Big Data Mining and Analytics

Research Article

Label distribution learning by mining deep semantic label-specific-features

Jing Wang[1,2,✉], Jinyao Liu[3], Lingqi Liu[1]

Cite this article: Wang J, Liu JY, Liu LQ. *Big Data Min. Anal.* https://doi.org/10.26599/BDMA.2025.9020114

ABSTRACT: Label Distribution Learning (LDL) is a novel machine learning paradigm that effectively addresses label ambiguity in big data analytics. Despite its success, LDL faces two persistent challenges: feature redundancy in high-dimensional spaces and computational inefficiency with large-scale label sets. Although existing methods alleviate these issues by exploiting label correlations, they generally work in the raw feature space, ignoring the critical role of discriminative Label-Specific-Features (LSF). To address this limitation, we propose a novel LDL approach that considers LSF through mining deep semantics, termed LAMPS. First, the label-semantic encoding module jointly learns label-specific and distribution-level semantics via deep graph autoencoders. Second, the learned deep semantics are fused into an LDL process that considers LSF and distribution-level information. We conduct extensive experiments on 12 real-world datasets. The experimental results demonstrate that LAMPS achieves statistically superior improvements over state-of-the-art LDL baselines in 89.4% of the cases and attains 3.2× speedup on the largest dataset, demonstrating its advantages for processing large-scale LDL data.

KEYWORDS: Label Distribution Learning (LDL); Label-Specific Features (LSF); feature selection; graph autoencoder; data mining

1 Introduction

Label Distribution Learning (LDL) [1] has emerged as an effective framework for addressing the challenge of label ambiguity in big data analytics [2–4]. Unlike Single-Label Learning (SLL) and Multi-Label Learning (MLL) paradigms—which identify which label(s) characterize an instance—LDL associates each instance with a label distribution. Concretely, a label distribution is a multivariate vector, whose elements are called the label description degrees and quantify the relative importance of all labels [1]. For example, Fig. 1 shows a multi-label image, in which "Land", "Water", "Cloud", and "Sky" are the ground-truth labels but have varying importance, e.g., "Land" $\succ$ "Water". MLL assigns all ones to them and fails to consider their different importance. In comparison, in LDL, the image is assigned with label distribution, with degrees of 0.268, 0.191, 0.135, and 0.406 for "Sky", "Cloud", "Water", and "Land", respectively, as shown in Fig. 1, which effectively represents their different importance. Overall, the shift from deterministic labels to continuous distributions overcomes representational limitations inherent in SLL and MLL, enabling richer modeling of semantic nuances in complex real-world data and mitigating label ambiguity.

Since its proposal, LDL has already found wide applications in big data analytics [5, 6], such as Facial Expression Recognition (FER) [7], age estimation [8, 9], and crowd-source learning [10]. For example, a facial image often blends complex combinations of basic emotions (e.g., happiness, sadness, surprise, fear, anger, and disgust), and the varying intensities of these emotions collectively form an emotion distribution [7]. Compared with single-label and multi-label, label distribution provides a nuanced representation for expression analysis. Then, FER can be regarded as an emotion distribution learning problem [7, 11]. Moreover, due to the smooth change of facial appearance, researchers have converted the ground-truth age to a label distribution and cast age estimation as an LDL problem [8, 9].

Recently, researchers have developed various LDL approaches across many dimensions, e.g. model architecture development to enhance learning capabilities and label correlation exploitation to resolve label dimensionality issues. For instance, Shen et al. [9] developed differentiable forest models for label distribution modeling, while Jin et al. [12] proposed Graph LDL (GLDL) leveraging Graph Convolutional Networks (GCNs) to explicitly model instance relationships and facilitate label correlation organization. In addition, Jia et al. [13] first considered label correlation for LDL by weighting the model parameters with Pearson's correlation coefficients among labels. Ren et al. [14] considered both global and local label correlations by exploiting the low-rank structure of label space. Wang and Geng [15] learned the label manifold to model both global and local label correlations.

While existing LDL algorithms have already demonstrated success across various applications, most implementations rely exclusively on generic feature representations shared among all class labels. However, such feature utilization may lead to suboptimal performance due to the interference between competing class labels. As illustrated in Fig. 1, critical feature

[1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China. [2] The First Affiliated Hospital of Jinzhou Medical University, Jinzhou 121002, China. [3] Chien-Shiung Wu College, Southeast University, Nanjing 210096, China.

✉ Corresponding author. E-mail: wangjing91@seu.edu.cn

Received: July 9, 2025; Revised: October 4, 2025; Accepted: November 3, 2025

© The Author(s) 2026. This is an open access article under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0, http://creativecommons.org/licenses/by/4.0/).

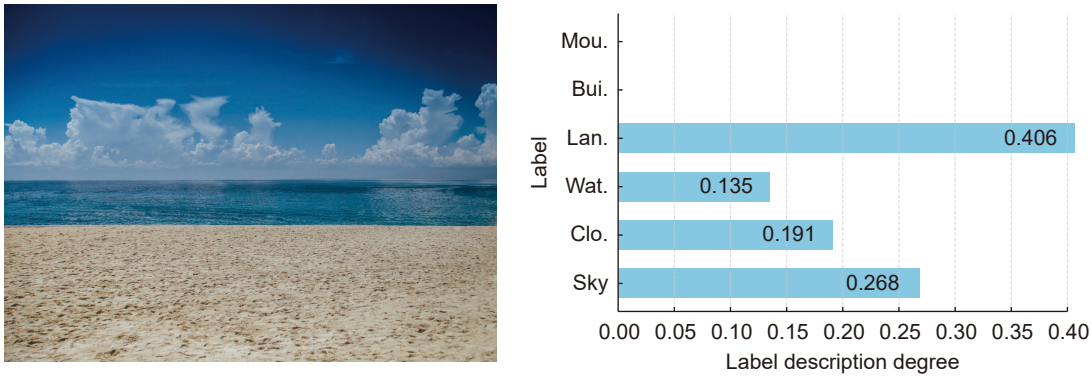


Fig. 1 Example of label distribution: (a) the image and (b) its label distribution with degrees of 0.268, 0.191, 0.135, and 0.406 for "Sky", "Clo.", "Wat.", and "Lan.", respectively, and zeros for the remaining labels. Here, "Clo.", "Wat.", "Lan.", "Bui", "Mou." represent "Cloud", "Water", "Land", "Building", and "Mountain", respectively.

differentiation requirements are involved, for example, the chrominance features dominate in distinguishing "Sky" from "Land", yet prove ineffective for differentiating "Sky" from "Water". Meanwhile, texture-based features may introduce detrimental effects when learning "Sky" given the typically homogeneous texture of clear skies. However, learning on the original raw features totally ignores such discriminative features, which would lead to sub-optimal performance.

To solve the above challenge, researchers have proposed feature selection [16] and Label-Specific Features (LSF) [17]. Feature selection selects subsets of features from original ones, while LSF learns discriminative features for each class label [17]. As illustrated in Fig. 2, different labels are highly correlated with different subsets of features; for example, "Sky" is highly correlated to the zeroth, third, fourth, and fifth dimensional features, which are called the LSF for "Sky". LSF has emerged as a promising solution for MLL, where each class label is associated with its most discriminative feature subset. It enables targeted noise mitigation and interference reduction, thereby enhancing both model accuracy and computational efficiency. LSF has witnessed its success in the field of MLL [17, 18].

Recently, some researchers have also considered LSF in the field of LDL. To name just a few, Ren et al. [14] selected common and label-specific features from the raw features. Lin et al. [19] considered both $L_{2,1}$-regularization and label correlation to mine label-specific and common features. Zhang et al. [20] generated LSF for each label by considering label-value rankings. While

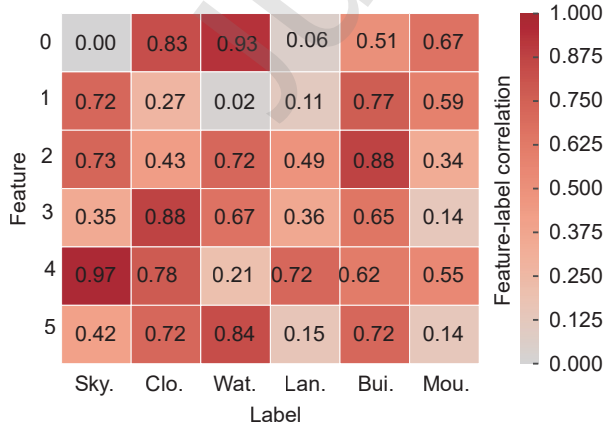


Fig. 2 Illustration of label-specific features. Suppose the image has six-dimensional features with six-dimensional labels, "Sky", "Clo.", "Wat.", "Lan.", "Bui.", and "Mou". The correlation between different labels and different features is diverse. Some features are more discriminative for some labels while others are less.

these methods have improved the performance of LDL, the LSF learning processes can not be combined with a deep learning framework, which limits the learning and representation ability of these works.

In this paper, we propose label distribution learning with deep semantic label-specific features (termed LAMPS). LAMPS has two novel components: (1) the label-semantic encoding module that learns label-specific and distribution-level semantics, and (2) the LDL module guided by the learned semantics. In particular, the label-semantic encoding module operates through two stages. First, multi-granularity instance topology construction captures both label-specific and distribution-level structural patterns. Second, these topological representations are learned in an unsupervised manner by a graph autoencoder. The label-semantic encoding module jointly yields label-specific and distribution-level semantics. Next, the LDL module learns label distribution guided by the two semantic encodings (the original features are projected into the label-specific and distribution-level semantic space). The learned LSFs are discriminative and of lower dimension compared with the original features, which strengthens the performance of LDL. Finally, we conduct extensive experiments on 12 real-world datasets. The experimental results and analysis demonstrate the superior performance, better efficiency, and robustness of LAMPS.

Overall, our contributions can be summarized as follows.

(1) We propose an end-to-end framework that simultaneously captures label-specific features and cross-label distribution semantics through a joint learning process.

(2) We develop a label-specific feature encoder using graph auto-encoding, enabling effective nonlinear mapping of original feature spaces.

(3) We conduct extensive experiments on 12 real-world datasets, demonstrating that LAMPS significantly outperforms state-of-the-art baselines in prediction accuracy while remaining computationally efficient, particularly on tasks with high-dimensional labels.

We organize the rest of the paper as follows. First, we briefly review the related work in Section 2. Second, we present the LAMPS approach in Section 3. Third, we report the experimental results in Section 4. Finally, we conclude this paper in Section 5.

2 Related work

This paper is related to two lines of work, i.e., label distribution learning and LSF learning, which are discussed below.

