LEARNING LABEL DISTRIBUTION WITH SUBTASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Label distribution learning (LDL) is a novel learning paradigm that emulates label polysemy by assigning label distributions over the label space. However, recent LDL work seems to exhibit a notable contradiction: 1) some existing LDL methods employ auxiliary tasks to enhance performance, which narrows their focus to specific domains, thereby lacking generalization capability; 2) conversely, LDL methods without auxiliary tasks rely on losses tailored solely to label distributions of the primary task, lacking additional supervised information to guide the learning process. In this paper, we propose $\mathcal{S}$-LDL, a novel and minimalist solution that partitions the label distribution of the primary task into subtask label distributions, i.e., a form of pseudo-supervised information, to reconcile the above contradiction. $\mathcal{S}$-LDL encompasses two key aspects: 1) an algorithm capable of generating subtasks without any extra knowledge, with subtasks deemed valid and reconstructable via our analysis; and 2) a plug-and-play framework seamlessly compatible with existing LDL methods, and even adaptable to derivative tasks of LDL. Experiments demonstrate that $\mathcal{S}$-LDL is effective and efficient. To the best of our knowledge, this represents the first endeavor to address LDL via subtasks. The code will soon be available on GitHub to facilitate reproducible research.

1 INTRODUCTION

Multi-label learning (MLL) (Zhang and Zhou, 2013) handles label polysemy in a binary manner, whereas label distribution learning (LDL) (Geng, 2016) offers a more nuanced perspective by answering: "*How much does each label y describe the instance $\boldsymbol{x}$?*". This is accomplished through the concept of a *label distribution* $\boldsymbol{d}$, which is a form of probability simplex that assigns a real value (i.e., *description degree* $d_{\boldsymbol{x}}^{y}$) to each label of each instance. This form introduces a quantitative manner to address label polysemy and extends LDL's practical applications to a wider range, e.g., counting (or grading) (Geng et al., 2013; Wu et al., 2019), sentiment analysis (Chen et al., 2020; Le et al., 2023), segmentation (Gao et al., 2017; Li et al., 2023b), etc. Concurrently, more and more derivative tasks of LDL (González et al., 2021b; Lu and Jia, 2022; Wang, Jing and Geng, Xin, 2019; Xu and Zhou, 2017; Xu et al., 2019) are emerging to offer assistance in various real-world dilemmas.

However, LDL encounters a spectrum of challenges: 1) label distributions are bound by two constraints, non-negativity (i.e., $d_{\boldsymbol{x}}^{y} \geq 0$) and sum-to-one (i.e., $\sum_{y \in \mathcal{Y}} d_{\boldsymbol{x}}^{y} = 1$), and are often formed from mixture distributions, posing significant hurdles for fitting, particularly when employing a maximum entropy model (Shen et al., 2017); 2) label distribution matrices are usually obtained via crowdsourcing, which is time-consuming and labor-intensive, so one often copes with scarce and low-quality datasets (Wang et al., 2023). These two key issues stand as formidable barriers to performance improvement in LDL.

With the widespread use of multi-task learning, some LDL work tries to compensate for performance from the perspective of *auxiliary tasks*, which are learned concurrently alongside the primary task, thereby refining its representations and ultimately boosting performance. Unfortunately, though these methods can exploit additional supervised information, they 1) do not address the first key issue mentioned above; and 2) require extra knowledge (e.g., facial characteristics (Chen et al., 2020), pathology criteria (Wu et al., 2019), emotion wheel theory in psychology (Yang et al., 2017a), etc.) or similar domain-specific data (Zhao et al., 2023b), limiting their generalization capability to those corresponding specific domains. Conversely, LDL methods that do not take advantage of auxiliary tasks, despite their efforts in loss function engineering and network structure design, they 1) do not address the second key issue mentioned above; and 2) focus solely on one aspect of label correlations

(e.g., correlation of local instances (Jia et al., 2019), ranking relation (Jia et al., 2023), suboptimal label (Wang, Jing and Geng, Xin, 2019), etc.), each with its own set of limitations.

