4.1 Graph Filtering versus Graph Metric

Below we detailedly introduce the graph signal filtering function $\mathcal{F}$ and the graph metric learning function $\mathcal{M}$, and derive the consistency of their learning process that the metric learning could be viewed as signal filtering and vice versa.

Graph Signal Filtering In spectral graph theory, one main operator is the graph Laplacian operator, defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. The popular option is to normalize graph Laplacian, i.e.,

$$\mathbf{L}^{\text{norm}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{2}$$

where each edge A_{ij} is multiplied by a factor $\frac{1}{\sqrt{D_{ii}D_{jj}}}$, and $\mathbf{I}$ is an identity matrix. Unless otherwise specified, below we use the normalized version. Due to the symmetric and positive definite (SPD) property, the graph Laplacian $\mathbf{L}$ is with a complete set of orthonormal eigenvectors. Formally, we can decompose the Laplacian matrix into

$$\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^\top, \tag{3}$$

where $\Lambda = diag([\lambda_1, \lambda_2, \cdots, \lambda_n])$ with the spectrum $\lambda_i \geq 0$. In analogy to the classic Fourier transform, the graph Fourier transform and its inverse transform are defined as [27]

$$\widehat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}, \quad \mathbf{x} = \mathbf{U}\widehat{\mathbf{x}}, \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a graph signal of spatial domain, and $\widehat{\mathbf{x}}$ is the corresponding frequency signal.

Let $\mathcal{F}(\cdot)$ denote the filter function on graph $\mathcal{G}$, we can define the frequency response on an input signal $\mathbf{x}$ as $\widehat{z}(\lambda_l) = \widehat{x}(\lambda_l)\widehat{\mathcal{F}}(\lambda_l)$, and the inverse graph Fourier transform [27] as $z(i) = \sum_{l=1}^{N} \widehat{x}(\lambda_l)\widehat{\mathcal{F}}(\lambda_l)u_l(i)$, where $\widehat{z}(\lambda_l), \widehat{x}(\lambda_l), \widehat{\mathcal{F}}(\lambda_l)$ are the Fourier coefficients w.r.t the spectrum λ_l. In matrix form, the filtering process can be rewritten as

$$\mathbf{z} = \widehat{\mathcal{F}}(\mathbf{L})\mathbf{x} = \mathbf{U}diag[\widehat{\mathcal{F}}(\lambda_1), \cdots, \widehat{\mathcal{F}}(\lambda_n)]\mathbf{U}^\top \mathbf{x}. \tag{5}$$

Given the input signal $\mathbf{x}$ and the output response $\mathbf{z}$, our aim is to learn the filter function $\widehat{\mathcal{F}}(\cdot)$ in frequency domain, which suffers high-burden eigenvalue decomposition. To bypass it, we use a low order polynomial to approximate $\widehat{\mathcal{F}}(\cdot)$, formally, $\widehat{\mathcal{F}}(\lambda_l) = \sum_{k=0}^{K-1} \theta_k \lambda_l^k$, where $\theta = [\theta_0, \theta_1, \cdots, \theta_{K-1}]^\top \in \mathbb{R}^K$ is a vector of parameters w.r.t the polynomial coefficients, and K is the order number. By plug it into Eqn. (5), we can have

$$\begin{aligned}\mathbf{z} &= \mathbf{U}diag[\sum_{k=0}^{K-1} \theta_k \lambda_1^k, \cdots, \sum_{k=0}^{K-1} \theta_k \lambda_n^k]\mathbf{U}^\top \mathbf{x} \\ &= \sum_{k=0}^{K-1} \theta_k \mathbf{U}diag[\lambda_1^k, \cdots, \lambda_n^k]\mathbf{U}^\top \mathbf{x} = \sum_{k=0}^{K-1} \theta_k \mathbf{L}^k \mathbf{x}.\end{aligned}$$