2.1 Label distribution learning

LDL was first proposed by Geng [1] to address label ambiguity by

Just Accepted

associating each instance with a label distribution, instead of logical labels. Concretely, label distribution is a multivariate vector, whose elements are called label description degrees and indicate the relative importance of all labels. As a result, it enables a more fine-grained label-semantic representation of each label [1]. Geng also proposed three implementation strategies for LDL, including Problem Transformation (PT), Algorithm Adaptation (AA), and Specialized Algorithm (SA) [1]. In detail, PT converts LDL problems into existing frameworks, AA extends conventional algorithms to learn label distribution, and SA considers the characteristics of LDL. Representative LDL algorithms include PT-Bayes, AA-kNN, and SA-BFGS, among others.

Since its proposal, LDL has gained popularity across various domains for handling semantic ambiguity in complex annotation scenarios. For example, LDL could effectively model continuous variations in facial expression recognition [7, 21] and age estimation [9]. Li and Deng [7] employed label distribution to represent the blended emotions of facial images in-the-wild. Shen et al. [9] cast facial age estimation as an LDL problem and designed a differential forest to learn such label distribution. LDL has also been applied to emotion analysis in NLP, where sentences often express multiple emotions with varying intensities [22, 23]. Zhang et al. [23] developed a multi-task convolutional neural network to learn text emotion distribution. Moreover, Geng and Hou [24] formulated pre-release movie sentiment forecasting as an LDL problem.

Recent advancements in LDL demonstrate some new directions. For example, Wang and Geng [15] developed a manifold learning method to capture global and local label correlations in a data-driven framework. Wen et al. [25] modeled sequential labels through spatio-temporal and semantic ordering for structured LDL tasks and proposed the ordinal LDL problem. Qian et al. [26] designed a neighborhood-based feature selection method, which further enhances the performance by measuring feature-label correlations. These innovations reflect the continuously evolving methodology for complex real-world applications. Wang et al. [27] introduced fuzzy label correlation into LDL, and each instance blends multiple local label correlations.

Although the above works have enhanced the performance of LDL, these methods are derived from the original features. On one hand, these methods neglect feature redundancy in the original feature space. Meanwhile, they ignore that one label may only correlate with a subset of the original features, i.e., its own LSF. As a result, these methods may have sub-optimal performance for complex applications. In comparison, our work learns the most discriminative features for each label.

2.2 Label-specific feature learning

LSF learning is a commonly used strategy in MLL, which exploits the feature space to provide relevant and discriminative features for each class label, because different class labels usually carry specific features of their own. LSF was first proposed by Zhang and Wu [17] to deal with MLL. They designed a representative algorithm called LIFT, which clusters positive and negative instances of each label, respectively, and then constructs label-specific features by querying the clustering results. LSF generally has lower dimensions than the original features, which help reduce redundant features. Subsequent works like LLSF [28] leverage second-order label correlation to learn label-specific features for multi-label classification. Besides LLSF, MLFC [29] and LF-LPLC [30] also combine label-specific features and label correlation for MLL. MLFC establishes an optimization framework integrating label correlations via auxiliary features, while LF-LPLC adapts a similar approach to LIFT to learn the label-specific features and exploits the local correlation between each pair of labels by means of nearest neighbor techniques. Moreover, numerous studies [31–36] have demonstrated that the construction of LSF is an efficient strategy for MLL.

Recently, LSFs have also been adopted in LDL. For example, LDLSF [14] jointly optimizes shared and label-specific features through unified representation learning. González et al. [37] integrated prototype selection with feature engineering, and Shu et al. [38] designed a dimensionality reduction framework to learn LSF. Qian et al. [39] employed feature weight fusion with local label correlations, and Zhang et al. [20] extended the LIFT paradigm to LDL through feature subspace fusion. Li et al. [40] developed a two-stage architecture that filters irrelevant features via label-independent prediction, and Lin et al. [41] designed a hierarchical feature selection method. Moreover, Wang et al. [5] first conducted fuzzy C-means clustering and then used the clusters to derive LSF. The above methods derive LDL models with the learned LSF, which are more discriminative and with lower dimensions.

3 LAMPS approach

In this section, we present the details of the LAMPS approach. First, we introduce some preliminaries. Second, we explain the method formulation. Third, we elaborate on the optimization process.

3.1 Preliminaries

Let $\mathcal{X} \in \mathbb{R}^q$ be the input space of q-dimensional and $\mathcal{Y} = \{y_1, y_2, \ldots, y_p\}$ be the complete set of labels with p classes. Let $\boldsymbol{x}$ be the instance variable. In the settings of LDL, each instance $\boldsymbol{x}$ is associated with label distribution $\boldsymbol{d} = [d_{\boldsymbol{x}}^{y_1}, d_{\boldsymbol{x}}^{y_2}, \ldots, d_{\boldsymbol{x}}^{y_p}]^{\mathrm{T}}$, where $d_{\boldsymbol{x}}^{y_j}$ is called the label description degree that satisfies $d_{\boldsymbol{x}}^{y_j} \in [0, 1]$ and $\sum_{j=1}^{p} d_{\boldsymbol{x}}^{y_j} = 1$ [1].

Let $S = \{(\boldsymbol{x}_1, \boldsymbol{d}_1), (\boldsymbol{x}_2, \boldsymbol{d}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{d}_n)\}$ be a training set with n examples, where $\boldsymbol{x}_i$ is the i-th training instance, and $\boldsymbol{d}_i = [d_{\boldsymbol{x}_i}^{y_1}, d_{\boldsymbol{x}_i}^{y_2}, \ldots, d_{\boldsymbol{x}_i}^{y_p}]$ is the assigned label distribution. Given a loss function $\ell : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}$, LDL can be formulated into the optimization problem in Formula (1):

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \ell(\boldsymbol{d}_i, f(\boldsymbol{x}_i; \boldsymbol{W})) \tag{1}$$

where $f(\cdot)$ is a parametric model with parameter $\boldsymbol{W}$, such as neural network [2] and the maximum entropy model [1, 5].

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ be the feature matrix and $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n]^{\mathrm{T}}$ the label distribution matrix. We define by the identity matrix $\boldsymbol{I}$ and the set $[n]$ i.e., $\{1, 2, \ldots, n\}$. Table 1 summarizes the main notations used.

3.2 Method formulation

Figure 3 illustrates the overall framework of the LAMPS approach that consists of two modules:

(1) **Label-semantic encoding module**: LAMPS employs the GCN-based graph autoencoder to encode the embeddings of label-specific semantic and distribution-semantic.

(2) **Label distribution learning module**: Guided by the learned label-semantic encoding, LAMPS learns LSF and the weight matrix of the model, enabling accurate label distribution prediction.

For convenience, we assume that the feature space and the label space are linearly correlated. Then, the goal of LDL is to find a

Just Accepted

Table 1 Definition of the used notations.

Symbol	Definition
$\mathcal{X}$ and $\mathcal{Y}$	Feature space and label space
$\boldsymbol{x}$ and $\boldsymbol{d}$	Instance variable and label distribution
$\boldsymbol{x}_i$ and $\boldsymbol{d}_i$	i-th training instance and label distribution
$\boldsymbol{D}$	Label distribution matrix
$\hat{\boldsymbol{D}}$	Predicted label distribution matrix
$\boldsymbol{X}$	Feature matrix
$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{Z}_i$	Label semantic for the i-th label
$\boldsymbol{E}$	Distribution-level semantic
$\boldsymbol{I}$	Identity matrix
$\|\cdot\|_{\mathrm{F}}$	Frobenius norm
$[n]$	Set of $\{1, 2, \ldots, n\}$

parameter $\boldsymbol{W} \in \mathbb{R}^{q\times p}$ that can predict the label distribution similar to $\boldsymbol{d}_i$ given the instance $\boldsymbol{x}_i$. The model's output can be represented by Eq. (2):

$$\hat{\boldsymbol{D}} = \boldsymbol{X}\boldsymbol{W} \tag{2}$$

where $\hat{\boldsymbol{D}}$ is the predicted label distribution matrix. We utilize the Frobenius norm of the weight parameter to control the complexity of the objective function and mitigate the risk of overfitting. We could use any loss function for LDL, such as Kullback–Leibler (KL)-divergence and L_2-norm loss, to learn label distribution. Here, we use the L_2-norm loss as the loss function, and formulate LDL as Formula (3):

$$\min_{\boldsymbol{W}} \|\boldsymbol{X}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{W}\|_{\mathrm{F}}^2 \tag{3}$$

where λ is the regularization parameter. Formula (3) fits an LDL model with the original features.

3.2.1 *Label-semantic encoding*

Next, we present the details of the label-semantic encoding module. LSF assumes that some labels may be determined by some discriminative features. Extracting features from the original feature space may lead to imperfect consideration and large computational complexity. To solve these problems, we introduce a graph autoencoder to generate label-specific embeddings automatically from data.

Graph Autoencoder (GAE) [42, 43] is a framework for unsupervised learning on graph-structured data based on autoencoder. In detail, we employ a two-layer GCN encoder and an inner product decoder. The structure of GCN is defined in Eq. (4):

$$\mathrm{GCN}(\boldsymbol{X}, \boldsymbol{A}) = \tilde{\boldsymbol{A}} \cdot \mathrm{ReLU}(\tilde{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}_1) \cdot \boldsymbol{W}_2 \tag{4}$$

where $\boldsymbol{A}$ is the adjacency matrix, $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ and

