Generative Label Enhancement via Renormalization Group

Chao Tan, *Member, IEEE,* Sheng Chen, *Life Fellow, IEEE,* Mengjiao Kai, Shangce Gao, *Senior Member, IEEE,* and Xin Geng, *Senior Member, IEEE*

***Abstract*—Applying the label distribution learning to solve the challenging label ambiguity problem requires datasets with label distributions. Unfortunately, most real-world datasets only contain logical or binary labels, and manual annotation of label distributions is very costly. Label enhancement (LE) is an effective approach of 'enhancing' binary labels into the related label distributions. However, LE algorithms in the existing literature cannot fully mine the feature space to extract the intrinsic structural information, which limits their performance. To tackle this critical issue, we design a generative LE model that uses a renormalization group (RG) based probabilistic graph model to perform feature extraction on data, called RG4LE. Unlike existing methods, the restricted Boltzmann machine (RBM) we adopted undergoes preliminary unsupervised training to efficiently extract high-level features. Furthermore, we introduce label information through label propagation that explicitly models semantic relationships between labels via a learned state transition matrix, which aligns the feature space and label space to capture richer structural information. We also incorporate a mutual information maximization mechanism which provides supervisory information for the RBM, ensures that the extracted features are maximally informative about the label distributions, facilitates bidirectional information flow, and mitigates information loss. Comprehensive experiments using real-world datasets confirm superiority of our method over best existing LE models.**

***Index Terms*—Label enhancement, Neural Network, Renormalization group, Restricted Boltzmann machine.**

I. Introduction

In recent years, machine learning has become one of the most active areas of science and technology [1], [2]. Its practical impact can be seen in text-oriented applications [3] and intelligent transportation systems [4], as well as in financial data analysis [5] and computer-assisted medical diagnosis [6]. Real-world machine learning tasks often encounter multi-label samples, where each example is assigned with multiple labels, breaking the restriction in conventional single-label learning that an example can only belong to one category. Multi-label learning (MLL) is particularly effective in the areas where ambiguous data are common [7], including image recognition [8], text classification [9], and target detection [10].

In MLL, the association strength between various labels and an example is binarily categorized as relevant (typically denoted as '1') or irrelevant (typically denoted as '0'). Hence, these labels are termed 'logical labels'. In the category information annotated by logical labels, there is no differentiation between relevant labels or between irrelevant labels. Yet real-world tasks are often not so simple. In particular, for relevant labels, although they are related to categories of the data object, their 'relevance strength' tend to be significantly different. Representing all of them with the undifferentiated '1' in logical labels results in losing such important categorical information. In response to this prominent problem, label distribution learning (LDL) [11] was proposed. This learning paradigm adopts a probability-distribution like data structure, known as label distribution, to depict an example's categorical information, instead of using logical labels. Therefore, a label distribution explicitly expresses the association strength of each label to its assigned data object using a continuous 'degree'. This naturally meets the need for varying label strengths. Relevant and irrelevant labels are no longer rigidly differentiated in the label distribution. Generally, a threshold can be set: labels with degrees above it are deemed relevant, otherwise irrelevant.

The prerequisite for applying the LDL paradigm is that each sample in the training set needs to be annotated with a label distribution. Although in many applications, the data itself inherently carries a label distribution, this label distribution is unknown or inexplicit, and annotating label distributions to these applications encounters great difficulty. On one hand, assigning degrees to all labels for each example leads to high annotation costs. On the other hand, there is often no objectively quantified criterion for the degrees describing label relevance to examples. Hence, abundant ambiguous real-world data continues to be annotated with simple logical labels. Nevertheless, the supervisory information encoded by logically labeled data can be viewed as being governed by an underlying label distribution, even though this distribution is not directly observed. By recovering such an implicit label distribution automatically in some systematical way, LDL can then be applied to these insufficiently labeled logical-tag data without extra data annotation burden, thereby expanding the applicability of LDL.

This work was supported by the National Natural Science Foundation of China (Grants 62476135, U24A20324), by the Jiangsu Science Foundation (BK20243012, BG2024036), and by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grants JP25K21298, JP25K24386, JP26K03333, JP26K23843, and JP25K03179. (*Corresponding author: Chao Tan*).

C. Tan (73022@njnu.edu.cn) and M. Kai (242202042@njnu.edu.cn) are with the School of Computer and Electronic Information/School of Artificial Intelligence, Nanjing Normal University, Nanjing 210023, China.

S. Chen (sqc@ecs.soton.ac.uk) is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

S. Gao (gaosc@eng.u-toyama.ac.jp) is with the Faculty of Engineering, University of Toyama, Gofuku 3190, Toyama, Japan.

X. Geng (xgeng@seu.edu.cn) is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China.

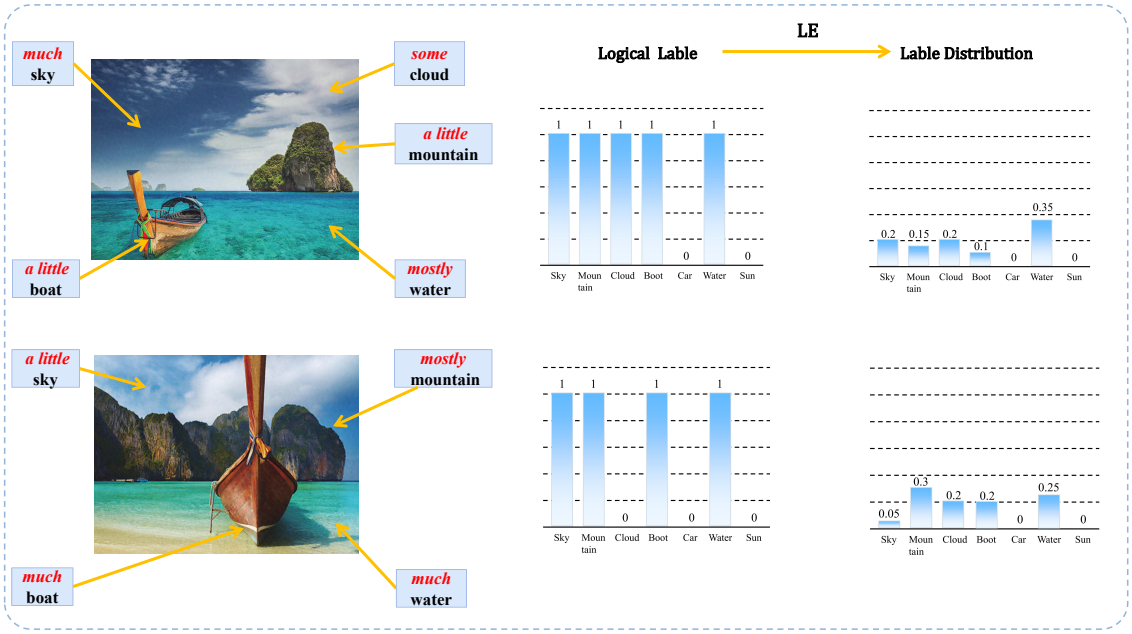


Fig. 1. Illustration of label enhancement task.

This recovery process 'enhances' the given simple binary labels of each sample into label distributions taking values in the interval $[0, 1]$, which contain more categorical supervision information. Xu et al. [12] referred to this process as label enhancement (LE). Since logical labels are either 0 or 1, the logical label space can be expressed as $\mathcal{L}=\{0,1\}^c$ for the data annotated with c logical labels. It can be seen that all samples can only distribute at the vertices of the unit hypercube in this logical label space. By contrast, in the label distribution space obtained after LE, denoted by $\mathcal{D}$, each dimension represents a degree in $[0, 1]$, and therefore $\mathcal{D}=[0, 1]^c$. That is, samples in the label distribution space distribute within the unit hypercube. Compared to the original $\mathcal{L}$, $\mathcal{D}$ contains more categorical supervision information. This additional supervisory information is mainly derived from two aspects: the neighborhood structure among instances in the feature space and the dependency patterns among labels. To intuitively illustrate the core task addressed by LE, Fig. 1 provides an example. It shows how LE transforms logical labels (indicating simple presence/absence of objects like 'boat', 'mountain', 'water', 'sky') into a more informative label distribution. This distribution quantifies the degree or strength of association (e.g. 'much', 'mostly', 'a little', 'some') between the sample and each label, providing richer supervision for learning tasks. Therefore, the study on LE can not only provide strong support for expanding the applicability of the LDL paradigm, but also has important implications for exploring the essence of categorical supervision information.

Prior to the proposal of the LE concept, some methods, although not specifically targeting LE, could partially achieve its functionality. However, it has been found that the algorithms specifically designed for LE achieve significantly better performance compared to those existing methods not specifically designed for this purpose. This highlights the necessity of studying specialized LE algorithms. These LE algorithms are designed based on the intrinsic characteristics of the LE problem, such as the inherent generation mechanism of LE information, and they adapt to specific learning algorithms to improve the generalization performance.

Since samples are distribute at the vertices of the unit hypercube in the logical label space, existing LE algorithms primarily use graph models to represent the relationships between samples. They then mine the relevance between samples according to their topological relations in graph models. Based on the relevance information between examples, they further infer the correlations between labels. The inferred relationships are then used to 'change' the original binary labels into continuous label distributions. Specifically, the method in [12] jointly exploits sample features and the available logical-label annotations to estimate a distribution over labels for each instance. Earlier mainstream methods were based on embedding-based approaches, which learn an embedding space using sample feature information and label semantic information, and then achieve recognition of new samples in this embedding space through nearest neighbors. This paper will comprehensively consider sample feature information and label information to generate label distributions for unseen classes from seen class samples based on LE. These label distributions can then serve as supervision information to learn richer knowledge about unseen classes.

With the continuous advancement of neural network (NN) technologies, Sung et al. [13] designed the RNet model to learn a transferable deep measure for comparing the relationships between sample features and label semantic information, laying the foundation for subsequent embedding-based MLL research. Deep NN (DNN) [14] is adopted to extract image features, with its output layer dimension being consistent with that of the extracted image features. The feature concatenation module then fuses together the labels' semantic features in the embedding space and the sample feature information. The

fused features go through a relation module to compute the relationship between sample features and label semantic information. With the aid of label semantic information, generative models [15–17] generate sample features for new samples. These generated new sample features for unseen labels and the sample features of seen labels are jointly utilized to train the classification model. By introducing generated samples of unseen labels, the model's generalization capability to new samples is enhanced.

In this paper, we will consider generating sample label distributions of new samples based on LE, mining the label distributions of seen category samples with respect to new samples. During training, the generated label distributions serve as supervision information about new samples to transfer information regarding these categories. Moreover, based on the smoothness assumption [18], i.e., similar samples likely share the same labels, the topological information of the sample feature space is transferred to the label space to constrain the generated label distributions regarding new samples. In this way, the feature information of seen category samples is fully utilized and this information is transferred to new samples. Hence, based on the inherent generation mechanism of LE information and introduction of the label distributions for known samples as supervision information to learn new samples during the training process, our LE model enhances the model's cognition of new samples.

Many interdisciplinary studies [19–21] have demonstrated a connection between renormalization group (RG) theory and machine learning. Sante et al. [19] highlighted the resemblance between RG and DNN structures, where structural features are marginalized into hidden degrees of freedom. Mehta and Schwab [20] suggested that machine learning algorithms can employ schemes akin to generalized RG to learn features from data. To illustrate this viewpoint, the authors of [20] utilized restricted Boltzmann machines (RBMs) to establish a mapping between variational RG and deep learning architectures. More recently, RG4LDL applied an RBM-based RG framework to label distribution learning, demonstrating its effectiveness in extracting compact representations and improving label distribution prediction [22]. Real-world high-dimensional LE models face related complexity and optimization challenges due to redundant degrees of freedom. Motivated by the aforementioned progress, this paper further investigates how an RG-inspired mechanism can be adapted to the LE task.

The advantage of adopting an RG-inspired mechanism for LE lies in its compatibility with the goal of recovering continuous label distributions from binary labels. In LE, not all feature variations are equally useful for estimating the latent label distribution, and nuisance variations may distort the sample topology used for label propagation (LP). Conventional dimensionality reduction or feature extraction techniques usually preserve variance, reconstruction fidelity, or local geometry in a task-agnostic manner. By contrast, RG-inspired coarse graining maps the original sample variables $\boldsymbol{X}$ to hidden variables $\boldsymbol{H}$ by retaining relevant degrees of freedom and suppressing irrelevant ones. In RG4LE, this mapping is implemented by an RBM and is further guided by the label-distribution information obtained from LP and mutual information maximization, so that the learned representation is not only compact but also informative for label-distribution recovery.