Further, we may extend it to multi-dimensional signals $\mathbf{X}$, each of which is with different parameter, formally,

$$\mathbf{z} = \sum_{k=0}^{K-1} \sum_{j=1}^{d} \Theta_{kj} \mathbf{L}^k \mathbf{X}_{*j} = \sum_{k=0}^{K-1} \mathbf{L}^k \mathbf{X} \mathbf{w}^{(k)}, \tag{6}$$

$$\text{s.t. }, \quad \mathbf{w}^{(k)} = [\Theta_{k1}, \Theta_{k2}, \cdots, \Theta_{kd}]^\top, \tag{7}$$

where $\mathbf{X}_{*j}$ takes the j-th column of the matrix $\mathbf{X}$, Θ is the parameter to be learnt, and $\mathbf{w}^{(k)}$ is associated to the k-order term of the polynomial of graph Laplacian.

Graph Wasserstein Metric Learning Below we derive that $\mathbf{w}^{(k)}$ is also the parameters to be learnt in metric learning. To simply the derivation, we consider the k-order case and meantime omit the superscript of $\mathbf{w}^{(k)}$. For a pair of graphs $\mathcal{G}_1, \mathcal{G}_2$, the filtering response in the k-order case may be written as

$$\widetilde{\mathbf{x}}_1 = \mathbf{L}_1^k \mathbf{X}_1 \mathbf{w}_1, \quad \widetilde{\mathbf{x}}_2 = \mathbf{L}_2^k \mathbf{X}_2 \mathbf{w}_2, \tag{8}$$

where $\widetilde{\mathbf{x}}_1 \in \mathbb{R}^{n_1}, \widetilde{\mathbf{x}}_2 \in \mathbb{R}^{n_2}$ are one-dimensional signal of all nodes, and $\mathbf{w}_1 \in \mathbb{R}^{d_1}, \mathbf{w}_2 \in \mathbb{R}^{d_2}$ are the graph filtering parameters for the k-th polynomial term case.

We use the 2^{th} Wasserstein distance (abbreviated as W_2) for the output signals $\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2$. Note that each node only carries with one signal, and multi-channel signals could be easily extended. Formally, when all nodes of one graph is viewed a set of signals, we define second-order statistic distance as follows

$$\mathcal{D}=\|\mu_1-\mu_2\|_2^2+tr(\Sigma_1+\Sigma_2-2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}), \tag{9}$$

$$\text{s.t. ,} \quad \mu_1 = \frac{1}{n_1}\mathbf{1}_{n_1}^\top \widetilde{\mathbf{x}}_1, \quad \mu_2 = \frac{1}{n_2}\mathbf{1}_{n_2}^\top \widetilde{\mathbf{x}}_2, \tag{10}$$

$$\Sigma_1 = \frac{1}{n_1}(\widetilde{\mathbf{x}}_1 - \mu_1)^\top(\widetilde{\mathbf{x}}_1 - \mu_1), \tag{11}$$

$$\Sigma_2 = \frac{1}{n_2}(\widetilde{\mathbf{x}}_2 - \mu_2)^\top(\widetilde{\mathbf{x}}_2 - \mu_2). \tag{12}$$

By integrating Eqn. (8), Eqn. (10), Eqn. (11) and Eqn. (12) into the distance metic in Eqn. (9), we can derive out the following formulas

$$\mu_1^\top \mu_1 = \mathbf{w}_1^\top \mathbf{X}_1 \mathcal{K}_{\mu_1} \mathbf{X}_1 \mathbf{w}_1, \tag{13}$$

$$\mu_2^\top \mu_2 = \mathbf{w}_2^\top \mathbf{X}_2 \mathcal{K}_{\mu_2} \mathbf{X}_2 \mathbf{w}_2, \tag{14}$$

$$\mu_1^\top \mu_2 = \mathbf{w}_1^\top \mathbf{X}_1 \mathcal{K}_{\mu_1\mu_2} \mathbf{X}_2 \mathbf{w}_2, \tag{15}$$

$$\Sigma_1 = \mathbf{w}_1^\top \mathbf{X}_1 \mathcal{K}_{\Sigma_1} \mathbf{X}_1 \mathbf{w}_1, \tag{16}$$