The generalizability across various domains appears to conflict with the ability to exploit additional data, so benefiting from both simultaneously seems elusive. However, we can still see the light from some MLL methods, which partition the label space and apply operations on these subspaces (Tsoumakas et al., 2008; 2010). These methods construct *subtasks* without involving extra knowledge and exhibit applicability across various domains. Intuitively, in the context of LDL, reliable supervised information can be generated from these subtasks, which can eventually be aggregated and reconstructed to the information of the primary task via ensemble strategies. Although existing label distribution ensemble practices demonstrate promising performance (González et al., 2021a; Shen et al., 2017), they focus only on the supervised information of the primary task.

In this paper, we introduce $\mathcal{S}$-LDL, a novel and minimalist label distribution learning algorithm that constructs and exploits subtasks, to reconcile the contradiction between the generalizability across various domains and the ability to exploit additional data. Serving as auxiliary tasks, subtasks 1) provide different views of the primary task distribution, rendering the mixture of distributions more traceable (i.e., the key issue one); 2) furnish additional supervised data to mitigate the scarcity and ambiguity inherent in LDL datasets (i.e., the key issue two); 3) require no extra knowledge from specific domains; and 4) emphasize various label correlations via partitioning of the label space.

The main contributions of this paper are outlined below: 1) we propose $\mathcal{S}$-LDL, which is considered the first endeavor to address LDL via subtasks; 2) our analysis shows the validity and reconstructability of these subtasks; 3) we present a plug-and-play framework seamlessly compatible with existing LDL methods, and adaptable to derivative tasks of LDL; and 4) the code will be available on GitHub soon, facilitating reproducible research endeavors.

2 Related work

LDL Our work is mainly related to LDL. Initially employed to tackle age estimation (Geng et al., 2013), LDL has evolved into a novel machine learning paradigm (Geng, 2016), which is supported by theoretical underpinnings (Wang and Geng, 2019) and features various derivative tasks (González et al., 2021b; Lu and Jia, 2022; Wang, Jing and Geng, Xin, 2019; Xu and Zhou, 2017; Xu et al., 2019). Most methods focus on improving performance via loss function engineering (Jia et al., 2019; 2023; Ren et al., 2019; Wen et al., 2023) or efficient model structures (González et al., 2021a; Jin et al., 2024; Shen et al., 2017; Yang et al., 2017b), while some work is dedicated to practical application scenarios (Gao et al., 2017; Li et al., 2023a; Shirani et al., 2019; Wu et al., 2019). However, the scarcity of label distribution datasets and the complexity of the label distribution itself make it difficult to further improve performance, at which point one may think of leveraging auxiliary tasks.

LDL with auxiliary tasks While there are LDL methods that leverage auxiliary tasks to enhance performance, they often rely on knowledge from disparate domains, extending beyond the scope of the LDL task. For example, LDL-ALSG (Chen et al., 2020) designs auxiliary tasks dedicated to facial emotion recognition, necessitating the use of external tools to extract facial points and action units from human faces. Wu et al. (2019) exploit the Hayashi criterion, a rule for counting and grading in acne lesions, which results in their method being only applicable in a small branch of the dermatology field. Yang et al. (2017a) employ a multi-task framework for image emotion classification, designing constraints inspired by Mikel's wheel, a psychological emotion model, which also suffers from similar limitations. As LDL methods of transfer learning, GLDL (Zhao et al., 2023b) utilizes data from one or more source domains, which is not easy to obtain in practical applications. The need for specific extra knowledge significantly narrows the application scenarios of these methods.

