Multi-Instance Multi-Graph Dual Embedding Learning

Jia Wu[†⋆], Xingquan Zhu[‡], Chengqi Zhang[†], Zhihua Cai[⋆]

[†] *Centre for Quantum Computation & Intelligent Systems, FEIT, University of Technology, Sydney, Australia*
[‡]*Dept. of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, USA*
[⋆]*Dept. of Computer Science, China University of Geosciences, Wuhan, China*
{jia.wu@student., chengqi.zhang@}uts.edu.au; xzhu3@fau.edu; zhcai@cug.edu.cn

***Abstract*—Multi-instance learning concerns about building learning models from a number of labeled instance bags, where each bag consists of instances with unknown labels. A bag is labeled positive if one or more multiple instances inside the bag is positive, and negative otherwise. For all existing multi-instance learning algorithms, they are only applicable to the setting where instances in each bag are represented by a set of well defined feature values. In this paper, we advance the problem to a multi-instance multi-graph setting, where a bag contains a number of instances and graphs in pairs, and the learning objective is to derive classification models from labeled bags, containing both instances and graphs, to predict previously unseen bags with maximum accuracy. To achieve the goal, the main challenge is to properly represent graphs inside each bag and further take advantage of complementary information between instance and graph pairs for learning. In the paper, we propose a Dual Embedding Multi-Instance Multi-Graph Learning (*DE-MIMG*) algorithm, which employs a dual embedding learning approach to (1) embed instance distributions into the informative subgraphs discovery process, and (2) embed discovered subgraphs into the instance feature selection process. The dual embedding process results in an optimal representation for each bag to provide combined instance and graph information for learning. Experiments and comparisons on real-world multi-instance multi-graph learning tasks demonstrate the algorithm performance.**

***Keywords*—Classification; Multi-instance; Multi-graph; Graph; Embedding;**

I. Introduction

Multi-instance (MI) learning is a special leaning task where label is only available for a bag of instances. This problem is originated from drug activity prediction [1] and has been extended to many real-world applications, such as content-based image retrieval and text categorization, mainly because MI learning can accommodate label ambiguity and does not require label for each single instance. For example, in content-based image retrieval, an image (*i.e.* a bag) can be labeled as positive if any region (*i.e.* an instance) of the image contains an interesting object, and negative otherwise. Many MI learning methods exist and solutions can be roughly separated into two categories: (1) upgrading normal single-instance learning methods to handle MI data, including lazy learning [2] and kernel method [3], *etc.*; and (2) specifically designed methods to tackle MI learning, such as diverse density [4] and its variants with expectation maximization [5], bagging [6], and boosting approaches [7]. Some recent researches consider relations between instances in each bag [8], [9], and use kernel based approaches to solve the problem.

For existing MI learning algorithms, a perquisite is that instances in each bag are represented by features in a tabular instance-feature format, as shown in Figure 1. Such setting inherently forbids multi-instance learning from being applied to complex data environments, where each observation may contain features and dependency structures (graphs), as shown in Figure 2. For example, an online webpage may contain images and additional text descriptions, where text can be represented as instances and images can be represented as graphs (which are better than simply representing the whole image by using visual features such as color histograms or textures) [10]. As a result, a webpage can be regarded as a bag containing a number of mixed instances and graphs in pairs, each representing a portion of the webpage content. For each viewer, a webpage is interesting if one or multiple parts of the content (text or image) is interesting to him/her (*i.e.* A bag is positive if one or multiple pairs of instances and graphs inside the bag is positive). The same problem setting can be generalized to many other domains, such as online review based product recommendation, where each product receives many customer reviews. For each review composed of review summary and detailed text descriptions, we can use bag-of-words to represent review summary as an instance, and employ a graph to represent detailed review descriptions (this graph representation has better performance than bag-of-words representation in preserving contextual information in text [11]). As a result, each review contains an instance and graph pair, and a product can be represented as a bag of instances and graphs. Assume customers mainly concern about several key properties, such as "affordability" and "durability", of the product. A product (*i.e.* a bag) can be labeled as positive if it receives very positive review in any of these properties, and negative otherwise. As a result, we can use proposed approach to help recommend products to customers. Similarly, for a scientific publication, its title can be represented as an instance and the abstract can be represented as a graph. So each publication is represented as an instance-graph pair. A paper and all references cited in the paper can form a bag,

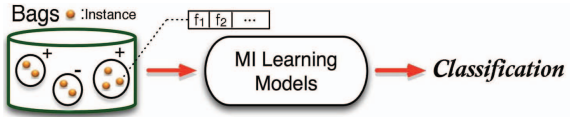

Fig. 1. Multi-instance learning where each bag contains several instances.

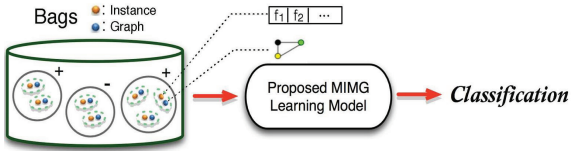

Fig. 2. Multi-instance multi-graph learning where each bag contains a number of instances and graphs in pairs.

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.121

which can be labeled as positive, if the paper or any of the reference cited in the paper is relevant to a specific topic.

The coexistence of instance and graph representation inside each bag provides a more powerful data representation capability than traditional multi-instance bags, it also raises significant challenges for deriving learning models from labeled bags for classification. Because, in this setting, the instances and graphs carry complementary yet redundant knowledge to describe the same object (*e.g.* a publication or a product). Effective learning models cannot be achieved without combining graph and instance information inside each bag for learning.

Indeed, to date, existing graph classification approaches mainly rely on two types of approaches [12], including (1) global distance based methods (*e.g.* graph kernels, graph embedding, and graph transformation), or (2) local subgraph feature based approaches, to represent graphs into a vector space so that generic supervised learning methods, such as decision tree, can be applied for classification. For all existing graph classification methods, they require each graph to be explicitly labeled and cannot handle the multi-graph (MG) setting where labels are only available for a group of graphs (*i.e.* a bag). On the other hand, existing MI learning algorithms cannot handle structured data, such as graphs, and are only applicable to bags with tabular instance-feature representations. This naturally raises the needs of designing new methods to handle bags containing mixed instances and bags.

For bags with mixed objects (*i.e.* instances or graphs), a straightforward solution is to treat objects as two separated views, so the multi-instance multi-graph (MIMG) setting can be solved by separating each bag into two sub-bags, each consisting of instances or graphs, respectively. Such a trivial solution, however, suffers from several major disadvantages, including (1) the instances and graphs are two views to provide complementary information for each bag. By separating instances and graphs into two sub-bags, it inherently treats each view in separated way (without allowing them to benefit each other) which will result in suboptimal learning results (as we will demonstrate in Section VI); (2) even if we can separate an bag as an instance and a graph sub-bag, there is no solution available to support multi-graph learning, where labels are only available for a bag of graphs. Completely discarding graphs in each bag is clearly not an option as it will result in severe information loss.

A slightly more intelligent design in solving MIMG learning is to transfer graphs into instances by using subgraph features [13] discovered from all graphs, so the problem is converted to a multi-instance multi-view (an instance feature view and a graph feature view) learning issue. This solution is still least effective mainly because (1) without utilizing valuable instance information and bag constraints, the discovered subgraph features cannot effectively characterize graphs inside (and between) bags; and (2) although there are many works related to multi-view learning [14], no systematic work exists for feature based multi-instance multi-view learning, and the only reference we found is [15] which treats each view in an independent way (as we described in the above paragraph).

Motivated by the above observations, in this paper, we propose a multi-instance multi-graph learning framework which allows each bag to contain pairs of instances and graphs. The key challenge, under the new data setting, is twofold:

- **Instance and Graph Feature Exploration:** Because graphs contain structured information, we need to explore effective features to represent graphs for MIMG learning. On the other hand, instance features may also contain redundancy and low quality features should be excluded to ensure that instances and graphs form consistent representation for learning. In a bag constrained environment, both instances and graphs provide complementary yet redundant information to describe each bag, so instance and graph feature exploration needs to be carried out in a mutually beneficial way with graph feature exploration taking instance distributions into consideration and instance feature exploration also considering graph features.
- **Multi-instance Multi-graph Bag Representation:** For bags with mixed instances and graphs, we need to find effective approaches to represent the bag, by using explored instance and graph features, for multi-instance multi-graph learning.

To solve the above challenges, we propose to use dual embedding to explore optimal graph features (*i.e.* subgraphs or subgraph features) and instance features for MIMG learning. More specifically, to select effective subgraph features, we embed instance distributions into the objective function to ensure that selected subgraph features are consistent with instance representation. Meanwhile, for instances in bags, we also explore optimal instance features by embedding graph distribution information. The dual embedding optimization ensures that the complementary information between graphs and instances can be fully utilized to explore a good feature representation for each bag to support MIMG learning. Experiments on two real-world learning tasks (scientific publication and online product recommendation) confirm the effectiveness of the proposed designs.

The remainder of the paper is organized as follows. Preliminary concepts and problem statement are addressed in Section II, followed by the overall framework in Section III. The dual embedding feature exploration framework is reported in Section IV. Section V outlines the proposed DEMIMG framework, followed by experiments in Section VI. We conclude the paper in Section VII.