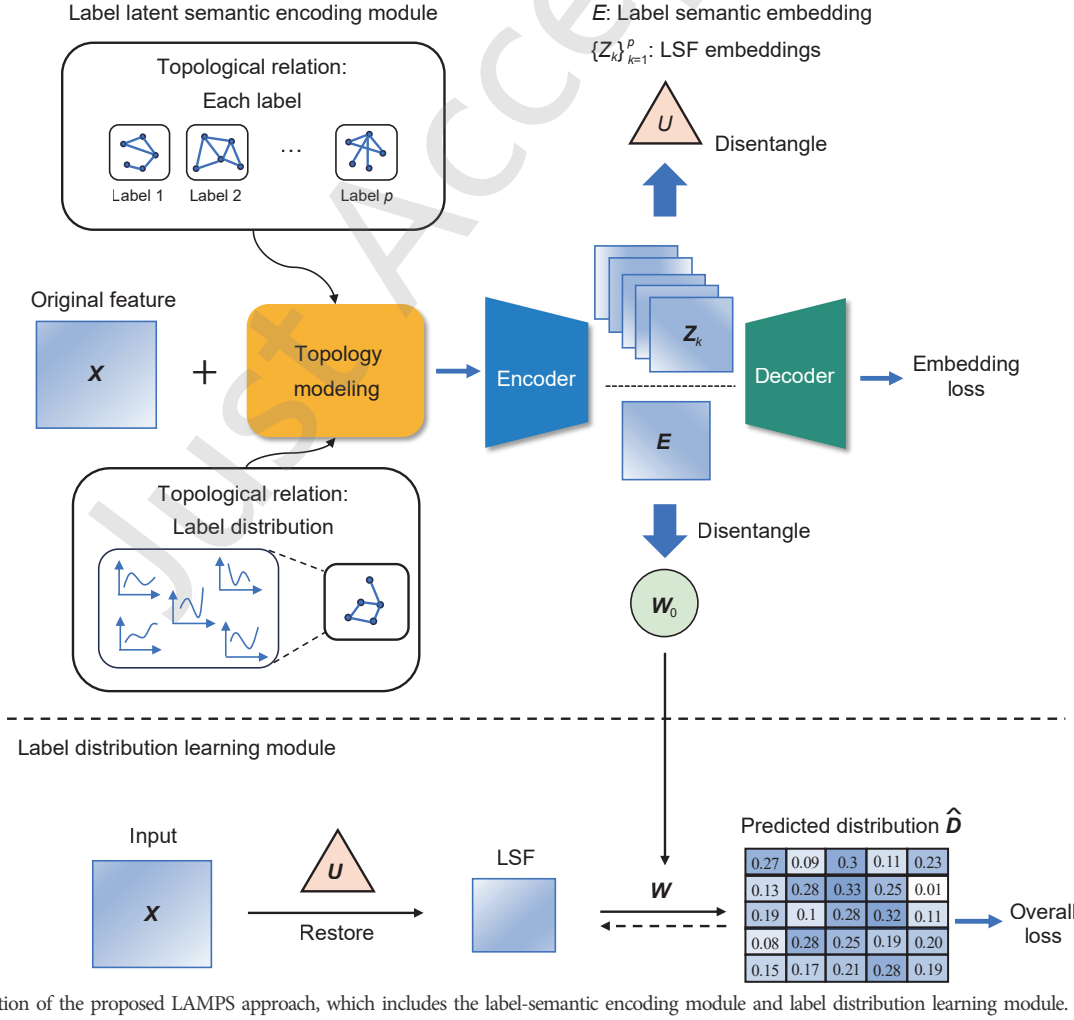


Fig. 3 Illustration of the proposed LAMPS approach, which includes the label-semantic encoding module and label distribution learning module. First, the label-semantic encoding module learns label-specific and distribution-level embeddings with a graph autoencoder. Second, guided by the learned embedding, the label distribution learning module learns LSF and optimizes the model parameters.

Just Accepted

$\boldsymbol{W}_1 \in \mathbb{R}^{q\times h_1}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{h_1\times h_2}$ denote the learnable parameters. $\tilde{\boldsymbol{A}} = \boldsymbol{D}_{\boldsymbol{A}}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}_{\boldsymbol{A}}^{-\frac{1}{2}}$ is the symmetrically normalized adjacency matrix, where $\boldsymbol{D}_{\boldsymbol{A}}$ means the degree matrix of $\boldsymbol{A}$. We denote the latent variables by $\boldsymbol{z}_i$ with $\boldsymbol{Z} \in \mathbb{R}^{n\times h_2}$ as the latent semantic matrix, i.e.,

$$\boldsymbol{Z} = \mathrm{GCN}(\boldsymbol{X},\ \boldsymbol{A}) \tag{5}$$

The generative model of the decoder is given by an inner product between latent variables. The embeddings $\boldsymbol{Z}$ and the reconstructed adjacency matrix $\hat{\boldsymbol{A}}$ are calculated as Eq. (6):

$$\hat{\boldsymbol{A}} = \sigma(\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}) \tag{6}$$

where σ is the Sigmoid function.

First, we define all instances as the node set $\mathcal{V} = \{v_i\}_{i=1}^n$, where each node v_i denotes instance $\boldsymbol{x}_i$ with label distribution d_i. We construct two graphs through two edge generation methods:

- **Label-specific edges**. For each label y_k, the edge between nodes v_i and v_j is determined by the Euclidean distance between their label description degrees $d_{\boldsymbol{x}_i}^{y_k}$ and $d_{\boldsymbol{x}_j}^{y_k}$. In detail, v_i is connected to v_j if $d_{\boldsymbol{x}_i}^{y_k}$ is among the k nearest neighbors of $d_{\boldsymbol{x}_j}^{y_k}$. That generates an adjacency matrix $\boldsymbol{A}_k$. We generate p adjacency matrices $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_p\}$ (with self-connection).
- **Distribution-level edges**. The edge between v_i and v_j is determined by the Euclidean distance between $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$. In detail, v_i is connected to v_j if $\boldsymbol{d}_i$ is among the k nearest neighbors of $\boldsymbol{d}_j$. That generates an adjacency matrix $\boldsymbol{A}_0$ (with self-connection).

Label-specific edges help learn label-specific embeddings, while distribution-level edges help learn global distribution embeddings.

We use the adjacency matrices $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_p\}$ and node features $\boldsymbol{X}$ to learn label-specific semantic embeddings through topology reconstruction. Likewise, we use the adjacency matrix $\boldsymbol{A}_0$ and node features $\boldsymbol{X}$ to learn (global) distribution-level semantic embeddings. We train GAE by reconstructing the adjacency matrices with the outputs of the decoder, which can be formulated as Formula (7):

$$\min_{\boldsymbol{W}_1,\boldsymbol{W}_2} \frac{1}{p}\sum_{j=1}^{p} \|\hat{\boldsymbol{A}}_j - \boldsymbol{A}_j\|_{\mathrm{F}}^2 + \|\hat{\boldsymbol{A}}_0 - \boldsymbol{A}_0\|_{\mathrm{F}}^2 \tag{7}$$

where $\hat{\boldsymbol{A}}_j$ is the output of GAE corresponding to $\boldsymbol{A}_j$.

3.2.2 *Label distribution learning by mining deep semantics*

After finishing training the GAE, we obtain the label-specific and distribution-level semantic embedding. The learned embedding has lower dimensions than the original features; therefore, learning LDL models in such an embedding space is more effective and reduces feature redundancy.

To consider LSF in LDL, we project the features of the input space onto the label-specific feature space. We define a weight matrix $\boldsymbol{U}$ and minimize the difference between $\boldsymbol{X}\boldsymbol{U}$ and $\boldsymbol{Z}_k$ as Formula (8):

$$\min_{\boldsymbol{U}} \sum_{k=1}^{p} \|\boldsymbol{X}\boldsymbol{U}\text{-}\boldsymbol{Z}_k\|_{\mathrm{F}}^2 \tag{8}$$

where $k = 1, 2, \ldots, p$. Meanwhile, instead of fitting an LDL model in the original feature space as Formula (3), we derive an LDL model in the label-specific semantic space, which can be recast as Formula (9):

$$\min_{\boldsymbol{W},\boldsymbol{U}} \sum_{k=1}^{p} \|\boldsymbol{X}\boldsymbol{U}\text{-}\boldsymbol{Z}_k\|_{\mathrm{F}}^2 + \|\boldsymbol{X}\boldsymbol{U}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 \tag{9}$$

In addition to label-specific semantics, our framework jointly learns distribution-level semantics through a graph autoencoder, which generates a latent representation $\boldsymbol{E} \in \mathbb{R}^{n\times h_2}$ that captures cross-label relationships. To align these semantics with label distributions, we learn a projection matrix $\tilde{\boldsymbol{U}}$ by Formula (10):

$$\min_{\tilde{\boldsymbol{U}}} \|\boldsymbol{X}\tilde{\boldsymbol{U}} - \boldsymbol{E}\|^2 \tag{10}$$

which effectively mines global distribution-level semantics. Similarly, we derive an LDL model in this semantic space, which recasts Formula (10) as Formula (11):

$$\min_{\boldsymbol{W},\tilde{\boldsymbol{U}}} \|\boldsymbol{X}\tilde{\boldsymbol{U}} - \boldsymbol{E}\|^2 + \|\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 \tag{11}$$

Next, we replace the original feature space by the label-specific and distribution-level semantic space. We formulate the problem of LAMPS as Formula (12):

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{U},\tilde{\boldsymbol{U}}} &\|\boldsymbol{X}\boldsymbol{U}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 + \|\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 \\ &+ \lambda_1\sum_{k=1}^{p}\|\boldsymbol{X}\boldsymbol{U}\text{-}\boldsymbol{Z}_k\|_{\mathrm{F}}^2 + \lambda_2\|\boldsymbol{X}\tilde{\boldsymbol{U}} - \boldsymbol{E}\|_{\mathrm{F}}^2 \\ &+ \lambda_3\left(\|\boldsymbol{W}\|_{\mathrm{F}}^2 + \|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\tilde{\boldsymbol{U}}\|_{\mathrm{F}}^2\right)\end{aligned} \tag{12}$$

where λ_1 and λ_2 are trade-off parameters, and λ_3 is a regularization parameter. Compared with Formula (3), LAMPS derives an LDL model in the joint label-specific and distribution-level semantic spaces.

The first two terms derive LDL models in the label-specific and distribution-level semantic spaces. The third term denotes the loss mappings from the input space to the label-specific semantic spaces for all labels. The fourth term is the loss mapping from the input space to the distribution-level semantic space. The last three terms are regularization on the parameters.

For an unknown instance $\boldsymbol{x}$, we fuse the prediction from the label-specific and distribution-level semantic spaces as the predicted label distribution $\hat{\boldsymbol{d}}$ in Eq. (13):

$$\hat{\boldsymbol{d}} = \frac{1}{2}\boldsymbol{W}^{\mathrm{T}}\left(\boldsymbol{U}^{\mathrm{T}} + \tilde{\boldsymbol{U}}^{\mathrm{T}}\right)\boldsymbol{x} \tag{13}$$

To satisfy the constraint of label distribution, we project the prediction onto a probability simplex [44] and obtain the final prediction.

3.3 Optimization

In this subsection, we elaborate on the optimization process. We employ alternating optimization to solve the optimization Formula (12). In each iteration, we solve the subproblem with one variable from $\{\boldsymbol{W}, \boldsymbol{U}, \tilde{\boldsymbol{U}}\}$ and fix the other two. Since each subproblem is a convex optimization problem, Formula (12) can be efficiently solved. Algorithm 1 summarizes the LAMPS approach.

