Multi-instance Ensemble Learning with Discriminative Bags

Mei Yang, Yu-Xuan Zhang, Xizhao Wang, *Fellow, IEEE*, and Fan Min, *Member, IEEE*

***Abstract*—Multi-instance learning (MIL) is more general and challenging than traditional supervised learning in that labels are given at the bag level. The popular feature mapping approaches convert each bag into an instance in the new feature space. However, most of them hardly maintain the distinguishability of bags, and the MIL model does not support self-reinforcement. In this paper, we propose the multi-instance ensemble learning with discriminative bags (ELDB) algorithm with two new techniques. The bag selection technique obtains discriminative bag set (dBagSet) according to two parts. First, considering the space and label distribution of the data, the bag selection process is optimized through discriminative analysis to obtain the basic dBagSet. Second, with the state and action transfer strategy, a dBagSet with better distinguishability is obtained through self-reinforcement. The ensemble technique trains a series of classifiers with these dBagSets and obtains the final weighted model. Experimental results show that ELDB is superior to the state-of-the-art MIL mapping solutions.**

***Index Terms*—Distinguishability, ensemble learning, mapping, multi-instance learning and self-reinforcement.**

I. Introduction

COMPARED with traditional single-instance learning (SIL), the processing data of multi-instance learning (MIL) [1] is a set of bags with bag-level labels instead of a set of instances with instance-level labels. In applications, a bag is labeled as positive if it contains at least one positive instance, otherwise it is negative. Accordingly, the task of MIL is to classify bags instead of individual instances. Over the years, MIL has already been widely applied in many domains, such as drug activity prediction [2], [3], image retrieval [4], [5], image classification [6], [7] and text classification [8], [9].

Existing MIL learners can be roughly divided into three categories [10]: a) Instance-based approaches seek a function to discriminate instances through a SIL classifier, and establish MIL assumptions to link the predicted instance label with the bag label [11], [12]. The label of the instance is unknown and the adaptability of MIL assumptions is weak. Hence the prediction error of the instance label directly affects the prediction of the bag label. b) Bag-based approaches measure the distance between bags and train a bag-level classification model [8], [13]. They do not need to consider instance-level labels. In contrast, the bag-level distance measure has a great impact on model performance. c) Mapping-based approaches map bags into the new feature space according to statistics information or clustering technologies. Then bag-level labels are predicted according to the corresponding mapping vectors [2], [3], [14]. Through space mapping, the influence of distance measurement can be reduced to a certain extent.

The last category can be further divided into the following subcategories. Statistic-based mapping approaches represent each bag as a single vector with one or more statistic values [13], [15]. Kernel-based mapping approaches focus on designing a kernel for the mapping [9], [16]. Instance-based mapping approaches transform each bag to a single instance via instance selection techniques [4], [17]. Bag-based mapping approaches transform each bag according to its spatial relationship [3]. Unfortunately, most existing mapping-based approaches hardly maintain the distinguishability of bags in the new feature space. Additionally, the MIL model does not support self-reinforcement, which means it cannot learn more to improve its distinguishability. Therefore, bag-based mapping methods face the following challenges: a) How to improve the distinguishability of bags in the new feature space? and b) How does the model acquire the self-reinforcement ability?

In this paper, we propose the multi-instance ensemble learning with discriminative bags (ELDB) algorithm to handle these issues. Figure 1 compares ELDB with the traditional bag-based mapping (TBBM) algorithm. The differences include: a) TBBM only generates a key bag set (kBagSet) by considering the spatial distribution of the data, while ELDB generates a discriminative bag set (dBagSet) by further considering the label information of the data; b) ELDB introduces a self-reinforcement mechanism to learn and update the existing dBagSet. Consequently, the updated dBagSet will have higher distinguishability; and c) ELDB generates a series of weighted models with the ensemble technique.

By following MIL mapping-based ensemble methods, the ensemble technique aims to improve the classification performance and stability of the model. These methods, such as the clustering-based and hierarchical sampling methods [18], [19], commonly train numerous single-instance classifiers by repeating the mapping process. Furthermore, each classifier can assign a weight by considering its own contribution [20]. As a result, in our case, it is more accurate to predict the label of the bag through the weighted ensemble model.

This work was supported by the National Natural Science Foundation of China (62006200), Natural Science Foundation of Sichuan Province (2019YJ0314) and Sichuan Province Youth Science and Technology Innovation Team (2019JDTD0017). (*Corresponding author: Fan Min.*)

Mei Yang is with the School of Computer Science, Southwest Petroleum University, Chengdu 610500, China (e-mail: yangmei@swpu.edu.cn).

Yu-Xuan Zhang is with the School of Computer Science, Southwest Petroleum University, Chengdu 610500, China (e-mail: 201921000434@stu.swpu.edu.cn).

Xizhao Wang is with Institute of Big Data, Shenzhen University, Shenzhen 518060, China (e-mail: xizhaowang@ieee.org).

Fan Min is with the School of Computer Science; Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu 610500, China (e-mail: minfan@swpu.edu.cn).

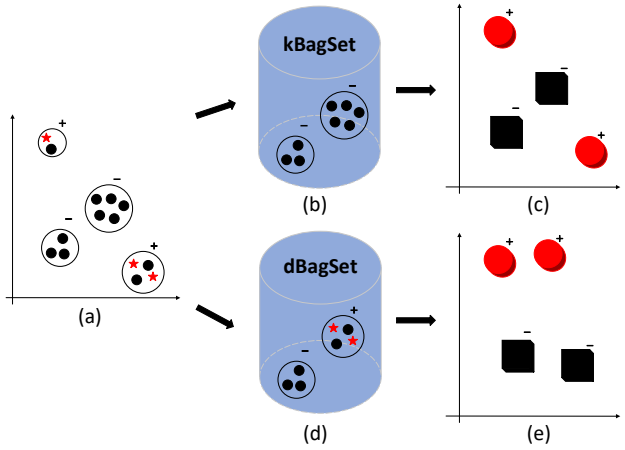

Fig. 1. The process of TBBM and ELDB: a) Original bag space; b) kBagSet for traditional bag mapping; c) New feature space with kBagSet; d) dBagSet for discriminative bag mapping; and e) New feature space with dBagSet.

Experiments are undertaken on thirty-eight MIL classification datasets to quantify the performance of ELDB. These datasets are selected from different application areas, such as drug activity prediction, mutagenicity prediction, image retrieval and text categorization. The experimental results show that ELDB is superior to rival algorithms in general and has higher stability.

There are two features of ELDB:

1) **A new discriminative bag selection method:** a) Compared with the state-of-the-art mapping-based methods, the mapping vectors of bags generated by ELDB have better distinguishability in terms of theoretical analysis; and b) Compared with instance-based discriminative mapping methods, ELDB has higher scalability in terms of time complexity analysis and experiments.
2) **A classifier ensemble method:** Multiple weighted models jointly determine the label of the bag. Consequently, the final prediction is more accurate and stable.

The rest of this paper is organized as follows. Section II introduces basic notations and some MIL methods directly related to our work. Section III proposes the ELDB algorithm. Section IV describes the comparison algorithms, the datasets used, and the experimental results and discussions. Section V concludes and points out some future issues.

II. Preliminaries

Table I captures some important notations used throughout the paper. A multi-instance dataset is denoted by $\mathcal{T} = \{\boldsymbol{B}_i\}_{i=1}^{N}$, where $\boldsymbol{B}_i = \{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}$ is a bag, $\boldsymbol{x}_{ij} \in \mathbb{R}^d$ is an instance, and d is the dimension of each instance. $\boldsymbol{Y} = [y_1, \ldots, y_N]$, where $y_i \in \{-1, +1\}$ is the label of $\boldsymbol{B}_i$.

Three lines of researches are directly related to our work. They are instance-based discriminative analysis, mapping function construction and classifier ensembling.

Instance-based discriminative analysis. The mapping method with instance selection is one of the processing strategies for the MIL classification problem. The core part

TABLE I
Notations.

Notation	Meaning
$\mathcal{T} = \{\boldsymbol{B}_i\}_{i=1}^{N}$	The dataset
$\mathcal{T}_d \subset \mathcal{T}$	The basic dataset
$\mathcal{T}_s = \mathcal{T} \setminus \mathcal{T}_d$	The update dataset
$\mathcal{T}_e \subset \mathcal{T}$	The discriminative bag set (dBagSet)
$\boldsymbol{Y} = [y_1, \ldots, y_N]$	The label vector
$\boldsymbol{B}_i = \{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}$	The i-th bag of $\mathcal{T}$
$\boldsymbol{x}_{ij}$	The j-th instance of $\boldsymbol{B}_i$
y_i	The label of $\boldsymbol{B}_i$
N	The cardinality of $\mathcal{T}$
ψ	The cardinality of $\mathcal{T}_e$
n_i	The cardinality of $\boldsymbol{B}_i$
$\boldsymbol{b}_i$	The mapping vector of $\boldsymbol{B}_i$

is to design a mapping function based on selected instances and to transform bags into the new feature space. The simplest approach is to construct a mapping function from all instances in the intermediate instance pool [4]. The similarity between bag $\boldsymbol{B}_i$ and instance $\boldsymbol{x} \in \boldsymbol{X} = \bigcup_{i=1}^{N} \boldsymbol{B}_i$ is defined as

$$f_s^C(\boldsymbol{B}_i, \boldsymbol{x}) = \min_j \exp(-\lambda \|\boldsymbol{x}_{ij} - \boldsymbol{x}\|^2). \tag{1}$$

One disadvantage of this method is that the number of instances largely determines the time cost of the algorithm. Two strategies to deal with this problem are as follows [21]. One is to select an instance from each positive bag via kernel density estimation. The other is to select the most positive instance and the least negative instance from all instances. However, neither of them considers the distinguishability of bags in new feature space.

To handle this problem, a clustering-based strategy is designed to explore correlations between positive and negative concepts [22]. An instance evaluation criterion is proposed to select the most discriminative instances [2]. In addition, a similarity function different from Eq. (1) is defined as follows:

$$f_s^W(\boldsymbol{B}_i, \boldsymbol{x}) = \max_j \exp(-\lambda \|\boldsymbol{x}_{ij} - \boldsymbol{x}\|^2). \tag{2}$$

We borrow the core idea of these two methods and design the bag selection technique with discriminative analysis and self-reinforcement mechanism.

Mapping function. With a dBagSet $\mathcal{T}_e = \{\boldsymbol{B}_{\zeta_k}\}_{k=1}^{\psi} \subset \mathcal{T}$, $\boldsymbol{B}_i \in \mathcal{T}$ is mapped to a new feature space as follows [3]:

$$f_b(\boldsymbol{B}_i, \mathcal{T}_e) \mapsto \boldsymbol{b}_i = [b_{i\zeta_1}, \ldots, b_{i\zeta_\psi}], \tag{3}$$

where $1 \le \zeta_k \le N$ and $b_{i\zeta_k}$ is the correlation value between $\boldsymbol{B}_i$ and $\boldsymbol{B}_{\zeta_k}$. Consequently, the dataset is mapped to a new one as

$$f_m(\mathcal{T}, \mathcal{T}_e) \mapsto V = \{\boldsymbol{b}_i\}_{i=1}^{N}. \tag{4}$$

Here we formulate the correlation function of two bags as average Hausdorff distance [3]. In addition, we introduce a simple distance:

$$b_{ik} = \|\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_k\|, \tag{5}$$

where $\overline{\boldsymbol{x}}_i = \sum_{j=1}^{n_i} \boldsymbol{x}_{ij}/n_i$.