$$\Sigma_2 = \mathbf{w}_2^\top \mathbf{X}_2 \mathcal{K}_{\Sigma_2} \mathbf{X}_2 \mathbf{w}_2, \tag{17}$$

$$(\Sigma_1\Sigma_2)^{1/2} \geq \mathbf{w}_1^\top \mathbf{X}_1^\top \mathcal{K}_{\Sigma_1\Sigma_2} \mathbf{X}_2 \mathbf{w}_2 \tag{18}$$

where each kernel term $\mathcal{K}$ is defined as

$$\mathcal{K}_{\mu_1} = \frac{1}{n_1^2}(\mathbf{L}_1^k)^\top \mathbf{1}_{n_1}\mathbf{1}_{n_1}^\top \mathbf{L}_1^k, \tag{19}$$

$$\mathcal{K}_{\mu_2} = \frac{1}{n_2^2}(\mathbf{L}_2^k)^\top \mathbf{1}_{n_2}\mathbf{1}_{n_2}^\top \mathbf{L}_2^k, \tag{20}$$

$$\mathcal{K}_{\mu_1\mu_2} = \frac{1}{n_1 n_2}(\mathbf{L}_1^k)^\top \mathbf{1}_{n_1}\mathbf{1}_{n_2}^\top \mathbf{L}_2^k, \tag{21}$$

$$\mathcal{K}_{\Sigma_1} = \frac{1}{n_1}(\mathbf{L}_1^k - \frac{1}{n_1}\mathbf{1}\mathbf{1}^\top \mathbf{L}_1^k)^\top(\mathbf{L}_1^k - \frac{1}{n_1}\mathbf{1}\mathbf{1}^\top \mathbf{L}_1^k), \tag{22}$$

$$\mathcal{K}_{\Sigma_2} = \frac{1}{n_2}(\mathbf{L}_2^k - \frac{1}{n_2}\mathbf{1}\mathbf{1}^\top \mathbf{L}_2^k)^\top(\mathbf{L}_2^k - \frac{1}{n_2}\mathbf{1}\mathbf{1}^\top \mathbf{L}_2^k), \tag{23}$$

$$\mathcal{K}_{\Sigma_1\Sigma_2} = \frac{1}{\sqrt{n_1 n_2}}(\mathbf{L}_1^k - \frac{1}{n_1}\mathbf{1}\mathbf{1}\mathbf{L}_1^k)^\top(\mathbf{L}_2^k - \frac{1}{n_2}\mathbf{1}\mathbf{1}\mathbf{L}_2^k). \tag{24}$$

In the above formulas, we can easily derive them except Eqn. (18). Next we give the derivation process of $(\Sigma_1\Sigma_2)^{1/2}$. We denote $\widetilde{\mathbf{x}}_1' = \widetilde{\mathbf{x}}_1 - \mu_1$ and $\widetilde{\mathbf{x}}_2' = \widetilde{\mathbf{x}}_2 - \mu_2$, and suppose the same dimensions (i.e., $n_1 = n_2$)[1], and then can

[1] We can pad zero values to one of them to produce the same dimensions for them.

have

$$(\Sigma_1\Sigma_2)^{1/2} = (\frac{1}{n_1 n_2}(\widetilde{\mathbf{x}}_1')^\top \widetilde{\mathbf{x}}_1'(\widetilde{\mathbf{x}}_2')^\top \widetilde{\mathbf{x}}_2')^{1/2} \tag{25}$$

$$\geq \frac{1}{\sqrt{n_1 n_2}}(\widetilde{\mathbf{x}}_1')^\top \widetilde{\mathbf{x}}_2', \tag{26}$$

where this inequation employs Cauchy inequality: $\sum_{i=1}^n a_i^2 \sum_{i=1}^n b_i^2 \geq (\sum_{i=1}^n a_i b_i)^2$. Next we plug Eqn. (8) and Eqn. (10) into the above equation and define the kernel term in Eqn. (24). After a series of derivation, we can reach the final Eqn. (18).