MLL with partitioning of the label space For reference, there exist MLL methods based on partitioning of the label space, which can construct multi-label subtasks without involving additional knowledge and can be widely used in various domains. The most classic related work is that of HOMER (Tsoumakas et al., 2008) and RAkEL (Tsoumakas et al., 2010), the former forms a hierarchy of label subspaces while the latter randomly selects label subspaces. Many subsequent papers have been inspired by them (Prabhu et al., 2018; Read et al., 2013; Wang et al., 2021). Read et al. (2014) present a general framework of label subspaces and provide some theoretical justification for it. Since

Table 1: Key notation and terminology in this paper

Symbol	Description	Example
$\mathcal{Y} = \{y_j\}_{j=1}^{L}$	Label space (L labels)	$\mathcal{Y} = \{\texttt{HA}, \texttt{SA}, \texttt{SU}, \texttt{AN}, \texttt{DI}, \texttt{FE}\}$[1]
$\mathcal{Y}^{\circ}$	Subtask label spaces	$\{\cdots, \mathcal{Y}^{(t)} = \{\texttt{HA}, \texttt{SA}, \texttt{SU}, \texttt{FE}\}, \cdots\}$
$d_{\boldsymbol{x}_i}^{y_j}$	Description degree of $\boldsymbol{x}_i$ about y_j	$d_{\boldsymbol{x}_i}^{y_0} = 0.4$, i.e., HA describes $\boldsymbol{x}_i$ by 0.4
$\boldsymbol{d}_i = (d_{\boldsymbol{x}_i}^{y_j})_{j=1}^{L}$	Label distribution of $\boldsymbol{x}_i$	$\boldsymbol{d}_i = (0.4, 0.05, 0.3, 0.1, 0.1, 0.05)$
$\boldsymbol{D} = (\boldsymbol{d}_i)_{i=1}^{N} = (\boldsymbol{d}_{\bullet j})_{j=1}^{L}$	Distribution matrix (N samples)	$(\cdots, \boldsymbol{d}_i, \cdots)$
$\boldsymbol{d}_i^{(t)} = (d_{\boldsymbol{x}_i}^{(t)y_j})_{j=1}^{\lvert\mathcal{Y}^{(t)}\rvert}$	Subtask label distribution	$\boldsymbol{d}_i^{(t)} = (0.5, 0.0625, 0.375, 0.0625)$
$\mathcal{D}^{\circ}$	Subtask distribution matrices	$\{\cdots, \boldsymbol{D}^{(t)}, \cdots\}$
$\boldsymbol{M} = (\boldsymbol{m}_t)_{t=1}^{T} = (M_{tj})$	Mask matrix (T anticipated tasks)	$(\cdots, \boldsymbol{m}_t = (1, 1, 1, 0, 0, 1), \cdots)$

label distribution contains rich knowledge, we can follow the patterns of these methods to construct label distribution subtasks.

LDL with ensemble strategy It is imperative to aggregate the output of subtasks. Fortunately, ensemble-based LDL methods have demonstrated promising performance. For instance, LDLFs (Shen et al., 2017) learns different label distributions on the leaf nodes of differentiable decision trees and learns weights that aggregate these label distributions. DF-LDL (González et al., 2021a) aggregates the label distribution of output of multiple base models by simple averaging, while Zhai et al. (2018) focus on aggregating the results of various neural networks via a combining learner. However, 1) the above methods are not suitable for incomplete label spaces (i.e., subtask label spaces); and 2) *none* of them involve the partitioning of the label space, therefore no extra supervised information of label distributions is constructed.

Drawing from the analysis of the aforementioned related work, we introduce $\mathcal{S}$-LDL, which leverages pseudo-supervised information from subtasks to eliminate reliance on additional knowledge from disparate domains, and facilitates the creation of a novel knowledge dimension in a generic framework.