II. Preliminaries and Problem Statement

In this section, we first introduce some important notations and definitions, then state our research problem.

DEFINITION 1 **Connected Graph**: A graph is represented as $G = (\mathcal{V}, E, \mathcal{L}, l)$ where $\mathcal{V}$ is a set of vertices $\mathcal{V} = \{v_1, \cdots, v_{n_v}\}$, $E \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, and $\mathcal{L}$ is the set of labels for the vertices and edges. $l : \mathcal{V} \cup E \rightarrow \mathcal{L}$ is the function assigning labels to the vertices and edges. A connected graph is a graph such that there is a path between any pair of vertices.

DEFINITION 2 **Bag**: A bag $B_i = \{B_i^{(I)}, B_i^{(G)}\}$ contains a number of objects (paired instances and graphs), in which $B_i^{(I)}$ and $B_i^{(G)}$ denote sub-bag with instances or graphs in B_i, respectively. A bag B_i's label is denoted by $y_i \in \mathcal{Y}$, with $\mathcal{Y} = \{-1, +1\}$. A bag is labeled positive if one or

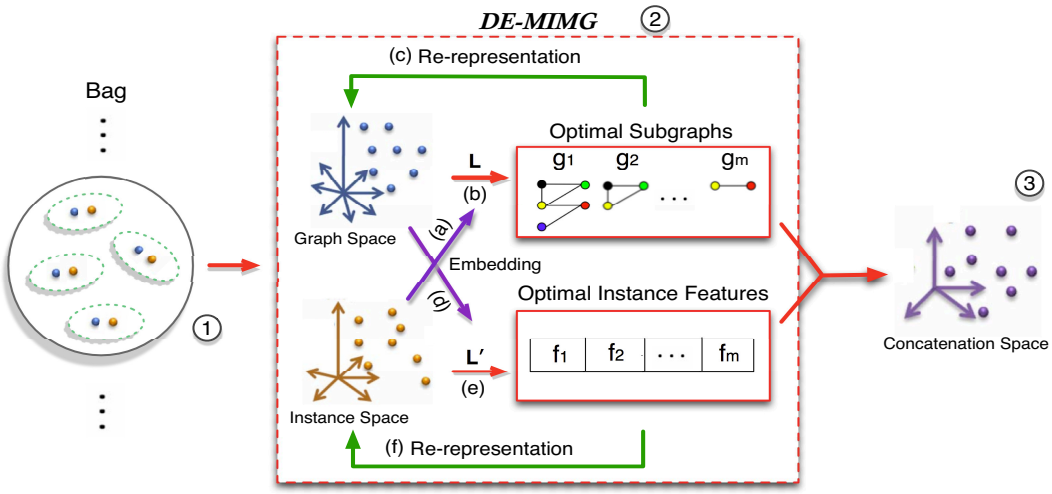

Fig. 3. A conceptual view of DE-MIMG: The overall framework of DE-MIMG is to find optimal representation to convert a bag ① as an instance in the new feature space ③ by using dual embedding strategy ②. The dual embedding consists of two parts: 1) finding subgraph features by embedding instance distributions, and 2) selecting instance features by embedding graph distribution information. The former first builds an instance embedded matrix L (a), which embeds instance distributions to help discover informative subgraphs (b), which are further used to re-represent graphs, as shown in (c). Alternately, the latter builds a graph embedded matrix L' (d) to find an optimal set of instance features (e) to re-represent instances in refined instance space (f). The optimal subgraph features and instance features are concatenated to represent each bag for MIMG learning.

multiple objects in the bag is positive, and negative otherwise. $\boldsymbol{\mathcal{B}} = \{B_1, \cdots, B_p\}$ denotes the set of p bags, and the aggregation of all graphs and all instances in $\boldsymbol{\mathcal{B}}$ is denoted by $\mathcal{G} = \{G_1, \cdots, G_q\}$ and $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_q\}$, respectively, where q is the total number of objects. Similarly, the set of all positive (or negative) bags can be denoted by $\boldsymbol{\mathcal{B}}^+$ (or $\boldsymbol{\mathcal{B}}^-$).

DEFINITION 3 **Subgraph**: Let $G = (\mathcal{V}, E, \mathcal{L}, l)$ and $g = (\mathcal{V}', E', \mathcal{L}', l')$ each denotes a connected graph. g is a subgraph of G, *i.e.*, $g \subseteq G$, iff there exists an injective function $\varphi : \mathcal{V}' \to \mathcal{V}$ s.t. (1) $\forall v \in \mathcal{V}', l'(v) = l(\varphi(v))$; (2) $\forall (u,v) \in E', (\varphi(u), \varphi(v)) \in E$ and $l'(u,v) = l(\varphi(u), \varphi(v))$. If g is a subgraph of G, G is a supergraph of g.

DEFINITION 4 **Subgraph Feature Representation for Graph and Bag**: Let $\mathcal{S}_g = \{g_1, \cdots, g_s\}$ denote a set of subgraphs discovered from a given graph set $\mathcal{G}$. For each graph G_i, we use a subgraph feature vector $\mathbf{h}_i^G = [(h_i^{g_1})^G, \cdots, (h_i^{g_s})^G]^\top$ to represent G_i in graph domain, where $(h_i^{g_k})^G = 1$ iff g_k is a subgraph of G_i (*i.e.* $g_k \subseteq G_i$) and $(h_i^{g_k})^G = 0$ otherwise.

Without considering instances, a bag B_i can be represented by a subgraph feature vector $\mathbf{h}_i^B = [(h_i^{g_1})^B, \cdots, (h_i^{g_s})^B]^\top$, where $(h_i^{g_k})^B = 1$ iff g_k is a subgraph of any graph G in bag B_i (*i.e.* $\exists G \subseteq B_i \wedge G \supseteq g_k$) and $(h_i^{g_k})^B = 0$ otherwise.

DEFINITION 5 **Instance Feature Representation for Bag**: According to multi-instance bag constraint, all instances in a negative bag are negative. So a negative bag $B_i \in \boldsymbol{\mathcal{B}}^-$ can be represented by an instance feature vector $\mathbf{x}_i^B = \sum_{j=1}^{m_i} \mathbf{x}_i^j / m_i$, where m_i denotes the number of instances in B_i and $\mathbf{x}_i^j$ is the jth instance in B_i. For a positive bag $B_i \in \boldsymbol{\mathcal{B}}^+$, the instance with the largest distance from negative *bags* is used to denote the positive bag. So the instance feature vector for a positive bag is $\mathbf{x}_i^B = \arg\min_{\mathbf{x}_i^j \in B_i} \sum_{\mathbf{x}_k^B \in \boldsymbol{\mathcal{B}}^-} exp(-\|\mathbf{x}_i^j - \mathbf{x}_k^B\|^2/t)$.

Given a set of labeled bags $\boldsymbol{\mathcal{B}}$, MIMG learning **aims** to build a prediction model from $\boldsymbol{\mathcal{B}}$ to predict previously unseen bags with maximum accuracy.

III. Overall Framework of MIMG Learning

Figure 3 shows the overall framework of the proposed MIMG learning algorithm with two major steps to solve the key challenges identified in Section I: (1) **Dual Embedding Feature Exploration:** To explore informative subgraphs and instance features, under a bag constrained setting, we employ a dual embedding strategy as follows: *a) Instance embedded subgraph feature exploration:* In graph domain, we embed instance distributions into the objective function to select a set of informative subgraphs to represent each bag B_i in graph space (as shown in steps (a), (b) and (c) in Figure 3); *b) subgraph embedded instance feature exploration:* In instance domain, we embed graph distribution information into the objective function to select important instance features to represent each bag B_i in instance space (as shown in steps (d), (e) and (f) in Figure 3); and (2)**Multi-instance Multi-graph Bag Representation:** Based on the explored optimal subgraphs and instance features, a concatenation strategy is used to map bags into a new mixed feature space. By doing so, each multi-instance multi-graph bag can be converted into one feature vector in the concatenated feature space [16].

After transferring bags into feature vectors, generic supervised learning algorithms, such as decision trees, can be applied for multi-instance multi-graph classification. In the following sections, we first propose our dual embedding feature exploration module, and then discuss detailed algorithm.

IV. Dual Embedding Feature Exploration

Feature (including subgraphs or instance features) exploration for MIMG leaning intends to find a set of most informative features to represent bags. This process has three main challenges: (1) How to utilize bag labels to find informative features? (2) How to tackle label ambiguity in positive bags, where genuine labels of positive graphs or instances are unknown, to find informative features? and (3) How to embed

instance distribution information into the subgraph feature exploration process, and vice versa?

To solve the above challenges, our dual embedding feature exploration framework first embeds instance distributions into the subgraph mining process to find initial subgraphs to represent bags in graph space. Alternatively, we also embed graph distribution information into the instance feature selection process to find good instance features to represent bags in instance space. By repeating the alternatively embedding process, we achieve optimal subgraphs and instance features to represent bags for learning.