Now we can obtain the upper bound of Wasserstein distance metric, i.e.,

$$\begin{aligned}\mathcal{D} \leq\ & \mathbf{w}_1^\top \mathbf{X}_1^\top (\mathcal{K}_{\mu_1} + \mathcal{K}_{\Sigma_1})\mathbf{X}_1\mathbf{w}_1 \\ & + \mathbf{w}_2^\top \mathbf{X}_2^\top (\mathcal{K}_{\mu_2} + \mathcal{K}_{\Sigma_2})\mathbf{X}_2\mathbf{w}_2 \\ & - 2\mathbf{w}_1^\top \mathbf{X}_1^\top (\mathcal{K}_{\mu_1\mu_2} + \mathcal{K}_{\Sigma_1\Sigma_2})\mathbf{X}_2\mathbf{w}_2.\end{aligned} \tag{27}$$

The above bound is obviously the metric learning in Wasserstein space if we extend $\mathbf{w}_1, \mathbf{w}_2$ to multi-channel responses. Therefore, the Wasserstein metric learning is consistent with graph signal filtering. In other words, the aim of metric learning is to learn graph filters, and vice verse.

Wasserstein Correlation Analysis Given M pairs of matching graphs, $\{(\mathcal{G}_1^{(m)}, \mathcal{G}_2^{(m)})\}|_{m=1}^M$, we expect to learn the projection to make their as closer as possible, formally,

$$\arg\min_{\mathbf{w}_1,\mathbf{w}_2} \quad \sum_{m=1}^M \mathcal{D}(\mathcal{G}_1^{(m)}, \mathcal{G}_2^{(m)}). \tag{28}$$

We replace $\mathcal{D}$ with Eqn. (27), and then the objective function can be rewritten as

$$\arg\min_{\mathbf{w}_1,\mathbf{w}_2} \quad \mathbf{w}_1^\top \mathcal{C}_1 \mathbf{w}_1 + \mathbf{w}_2^\top \mathcal{C}_2 \mathbf{w}_2 - 2\mathbf{w}_1^\top \mathcal{C}_{12}\mathbf{w}_2, \tag{29}$$

where

$$\mathcal{C}_1 = \sum_{m=1}^M (\mathbf{X}_1^{(m)})^\top (\mathcal{K}_{\mu_1} + \mathcal{K}_{\Sigma_1})\mathbf{X}_1^{(m)}, \tag{30}$$

$$\mathcal{C}_2 = \sum_{m=1}^M (\mathbf{X}_2^{(m)})^\top (\mathcal{K}_{\mu_2} + \mathcal{K}_{\Sigma_2})\mathbf{X}_2^{(m)}, \tag{31}$$

$$\mathcal{C}_{12} = \sum_{m=1}^M (\mathbf{X}_1^{(m)})^\top (\mathcal{K}_{\mu_1\mu_2} + \mathcal{K}_{\Sigma_1\Sigma_2})\mathbf{X}_2^{(m)}. \tag{32}$$

An elegant alternative of the solution is to maximize the following objective function

$$\arg\max_{\mathbf{w}_1,\mathbf{w}_2} \frac{\mathbf{w}_1^\top \mathcal{C}_{12}\mathbf{w}_2}{\sqrt{\mathbf{w}_1^\top \mathcal{C}_1 \mathbf{w}_1}\sqrt{\mathbf{w}_2^\top \mathcal{C}_2 \mathbf{w}_2}}. \tag{33}$$

which finally falls into the category of canonical correlation analysis. Hence, this maximum optimization has a closed-form solution, which can be derived as the eigenvalue decomposition from

$$\mathcal{C}_1^{-1}\mathcal{C}_{12}\mathcal{C}_2^{-1}\mathcal{C}_{21}\mathbf{w}_1 = \rho^2\mathbf{w}_1 \tag{34}$$

$$\mathcal{C}_2^{-1}\mathcal{C}_{21}\mathcal{C}_1^{-1}\mathcal{C}_{12}\mathbf{w}_2 = \rho^2\mathbf{w}_2 \tag{35}$$

where $\mathbf{w}_1, \mathbf{w}_2$ are eigenvectors and ρ is the correlation coefficient.