RBM [23] is a stochastic NN typically trained by unsupervised learning and has its root in statistical mechanics. In RBM, an energy function measures the system's overall state, with its minimum corresponding to the most stable state. Statistical mechanics principles show that a physical probability distribution can be transferred into a suitable energy model, leveraging energy models' inherent properties. Our proposed generative LE model, called RG4LE, is an efficient RG-inspired framework for LE. It employs RBM-type NNs to model label probabilities conditioned on samples, learning latent representations that preserve maximal label correlation information. Crucially, these representations are optimized through bidirectional information flow between feature and label spaces.

Our RG4LE introduces dual innovations: we use RBM for initial unsupervised feature extraction, followed by a mutual information maximization mechanism that injects supervisory signals to align features with label distributions; and we introduce LP enhanced by a learned label-state transition matrix, which explicitly models semantic relationships between labels. This allows LP to leverage both feature-space topology and label-space semantics for probabilistic enhancement of logical labels. Our RG4LE is the first application of RBM theory to enhance logical labels by enriching their probability distributions. The feature space obtained after RBM training strengthens class supervision information by leveraging the topological structure between sample and label spaces. On the other hand, the LP algorithm better captures the manifold structure between samples to propagate labels, and therefore it enhances the original logical labels with a probabilistic interpretation. Our main contributions are given below.

1) This work incorporates an RBM into the LE framework to progressively learn compact representations that preserve task-relevant information while suppressing redundant feature variations. The RBM undergoes preliminary unsupervised training for feature extraction, and a mutual information mechanism subsequently provides supervisory signals to align features with label distributions. Our RG-inspired model excels at processing continuous label distributions. This constitutes the first application of RBM for logical-label-to-distribution enhancement.

2) RG4LE's significance lies in unifying three key components: label-state transition matrix explicitly modeling semantic label relationships; semantics-enhanced LP leveraging this matrix; and mutual information maximization reinforcing feature-distribution dependency. This design ensures accurate recovery of continuous label distributions from binary labels. Experiments using 13 real-world multilabel datasets further confirm the superiority of the proposed RG4LE, which consistently outperforms representative best existing baselines.

II. Related Works

A. Label Enhancement Learning

Various LE models can be grouped into two categories: algorithms based on fuzzy logic and algorithms based on

graph models. The fuzzy-based LE models, such as LE model using fuzzy clustering [24] and LE method with kernel model [25], derive soft membership values for individual label classes through fuzzy operations or clustering procedures. Although such soft assignments relax the binary nature of logical labels, they do not explicitly explain how the resulting memberships correspond to a valid label distribution. The graph-based LE algorithms, including label propagation (LP) based LE [26], manifold learning (ML) based LE [27], LE using graph Laplacian (GLLE) [12], LE based on sample correlation (LESC) [28], privileged LE for MLL (PLEML) [29] and LE via the preservation of positive and negative label relationships (LE-PNLR) [30], use graph models to model the samples' topological structure and establish the relationship between sample relevance and label relevance to enhance binary labels into continuous label distributions. Recent studies further explored the role of LE in MLL scenarios. For instance, SMLE [31] applies LE to semi-supervised MLL by combining pseudo-labeling, label propagation, and local label correlations, while interactive fusion LE [32] integrates the LE process with multi-label model training through interaction and alignment mechanisms.

Table I presents a comparison of various LE methods based on how many of the following LE functionalities they possess. Feature structure utilization (FSU) assesses whether the algorithm utilizes the structural information of features to enhance predictive performance. Smooth assumption (SA) refers to the assumption that neighboring samples have similar labels. This assumption helps to manage noise and improve model stability and generalization. Label correlation modeling (LCM) evaluates whether the algorithm models label correlations, which helps to boost classification accuracy. Label semantics modeling (LSM) focuses on whether the algorithm understands and applies label semantics in the learning process. Generative feature modeling (GFM) considers whether the algorithm uses generative models to better represent features, thereby enhancing its generative and expressive power. It can be seen that our RG4LE possesses the most LE functionalities.

B. Renormalization Group

The RG, a conceptual framework from theoretical physics, provides a formalized tool to explore the changes of physical systems at different scales. Changes across scales are defined as scale transformations. RG theory enables studying the variations of dynamical systems across different scales. Petermann [33] envisioned and conceptually introduced RG in quantum field theory (QFT), where divergences in computations leads to unphysical infinite results. Using RG, divergences are absorbed into fewer measurable quantities to obtain finite results. RG has similarities to DNN structure in that structural features are marginalized into hidden degrees of freedom [19], and it makes modeling closer to real essence.

TABLE I
COMPARISON OF LE METHODS BY LE FUNCTIONALITIES.

LE algorithm	FSU	SA	LCM	LSM	GFM
LP [26]	✓	✓			
ML [27]	✓	✓	✓		
GLLE [12]	✓	✓	✓		
PLEML [29]	✓	✓	✓		
LESC [28]	✓	✓	✓		
LE-PNLR [30]	✓	✓			
Our RG4LE	✓	✓	✓	✓	✓

RG is closely related to self-similarity. Mehta and Schwab [20] proposed that machine learning algorithms may employ a scheme analogous to a generalized RG to extract features from data. Specifically, RG bears a similarity to DNN structures, where structural features are marginalized into hidden degrees of freedom. In particular, lower-level features are fed through higher layers of DNN to generate more abstract higher-level features. In this feature extraction process, a DNN learns to ignore irrelevant features while retaining relevant ones progressively layer by layer. This coarse-graining process resembles RG, which extracts relevant features of a physical system by integrating short-distance degrees of freedom to describe large-scale phenomena. To illustrate this viewpoint, the work [20] established a mapping between variational RG and deep learning architectures based on the construction of a RBM.

However, identifying key degrees of freedom is not straightforward. Koch-Janusz and Ringel [34] proposed a model-agnostic, information-theoretic feature NN for real-space RG processes. Based on machine learning algorithms, the NN can identify relevant degrees of freedom within spatial regions and iteratively execute RG coarse-graining steps. The input data obeys a Boltzmann distribution and no further knowledge about microscopic details of the system is needed. The NN's parameters are learned through a training algorithm by calculating the mutual information between spatial separation regions.

C. Information-Theoretic Neural Network

By identifying relevant degrees of freedom and iteratively removing 'irrelevant' degrees of freedom, information-theoretic NNs are able to extract intricate information about relevant degrees of freedom between spatial layers, thereby capturing the self-similarity of the system near critical points and obtaining a universal, low-energy effective model.

Information-theoretic NNs provide a mechanism for extracting relevant degrees of freedom analogous to RG in particle physics. RG principle inspires more effective machine learning models that identify multi-scale relevant features and universal representations. Identifying task-relevant degrees of freedom, however, remains nontrivial. One possible solution is to employ an unsupervised neural model that progressively extracts the informative components of the input variables. Consider a system described by a collection of random variables $\boldsymbol{X} = \{\boldsymbol{x}_i\}$. A lower-dimensional set of coarse-grained variables $\boldsymbol{H} = \{\boldsymbol{h}_j\}$ can be introduced to summarize the information contained in $\boldsymbol{X}$. The dependence of the 'hidden' variables $\boldsymbol{H}$ on the original system variables $\boldsymbol{X}$ is learned by optimizing an information-theoretic criterion, which can naturally be implemented using a NN. The resulting conditional model is represented by a Boltzmann distribution parameterized through an energy function $E(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$

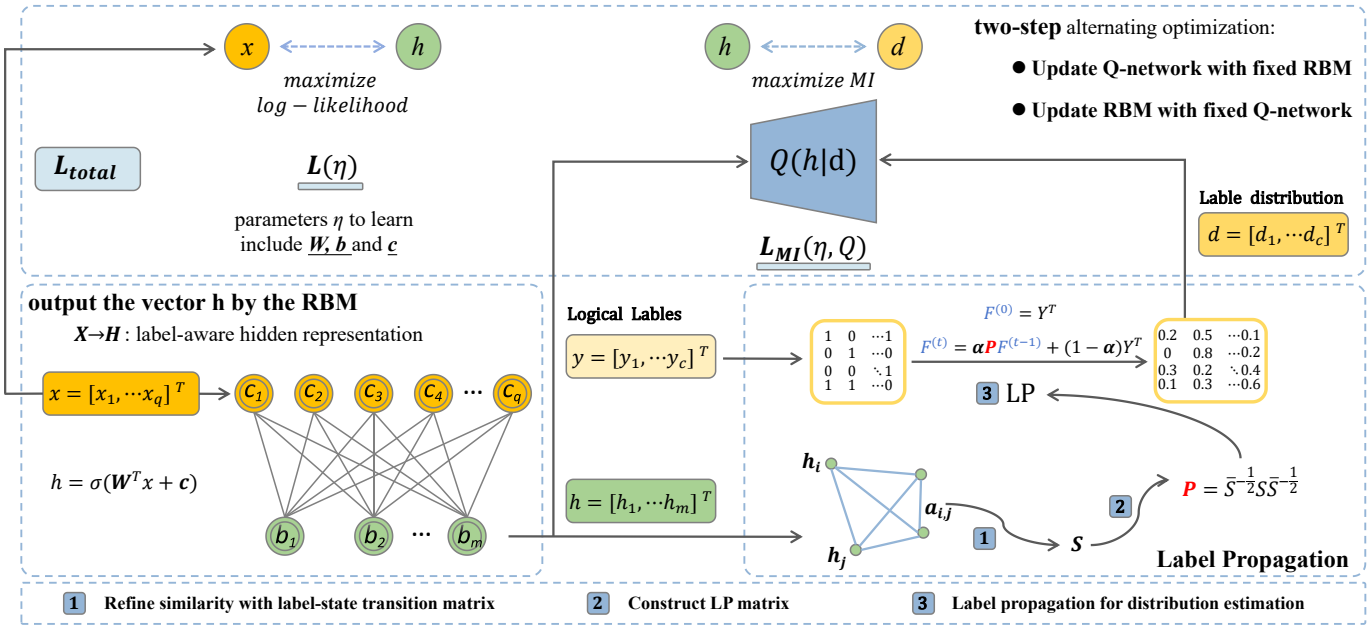


Fig. 2. Proposed RG4LE consists of three key components: 1) Feature extraction via RBM to capture high-level representations; 2) Label propagation enhanced by a learned label-state transition matrix that models semantic label relationships; and 3) Mutual information maximization aligning the feature and label spaces bidirectionally.

denotes the NN's trainable parameters. From an information-theoretic perspective, mutual information is preserved under invertible reparameterizations of $\boldsymbol{X}$ or $\boldsymbol{H}$, while non-invertible transformations may discard information. This property motivates the use of an RBM to learn compact hidden variables that retain information relevant to the subsequent label-distribution recovery task.

III. Proposed RG4LE

Essentially, LE uses the feature information and logical labels provided in the dataset to infer the label distribution for each sample. We utilize the sample feature information mined by RBM and the logical label information of known samples to learn an embedding space. In this space, unseen class samples are identified through nearest neighbor search. In this way, by unsupervised NN feature learning on samples with both known and unknown logical labels, the general MLL problem can be transformed into a supervised classification problem. Moreover, via LE, the label distribution information generated from the logical labels of known samples is used to enhance the generalization capability of the classification model on learning unknown samples.

A. Label Enhancement Problem Formulation

The set of n feature samples is defined by $\boldsymbol{X} = \left[\boldsymbol{x}_1\ \boldsymbol{x}_2 \cdots \boldsymbol{x}_n\right] \in \mathbb{R}^{q\times n}$, where q is the feature dimension. The binary label vector of sample $\boldsymbol{x}_i$ is given by $\boldsymbol{y}_i = \left[y_{i,1}\ y_{i,2} \cdots y_{i,c}\right]^{\mathrm{T}} \in \{0,1\}^c$, where c is the number of labels. If the label $y_{i,k}$ is relevant to $\boldsymbol{x}_i$, then $y_{i,k} = 1$; otherwise $y_{i,k} = 0$. Let $d_{\boldsymbol{x}}^{y}$ denote the description degree of label y to feature $\boldsymbol{x}$. Then the ground-truth label distribution of $\boldsymbol{x}_i$ is represented by $\boldsymbol{d}_i = \left[d_{\boldsymbol{x}_i}^{y_{i,1}}\ d_{\boldsymbol{x}_i}^{y_{i,2}} \cdots d_{\boldsymbol{x}_i}^{y_{i,c}}\right]^{\mathrm{T}}$. Given a binary label tagged training dataset $\boldsymbol{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i), 1 \le i \le n\}$, LE recovers the label distribution $\widehat{\boldsymbol{d}}_i$ of $\boldsymbol{x}_i$, thereby converting the logically labeled dataset $\boldsymbol{S}$ to the label distribution tagged dataset $\widehat{\boldsymbol{D}} = \left\{\left(\boldsymbol{x}_i, \widehat{\boldsymbol{d}}_i\right), 1 \le i \le n\right\}$, with the probability like condition $\widehat{d}_{\boldsymbol{x}_i}^{y_{i,j}} \in [0,1]$ and $\sum_{j=1}^{c} \widehat{d}_{\boldsymbol{x}_i}^{y_{i,j}} = 1$. The goal of this LE process is to make the estimated label distributions $\{\widehat{\boldsymbol{d}}_i, 1 \le i \le n\}$ as close as possible to the (unknown) ground-true label distributions $\{\boldsymbol{d}_i, 1 \le i \le n\}$.