Assume a set of graphs $\mathcal{G}$ and the corresponding instance set $\mathcal{X}$ are collected from bag set $\boldsymbol{\mathcal{B}}$. Let $\mathcal{S}_g$ denotes the complete set of subgraphs discovered from $\mathcal{G}$, with $\mathcal{S}_f$ denoting the whole instance feature space in bags. Our dual embedding feature exploration **aims** to find a set of most informative features $\boldsymbol{\mathcal{F}} = \mathbf{g} \cup \mathbf{f}$, ($\mathbf{g} \subseteq \mathcal{S}_g$, $\mathbf{f} \subseteq \mathcal{S}_f$). To this end, we define $\mathcal{J}(\cdot)$ as an evaluation function to measure the informativeness of $\boldsymbol{\mathcal{F}}$. So the objective function of feature exploration, which contains two terms $\mathcal{J}(\mathbf{g})$ and $\mathcal{J}(\mathbf{f})$, is defined in Eq. (1), where $|\cdot|$ represents the cardinality of the set, and m specifies the maximum number selected subgraphs or instance features.

$$\begin{cases} \mathbf{g}^\star = \arg\max\limits_{\mathbf{g}\subseteq\mathcal{S}_g} \Big(\mathcal{J}(\mathbf{g})\Big) & s.t.\ |\mathbf{g}| \le m \\ \mathbf{f}^\star = \arg\max\limits_{\mathbf{f}\subseteq\mathcal{S}_f} \Big(\mathcal{J}(\mathbf{f})\Big) & s.t.\ |\mathbf{f}| \le m \end{cases} \tag{1}$$

A. Embedding Feature Evaluation Criteria

Feature Evaluation Criteria(FEC): In order to maximize the evaluation $\mathcal{J}(\mathbf{g})$ or $\mathcal{J}(\mathbf{f})$ of a feature set, we impose constraints to the bag and object (*i.e.* instance or graph) levels as follows: (a) bag level *must-link:* For any two bags B_i and B_j, if they have the same labels, we form a pairwise *must-link* constraint between them. Because each bag is associated with a known class label (positive or negative), the features should ensure that bags with the same label have high similarity with each other; (b) bag level *cannot-link:* If B_i and B_j have different labels, we form a *cannot-link* constraint between them; (c) object level *must-link:* In MIMG scenarios, only objects in negative bags are genuinely negative. The selected features should ensure that objects in each negative bags are similar to each other so they can share certain commonality of being negative; (d) object level *separability:* for all objects in positive bags, their genuine labels are unknown, although at least one of them must be positive. In this case, we use Principle Component Analysis (PCA) principle [17] and seek to find features which preserve the diverse information in the positive bags (*i.e.* objects in positive bags are maximally separable). Intuitively, (a) and (b) only consider constrains from labeled bags, and tend to select the optimal features based on bag labels. They are similar to the Linear Discriminant Analysis (LDA) [18] criterion. To further take the data distributions inside each bag into consideration, we also add object level constraints (c) and (d) to ensure that features can make objects in each negative bag close to each other, and maximally separated in each positive bag.

Dual Embedding Strategy: To take full advantage of the complementary information between graphs and instances in each bag, we propose a dual embedding strategy, as shown in part ② of Figure 3. In summary, the embedding process is to utilize instance distributions to help find informative subgraph features to represent graphs (*i.e.* the instance embedding, a conceptual view of which is shown in Figure 4), and further use the graph distribution information to help find optimal instance features (*i.e.* the graph embedding). Such a dual embedding process is carried out in an iterative way, until the whole process converges. As a result, it will produce optimal subgraph features and instance features to represent each bag.

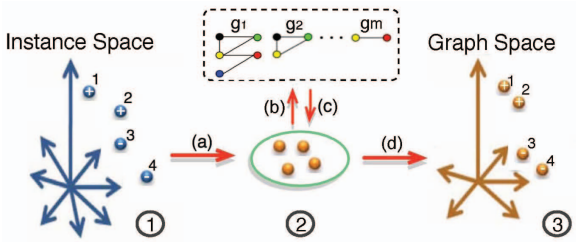

Fig. 4. The instance embedding is to help mine optimal subgraphs ② to represent graphs ③ under the FEC rule (*i.e.* objects with the same label are close to each other, and are separated from objects with different labels.) and also keep the distribution information in instance domain ①. The process starts from (a) using instance distributions to discover optimal subgraphs (b), and further represent graphs in graph space (d).

B. Embedding FEC for Subgraph Feature Exploration

1) ***Optimization Framework****:* The objective function $\mathbf{g}^\star$ in Eq. (1) indicates that the optimal subgraph features $\mathbf{g}^\star$ should have maximum discriminative power, *i.e.*, $\max(\mathcal{J}(\mathbf{g}))$, among all alternative subgraph sets with the same size. In order to derive solutions to find such subgraph features, we first formally introduce notations as follows:

- $\mathcal{H}_B$: the matrix consisting of binary subgraph feature vectors using S_g to represent all bags in $\boldsymbol{\mathcal{B}}$. $\mathcal{H}_B = [\mathbf{h}_1^B, \cdots, \mathbf{h}_p^B] = [\boldsymbol{f}_{g_1}^B, \cdots, \boldsymbol{f}_{g_s}^B]^\top \in \{0,1\}^{s\times p}$, where $\boldsymbol{f}_{g_k}^B$, $g_k \in S_g$, is an indicator vector of subgraph g_k with respect to all bags B_i in $\boldsymbol{\mathcal{B}}$, *i.e.*, $\boldsymbol{f}_{g_k}^B = [f_{g_k}^{B_1}, \cdots, f_{g_k}^{B_p}]^\top$, where $f_{g_k}^{B_i} = 1$ iff $\exists G \subseteq B_i \wedge G \supseteq g_k$ and $f_{g_k}^{B_i} = 0$ otherwise.
- $\mathcal{H}_G$: the matrix using subgraphs to represent all graphs in bags based on S_g, $\mathcal{H}_G = [\mathbf{h}_1^G, \cdots, \mathbf{h}_q^G] = [\boldsymbol{f}_{g_1}^G, \cdots, \boldsymbol{f}_{g_s}^G]^\top \in \{0,1\}^{s\times q}$, where $\boldsymbol{f}_{g_k}^G$, $g_k \in S_g$, is an indicator vector of subgraph g_k with respect to all graphs in $\boldsymbol{\mathcal{B}}$, *i.e.*, $\boldsymbol{f}_{g_k}^G = [f_{g_k}^{G_1}, \cdots, f_{g_k}^{G_q}]^\top$, where $f_{g_k}^{G_i} = 1$ iff $g_k \subseteq G_i$ and $f_{g_k}^{G_i} = 0$ otherwise.
- A, B, C and D: $A = \{(i,j)|y_i y_j = 1\}$ denotes the *must-link* pairwise bag constraint sets with $B = \{(i,j)|y_i y_j = -1\}$ denoting the *cannot-link* pairwise sets. $C = \{(G_i, G_j)|G_i, G_j \in \boldsymbol{\mathcal{B}}^-\}$ denotes the *musk-link* pairwise constraint in negative bag set $\boldsymbol{\mathcal{B}}^-$, with $D = \{(G_i, G_j)|G_i, G_j \in \boldsymbol{\mathcal{B}}^+\}$ denoting the *separability* pairwise constraint in positive bag set $\boldsymbol{\mathcal{B}}^+$.

Based on the above constraints, we derive an embedding criterion $\mathcal{J}(\mathbf{g})$, which consists of bag level $\mathcal{J}(\mathbf{g})^B$ and graph level $\mathcal{J}(\mathbf{g})^G$ as follows:

$$\begin{aligned} \mathcal{J}(\mathbf{g}) = \mathcal{J}(\mathbf{g})^B + \mathcal{J}(\mathbf{g})^G = \frac{1}{2}\sum\nolimits_{i,j} K_B\Big(\mathcal{D}_{\mathbf{g}}\mathbf{h}_i^B, \mathcal{D}_{\mathbf{g}}\mathbf{h}_j^B\Big) M_{i,j}^B \\ + \frac{1}{2}\sum\nolimits_{i,j} K_G\Big(\mathcal{D}_{\mathbf{g}}\mathbf{h}_i^G, \mathcal{D}_{\mathbf{g}}\mathbf{h}_j^G\Big) M_{i,j}^G \end{aligned} \tag{2}$$

For bag level, we use $M^B_{i,j}$ to embed instance distribution information between two bags in instance space $\mathbf{x}^B_i$ and $\mathbf{x}^B_j$ (so instance distributions is embedded in the subgraph feature discovery process). $K_B(\mathcal{D}_\mathbf{g}\mathbf{h}^B_i, \mathcal{D}_\mathbf{g}\mathbf{h}^B_j)$ (K_B for abbreviation), which is defined in Eq. (3), denotes distance between two bags B_i and B_j in subgraph feature space corresponding to the bag level criteria FEC (a) and (b) in Section IV-A.