3.3.1 *W-subproblem*

When $\boldsymbol{U}$ and $\tilde{\boldsymbol{U}}$ are fixed, the subproblem regarding $\boldsymbol{W}$ can be formulated as Formula (14):

$$\min_{\boldsymbol{W}} \|\boldsymbol{X}\boldsymbol{U}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 + \|\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{W} - \boldsymbol{D}\|_{\mathrm{F}}^2 + \lambda_3\|\boldsymbol{W}\|_{\mathrm{F}}^2 \tag{14}$$

The gradient of the above objective regarding $\boldsymbol{W}$ is defined in Eq. (15):

Just Accepted

Algorithm 1 LAMPS approach

Parameter: λ_1, λ_2, and λ_3

Input: $\boldsymbol{X}$, $\boldsymbol{D}$, $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, …, $\boldsymbol{Z}_p$, and $\boldsymbol{E}$

Output: $\boldsymbol{W}$, $\boldsymbol{U}$, and $\tilde{\boldsymbol{U}}$

1: Randomly initialize $\boldsymbol{W}$, $\boldsymbol{U}$, and $\tilde{\boldsymbol{U}}$;
2: **Repeat**
3: Update $\boldsymbol{W}$ according to Eq. (17);
4: Update $\boldsymbol{U}$ according to Eq. (20);
5: Update $\tilde{\boldsymbol{U}}$ according to Eq. (23);
6: **until** convergence;
7: **return** $\boldsymbol{W}$, $\boldsymbol{U}$, and $\tilde{\boldsymbol{U}}$

$$\nabla_W = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{U}\boldsymbol{W}-\boldsymbol{D}) + \tilde{\boldsymbol{U}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{W}-\boldsymbol{D}) + \lambda_3\boldsymbol{W} \quad (15)$$

Setting it to zero, we could derive a closed-form solution for $\boldsymbol{W}$, i.e.,

$$\boldsymbol{W} = \left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{U} + \tilde{\boldsymbol{U}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\tilde{\boldsymbol{U}} + \lambda_3\boldsymbol{I}\right)^{-1}(\boldsymbol{U}^{\mathrm{T}} + \tilde{\boldsymbol{U}}^{\mathrm{T}})\boldsymbol{X}^{\mathrm{T}}\boldsymbol{D} \quad (16)$$

where $\boldsymbol{I}$ is the identity matrix. Here, we apply gradient descent to update $\boldsymbol{W}$. In the t-th step, $\boldsymbol{W}$ is updated by Eq. (17):

$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} - \eta_W^{(t)}\nabla_W \quad (17)$$

where $\eta_W^{(t)}$ is the step length obtained from line search [45].

3.3.2 U-subproblem

When $\boldsymbol{W}$ and $\tilde{\boldsymbol{U}}$ are fixed, the subproblem regarding $\boldsymbol{U}$ is formulated as Formula (18):

$$\min_U \|\boldsymbol{X}\boldsymbol{U}\boldsymbol{W}-\boldsymbol{D}\|_{\mathrm{F}}^2 + \lambda_1\sum_{k=1}^{p}\|\boldsymbol{X}\boldsymbol{U}\text{-}\boldsymbol{Z}_k\|_{\mathrm{F}}^2 + \lambda_3\|\boldsymbol{U}\|_{\mathrm{F}}^2 \quad (18)$$

The gradient of the Formula (18) is in Eq. (19):

$$\nabla_U = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{U}\boldsymbol{W}-\boldsymbol{D})\boldsymbol{W}^{\mathrm{T}} + \lambda_1\sum_k \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{U}-\boldsymbol{Z}_k) + \lambda_3\boldsymbol{U} \quad (19)$$

We apply gradient descent to update $\boldsymbol{U}$ as Eq. (20):

$$\boldsymbol{U}_t = \boldsymbol{U}_{t-1} - \eta_U^{(t)}\nabla_U \quad (20)$$

where $\eta_U^{(t)}$ is the step length by line search.

3.3.3 $\tilde{\boldsymbol{U}}$-subproblem

When $\boldsymbol{W}$ and $\boldsymbol{U}$ are fixed, the subproblem regarding $\tilde{\boldsymbol{U}}$ reduces to Formula (21):

$$\min_{\tilde{U}} \|\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{W}-\boldsymbol{D}\|_{\mathrm{F}}^2 + \lambda_2\|\boldsymbol{X}\tilde{\boldsymbol{U}}-\boldsymbol{E}\|_{\mathrm{F}}^2 + \lambda_3\|\tilde{\boldsymbol{U}}\|_{\mathrm{F}}^2 \quad (21)$$

The gradient of Formula (21) can be calculated by Eq. (22):

$$\nabla_{\tilde{U}} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{W}-\boldsymbol{D})\boldsymbol{W}^{\mathrm{T}} + \lambda_2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\tilde{\boldsymbol{U}}-\boldsymbol{E}) + \lambda_3\tilde{\boldsymbol{U}} \quad (22)$$

Next, we apply gradient descent with line search to update $\tilde{\boldsymbol{U}}$ by

$$\tilde{\boldsymbol{U}}_t = \tilde{\boldsymbol{U}}_{t-1} - \eta_{\tilde{U}}^{(t)}\nabla_{\tilde{U}} \quad (23)$$

where $\eta_{\tilde{U}}^{(t)}$ is the step length by line search.

3.3.4 Computational cost

LAMPS needs computation complexity $\mathcal{O}(n^2 + nqp)$.

(1) The label-semantic encoding module has complexity $\mathcal{O}(n^2)$, since training the GAE costs $\mathcal{O}(n^2)$.

(2) The LDL module has computation complexity $\mathcal{O}(nqp)$. First, calculation of ∇_W needs complexity $\mathcal{O}(nqp)$, yielding complexity $\mathcal{O}(nqp)$ for the procedure of updating $\boldsymbol{W}$. Similarly, the procedures of updating $\boldsymbol{U}$ and $\tilde{\boldsymbol{U}}$ have complexity $\mathcal{O}(nqp)$.

Therefore, the total computational complexity is $\mathcal{O}(n^2 + nqp)$. The computation bottleneck of LAMPS lies in training the GAE, which needs quadratic complexity regarding n ($\mathcal{O}(n^2)$ complexity). Meanwhile, we can efficiently implement GAE on GPU with the PyTorch framework, which alleviates the high complexity of training GAE.

4 Experiment

In this section, we thoroughly evaluate the performance of LAMPS on 12 real-world datasets in comparison with several state-of-the-art LDL approaches over six different metrics.

4.1 Experimental configurations

4.1.1 Datasets

We conduct experiments on 12 real-world datasets with label distribution, as shown in Table 2. There are five datasets with different dimensions of labels on Yeast biological experiments [46], four facial expression datasets, one multi-label image database of natural scenes, one dataset for opinion prediction of pre-release movies, and one dataset for text emotion recognition. Table 2 summarizes some brief statistics of these datasets [1].

In detail, the first five datasets (from Yeast_cdc to Yeast_spo5) are collected from the biological experiments in budding yeast [46]. The yeast dataset includes 2465 yeast genes, which are represented by a phylogenetic profile vector of length 24, while having different dimensions of label distribution ranging from 3 to 15.

The sixth and seventh datasets (SJAFFE and SBU_3DFE) are extensions of two widely used facial expression image databases, i.e., JAFFE [47] and BU_3DFE [48]. SJAFFE contains 213 grayscale images with 243-dimensional features, and each image is annotated by 60 individuals on six basic emotions (happiness, sadness, surprise, fear, anger, and disgust) using a five-level scale. The average scores representing emotion intensity are regarded as the label distribution. SBU_3DFE includes 2500 images with 243-dimensional feature vectors. Likewise, each is scored by 23 annotators and the average scoring is treated as the label distribution.

Table 2 Statistics for 12 real-world datasets.

Index	Dataset	Number of examples	Number of features	Number of labels
1	Yeast_cdc	2 465	24	15
2	Yeast_diau	2 465	24	7
3	Yeast_heat	2 465	24	6
4	Yeast_spo	2 465	24	6
5	Yeast_spo5	2 465	24	3
6	SJAFFE	213	243	6
7	SBU_3DFE	2 500	243	6
8	Natural_Scene	2 000	294	9
9	Movie	7 755	1 869	5
10	Twitter_ldl	10 040	200	8
11	Flickr_ldl	11 150	200	8
12	Ren-Cecps	32 420	200	8

Just Accepted

The eighth dataset, Natural_Scene, involves 2000 natural scene images represented by 294-dimensional feature vectors [1]. Each image is labeled by 10 individuals across nine categories (plant, sky, cloud, snow, building, desert, mountain, water, and sun). Then, the inconsistent rankings are transformed into a label distribution via a nonlinear programming process [1].

The ninth dataset, Movie, consists of 7755 movies with 54 million ratings from 478 656 users [24]. For each movie, its label distribution represents the rating level percentages from one to five, with 1869-dimensional features extracted from the metadata, such as genre, directors, actors, and budget [24].

The tenth and eleventh datasets (Twitter_ldl and Flickr_ldl) are two image databases for visual sentiment analysis [49]. We preprocess these two datasets as Ref. [15].

The last dataset, Ren-Cecps, is a large-scale Chinese corpus of 32 420 blogs on emotional expression analysis [50]. Each is annotated with an eight-dimensional (emotion) label distribution.

4.1.2 Evaluation measures

Following Geng [1], we adopt six measures to evaluate the performance of the LDL algorithms, including four distance metrics and two similarity metrics. Chebyshev, Clark, Canberra, and KL-divergence [51] are in one group to measure the distance with lower values meaning better performance ("↓" means the lower the better); Cosine and Intersection [51] are in the other group to measure similarity with higher values indicating better performance ("↑" means the higher the better). For two label distributions $\boldsymbol{p}$ and $\boldsymbol{q}$, the formulas of the six measures are shown in Table 3, where p_i and q_i denote the i-th element of $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively. These six metrics comprehensively measure the performance of LDL algorithms from different perspectives.

4.1.3 Baselines and settings

We compare LAMPS with one representative LDL baseline SA-BFGS [1], and four state-of-the-art LDL algorithms, i.e., LDL-LDM [15], Adam-LDL-SCL [52], LDLSF [14], and LDL-LDSF [5]. The details of these comparison methods are as follows.

(1) SA-BFGS [1]: SA-BFGS takes the characteristics of label distribution into consideration, employing the maximum entropy model to learn label distribution.

(2) LDLSF [14]: LDLSF jointly learns the common and label-specific features to improve the performance of LDL.

(3) Adam-LDL-SCL [52]: Adam-LDL-SCL first encodes local correlations in the label space as additional features. Then, it jointly learns the label correlation encoding and label distribution.

(4) LDL-LDM [15]: LDL-LDM exploits the manifold structure of label distribution to model label correlation. It considers both global and local label correlation by exploiting label distribution manifold.

(5) LDL-LDSF [5]: LDL-LDSF first performs fuzzy C-means clustering and then leverages the cluster structures to supervise the learning of LSF.

In particular, SA-BFGS[†] is a representative LDL baseline without LSF and label correlation. Adam-LDL-SCL and LDL-LDM[‡] are two state-of-the-art LDL methods that learn label correlation in the LDL process. LDLSF and LDL-LDSF learn LSF in the LDL process. Therefore, there is diversity among the comparing algorithms.