Fig. 2 depicts our proposed RG4LE, which consists of the feature extraction via RBM, supervised learning via LP, and variational mutual information maximization.

B. Feature Extraction via RBM

The RBM is first pretrained by unsupervised learning to capture data topology from the sample set $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{q\times n}$. In the RG interpretation, $\boldsymbol{X}$ corresponds to the original visible degrees of freedom, while the hidden representation $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] \in \mathbb{R}^{m\times n}$ corresponds to coarse-grained variables. These hidden variables provide a compact representation of the original samples and serve as the feature basis for subsequent LP and mutual information maximization. Typically, $m \le q$. Based on this formulation, the joint probability of the visible and hidden variables can be expressed as a Boltzmann distribution determined by the energy function $E(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{\eta})$, with $\boldsymbol{\eta}$ representing the model's trainable parameters. From the RG perspective, learning the RBM amounts to extracting relevant degrees of freedom from $\boldsymbol{X}$ while suppressing less informative variations. The label relevance of the learned representation is further strengthened by the subsequent LP and mutual information maximization modules.

RBMs belong to a class of stochastic NNs with a two-layer structure, comprising a visible layer of q nodes and a hidden layer of m nodes. The two layers have symmetric connections between them but no within-layer connections or feedback. Collect the symmetric connection weights between the two layers in the matrix $\boldsymbol{W} = \left[w_{j,l}\right] \in \mathbb{R}^{q\times m}$, the bias terms of the q visible nodes in the vector $\boldsymbol{b} = \left[b_1, \cdots, b_q\right]^{\mathrm{T}}$ and the bias terms of the m hidden nodes in the vector $\boldsymbol{c} = \left[c_1, \cdots, c_m\right]^{\mathrm{T}}$.

Then the parameters to learn are $\boldsymbol{\eta} = (\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{c}) \in \Omega_\eta$, where $\Omega_\eta = \mathbb{R}^{q\times m} \times \mathbb{R}^q \times \mathbb{R}^m$. After the RBM is trained, given an input vector $\boldsymbol{x} = \big[x_1, \cdots x_q\big]^{\mathrm{T}}$, the output vector $\boldsymbol{h} = \big[h_1, \cdots h_m\big]^{\mathrm{T}}$ of the RBM, which represents the extracted feature vector for $\boldsymbol{x}$, can be represented as

$$\boldsymbol{h} = \sigma\big(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}\big) = \sigma\big(f(\boldsymbol{x}, \boldsymbol{\eta})\big), \tag{1}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

For unsupervised RBM training using only the sample data, the dependencies between the visible and hidden variables are characterized by the following energy function [35]

$$E(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{\eta}) = -\sum_{j=1}^{q}\sum_{l=1}^{m} x_j w_{j,l} h_l - \sum_{j=1}^{q} b_j x_j - \sum_{l=1}^{m} c_l h_l. \tag{2}$$

It can be seen that the joint probability distribution of $\boldsymbol{X}$ and $\boldsymbol{H}$ is given by the Boltzmann weights. Accordingly, the RBM parameters $\boldsymbol{\eta}$ are estimated during training by maximizing the likelihood function of the observed data samples:

$$\begin{aligned}\max_{\boldsymbol{\eta}\in\Omega_\eta} L(\boldsymbol{\eta}) =& \frac{1}{n}\sum_{i=1}^{n} \log \sum_{k=1}^{n} P\left(\boldsymbol{x}_i, \boldsymbol{h}_k|\boldsymbol{\eta}\right) \\ =& \frac{1}{n}\sum_{i=1}^{n} \log \frac{\sum\limits_{k=1}^{n} \exp\left(-E\left(\boldsymbol{x}_i, \boldsymbol{h}_k|\boldsymbol{\eta}\right)\right)}{\sum\limits_{i=1}^{n}\sum\limits_{k=1}^{n} \exp\left(-E\left(\boldsymbol{x}_i, \boldsymbol{h}_k|\boldsymbol{\eta}\right)\right)}.\end{aligned} \tag{3}$$

In (3), the joint probability of $\boldsymbol{x}$ and $\boldsymbol{h}$, $P(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{\eta})$, characterizes the joint probabilistic dependence between the visible variable $\boldsymbol{x}$ and its hidden representation $\boldsymbol{h}$.

C. Supervised Learning via Label Propagation

In Section III-B, we apply RG to preprocess data for feature extraction, identifying important degrees of freedom to extract the relevant features for assisting the LE model. The feature space obtained after RBM training strengthens class supervision information, namely, the relevance between the topological structures of examples in sample space and labels in label space. By leveraging the feature representations learned by the unsupervised RBM, the LP algorithm is able to capture the manifold structure between samples and propagate logical label information more accurately, thus enhancing the original logical labels with a probabilistic interpretation. More specifically, after obtaining the parameters $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{c}$, the output vector $\boldsymbol{h} = \big[h_1, \cdots h_m\big]^{\mathrm{T}}$ of the RBM, given by (1), represents the extracted feature vector for sample $\boldsymbol{x}$. To introduce label information, an LP algorithm is used to fuse the features extracted by the RBM with the label information contained in the logical labels. The process is as follows.

a) First, the association graph $\mathcal{G} = <\mathcal{V}, \mathcal{E}>$ is constructed using the features extracted by the RBM from the training sample set $\boldsymbol{X}$, where the set of vertices $\mathcal{V} = \{\boldsymbol{h}_i\}_{i=1}^n$ and $\mathcal{E}$ is the corresponding edge set.

b) Second, a sample correlation matrix $\boldsymbol{A} = \big[a_{i,j}\big] \in \mathbb{R}^{n\times n}$ is constructed using the extracted feature vectors $\{\boldsymbol{h}_i\}_{i=1}^n$ as

$$a_{i,j} = \begin{cases} \exp\Big(-\frac{\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2}{2}\Big), & i \neq j, \\ 0, & i = j, \end{cases} \quad 1 \le i, j \le n. \tag{4}$$

Here $a_{i,j}$ is interpreted as the similarity' between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. It can be seen that $\boldsymbol{A}$ is constructed based on the feature space's topological information, namely, the low-dimensional coordinates $\boldsymbol{h}_i$ of the feature space. Based on the smoothness assumption, which generally holds for real-world data, the feature space's topological structure is preserved in the label space. Consequently, the pairwise relationships encoded in the feature space can provide structural guidance for estimating the labels of previously unseen samples.

c) Third, the LP algorithm iteratively propagates the label information through the graph constructed by feature similarities to obtain the predicted labels.

Existing LP algorithms use the sample similarity matrix $\boldsymbol{A}$ to construct the LP matrix as $\boldsymbol{P} = \bar{\boldsymbol{A}}^{-\frac{1}{2}}\boldsymbol{A}\bar{\boldsymbol{A}}^{-\frac{1}{2}}$. Here, $\bar{\boldsymbol{A}} = \mathrm{diag}\big\{\bar{a}_1, \cdots, \bar{a}_n\big\}$ denotes the degree matrix associated with $\boldsymbol{A}$, whose i-th diagonal entry is given by $\bar{a}_i = \sum_{j=1}^n a_{i,j}$, $1 \le i \le n$. This LP matrix only uses the information in $\boldsymbol{A}$, and it does not use the category information provided by logical labels. $\boldsymbol{A}$ merely captures the affinity between samples in the feature space, but fails to model their semantic relationships within the logical label space. For instance, consider two image samples whose logical labels both contain categories such as 'mountain', 'water', 'flower' and 'bird'. This co-occurrence pattern strongly indicates that they belong to the natural landscape domain. Consequently, additional categories present in the first sample are highly likely to manifest in the second sample. This similarity in their logical label space validates the rationality of performing LP between these two samples. This semantic similarity is reflected in the logical label structure and can be used as an effective guide for the LP process.

To incorporate these similarities, we construct a label-state transition matrix $\boldsymbol{T} = \big[T_{i,j}\big] \in \{0, 1\}^{n\times n}$. Specifically, given the label matrix $\boldsymbol{Y} = \big[\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\big] = [Y_{k,i}] \in \{0, 1\}^{c\times n}$, where $Y_{k,i}$ denotes whether the i-th sample is annotated with the k-th label, we define the intersection matrix $\boldsymbol{I} = \big[I_{i,j}\big] \in \mathbb{Z}^{n\times n}$, the total label count vector $\bar{\boldsymbol{c}} = \big[\bar{c}_i\big] \in \mathbb{N}^n$, and the union matrix $\boldsymbol{U} = \big[U_{i,j}\big] \in \mathbb{Z}^{n\times n}$, which are computed according to:

$$\boldsymbol{I} = \boldsymbol{Y}^{\mathrm{T}} \cdot \boldsymbol{Y},\ \bar{c}_i = \sum_{k=1}^{c} Y_{k,i},\ U_{i,j} = \bar{c}_i + \bar{c}_j - I_{i,j}. \tag{5}$$

Based on these matrices, the Jaccard similarity matrix $\boldsymbol{J} = \big[J_{i,j}\big] \in \mathbb{R}^{n\times n}$ for each pair of logical labels is computed as:

$$J_{i,j} = \begin{cases} \frac{I_{i,j}}{U_{i,j}}, & \text{if } U_{i,j} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

After obtaining $\boldsymbol{J}$, the label-state transition matrix $\boldsymbol{T}$ is constructed according to:

$$T_{i,j} = \begin{cases} 1, & \text{if } J_{i,j} > \tau \text{ and } \bar{c}_i < \bar{c}_j, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where τ is the label association threshold (default value 0.5). The diagonal elements of $\boldsymbol{T}$ are set to 0 to prevent a sample from propagating labels to itself, thereby avoiding redundant computations. $T_{i,j} = 1$ indicates that the label is allowed to propagate from sample j to sample i. The hard threshold and asymmetric condition in (7) are used to reduce low-confidence

propagation. Specifically, $J_{i,j} > \tau$ keeps only sufficiently reliable label-state associations, while $\bar{c}_i < \bar{c}_j$ encourages labels to propagate from samples with richer logical label states to samples with fewer labels. This design is consistent with the goal of enhancing incomplete logical labels. It may, however, introduce bias near the threshold boundary or favor samples with more annotated labels as propagation sources. Symmetric or probabilistic transition variants could provide more balanced or smoother propagation, but may also introduce weak label associations and increase propagation noise.

The label-state transition matrix is utilized in PageRank calculations to control both the path and weight of LP. Define the column normalized label-state transition matrix $\boldsymbol{Q}=\big[Q_{i,j}\big]\in\mathbb{R}^{n\times n}$ as:

$$Q_{i,j}=\frac{T_{i,j}}{\sum_{k=1}^{n}T_{k,j}}, \tag{8}$$

and $\boldsymbol{pr}=[pr_i]\in\mathbb{R}^n$ for the importance scores of nodes. Initially, $\boldsymbol{pr}^{(0)}=\big[\frac{1}{n},\cdots,\frac{1}{n}\big]^{\mathrm{T}}\in\mathbb{R}^n$ for equal importance score. Then, $\boldsymbol{pr}$ is iteratively updated according to

$$\boldsymbol{pr}^{(t+1)}=\frac{1-d_f}{n}\mathbf{1}_n+d_f\cdot\boldsymbol{Q}\cdot\boldsymbol{pr}^{(t)}, \tag{9}$$

where $\mathbf{1_n}$ is the all-1-element n-dimensional vector, and $d_f\in(0,\,1)$ is the damping factor controlling the graph defined by the normalized label-state transition matrix $\boldsymbol{Q}$. Higher d_f prioritizes propagation through $\boldsymbol{Q}$'s topology, and $1-d_f$ represents the probability of random jumps that prevent propagation trapping. PageRank ($\boldsymbol{pr}$) is utilized to refine the original sample similarity matrix $\boldsymbol{A}$, so as to ensure more accurate LP. The refined sample similarity matrix $\boldsymbol{S}$ is calculated as

$$\boldsymbol{S}=\boldsymbol{A}\odot\left(\frac{\boldsymbol{pr}\cdot\boldsymbol{pr}^{\mathrm{T}}}{\max\{pr_1,\cdots,pr_n\}}\right). \tag{10}$$

Unlike the LE algorithms in the existing literature, we construct the LP matrix by $\boldsymbol{P}=\bar{\boldsymbol{S}}^{-\frac{1}{2}}\boldsymbol{S}\bar{\boldsymbol{S}}^{-\frac{1}{2}}$, in which $\bar{\boldsymbol{S}}=\mathrm{diag}\big\{\bar{s}_1,\cdots,\bar{s}_n\big\}$ denotes the diagonal matrix with the diagonal elements $\bar{s}_i=\sum_{j=1}^{n}s_{i,j}$, $1\le i\le n$.