At graph (or instance) level, we use $M^G_{i,j}$ to embed instances $\mathbf{x}_i$ and $\mathbf{x}_j$ in bags into the objective function for subgraph mining. $K_G(\mathcal{D}_\mathbf{g}\mathbf{h}^G_i, \mathcal{D}_\mathbf{g}\mathbf{h}^G_j)$ (K_G for abbreviation), which is defined in Eq. (4), represents distance between two graphs G_i and G_j corresponding to the graph level criteria FEC (c) and (d) in Section IV-A. $\mathcal{D}_\mathbf{g} = diag(d(\mathbf{g}))$ is a diagonal matrix indicating which subgraph features $\mathbf{g}$ are selected from S_g to represent the bags or graphs, $d(\mathbf{g})_i = I(g_i \in \mathbf{g})$. To calculate $M^B_{i,j}$ and $M^G_{i,j}$, we adopt a radial basis function to measure $M^B_{i,j} = exp(-\|\mathbf{x}^B_i - \mathbf{x}^B_j\|^2/t)$ and $M^G_{i,j} = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t)$.

$$K_B = \begin{cases} -\|\mathcal{D}_\mathbf{g}\mathbf{h}^B_i - \mathcal{D}_\mathbf{g}\mathbf{h}^B_j\|^2, y_i y_j = 1 \\ \|\mathcal{D}_\mathbf{g}\mathbf{h}^B_i - \mathcal{D}_\mathbf{g}\mathbf{h}^B_j\|^2, y_i y_j = -1 \end{cases} \tag{3}$$

$$K_G = \begin{cases} -\|\mathcal{D}_\mathbf{g}\mathbf{h}^G_i - \mathcal{D}_\mathbf{g}\mathbf{h}^G_j\|^2, \forall G_i, G_j \in \boldsymbol{\mathcal{B}}^- \\ \|\mathcal{D}_\mathbf{g}\mathbf{h}^G_i - \mathcal{D}_\mathbf{g}\mathbf{h}^G_j\|^2, \forall G_i, G_j \in \boldsymbol{\mathcal{B}}^+ \end{cases} \tag{4}$$

By using a bag level instance embedded matrix $W_B = [W^B_{ij}]^{p\times p}$ and a graph level instance embedded matrix $W_G = [W^G_{ij}]^{q\times q}$. Eq. (2) can be rewritten as follows,

$$\begin{aligned} \mathcal{J}(\mathbf{g}) = &\frac{1}{2}\sum_{i,j}\|\mathcal{D}_\mathbf{g}\mathbf{h}^B_i - \mathcal{D}_\mathbf{g}\mathbf{h}^B_j\|^2 W^B_{i,j} \\ &+ \frac{1}{2}\sum_{i,j}\|\mathcal{D}_\mathbf{g}\mathbf{h}^G_i - \mathcal{D}_\mathbf{g}\mathbf{h}^G_j\|^2 W^G_{i,j} \end{aligned} \tag{5}$$

where $W^B_{ij} = \{-M^B_{i,j}/|A|, y_i y_j = 1;\ M^B_{i,j}/|B|, y_i y_j = -1; 0, otherwise\}$, and $W^G_{ij} = \{-M^G_{i,j}/|C|, \forall G_i, G_j \in \boldsymbol{\mathcal{B}}^-; M^G_{i,j}/|D|, \forall G_i, G_j \in \boldsymbol{\mathcal{B}}^+; 0, otherwise\}$.

Through the above process, the distribution information in instance domain is embedded in the two matrices W_B for bag level and W_G for instance level (analogous to graph level). For bag level evaluation criterion $\mathcal{J}(\mathbf{g})^B$, We have

$$\begin{aligned} \mathcal{J}(\mathbf{g})^B &= \frac{1}{2}\sum\nolimits_{i,j}\|D_\mathbf{g}\mathbf{h}^B_i - D_\mathbf{g}\mathbf{h}^B_j\|^2 W^B_{i,j} \\ &= tr(\mathcal{D}^\top_\mathbf{g}\mathcal{H}_B(D_B - W_B)\mathcal{H}^\top_B\mathcal{D}_\mathbf{g}) \\ &= tr(\mathcal{D}^\top_\mathbf{g}\mathcal{H}_B L_B \mathcal{H}^\top_B\mathcal{D}_\mathbf{g}) \end{aligned} \tag{6}$$

where $tr(|\cdot|)$ is the trace operator for a matrix, D_B is a diagonal matrix generated from W_B, *i.e.,* $D^B_{i,i} = \sum_j W^B_{ij}$. $L_B = [L^B_{i,j}]^{p\times p} = D_B - W_B$ is a Laplacian matrix.

By combining bag level $\mathcal{J}(\mathbf{g})^B$ and graph level evaluation criterion $\mathcal{J}(\mathbf{g})^G$, which shares the same derivation with $\mathcal{J}(\mathbf{g})^B$. Eq.(5) can be rewritten as follows,

$$\begin{aligned} \mathcal{J}(\mathbf{g}) &= tr(\mathcal{D}^\top_\mathbf{g}\mathcal{H}_B L_B \mathcal{H}^\top_B\mathcal{D}_\mathbf{g}) + tr(\mathcal{D}^\top_\mathbf{g}\mathcal{H}_G L_G \mathcal{H}^\top_G\mathcal{D}_\mathbf{g}) \\ &= tr\Big(\mathcal{D}^\top_\mathbf{g}(\mathcal{H}_B L_B \mathcal{H}^\top_B + \mathcal{H}_G L_G \mathcal{H}^\top_G)\mathcal{D}_\mathbf{g}\Big) \\ &= tr\left(\mathcal{D}^\top_\mathbf{g}\begin{bmatrix}\mathcal{H}_B & \mathcal{H}_G\end{bmatrix}\begin{bmatrix}L_B & 0 \\ 0 & L_G\end{bmatrix}\begin{bmatrix}\mathcal{H}^\top_B \\ \mathcal{H}^\top_G\end{bmatrix}\mathcal{D}_\mathbf{g}\right) \\ &= tr(\mathcal{D}^\top_\mathbf{g}\mathcal{H}L\mathcal{H}^\top\mathcal{D}_\mathbf{g}) = \sum\nolimits_{g_k\in\mathbf{g}} \boldsymbol{f}^\top_{g_k} L \boldsymbol{f}_{g_k} \end{aligned} \tag{7}$$

where $L_G = [L^G_{i,j}]^{q\times q} = D_G - W_G$ is known as a Laplacian matrix, which encodes the instance embedded objective function at the graph level. According to Eq. (7), it is

$$\mathcal{H} = \begin{bmatrix}\mathcal{H}_\mathcal{B} & \mathcal{H}_\mathcal{G}\end{bmatrix}, L = \begin{bmatrix}L_B & 0 \\ 0 & L_G\end{bmatrix}, \boldsymbol{f}_{g_k} = \begin{bmatrix}\boldsymbol{f}^B_{g_k} \\ \boldsymbol{f}^G_{g_k}\end{bmatrix}, W = \begin{bmatrix}W_B & 0 \\ 0 & W_G\end{bmatrix} \tag{8}$$

where $\boldsymbol{f}_{g_k}$ is an indicator vector of subgraph g_k with respect to the data combined with bags and graphs $\mathcal{H}$. By denoting the function as $z(g_k, L) = \boldsymbol{f}^\top_{g_k} L \boldsymbol{f}_{g_k}$, the problem of maximizing $\mathcal{J}(\mathbf{g})$ in Eq. (1) is equal to find a subset of subgraphs that can maximize the sum of $z(g_k, L)$, which can be represented as:

$$\mathbf{g}^\star = \max_\mathbf{g} \sum\nolimits_{g_k\in\mathbf{g}} z(g_k, L) \qquad s.t.\ \ \mathbf{g}\subseteq S_g, |\mathbf{g}| \le m. \tag{9}$$

DEFINITION 6 **eScore:** Given a multi-instance multi-graph dataset and two instance embedded matrices W_B and W_G. Assume L denotes a Laplacian matrix defined as $L = D - W$, where W is the matrix composed of W_B and W_G in Eq. (8) and D is a diagonal matrix, $D_{i,i} = \sum_j W_{i,j}$, the informativeness score of a subgraph g_k can be measured by:

$$r(g_k) = z(g_k, L) = \boldsymbol{f}^\top_{g_k} L \boldsymbol{f}_{g_k} \tag{10}$$

Since the Laplacian matrix L is positive semi-definite [19], for any subgraph g_k, $\boldsymbol{f}^\top_{g_k} L \boldsymbol{f}_{g_k} \ge 0$, *i.e.,* $r(g_k) \ge 0$. To find subgraph set $\mathbf{g}$ which maximizes the criterion $\mathcal{J}(\mathbf{g})$ defined in Eq. (7), we can calculate eScore of each subgraph in $\mathcal{S}_g$ and sort them, according to their eScore values, in a descending order, *i.e.*, $r(g_1) \ge r(g_2) \cdots \ge r(g_s)$. Then by using the top-$m$ features $\mathbf{g} = \{g_1, \cdots, g_m\}$, we can maximize $\mathcal{J}(\mathbf{g})$.