3 Subtask Construction

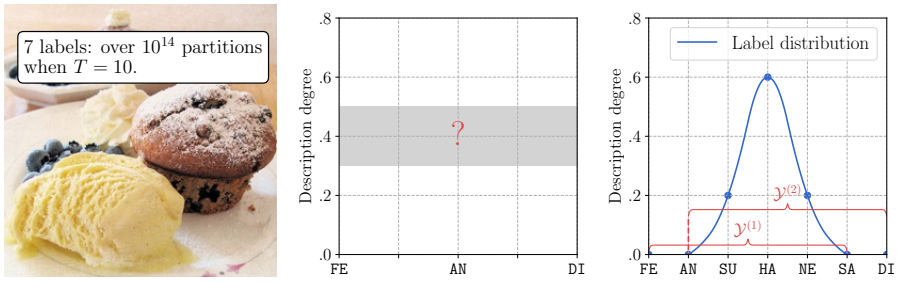
3.1 Preliminary

Notation Vectors are denoted by lowercase bold letters, e.g., $\boldsymbol{v}$, and the corresponding regular letter with subscript i, i.e., v_i, indicates its i-th element. Matrices are denoted by uppercase bold letters, e.g., $\boldsymbol{A}$. The row vector $\boldsymbol{a}_i$ indicates its i-th row and the column vector $\boldsymbol{a}_{\bullet j}$ indicates its j-th column. A_{ij} is the element in i-th row and j-th column of $\boldsymbol{A}$. The superscript (t) indicates that a symbol corresponds to the t-th subtask. Table 1 outlines the key notation in this paper.

Problem definition Let $\boldsymbol{x} \in \mathbb{R}^P$ denote the feature of the instance and $\boldsymbol{d} \in \Delta^{L-1}$ denote the label distribution, where $\Delta^{k-1} \triangleq \{\boldsymbol{v} \in \mathbb{R}^k \mid \boldsymbol{1}\boldsymbol{v}^{\mathrm{T}} = 1, \boldsymbol{v} \geq 0\}$ is the $(k-1)$-dimensional probability simplex. The goal of LDL is to find a mapping $\zeta : \boldsymbol{x} \mapsto \boldsymbol{d}$. In this paper, we partition the label space $\mathcal{Y}$ corresponding to $\boldsymbol{d}$ to obtain the subtask label space set $\mathcal{Y}^{\circ}$, then accordingly generate pseudo-supervised information, i.e., subtask distribution matrix set $\mathcal{D}^{\circ}$, to guide the learning of ζ.

Technical challenges Our first challenge arises from *the exponential growth in partitions* as the number of labels increases (Tsoumakas et al., 2010). When generating T tasks from a label space with L labels, the number of unique partitions is given by $(2^L-L-2)!/(T!(2^L-L-2-T)!)$. This makes it impractical to calculate metric for each case to select subtasks. We tackle this challenge in a mask matrix learning manner. The second challenge lies in *discerning reasonable partitions*. Since the label distribution matrix is usually imbalanced in average description degree (Zhao et al., 2023a), some partitions exhibit unreasonable local ignorance. As a result, the corresponding spaces struggle to handle the majority of instances, because 1) theoretically, there is no objective standard for the degree of negative correlation; and 2) empirically, weakly or negatively correlated information is

[1] HA, SA, SU, AN, DI, FE, and NE represent the seven common emotions in sentiment analysis datasets, namely happiness, sadness, surprise, anger, disgust, fear, and neutral, respectively.



(a) Huge number of partitions (b) Unreasonable ignorance (c) Lacking new knowledge [2]

Figure 1: **(a)** is sourced from the `emotion6` (Yang et al., 2017b) dataset, which has only 7 labels, but the number of potential partitions is huge. **(b)** exemplifies a subtask label space $\{\texttt{FE}, \texttt{AN}, \texttt{DI}\}$, which is challenging to describe (a). **(c)**'s two subtask label spaces encompass all descriptive information, meaning no new knowledge is generated about (a). This example vividly illustrates the limitations that may result from local ignorance and lack of diversity in subtask label spaces.

easily overlooked by human annotators in crowdsourced datasets. To mitigate this, we incorporate the description degree as a metric for the reliability of supervised information in guiding the generation of subtask masks. The third challenge is *avoiding analogous pseudo-supervised information*, i.e., to generate label distributions containing new knowledge (González et al., 2021b). This necessitates fostering richness and diversity in both subtask label spaces and distributions. To achieve this objective, we 1) minimize pairwise similarity among subtask masks; and 2) normalize each subtask label distribution to yield brand new insights distinct from the label distribution of the primary task. Fig. 1 portraits an illustrative example of these challenges.