Classifier ensembling. Zhou et al. [18] showed that the multi-instance representation can be adapted to SIL. In their setting, by repeating the clustering-based mapping strategy

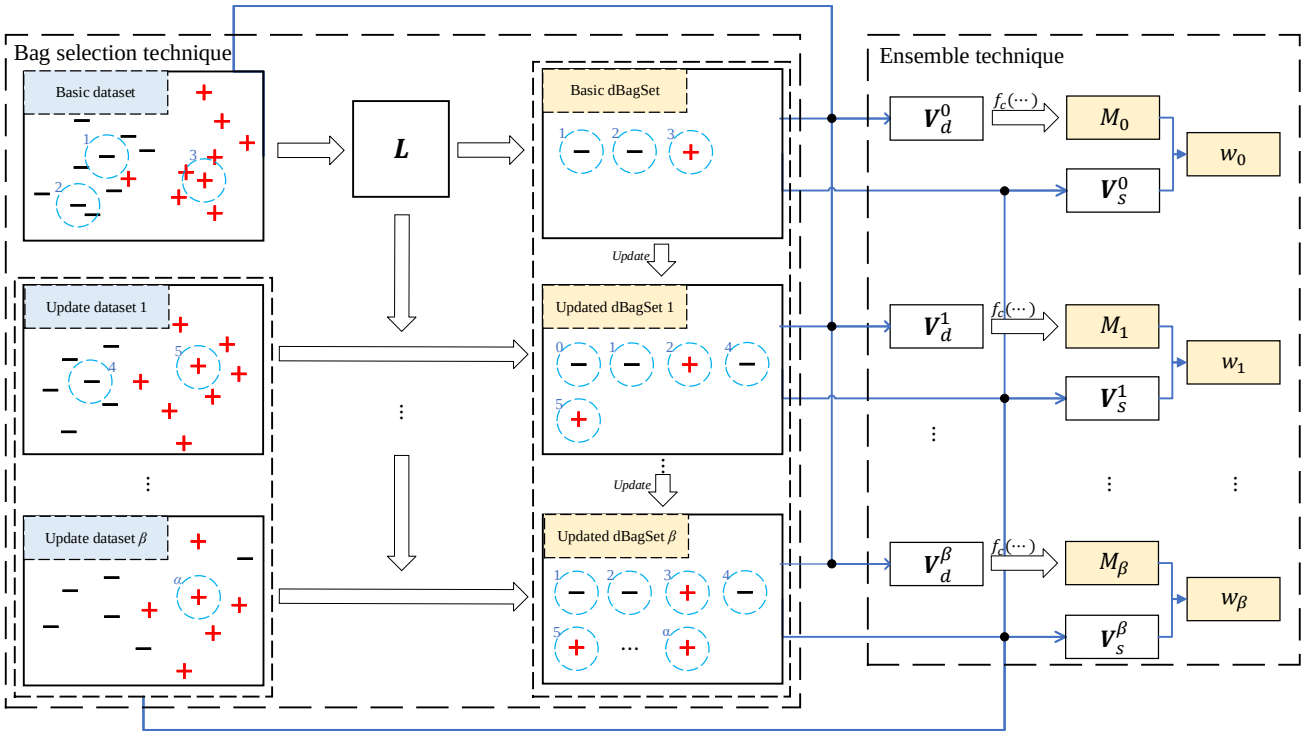

Fig. 2. The overall framework of ELDB with bag selection and ensemble techniques. The symbols "+" and "−" represent the positive and negative bags respectively. The dotted circles with number represent the selected bags. The variable $\boldsymbol{L}$ represents the learned discriminative matrix and will be used to update the basic dBagSet. The solid lines with the hollow 90°- and the hollow-arrow represent the mapping of basic dataset to the set of single-instances $\boldsymbol{V}_d^i$ and $\boldsymbol{V}_s^i$ based on a dBagSet respectively. The weight w_i is the performance value of the model M_i on $\boldsymbol{V}_s^i$.

with different clustering centers, many classifiers can be combined into an ensemble for prediction. By contrast, we introduce the self-reinforcement mechanism and assign a weight to each classifier.

III. The Proposed Algorithm

In this section, we will first present the overall framework, and then introduce two crucial techniques, namely the bag selection technique and ensemble technique.

A. Overall Framework

Figure 2 presents the overall framework of ELDB. To improve the classification performance and the stability of MIL single-model (such as [2], [3], [16]), we randomly divide the dataset $\mathcal{T}$ into the basic dataset $\mathcal{T}_d = \{\boldsymbol{B}_{\xi_i}\}_{i=1}^{N_d}$ and the update dataset $\mathcal{T}_s = \{\boldsymbol{B}_{\xi_i}\}_{i=N_d+1}^{N}$, where $1 \leq \xi_i \leq N$. Then the weighted ensemble model is learned through two crucial techniques.

First, the bag selection technique with two parts is used to generate the dBagSet $\mathcal{T}_e \subset \mathcal{T}$. One part is the discriminative analysis. By considering the spatial and label distribution of $\mathcal{T}_d$, we generate the discriminative matrix $\boldsymbol{L}$ and the basic dBagSet $\mathcal{T}_e^0$. Another part is the self-reinforcement mechanism. For the subset of $\mathcal{T}_s$, the self-reinforcement mechanism is employed to determine whether a dBagSet can be updated. Consequently, we can obtain the updated dBagSets with higher distinguishability.

Second, the ensemble technique integrates these dBagSets according to a SIL classifier $f_c(\cdots)$ and the specified mapping function: a) Basic and update datasets will be mapped as the sets of single-instances $\boldsymbol{V}_d^i$ and $\boldsymbol{V}_s^i$ respectively. b) The single-instances model M_i is trained with $\boldsymbol{V}_d^i$ and the label vector $[y_{\xi_1}, \ldots, y_{\xi_{N_d}}]$; and c) The model weight w_i is computed based on M_i and $\boldsymbol{V}_s^i$. Finally, the weighted ensemble model is obtained by integrating dBagSets, models and weights.

B. The Bag Selection Technique

The bag selection technique includes two parts. First, the discriminative analysis technique generates the basic dBagSet. Second, the self-reinforcement mechanism provides a strategy for updating dBagSet.

1) Discriminative Analysis: We rewrite the instance evaluation criterion [2] as the discriminative analysis for bags as follows. To obtain the basic dBagSet with the discriminative power according to the spatial and label distribution of the dataset, we need to compute

$$\max_{\mathcal{T}_e \subseteq \mathcal{T}_d \subset \mathcal{T}} \sum_{y_{\xi_i} \neq y_{\xi_j}} d\left(f_b\left(\boldsymbol{B}_{\xi_i}, \mathcal{T}_e\right), f_b\left(\boldsymbol{B}_{\xi_j}, \mathcal{T}_e\right)\right), \quad (6)$$

and

$$\min_{\mathcal{T}_e \subseteq \mathcal{T}_d \subset \mathcal{T}} \sum_{y_{\xi_i} = y_{\xi_j}} d\left(f_b\left(\boldsymbol{B}_{\xi_i}, \mathcal{T}_e\right), f_b\left(\boldsymbol{B}_{\xi_j}, \mathcal{T}_e\right)\right), \quad (7)$$

where $d(\cdot, \cdot)$ denotes the distance between two mapping vectors. In case that $\mathcal{T}_e$ is known, we also let $d_{ij} = d(f_b(\boldsymbol{B}_{\xi_i}, \mathcal{T}_e), f_b(\boldsymbol{B}_{\xi_j}, \mathcal{T}_e))$.

To transform the multi-objective optimization into single-objective, the bag-link matrix $\Delta = [\delta_{ij}]_{N_d \times N_d}$ is introduced, where

$$\delta_{ij} = \begin{cases} \lambda_{ij}, & y_{\xi_i} \neq y_{\xi_j}; \\ -\lambda_{ij}, & y_{\xi_i} = y_{\xi_j}, \end{cases} \tag{8}$$

where $\lambda_{ij} > 0$ is a scale parameter. Consequently, the combined optimization objective is

$$\max_{\mathcal{T}_e \subseteq \mathcal{T}_d \subset \mathcal{T}} \mathcal{J}(\mathcal{T}_d, \mathcal{T}_e) = \frac{1}{2} \sum_{\boldsymbol{B}_{\xi_i}, \boldsymbol{B}_{\xi_j} \in \mathcal{T}_d} d_{ij} \delta_{ij}. \tag{9}$$

It represents the distinguishability of all bags belonging to $\mathcal{T}_d$ in the new feature space. However, the current problem is how to find $\mathcal{T}_e$. The simplest method is to traverse each non-empty subset $\mathcal{T}_e \subseteq \mathcal{T}_d$, but the time complexity $O(2^N)$ is unacceptable.

To tackle this problem, the diagonal bag selection matrix $\boldsymbol{Q} = [q_{ij}]_{N_d \times N_d}$ is introduced, where $q_{ij} = 1$ if $i = j$ and $\boldsymbol{B}_{\xi_i} \in \mathcal{T}_e$; otherwise 0. More specifically, d_{ij} is formulated as

$$d_{ij} = \|\boldsymbol{Q}\boldsymbol{b}^*_{\xi_i} - \boldsymbol{Q}\boldsymbol{b}^*_{\xi_j}\|^2, \tag{10}$$

where $\boldsymbol{b}^*_{\xi_i} = f_b(\boldsymbol{B}_{\xi_i}, \mathcal{T}_d)$ according to Eq. (3). Then we have

$$\begin{aligned} \mathcal{J}(\mathcal{T}_d, \mathcal{T}_e) = \frac{1}{2} \sum_{i,j} \Big(& \left(\boldsymbol{b}^*_{\xi_i}\right)^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{b}^*_{\xi_i} + \left(\boldsymbol{b}^*_{\xi_j}\right)^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{b}^*_{\xi_j} - \\ & \left(\boldsymbol{b}^*_{\xi_i}\right)^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{b}^*_{\xi_j} - \left(\boldsymbol{b}^*_{\xi_j}\right)^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{b}^*_{\xi_i} \Big) \delta_{ij}. \end{aligned} \tag{11}$$

For Eq. (8), the simplest setting is used here, i.e., $\forall i, j : \lambda_{ij} = 1$. Additionally, let $\Gamma = [\gamma_{ij}]_{N_d \times N_d}$ be a diagonal matrix, where $\gamma_{ii} = \sum_j \delta_{ij}$. So we have

$$\begin{aligned} \mathcal{J}(\mathcal{T}_d, \mathcal{T}_e) &= \sum_{i,j} \left(\left(\boldsymbol{b}^*_{\xi_i}\right)^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{b}^*_{\xi_i} - \left(\boldsymbol{b}^*_{\xi_i}\right)^{\mathrm{T}} \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{b}^*_{\xi_j} \right) \delta_{ij} \\ &= tr\left(\boldsymbol{Q}^{\mathrm{T}} \boldsymbol{V}^*_d (\Gamma - \Delta) \left(\boldsymbol{V}^*_d\right)^{\mathrm{T}} \boldsymbol{Q} \right) \\ &= tr\left(\boldsymbol{Q}^{\mathrm{T}} \boldsymbol{V}^*_d \boldsymbol{L} \left(\boldsymbol{V}^*_d\right)^{\mathrm{T}} \boldsymbol{Q} \right) \\ &= \sum_{\boldsymbol{B}_{\xi_k} \in \mathcal{T}_e} \boldsymbol{b}^*_{\xi_k} \boldsymbol{L} \left(\boldsymbol{b}^*_{\xi_k}\right)^{\mathrm{T}}, \end{aligned} \tag{12}$$

where $\boldsymbol{V}^*_d = f_m(\mathcal{T}_d, \mathcal{T}_d)$ is computed according to Eq. (4). Consequently, the distinguishability score p_k of $\boldsymbol{B}_k \in \mathcal{T}$ is defined as

$$p_k = \boldsymbol{b}^*_k \boldsymbol{L} \left(\boldsymbol{b}^*_k\right)^{\mathrm{T}}, \tag{13}$$

and $\boldsymbol{L}$ serves as the discriminative matrix. Furthermore, the original optimization problem becomes

$$\max_{\mathcal{T}_e \subseteq \mathcal{T}_d \subset \mathcal{T}} \sum_{\boldsymbol{B}_{\xi_k} \in \mathcal{T}_e} p_{\xi_k}. \tag{14}$$

To obtain the basic dBagSet $\mathcal{T}_e$, we calculate the score of each bag $\boldsymbol{B}_{\xi_i} \in \mathcal{T}_d$, and then select ψ bags with the highest score as the elements of $\mathcal{T}_e$. The whole process is represented as

$$\mathcal{T}_e = \mathrm{bagSelection}(\mathcal{T}_d, \psi). \tag{15}$$

By considering the solution interval of Eq. (14), we design four types of dBagSet initialization modes as follows:

1) **Global (g)** uses all bags to generate dBagSet;
2) **Positive (p)** only uses all positive bags;
3) **Negative (n)** only uses all negative bags; and
4) **Balance (b)** chooses equal amount of positive and negative discriminative bags.