*2) **Subgraph Exploration**:* One of the most straightforward solutions for finding an optimal subgraph set is exhaustive enumeration, *i.e.*, all subgraphs in a graph dataset will be enumerated with their eScore values being calculated for ranking. However, the number of subgraphs grows exponentially with the size of graphs in bags, which makes the exhaustive enumeration approach impractical for real-world data. Alternatively, we employ a Depth-First-Search (DFS) based algorithm gSpan [20] to iteratively enumerate subgraphs. The key idea of gSpan is to first assign a unique minimum DFS code to each graph, and then discover all frequent subgraphs by a preorder traversal of the tree. Some recent graph classification approaches [19], [21] incorporate constraints to prune the search space of gSpan. In this paper we derive a upper bound for using eScore to prune the search space in the DFS code tree, which could be defined as follows:

THEOREM 1 **Upper bound of eScore:** Given two subgraphs g_k, $g_k{}' \in S_g$, g'_k is a supergraph of g_k (*i.e.* $g'_k \supseteq g_k$). The eScore value g'_k ($r(g'_k)$) is bounded by $\hat{r}(g_k)$, *i.e.,* $r(g'_k) \le \hat{r}(g_k)$, where $\hat{r}(g_k)$ is defined as follows:

$$\hat{r}(g'_k) \triangleq \boldsymbol{f}^\top_{g_k} \hat{L} \boldsymbol{f}_{g_k} \tag{11}$$

where the instance embedded matrix $\hat{L}$ is defined as $\hat{L}_{ij} \triangleq max(0, L_{ij})$.

Proof:

$$\begin{aligned} r(g'_k) =& \boldsymbol{f}^\top_{g'_k} L \boldsymbol{f}_{g'_k} = \begin{bmatrix}(\boldsymbol{f}^B_{g'_k})^\top & (\boldsymbol{f}^B_{g'_k})^\top\end{bmatrix}\begin{bmatrix}L_B & 0 \\ 0 & L_G\end{bmatrix}\begin{bmatrix}\boldsymbol{f}^B_{g'_k} \\ \boldsymbol{f}^G_{g'_k}\end{bmatrix} \\ =& (\boldsymbol{f}^B_{g'_k})^\top L_B \boldsymbol{f}^B_{g'_k} + (\boldsymbol{f}^G_{g'_k})^\top L_G \boldsymbol{f}^G_{g'_k} \\ =& \sum_{i,j:B_i,B_j\in\mathcal{B}(g'_k)} L^B_{ij} + \sum_{i,j:G_i,G_j\in\mathcal{G}(g'_k)} L^G_{ij} \end{aligned} \tag{12}$$

Algorithm 1 ESFE: Embedding Subgraph Feature Exploration

Input:
$\mathcal{G}$: A graph dataset;
$\mathcal{X}$: An instance dataset;
min_sup: The threshold of the frequent subgraph;
m: the number of subgraph features to be selected;
Output:
$\mathbf{g} = \{g_1, \cdots, g_m\}$: A set of subgraph features;
1: $\mathbf{g} = \emptyset$, $\tau = 0$;
2: **while** Recursively visit the DFS Code Tree in gSpan **do**
3: $g_k \leftarrow$ current visited subgraph in DFS code tree of $\mathcal{G}$;
4: **if** $freq(g_k) < min\_sup$, **then**
5: **return**;
6: $L \leftarrow$ Apply $\mathcal{G}$ and $\mathcal{X}$ to obtain the embedding matrix;
7: $r(g_k) \leftarrow$ Apply L to compute the eScore of subgraph g_k;
8: **if** $|\mathbf{g}| < m$ or $r(g_k) > \tau$, **then**
9: $\mathbf{g} \leftarrow \mathbf{g} \bigcup g_k$;
10: **if** $|\mathbf{g}| \geq m$, **then**
11: $\mathbf{g} \leftarrow \mathbf{g}/\arg\min_{g_i \in \mathbf{g}} r(g_i)$;
12: $\tau = \min_{g_i \in \mathbf{g}} r(g_i)$;
13: **if** $\hat{r}(g_k) \geq \tau$, **then**
14: Depth-first search the subtree rooted from node g_k;
15: **end while**
16: **return** **g**;

where $\mathcal{B}(g'_k) \triangleq \{B_i | g'_k \subseteq G_j \in B_i, 1 \leq i \leq p\}$ and $\mathcal{G}(g'_k) \triangleq \{G_j | g'_k \subseteq G_j, 1 \leq j \leq q\}$. Since g'_k is the supergraph of g_k (*i.e.* $g'_k \supseteq g_k$), according to the anti-monotonic property, we have $\mathcal{B}(g'_k) \subseteq \mathcal{B}(g_k)$ and $\mathcal{G}(g'_k) \subseteq \mathcal{G}(g_k)$. Besides, $\hat{L}^B_{ij} \triangleq max(0, L^B_{ij})$ and $\hat{L}^G_{ij} \triangleq max(0, L^G_{ij})$, so $\hat{L}^B_{ij} \geq L^B_{ij}$ and $\hat{L}^G_{ij} \geq L^G_{ij}$. Both $\hat{L}^B_{ij}$ and $\hat{L}^G_{ij}$ are great than or equal to zero. Thus, Eq. 12 could be rewritten as

$$\begin{aligned} r(g'_k) &= \sum_{i,j:B_i,B_j \in \mathcal{B}(g'_k)} L^B_{ij} + \sum_{i,j:G_i,G_j \in \mathcal{G}(g'_k)} L^G_{ij} \\ &\leq \sum_{i,j:B_i,B_j \in \mathcal{B}(g'_k)} \hat{L}^B_{ij} + \sum_{i,j:G_i,G_j \in \mathcal{G}(g'_k)} \hat{L}^G_{ij} \\ &\leq \sum_{i,j:B_i,B_j \in \mathcal{B}(g_k)} \hat{L}^B_{ij} + \sum_{i,j:G_i,G_j \in \mathcal{G}(g_k)} \hat{L}^G_{ij} \\ &\leq \boldsymbol{f}^\top_{g_k} \hat{L} \boldsymbol{f}_{g_k} = \hat{r}(g_k) \end{aligned} \tag{13}$$

Thus, for any $g'_k \supseteq g_k$, $r(g'_k) \leq \hat{r}(g_k)$. ■

The above upper bound can be utilized to prune DFS code tree in gSpan by using branch-and-bound pruning. Algorithm 1 lists the proposed embedding subgraph feature exploration method, which starts from an empty feature set **g** and a minimum eScore $\tau = 0$. The algorithm continuously enumerates subgraphs by recursively visiting the DFS code tree in gSpan. If a subgraph g_k is not a frequent subgraph, both g_k and its subtree will be pruned (line 4-5). Otherwise, we calculate g_k's eScore value $r(g_k)$ using the embedding matrix L, which embeds the instance distribution information. If $r(g_k)$ is larger than τ which is the minimum eScore of the current set **g**, or **g** has less than m subgraphs (*i.e.* **g** is not full), g_k is added to the subgraph set **g** (lines 8-9). If the size of **g** exceeds the predefined value m, we need to remove one subgraph with the least discriminative power (lines 10-11). After that, the upper bound pruning module will check if $\hat{r}(g_k)$ is less than the threshold τ, if so, it means that the eScore value of any supergraph g'_k of g_k (*i.e.* $g'_k \supseteq g_k$) will not be greater than τ. Therefore we can safely prune subtrees rooted from g_k in the search space. If $\hat{r}(g_k)$ is indeed greater than the threshold τ, the depth-first search will continue by following the children of g_k (line 13-14), until the subgraph mining process is completed.

Algorithm 2 EIFE: Embedding Instance Feature Exploration

Input:
$\mathcal{X}$: An instance dataset;
$\mathcal{G}$: A graph dataset;
m: the number of features to be selected;
Output:
$\mathbf{f} = \{f_1, \cdots, f_m\}$: A set of features;
1: $\mathbf{g} = \emptyset$, $\rho = 0$;
2: **for** each feature f_k in **f** **do**
3: $L' \leftarrow$ Apply $\mathcal{X}$ and $\mathcal{G}$ to obtain the embedding matrix;
4: $r(f_k) \leftarrow$ Apply L' to compute the eScore of feature f_k;
5: **if** $|\mathbf{f}| < m$ or $r(f_k) > \rho$, **then**
6: $\mathbf{f} \leftarrow \mathbf{f} \bigcup f_k$;
7: **if** $|\mathbf{f}| \geq m$, **then**
8: $\mathbf{f} \leftarrow \mathbf{f}/\arg\min_{f_i \in \mathbf{f}} r(f_i)$;
9: $\rho = \min_{f_i \in \mathbf{f}} r(f_i)$;
10: **end for**
11: **return** **f**;

C. Embedding FEC for Instance Feature Selection

In this section, we focus on instance feature selection by embedding graph distribution information into the object function. Because in MIMG setting, each bag consists of an instance-graph pair with complementary information from instance and graph domains, we use the graph embedded informativeness score to evaluate instance feature as follows:

$$\begin{aligned} \mathcal{J}(\mathbf{f}) = &\frac{1}{2} \sum_{i,j} K_B \left(\mathcal{D}_{\mathbf{f}} \mathbf{x}^B_i, \mathcal{D}_{\mathbf{f}} \mathbf{x}^B_j \right) Q^B_{i,j} \\ &+ \frac{1}{2} \sum_{i,j} K_G \left(\mathcal{D}_{\mathbf{f}} \mathbf{x}_i, \mathcal{D}_{\mathbf{f}} \mathbf{x}_j \right) Q^G_{i,j} \end{aligned} \tag{14}$$

where $\mathcal{J}(\mathbf{f})$ is an evaluation function, which is similar to $\mathcal{J}(\mathbf{g})$, to estimate the informativeness of feature set **f**, $\mathcal{D}_{\mathbf{f}} = diag(d(\mathbf{f})), d(\mathbf{f})_i = I(f_i \in \mathbf{f})$. $Q^B_{i,j}$ denotes the distance between two bags B_i and B_j in graph space, and $Q^G_{i,j}$ represents the distance between two graphs G_i and G_j by using selected subgraph features.