Consequently, those eigenvectors with large correlation coefficients may be chosen as multi-channel projection functions. Further, with the change of the order k, we can learn the corresponding filtering functions also metrics.

5 Graph Generation

In this section, we introduce how we generate graphs on the MovieGraphs dataset. As structural difference exists between videos and descriptions of movies, different graphs are constructed accordingly.

5.1 Graph construction on videos

Formally, a graph can be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. Following the configuration of the Moviegraphs dataset, four types of nodes, which correspond to the nodes in the manually annotated graph of the dataset, are taken into account for video clips denoted as M. Nodes of character and attribute are denoted with the notations v^{ch} and v^{att} respectively. Another two independent nodes named scene and situation are specifically denoted as v^{sc} and v^{si}. Below, we introduce how we learn embeddings of these nodes, and set up connections between them.

Scene and situation. Scene and situation provide the context of the video. For each video clip, the Resnet is used to extract features from those frames, and features of every ten frames are averaged as the representation of scene and situation denoted as $\mathbf{x}_{sc} \in \mathbb{R}^{2048}$ and $\mathbf{x}_{si} \in \mathbb{R}^{2048}$, which is similar with the previous work [30].

Character. In the graph retrieval video task, in order to obtain the features of different types of nodes, e.g. facial expression and age, we first perform face detection on each frame [1], and then construct multiple clusters where each cluster is formed by those faces belonging to the same person. Moreover, we assign each cluster with one name according to the actor list in IMDB by comparing the features between the cluster and the actor picture (also provided in IMDB). Specifically, in the process of constructing face clusters, facial features are first extracted, and accordingly the Euclidean distances are calculated between faces for comparison. Also, a threshold is set to determine whether they are the same person. For each face cluster not aligned with an actor name, we randomly choose an unassigned name in the actor list for it.

Attribute. For each face cluster, its attribute node include age, gender, emotion, etc. Each attribute node is represented by the extracted feature of the

predicted attribute value [23] (e.g. "male" for gender), which is formally denoted as $\mathbf{x}_{att} \in \mathbb{R}^{300}$. These nodes form a graph where nodes with similar embeddings are connected.

5.2 Graph construction on Descriptions

In moviegraphs dataset, each video clip has a natural language description. To construct one graph for each description of the movie clip, after spliting the sentence and removing stopwords and notations, we statistics the total words while keep previous order to obtain a small corporus. Here we regard each word as graph node v^{des}, and each node in the textual graph has the representation of a fixed length by using GloVe embeddings [23]. Moreover, the intense of the edge between nodes is defined as the similarity between their embeddings.

6 Experiments

We conduct experiments on the MovieGraphs dataset with our proposed GWCA. The performance of GWCA is also compared with the results of those retrieval tasks in MovieGraphs [30]. Moreover, we conduct an ablation study to discuss the influence of different distance metrics and different orders of receptive fields.

6.1 Dataset and settings

MovieGraphs dataset consists of 51 movies with annotated textual description and graphs. Each movie is split into multiple rough scenes and then manually refined. As a result, the dataset contains 7637 clips in total and each clip has an annotated description and graph. There are 35 words on average in the description, and the average number of nodes per graph is also about 35. In the experiment, following the protocol in [30], the dataset is split into 5050 clips for training, 1060 clips for validation and 1527 clips for testing.

Two retrieval tasks are evaluated to test the performance: (1) descriptions retrieval using annotated graphs as queries, and (2) video clips retrieval using annotated graphs as queries. In the test stage, for each query graph, we search all the descriptions/video clips to find the most similar one. Following the previous work [30], we use "Recall" as the evaluation metric, and calculate the Recall@1(R@1), Recall@5(R@5) and Recall10(R@10) to explore the effectiveness of our GWCA. There R@K stands for the fraction of correct predicted results in the top K predictions.

6.2 The Comparison Results

We compare our proposed GWCA with those state-of-the-art methods, and the results are shown in Table 1.

Description Retrieval using graphs as queries. The results of description retrieval with query graphs are shown in Table 1, from the first row to third row

Table 1. The comparison results of two different retrieval tasks.