We will compare these modes through experimentation.

2) Self-reinforcement Mechanism: We borrow the idea of [23] to design self-reinforcement mechanism for dBagSet updating. Let $\mathcal{T}^i_e$ be the i-th updated state of the dBagSet. Specifically, $\mathcal{T}^0_e$ is obtained according to Eq. (15). Let action a_i indicate whether a state $\mathcal{T}^i_e$ will be updated.

Algorithm 1 presents the pseudo code of the self-reinforcement mechanism. Line 1 sets action a_i to 0. Line 2 computes the score of $\boldsymbol{B}_{\zeta_k} \in \mathcal{T}^i_e$ and obtains the index $\tau = \arg\min_{\boldsymbol{B}_{\zeta_k} \in \mathcal{T}^i_e} p_{\zeta_k}$. Lines 3-13 traverse each bag $\boldsymbol{B}_{\xi_j} \in \mathcal{T}'$. If $p_{\xi_j} \leq p_{\zeta_\tau}$, the state will not be changed; otherwise the action a_i is set to 1. For different action modes, the corresponding operations are:

- **Addition (a)**: Update $\mathcal{T}^i_e$ by adding the selected bag.
- **Replacement (r)**: Update $\mathcal{T}^i_e$ by replacing bag and re-computing p^*_τ.

In fact, the update strategies of these two action modes are completely different. The action "a" uses a greedy strategy of finding as many bags with high distinguishability as possible. While the action "r" introduces a competitive strategy to exclude some bags. Additionally, we mark the algorithm name as aELDB when selecting the action mode "a"; otherwise rELDB.

The discriminative analysis and self-reinforcement mechanism will jointly participate in the construction of the weighted ensemble model.

Algorithm 1 selfReinforcement($\mathcal{T}^i_e, \mathcal{T}', m$).

Input:
State $\mathcal{T}^i_e$;
$\mathcal{T}' \subseteq \mathcal{T}_s \subset \mathcal{T}$;
Action mode m ("a" or "r");
Output:
The updated state $\mathcal{T}^i_e$;
Action a_i;

1: $a_i = 0$;
2: $\tau = \arg\min_{\boldsymbol{B}_{\zeta_k} \in \mathcal{T}^i_e} p_{\zeta_k}$, where $\boldsymbol{B}_{\zeta_\tau} \in \mathcal{T}^i_e$ and p_{ζ_k} is computed according to Eq. (13);
3: **for** ($\boldsymbol{B}_{\xi_j} \in \mathcal{T}'$) **do**
4: **if** ($p_{\xi_j} > p_{\zeta_\tau}$) **then**
5: **if** ($m ==$ "a") **then**
6: $\mathcal{T}^i_e \leftarrow \mathcal{T}^i_e \cup \{\boldsymbol{B}_{\xi_j}\}$; // Add
7: **else**
8: $\mathcal{T}^i_e \leftarrow \mathcal{T}^i_e \cup \{\boldsymbol{B}_{\xi_j}\} \setminus \{\boldsymbol{B}_{\zeta_\tau}\}$; // Replace
9: Update τ and p_{ζ_τ};
10: **end if**
11: $a_i = 1$;
12: **end if**
13: **end for**
14: **return** $\mathcal{T}^i_e$, a_i;

C. The Ensemble Technique

For ease of description, the input/output relationships of a SIL classifier (e.g. kNN) used are formulated as follows:

$$M = f_c^{\text{model}}(\boldsymbol{V}, \boldsymbol{Y}), \tag{16}$$

$$w = f_c^{\text{weight}}(\boldsymbol{V}, \boldsymbol{Y}, M), \tag{17}$$

$$\hat{y}_i = f_c^{\text{predict}}(\boldsymbol{b}_i, M). \tag{18}$$

All inputs have been defined in Section II. For the output, M is the trained single-instance classification model, w is the value of performance measure (e.g. F1-measure) and $\hat{y}_i$ is the predicted label of $\boldsymbol{B}_i$.

Algorithm 2 lists the ELDB algorithm. The initialization is implemented in Lines 1-2. The dataset $\mathcal{T}$ is randomly split into two parts $\mathcal{T}_d$ and $\mathcal{T}_s$. $\mathcal{T}_d$ will be used for the discriminative analysis and $\mathcal{T}_s$ for the self-reinforcement mechanism.

Algorithm 2 ELDB ($\mathcal{T}, \boldsymbol{Y}, \alpha, \psi, t, m$)

Input:
Dataset $\mathcal{T}$;
Label vector $\boldsymbol{Y}$;
Proportion of basic dataset α;
Size of dBagSet ψ;
Size of batch t;
Action mode m ("a" or "r");

Output:
Weighted ensemble model $\mathcal{M}$;

1: $\mathcal{T}_d = \{\boldsymbol{B}_{\xi_i}\}_{i=1}^{N_d}$, where $N_d = \lfloor \alpha \times N \rfloor$ and $N = |\mathcal{T}|$;
2: $\mathcal{T}_s = \{\boldsymbol{B}_{\xi_i}\}_{i=N_d+1}^{N}$;
// Step 1. Basic state and parameter computation.
3: $\mathcal{T}_e^0$ = bagSelection ($\mathcal{T}_d, \psi$) according to Eq. (15);
4: $\boldsymbol{V}_d^0 = f_m(\mathcal{T}_d, \mathcal{T}_e^0)$ according to Eq. (4);
5: $\boldsymbol{V}_s^0 = f_m(\mathcal{T}_s, \mathcal{T}_e^0)$;
6: $\boldsymbol{Y}_d = [y_{\xi_1}, \ldots, y_{\xi_{N_d}}]$;
7: $\boldsymbol{Y}_s = [y_{\xi_{N_d+1}}, \ldots, y_{\xi_N}]$;
8: $M_0 = f_c^{\text{model}}(\boldsymbol{V}_d^0, \boldsymbol{Y}_d)$ according to Eq. (16);
9: $w_0 = f_c^{\text{weight}}(\boldsymbol{V}_s^0, \boldsymbol{Y}_s, M_0)$ according to Eq. (17);
10: Update M_0 by recomputing $f_c^{\text{model}}(\boldsymbol{V}^0, \boldsymbol{Y})$, where $\boldsymbol{V}^0 = f_m(\mathcal{T}, \mathcal{T}_e^0)$;
// Step 2. Weighted ensemble model construction.
11: $\mathcal{M} = \{(M_0, w_0, \mathcal{T}_e^0)\}$;
12: $n_t = \lceil (N - N_d)/t \rceil$;
13: **for** ($i \in [1..n_t]$) **do**
14: $\quad \mathcal{T}' = \{\boldsymbol{B}_{\xi_{((i-1)\times t+N_d+1)}}, \ldots, \boldsymbol{B}_{\xi_{(i\times t+N_d)}}\} \subseteq \mathcal{T}_s$;
15: $\quad \mathcal{T}_e^i$, a_i = selfReinforcement ($\mathcal{T}_e^{i-1}$, $\mathcal{T}'$, m) according to Algorithm 1;
16: $\quad$ **if** ($a_i == 1$) **then**
17: $\quad\quad$ Compute V_d^i and V_s^i based on $\mathcal{T}_e^i$;
18: $\quad\quad M_i = f_c^{\text{model}}(\boldsymbol{V}_d^i, \boldsymbol{Y}_d)$;
19: $\quad\quad w_i = f_c^{\text{weight}}(\boldsymbol{V}_s^i, \boldsymbol{Y}_s, M_i)$;
20: $\quad\quad$ Update M_i by recomputing $f_c^{\text{model}}(\boldsymbol{V}^i, \boldsymbol{Y})$;
21: $\quad\quad \mathcal{M} \leftarrow \mathcal{M} \cup \{(M_i, w_i, \mathcal{T}_e^i)\}$;
22: $\quad$ **end if**
23: **end for**
24: **return** $\mathcal{M}$;

The basic state and parameter computation are implemented in Lines 3-10. Line 3 generates the state $\mathcal{T}_e^0$ according to the discriminative analysis with $\mathcal{T}_d$. Lines 4-5 map $\mathcal{T}_s$ and $\mathcal{T}_d$ to the sets of single-instances $\boldsymbol{V}_s$ and $\boldsymbol{V}_d$ respectively. Lines 6-7 assign the corresponding label vector. Line 8 trains a classification model M_0 according to $\boldsymbol{V}_d^0$ and its corresponding label vector $\boldsymbol{Y}_d$. Line 9 computes the weight of the model M_0. To make the most of $\mathcal{T}$'s information, Line 10 retrains the model M_0.

The weighted ensemble model construction is implemented in Lines 11-23. Line 11 records the basic model M_0 and its weight w_0 and state $\mathcal{T}_e^0$. Line 12 computes the value of loops n_l. Lines 13-23 are the main loop. For each loop, Line 14 chooses a subset $\mathcal{T}'$ of $\mathcal{T}_s$. Line 15 updates the state $\mathcal{T}_e^i$ and gets the current action a_i according to the self-reinforcement mechanism. Lines 16-22 update records when $a_i == 1$.

With the weighted ensemble model $\mathcal{M}$, the label of the bag $\boldsymbol{B}_i$ can be predicted as

$$\hat{y}_i = \text{sign}\left(\sum_j w_j Y_{ij}\right), \tag{19}$$

where

$$Y_{ij} = f_c^{\text{predict}}\left(f_b\left(\boldsymbol{B}_i, \mathcal{T}_e^i\right), M_i\right), \tag{20}$$

where $f_b(\cdot,\cdot)$ is the mapping function according to Eq. (3) and $\text{sign}(x) = 1$ if $x \geq 0$; otherwise -1.

D. Analysis and Discussions

In this subsection, we analyze properties of ELDB and discuss the reason of ensembling.

1) Properties of ELDB: For traditional bag-based mapping methods, kBagSet is generated according to the spatial distribution of the dataset. However, these methods do not consider the distinguishability of bags in the new feature space. Here *distinguishability* is represented by: a) The bags with the same label are similar to each other; and b) The bags with the different label should represent the disparity among them.

To enhance *distinguishability*, we introduce discriminative analysis by considering the spatial and label distribution of the data. If Eq. (14) is optimized, then $\forall \boldsymbol{B}_{\xi_i}, \boldsymbol{B}_{\xi_j}, \boldsymbol{B}_{\xi_k} \in \mathcal{T}_d, y_{\xi_i} = y_{\xi_j}$ and $y_{\xi_i} \neq y_{\xi_k}$, we have $d_{ij} \leq d_{ik}$.

In addition, we design the self-reinforcement mechanism to obtain the updated dBagSet with higher distinguishability. Let $\mathcal{T}_e^i$ and $\mathcal{T}_e^{i+1}$ denote the previous state and the updated state respectively. Their two properties are as follows:

- For action mode "a", $\sum_{j=1}^{|\mathcal{T}_e^i|} p_{\zeta_j} < \sum_{k=1}^{|\mathcal{T}_e^{i+1}|} p_{\zeta_k}$, where p_{ζ_j} and p_{ζ_k} are computed scores according to Eq. (14).
- For action mode "r", $\exists \boldsymbol{B}_{\zeta_\tau} \in \mathcal{T}_e^i, \forall \boldsymbol{B}_{\zeta_k} \in \mathcal{T}_e^{i+1}, p_{\zeta_\tau} < p_{\zeta_k}$.