Using the same derivation as $\mathcal{J}(\mathbf{g})$, the related informativeness score of a feature f_k can be represented as:

$$r(f_k) = \boldsymbol{f}^\top_k L' \boldsymbol{f}_k \tag{15}$$

where L' encodes the distribution information in graph domain, and $\boldsymbol{f}_k$ is an indicator vector the same with $\boldsymbol{f}_{g_k}$. By choosing the top-m features with high $r(f_k)$ values, $\mathcal{J}(\mathbf{f})$ can be maximized. The detailed process of embedding instance feature selection is summarised in Algorithm 2.

D. Feature Concatenation

After the exploration of the optimal subgraphs and instance features, we employ a concatenation strategy to combine subgraphs and instance features to represent each bag. In this process, each bag is converted into a vector space by using selected optimal subgraphs and instance features. More specifically, we use subgraph feature representation, as shown in Definition 4, to convert each bag into one instance. The same bag in instance domain is also converted into one instance by using selected instance features, per Definition 5. After that, the subgraph features and instance features of the bag are concatenated as one feature domain to represent the bag.

Algorithm 3 DE-MIMG: MIMG Dual Embedding Learning

Input:

$\boldsymbol{\mathcal{B}} = \{B_1^{(I)}, \cdots, B_p^{(I)}, B_1^{(G)}, \cdots, B_p^{(G)}\}$: a bag set;

m: The number of subgraphs or instance features to be selected;

min_sup: The minimum support threshold;

Output:

The target class label y_t of a test bag B_t based on a set of selected concatenate features $\boldsymbol{\mathcal{F}} = \{\mathbf{g}, \mathbf{f}\} = \{g_1, \cdots, g_m, f_1, \cdots, f_m\}$;

// Training Phase:

1: $\mathcal{X}, \mathcal{G} \leftarrow$ Apply the *MIMG* data $\boldsymbol{\mathcal{B}}$ to obtain the instance data and graph data, respectively.

2: **while** No changes for L in the previous iteration **do**

// Update instance embedded subgraphs:

3: $\mathbf{g} \leftarrow ESFE(m, min\_sup, \mathcal{G}, \mathcal{X})$; //Algorithm 1

4: $\mathcal{G} \leftarrow$ Apply the subgraphs $\mathbf{g}$ to represent the graphs.

// Update graph embedded instance features:

5: $\mathbf{f} \leftarrow EIFE(m, \mathcal{X}, \mathcal{G})$; //Algorithm 2

6: $\mathcal{X} \leftarrow$ Apply the instance features $\mathbf{f}$ to represent the instances.

7: **end while**

8: $\boldsymbol{\mathcal{F}} \leftarrow$ Apply the selected optimal subgraphs $\mathbf{g}$ and instance features $\mathbf{f}$ to obtain the concatenated features.

9: $\mathcal{X}^\star \leftarrow$ Apply the concatenated feature set $\boldsymbol{\mathcal{F}}$ to obtain bag constrained *MIMG* training data;

10: $\psi \leftarrow$ a classifier built from $\mathcal{X}^\star$;

// Test Phase:

11: $\mathbf{x}_t^\star \leftarrow$ bag constrained feature representation for test bag B_t;

12: $y_t \leftarrow h(\mathbf{x}_t^\star | \psi)$;

13: **return** y_t;

V. DE-MIMG: Multi-Instance Multi-Graph Dual Embedding Learning

Algorithm 3 lists detailed procedures of the proposed MIMG framework with interactive embedding between instance and graph domains. At the very first step, the algorithm finds a set of subgraphs to represent the bag and graphs in graph domain by a depth-first search using Algorithm 1 (line 3), because the initial graphs have no feature representation. It is cleat that the quality of initial selected subgraphs is not optimal because the embedded instance distributions are based on the whole feature space, which may contain a significant amount of noise and redundancy. Nevertheless, it provides a good starting point for MIMG dual embedding learning.

With the help of initial subgraph features, we can obtain optimal instance features in instance domain under same embedding optimal framework utilizing Algorithm 2 (line 5). By using embedding learning in an iterative way, we can obtain the final optimal subgraphs and instance features in graph and instance domain respectively, until the distributions in both domains are stable. Through the concatenation of subgraphs and instance features, we can obtain a mixed feature set $\boldsymbol{\mathcal{F}}$, which is utilized to represent the bags. As a result, the original bags are represented as bag constrained mixed features (lines 8-9), which help train a classifier ψ (line 10). At the test phase, a test bag B_t is transferred into a feature vector by using $\boldsymbol{\mathcal{F}}$, and is predicted by the classifier ψ to obtain its class label y_t (lines 11-12).

The major advantage of DE-MIMG is threefold:(1) the dual embedding of instance and graph information provides effective way to leverage two complementary domains (graphs and instances), in a bag constrained environment, to achieve a common learning goal; (2) the bag and graph (or instance) level constraints fully utilize the subgraphs or instance features to find a set of most informative features to represent bags. (3) the upper bound of eScore in graph domain helps prune subgraph search space.

VI. Experiments

In this section, we report experiments on two real-world applications to validate the effectiveness of DE-MIMG for multi-instance multi-graph learning.

A. DataSets

Online Product Review Dataset is downloaded from Stanford Large Network Dataset Collection [1]. The food review dataset *Fine Foods* contains numerous food related reviews from Amazon. Each review is associated with some attributes such as product ID, reviewer ID, review score (rating of the product varying from 1 to 5), review summary, and detailed text descriptions [22]. For each product, if the average score over all reviews is great or equal to 4, we believe that one or multiple key properties (*e.g.* "affordability" and "durability") of the product should receive very positive reviews. On the contrary, a customer may be not interested in a product, if all of the review scores are less than 4. By doing so, we can recommend good products to users based on the review summary and text descriptions. For review text, we use fuzzy cognitive map (E-FCM) [23] to form a graph representation with each node denoting one keyword and edges representing correlations between keywords. To reduce edge density for each graph, all edges whose correlation values less than a certain threshold (0.006) are discarded. We choose 400 food products, each of which containing 1 to 10 reviews, to form 200 positive (average score $\geq$ 4) bags (with 1151 pairs of instances and graphs in all positive bags) and 200 negative (score$<$ 4) bags (with 1106 pairs of instances and graphs in all negative bags).

DBLP Dataset. The DBLP dataset contains bibliography data in computer science [2]. Each record in DBLP is composed of a number of attributes such as abstract, authors, year, venue, title, and references [24]. To build bags, we select papers published in Artificial Intelligence (AI: IJCAI, AAAI ,NIPS, UAI, COLT, ACL, KR, ICML, ECML and IJCNN) and Computer Vision (CV: ICCV, CVPR, ECCV, ICPR, ICIP, ACM Multimedia and ICME) fields to form a MIMG learning task. The goal is to predict which field (AI or CV) a paper belongs to by using the title (converted to instance representation) and abstract (converted to graph representation) of each paper and references cited in the paper. So each paper is a bag and each instance inside the bag denotes either the paper's title or the title of a reference cited in the paper, and each graph inside the bag denotes either the paper's abstract or the abstract of a reference cited in the paper. The graph representation for abstract is similar to the above online product review dataset. Notice that AI and CV are overlapped in many aspects, such as machine learning, optimization and data mining, which makes a challenging MIMG task. The original DBLP dataset contains a significant number of papers without any references. We choose 400 papers, each of which containing 1 to 10 references, to form the corresponding multi-instance multi-graph datasets with positive (AI) bags with 1151 pairs of

[1] http://snap.stanford.edu/data/.

[2] http://dblp.uni-trier.de/xml/.

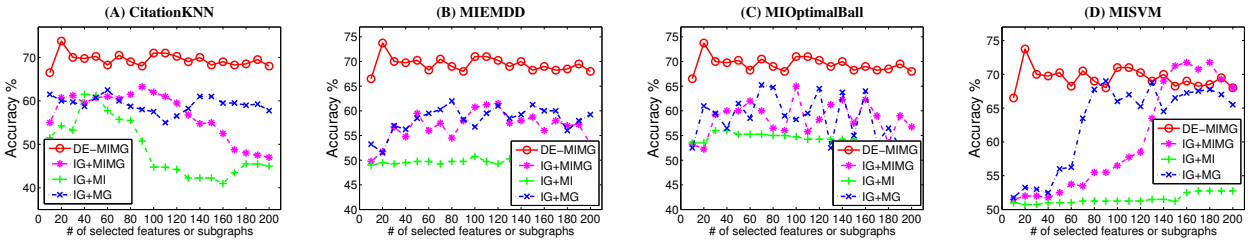

Fig. 5. Accuracy comparisons on **Online Product Review dataset** by using proposed dual embedding algorithm DE-MIMG and IG based baselines on separated multi-instance setting IG+MI, separated multi-graph setting IG+MG and multi-instance multi-graph learning IG+MIMG on generic multi-instance (MI) learning methods: (A) CitationKNN; (B) MIEMDD; (C) MIOptimallBall; and (D) MISVM.