3.2 Learning Subtask Masks

Let $\boldsymbol{M} \in \{0, 1\}^{T \times L}$ denote the subtask mask matrix, where T represents the number of anticipated tasks. To ensure that the subtask label spaces contain as reliable information as possible, the learning of the subtask mask matrix can be converted into this problem: $\arg\max_{\boldsymbol{M}} \|\boldsymbol{D}\boldsymbol{M}^\top\|_{\mathrm{F}}$.

Obviously, a senseless solution is $\boldsymbol{m}_t = \mathbf{1}$ where $t = 1, \cdots, T$, i.e., all pseudo-supervised information is equivalent to the primary task information. Therefore, solving the above problem alone is inappropriate. To address this, we consider pairwise similarity among subtask masks. We also employ exponential tricks to convert maximization into minimization. Finally, $\boldsymbol{M}$ is calculated as

$$\boldsymbol{M}^* = \arg\min_{\boldsymbol{M}} \left(\frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} \exp\left(-\boldsymbol{d}_i \boldsymbol{m}_t^\top\right) + \frac{2\lambda}{(T(T-1))} \sum_{i,j,i\neq j} \frac{\boldsymbol{m}_i \boldsymbol{m}_j^\top}{\|\boldsymbol{m}_i\| \, \|\boldsymbol{m}_j\|} \right), \quad \text{s.t. } M_{tj} \in \{0, 1\};\ t = 1, \cdots, T;\ j = 1, \cdots, L, \tag{1}$$

where λ is a trade-off parameter. Eq. (1) is slightly more complicated than conventional integer programming. For convenience, we solve it using the stochastic gradient descent (SGD) method, with its constraint enforced via sigmoid (a conversion threshold is set, where outputs greater than it are set to 1, while those below are set to 0). Refer to Section 4.1 for an analysis of the validity of Eq. (1).

3.3 Generating Subtask Distributions

We slice the label distribution matrix according to the subtask label space. To generate diversified subtask label distributions, we perform normalization on each subtask distribution with

$$[\mathcal{N}_{\mathrm{SUM}}(\boldsymbol{v})]_j = \frac{v_j}{\sum_{i=1}^{|\boldsymbol{v}|} v_i}. \tag{2}$$

[2] Despite the discrete label space, in the field of LDL, the label distribution is intentionally plotted as a curve, to distinguish it from the logical labels.

Algorithm 1 Subtask construction

Input: Input matrix $\boldsymbol{D}$, trade-off parameter λ, anticipated number of subtasks T.
Output: Subtask distribution matrices $\mathcal{D}^\circ$ (with corresponding subtask label spaces $\mathcal{Y}^\circ$).

Initialization: $\mathcal{Y}^\circ = \{\varnothing\}$, $\mathcal{D}^\circ = \{\varnothing\}$;
Calculate $\boldsymbol{M}$ using SGD; ▷ (Eq. (1))
for $t = 1$ **to** T **do**
  $\mathcal{Y}^{(t)} \leftarrow \{y_j\}$ **if** $M_{tj} = 1$;
  if $|\mathcal{Y}^{(t)}| = L$ **or** $|\mathcal{Y}^{(t)}| \leq 1$ **then**
    continue; /* Ignore invalid subtask masks. */
  end if
  /* The clip$(x,\ a,\ b)$ function limits x to be within $[a,\ b]$. */
  $\boldsymbol{D}^{(t)} \leftarrow \text{clip}(\boldsymbol{d}_{\bullet j},\ \varepsilon,\ 1)$ **if** $y_j \in \mathcal{Y}^{(t)}$; /* ε is a very small positive number. */
  for $i = 1$ **to** N **do**
    Normalization: $\boldsymbol{d}_i^{(t)} \leftarrow \mathcal{N}_{\text{SUM}}(\boldsymbol{d}_i^{(t)})$; ▷ (Eq. (2))
  end for
  $\mathcal{Y}^\circ = \mathcal{Y}^{(t)} \cup \mathcal{Y}^\circ$, $\mathcal{D}^\circ = \boldsymbol{D}^{(t)} \cup \mathcal{D}^\circ$;
end for