Consequently, $\mathcal{T}_e^{i+1}$ can make the bag more distinguishable in the new feature space by comparing with $\mathcal{T}_e^i$.

2) Why ensembling: Let $\mathcal{T}_E = \{\mathcal{T}_e^i\}_{i=0}^{\beta}$ be dBagSet union. A straightforward question is, why not only use the state $\mathcal{T}_e^\beta$ with higher distinguishability? On the one hand, with the update of $\mathcal{T}_e^i \in \mathcal{T}_E$, the following events are inevitable: a) For the addition mode, $\lim_{\beta\to\infty} |\mathcal{T}_e^0|/\beta_\infty = 0$, where β_∞ denotes

the cardinality of $\mathcal{T}_e^\beta \setminus \mathcal{T}_e^0$; and b) For the replacement mode, $\lim_{\beta\to\infty} |\mathcal{T}_e^\beta \cap \mathcal{T}_e^0| = 0$. Therefore, the model based on $\mathcal{T}_e^\beta$ may lose the information of the basic state $\mathcal{T}_e^0$. On the other hand, the ensemble model obtained based on $\mathcal{T}_E$ will reduce the uncertainty of a single model that completely depends on $\mathcal{T}_e^i$. With this process, the stability of the model is enhanced.

IV. Experiments

In this section, we report five groups of experimental results to analyze the effectiveness of the ELDB algorithm. Section IV-A describes the characteristics and parameter settings of comparison algorithms. Section IV-B explains the features of seven types of datasets used in experiments. Section IV-C compares the performances of ELDB with seven state-of-the-art algorithms. Section IV-D presents the results of statistical significance comparisons. Section IV-E analyzes the parameter sensitivity of ELDB. Section IV-F compares the time cost of all rival methods. Section IV-G conducts experiments on large-scale datasets. Through these experiments, we aim to answer:

1) Is ELDB more accurate than rival algorithms?
2) What are the advantages and disadvantages of ELDB?
3) How sensitive is ELDB to parameter settings?
4) How efficient is ELDB?

The experimental environment is the Windows 10 64-bit operating system, 16 GB memory, AMD Ryzen 7 4800U CPU 1.8GHz, Python 3.9.2. The Python source code is available at https://github.com/InkiInki/ELDB.

For each dataset, the average F1-measure of ten times 10-fold cross validation (10CV) and its standard deviation (the value with "$\pm$") are reported.

A. Comparison Algorithms

We compare ELDB with four types of state-of-the-art mapping-based MIL algorithms.

1) Statistic-based: We select Simple-MI [15] as the comparison. It uses the mean vector of the bag as the representation of the bag itself. Thus, it does not need parameter settings.

2) Kernel-based: For comparison, the miFV algorithm is used [16]. It uses Gaussian mixture model (GMM) to extract the information of instance space, and then encodes each bag to a single-single. For its parameter settings, the number of components for GMM is enumerated in $\{1, 2, 3\}$ and the PCA energy is set to 1.

3) Instance-based: The crucial step of this type of methods is how to find key instances. For comparison, the StableMIL, miVLAD, MILFM and MILDM algorithms are used [2], [17], [22], [24]. For the parameter settings of StableMIL, the threshold τ is set to 0.25. For miVLAD, the number of clustering centers for kMeans is enumerated in $\{1, 2\}$ and the PCA energy is set to 1. For MILFM and MILDM, the similarity function is formulated as Eq. (2), and γ is enumerated in $\{0.1, \ldots, 1.0\}$. Besides, the number of clustering centers for MILFM and the number of discriminative instances for MILDM are set to 40 and the number of bags, respectively.

4) Bag-based: This type of methods differ from instance-based mapping methods in that the key is bags rather than instances. The BAMIC [3] algorithm is used for comparison. In particular, ELDB is one of these algorithms. The parameter settings of BAMIC and ELDB are: The distance functions are formulated as average Hausdorff distance [3] and Eq. (5); the number of selected bags is set to $r \times \min\{N, 100\}$, where r is enumerated in $\{0.1, \ldots, 1.0\}$, N is the size of dataset. The additional parameters for ELDB were: a) Proportion of basic dataset $\alpha = 0.75$; and b) Size of batch $t = N(1-\alpha)/2$;

By mapping the dataset $\mathcal{T}$ into the set of new feature vectors $\boldsymbol{V}$, we need to employ a SIL classifier to train a single-instance model. The details are implemented in the subsection III-C. In our settings, kNN, J48 and SVM are employed.

B. Experimental Datasets

Two drug activity prediction [1], two mutagenicity prediction [25], three image retrieval [26], [27], one medical image [28], [29], one text [8], two original image [4], [6] and three biocreative datasets [30], [31] are utilized to verify the performances of ELDB. Table II lists some details for these datasets[1]. The symbol "#" means that the original images are transformed to bags based on bag generators [32]. In the following parts, we will briefly introduce the domain knowledge of these datasets.

TABLE II
Detailed properties of the used datasets.

Name	Bags	Instances	Attributes	Max / Min
Musk1	92	476	166	40/2
Musk2	102	6,598	166	1,044/1
Mutagenesis1	188	10,486	7	88/28
Mutagenesis2	42	2,132	7	86/26
Elephant	200	1,391	230	13/2
Fox	200	1,320	230	13/2
Tiger	200	1,220	230	13/1
Messidor	1,200	12,352	687	12/8
Newsgroups	2,000	80,137	200	84/8
Component	3,130	36,894	200	53/1
Function	5,242	55,536	200	51/1
Corel	10,000	#	#	#
GHIM	10,000	#	#	#
Process	11,718	118,417	200	57/1

1) Drug Activity Prediction: By studying a collection of existing molecules, drug activity prediction attempts to predict whether new molecules can be manufactured medicines. [1]. Each molecule can be represented as a bag, and its eligibility to manufacture drugs depends on its instances. During the learning procedure, a molecule is positive if at least one instance inside can be used to manufacture a drug; otherwise

[1] The original images of the messidor dataset can be found at http://www.adcis.net/en/third-party/messidor/. The text dataset can be found at http://www.lamda.nju.edu.cn/data_MItext.ashx/. Some processed datasets of Corel and GHIM can be found at https://github.com/InkiInki/Data001. The others can be found at http://www.figshare.com/articles/MIProblems_A_repository_of_multiple_instance_learning_datasets/6633983.

negative. Musk1 and musk2 are MIL real-world drug activity prediction datasets.

2) Mutagenicity Prediction: One way for predicting carcinogenicity is to predict compound molecular mutagenicity. The main challenge is how to effectively detect these molecules [25], [33]. The MIL datasets mutagenesis1 and mutagenesis2 are two versions of mutagenicity prediction.

3) Image Retrieval: The problem of content-based image retrieval includes identifying the intended target object in the image. The difficulty is that the image comprises a large number of diverse things. In the SIL setting, the image may not be retrieved well. Fortunately, this problem is sutable to MIL scenarios: Each image can be viewed as a bag of segments containing one or more regions [27]. The goal is to distinguish whether the image contains an object of interest. The elephant, fox, and tiger [26] datasets are used in our research.

4) Medical Image: Messidor is a medical classification dataset that includes 546 healthy patients and 654 diabetes fundus images [28]. Its purpose is to detect whether the image contains lesions, which is consistent with MIL's application to image retrieval datasets [29], [34].

5) Text: There are twenty sub-datasets in the newsgroups text dataset, such as alt.atheish, comp.graphics and misc.forsale [8]. Each of sub-dataset contains 50 positive and 50 negative bags. TFIDF features[35] represent each instance in the bag as a 200-dimensional vector. The goal of the text dataset is to determine whether a particular bag contains target newsgroups information.

6) Corel and GHIM: To evaluate ELDB and the rival algorithms, we sample 20 and all categories from Corel database with 100 categories [4], [36] and GHIM database with 20 categories [6], respectively. For all images in these databases, we need to build the MIL model to distinguish the selected positive and negative categories. Furthermore, these images have been segmented by the SB system [32], [37]. Figure 3 shows sample images taken from the Corel database.

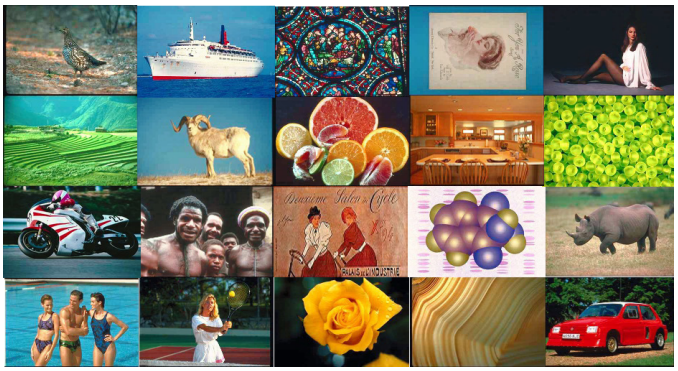

Fig. 3. Sample images taken from Corel database. From left to right and top to bottom are: 1) Bird; 2) Ship; 3) Church; 4) Illustration; 5) Girl; 6) Filed; 7) Goat; 8) Fruit; 9) Hall; 10) Beads; 11) Motorcycle; 12) People; 13) Poster; 14) Molecular; 15) Rhino; 16) Bikini; 17) Tennis; 18) Flower; 19) Texture; and 20) Car.

7) Biocreative: Biocreative is a text categorization dataset [30], [31]. The task is to figure out whether some Gene Ontology codes should be used to annotate a pair of genes. For evaluation, component, function and process datasets are employed.

C. Performance Comparisons

Tables III and IV compare the performance of ELDB and the four types of comparison algorithms. The variables n and d represent the number of instances and dimensions for MIL datasets, respectively. The action modes, addition and replacement of self-enhancement mechanism, are represented by the symbols "a" and "r" of aELDB and rELDB. With the little black bullet, the best performance value for each dataset is emphasized. *Average* denotes the average classification performance across all datasets. *Mean rank* indicates the average of the classification performance ranking of the current algorithm on each dataset.

The experimental results demonstrate that ELDB can be applied to most MIL classification tasks and has the greatest overall classification performance, according to results on each dataset, *average* and *mean rank*; and the standard deviation of ELDB is second only to miFV, according to the average classification results. The following could be the reason: For starters, the self-reinforcement mechanism can improve a dBagSet's overall distinguishability and make mapping results of bags more discriminative. The final model, on the other hand, is made up of a number of weighted models. The label of a bag can be predicted more correctly and stably by taking into account the influence of these models.

Some outcomes necessitate extra attention: a) On the text datasets, the effects of ELDB are significantly better than StableMIL, MILFM and MILDM. The reason may be that the three comparison methods are looking for key instances in the specified instance space. For example, the key instances of StableMIL are positive instances that can change the label of negative bags. However, the text dataset is sparse and the attribute values of instances are small. As a result, some negative instances may have similar characteristics to the positive instances. This may lead to unsatisfactory mapping results of these methods. While ELDB designs the discriminative analysis and the self-reinforcement mechanism to keep the bag distinct in the new feature space. The feature makes ELDB unaffected by characteristics such as dataset sparsity. and b) The classification results of aELDB is overall better than rELDB. The cause for this could be that the action mode "r" ignores some bags with high distinguishability when specifying the cardinality of dBagSet.

D. Statistical Significance Comparisons

Table V summarizes the p-value of two-tailed t-test between ELDB and all comparative algorithms in this experimental scenario. The 95% confidence level ($\alpha = 0.05$) is used to calculate all paired t-test values. According to statistical theory, there is no significant difference between two algorithms when the p-value is greater than 0.05.

The majority of the p-values in the second column are greater than 0.05. As a result, aELDB and rELDB are only two types of ELDB. Similarly, the results of the sixth column demonstrate that miVLAD is the closest approach to ELDB.

TABLE III
F1-MEASURE WITH STANDARD DEVIATIONS ON FIVE TYPES OF MIL DATASETS.