Next, we define $\boldsymbol{F}=[f_{i,j}]\in\mathbb{R}^{n\times c}$. Each row of $\boldsymbol{F}$ is related to a sample and each column of $\boldsymbol{F}$ is related to a label. $\boldsymbol{F}$ is known as the label importance matrix. Based on the logical label matrix $\boldsymbol{Y}=\big[\boldsymbol{y}_1,\cdots,\boldsymbol{y}_n\big]\in\{0,1\}^{c\times n}$, the initial label importance matrix is given by $\boldsymbol{F}^{(0)}=\boldsymbol{Y}^{\mathrm{T}}\in\mathbb{R}^{n\times c}$. The RBM feature matrix $\boldsymbol{H}=\big[\boldsymbol{h}_1,\cdots,\boldsymbol{h}_n\big]\in\mathbb{R}^{m\times n}$ contributes to LP through the sample similarity matrix $\boldsymbol{S}$ and the propagation matrix $\boldsymbol{P}$. The label information is then propagated over the sample graph through $\boldsymbol{P}$:

$$\boldsymbol{F}^{(t)}=\alpha\boldsymbol{P}\boldsymbol{F}^{(t-1)}+(1-\alpha)\boldsymbol{Y}^{\mathrm{T}}. \tag{11}$$

The trade-off coefficient $\alpha\in(0,\ 1)$ determines the relative contributions of the propagated information from $\boldsymbol{P}$ and the original supervision provided by $\boldsymbol{Y}$. Since $\boldsymbol{F}^{(t)}=(\alpha\boldsymbol{P})^t\boldsymbol{Y}^{\mathrm{T}}+(1-\alpha)\sum_{i=0}^{t-1}(\alpha\boldsymbol{P})^i\boldsymbol{Y}^{\mathrm{T}}$, it converges to

$$\boldsymbol{F}^{\star}=(1-\alpha)(\underline{\boldsymbol{I}}_n-\alpha\boldsymbol{P})^{-1}\boldsymbol{Y}^{\mathrm{T}}. \tag{12}$$

Here $\underline{\boldsymbol{I}}_n$ denotes the $(n\times n)$ identity matrix. After completing the iterative propagation, we normalize each row of $\boldsymbol{F}^{\star}$ to recover the label distribution according to

$$\widehat{d}_{\boldsymbol{x}_i}^{y_{i,j}}=\frac{f_{i,j}^{\star}}{\sum_{k=1}^{c}f_{i,k}^{\star}},\ 1\le i\le n,\ 1\le j\le c. \tag{13}$$

D. Information Maximization for RG4LE

The previously introduced preprocessing of feature data using RG and the enhancement of logical labels into label distributions via LP represent forward information flow in RG4LE. We now design a bidirectional or reverse flow of information to avoid information loss inherent in a purely unidirectional pipeline. Specifically, we guide the RBM training by maximizing the mutual information $\mathcal{I}\big(\boldsymbol{h};\widehat{\boldsymbol{d}}\big)$, so that the hidden features $\boldsymbol{H}$ extracted by the RBM explicitly focus on features relevant to the label distribution $\widehat{\boldsymbol{D}}$ generated through LP. For notational simplicity, we drop $\widehat{\ }$ in the sequel.

1) Motivation for Mutual Information: The aforementioned forward information flow has two limitations. First, feature-label space misalignment arises because the RBM, trained solely on the sample dataset without supervision information, may extract features statistically independent of the true label distribution. Second, information attenuation occurs as LP relies exclusively on the topological structure of the extracted features. If the extracted features fail to adequately encode label correlations, LP can propagate noise or suboptimal distributions, particularly for ambiguous or rare labels. The mutual information between $\boldsymbol{H}$ and $\boldsymbol{D}$, which quantifies the statistical dependence between the extracted features and the label distributions, is calculated according to

$$\mathcal{I}(\boldsymbol{h};\boldsymbol{d})=\mathcal{H}(\boldsymbol{h})-\mathcal{H}(\boldsymbol{h}|\boldsymbol{d}), \tag{14}$$

where $\mathcal{H}(\cdot)$ denotes the entropy. Maximizing $\mathcal{I}(\boldsymbol{h};\boldsymbol{d})$ forces the feature representations to prioritize label-discriminative features, effectively aligning the feature and label spaces. This mitigates information loss in two ways: 1) it encourages the RBM to extract features relevant for label discrimination during training, and 2) it provides implicit supervision to the LP process through the joint optimization of features and label distributions.

2) Maximizing Variational Mutual Information: Computing the mutual information $\mathcal{I}(\boldsymbol{h};\boldsymbol{d})$ directly requires the posterior distribution $\mathcal{P}(\boldsymbol{h}|\boldsymbol{d})$, which is difficult to sample and estimate. Since direct maximization is challenging, we employ an auxiliary distribution $\mathcal{Q}(\boldsymbol{h}|\boldsymbol{d})$ to approximate $\mathcal{P}(\boldsymbol{h}|\boldsymbol{d})$, resulting in a variational lower bound for mutual information. Following the inference from [36], we establish a lower bound of the mutual information between $\boldsymbol{H}$ and $\boldsymbol{D}$:

$$\begin{aligned}\mathcal{I}(\boldsymbol{h};\boldsymbol{d})&=\mathcal{H}(\boldsymbol{h})-\mathcal{H}(\boldsymbol{h}|\boldsymbol{d})\\&=\mathbb{E}_{\boldsymbol{d}\sim LP(\boldsymbol{d}|\boldsymbol{h})}\big[\mathbb{E}_{\boldsymbol{h}'\sim q_\phi(\boldsymbol{h}|\boldsymbol{d})}[\log\mathcal{P}(\boldsymbol{h}'|\boldsymbol{d})]\big]+\mathcal{H}(\boldsymbol{h})\\&=\mathbb{E}_{\boldsymbol{d}\sim LP(\boldsymbol{d}|\boldsymbol{h})}\big[D_{KL}\big(\mathcal{P}(\cdot|\boldsymbol{d})|\mathcal{Q}(\cdot|\boldsymbol{d})\big)+\\&\quad\ \mathbb{E}_{\boldsymbol{h}'\sim q_\phi(\boldsymbol{h}|\boldsymbol{d})}[\log\mathcal{Q}(\boldsymbol{h}'|\boldsymbol{d})]\big]+\mathcal{H}(\boldsymbol{h})\\&\ge\mathbb{E}_{\boldsymbol{d}\sim LP(\boldsymbol{d}|\boldsymbol{h})}\big[\mathbb{E}_{\boldsymbol{h}'\sim q_\phi(\boldsymbol{h}|\boldsymbol{d})}[\log\mathcal{Q}(\boldsymbol{h}'|\boldsymbol{d})]\big]+\mathcal{H}(\boldsymbol{h}).\end{aligned} \tag{15}$$

In (15), $\mathcal{Q}(\boldsymbol{h}|\boldsymbol{d})$ denotes the auxiliary distribution represented by a NN whose input is the label distribution $\boldsymbol{d}$. The non-negative Kullback-Leibler (KL) divergence ($D_{KL} \geq 0$) quantifies the information loss incurred when using $\mathcal{Q}$ to represent $\mathcal{P}$. As the auxiliary distribution $\mathcal{Q}$ approaches the true posterior distribution $\mathcal{P}$, $\mathbb{E}_{\boldsymbol{d}\sim LP(\boldsymbol{d}|\boldsymbol{h})}\big[D_{KL}\big(\mathcal{P}(\cdot|\boldsymbol{d})|\mathcal{Q}(\cdot|\boldsymbol{d})\big)\big] \to 0$, and can be discarded to derive a tractable lower bound. The entropy term $\mathcal{H}(\boldsymbol{h})$ does not contain optimization variables and is treated as a constant. This approach bypasses the explicit computation of the posterior $\mathcal{P}(\boldsymbol{h}|\boldsymbol{d})$. Nonetheless, we still need to sample from the posterior distribution appearing within the inner expectation. We apply the lemma used in InfoGAN [36] to overcome this sampling difficulty.

Lemma 1. *[36] For random variables X and Y, and a function $f(x, y)$ under suitable regularity conditions,*

$$\mathbb{E}_{x\sim X, y\sim Y|x}[f(x, y)] = \mathbb{E}_{x\sim X, y\sim Y|x, x'\sim X|y}[f(x', y)].$$

According to Lemma 1, we redefine the mutual information's variational lower bound as:

$$\begin{aligned} L_{MI}(\boldsymbol{\eta}, Q) =& \mathbb{E}_{\boldsymbol{d}\sim LP(\boldsymbol{d}|\boldsymbol{h})}\big[\mathbb{E}_{\boldsymbol{h}'\sim q_\phi(\boldsymbol{h}|\boldsymbol{d})}[\log \mathcal{Q}(\boldsymbol{h}'|\boldsymbol{d})]\big] + \mathcal{H}(\boldsymbol{h}) \\ =& \mathbb{E}_{\boldsymbol{d}\sim RBM_{\boldsymbol{\eta}}(\boldsymbol{h}|\boldsymbol{x}), \boldsymbol{d}\sim LP(\boldsymbol{d}|\boldsymbol{h})}[\log \mathcal{Q}(\boldsymbol{h}|\boldsymbol{d})] + \mathcal{H}(\boldsymbol{h}) \\ \leq& \mathcal{I}(\boldsymbol{h}; \boldsymbol{d}), \end{aligned} \tag{16}$$

where $\boldsymbol{\eta}$ includes all the trainable parameters of the RBM, and Q denotes the NN for the auxiliary distribution $\mathcal{Q}(\cdot)$.

3) Overall Objective Function: The overall objective function is constructed by weighting the RBM reconstruction loss with the mutual information's variational lower bound:

$$L_{total}(\boldsymbol{\eta}, Q) = L_{RBM}(\boldsymbol{\eta}) - \lambda L_{MI}(\boldsymbol{\eta}, Q), \tag{17}$$

where L_{RBM} denotes the RBM reconstruction loss, and λ is a trade-off weight controlling the mutual information term. In this objective, the LP branch connects the hidden representation and the mutual information term by generating the current label-distribution estimate $\widehat{\boldsymbol{D}}$ from the RBM feature representation $\boldsymbol{H}$ and the logical labels. The estimate $\widehat{\boldsymbol{D}}$ is then used as the conditional input of the auxiliary distribution $\mathcal{Q}(\boldsymbol{h}|\boldsymbol{d})$ in the variational mutual information lower bound. To avoid direct competition between the Q-NN and the RBM, which can lead to training instability, we adopt a two-step alternating optimization strategy. During each step, the parameters of the branch not being optimized are fixed, and the LP output is treated as a fixed label-distribution estimate rather than a differentiable path for back-propagation. This stop-gradient treatment reduces the risk of unstable mutual adaptation among the RBM, LP, and Q-NN, thereby helping to avoid degenerate solutions.

a) Optimize the Q-NN while fixing the RBM parameters $\boldsymbol{\eta}$. The objective is to maximize the mutual information lower bound: $\max_Q L_{MI}(\boldsymbol{\eta}, Q)$, thereby enhancing the approximation quality of the true posterior by the Q-NN.

b) Optimize the RBM parameters to minimize the joint objective function: $\min_{\boldsymbol{\eta}}(L_{RBM}(\boldsymbol{\eta}) - \lambda L_{MI}(\boldsymbol{\eta}, Q))$, while fixing the Q-NN.

IV. Experiments

A. Setup of Experiments

1) Datasets: We select 13 real-world multilabel datasets tagged with ground-truth label distributions, which are widely adopted in evaluating LE and LDL algorithms [11]. Table II summarizes these datasets. The datasets include the data collected from ten yeast biology experiments (Yeast-alpha to Yeast-spoem), two facial expression image data (SBU_3DFE and SJAFFE), and one natural scene recognition data (Natural_Scene). Since these datasets are only tagged with ground-truth label distributions, logical labels have to be generated from them by cumulative-threshold binarization: for each sample, labels are sorted by their description degrees in descending order and selected as positive labels until their cumulative description degree reaches or exceeds 0.5; the selected labels are assigned 1 and the remaining labels are assigned 0.