Method	Description			Method	Video		
	R@1	R@5	R@10		R@1	R@5	R@10
GloVe,idf·max-sum	61.3	81.6	86.9	sc	1.1	4.3	7.7
GloVe,max-sum	62.1	81.3	87.2	sc,si	1.0	5.4	8.7
TF·IDF	61.6	83.8	89.7	sc,si,a	2.2	9.4	15.5
GWCA	**67.6**	**87.8**	**91.9**	sc,si,a(ours)	**2.4**	**10.1**	**16.2**

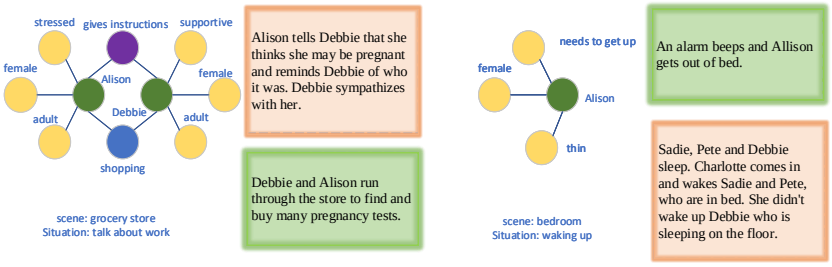

Fig. 2. The results for retrieved descriptions with graphs. We show the top-2 retrieved clips. The sub-graphs indicate the query graphs. The green boxes indicate the ground-truth and the red boxes indicate the quite similar one.

in the second column. For the compared methods, GloVe means that the GloVe word embedding is employed; max-sum and idf·max-sum are pooling strategies with word embedding. Specifically, idf·max-sum weights words with rarity. The previous method [30] finds the best matching word in description for each word in the manually annotated graph, and sum up them to compute the similarity. Then this processed score is fed into the loss function. According to Table 1, GloVe with the pooling strategy of max-sum achieves limited performance gain comparing with GloVe. TF·IDF, which uses an identity sparse matrix to initialize features, performs better than GloVe. Among these compared methods, our GWCA shows the best performance, where the score of Recall@1 is about 5.5% higher than GloVe with max-sum pooling. Besides, for those words with similar meaning/embedding which are sometimes confusing, our GWCA is still effective enough to compute the correlation and fulfill the retrieval task well. This observation demonstrates that GWCA successfully formulates graph signal encoding together with graph distance metric learning into a unified model. Besides, we show some retrieval examples in Fig. 2.

Video Retrieval using graphs as queries. This experiment aims to measure the performance of our method to retrieve videos based on the given annotated graphs. The result is shown in the third and fourth columns in Table 1. The characters 'si', 'sc' and 'a' indicate that we start with the scene, situation, attributes, and characters as part of graphs, while their corresponding methods all compute the cosine similarity to measure the distances between nodes. The reported result of our GWCA employs all the four kinds of nodes, and achieves