Datasets	$n \times d$	Simple-MI	StableMIL	miFV	miVLAD	MILFM	MILDM	BAMIC	**aELDB**	**rELDB**
Musk1	476×166	$83.69_{\pm 4.30}$	$84.39_{\pm 1.04}$	$88.70_{\pm 2.50}$	$85.69_{\pm 3.15}$	$87.63_{\pm 1.57}$	$89.52_{\pm 2.99}$	$88.48_{\pm 3.40}$	$89.85_{\pm 1.47}$	$90.21_{\pm 1.58}\bullet$
Musk2	$6,598 \times 166$	$76.13_{\pm 3.10}$	$78.56_{\pm 2.81}$	$74.80_{\pm 3.36}$	$71.86_{\pm 2.62}$	$82.90_{\pm 3.18}$	$89.06_{\pm 3.50}\bullet$	$86.84_{\pm 2.21}$	$88.58_{\pm 3.48}$	$85.71_{\pm 3.93}$
Mutagenesis1	$10,486 \times 7$	$87.67_{\pm 1.10}$	$87.88_{\pm 1.43}$	$89.70_{\pm 1.14}\bullet$	$87.77_{\pm 1.50}$	$88.52_{\pm 1.07}$	$88.05_{\pm 0.98}$	$86.00_{\pm 1.25}$	$88.66_{\pm 1.24}$	$88.54_{\pm 1.59}$
Mutagenesis2	$2,132 \times 7$	$57.12_{\pm 9.04}$	$70.24_{\pm 2.24}\bullet$	$50.00_{\pm 6.26}$	$55.33_{\pm 10.68}$	$63.10_{\pm 10.93}$	$54.26_{\pm 4.10}$	$53.32_{\pm 10.53}$	$54.30_{\pm 10.44}$	$51.98_{\pm 12.73}$
Elephant	$1,391 \times 230$	$83.81_{\pm 1.36}$	$84.20_{\pm 4.23}$	$84.80_{\pm 0.83}$	$84.88_{\pm 1.68}$	$83.23_{\pm 2.26}$	$82.45_{\pm 2.01}$	$84.36_{\pm 1.29}$	$85.20_{\pm 1.11}\bullet$	$84.31_{\pm 1.22}$
Fox	$1,320 \times 230$	$61.55_{\pm 1.86}$	$57.20_{\pm 3.34}$	$63.87_{\pm 1.08}$	$65.48_{\pm 1.32}$	$59.93_{\pm 1.84}$	$60.65_{\pm 2.24}$	$64.69_{\pm 2.00}$	$66.62_{\pm 1.13}\bullet$	$64.82_{\pm 1.42}$
Tiger	$1,220 \times 230$	$80.47_{\pm 2.23}$	$64.64_{\pm 3.15}$	$80.37_{\pm 0.91}$	$85.16_{\pm 1.34}\bullet$	$80.01_{\pm 1.86}$	$78.50_{\pm 2.92}$	$79.86_{\pm 0.91}$	$79.59_{\pm 0.98}$	$76.70_{\pm 1.31}$
Messidor	$12,352 \times 687$	$71.34_{\pm 0.25}$	$68.61_{\pm 0.87}$	$74.17_{\pm 0.57}\bullet$	$71.30_{\pm 0.68}$	$65.77_{\pm 0.74}$	$62.72_{\pm 1.23}$	$71.57_{\pm 0.70}$	$73.36_{\pm 0.61}$	$72.46_{\pm 0.87}$
Alt.atheism	$5,443 \times 200$	$48.66_{\pm 6.71}$	$54.20_{\pm 7.76}$	$82.66_{\pm 1.29}$	$84.65_{\pm 1.92}\bullet$	$53.70_{\pm 5.94}$	$56.72_{\pm 5.19}$	$84.02_{\pm 1.48}$	$84.44_{\pm 1.72}$	$83.45_{\pm 2.50}$
Comp.graphics	$3,094 \times 200$	$49.27_{\pm 6.78}$	$44.35_{\pm 6.24}$	$79.34_{\pm 2.72}$	$79.83_{\pm 3.04}$	$44.55_{\pm 5.08}$	$51.11_{\pm 4.41}$	$79.10_{\pm 2.15}$	$80.04_{\pm 2.97}\bullet$	$78.82_{\pm 2.88}$
Comp.os.ms	$5,175 \times 200$	$53.07_{\pm 5.33}$	$45.66_{\pm 5.28}$	$70.22_{\pm 2.98}$	$69.92_{\pm 3.73}$	$49.27_{\pm 3.87}$	$51.92_{\pm 5.83}$	$70.13_{\pm 2.22}$	$71.62_{\pm 2.14}\bullet$	$70.10_{\pm 3.34}$
Comp.sys.mac	$4,473 \times 200$	$47.52_{\pm 4.81}$	$48.11_{\pm 6.58}$	$74.64_{\pm 2.12}$	$77.03_{\pm 2.92}$	$51.65_{\pm 6.61}$	$52.39_{\pm 5.31}$	$81.14_{\pm 3.09}$	$81.12_{\pm 2.31}$	$81.67_{\pm 2.89}\bullet$
Misc.forsale	$5,306 \times 200$	$49.52_{\pm 5.66}$	$44.76_{\pm 3.59}$	$69.07_{\pm 2.23}\bullet$	$68.27_{\pm 2.53}$	$49.10_{\pm 5.56}$	$47.93_{\pm 6.89}$	$66.69_{\pm 5.78}$	$68.07_{\pm 3.31}$	$68.60_{\pm 3.31}$
Rec.sport.baseball	$3,358 \times 200$	$61.82_{\pm 4.92}$	$47.25_{\pm 4.05}$	$82.49_{\pm 1.96}$	$82.99_{\pm 1.17}$	$51.06_{\pm 5.00}$	$54.35_{\pm 5.50}$	$82.95_{\pm 1.46}$	$83.80_{\pm 2.02}\bullet$	$82.43_{\pm 1.60}$
Rec.sport.hockey	$1,982 \times 200$	$73.42_{\pm 3.18}$	$44.30_{\pm 5.91}$	$87.08_{\pm 1.65}$	$91.46_{\pm 1.90}\bullet$	$47.60_{\pm 3.16}$	$51.29_{\pm 5.44}$	$88.52_{\pm 3.47}$	$89.76_{\pm 2.62}$	$90.26_{\pm 1.75}$
Sci.electronics	$3,192 \times 200$	$44.21_{\pm 5.54}$	$45.18_{\pm 3.85}$	$91.19_{\pm 0.93}$	$92.60_{\pm 1.28}$	$55.15_{\pm 5.10}$	$52.76_{\pm 7.21}$	$92.29_{\pm 0.86}$	$92.06_{\pm 1.08}$	$92.61_{\pm 0.74}\bullet$
Sci.med	$3,045 \times 200$	$59.69_{\pm 5.19}$	$46.54_{\pm 6.17}$	$81.47_{\pm 1.77}$	$80.78_{\pm 3.88}$	$49.98_{\pm 4.98}$	$54.80_{\pm 4.85}$	$82.13_{\pm 3.77}$	$82.99_{\pm 2.20}$	$83.18_{\pm 2.20}\bullet$
Sci.space	$3,655 \times 200$	$62.97_{\pm 5.34}$	$44.45_{\pm 5.72}$	$86.53_{\pm 2.22}\bullet$	$81.90_{\pm 3.34}$	$45.02_{\pm 4.37}$	$57.58_{\pm 6.35}$	$82.09_{\pm 0.68}$	$81.27_{\pm 2.46}$	$80.13_{\pm 3.25}$
Average		$64.00_{\pm 4.26}$	$58.92_{\pm 4.13}$	$78.39_{\pm 2.03}$	$78.72_{\pm 2.70}$	$61.57_{\pm 3.51}$	$63.11_{\pm 4.83}$	$79.12_{\pm 2.63}$	$80.07_{\pm 2.41}\bullet$	$79.22_{\pm 2.73}$
Mean rank		6.50	7.67	3.83	3.67	6.72	6.39	4.06	2.56$\bullet$	3.61

TABLE IV
F1-MEASURE WITH STANDARD DEVIATIONS ON COREL AND GHIM DATABASES.