2) Performance Measures: Two common classes of metrics to assess LE algorithms are distance-based metrics, which measure the distance or closeness of the estimated label distributions to the ground-truth, and distribution similarity measures, which evaluate the similarity between the estimated label distributions and the ground-truth. In the experiments, four distance-based metrics are adopted, which are Chebyshev distance (Cheb), Clark distance (Clark), Canberra distance (Canber) and KL divergence (KLdiv), while two distribution similarity metrics are adopted, which are cosine coefficient (Cosine) and intersectional similarity (Intersec). For the distance-based metrics, smaller values indicate that the estimated label distributions are closer to the ground truth, and this preference is denoted by '↓'. By contrast, for the distribution similarity metrics, larger values correspond to better performance and are marked by '↑'.

3) Comparison Benchmarks: Our RG4LE is compared with six representative LE models: the LP algorithm [26], the ML algorithm [27], the GLLE [12], the PLEML [29], the LESC [28], and the LE-PNLR [30].

4) Algorithmic Parameters: The key hyperparameters are set as follows. The number of RBM hidden units is equal to the number of labels, i.e., $m = c$. The RBM is pretrained for 500 epochs with two-step contrastive divergence. For LP, the Gaussian kernel width is chosen to be $\sigma = 1.0$, the smoothing parameter is chosen to be $\alpha = 0.9$, the PageRank damping factor is chosen to be $d = 0.85$, and the label-state

TABLE II
13 Datasets used in the experiments. These datasets are tagged with ground-truth label distributions.

Dataset	No. examples (n)	No. features (q)	No. labels (c)
Yeast-alpha	2465	24	18
Yeast-cdc	2465	24	15
Yeast-elu	2465	24	14
Yeast-diau	2465	24	7
Yeast-heat	2465	24	6
Yeast-spo	2465	24	6
Yeast-cold	2465	24	4
Yeast-dtt	2465	24	4
Yeast-spo5	2465	24	3
Yeast-spoem	2465	24	2
SBU_3DFE	2500	243	6
SJAFFE	213	243	6
Natural-Scene	2000	294	9

TABLE III
COMPARISON OF LE PREDICTIVE ACCURACY MEASURED BY SIX METRICS FOR 7 LE ALGORITHMS.

dataset	LP	ML	GLLE	PLEML	LESC	LE-PNLR	RG4LE	LP	ML	GLLE	PLEML	LESC	LE-PNLR	RG4LE
	Chebyshev distance(↓)							Clark distance(↓)						
Yeast-alpha	0.0400(7)	0.0387(6)	0.0192(5)	0.0137(2)	0.0169(4)	0.0151(3)	**0.0129(1)**	0.4322(6)	0.6025(7)	0.3304(5)	0.2147(2)	0.2823(4)	0.2503(3)	**0.1871(1)**
Yeast-cdc	0.0420(6)	0.0475(7)	0.0217(4)	0.0167(2)	0.0198(3)	0.0222(5)	**0.0145(1)**	0.3803(6)	0.5593(7)	0.3018(4)	0.2191(2)	0.2727(3)	0.3249(5)	**0.2053(1)**
Yeast-cold	0.1370(7)	0.1207(6)	0.0650(4)	0.0540(2)	0.0572(3)	0.0820(5)	**0.0405(1)**	0.1805(5)	0.3224(7)	0.1738(4)	0.1465(2)	0.1552(3)	0.2007(6)	**0.1114(1)**
Yeast-diau	0.0990(6)	0.2011(7)	0.0530(5)	0.0415(2)	0.0419(4)	0.0416(3)	**0.0357(1)**	0.2841(5)	0.7276(7)	0.2964(6)	0.2222(2)	0.2302(3)	0.2409(4)	**0.1936(1)**
Yeast-dtt	0.1280(7)	0.1073(6)	0.0518(5)	0.0372(2)	0.0466(4)	0.0387(3)	**0.0281(1)**	0.1902(6)	0.2953(7)	0.1413(5)	0.1012(2)	0.1278(4)	0.1058(3)	**0.0773(1)**
Yeast-elu	0.0440(6)	0.0499(7)	0.0221(5)	0.0165(2)	0.0208(4)	0.0170(3)	**0.0143(1)**	0.3642(6)	0.5340(7)	0.2845(5)	0.2042(2)	0.2617(4)	0.2114(3)	**0.1692(1)**
Yeast-heat	0.0860(6)	0.0915(7)	0.0478(5)	0.0433(2)	0.0466(4)	0.0436(3)	**0.0360(1)**	0.2144(6)	0.3823(7)	0.2082(5)	0.1871(3)	0.2037(4)	0.1869(2)	**0.1492(1)**
Yeast-spo	0.0900(6)	0.0953(7)	0.0608(4)	0.0603(3)	0.0609(5)	0.0586(2)	**0.0484(1)**	0.5585(7)	0.4030(6)	0.2618(5)	0.2558(3)	0.2596(4)	0.2510(2)	**0.2068(1)**
Yeast-spo5	0.1140(5)	0.1514(6)	0.0980(4)	0.0921(2)	0.0933(3)	0.6608(7)	**0.0620(1)**	0.2741(5)	0.3015(6)	0.1943(4)	0.1855(2)	0.1871(3)	0.6839(7)	**0.1306(1)**
Yeast-spoem	0.1630(6)	0.1319(5)	0.0870(2.5)	0.1170(4)	0.0870(2.5)	0.7099(7)	**0.0449(1)**	0.2718(6)	0.2036(5)	0.1321(3)	0.1757(4)	0.1295(2)	0.6375(7)	**0.0707(1)**
SBU_3DFE	0.1230(3.5)	0.1868(7)	0.1230(3.5)	0.1228(2)	0.1231(5)	0.1388(6)	**0.1141(1)**	0.5810(6)	0.7861(7)	0.3818(4)	0.3689(2)	0.3785(3)	0.4077(5)	**0.3377(1)**
SJAFFE	0.1070(5)	0.2188(7)	0.0845(2)	0.0885(3)	**0.0692(1)**	0.1193(6)	0.0989(4)	0.3140(2)	0.8055(7)	0.3633(4)	0.3775(5)	**0.2763(1)**	0.4211(6)	0.3588(3)
Natural-Scene	**0.2750(1)**	0.2990(2)	0.3353(4)	0.3384(5)	0.3417(6)	0.3574(7)	0.3177(3)	2.4828(7)	2.4520(2)	2.4609(3)	2.4659(5)	2.4649(4)	2.4758(6)	**2.4459(1)**
Average Rank	5.5000(6)	6.1538(7)	4.0769(4)	2.5385(2)	3.7308(3)	4.6154(5)	**1.3846(1)**	5.6154(6)	6.3077(7)	4.3846(4)	2.7692(2)	3.2308(3)	4.5385(5)	**1.1538(1)**
	Canberra(↓)							Kullback-Leibler divergence(↓)						
Yeast-alpha	1.7068(6)	2.0181(7)	1.1135(5)	0.6981(2)	0.9514(4)	0.8252(3)	**0.6413(1)**	0.1210(7)	0.0550(6)	0.0130(5)	0.0057(2)	0.0080(4)	0.0074(3)	**0.0051(1)**
Yeast-cdc	1.3532(6)	1.7591(7)	0.9442(4)	0.6545(2)	0.8405(3)	1.0382(5)	**0.5546(1)**	0.1110(7)	0.0609(6)	0.0140(4)	0.0073(2)	0.0100(3)	0.0144(5)	**0.0058(1)**
Yeast-cold	0.3241(5)	0.5598(7)	0.3016(4)	0.2527(2)	0.2680(3)	0.3489(6)	**0.1909(1)**	0.1030(6)	0.5560(7)	0.0190(4)	0.0135(2)	0.0150(3)	0.0238(5)	**0.0077(1)**
Yeast-diau	0.6425(5)	1.6538(7)	0.6734(6)	0.4772(2)	0.5021(3)	0.5362(4)	**0.4133(1)**	0.1270(6)	0.1934(7)	0.0270(5)	0.0158(2)	0.0170(3)	0.0176(4)	**0.0119(1)**
Yeast-dtt	0.3560(6)	0.5070(7)	0.2458(5)	0.1747(2)	0.2229(4)	0.1830(3)	**0.1319(1)**	0.1030(7)	0.0648(6)	0.0130(5)	0.0066(2)	0.0100(4)	0.0071(3)	**0.0039(1)**
Yeast-elu	1.2612(6)	1.6263(7)	0.8692(5)	0.6014(2)	0.7906(4)	0.6299(3)	**0.4903(1)**	0.1090(7)	0.0567(6)	0.0130(5)	0.0064(2)	0.0090(4)	0.0069(3)	**0.0050(1)**
Yeast-heat	0.4706(6)	0.7826(7)	0.4203(5)	0.3741(3)	0.4110(4)	0.3717(2)	**0.2810(1)**	0.0890(7)	0.0656(6)	0.0170(5)	0.0134(2.5)	0.0155(4)	0.0134(2.5)	**0.0092(1)**
Yeast-spo	1.2341(7)	0.8440(6)	0.5422(5)	0.5281(3)	0.5329(4)	0.5188(2)	**0.4187(1)**	0.0840(6)	0.5320(7)	0.0290(5)	0.0271(3)	0.0280(4)	0.0252(2)	**0.0168(1)**
Yeast-spo5	0.4013(5)	0.4664(6)	0.3018(4)	0.2849(2)	0.2884(3)	0.6337(7)	**0.1974(1)**	0.0420(5)	0.0811(6)	0.0340(4)	0.0299(2)	0.0310(3)	0.6768(7)	**0.0140(1)**
Yeast-spoem	0.3655(6)	0.2800(5)	0.1840(4)	0.1837(3)	0.1801(2)	0.6129(7)	**0.0967(1)**	0.0670(5)	0.5030(6)	0.0270(2.5)	0.0459(4)	0.0270(2.5)	0.8803(7)	**0.0074(1)**
SBU_3DFE	1.2463(6)	1.6593(7)	0.8409(4)	0.7866(2)	0.8039(3)	0.8817(5)	**0.7462(1)**	0.1050(6)	0.2489(7)	0.0690(3)	0.0659(2)	0.0692(4)	0.0900(5)	**0.0540(1)**
SJAFFE	1.0708(6)	1.6894(7)	0.7518(2)	0.7876(4)	**0.5606(1)**	0.8789(5)	0.7529(3)	0.0770(6)	0.2513(7)	0.0500(4)	0.0494(3)	**0.0290(1)**	0.0727(5)	0.0486(2)
Natural-Scene	6.7810(2)	**6.7217(1)**	6.8511(4)	6.8717(5)	6.8780(6)	6.8859(7)	6.7852(3)	1.5950(5)	2.2757(6)	2.6630(7)	0.9787(2)	1.1663(4)	1.0153(3)	**0.8617(1)**
Average Rank	5.5385(6)	6.2308(7)	4.3846(4)	2.6154(2)	3.3846(3)	4.5385(5)	**1.3077(1)**	6.1538(6)	6.3846(7)	4.5000(5)	2.3462(2)	3.3462(3)	4.1923(4)	**1.0769(1)**
	Cosine coefficient(↑)							Intersec(↑)						
Yeast-alpha	0.9814(6)	0.9530(7)	0.9876(5)	0.9944(2)	0.9905(4)	0.9927(3)	**0.9950(1)**	0.9074(6)	0.8898(7)	0.9386(5)	0.9615(2)	0.9473(4)	0.9547(3)	**0.9635(1)**
Yeast-cdc	0.9828(6)	0.9468(7)	0.9875(4)	0.9930(2)	0.9896(3)	0.9861(5)	**0.9947(1)**	0.9122(6)	0.8836(7)	0.9376(4)	0.9569(2)	0.9445(3)	0.9317(5)	**0.9636(1)**
Yeast-cold	0.9847(4)	0.9429(7)	0.9827(5)	0.9873(2)	0.9859(3)	0.9757(6)	**0.9929(1)**	0.9213(5)	0.8646(7)	0.9250(4)	0.9376(2)	0.9338(3)	0.9100(6)	**0.9533(1)**
Yeast-diau	0.9805(5)	0.8427(7)	0.9750(6)	0.9854(2)	0.9844(3)	0.9836(4)	**0.9892(1)**	0.9128(5)	0.7557(7)	0.9052(6)	0.9335(2)	0.9301(3)	0.9255(4)	**0.9428(1)**
Yeast-dtt	0.9835(6)	0.9515(7)	0.9884(5)	0.9937(2)	0.9901(4)	0.9932(3)	**0.9965(1)**	0.9134(6)	0.8779(7)	0.9393(5)	0.9570(2)	0.9448(4)	0.9548(3)	**0.9677(1)**
Yeast-elu	0.9829(6)	0.9489(7)	0.9879(5)	0.9906(3)	0.9896(4)	0.9932(2)	**0.9952(1)**	0.9120(6)	0.8839(7)	0.9383(5)	0.9575(2)	0.9439(4)	0.9554(3)	**0.9654(1)**
Yeast-heat	0.9861(4)	0.9454(7)	0.9845(6)	0.9872(2)	0.9851(5)	0.9871(3)	**0.9915(1)**	0.9237(6)	0.8718(7)	0.9310(5)	0.9385(3)	0.9324(4)	0.9389(2)	**0.9531(1)**
Yeast-spo	0.9386(6)	0.8397(7)	0.9747(4)	0.9747(4)	0.9747(4)	0.9763(2)	**0.9845(1)**	0.8184(6)	0.7614(7)	0.9105(5)	0.9130(3)	0.9121(4)	**0.9763(1)**	0.9316(2)
Yeast-spo5	0.9686(5)	0.9359(6)	0.9713(4)	0.9736(2)	0.9732(3)	0.6657(7)	**0.9884(1)**	0.8855(5)	0.7486(6)	0.9020(4)	0.9079(2)	0.9067(3)	0.6642(7)	**0.9380(1)**
Yeast-spoem	0.9503(5)	0.8530(6)	0.9782(2)	0.9620(4)	0.9780(3)	0.6121(7)	**0.9945(1)**	0.8367(5)	0.7681(6)	0.9130(2.5)	0.9109(4)	0.9130(2.5)	0.6906(7)	**0.9551(1)**
SBU_3DFE	0.9220(5)	0.8435(7)	0.9304(4)	0.9344(2)	0.9319(3)	0.9140(6)	**0.9464(1)**	0.8096(6)	0.7414(7)	0.8531(4)	0.8570(2)	0.8542(3)	0.8409(5)	**0.8670(1)**
SJAFFE	0.9410(5)	0.8231(7)	0.9594(2)	0.9576(3)	**0.9731(1)**	0.9315(6)	0.9539(4)	0.8361(6)	0.7251(7)	0.8757(2)	0.8718(4)	**0.9050(1)**	0.8501(5)	0.8730(3)
Natural-Scene	0.7264(5)	0.6610(6)	**0.7789(1)**	0.7738(2)	0.7602(3)	0.6376(7)	0.7344(4)	0.4512(4)	0.3307(7)	**0.5226(1)**	0.5214(2)	0.5107(3)	0.4220(6)	0.4449(5)
Average Rank	5.2308(6)	6.7692(7)	4.0769(4)	2.4615(2)	3.3077(3)	4.6923(5)	**1.4615(1)**	5.5385(6)	6.8462(7)	4.0385(4)	2.4615(2)	3.1923(3)	4.3846(5)	**1.5385(1)**