Algorithm 2 $\mathcal{S}$-LDL (shallow regime)

Input: Feature matrix $\boldsymbol{X}$, label distribution matrix $\boldsymbol{D}$, testing instance $\boldsymbol{x}'$.
Output: Predicted label distribution $\boldsymbol{d}'$ for instance $\boldsymbol{x}'$.

Initialize parameter of each estimator;
$\mathcal{D}^\circ \leftarrow \texttt{SC}(\boldsymbol{D})$;
for $t = 1$ **to** $|\mathcal{D}^\circ|$ **do**
  Fit an estimator $f^{(t)}$ on dataset $\{\boldsymbol{X},\ \boldsymbol{D}^{(t)}\}$;
  $\boldsymbol{d}^{(t)\prime} \leftarrow f^{(t)}(\boldsymbol{x}')$;
end for
Concatenate $\boldsymbol{X}$ and all of the $\boldsymbol{D}^{(t)}$s to get $\boldsymbol{Z}$, where $t = 1,\ \cdots,\ |\mathcal{D}^\circ|$;
Fit an estimator f on dataset $\{\boldsymbol{Z},\ \boldsymbol{D}\}$;
Concatenate $\boldsymbol{x}'$ and all of the $\boldsymbol{d}^{(t)\prime}$s to get $\boldsymbol{z}'$, where $t = 1,\ \cdots,\ |\mathcal{D}^\circ|$;
$\boldsymbol{d}' \leftarrow f(\boldsymbol{z}')$;

The rationale for utilizing $\mathcal{N}_{\text{SUM}}$ as the normalization function can be found in Section 4.2. The overall subtask construction process, denoted by SC, is illustrated in Alg. 1. Then, one can naturally come up with an adaptive LDL pipeline based on the shallow regime, as depicted in Alg. 2.

4 Analysis about subtask construction

In this section, we analyze the subtask construction algorithm SC by studying the following questions:

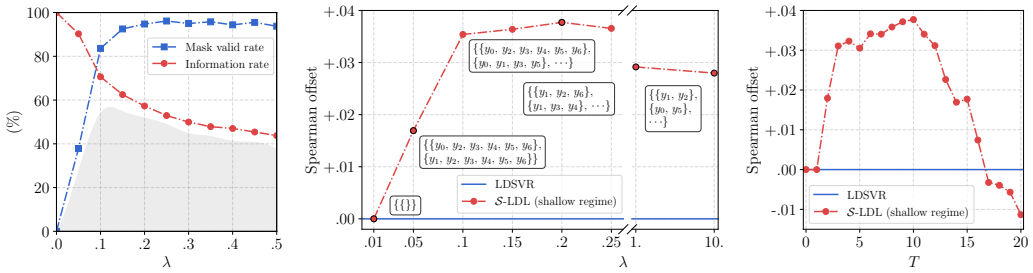
- $\mathcal{Q}_1$: Are the subtask spaces provided by Eq. (1) valid for performance improvement? Can one configure λ and T in Eq. (1) without any prior knowledge?
- $\mathcal{Q}_2$: Are the subtask label distributions provided by Eq. (2) reconstructable? Can one replace Eq. (2) with other normalization functions?
- $\mathcal{Q}_3$: What is the overall time complexity of SC? Is it practical for large-scale datasets?

The validity, reconstructability, and complexity analysis are conducted for $\mathcal{Q}_1$, $\mathcal{Q}_2$, and $\mathcal{Q}_3$, respectively.