Positive class	Negative class	$n \times d$	Simple-MI	StableMIL	miFV	miVLAD	MILFM	MILDM	BAMIC	**aELDB**	**rELDB**
Bird	Ship	$3,200 \times 12$	$88.37_{\pm 0.98}$	$90.04_{\pm 0.74}$	$92.54_{\pm 0.94}$	$89.04_{\pm 0.83}$	$93.55_{\pm 1.17}$	$91.59_{\pm 1.35}$	$95.32_{\pm 0.78}\bullet$	$94.89_{\pm 1.26}$	$95.02_{\pm 0.83}$
Church	Illustration	$3,200 \times 12$	$94.79_{\pm 0.76}$	$100.00_{\pm 0.00}\bullet$	$97.56_{\pm 0.77}$	$94.44_{\pm 0.45}$	$95.41_{\pm 0.35}$	$94.17_{\pm 0.65}$	$91.61_{\pm 0.99}$	$95.60_{\pm 0.96}$	$95.13_{\pm 0.84}$
Girl	Field	$3,200 \times 12$	$96.17_{\pm 0.54}$	$97.25_{\pm 0.78}$	$96.39_{\pm 0.54}$	$96.77_{\pm 1.26}$	$98.62_{\pm 0.34}$	$95.82_{\pm 0.68}$	$98.29_{\pm 0.65}$	$98.70_{\pm 0.37}\bullet$	$98.49_{\pm 0.65}$
Goat	Fruit	$3,200 \times 12$	$77.88_{\pm 1.49}$	$86.64_{\pm 0.77}$	$88.84_{\pm 1.39}$	$76.92_{\pm 2.10}$	$88.13_{\pm 1.40}$	$86.40_{\pm 2.18}$	$89.84_{\pm 1.78}$	$89.07_{\pm 1.70}$	$88.92_{\pm 1.25}\bullet$
Hall	Beads	$3,200 \times 12$	$87.70_{\pm 0.97}$	$80.80_{\pm 0.54}$	$87.53_{\pm 1.53}$	$83.99_{\pm 2.99}$	$88.78_{\pm 1.14}\bullet$	$91.45_{\pm 1.53}$	$86.13_{\pm 1.26}$	$87.65_{\pm 0.58}$	$86.56_{\pm 1.03}$
Motorcycle	People	$3,200 \times 12$	$75.91_{\pm 2.29}$	$78.85_{\pm 2.74}$	$81.27_{\pm 0.85}$	$73.93_{\pm 1.69}$	$83.16_{\pm 1.60}$	$85.35_{\pm 1.54}$	$85.44_{\pm 1.33}$	$85.38_{\pm 1.47}$	$85.47_{\pm 0.97}\bullet$
Poster	Molecular	$3,200 \times 12$	$92.96_{\pm 1.13}$	$82.78_{\pm 1.31}$	$90.64_{\pm 1.09}$	$95.42_{\pm 1.02}$	$99.54_{\pm 0.08}$	$99.65_{\pm 0.25}$	$99.83_{\pm 0.22}$	$100.00_{\pm 0.00}\bullet$	$99.86_{\pm 0.29}$
Rhino	Bikini	$3,200 \times 12$	$83.83_{\pm 1.62}$	$94.51_{\pm 0.74}$	$97.30_{\pm 0.50}\bullet$	$85.20_{\pm 2.46}$	$96.18_{\pm 0.50}$	$94.79_{\pm 0.53}$	$96.98_{\pm 0.96}$	$96.31_{\pm 0.46}$	$96.34_{\pm 0.77}$
Tennis	Flower	$3,200 \times 12$	$93.62_{\pm 0.70}$	$93.49_{\pm 0.56}$	$93.48_{\pm 1.05}$	$93.07_{\pm 0.78}$	$93.90_{\pm 0.44}$	$90.15_{\pm 1.13}$	$95.79_{\pm 1.00}\bullet$	$95.38_{\pm 1.21}$	$95.24_{\pm 1.00}$
Texture	Car	$3,200 \times 12$	$96.23_{\pm 0.51}$	$88.93_{\pm 2.01}$	$95.42_{\pm 0.39}$	$94.26_{\pm 0.94}$	$94.22_{\pm 0.71}$	$96.36_{\pm 0.67}\bullet$	$95.51_{\pm 0.99}$	$95.66_{\pm 1.09}$	$95.28_{\pm 0.77}$
Aircraft	Forbidden City	$16,000 \times 12$	$84.84_{\pm 0.46}$	$86.74_{\pm 0.62}$	$86.16_{\pm 0.28}$	$77.94_{\pm 0.63}$	$89.33_{\pm 0.59}$	$79.47_{\pm 0.29}$	$89.69_{\pm 0.65}$	$90.30_{\pm 0.88}\bullet$	$89.61_{\pm 0.60}$
Beetle	Horse	$16,000 \times 12$	$85.06_{\pm 0.40}$	$85.87_{\pm 0.40}$	$85.34_{\pm 0.25}$	$85.63_{\pm 0.61}$	$88.05_{\pm 0.40}$	$83.81_{\pm 0.48}$	$88.57_{\pm 0.60}\bullet$	$88.38_{\pm 0.45}$	$88.01_{\pm 0.72}$
Beach	Chicken	$16,000 \times 12$	$97.59_{\pm 0.36}\bullet$	$91.92_{\pm 0.45}$	$95.70_{\pm 0.22}$	$93.75_{\pm 0.82}$	$95.19_{\pm 0.31}$	$94.31_{\pm 0.21}$	$95.67_{\pm 0.31}$	$95.78_{\pm 0.50}$	$95.82_{\pm 0.52}$
Church	Snow mountain	$16,000 \times 12$	$65.30_{\pm 0.65}$	$83.72_{\pm 1.41}$	$78.18_{\pm 0.38}$	$71.84_{\pm 0.96}$	$92.03_{\pm 0.57}\bullet$	$83.89_{\pm 0.54}$	$86.48_{\pm 0.78}$	$87.23_{\pm 0.69}$	$86.73_{\pm 0.99}$
Dragonfly	Butterfly	$16,000 \times 12$	$69.85_{\pm 0.44}$	$94.05_{\pm 0.26}\bullet$	$73.23_{\pm 0.32}$	$74.96_{\pm 0.72}$	$75.31_{\pm 0.67}$	$71.40_{\pm 0.88}$	$75.82_{\pm 1.30}$	$77.77_{\pm 0.60}$	$76.49_{\pm 0.48}$
Flower	Tree	$16,000 \times 12$	$84.89_{\pm 0.21}$	$90.59_{\pm 0.67}$	$94.59_{\pm 0.33}$	$83.48_{\pm 0.86}$	$95.11_{\pm 0.26}$	$80.41_{\pm 0.62}$	$95.05_{\pm 0.60}\bullet$	$94.83_{\pm 0.46}$	$94.59_{\pm 0.54}$
Field	Sailing	$16,000 \times 12$	$96.00_{\pm 0.18}$	$93.20_{\pm 1.25}$	$97.99_{\pm 0.18}$	$90.41_{\pm 0.94}$	$98.41_{\pm 0.18}$	$98.45_{\pm 0.10}$	$98.63_{\pm 0.23}\bullet$	$98.60_{\pm 0.29}$	$98.59_{\pm 0.16}$
Fireworks	Dusk	$16,000 \times 12$	$85.78_{\pm 0.58}$	$93.55_{\pm 0.46}$	$91.15_{\pm 0.40}$	$85.60_{\pm 0.86}$	$94.71_{\pm 0.29}$	$80.72_{\pm 0.33}$	$94.84_{\pm 0.68}$	$95.03_{\pm 0.71}\bullet$	$94.60_{\pm 0.45}$
Great Wall	Car	$16,000 \times 12$	$75.03_{\pm 0.63}$	$84.06_{\pm 0.76}$	$81.96_{\pm 0.33}$	$79.24_{\pm 0.62}$	$87.50_{\pm 0.49}\bullet$	$82.10_{\pm 0.50}$	$87.23_{\pm 0.88}$	$86.30_{\pm 1.00}$	$85.78_{\pm 0.84}$
Motorcycle	Big Ship	$16,000 \times 12$	$92.30_{\pm 0.26}$	$90.66_{\pm 0.38}$	$92.87_{\pm 0.17}$	$90.35_{\pm 0.34}$	$91.14_{\pm 0.37}$	$87.06_{\pm 0.33}$	$92.97_{\pm 0.33}\bullet$	$92.71_{\pm 0.34}$	$92.45_{\pm 0.51}$
	Average		$86.21_{\pm 0.76}$	$89.38_{\pm 0.84}$	$89.91_{\pm 0.60}$	$85.81_{\pm 1.09}$	$91.91_{\pm 0.59}$	$88.37_{\pm 0.74}$	$91.98_{\pm 0.82}$	$92.28_{\pm 0.75}\bullet$	$91.95_{\pm 0.71}$
	Mean rank		6.70	6.15	5.35	7.70	3.95	6.45	2.90	2.40$\bullet$	3.40

They are, however, two fundamentally different approaches. In comparison to the other approaches, the remaining columns reveal that ELDB is statistically considerably better (including the same type of algorithm BAMIC). In other words, ELDB outperforms the state-of-the-art MIL mapping methods in most cases.

E. Parameter Analysis

Figures 4-5 show the experimental results of parameter analysis on two types of datasets, including the image retrieval and the text classification tasks. The symbols "k", "j" and "s" denote the used single-instance classifier kNN, J48 and SVM respectively; "a" and "r" denote the addition and replacement modes respectively; and "g", "p", "n" and "b" denote four bag initialization modes respectively. Specially, the black long solid line represents the upper and lower limits of the F1-measure value for the current experiment.

For all of these images, the abscissa represents the proportion of discriminative bag selection, whereas the ordinate represents F1-measure values. The following summarizes the impact of parameters on ELDB:

1) **Action modes**: Action "a" performs better in terms of classification performance than action "r". However, the complexity of "a" is greater than that of "r" with the given basic dBagSet.
2) **Distance function**: The average Hausdorff distance and Eq.(5) yield approximately the same number of optimal results on image retrieval and text datasets.
3) **Bag selection mode**: The effect of the global mode "g" is better when the number of discriminative bags ψ is small. As the ψ increases, there is no significant difference between four modes.
4) **Classifier**: kNN and SVM are more suitable for these datasets than J48.

TABLE V

TWO-TAILED t-TEST RESULTS FOR ELDB *vs.* FOUR TYPES MAPPING-BASED MIL METHODS ON A TOTAL OF EIGHTEEN DATASETS SHOWN IN TABLE III. THE PROPOSED BAG-BASED AELDB AND RELDB ARE DENOTED BY AE AND RE, RESPECTIVELY. THE STATISTIC-BASED SIMPLE-MI IS ABBREVIATED AS SM. THE KERNEL-BASED MIFV IS DENOTED BY FV. THE INSTANCE-BASED STABLEMIL, MIVLAD, MILFM AN MILDM ARE DENOTED BY ST, VL, FM AND DM, RESPECTIVELY. THE BAG-BASED BAMIC IS DENOTED BY BA.

Datasets	AE-RE	AE-SM	AE-ST	AE-FV	AE-VL	AE-FM	AE-DM	AE-BA
Musk1	$3.19e-02$	$3.56e-05$	$4.38e-02$	$7.10e-04$	$4.56e-05$	$9.23e-01$	$1.31e-05$	$4.67e-07$
Musk2	$2.09e-01$	$1.76e-06$	$1.11e-03$	$9.33e-01$	$9.71e-08$	$4.75e-08$	$2.64e-04$	$5.54e-06$
Mutagenesis1	$2.46e-01$	$9.27e-05$	$7.80e-08$	$1.27e-06$	$6.55e-02$	$3.87e-05$	$9.82e-02$	$2.07e-11$
Mutagenesis2	$7.49e-01$	$1.76e-02$	$1.28e-03$	$2.42e-01$	$2.21e-01$	$5.25e-01$	$1.95e-01$	$4.46e-05$
Elephant	$1.01e-03$	$2.31e-11$	$6.35e-06$	$7.99e-08$	$2.02e-09$	$1.62e-04$	$1.82e-02$	$8.50e-03$
Fox	$3.83e-01$	$2.66e-07$	$1.18e-03$	$2.27e-07$	$2.88e-05$	$1.63e-06$	$5.01e-01$	$3.85e-04$
Tiger	$8.73e-01$	$4.52e-08$	$1.07e-02$	$1.40e-06$	$1.70e-09$	$2.48e-02$	$4.41e-04$	$4.05e-04$
Messidor	$1.89e-07$	$4.10e-01$	$8.90e-03$	$4.58e-03$	$4.57e-05$	$7.77e-03$	$6.28e-02$	$3.06e-11$
Alt.atheism	$2.03e-02$	$1.87e-23$	$1.77e-08$	$8.99e-16$	$8.17e-02$	$1.97e-21$	$9.85e-12$	$8.72e-02$
Comp.graphics	$9.30e-01$	$2.54e-21$	$5.50e-07$	$2.21e-06$	$8.17e-01$	$4.41e-12$	$9.33e-18$	$2.22e-05$
Comp.os.ms	$3.13e-01$	$4.21e-15$	$9.37e-08$	$5.98e-10$	$3.19e-03$	$3.73e-16$	$1.06e-11$	$5.47e-02$
Comp.sys.mac	$8.32e-04$	$1.96e-20$	$3.90e-08$	$5.19e-08$	$6.55e-02$	$3.87e-22$	$7.78e-12$	$9.43e-06$
Misc.forsale	$5.65e-01$	$1.14e-17$	$6.64e-08$	$1.73e-10$	$5.78e-03$	$5.45e-14$	$9.33e-15$	$2.38e-03$
Rec.sport.baseball	$3.66e-02$	$3.31e-21$	$1.83e-10$	$1.63e-14$	$1.72e-03$	$1.27e-18$	$1.61e-13$	$8.10e-02$
Rec.sport.hockey	$5.22e-01$	$8.14e-21$	$2.58e-13$	$6.35e-08$	$1.98e-04$	$1.33e-26$	$5.23e-16$	$8.40e-12$
Sci.electronics	$9.13e-01$	$1.87e-16$	$1.44e-09$	$4.76e-01$	$1.72e-01$	$1.87e-16$	$9.89e-22$	$3.55e-04$
Sci.med	$2.35e-01$	$1.20e-22$	$1.06e-09$	$1.61e-12$	$2.26e-01$	$6.02e-15$	$8.26e-16$	$1.29e-03$
Sci.space	$5.56e-02$	$1.04e-21$	$2.89e-09$	$2.19e-11$	$9.99e-04$	$7.13e-23$	$1.66e-21$	$5.79e-01$
Accept / Reject	12/6	1/17	0/18	3/15	7/11	2/16	4/14	4/14
Datasets	RE-AE	RE-SM	RE-ST	RE-FV	RE-VL	RE-FM	RE-DM	RE-BA
Musk1	$3.19e-02$	$6.86e-06$	$3.06e-01$	$3.71e-05$	$2.42e-06$	$5.81e-02$	$8.71e-07$	$1.42e-07$
Musk2	$2.09e-01$	$2.81e-06$	$1.19e-04$	$4.16e-01$	$1.10e-07$	$9.92e-08$	$5.37e-04$	$3.89e-05$
Mutagenesis1	$2.46e-01$	$4.41e-08$	$1.63e-06$	$2.88e-07$	$2.37e-01$	$4.60e-05$	$3.78e-03$	$2.82e-11$
Mutagenesis2	$7.49e-01$	$7.32e-05$	$1.31e-03$	$1.46e-01$	$1.50e-01$	$5.33e-01$	$5.89e-02$	$9.77e-09$
Elephant	$1.01e-03$	$4.86e-09$	$3.30e-06$	$4.09e-09$	$7.81e-09$	$1.69e-06$	$6.11e-05$	$5.33e-05$
Fox	$3.83e-01$	$4.54e-06$	$6.72e-04$	$4.72e-06$	$2.33e-05$	$1.90e-05$	$7.18e-01$	$4.40e-04$
Tiger	$8.73e-01$	$7.96e-08$	$5.26e-02$	$2.19e-06$	$3.53e-09$	$2.11e-02$	$4.26e-04$	$4.38e-04$
Messidor	$1.89e-07$	$7.90e-07$	$7.03e-01$	$2.00e-07$	$9.27e-04$	$1.12e-07$	$2.71e-05$	$7.83e-12$
Alt.atheism	$2.03e-02$	$9.31e-23$	$1.49e-08$	$8.25e-15$	$5.33e-01$	$1.05e-20$	$1.34e-11$	$2.37e-04$
Comp.graphics	$9.30e-01$	$1.39e-21$	$2.48e-08$	$1.97e-06$	$7.49e-01$	$7.75e-12$	$2.46e-17$	$1.06e-05$
Comp.os.ms	$3.13e-01$	$6.03e-20$	$7.31e-08$	$2.66e-10$	$9.46e-03$	$4.85e-14$	$2.50e-10$	$1.92e-01$
Comp.sys.mac	$8.32e-04$	$1.55e-18$	$1.51e-08$	$7.62e-03$	$5.19e-01$	$2.49e-19$	$2.47e-12$	$7.06e-07$
Misc.forsale	$5.65e-01$	$1.42e-17$	$1.31e-07$	$4.37e-10$	$1.12e-02$	$6.76e-15$	$6.66e-15$	$2.01e-03$
Rec.sport.baseball	$3.66e-02$	$2.28e-21$	$2.90e-08$	$1.73e-14$	$2.58e-02$	$1.27e-15$	$1.34e-11$	$2.41e-05$
Rec.sport.hockey	$5.22e-01$	$1.36e-21$	$1.37e-13$	$9.63e-09$	$9.39e-05$	$5.56e-21$	$7.70e-20$	$2.07e-08$
Sci.electronics	$9.13e-01$	$4.10e-16$	$5.39e-09$	$5.72e-01$	$2.30e-01$	$4.10e-16$	$1.61e-20$	$8.79e-04$
Sci.med	$2.35e-01$	$8.67e-25$	$3.63e-09$	$3.78e-11$	$1.63e-02$	$2.85e-17$	$1.88e-13$	$6.52e-04$
Sci.space	$5.56e-02$	$3.36e-25$	$8.46e-10$	$2.79e-10$	$1.70e-02$	$1.92e-25$	$2.29e-20$	$1.23e-02$
Accept / Reject	12/6	0/18	3/15	3/15	6/12	2/16	2/16	1/17