The code for these algorithms is shared by the authors, and we use the suggested parameter settings as in the corresponding literature. Table 4 summarizes the used parameter settings in the experiments.

Table 3 Formula of six LDL evaluation metrics ($\boldsymbol{p}$ and $\boldsymbol{q}$ are two label distribution), where ↓ (↑) denotes the smaller (larger) the better.

Metric	Formula
Chebyshev (↓)	$\max_i \lvert p_i - q_i \rvert$
Clark (↓)	$\sqrt{\sum_i (p_i - q_i)^2 / (p_i + q_i)^2}$
Canberra (↓)	$\sum_i \lvert p_i - q_i \rvert / (p_i + q_i)$
KL-divergence (↓)	$\sum_i p_i \ln p_i / q_i$
Cosine (↑)	$\sum_i \frac{p_i q_i}{(\sqrt{\sum_i p_i^2} \sqrt{\sum_i q_i^2})}$
Intersection (↑)	$\sum_i \min(p_i, q_i)$

4.2 Experimental results and discussion

Tables 5 and 6 tabulate the comparative results across six methods on 12 datasets in terms of six LDL metrics, where the best results are highlighted in bold. Furthermore, we conduct the pairwise t-tests and use •/○ to indicate whether LAMPS is statistically superior/inferior to the comparing methods at a significance level of 0.05. The final rows summarize win/tie/loss counts.

According to Tables 5 and 6, LAMPS ranks first in 86.1% of cases (62 of 72) and achieves statistically better performance than other methods in 89.4% of comparisons (322 wins out of 360 on t-tests). To perform a comparative analysis, we conduct the Friedman tests [53] on each metric to test the equal performance of these methods. The p-values and the statistics are reported in the last column of Table 7. As shown in Table 7, for each metric, the null hypothesis that the comparing methods have equal performance is clearly rejected at the significance level of 0.05.

Next, we conduct post hoc tests [53] to further investigate the relative performance of LAMPS. We perform Bonferroni-Dunn tests [53] with LAMPS as the control method. One algorithm has statistically different performance from LAMPS if its mean rank differs from that of LAMPS by at least one Critical Difference (CD). For the tests, CD equals 1.97 at a significance level of 0.05. Figure 4 visualizes the results of the CD-diagrams in terms of six metrics. In each subfigure, the average ranks of all the methods are plotted on the axis (non-overlapping intervals indicate significant performance differences).

According to Tables 5 and 6 and Fig. 4, we can make the following observations.

(1) SA-BFGS is a representative baseline without LSF and label correlation. LAMPS significantly outperforms SA-BFGS across all metrics. On one hand, LAMPS learns LSF by mining label-specific semantics, on the other hand, LAMPS learns distribution-level semantics that model label correlation.

(2) LDL-LDM and Adam-LDL-SCL exploit label correlation, which helps enhance their performance. Although these two methods demonstrate comparable performance against LAMPS on some metrics, they derive LDL models in the original feature

[†] Code: https://palm.seu.edu.cn/xgeng/LDL/resource/LDLPackage_v1.2.zip

[‡] Code: https://github.com/wangjing4research/LDL-LDM

Table 4 Summary of the parameter settings for the comparison methods.

Method	Setting
Adam-LDL-SCL	$\lambda_1, \lambda_2, \lambda_3 \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}, m \in [0, 14]$
LDLSF	$\lambda_1, \lambda_2, \lambda_3 \in \{10^{-6}, 10^{-5}, \ldots, 10^{-1}\}, g \in [1, 14]$
LDL-LDM	$\lambda_1, \lambda_2, \lambda_3 \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}, g \in [1, 14]$
LDL-LDSF	$m = 2, \lambda = 0.1, g \in [2, 10]$
LAMPS	$\lambda_1, \lambda_2, \lambda_3 \in \{10^{-7}, 10^{-6}, \ldots, 10^{-1}\}$

Uncorrected Proof

Table 5 Results (mean±std) of the comparison methods in terms of Chebyshev, Clark, and Canberra, where the best results are highlighted in bold. We conduct the pairwise t-tests and use •/○ to indicate whether LAMPS is statistically superior/inferior to the comparing methods (at a significance level of 0.05). The win/tie/loss counts are summarized in the last row.

Metric	Dataset	SA-BFGS	LDL-LDM	Adam-LDL-SCL	LDLSF	LDL-LDSF	LAMPS
Chebyshev (↓)	Yeast_cdc	0.0168±0.0000•	0.0163±0.0016•	0.0166±0.0001•	0.0162±0.0011•	0.0165±0.0010•	**0.0160±0.0024**
	Yeast_diau	0.0373±0.0000•	0.0374±0.0012•	0.0370±0.0002•	0.0370±0.0000•	0.0370±0.0001•	**0.0369±0.0000**
	Yeast_heat	0.0427±0.0000•	0.0423±0.0008•	0.0421±0.0004	0.0423±0.0000•	0.0420±0.0001	**0.0420±0.0003**
	Yeast_spo	0.0583±0.0002•	0.0582±0.0028•	0.0586±0.0009•	0.0584±0.0001•	0.0585±0.0002•	**0.0581±0.0000**
	Yeast_spo5	0.0924±0.0000•	**0.0914±0.0038**	0.0921±0.0008•	0.0915±0.0001•	0.0916±0.0005•	**0.0914±0.0001**
	SJAFFE	0.0956±0.0047•	0.1152±0.0105•	0.0889±0.0070•	0.0875±0.0012•	0.0863±0.0015•	**0.0834±0.0015**
	SBU_3DFE	0.1144±0.0010•	0.1324±0.0032•	0.1309±0.0016•	0.1086±0.0002•	0.1152±0.0007•	**0.1079±0.0004**
	Natural_Scene	0.3463±0.0045•	0.3181±0.0084•	0.3551±0.0014•	0.3201±0.0009•	0.3091±0.0008•	**0.2857±0.0005**
	Movie	0.1498±0.0018•	0.1236±0.0058•	0.1639±0.0149•	0.1308±0.0000•	0.1420±0.0008•	**0.1067±0.0004**
	Flickr_ldl	0.2773±0.0013•	0.2735±0.0038•	0.2750±0.0080•	0.2691±0.0009•	0.2704±0.0101•	**0.2664±0.0000**
	Twitter_ldl	0.2911±0.0069•	0.2802±0.0029•	0.2939±0.0079•	0.2935±0.0017•	0.2800±0.0038•	**0.2786±0.0104**
	Ren-Cecps	0.2193±0.0023•	0.1825±0.0080•	0.1775±0.0168•	0.1812±0.0000•	0.1717±0.0000•	**0.1651±0.0005**
Clark (↓)	Yeast_cdc	0.2178±0.0002•	0.2160±0.0002•	0.2152±0.0009•	0.2161±0.0009•	0.2160±0.0003•	**0.2131±0.0000**
	Yeast_diau	0.2011±0.0003•	0.1993±0.0056•	0.1987±0.0008•	0.2008±0.0002•	0.2000±0.0007•	**0.1986±0.0000**
	Yeast_heat	0.1837±0.0000•	0.1830±0.0021•	0.1827±0.0017•	0.1829±0.0001•	0.1830±0.0001	**0.1823±0.0007**
	Yeast_spo	0.5147±0.0018•	0.5134±0.0282•	0.5157±0.0064•	0.5139±0.0004•	0.5130±0.0005•	**0.5127±0.0011**
	Yeast_spo5	0.1888±0.0000•	**0.1842±0.0088**○	0.1858±0.0019○	0.1843±0.0002○	0.1840±0.0005○	0.1861±0.0000
	SJAFFE	0.3702±0.0178•	0.4071±0.0245•	0.3289±0.0146•	0.3597±0.0020•	0.3119±0.0017•	**0.3051±0.0006**
	SBU_3DFE	0.3925±0.0017•	0.3993±0.0151•	0.3959±0.0036•	**0.3704±0.0006**○	0.3710±0.0007○	0.3755±0.0049
	Natural_Scene	2.4071±0.0245•	2.4184±0.0015•	2.4757±0.0042•	2.4188±0.0074•	2.0725±0.0173•	**1.7544±0.0149**
	Movie	0.6612±0.0060•	0.5486±0.0210•	0.6616±0.0348•	0.5688±0.0034•	0.5573±0.0279•	**0.4834±0.0015**
	Flickr_ldl	2.2279±0.0052•	2.2282±0.0279•	2.2119±0.0119•	2.2005±0.0208•	2.1877±0.0011•	**2.1707±0.0086**
	Twitter_ldl	2.5055±0.0079•	2.4620±0.0169•	2.4559±0.0119•	2.4507±0.0088•	2.4557±0.0017•	**2.4335±0.0020**
	Ren-Cecps	1.2144±0.0023•	0.9245±0.0080•	0.9754±0.0168•	0.9127±0.0000•	0.9003±0.0000•	**0.8510±0.0005**
Canberra (↓)	Yeast_cdc	0.6484±0.0032•	0.6347±0.0012•	0.6442±0.0004•	0.6331±0.0006•	0.6300±0.0001•	**0.6286±0.0002**
	Yeast_diau	0.4387±0.0001•	0.4339±0.0132•	0.4307±0.0018•	0.4309±0.0005•	0.4291±0.0000•	**0.4283±0.0001**
	Yeast_heat	0.3743±0.0001•	0.3679±0.0045•	0.3647±0.0028•	0.3643±0.0003•	0.3651±0.0005•	**0.3623±0.0002**
	Yeast_spo	0.0322±0.0003•	0.0248±0.0023•	0.0248±0.0005•	0.0246±0.0000	0.0247±0.0000•	**0.0240±0.0002**
	Yeast_spo5	0.2897±0.0000•	**0.2829±0.0126**○	0.2893±0.0026•	0.2831±0.0003○	0.2897±0.0002•	0.2863±0.0005
	SJAFFE	0.7577±0.0344•	0.8525±0.0718•	0.6781±0.0335•	0.7135±0.0041•	0.6538±0.0051•	**0.6308±0.0013**
	SBU_3DFE	0.8290±0.0047•	0.8693±0.0357•	0.8606±0.0083•	**0.7716±0.0014**○	0.7720±0.0011	0.7782±0.0064
	Natural_Scene	6.8758±0.0263•	6.6381±0.0624•	6.8220±0.0215•	6.7749±0.0023•	6.7750±0.0397•	**4.0456±0.0450**
	Movie	1.1965±0.0080•	1.0491±0.0408•	1.2748±0.0678•	1.2088±0.0000•	1.0887±0.0049•	**0.9123±0.0034**
	Flickr_ldl	5.5269±0.0019•	5.5119±0.0115•	5.5547±0.0677•	5.5111±0.0009•	5.5433±0.0079•	**5.0894±0.0203**
	Twitter_ldl	6.5977±0.0237	**6.4793±0.0459**○	6.6651±0.0338	6.6693±0.0694	6.6799±0.0793•	6.6703±0.0894
	Ren-Cecps	2.7931±0.0023•	2.4582±0.0080•	2.1677±0.0168•	**2.1287±0.0000**○	2.1670±0.0147•	2.1510±0.0005
Win/tie/loss counts		35/1/0	32/1/3	33/2/1	29/2/5	31/3/2	—

space and neglect LSF. In comparison, LAMPS learns LSF, which yields the superior performance of LAMPS.