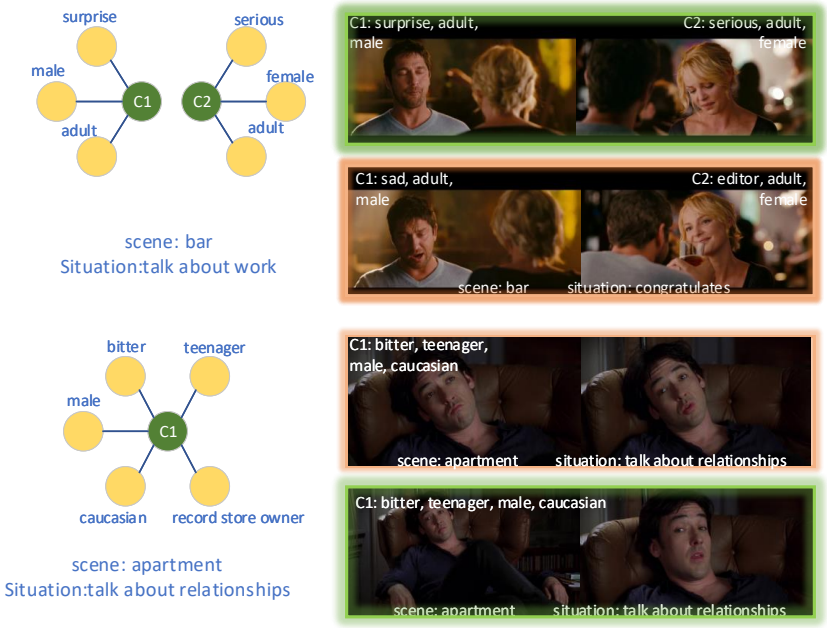

Fig. 3. The results for retrieved video clips with graphs. We show the top-2 retrieved clips. The red boxes indicate results that are quite similar in meaning to the query, and the green boxes indicate ground-truth.

the best performance. According to the shown result, we have the following observations: (1) the four different nodes, i.e. situation, scene, attribute and characters, all contribute to the video retrieval; (2) compared with other methods, our GWCA is advantageous in understanding the graph structure as it jointly encapsulates graph signal filtering and Wasserstein metric learning into a unified mathematic model helps to enhance the node representation ability. Some examples of the retrieved videos are visualized in Fig. 3 for the intuitive impression of our GWCA.

6.3 Ablation Study

In this section, we dissect our algorithm by conducting ablation analysis. Specifically, we evaluate how the modules, i.e. the graph Wasserstein metric, the order of receptive fields, and the dimension of the features of nodes, promote the retrieval. For this purpose, we conduct the following additional experiments:

(1) Comparing the performance between Graph Wasserstein Metric and cosine distance with different algorithms, e.g. PCA, CCA, and also the original feature without learning. The result is shown in Table 2.

(2) Comparing the performance of GWCA under different values of the order of receptive field k. Please see the result in Fig. 4. Fusion means the features of those orders that are lower than k are fused together as the feature.
(3) Comparing the performance of GWCA under different dimensions of node features. The result is shown in Fig. 5.

According to the results, we have the following observations:

(1) Graph Wasserstein Metric effectively promotes the performance in distance measurement between graph. Specifically, for both PCA and CCA, higher performances are achieved with Graph Wasserstein Metric than cosine similarity.
(2) The order of receptive field k also influences the performance. We focus on the situation of fusion as it achieves higher performance in Fig. 4. As it is shown, we can see that the best performance is achieved when k equals 2, otherwise the performance drops.
(3) The dimension of node feature is also an important factor influencing the retrieval performance. According to Fig. 5, the performance varies with different dimensions, and the best performance is achieved when the dimension is set to 240.

Table 2. The comparison results between cosine distance and Graph Wasserstein Metric using different algorithms.

Method		R@1	R@5	R@10
Ori Feature	cos	4.07	12.4	19.6
PCA	cos	35.9	55.5	64.8
	w-2	62.1	77.7	86.7
CCA	cos	61.9	78.3	85.3
	w-2	66.4	82.7	88.3
GWCA		**67.6**	**87.8**	**91.9**

7 Conclusion

In this paper, a Graph Wasserstein Correlation Analysis (GWCA) method was proposed to deal with the comparisons of pairwise movie graphs and show the effectiveness. We relabel some content ourselves after downloading the existing data, and then use GWCA to formulate graph signal encoding together with graph distance metric learning on this dataset. In this way, graph signal filtering and Wasserstein metric learning are jointly encapsulated into a unified model, which efficiently preserves the proximity of the nodes of graph and empowering the learned node representations. Extensive experiments and our visualizations analyze our method and we believe that our contribution can be applied to many domains.

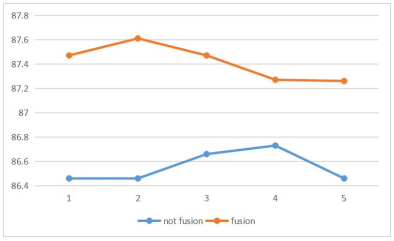

Fig. 4. Test Recall@5 with different orders, using GWCA in the description retrieval task.