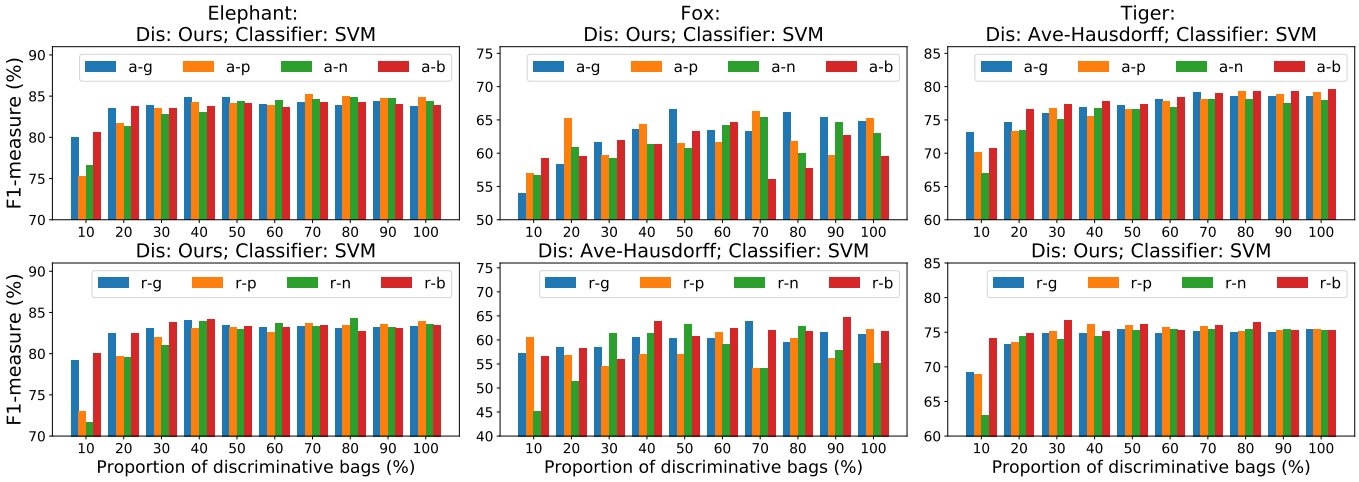

Fig. 4. Parameter analysis of ELDB with different proportion of discriminative bags, best distance functions, best classifier and four bag initialization modes for image classification task: Elephant, fox and tiger.

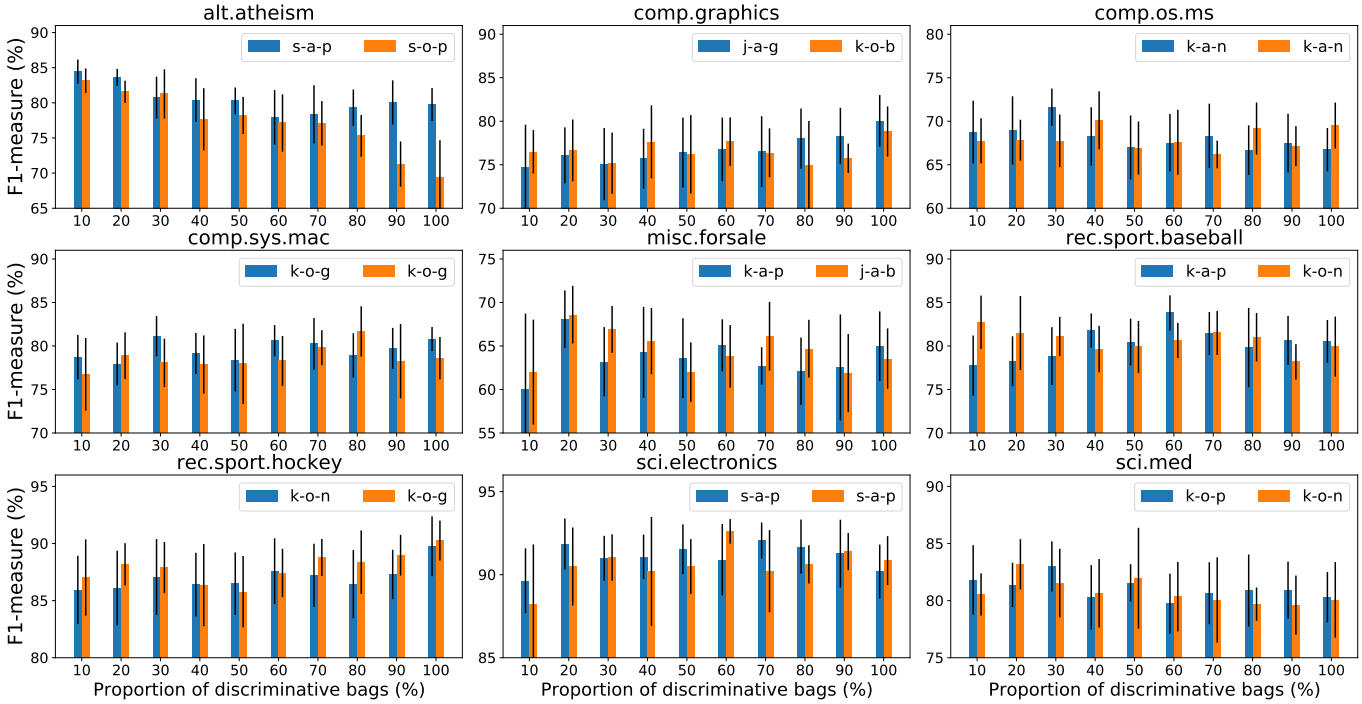

Fig. 5. Parameter analysis of ELDB with different proportion of discriminative bags, best distance functions, best classifier and best bag initialization mode for text classification task: Alt.atheism, comp.graphics, comp.os.ms and so on.

TABLE VI
TOTAL CPU RUNTIME OF ONE TIME 10CV FOR THE COMPARED ALGORITHMS ON EIGHT MIL CLASSIFICATION DATASETS (MEASURED IN MILLISECONDS).

Datasets	(d, n, N)	Simple-MI	StableMIL	miFV	miVLAD	MILFM	MILDM	BAMIC	**aELDB**	**rELDB**
Musk1	$(166, 476, 92)$	55.06	7501.83	1524.38	382.35	5379.90	3522.21	721.66	1325.20	766.69
Musk2	$(166, 6598, 102)$	61.06	166306.33	4744.31	915.83	506113.52	532734.74	958.87	2204.49	1279.38
Mutagenesis1	$(7, 10486, 188)$	39.03	> 1000000	3979.62	1123.02	> 1000000	> 1000000	1723.57	4587.94	3057.24
Mutagenesis2	$(7, 2132, 42)$	22.02	46816.61	1218.10	420.39	47659.60	55366.38	943.87	1437.93	530.12
Bird&ship	$(12, 3200, 200)$	42.04	131970.47	1918.75	589.54	108786.59	106786.23	2536.31	11289.05	8542.93
Beetle&Horse	$(12, 16000, 2000)$	310.29	733659.69	8132.39	2084.90	> 1000000	> 1000000	29030.42	71802.32	55473.77
Alt.atheism	$(200, 5443, 100)$	52.06	58900.60	4576.17	823.76	300652.63	263707.36	1035.94	960.32	944.21
Component	$(200, 36894, 3130)$	2643.41	> 1000000	73009.44	11767.72	> 1000000	> 1000000	32957.00	60414.98	46626.24

F. Time Complexity and Efficiency Comparisons

We compare the time complexity and runtime of ELDB to that of seven rival algorithms. For ELDB, the weighted ensemble model construction costs $O(dN^2)$, where d is the dimension and N is the size of the dataset. By contrast, Simple-MI costs $O(dN)$, StableMIL costs $O(dn^2)$, miFV costs $O(dn)$, miVLAD costs $O(dn)$, MILFM costs $O(dn^2)$, MILDM costs $O(dn^2)$, BAMIC costs $O(dN^2)$, where n is the size of instance space. Normally, $N \ll n$, so $O(N^2) \ll O(n^2)$. To verify the theoretical analysis, we compare the CPU runtime on eight datasets for these methods, as shown in Table VI. The results show that our method has similar runtime with miFV and BAMIC algorithms. Besides, even on the small-scale dataset, the runtime of StableMIL, MILFM and MILDM are relatively large.

G. Large-scale Datasets

We make recommendations for ELDB parameter settings for large-scale datasets based on the above experimental results. For the action mode, "a" performs better than "r" in terms of the classification performance. However, the dimension of the mapping vector based on "r" is relatively low. Therefore, the action mode can use "r" on the large-scale datasets; otherwise "a". In addition, the distance function can use the Eq. (5). The bag selection mode can use the global mode "g". The number of discriminative bags ψ can be set to the number of bags N times the proportion of basic dataset α, where α can be set to 0.75. The size of the batch can be set to $N \times (1-\alpha)/2$. The single-instance classifier can use kNN or SVM.

As shown in Table VII, we only compare rELDB with four rival algorithms, because StableMIL, MILFM and MILDM have relatively high time complexity. The experimental results show that rELDB has good performance on large datasets, particularly on function dataset.

H. Discussions

We can now answer the four questions proposed at the start of this section.