association threshold is set to $\tau = 0.5$. For mutual information maximization, the trade-off weight is set to $\lambda = 0.1$, with 5 alternating optimization rounds. All the experiments are carried out with Python 3.8 and Torch 1.13.1 on a PC with an Intel(R) Core(TM) i5-12400 2.50 GHz processor.

B. Experimental Results with Discussion

For each dataset, cumulative-threshold binarization is utilized to generate logical labels from the ground-truth label distributions. Each LE method is then applied to the complete logically labeled dataset to recover the label distributions of all samples. The estimated label distributions are compared with the ground truth using the 6 evaluation metrics. To reduce the influence of stochastic variation, each experiment is repeated ten times to report the average performance.

1) LE Predictive Accuracy Results: The label predictive performance of the 7 LE methods are listed in Table III. The results are reported in the form of mean (rank), where the value in parentheses denotes the rank and a smaller value indicates better result. The best performance in each comparison is highlighted in bold. The LE prediction accuracy results presented in Table III convincingly demonstrate that our RG4LE consistently outperforms the other six LE benchmarks.

2) Ablation study: In order to assess the contributions made by the components of RG4LE, an ablation study was conducted. Specifically, we evaluate the impact of the label-state transition matrix and mutual information maximization mechanisms on LE performance. The ablation study results are presented in Table IV, which compares the complete RG4LE framework (top row) with two ablated versions: RG4LE without the label-state transition matrix (middle row) and RG4LE without mutual information maximization (bottom row).

The results indicate that omitting the label-state transition matrix leads to a significant decline in model performance. This deterioration can be attributed to the absence of mechanisms to capture the relationships and transition patterns between labels, which play a critical role in enhancing predictive accuracy. Similarly, removing the mutual information maximization component prevents the RBM from extracting informative features from the raw data. Without this mechanism, the relationship between the label distribution and the extracted feature vectors is not adequately captured. The only exception is the Intersection value on Yeast-alpha, where the version without mutual information maximization is marginally higher than RG4LE; nevertheless, RG4LE remains superior on the other five metrics for this dataset and shows the best overall performance across datasets. Overall the ablation study highlights the pivotal role of the label-state transition matrix and mutual information maximization components in the RG4LE framework. The three modules depicted in Fig. 2

interact through a bidirectional information flow. Specifically, the RBM first extracts the hidden representation $\boldsymbol{H}$ from the original samples; LP then uses $\boldsymbol{H}$ together with the label-state transition matrix to estimate the label distribution $\widehat{\boldsymbol{D}}$; finally, mutual information maximization uses the estimated label-distribution information as a supervisory signal to guide RBM training, making the learned features more informative for subsequent LP. This interaction explains why removing either the transition matrix or the mutual information term weakens the overall enhancement performance.

3) Friedman Test and Critical Difference Diagram: A standard nonparametric procedure for comparing multiple algorithms across multiple datasets is Friedman test [37]. We employ this test to assess whether the performance differences among the 7 LE methods reported in Table III are statistically significant. In this study, 7 algorithms are evaluated on 13 datasets. Table V summarizes the resulting Friedman statistics F_F. Since the F_F values for all the 6 metrics exceed the critical value, the null hypothesis that all the algorithms perform equivalently is rejected. Therefore, Nemenyi post-hoc test [37] is further applied to compare the average ranks of all algorithms. Figs. 3 to 8 present the resulting critical difference (CD) diagrams for the 6 evaluation metrics.

Given the significance level of 0.05 and 7 compared algorithms over 13 datasets, the CD value is 1.8870 for the Nemenyi test [37]. In the CD diagram for an evaluation metric over the 13 datasets, if the average order difference between two algorithms is smaller than the CD value, it is deemed that the two algorithms exhibit no significant difference, and the two algorithms are connected by a line segment in the CD diagram. Algorithms that are not directly linked to our RG4LE by a line segment in the CD diagram are deemed to exhibit significant performance difference from RG4LE. From Figs. 3 to 8, it can be seen that for Chebyshev, Clark, Canberra and KLdist metrics, only PLEML has direct line segments connected with RG4LE, while for cosine and Intersection metrics, only PLEML and LESC have direct line segments connected with RG4LE. The CD Diagram results therefore suggest that most of the LE performance results given in Table III are statistically significant, i.e., statistically, RG4LE consistently achieves better LE predictive accuracy.

4) Statistical Validation of LE Predictive Performance: In order to reveal the statistical relationships among the 7 compared LE models over 13 datasets, we adopt Wilcoxon signed-rank test [37]. Specifically, we use this test to validate whether our RG4LE performs significantly better than the other 6 LE models. Table VI presents the Wilcoxon signed-rank test results, with the p-values for the corresponding tests given in the brackets. As can be concluded by the results of Table VI, it is statistically significant that our RG4LE outperforms all the other six LE models in all the six metrics, with the exception of LESC, which is tie with RG4LE in cosine and Intersection metrics.

We further employ the Bayesian signed-rank test [38] to quantify the extent to which RG4LE performs better than the other 6 benchmark methods. Compared with the conventional Wilcoxon signed-rank test, the Bayesian analysis provides a more informative probabilistic characterization of the pairwise performance differences. Table VII reports the corresponding results. For each comparison, the triplet [a,b,c] denotes the posterior probabilities that RG4LE performs better than, is practically equivalent to, or performs worse than the competing method, respectively, namely, [WIN,TIE,LOSE]. The analysis adopts an equivalent-performance assumption as the prior

TABLE IV
ABLATION STUDY FOR ASSESSING CONTRIBUTIONS MADE BY COMPONENTS OF RG4LE. FOR EACH DATASET, TOP ROW IS RG4LE, MIDDLE ROW IS RG4LE WITHOUT LABEL-STATE TRANSITION MATRIX, AND BOTTOM ROW IS RG4LE WITHOUT MI MAXIMIZATION.

dataset	Cheb ↓	Clark ↓	Canber ↓	KLdiv ↓	Cosine ↑	Intersec ↑
	0.0129	**0.2034**	**0.6413**	**0.0051**	**0.9950**	0.9635
Yeast-alpha	0.0152	0.2454	0.7703	0.0068	0.9935	0.9578
	0.0131	0.2042	0.6612	0.0054	0.9949	**0.9649**
	0.0145	**0.1871**	**0.5546**	**0.0058**	**0.9947**	**0.9636**
Yeast-cdc	0.0159	0.2144	0.6575	0.0064	0.9939	0.9567
	0.0155	0.2053	0.6125	0.0064	0.9939	0.9597
	0.0405	**0.1114**	**0.1909**	**0.0077**	**0.9929**	**0.9533**
Yeast-cold	0.0518	0.1346	0.2325	0.0110	0.9893	0.9415
	0.0518	0.1348	0.2331	0.0110	0.9893	0.9415
	0.0357	**0.1936**	**0.4133**	**0.0119**	**0.9892**	**0.9428**
Yeast-diau	0.0468	0.2606	0.5906	0.0200	0.9812	0.9175
	0.0440	0.2534	0.5806	0.0201	0.9813	0.9191
	0.0281	**0.0773**	**0.1319**	**0.0039**	**0.9965**	**0.9677**
Yeast-dtt	0.0322	0.0871	0.1484	0.0048	0.9955	0.9635
	0.0319	0.0867	0.1477	0.0048	0.9955	0.9637
	0.0143	**0.1692**	**0.4903**	**0.0050**	**0.9952**	**0.9654**
Yeast-elu	0.0162	0.1998	0.5935	0.0061	0.9940	0.9578
	0.0154	0.1901	0.5581	0.0056	0.9947	0.9606
	0.0360	**0.1492**	**0.2870**	**0.0092**	**0.9915**	**0.9531**
Yeast-heat	0.0368	0.1575	0.3127	0.0093	0.9912	0.9489
	0.0361	0.1527	0.2956	0.0093	0.9912	0.9516
	0.0484	**0.2068**	**0.4187**	**0.0168**	**0.9845**	**0.9316**
Yeast-spo	0.0511	0.2167	0.4413	0.0191	0.9823	0.9276
	0.0502	0.2153	0.4426	0.0187	0.9827	0.9278
	0.0620	**0.1306**	**0.1974**	**0.0140**	**0.9884**	**0.9380**
Yeast-spo5	0.0854	0.1694	0.2631	0.0253	0.9775	0.9146
	0.0774	0.1560	0.2411	0.0228	0.9803	0.9226
	0.0449	**0.0707**	**0.0967**	**0.0074**	**0.9945**	**0.9551**
Yeast-spoem	0.0686	0.1034	0.1433	0.0182	0.9850	0.9314
	0.0687	0.1038	0.1437	0.0186	0.9848	0.9313
	0.1141	**0.3377**	**0.7462**	**0.0540**	**0.9464**	**0.8670**
SBU_3DFE	0.1304	0.3724	0.8026	0.0777	0.9246	0.8542
	0.1299	0.3702	0.7987	0.0764	0.9258	0.8552
	0.0989	**0.3588**	**0.7529**	**0.0486**	**0.9539**	**0.8730**
SJAFFE	0.1099	0.3816	0.7856	0.0607	0.9424	0.8660
	0.1105	0.3830	0.7803	0.0605	0.9426	0.8669
	0.3177	**2.4459**	**6.7852**	**0.8617**	**0.7344**	**0.4449**
Natural-Scene	0.3520	2.4718	6.9234	1.0712	0.6269	0.3910
	0.3499	2.4710	6.9251	1.0692	0.6262	0.3897

TABLE V
FRIEDMAN STATISTICS F_F FOR SIX METRICS AND CRITICAL VALUE AT SIGNIFICANCE LEVEL OF 0.05. THERE ARE 7 COMPARED ALGORITHMS AND 13 DATASETS.

Evaluation metric	F_F	Critical value
Cheb	16.7712	2.227
Clark	23.6695	
Canber	19.4795	
KLdiv	47.1500	
Cosine	25.1594	
Intersec	28.4638	

TABLE VI
COMPARING LE PERFORMANCE OF OURRG4LE VERSUS SIX EXISTING LE ALGORITHMS USING WILCOXON SIGNED-RANK TEST, GIVEN SIGNIFICANCE LEVEL $\alpha = 0.05$. THE p-VALUES ARE GIVEN IN THE BRACKETS.