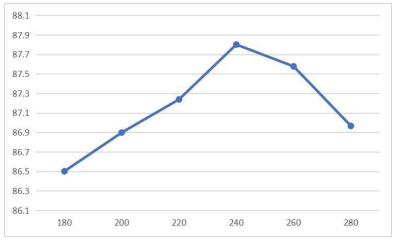

Fig. 5. Test Recall@5 with different dimensions, using GWCA in the description retrieval task.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants Nos. 61906094, 61972204), the Natural Science Foundation of Jiangsu Province (Grant Nos. BK20190019, BK20190452), and the fundamental research funds for the central universities (No. 30919011232).

References

1. `https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chinese.md/`
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003)
3. Bojanowski, P., Bach, F., Laptev, I., Ponce, J., Schmid, C., Sivic, J.: Finding actors and actions in movies. In: Proceedings of the IEEE international conference on computer vision. pp. 2280–2287 (2013)
4. Bruna, J., Zaremba, W., Szlam, A., Lecun, Y.: Spectral networks and locally connected networks on graphs. Computer Science (2014)
5. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation **4**(2), 490–530 (2005)
6. Chen, X., Zitnick, C.L.: Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654 (2014)
7. Chung, F.R.: Lectures on spectral graph theory. CBMS Lectures, Fresno **6**, 17–21 (1996)
8. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
9. Cour, T., Jordan, C., Miltsakaki, E., Taskar, B.: Movie/script: Alignment and parsing of video and text transcription. In: European Conference on Computer Vision. pp. 158–171. Springer (2008)
10. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)

11. Ding, L., Yilmaz, A.: Learning relations among movie characters: A social network perspective. In: European conference on computer vision. pp. 410–423. Springer (2010)
12. Donoho, D.L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proceedings of the National Academy of Sciences **100**(10), 5591–5596 (2003)
13. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: Generating sentences from images. In: European conference on computer vision. pp. 15–29. Springer (2010)
14. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bulletin of the American Mathematical Society **43**(4), 439–561 (2006)
15. Jiang, J., Cui, Z., Xu, C., Yang, J.: Gaussian-induced convolution for graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4007–4014 (2019)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
17. Lafon, S., Lee, A.B.: Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. IEEE transactions on pattern analysis and machine intelligence **28**(9), 1393–1403 (2006)
18. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
19. Li, B., Li, X., Zhang, Z., Wu, F.: Spatio-temporal graph routing for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8561–8568 (2019)
20. Narang, S.K., Ortega, A.: Lifting based wavelet transforms on graphs. In: Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference. pp. 441–444. Asia-Pacific Signal and Information Processing Association, 2009 Annual ... (2009)
21. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2text: Describing images using 1 million captioned photographs. In: Advances in neural information processing systems. pp. 1143–1151 (2011)
22. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4594–4602 (2016)
23. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
24. Ron, D., Safro, I., Brandt, A.: Relaxation-based coarsening and multiscale graph organization. Multiscale Modeling & Simulation **9**(1), 407–423 (2011)
25. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
26. Sankar, P., Jawahar, C., Zisserman, A.: Subtitle-free movie to script alignment. In: Proc. Brit. Mach. Vis. Conf. pp. 121–1 (2009)
27. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine **30**(3), 83–98 (2013)
28. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: Pro-

ceedings of the IEEE conference on computer vision and pattern recognition. pp. 4631–4640 (2016)
29. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. science **290**(5500), 2319–2323 (2000)
30. Vicol, P., Tapaswi, M., Castrejon, L., Fidler, S.: Moviegraphs: Towards understanding human-centric situations from videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8581–8590 (2018)
31. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
32. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015)
33. Yang, Y., Teo, C.L., Daumé III, H., Aloimonos, Y.: Corpus-guided sentence generation of natural images. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 444–454. Association for Computational Linguistics (2011)
34. Zhang, T., Cui, B., Cui, Z., Huang, H., Yang, J., Deng, H., Zheng, B.: 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). pp. 1657–1666. IEEE (2020)
35. Zhao, W., Cui, Z., Xu, C., Li, C., Zhang, T., Yang, J.: Hashing graph convolution for node classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 519–528 (2019)