(3) Among the baselines, LDLSF and LDL-LDSF learn LSF in the learning process. As a result, they achieve comparable performance to LAMPS on some metrics. However, LAMPS maintains superiority across multiple datasets. The reason may be that GAE mines superior deep semantics.

To summarize, LAMPS outperforms the comparison methods by a large margin, justifying its advantages for processing LDL data.

4.3 Ablation study

LAMPS has two novel components as follows:

(1) term of Label-Specific Semantics (LSS), i.e., $\sum_{k=1}^{p}\|\boldsymbol{XU}\text{-}\boldsymbol{Z}_k\|_{\mathrm{F}}^2$;

(2) term of Distribution-Level Semantics (DLS), i.e., $\|\boldsymbol{X}\tilde{\boldsymbol{U}}-\boldsymbol{E}\|_{\mathrm{F}}^2$.

Next, we conduct an ablation study to investigate the usefulness of LSS and DLS. In addition, we study the usefulness of the regularization term. We conduct an ablation study by setting the values of λ_1 and λ_2. We run LAMPS without LSS and DLS by setting $\lambda_1=0$ and $\lambda_2=0$, respectively. Similarly, we run LAMPS without regularization by setting $\lambda_3=0$.

Table 8 reports the results of the ablation study on SJAFFE in terms of six metrics. Moreover, we show an example on Natural_Scene and draw the ground-truth label distribution, as well as the label distribution predicted by LAMPS with (W/) and

Just Accepted

Table 6 Results (mean±std) of the comparison methods in terms of KL-divergence, Cosine, and Intersection, where the best results are highlighted in bold. We conduct the pairwise t-tests and use •/○ to indicate whether LAMPS is statistically superior/inferior to the comparing methods (at a significance level of 0.05). The win/tie/loss counts are summarized in the last row.

Metric	Dataset	SA-BFGS	LDL-LDM	Adam-LDL-SCL	LDLSF	LDL-LDSF	LAMPS
KL (↓)	Yeast_cdc	0.0072±0.0007•	0.0069±0.0017•	0.0070±0.0003•	0.0068±0.0062	0.0068±0.0001•	**0.0067±0.0003**
	Yeast_diau	0.0143±0.0002•	0.0137±0.0009•	0.0134±0.0001•	0.0130±0.0000•	0.0130±0.0000•	**0.0129±0.0002**
	Yeast_heat	0.0138±0.0000•	0.0130±0.0003•	0.0131±0.0003•	0.0126±0.0000•	0.0126±0.0000•	**0.0123±0.0007**
	Yeast_spo	0.0274±0.0000•	0.0258±0.0023•	0.0249±0.0005•	0.0246±0.0000•	0.0249±0.0000•	**0.0238±0.0023**
	Yeast_spo5	0.0300±0.0000•	0.0294±0.0021•	0.0298±0.0005•	0.0293±0.0001•	0.0290±0.0000	**0.0281±0.0007**
	SJAFFE	0.0550±0.0051•	0.0657±0.0098•	0.0442±0.0056•	0.0502±0.0026•	0.0413±0.0009•	**0.0324±0.0161**
	SBU_3DFE	0.0718±0.0008•	0.0773±0.0032•	0.0750±0.0012•	0.0602±0.0004	0.0602±0.0005	**0.0601±0.0023**
	Natural_Scene	0.9443±0.0101•	0.7772±0.0175•	0.9498±0.0064•	1.1288±0.0168•	0.9377±0.0168•	**0.6873±0.0013**
	Movie	0.1704±0.0044•	0.1097±0.0088•	0.1786±0.0200•	0.2417±0.0000•	0.1077±0.0028•	**0.0854±0.0004**
	Flickr_ldl	0.5133±0.0047•	0.5007±0.0018	0.5113±0.0029•	0.5117±0.0086•	0.5110±0.0029•	**0.5001±0.0045**
	Twitter_ldl	0.5577±0.0029•	0.5502±0.0100•	0.5477±0.0022•	0.5508±0.0230•	0.5443±0.0030•	**0.5408±0.0009**
	Ren-Cecps	0.3031±0.0032•	0.2745±0.0080•	0.2654±0.0168•	0.2627±0.0000•	0.2549±0.0001•	**0.2010±0.0005**
Cosine (↑)	Yeast_cdc	0.9965±0.0019•	0.9958±0.0009•	0.9942±0.0020•	0.9947±0.0062•	0.9947±0.0092•	**0.9967±0.0001**
	Yeast_diau	0.9857±0.0008•	0.9869±0.0008•	0.9874±0.0001•	0.9879±0.0003•	0.9880±0.0000	**0.9880±0.0000**
	Yeast_heat	0.9857±0.0000•	0.9879±0.0004•	0.9880±0.0002•	0.9880±0.0000•	0.9880±0.0000•	**0.9882±0.0003**
	Yeast_spo	0.9701±0.0000•	0.9748±0.0020•	0.9746±0.0005•	0.9759±0.0000•	0.9761±0.0000•	**0.9772±0.0001**
	Yeast_spo5	0.9734±0.0000•	0.9740±0.0018○	0.9738±0.0004	**0.9741±0.0001**○	0.9738±0.0001	0.9738±0.0001
	SJAFFE	0.9499±0.0046•	0.9620±0.0020•	0.9583±0.0056•	0.9597±0.0010•	0.9600±0.0071•	**0.9624±0.0017**
	SBU_3DFE	0.9298±0.0006•	0.9246±0.0029•	0.9264±0.0012•	**0.9440±0.0002**○	0.9430±0.0009•	0.9433±0.0002
	Natural_Scene	0.6795±0.0060•	0.7482±0.0034•	0.6651±0.0046•	0.7237±0.0011•	0.7559±0.0038•	**0.7675±0.0007**
	Movie	0.8984±0.0021•	0.9275±0.0060•	0.8808±0.0165•	0.9177±0.0000•	0.9297±0.0008•	**0.9443±0.0002**
	Flickr_ldl	0.8117±0.0104•	0.8128±0.0095•	0.8135±0.0073•	0.8230±0.0065•	0.8287±0.0077•	**0.8339±0.0008**
	Twitter_ldl	0.8401±0.0072•	0.8439±0.0025•	0.8430±0.0007•	0.8428±0.0009•	0.8420±0.0019•	**0.8552±0.0221**
	Ren-Cecps	0.7923±0.0023•	0.8045±0.0080•	0.8154±0.0168•	0.8227±0.0000•	0.8299±0.0000•	**0.8415±0.0005**
Intersection (↑)	Yeast_cdc	0.9584±0.0010•	0.9572±0.0018•	0.9576±0.0016•	0.9573±0.0020•	0.9580±0.0011•	**0.9588±0.0012**
	Yeast_diau	0.9402±0.0001•	0.9398±0.0020•	0.9404±0.0003	0.9403±0.0001•	0.9403±0.0011•	**0.9405±0.0018**
	Yeast_heat	0.9374±0.0000•	0.9401±0.0009•	0.9402±0.0004•	0.9402±0.0000•	0.9403±0.0000•	**0.9407±0.0001**
	Yeast_spo	0.9148±0.0000•	0.9155±0.0045•	0.9151±0.0010•	0.9154±0.0001•	0.9151±0.0001•	**0.9157±0.0004**
	Yeast_spo5	0.9068±0.0000•	**0.9086±0.0038**○	0.9079±0.0008•	0.9085±0.0001	0.9085±0.0001	0.9084±0.0007
	SJAFFE	0.8732±0.0058•	0.8549±0.0134•	0.8852±0.0068•	0.8852±0.0011•	0.8900±0.0011•	**0.8907±0.0007**
	SBU_3DFE	0.8505±0.0008•	0.8444±0.0058•	0.8459±0.0015•	0.8644±0.0002	0.8640±0.0003•	**0.8644±0.0001**
	Natural_Scene	0.5256±0.0043•	0.5283±0.0040•	0.4636±0.0015•	0.5397±0.0008•	0.5574±0.0010•	**0.5814±0.0013**
	Movie	0.7931±0.0023•	0.8245±0.0080•	0.7754±0.0168•	0.8127±0.0000•	0.8349±0.0000•	**0.8510±0.0005**
	Flickr_ldl	0.6448±0.0009•	0.6509±0.0007•	0.6443±0.0011•	0.6538±0.0028•	0.6559±0.0008	**0.6607±0.0093**
	Twitter_ldl	0.6559±0.0038•	0.6553±0.0047•	0.6606±0.0005•	0.6638±0.0042•	**0.6695±0.0074**	0.6693±0.0023
	Ren-Cecps	0.7031±0.0023•	0.7245±0.0080•	0.7254±0.0168•	0.7327±0.0000•	0.7339±0.0001•	**0.7610±0.0005**
Win/tie/loss count		36/0/0	33/1/2	34/2/0	30/4/2	29/7/0	—

Table 7 Friedman statistics χ_F^2 in terms of each metric and the p value at the significance level of 0.05.

Metric	χ_F^2	p-value
Chebyshev	31.19	8.59×10^{-6}
Clark	29.57	1.79×10^{-5}
Canberra	22.99	3.39×10^{-4}
KL-divergence	42.01	5.86×10^{-8}
Cosine	34.07	2.32×10^{-6}
Intersection	34.28	2.10×10^{-6}

without (W/O) LSS and DLS in Fig. 5.

From Table 8 and Fig. 5, we could make the following observations:

(1) When the LSS term is removed, the model performance declines by approximately 13.8% on average across six evaluation metrics. This demonstrates the critical role of LSS in capturing LSF. LAMPS struggles to accurately characterize the unique semantics of individual labels without LSS.

(2) When the DLS is removed, there is a significant performance drop, which justifies the role of DLS in processing data features and model learning. By modeling global distributional relationships among labels, DLS provides richer semantic information and enhances the performance of LAMPS.