RG4LE versus	evaluation metric					
	Cheb	Clark	Canber	KLdiv	Cosine	Intersec
LP	WIN[0.004638671875]	WIN[0.000732421875]	WIN[0.00048828125]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.00048828125]
ML	WIN[0.00048828125]	WIN[0.000244140625]	WIN[0.00048828125]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
GLLE	WIN[0.004638671875]	WIN[0.000244140625]	WIN[0.00048828125]	WIN[0.000244140625]	WIN[0.026611328125]	WIN[0.026611328125]
PLEML	WIN[0.005615234375]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.0478515625]	WIN[0.021484375]
LESC	WIN[0.013427734375]	WIN[0.013427734375]	WIN[0.010009765625]	WIN[0.013671875]	TIE[0.16259765625]	TIE[0.14208984375]
LE-PNLR	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.0126953125]

TABLE VII
COMPARING LE PERFORMANCE OF RG4LE VERSUS SIX EXISTING LE ALGORITHMS USING BAYESIAN SIGNED-RANK TEST, GIVEN SIGNIFICANCE LEVEL ROPE $= 0.0001$ AND DEFAULT PRIOR STRENGTH OF 0.6. THE THREE VALUES IN THE BRACKET ARE PROBABILITIES OF [WIN,TIE,LOSE].

RG4LE vs	performance metric					
	Cheb	Clark	Canber	KLdiv	Cosine	Intersec
LP	[1.0, 0.0, 0.0]	[0.99948, 0.0, 0.00052]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
ML	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
GLLE	[1.0, 0.0, 0.0]	[0.99926, 0.0, 0.00074]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[0.98454, 0.0, 0.01546]	[0.98586, 0.0, 0.01414]
PLEML	[1.0, 0.0, 0.0]	[0.9979, 0.0, 0.0021]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[0.97652, 2e-05, 0.02346]	[0.98768, 0.0, 0.01232]
LESC	[0.99688, 0.0, 0.00312]	[0.99438, 0.0, 0.00562]	[0.98714, 0.0, 0.01286]	[0.99344, 0.0, 0.00656]	[0.9212, 0.0, 0.0788]	[0.93048, 0.0, 0.06952]
LE-PNLR	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[0.99416, 0.0, 0.00584]

reference, with the prior strength set to 0.6. Two results are regarded as practically equivalent when their absolute difference falls within the region of practical equivalence, defined by rope $= 0.0001$. As shown in Table VII, RG4LE achieves dominant posterior probabilities of winning against the other 6 benchmarks across all six metrics.

5) Visualization of label distribution: We use the Yeast-spo5 dataset, which contains 3 labels, to provide a visual illustration of the label distributions estimated by the 7 LE models, in comparison with the truth label distribution. The description degrees associated with the 3 labels are mapped to the red, green, and blue channels of the RGB color space, respectively. Under this representation, the color assigned to each sample directly encodes its predicted label distribution. Consequently, the recovery quality of an LE method can be visually assessed by comparing the color structure formed by its predictions with that induced by the ground-truth over the sample manifold.

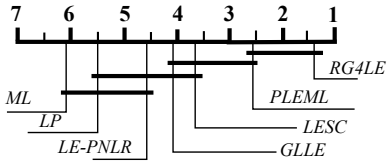
Fig. 9 depicts the truth label distribution and the 7 recovered label distributions shown as 3D RGB color patterns of the RGB color channels, with the three axes corresponding to the three components of the example. How accurate a predicted label distribution is reflected by how close the color pattern of the recovered label distribution to that of the truth label distribution. Our RG4LE achieves the closest visual alignment with the ground-truth label distribution by accurately replicating the cyan-green dominance in core regions (RGB $\approx$ [0.2,0.6,0.2]) and fully preserving the characteristic blue-to-yellow gradient transitions toward the periphery, including precise localization of salient red points. In contrast, LESC and GLLE exhibit excessive blue intensity (B-channel amplification) in central zones while ML manifests prevalent gray patches in these regions, all three methods showing fragmented transitional patterns. LE-PNLR suffers from global desaturation, while PLEML and LP show significant deviations from ground-truth with severely distorted color distributions. These collective discrepancies unequivocally confirm RG4LE's superior performance.

C. Computational Complexity Analysis

The computational complexity per iteration comparison for the 7 LE algorithms are presented in Table VIII. To verify this theoretical analysis, we record the runtimes imposed by the 7 LE models on the LE learning of the 13 datasets, and the results are listed in Table IX, which show that our RG4LE only imposes an average runtime of 0.3331 s over 13 datasets, representing a speedup of more than 60 times over ML, the second-fastest method. Experimental results convincingly verify the theoretical analysis given in Table VIII.

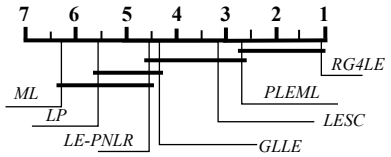
TABLE IX
COMPARISON OF RUNTIME [S] $\downarrow$ FOR 7 LE MODELS.

Algorithms	LP	ML	GLLE	PLEML	LESC	LE-PNLR	RG4LE
Yeast-alpha	96.7089(4)	10.4143(2)	2101.2288(6)	29.5964(3)	2468.2178(7)	1086.2167(5)	**0.2358(1)**
Yeast-cdc	91.9976(4)	28.0560(3)	2058.3899(6)	25.4105(2)	2620.3463(7)	658.9864(5)	**0.2227(1)**
Yeast-cold	85.7640(5)	25.9546(3)	2098.2072(6)	19.2865(2)	2592.1138(7)	39.7982(4)	**0.1777(1)**
Yeast-diau	90.4889(4)	22.2115(3)	2164.7457(6)	21.7103(2)	2587.7626(7)	113.4968(5)	**0.191(1)**
Yeast-dtt	85.0445(5)	24.5773(3)	2062.2618(6)	19.6530(2)	2568.7960(7)	38.9715(4)	**0.1774(1)**
Yeast-elu	91.1003(4)	31.2115(3)	1949.0746(6)	25.1244(2)	2244.6699(7)	602.7289(5)	**0.2183(1)**
Yeast-heat	88.0539(5)	33.0071(3)	2170.0425(6)	22.5063(2)	2324.9983(7)	83.4711(4)	**0.1839(1)**
Yeast-spo	88.1222(5)	21.0123(2)	2113.8297(6)	22.1987(3)	2491.1317(7)	83.4708(4)	**0.1844(1)**
Yeast-spo5	87.4979(5)	23.8147(2)	2234.4134(6)	32.4963(4)	2566.9491(7)	23.9176(3)	**0.1737(1)**
Yeast-spoem	100.8651(5)	7.5451(3)	2053.6103(6)	27.1619(4)	2654.3258(7)	16.2123(2)	**0.1683(1)**
Natural-Scene	71.9972(4)	15.5970(2)	1098.1414(6)	98.3183(5)	2654.3258(7)	69.75(3)	**1.1171(1)**
SJAFFE	0.6692(2)	2.2597(3)	49.4767(7)	17.1449(5)	40.8202(6)	5.35(4)	**0.1201(1)**
SBU_3DFE	109.8430(5)	13.2224(2)	2068.8012(6)	36.1094(3)	2726.1515(7)	78.46(4)	**1.1604(1)**
Average	83.7041(5)	19.9141(2)	1863.2479(6)	30.5167(3)	2349.2776(7)	223.1408(4)	**0.3331(1)**



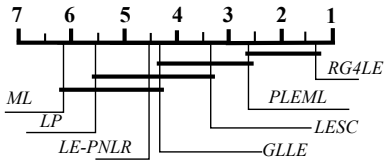
(a) Chebyshev distance

Fig. 3. CD diagrams of Nemenyi tests with Chebyshev distance metric over 13 datasets for 7 LE algorithms.



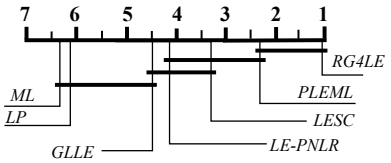
(b) Clark distance

Fig. 4. CD diagrams of Nemenyi tests with Clark distance metric over 13 datasets for 7 LE algorithms.



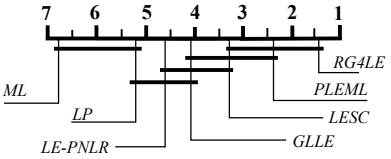
(c) Canberra distance

Fig. 5. CD diagrams of Nemenyi tests with Canberra distance metric over 13 datasets for 7 LE algorithms.



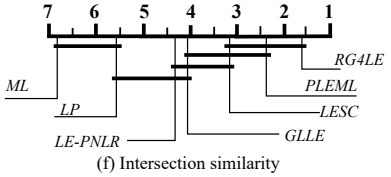
(d) Kullback-Leibler distance

Fig. 6. CD diagrams of Nemenyi tests with Kullback-Leibler divergence metric over 13 datasets for 7 LE algorithms.



(e) Cosine coefficient

Fig. 7. CD diagrams of Nemenyi tests with cosine coefficient metric over 13 datasets for 7 LE algorithms.



(f) Intersection similarity

Fig. 8. CD diagrams of Nemenyi tests with intersectional similarity metric over 13 datasets for 7 LE algorithms.

The LE predictive performance and the runtime results therefore confirm that our RG4LE is markedly more computationally efficient than the competing state-of-the-art LE algorithms, while achieving better LE accuracy.

V. Conclusions

This paper has proposed a new generative label enhancement model, called RG4LE, which pioneers three innovations. 1) RBM hybrid training combining unsupervised pretraining with mutual information maximization-based supervision. 2) Semantic-aware propagation via the label-state transition matrix. 3) Bidirectional alignment through mutual information maximization. This unified approach significantly advances LE performance while reducing computational burden. Extensive experimental results have convincingly demonstrated that our novel RG4LE offers superior LE performance over best existing LE algorithms, while imposing significantly lower computational complexity.

Future research will further examine the practical value of the label distributions recovered by RG4LE beyond standard LE evaluation protocols. One meaningful direction is to use the label distributions recovered by RG4LE from the original binary labels as richer soft supervision signals for downstream predictors, such as multi-label prediction models in image, text, or other application-oriented scenarios, and compare them with predictors trained directly from the original logical labels under the same experimental settings. Such an evaluation would help determine whether the recovered label distributions can improve supervised learning performance in application-oriented scenarios. In addition, recent developments in LE, LDL, and data-centric learning will be considered to further extend the applicability of RG4LE.

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science,* vol. 349, no. 6245, pp. 255–260, 2015.

[2] F. Y. Wang, X. Wang, L. X. Li, and L. Li, "Steps toward parallel intelligence," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 4, pp. 345–348, Oct. 2016.

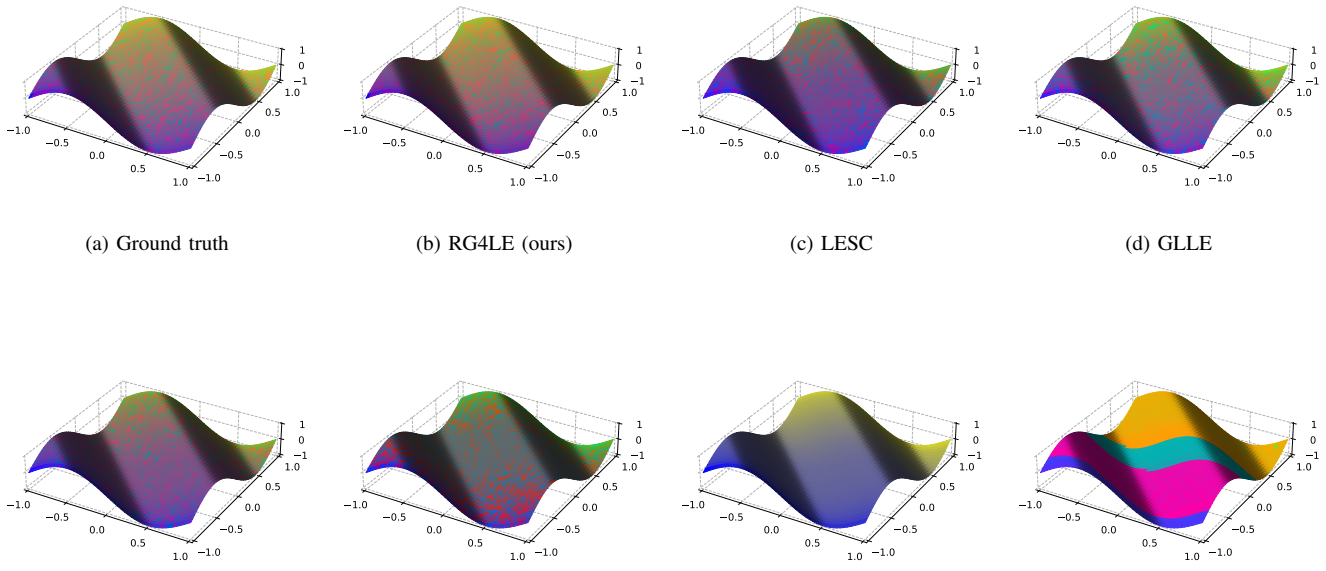
[3] Z. Li, *et al.*, "TEG-DB: A comprehensive dataset and benchmark of textual-edge graphs," in *Proc. NeurIPS 2024* (Vancouver, BC Canada), Dec. 10-15, 2024, pp. 60980–60998, 2024.