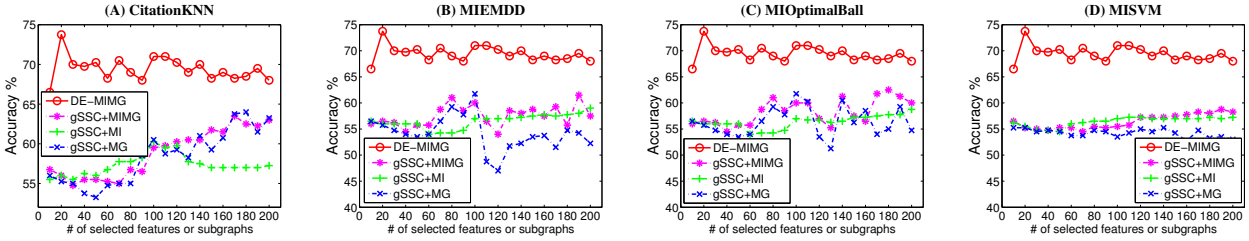

Fig. 6. Accuracy comparisons on **Online Product Review dataset** by using proposed dual embedding algorithm DE-MIMG and gSSC based baselines on separated multi-instance setting gSSC+MI, separated multi-graph setting gSSC+MG and multi-instance multi-graph learning gSSC+MIMG on generic multi-instance (MI) learning methods: (A) CitationKNN; (B) MIEMDD; (C) MIOptimallBall; and (D) MISVM.

TABLE I. PAIRWISE t-TEST RESULT ON ONLINE PRODUCT REVIEW DATASET. A DENOTES DE-MIMG, WITH B, C AND D DENOTING MI, MG AND MIMG SETTING. $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ *and* $\mathcal{H}_4$ DENOTE CITATIONKNN, MIEMDD, MIOPTIMALBALL AND MISVM, RESPECTIVELY.

	IG				gSSC		
	A-B	A-C	A-D		A-B	A-C	A-D
$\mathcal{H}_1$	2.7E-11	4.9E-13	6.5E-10	$\mathcal{H}_1$	1.8E-17	2.3E-10	1.2E-10
$\mathcal{H}_2$	3.9E-20	6.8E-12	1.5E-13	$\mathcal{H}_2$	1.7E-16	1.4E-13	3.0E-14
$\mathcal{H}_3$	1.5E-18	2.1E-10	3.0E-11	$\mathcal{H}_3$	7.6E-17	3.8E-14	10.0E-13
$\mathcal{H}_4$	3.7E-20	3.0E-4	1.5E-4	$\mathcal{H}_4$	2.7E-17	2.5E-20	1.2E-15

instances and graphs, and negative (CV) bags with 1150 pairs of instances and graphs.

B. Baseline Methods

For comparison purposes, we implement following baseline approaches, which are designed from two perspectives. In the first set of methods (Instance feature selection based approaches), we convert graphs in each bag into instances, by using frequent subgraphs, and then use traditional feature selection methods to select a number of features to represent each bag for learning. In the second set of methods (Graph feature selection based approaches), we employ a state-of-the-art method to select informative subgraphs to represent bags for learning. Our main objective is to validate whether the proposed dual embedding approach, which takes graphs and instances into consideration, is superior to any alternative which only focuses on individual instance or graph domain or a simple instance and graph concatenation strategy.

- **Instance feature selection based approaches (IG+MI, IG+MG, and IG+MIMG)**. In this set of methods, we use frequent subgraph mining to discover a set of frequent subgraphs from all bags to represent each graph as a graph-view instance. An IG (Information Gain) [25] based feature selection is further used to select m subgraphs and m instance features with the highest IG scores. After that, the learning is carried out by using information from different views: IG based multi-instance setting (IG+MI) applies existing MI learning methods to the selected m instance features. IG based multi-graph setting (IG+MG) uses m subgraphs to transfer each graph into one a graph-view instance, so a bag of graphs is converted into a bag of instances, through which the MI learning methods can be applied for learning. In addition, we also concatenate discovered subgraphs and instance features as one view and use MI learning approaches for learning (denoted by IG+MIMG).
- **Graph feature selection based approaches (gSSC+MI, gSSC+MG, and gSSC+MIMG**. In this set of methods, we use a recent semi-supervised subgraph feature selection approach gSSC [19], which employs a subgraph selection principle to ensure that graphs in different (or same) classes should be far from (or close to) each other, and unlabeled graphs should be well separated. By ignoring unlabeled graphs, gSSC can be used to select subgraphs as features. Once subgraph features are selected, the remaining designs are the same as the above instance feature selection based baseline approaches.

C. Experimental Setting

After finding optimal subgraphs and instance features under dual embedding strategy, a bag can be represented as one instance in the new feature space, with bag label being propagated to the new instance. As a result, any supervised learning methods can be applied to support MIMG classification (we use Decision Trees (J48) in our experiments). In addition to the above baseline methods, we also compare proposed DE-MIMG with implementations of existing multi-instance learning methods in [26], including lazy learning CitationKNN [2],

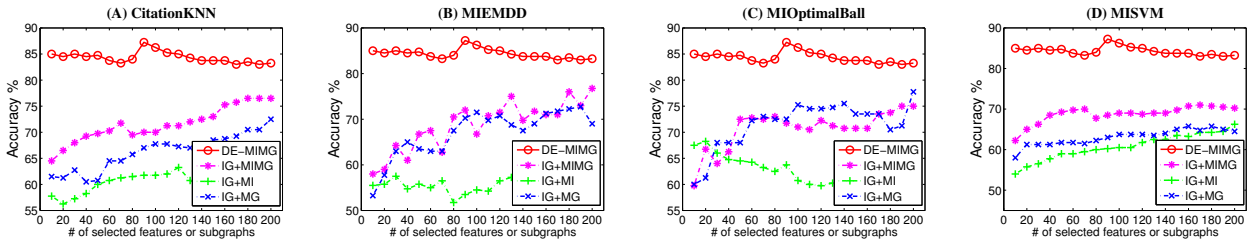

Fig. 7. Accuracy comparisons on **DBLP dataset** by using proposed dual embedding algorithm DE-MIMG and IG based baselines on separated multi-instance setting IG+MI, separated multi-graph setting IG+MG and multi-instance multi-graph learning IG+MIMG on generic multi-instance (MI) learning methods: (A) CitationKNN; (B) MIEMDD; (C) MIOptimallBall; and (D) MISVM.

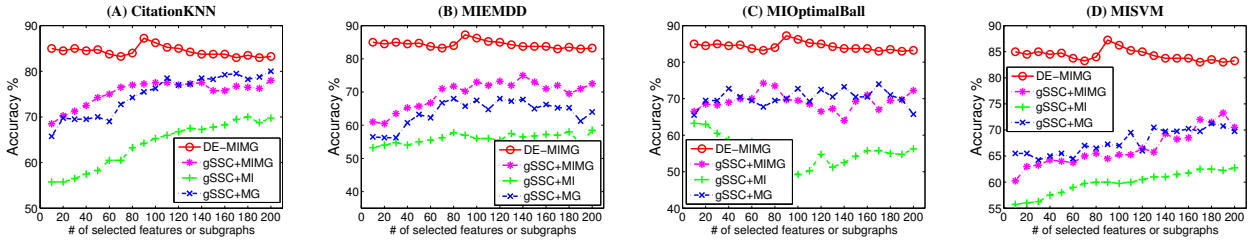

Fig. 8. Accuracy comparisons on **DBLP dataset** by using proposed dual embedding algorithm DE-MIMG and gSSC based baselines on separated multi-instance setting gSSC+MI, separated multi-graph setting gSSC+MG and multi-instance multi-graph learning gSSC+MIMG on generic multi-instance (MI) learning methods: (A) CitationKNN; (B) MIEMDD; (C) MIOptimallBall; and (D) MISVM.

TABLE II. PAIRWISE t-TEST RESULT ON DBLP DATASET. A DENOTES DE-MIMG, WITH B, C AND D DENOTING MI, MG AND MIMG SETTING, RESPECTIVELY. $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ *and* $\mathcal{H}_4$ DENOTE CITATIONKNN, MIEMDD, MIOPTIMALBALL AND MISVM, RESPECTIVELY.