(3) Removing the regularization term degrades the performance of LAMPS. The reason lies in that the absence of

Just Accepted

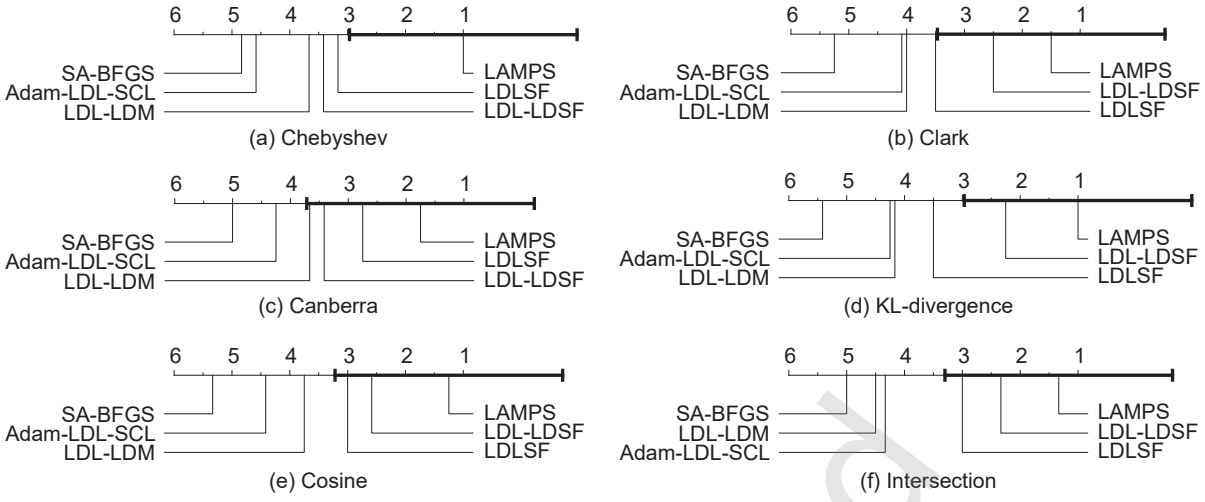


Fig. 4 CD diagrams of the Bonferroni-Dunn tests for learning full label distribution in terms of six metrics, where LAMPS serves as the control algorithm. The algorithms not connected with LAMPS are considered to have significantly different performance from LAMPS.

Table 8 Results of the ablation study on SJAFFE in terms of six metrics. ✔ and ✘ denote the inclusion and exclusion of the corresponding component, respectively.

Component			Chebyshev (↓)	Clark (↓)	Canberra (↓)	KL (↓)	Cosine (↑)	Intersection (↑)
LSS	DLS	Regularization						
✔	✘	✘	0.1176	0.3922	0.8559	0.0586	0.9383	0.8531
✘	✔	✘	0.1391	1.3285	1.9499	0.4270	0.9028	0.8083
✘	✘	✔	0.1224	0.4121	0.8958	0.0516	0.9315	0.8529
✔	✔	✘	0.1459	1.0968	1.7630	0.4520	0.8946	0.7998
✘	✔	✔	0.0874	0.3342	0.6520	0.0446	0.9606	0.8947
✔	✘	✔	0.1226	0.3879	0.8242	0.0643	0.9367	0.8560
✔	✔	✔	0.0834	0.3051	0.6308	0.0324	0.9624	0.8907

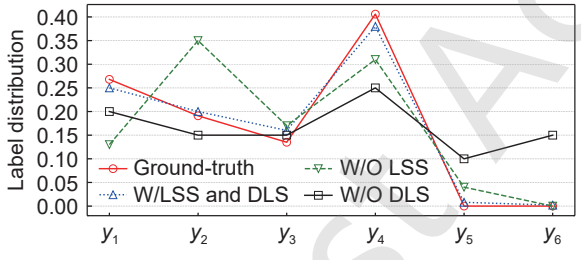


Fig. 5 Example on Natural_Scene to show the usefulness of LSS and DLS. We plot the ground-truth label distribution, as well as the label distribution by LAMPS with and without LSS and DLS.

regularization leads to overfitting on the training data and weakens generalization.

In summary, the results of the ablation study validate the usefulness of LSS and DLS, as well as the regularization term.

4.4 Running time

We study the running time of LAMPS. Table 9 compares the training/testing execution times of several LDL methods on the largest dataset Ren-Cecps. We run all methods on a Linux server with Intel i7-4790 CPU and 32 GB RAM. SA-BFGS, LDL-LDM, Adam-LDL-SCL, and LDL-LDSF are implemented in Python. We implement LAMPS with the PyTorch framework. Moreover, LDLSF is implemented with the MATLAB framework.

According to Table 9, LDLSF, LDL-LDSF, and LAMPS require additional time for LSF construction but achieve superior performance. LAMPS attains 3.2× speedup over LDLSF, which benefits from the use of neural networks. Although SA-BFGS has the least running time among all the methods, its speed comes at the cost of accuracy. In summary, LAMPS demonstrates state-of-the-art accuracy with competitive runtime efficiency.

Table 9 Training/test execution time of all compared algorithms (mean ± std) on the largest dataset Ren-Cecps.

(Unit: s)

Method	Training	Test
SA-BFGS	8.67±0.25	0.01±0.00
LDL-LDM	36.67±1.86	0.33±0.00
Adam-LDL-SCL	168.20±10.99	2.56±0.18
LDLSF	13 367.66±0.18	0.02±0.00
LDL-LDSF	25 079.39±20.91	0.05±0.00
LAMPS	4 221.39±0.28	2.64±0.11

4.5 Parameter analysis

LAMPS has several parameters: (1) λ_1, λ_2, and λ_3 control the influence of label-specific semantics, distribution-level semantics, and the regularization, respectively, and (2) h_1 and h_2 define the dimensions of the two-layer GCN.

To study the sensitivity of the parameters λ_1, λ_2, and λ_3, we run LAMPS on Ren-Cecps with varying values from the candidate set $\{10^{-7}, 10^{-6}, \ldots, 10^{-1}\}$. Figure 6 shows the performance of LAMPS in terms of KL-divergence. According to Fig. 6, LAMPS achieves better performance with $\lambda_1 \in [10^{-7}, 10^{-5}]$ and $\lambda_2 \in [10^{-7}, 10^{-5}]$. In addition, LAMPS achieves satisfactory performance when $\lambda_3 \in [10^{-3}, 10^{-1}]$. We could set $\lambda_1 = 10^{-5}$, $\lambda_2 = 10^{-5}$, and $\lambda_3 = 10^{-2}$, which yields better performance in the experiments. Generally speaking, LAMPS shows robust performance regarding λ_1, λ_2, and λ_3.

Furthermore, to investigate the sensitivity of h_1 and h_2, we run

Just Accepted

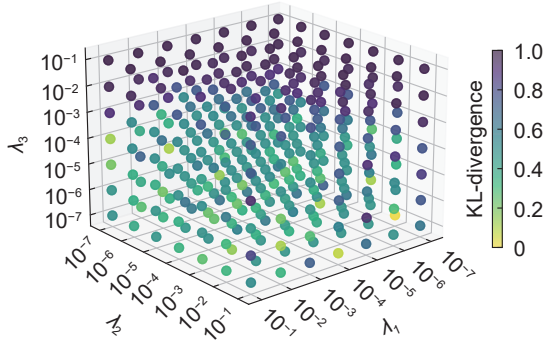


Fig. 6 Performance of LAMPS on Ren-Cecps in terms of KL-divergence with varying λ_1, λ_2, and λ_3 from the candidate set $\{10^{-7}, 10^{-6}, \ldots, 10^{-1}\}$.

LAMPS with varying values on Ren-Cecps in terms of KL-divergence, as reported in Fig. 7. According to Fig. 7, LAMPS is robust regarding h_2 and achieves better performance with $h_2 \leqslant 200$. As a result, in the experiments, we could set $h_1 = 150$ and $h_2 = 50$, which bring satisfactory performance.

5 Conclusion and discussion

In this paper, we present LAMPS, a novel LDL framework that jointly optimizes label-specific and distribution-level semantics through deep graph autoencoders. The key innovation lies in learning dual semantic representations, i.e., label-specific and distribution-level semantic encoding, which are integrated to enhance the original feature space. Then, the learned deep semantics enhance the performance of the LDL process. Extensive experiments across 12 benchmarks demonstrate the superiority of LAMPS, which achieves statistically significant improvements over existing methods in 89.4% of the comparisons while maintaining competitive efficiency, for example, LAMPS attains 3.2× speedup on the largest dataset, showing its superiority in processing large-scale LDL datasets.

While LAMPS achieves significant improvements for LDL, it still has several limitations. For example, the framework relies heavily on graph autoencoders to mine deep semantics, which may introduce computational overhead, particularly for very large-scale datasets. The performance of LAMPS depends on the quality of the constructed label-specific and distribution-level graphs, which may not generalize in cases of sparse or noisy labels. In addition, the current implementation assumes linear correlations between features and labels, potentially limiting its effectiveness for complex, nonlinear relationships. The predicted label distribution $\boldsymbol{XUW}$ may not satisfy the constraints of label description degrees. Moreover, the hyperparameters (λ_1, λ_2, and λ_3) require careful tuning, which could be challenging in practice without extensive validation.

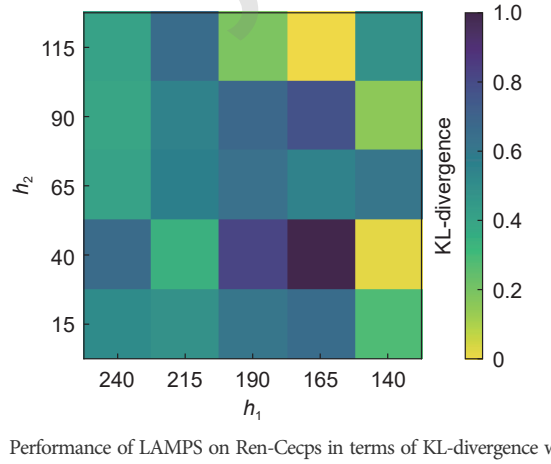


Fig. 7 Performance of LAMPS on Ren-Cecps in terms of KL-divergence with varying h_1 and h_2.

Future research will address these limitations by exploring several directions. We will incorporate more efficient graph representation techniques, such as scalable graph neural networks, to reduce computational costs and improve scalability. We could enhance the robustness of graph construction by integrating domain knowledge or adaptive edge weighting mechanisms that can better handle noisy or sparse label scenarios. Moreover, we will extend the framework to nonlinear models, for example, by integrating kernel methods or deep networks, which could further improve performance. We will investigate how to optimize the model parameters over a probability simplex [54] to satisfy the constraints of label description degrees.

Availability of Data and Materials

All data used in this study are publicly available.

Author Contributions

Jing Wang is responsible for conceptualization, formal analysis, and writing—review & editing. Jinyao Liu and Lingqi Liu are responsible for methodology, validation, and writing—original draft. All authors have approved the final manuscript.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62676087 and 62306073), the Natural Science Foundation of Liaoning Province (No. 2025-MSLH-214), and the Fundamental Research Funds for the Central Universities.