1) ELDB is more accurate than popular mapping-based MIL supervised classification algorithms, such as

TABLE VII
F1-MEASURE WITH STANDARD DEVIATIONS ON MIL BIOCREATIVE DATASETS.

Datasets	$n \times d$	Simple-MI	miFV	miVLAD	BAMIC	**rELDB**
Component	$36,894 \times 200$	$28.11_{\pm 0.74}$	$59.0_{\pm 0.78}$	$69.29_{\pm 0.76}\bullet$	$67.43_{\pm 1.03}$	$68.78_{\pm 0.74}$
Function	$55,536 \times 200$	$20.21_{\pm 0.75}$	$62.4_{\pm 0.66}$	$70.64_{\pm 0.79}$	$74.25_{\pm 0.80}$	$75.43_{\pm 0.39}\bullet$
Process	$118,417 \times 200$	$22.73_{\pm 0.45}$	$53.6_{\pm 0.28}$	$70.78_{\pm 0.44}\bullet$	$67.40_{\pm 1.30}$	$67.39_{\pm 0.70}$

Simple-MI and MILDM. This is validated by Tables III and IV.

2) ELDB has the best overall classification performance and second stability. The disadvantage of ELDB is that the weighted ensemble model becomes more complicated as the dBagSet in the model is updated. Therefore, ELDB has a higher time complexity than the similar method BAMIC. At present, we use reasonable parameter settings to alleviate this problem.
3) ELDB is insensitive to parameter settings, such as the action mode and bag selection mode. This is validated by Figs. 4 and 5.
4) ELDB is an effective algorithm. This is validated by Table VI.

V. CONCLUSION AND FURTHER WORK

This paper designs the ELDB algorithm with the new discriminative bag selection method and classifier ensemble strategy for supervised MIL classification tasks. In theory, the bag selection results are divided into two categories: Basic and updated dBagSets. The basic dBagSet is generated by taking into account the dataset's spacial and label distribution. Based on this dBagSet, the bags can be mapped into the new feature space and be easily separated from each other. In addition, we design the self-reinforcement mechanism to enhance the mapping-ability of the basic dBagSet and obtain the updated dBagSet. Naturally, this process can be gradually strengthened through updates.

The classifier ensemble strategy is designed to make full use of the generated dBagSet. For each dBagSet, we can train a single-instance classification model and assign a weight by considering its contribution. At last, all of these models can be combined to form a weighted ensemble model. The experimental results on thirty-eight datasets show that ELDB outperforms the state-of-the-art MIL mapping methods in terms of F1-measure and statistical significance. The parameter analysis shows how parameters affect ELDB and suggests parameter settings for large-scale datasets. The time complexity further demonstrates that ELDB provides an effective tradeoff between runtime efficiency and classification performance.

The following topics deserve further investigation:

1) A more effective self-reinforcement mechanism. Currently, only addition and replacement are available in action modes. In the future, we will examine more flexible learning modes.
2) A better weight assignation scheme. We currently use the model's classification performance as its weight. However, the model's weight may be related to the overall distinguishability of its corresponding dBagSet.
3) Classification performance improvement on datasets such as mutagenesis2, process and tiger.

REFERENCES

[1] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *AI*, vol. 89, no. 1-2, pp. 31–71, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370296000343

[2] J. Wu, S. R. Pan, X. Q. Zhu, C. Q. Zhang, and X. D. Wu, "Multi-instance learning with discriminative bag mapping," *TKDE*, vol. 30, no. 6, pp. 1065–1080, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8242668

[3] M.-L. Zhang and Z.-H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *APIN*, vol. 31, no. 1, pp. 47–68, 2009. [Online]. Available: https://link.springer.com/article/10.1007/s10489-007-0111-x#citeas

[4] Y. X. Chen, J. B. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *TPAMI*, vol. 28, no. 12, pp. 1931–1947, Dec. 2006. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1717454

[5] S. Conjeti, M. Paschali, A. Katouzian, and N. Navab, "Deep multiple instance hashing for scalable medical image retrieval," in *MICCAI*, 2017, pp. 550–558. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-66179-7_63

[6] G.-H. Liu, J.-Y. Yang, and Z. Y. Li, "Content-based image retrieval using computational visual attention model," *Pattern Recognit.*, vol. 48, no. 8, pp. 2554–2566, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320315000539

[7] X.-S. Wei, H.-J. Ye, X. Mu, J. X. Wu, C. H. Shen, and Z.-H. Zhou, "Multiple instance learning with emerging novel class," *TKDE*, pp. 1–1, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8896009/

[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. [Online]. Available: https://doi.org/10.1145/1553374.1553534

[9] B.-C. Xu, K. M. Ting, and Z.-H. Zhou, "Isolation set-kernel and its application to multi-instance learning," in *KDD*, 2019, pp. 941–949. [Online]. Available: https://doi.org/10.1145/3292500.3330830

[10] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, and S. Vluymans, *Multiple instance learning foundations and algorithms*, 2016. [Online]. Available: https://www.springer.com/gp/book/9783319477589

[11] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *NIPS*, 1998, pp. 570–576. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3008904.3008985

[12] S. Vluymans, D. S. Tarragó, Y. Saeys, C. Cornelis, and F. Herrera, "Fuzzy multi-instance classifiers," *TFS*, vol. 24, no. 6, pp. 1395–1409, 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7378303

[13] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *ICML*, 2002, pp. 179–186.

[14] Y. X. Chen and J. Z. Wang, *Categorization by learning and reasoning with regions*, 2004. [Online]. Available: https://doi.org/10.1007/1-4020-8035-2_6

[15] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *AI*, vol. 201, no. 4, pp. 81–105, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370213000581

[16] X.-S. Wei, J. X. Wu, and Z.-H. Zhou, "Scalable multi-instance learning," in *ICDM*, 2014, pp. 1037–1042. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7023443

[17] X.-S. Wei, J. X. Wu and Z.-H. Zhou, "Scalable algorithms for multi-instance learning," *TNNLS*, vol. 28, no. 4, pp. 975–987, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7398097

[18] Z.-H. Zhou and M.-L. Zhang, "Solving multi-instance problems with classifier ensemble based on constructive clustering," *KAIS*, vol. 11, pp. 155–170, Feb. 2007. [Online]. Available: https://doi.org/10.1007/s10115-006-0029-3

[19] H. N. Yuan, M. Fang, and X. Q. Zhu, "Hierarchical sampling for multi-instance ensemble learning," *TKDE*, vol. 25, no. 12, pp. 2900–2905, 2013. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6384531

[20] G. Chen, M. Giuliani, D. Clarke, A. Gaschler, and A. Knoll, "Action recognition using ensemble weighted multi-instance learning," in *ICRA*, 2014, pp. 4520–4525. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6907519

[21] Z. Y. Fu, A. Robles-Kelly, and J. Zhou, "MILIS: Multiple instance learning with instance selection," *TPAMI*, vol. 33, no. 5, pp. 958–977, 2011. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5557878

[22] R. C. Hong, M. Wang, Y. Gao, D. C. Tao, X. L. Li, and X. D. Wu, "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 669–680, 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6542696

[23] C. Liu, X. Xu, and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 45, no. 3, pp. 385–398, 2015. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6918520

[24] W. J. Zhang, L. Liu, and J. Y. Li, "Robust multi-instance learning with stable instances," *CoRR*, vol. abs/1902.05066, pp. 1682–1689, 2020. [Online]. Available: http://arxiv.org/abs/1902.05066

[25] A. Srinivasan, S. Muggleton, and R. King, "Comparing the use of background knowledge by inductive logic programming systems," in *ILP*, 1995, pp. 199–230. [Online]. Available: http://www.doc.ic.ac.uk/~shm/Papers/prg-tr-9-95.ps.gz

[26] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2002, pp. 561–568. [Online]. Available: http://papers.nips.cc/paper/2232-support-vector-machines-for-multiple-instance-learning

[27] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," *TPAMI*, vol. 30, no. 6, pp. 985–1002, 2008. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4420087

[28] E. Decencière, X. W. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay *et al.*, "Feedback on a publicly distributed image database: The messidor database," *Image Anal. Stereol.*, vol. 33, no. 3, pp. 231–234, 2014. [Online]. Available: https://www.ias-iss.org/ojs/IAS/article/view/1155

[29] M. Kandemir and F. A. Hamprecht, "Computer-aided diagnosis from weak supervision: A benchmarking study," *Comput. Med. Imaging Graphics*, vol. 42, pp. 44–50, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0895611114001852

[30] S. Ray and M. Craven, "Learning statistical models for annotating proteins with function information using biomedical text," *BMC Bioinf.*, vol. 6, no. 1, pp. 18–18, 2005. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-6-S1-S18

[31] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison," in *ICML*, 2005, pp. 697–704. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1102351.1102439

[32] X.-S. Wei and Z.-H. Zhou, "An empirical study on image bag generators for multi-instance learning," *Mach. Learn.*, vol. 105, no. 2, pp. 155–198, 2016. [Online]. Available: https://link.springer.com/article/10.1007/s10994-016-5560-1

[33] P. Reutemann, B. Pfahringer, and E. Frank, "A toolbox for learning from relational data with propositional and multi-instance learners," in *AJCAI*, 2005, pp. 1017–1023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-30549-1_95

[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *BMC Bioinf.*, vol. 60, no. 1, pp. 91–110, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[35] J. J. He, H. Gu, and Z. L. Wang, "Bayesian multi-instance multi-label learning using gaussian process prior," *Mach. Learn.*, vol. 88, no. 1C2, pp. 273–295, Jul. 2012. [Online]. Available: https://doi.org/10.1007/s10994-012-5283-x

[36] J. Amores, "MILDE: Multiple instance learning by discriminative embedding," *Mach. Learn.*, vol. 42, pp. 381–407, Feb. 2015. [Online]. Available: https://doi.org/10.1007/s10115-013-0711-1

[37] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *ICML*, 2001, pp. 425–432. [Online]. Available: https://openreview.net/forum?id=Hy-LyoWdWH

Mei Yang is an associate professor with School of Computer Science, Southwest Petroleum University. Her current research interests include multi-instance learning and recommender systems.

Yu-Xuan Zhang is a postgraduate student with School of Computer Science, Southwest Petroleum University, Chengdu, China. His current research interests include multi-instance learning and deep learning.

Xizhao Wang served in Hebei University as a professor and the dean of school of Mathematics and Computer Sciences before 2014. After 2014 Prof. Wang's worked as a professor in Big Data Institute of Shenzhen University. Prof. Wang's major research interests include uncertainty modeling and machine learning for big data. Prof. Wang has edited 10+ special issues and published 3 monographs, 2 textbooks, and 200+ peer-reviewed research papers. As a Principle Investigator (PI) or co-PI, Prof. Wang has completed 30+ research projects. Prof. Wang has supervised more than 100 M.phil. and Ph.D. students. Prof. Wang is an IEEE Fellow, the previous BoG member of IEEE SMC society, the chair of IEEE SMC Technical Committee on Computational Intelligence, the Chief Editor of Machine Learning and Cybernetics Journal, and associate editors for a couple of journals in the related areas. He was the recipient of the IEEE SMCS Outstanding Contribution Award in 2004 and the recipient of IEEE SMCS Best Associate Editor Award in 2006. He is the general Co-Chair of the 2002C2017 International Conferences on Machine Learning and Cybernetics, cosponsored by IEEE SMCS. Prof. Wang was a distinguished lecturer of the IEEE SMCS.

Fan Min (M09) received the M.S. and Ph.D. degrees from the School of Computer Science and Engineering, University of Electronics Science and Technology of China, Chengdu, China, in 2000 and 2003, respectively. He visited the University of Vermont, Burlington, Vermont, from 2008 to 2009. He is currently a professor with Southwest Petroleum University, Chengdu. He has published more than 100 refereed papers in various journals and conferences, including the Information Sciences, International Journal of Approximate Reasoning, and Knowledge-Based Systems. His current research interests include data mining, recommender systems, active learning and granular computing.