	IG				gSSC		
	A-B	A-C	A-D		A-B	A-C	A-D
$\mathcal{H}_1$	1.2E-22	2.5E-14	1.9E-11	$\mathcal{H}_1$	9.2E-13	4.5E-8	5.3E-11
$\mathcal{H}_2$	2.5E-17	1.1E-11	2.2E-10	$\mathcal{H}_2$	1.8E-23	4.2E-15	8.6E-12
$\mathcal{H}_3$	6.2E-19	2.9E-10	1.5E-11	$\mathcal{H}_3$	2.6E-17	2.4E-16	1.2E-15
$\mathcal{H}_4$	9.7E-17	4.2E-12	2.3E-19	$\mathcal{H}_4$	2.0E-19	6.2E-16	7.1E-14

diverse density with expectation maximization MIEMDD [5], boosting approach MIOptimalBall [7] and kernel method MISVM [3]. All reported results are based on 10 times 10-fold cross-validation with classification accuracy being used as the performance metric. Unless specified otherwise, the default parameter settings are as follows: minimum support threshold $min\_sup = 2\%$ for Online Product Review and $min\_sup = 3\%$ for DBLP dataset. All experiments are conducted on a Linux cluster computing node with an Interl(R) Xeon(R) @3.33GHZ CPU and 3GB fixed memory size.

D. Accuracy Comparisons

Our first set of experiments validate the learning performance by treating instances or graphs in bags as two independent views: a multi-instance (MI) setting and a multi-graph (MG) setting, respectively. In addition, we also compare DE-MIMG with simply concatenated MIMG learning approach. In Figures 5 and 6, we compare the performance of DE-MIMG with instance feature selection and graph feature selection based baseline approaches, by varying the number of subgraphs or instance features from 10 to 200 for Online Product Review dataset. The results on DBLP dataset are reported in Figures 7 and 8.

1) Accuracy Comparisons with MI Setting and MG Setting: Overall, the results demonstrate that given a MIMG task, treating graphs and instances in a separated way is not a good approach. In most cases, MG setting can achieve much better performance than MI setting on both Online Product Review and DBLP datasets. This is mainly attributed to the fact that graphs contain detailed text descriptions (paper abstract for DBLP and review descriptions for online products) and structure dependency between keywords, which is absent in instances. Figures 5-8 show that DE-MIMG is normally 10%-15% more accurate than baseline methods. This demonstrates that, by combining graphs and instances, DE-MIMG is able to find the most effective subgraphs and instance features to represent bags for classification.

In the first two columns of Tables I and II, we report the pairwise t-test (with confidence level $\alpha = 0.05$) to validate the statistical significance between our proposed method and baselines. Each entry (value) denotes the p-value for a t-test between two algorithms, and a p-value less than $\alpha = 0.05$ indicates that the difference is statistically significant. From the first two columns in each table, DE-MIMG statistically outperforms baselines, which separate graphs and instances in bags into isolated views, in all cases.

2) Accuracy Comparisons with Simple Graph and Instance Feature Concatenation: The major difference between DE-MIMG and MIMG (IG+MIMG and gSSC+MIMG) is that IG+MIMG and gSSC+MIMG concatenate features selected from isolated graph and instance domain to form a new representation, whereas DE-MIMG employs iterative embedding between graphs and instances to find new feature representation for bags. The results in Figures 5-8 show that although IG+MIMG and gSSC+MIMG are slightly superior to IG+MG, IG+MG, gSSC+MI, and gSSC+MG on DBLP dataset, their performances are inferior to DE-MIMG on both datasets, with 10% or more accuracy drop on average. This demonstrates that the complementary information between instances and graphs, in each bag, can help select effective features to represent bags for learning.

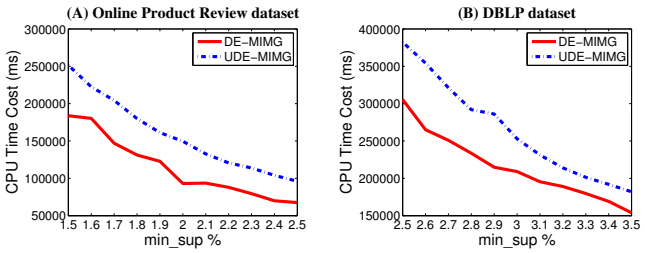

Fig. 9. Average CPU runtime for DE-MIMG *v.s.* unpruned UDE-MIMG with different min_sup under a fixed number of subgraphs m=200 on Online Product Review and DBLP datasets, respectively.

In the last column of Tables I and II, we report the pairwise t-test with confidence level $\alpha = 0.05$. The p-values (less than 0.05) in each entry confirm that DE-MIMG statistically significantly outperforms MIMG.

E. Efficiency of the Pruning Strategy

In order to demonstrate the efficiency of the proposed pruning module for searching subgraphs in DE-MIMG, as described in Section IV-B2, we implement a UDE-MIMG approach which does not have pruning module and compare its runtime performance with DE-MIMG. By doing so, we can empirically demonstrate the efficiency of the pruning module. In our implementation, UDE-MIMG first uses gSpan to find a set of frequent subgraphs, and then selects the optimal set of subgraphs by using the same criteria as DE-MIMG.

In Figures 9(A) and 9(B), we report the average CPU runtime performance with respect to different minimum support min_sup values (the number of selected subgraphs is fixed to 200) on Online Product Review and DBLP datasets, respectively. The results show that, as the min_sup values increase, the runtime of both DE-MIMG and UDE-MIMG decrease, mainly because a larger min_sup value will reduce the number of candidates for validation. DE-MIMG demonstrates much better runtime performance than an unpruned version. This is mainly attributed to the pruning module in DE-MIMG, which uses threshold min_sup and upper bound $\tau = \min_{g_i \in \mathbf{g}} r(g_i)$ (as shown in Algorithm 1) to dynamically prune the candidate set for better runtime efficiency.

VII. Conclusions

In this paper, we investigated a new multi-instance multi-graph (MIMG) learning task, where a bag contains a number of instances and graphs, and labels are only available for bags but not for individual instances or graphs. This problem setting is significantly more challenging than traditional multi-instance learning because no feature representation is immediately available to represent graphs in each bags. To address the challenge, we proposed a dual embedding feature evaluation criterion to find optimal subgraphs and instance features by mutually embedding instance distributions into subgraph mining and embedding graph distribution information into instance feature selection. The mutual embedding process results in an optimal set of subgraphs and instance features to represent bags for learning. Experiments and comparisons on real-world tasks show that the proposed DE-MIMG approach significantly outperforms baselines.

References

[1] T. Dietterich., R. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, 1997.

[2] J. Wang, "Solving the multiple-instance problem: A lazy learning approach," in *ICML*, 2000, pp. 1119–1125.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2003, pp. 561–568.

[4] O. Maron and T. Lozano-Prez, "A framework for multiple-instance learning," in *NIPS*, 1998, pp. 570–576.

[5] Q. Zhang and S. Goldman, "Em-dd: An improved multiple-instance learning technique," in *NIPS*, 2001, pp. 1073–1080.

[6] H. Yuan, M. Fang, and X. Zhu, "Hierarchical sampling for multi-instance ensemble learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 99, no. PrePrints, p. 1, 2012.

[7] P. Auer and R. Ortner, "A boosting approach to multiple instance learning," in *ECML*, 2004, pp. 63–74.

[8] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-i.i.d. samples," in *ICML*, 2009, pp. 1249–1256.

[9] D. Zhang, Y. Liu, L. Si, J. Zhang, and R. D. Lawrence, "Multiple instance learning on structured data," in *NIPS*, 2011, pp. 145–153.

[10] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in *CVPR*, 2007, pp. 1–8.

[11] R. Angelova and G. Weikum, "Graph-based text classification: learn from your neighbors," in *SIGIR*, 2006, pp. 485–492.

[12] S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Graph stream classification using labelled and unlabeled graphs," in *ICDE*, 2013.

[13] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *ICDM*, 2001, pp. 313–320.

[14] R. Memisevic, "On multi-view feature learning." in *ICML*, 2012.

[15] M. Mayo and E. Frank, "Experiments with multi-view multi-instance learning for supervised image classification." in *IVCNZ*, 2011, pp. 363–369.

[16] Z.-H. Zhou and M.-L. Zhang, "Solving multi-instance problems with classifier ensemble based on constructive clustering," *Knowl. Inf. Syst.*, vol. 11, pp. 155–170, 2007.

[17] M. Grbovic, C. Dance, and S. Vucetic, "Sparse principal component analysis with constraints," in *AAAI*, 2012, pp. 935–941.

[18] W.-K. Ching, D. Chu, L.-Z. Liao, and X. Wang, "Regularized orthogonal linear discriminant analysis," *Pattern Recognition*, vol. 45, no. 7, pp. 2719–2732, 2012.

[19] X. Kong and P. Yu, "Semi-supervised feature selection for graph classification," in *KDD*, 2010, pp. 793–802.

[20] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *ICDM*, 2002, pp. 721–724.

[21] X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," in *SIGMOD*, 2008, pp. 433–444.

[22] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in *WWW*, 2013, pp. 897–908.

[23] X. Luo, Z.X., J. Yu, and X. Chen, "Building association link network for semantic link on web resources," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 482–494, 2011.

[24] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *KDD*, 2008, pp. 990–998.

[25] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[26] I. Witten. and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann Publishers, 2005. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/