[4] M. Haider, *et al.*, "Networkgym: Reinforcement learning environments for multi-access traffic management in network simulation," in *Proc. NeurIPS 2024* (Vancouver, BC Canada), Dec. 10-15, 2024, pp. 106628–106648.

TABLE VIII
COMPUTATIONAL COMPLEXITY PER ITERATION COMPARISON FOR VARIOUS LE ALGORITHMS, WHERE I_{irwls} DENOTES THE NUMBER OF ITERATIONS FOR THE ITERATIVELY REWEIGHED LEAST SQUARES [39], I_{bfgs} DENOTES THE NUMBER OF ITERATIONS FOR THE BFGS ALGORITHM [40], AND $\mathsf{O}(\mathrm{SVM})$ DENOTES THE COMPUTATIONAL COMPLEXITY OF THE SUPPORT VECTOR MACHINE (SVM).

LE algorithm	Complexity per iteration
LP [26]	$\mathsf{O}(q\,n^2)$
ML [27]	$\mathsf{O}(n^2 + I_{\mathrm{irwls}}\,n^3)$
GLLE [12]	$\mathsf{O}((q + I_{\mathrm{bfgs}})n^2)$
PLEML [29]	$\mathsf{O}(n^2) + \mathsf{O}(\mathrm{SVM})$
LESC [28]	$\mathsf{O}((q + I_{\mathrm{bfgs}})n^2)$
LE-PNLR [30]	$\mathsf{O}(q\,c\,n^2)$
Our RG4LE	$\mathsf{O}(n\,q\,c)$

[5] S. Li, *et al.*, "CausalStock: Deep end-to-end causal discovery for news-driven multi-stock movement prediction," in *Proc. NeurIPS 2024* (Vancouver, BC Canada), Dec. 10-15, 2024, pp. 47432–47454.
[6] K. Zhang, *et al.*, "Artificial intelligence in drug development," *Nature Medicine*, vol. 31, pp. 45–59, 2025.
[7] Y. Lin, Z. Yu, K. Yang, Z. Fan, and C. L. P. Chen, "Boosting adaptive weighted broad learning system for multi-label learning," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 11, pp. 2204–2219, Nov. 2024.
[8] T. Zhao, *et al.*, "Onion irrigation treatment inference using a low-cost hyperspectral scanner," in *Proc. SPIE Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques and Applications VII* (Honolulu, HI, USA), Sep. 24-26, 2018, pp. 1–7.
[9] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science,* vol. 349, no. 6245, pp. 261–266, 2015.
[10] Y. Wang, *et al.*, "Lithium-ion battery face imaging with contactless walabot and machine learning," in *Proc. ICMA 2019* (Tianjin, China), Aug. 4-7, 2019, pp. 1067–1072.
[11] X. Geng, "Label distribution learning," *IEEE Trans. Knowledge and Data Eng.*, vol. 28, no. 7, pp. 1734–1748, 2016.
[12] N. Xu, Y. P. Liu, and X. Geng, "Label enhancement for label distribution learning," *IEEE Trans. Knowledge and Data Eng.*, vol. 33, no. 4, pp. 1632–1643, 2021.
[13] F. Sung, *et al.*, "Learning to compare: Relation network for few-shot learning," in *Proc. CVPR 2018* (Salt Lake City, UT, USA), Jun. 18-22, 2018, pp. 1199–1208.
[14] X. H. Wen and M. C. Zhou, "Evolution and role of optimizers in training deep learning models," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 10, pp. 2039–2042, Oct. 2024.
[15] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. ICLR 2017* (Toulon, France), Apr. 24-26, 2017, pp. 1–15.
[16] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 588–598, Jul. 2017.
[17] V. K. Verma, *et al.*, "Generalized zero-shot learning via synthesized examples," in *Proc. CVPR 2018* (Salt Lake City, UT, USA), Jun. 8-22, 2018, pp. 4281–4289.
[18] X. J. Zhu, *Semi-supervised Learning with Graphs*. PhD Thesis, CMU-LTI-05-192, Carnegie Mellon University, May 2005.
[19] D. D. Sante, *et al.*, "Deep learning the functional renormalization group," *Physical Review Lett.*, vol. 129, article ID 136402, pp. 1–7, 2022.
[20] P. Mehta and D. J. Schwab, "An exact mapping between the variational renormalization group and deep learning," *arXiv:1410.3831*, 2014. https://arxiv.org/abs/1410.3831
[21] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?," *J. Statistical Physics*, vol. 168, pp. 1223–1247, 2017.
[22] C. Tan, S. Chen, J. Zhang, Z. Xu, X. Geng, and G. Ji, "RG4LDL: Renormalization group for label distribution learning," *Knowl.-Based Syst.*, vol. 320, Art. no. 113666, 2025, doi: 10.1016/j.knosys.2025.113666.
[23] G. Montúfar, "Restricted Boltzmann machines: Introduction and review," in *Proc. IGAIA IV* (Liblice, Czech Republic), Jun. 12-17, 2016, pp. 75–115.
[24] E. N. Gayar, F. Schwenker, and G. Palm, "A study of the robustness of KNN classifiers trained using soft labels," in *Proc. ANNPR 2006* (Ulm, Germany), Aug. 31-Sep. 2, 2006, pp. 67–80.
[25] X. F. Jiang, Z. Yi, and J. C. Lv, "Fuzzy SVM with a new fuzzy membership function," *Neural Computing and*



(a) Ground truth (b) RG4LE (ours) (c) LESC (d) GLLE

(e) LE-PNLR (f) ML (g) PLEML (h) LP

Fig. 9. Ground-truth label distribution and estimated label distributions by the 7 LE models shown as RGB color patterns on Yeast-spo5 test manifold.

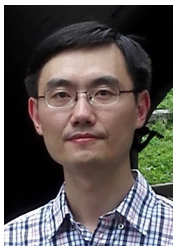
Applications, vol. 15, no. 3, pp. 268–276, 2006.
[26] M. L. Zhang, *et al.*, "Leveraging implicit relative labeling-importance information for effective multi-label learning," *IEEE Trans. Knowledge and Data Eng.*, vol. 33, no. 5, pp. 2057–2070, 2021.
[27] P. Hou, X. Geng, and M. L. Zhang, "Multi-label manifold learning," in *Proc. AAAI 2016* (Phoenix, AZ, USA), Feb. 12-17, 2016, pp. 1680–1686.
[28] H. Y. Tang, *et al.*, "Label enhancement with sample correlations via low-rank representation," in *Proc. AAAI 2020* (New York, NY, USA), Feb. 7-12, 2020. pp. 5932–5939.
[29] W. F. Zhu, X. Y. Jia, and W. W. Li, "Privileged label enhancement with multi-label learning," in *Proc. IJCAI 2020* (Yokohama, Japan), Jan. 7-15, 2020, pp. 2376–2382.
[30] X. Y. Jia, Y. N. Lu, and F. W. Zhang, "Label enhancement by maintaining positive and negative label relation," *IEEE Trans. Knowledge and Data Eng.*, vol. 35, no. 2, pp. 1708–1720, 2023.
[31] Q. Z. Ye, J. Zhang, H. R. Wu, T. L. Gu, C. L. P. Chen, and J. Y. Long, "SMLE: Semi-supervised multi-label learning with label enhancement," *IEEE Trans. Knowledge and Data Eng.*, 2025.
[32] X. Y. Zhao, Y. X. An, N. Xu, L. Qi, and X. Geng, "Interactive fusion label enhancement for multi-label learning," *ACM Trans. Knowledge Discovery from Data*, vol. 19, no. 7, pp. 1–23, 2025.
[33] A. Petermann, "La normalisation des constantes dans la théorie des quanta," *Helvetica Physica Acta*, vol. 26, pp. 499–520, 1952.
[34] M. Koch-Janusz and Z. Ringel, "Mutual information, neural networks and the renormalization group," *Nature Physics*, vol. 14, no. 6, pp. 578–582, 2018.
[35] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," *Neural Networks*, vol. 7700, pp. 599–619, 2012.
[36] X. Chen, *et al.*, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. NIPS 2016* (Barcelona, Spain), Dec. 5–10, 2016, pp. 2180–2188.
[37] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.
[38] A. Benavoli, *et al.*, "Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis," *J. Machine Learning Research*, vol. 18, no. 77, pp. 1–36, 2017.
[39] F. Párez-Cruz, *et al.*, "An IRWLS procedure for SVR," In *Proc. 10th European Signal Processing Conf.* (Tampere, Finland), Sep. 4-8, 2000, pp. 1–4.
[40] J. Nocedal and S.-J. Wright, *Numerical Optimization*. Springer: New York, 2006.

Chao Tan (Member, IEEE) received the BE and ME degrees in computer science and technology from Southeast University, in 2005 and 2009, respectively, and the PhD degree in computer science and technology from Tongji University, in 2015. She joined the Nanjing Normal University as a lecturer in 2015 and is a professor in the School of Computer and Electronic Information/School of Artificial Intelligence at present. She worked as a postdoctoral researcher in Southeast University from 2016 to 2020. From 2019 to 2020, she was a visiting researcher at School of Electronics and Computer Science, University of Southampton, U.K. She has published papers in IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Cybernetics(TCYB), Pattern Recognition, AAAI, IJCAI, etc. She has served as a reviewer or meta-reviewer for IEEE TKDE, AAAI, IJCAI, PR and related journals and conferences. Her research interests generally focus on machine learning, multi-label manifold learning and data mining.

Sheng Chen (M'90-SM'97-F'08) received the B.Eng. degree in control engineering from the Huadong Petroleum Institute, Dongying, China, in 1982, and the Ph.D. degree in control engineering from the City, University of London, London, U,K., in 1986, and the higher doctoral degree, D.Sc., from the University of Southampton, Southampton, U.K. in 2005. From 1986 to 1999, he held research and academic appointments with the University of Sheffield, Sheffield, U.K., University of Edinburgh, Edinburgh, U.K., and University of Portsmouth, Portsmouth, U.K. Since 1999, he has been with the Department of Electronics and Computer Science, University of Southampton, Southampton, U.K., where he holds the post of Professor in intelligent systems and signal processing. He is currently a Distinguished Adjunct Professor with King Abdulaziz University, Jeddah, Saudi Arabia. He has authored over 600 research papers. His current research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods and optimization. Dr. Chen is a Fellow of the Royal Academy of Engineering, London, U.K., a Fellow of the IET, and an ISI highly cited researcher in engineering (2004).

Mengjiao Kai received the bachelor's degree in software engineering from the Qingdao University, QingDao, China, in 2024. She is currently working toward the postgraduation degree with the Department of Computer and Electronic Information, Nanjing Normal University, Nanjing. Her research interests generally focus on machine learning, multi-label manifold learning and data mining.

Shangce Gao (Senior Member, IEEE) received his Ph.D. degree in Innovative Life Science from the University of Toyama, Toyama, Japan, in 2011. He is currently a Professor with the Faculty of Engineering at the University of Toyama. His research interests focus on brain-inspired neural networks and their applications in medical diagnosis and drug discovery. He serves as an Associate Editor for several international journals, including IEEE Transactions on Neural Networks and Learning Systems.



Xin Geng (Senior Member, IEEE) received the BSc and MSc degrees in Computer Science from Nanjing University, China, in 2001 and 2004, respectively, and the PhD degree from Deakin University, Australia in 2008. He is currently a Chair Professor with Southeast University, China. His research interests include machine learning, pattern recognition, and computer vision. He has published over 100 refereed articles in these areas. He has been an Associate Editor of IEEE Transactions on Multimedia, FCS, and MFC, a Steering Committee Member of PRICAI, the Program Committee Chair of conferences, such as PRICAI 2018 and VALSE 2013, the Area Chair of conferences, such as IJCAI, CVPR, ACMMM, and ICPR, and a Senior Program Committee Member of conferences, such as IJCAI, AAAI, and ECAI. He is a Distinguished Fellow of IETI.