Declaration of Competing Interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] X. Geng, Label distribution learning, *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, 2016.

[2] B. B. Gao, C. Xing, C. W. Xie, J. Wu, and X. Geng, Deep label distribution learning with label ambiguity, *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2825–2838, 2017.

[3] Z. Deng, T. Li, D. Deng, K. Liu, Z. Luo, and P. Zhang, Feature selection for handling label ambiguity using weighted label-fuzzy relevancy and redundancy, *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 8, pp. 4436–4447, 2024.

[4] Y. He, J. Chen, J. Shen, P. Fournier-Viger, and J. Z. Huang, Density estimation-based method to determine sample size for random sample partition of big data, *Front. Comput. Sci.*, vol. 18, no. 5, p. 185322, 2024.

[5] X. Wang, J. D. Peter, A. Slowik, F. Zhang, and X. Xue, Learning fuzzy label-distribution-specific features for data processing, *IEEE Trans. Fuzzy Syst.*, vol. 33, no. 1, pp. 365–376, 2025.

[6] W. Yan, M. Wu, Y. Zhou, Q. Zheng, J. Chen, H. Cheng, and J. Zhu, Label distribution-driven multi-view representation learning, *Inf. Fusion*, vol. 115, p. 102727, 2025.

[7] S. Li and W. Deng, Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning, *Int. J. Comput. Vis.*, vol. 127, no. 6, pp. 884–906, 2019.

[8] X. Geng, C. Yin, and Z. H. Zhou, Facial age estimation by learning from label distributions, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2401–2412, 2013.

[9] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, and A. Yuille, Deep differentiable random forests for age estimation, *IEEE Trans. Pattern*

Just Accepted

Anal. Mach. Intell., vol. 43, no. 2, pp. 404–419, 2021.
[10] L. Jiang, H. Zhang, F. Tao, and C. Li, Learning from crowds with multiple noisy label distribution propagation, *IEEE Trans. Neural Network. Learn. Syst.*, vol. 33, no. 11, pp. 6558–6568, 2022.
[11] Y. Zhou, H. Xue, and X. Geng, Emotion distribution recognition from facial expressions, in *Proc. 23rd ACM Int. Conf. Multimedia*, Brisbane, Australia, 2015, pp. 1247–1250.
[12] Y. Jin, R. Gao, Y. He, and X. Zhu, GLDL: Graph label distribution learning, in *Proc. 38th AAAI Conf. Artificial Intelligence*, Vancouver, Canada, 2024, pp. 12965–12974.
[13] X. Jia, W. Li, J. Liu, and Y. Zhang, Label distribution learning by exploiting label correlations, in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 3310–3317.
[14] T. Ren, X. Jia, W. Li, L. Chen, and Z. Li, Label distribution learning with label-specific features, in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, Macao, China, 2019, pp. 3318–3324.
[15] J. Wang and X. Geng, Label distribution learning by exploiting label distribution manifold, *IEEE Trans. Neural Network. Learn. Syst.*, vol. 34, no. 2, pp. 839–852, 2023.
[16] Z. Wang, X. Xiao, and S. Rajasekaran, Novel and efficient randomized algorithms for feature selection, *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 208–224, 2020.
[17] M. L. Zhang and L. Wu, Lift: Multi-label learning with label-specific features, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, 2015.
[18] X. Xu, H. Wang, J. Yao, and Z. Zhang, Learning embedded label-specific features for partial multi-label learning, *Pattern Recognit.*, vol. 170, p. 112005, 2026.
[19] S. Lin, C. Wang, Y. Mao, and Y. Lin, Feature selection for label distribution learning under feature weight view, *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 5, pp. 1827–1840, 2024.
[20] Z. Zhang, J. Wang, and X. Geng, Label distribution learning by utilizing common and label-specific feature fusion space, *Int. J. Mach. Learn. Cybern.*, vol. 16, no. 3, pp. 1545–1558, 2025.
[21] A. Khelifa, H. Ghazouani, and W. Barhoumi, Label distribution learning for compound facial expression recognition in-the-wild: A comparative study, *Expert Syst.*, vol. 42, no. 2, p. e13724, 2025.
[22] D. Zhou, X. Zhang, Y. Zhou, Q. Zhao, and X. Geng, Emotion distribution learning from texts, in *Proc. Conf. Empirical Methods in Natural Language Processing*, Austin, TX, USA, 2016, pp. 638–647.
[23] Y. Zhang, J. Fu, D. She, Y. Zhang, S. Wang, and J. Yang, Text emotion distribution learning via multi-task convolutional neural network, in *Proc. 27th Int. Joint Conf. Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 4595–4601.
[24] X. Geng and P. Hou, Pre-release prediction of crowd opinion on movies by label distribution learning, in *Proc. 24th Int. Conf. Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 3511–3517.
[25] C. Wen, X. Zhang, X. Yao, and J. Yang, Ordinal label distribution learning, in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Paris, France, 2023, pp. 23424–23434.
[26] W. Qian, Y. Xiong, J. Yang, and W. Shu, Feature selection for label distribution learning via feature similarity and label correlation, *Inf. Sci.*, vol. 582, pp. 38–59, 2022.
[27] J. Wang, Z. Kou, Y. Jia, J. Lv, and X. Geng, Label distribution learning by exploiting fuzzy label correlation, *IEEE Trans. Neural Network. Learn. Syst.*, vol. 36, no. 5, pp. 8979–8990, 2025.
[28] J. Huang, G. Li, Q. Huang, and X. Wu, Learning label specific features for multi-label classification, in *Proc. IEEE Int. Conf. Data Mining*, Atlantic City, NJ, USA, 2015, pp. 181–190.
[29] J. Zhang, C. Li, D. Cao, Y. Lin, S. Su, L. Dai, and S. Li, Multi-label learning with label-specific features by resolving label correlations, *Knowl.-Based Syst.*, vol. 159, pp. 148–157, 2018.
[30] W. Weng, Y. Lin, S. Wu, Y. Li, and Y. Kang, Multi-label learning based on label-specific features and local pairwise label correlation, *Neurocomputing*, vol. 273, pp. 385–394, 2018.
[31] Y. Guo, F. Chung, G. Li, J. Wang, and J. C. Gee, Leveraging label-specific discriminant mapping features for multi-label learning, *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 2, p. 24, 2019.
[32] X. Y. Jia, S. S. Zhu, and W. W. Li, Joint label-specific features and correlation information for multi-label learning, *J. Comput. Sci. Technol.*, vol. 35, pp. 247–258, 2020.
[33] Z. Cheng and Z. Zeng, Joint label-specific features and label correlation for multi-label learning with missing label, *Appl. Intell.*, vol. 50, no. 11, pp. 4029–4049, 2020.
[34] J. Li, P. Li, X. Hu, and K. Yu, Learning common and label-specific features for multi-label classification with correlation information, *Pattern Recognit.*, vol. 121, p. 108259, 2022.
[35] Z. Chen and Z. Tan, Multi-label learning for label-specific features using correlation information with missing label, *Expert Syst. Appl.*, vol. 269, p. 126491, 2025.
[36] Y. Li, K. Wang, L. Tan, and F. Min, Label-specific disentanglement and correlation-guided fusion for multi-label classification, *Knowl. Inf. Syst.*, vol. 67, pp. 4991–5017, 2025.
[37] M. González, J. R. Cano, and S. García, ProLSFEO-LDL: Prototype selection and label-specific feature evolutionary optimization for label distribution learning, *Appl. Sci.*, vol. 10, no. 9, p. 3089, 2020.
[38] W. Shu, Q. Xia, and W. Qian, Label distribution feature selection based on label-specific features, *Appl. Intell.*, vol. 54, no. 19, pp. 9195–9212, 2024.
[39] W. Qian, Q. Ye, Y. Li, and S. Dai, Label distribution feature selection with feature weights fusion and local label correlations, *Knowl.-Based Syst.*, vol. 256, p. 109778, 2022.
[40] G. L. Li, H. R. Zhang, F. Min, and Y. N. Lu, Two-stage label distribution learning with label-independent prediction based on label-specific features, *Knowl.-Based Syst.*, vol. 267, p. 110426, 2023.
[41] Y. Lin, H. Liu, H. Zhao, Q. Hu, X. Zhu, and X. Wu, Hierarchical feature selection based on label distribution learning, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5964–5976, 2023.
[42] J. Zhang and Q. Xu, Attention-aware heterogeneous graph neural network, *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 233–241, 2021.
[43] X. Guo, D. Song, and F. Yang, Enhanced attention-driven dynamic graph convolutional network for extracting drug–drug interaction, *Big Data Mining and Analytics*, vol. 8, no. 1, pp. 257–271, 2025.
[44] W. Wang and M. A. Carreira-Perpinan, Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application, arXiv preprint arXiv:1309.1541, 2013.
[45] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
[46] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1998.
[47] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, Coding facial expressions with Gabor wavelets, in *Proc. 3rd IEEE Int. Conf. Automatic Face and Gesture Recognition*, Nara, Japan, 1998, pp. 200–205.
[48] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, A 3D facial expression database for facial behavior research, in *Proc. 7th Int. Conf. Automatic Face and Gesture Recognition*, Southampton, UK, 2006, pp. 211–216.
[49] J. Yang, M. Sun, and X. Sun, Learning visual sentiment distributions via augmented conditional probability neural network, in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 224–230.
[50] C. Quan and F. Ren, Construction of a blog emotion corpus for Chinese emotional expression analysis, in *Proc. Conf. Empirical Methods in Natural Language Processing*, Singapore, 2009, pp. 1446–1454.

Just Accepted

[51] S. H. Cha, Comprehensive survey on distance similarity measures between probability density functions, *Int. J. Math. Models Methods Appl. Sci.*, vol. 1, no. 4, pp. 300–307, 2007.
[52] X. Zheng, X. Jia, and W. Li, Label distribution learning by exploiting sample correlations locally, in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 4556–4563.
[53] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
[54] J. Chok and G. M. Vasil, Optimization over a probability simplex, *J. Mach. Learn. Res.*, vol. 26, no. 1, p. 73, 2025.



Jing Wang received the BEng degree from Suzhou University of Science and Technology, China, in 2013, the MEng degree from Northeastern University, China, in 2015, and the PhD degree from Southeast University, China, in 2021. He is currently an associate professor at School of Computer Science and Engineering, Southeast University, China. His research interests include pattern recognition and machine learning.



Jinyao Liu is currently an undergraduate student at Chien-Shiung Wu College, Southeast University, China. His research interests mainly include pattern recognition and machine learning.



Lingqi Liu received the BEng degree from Northeast Normal University, China, in 2022. She is currently a master's student at Southeast University, China. Her research interests include machine learning and data mining.

Just Accepted