4.1 Validity analysis

Eq. (1) manages the intricate task of selecting subtask spaces via λ and T. On the one hand, we strive to explain that it is useful for performance improvement to suppress local ignorance and increase diversity of each subtask label space simultaneously. On the other hand, we seek to determine the appropriate λ and T without any prior knowledge. To this end, we design the following two metrics.

Definition 1 (Information rate)**.** *We call it informative if $M_{tj} = 1$ where $t = 1,\ \cdots,\ T$ and $j = 1,\ \cdots,\ L$. Let I be the summation of information; we define the information rate as $I/(TL)$.*



(a) The two metrics w.r.t. λ (b) Spearman offset w.r.t. λ (c) Spearman offset w.r.t. T

Figure 2: Visualized results of the validity analysis. Results of **(a)** are the average of experiments on all datasets (introduced in the appendix), while results of **(b)** and **(c)** are on the `emotion6` dataset. The blue lines in (b) and (c) represent performance without auxiliary tasks.

Definition 2 (Mask valid rate)**.** *Let $\delta(\cdot, \cdot)$ be the Kronecker delta function. For all $t = 1, \cdots, T$, the following are considered counting of invalid masks: 1) $\delta(\boldsymbol{m}_t, \mathbf{1})$, or 2) $\delta(|\boldsymbol{m}_t|!, 1)$, or, 3) excluding masks in cases 1) and 2), for any remaining mask index i, $\delta(\boldsymbol{m}_i, \boldsymbol{m}_j)$, where $j = 1, \cdots, i$.*[3] *Let S be the summation of all invalid subtasks; we define the mask valid rate as $1 - S/T$.*

While it is unknown which metric is more important, we intuitively claim that higher values for both metrics are likely to lead to better performance. First, we calculate the average of these two metrics for all datasets with varying λ, results of which are shown in Fig. 2(a). The grey area depicts the average of the two metrics, implying that the appropriate value of λ may be greater than 0.1.

It is important to highlight that Alg. 2 relies on naive concatenation operations and is not tied to representation learning. Consequently, any performance improvement over the base estimator is solely attributed to the effects of the subtask label distributions. Therefore, we employ Alg. 2 as the "scaffolding" for our analysis. With λ varying and T fixed at 10, we conduct ten-fold experiments repeated 10 times on the `emotion6` dataset using Alg. 2. Here, $f^{(t)}$s and f are implemented by a representative LDL method, LDSVR (Geng and Hou, 2015). We record the average Spearman's coefficient (the higher the better). The results, which are shown in Fig. 2(b), support our claim. The panels from left to right display examples of subtask label spaces when the λ is 0.01, 0.05, 0.2, 1, and 10, respectively. When λ is suitable, label spaces are diverse and do not have excessive local ignorance; as λ decreases, label spaces tend to be homogeneous, and invalid masks account for the majority; as λ increases, the local ignorance of each label space becomes significant.

Besides, we also study the parameter sensitivity of T with λ fixed at 0.2. Results are shown in Fig. 2(c), illustrating that having a plethora of auxiliary tasks are detrimental to performance, which may be due to overfitting.

The validity analysis demonstrates that simultaneously avoiding local ignorance and homogeneity can lead to more efficient subtask label spaces, thereby improving performance. Without any prior knowledge, λ and T are recommended to be set to 0.2 and 10, respectively. Since the validity of Eq. (1) is ensured, one may wonder about the rationality and necessity of Eq. (2).

4.2 Reconstructability Analysis

We strive to choose a normalization function so that subtask label distributions retain more information, even efficacious enough to reconstruct the label distribution of the primary task. Theorem 1 illustrates that Eq. (2) is the only possibility.

Theorem 1. *Let each subtask label space form a connected graph with its each label as a node. Then merge these graphs according to their respective labels to form $\mathcal{G}$. If and only if $\mathcal{N}_{SUM}$ is used for normalization, the primary label distribution can be reconstructed from these subtask label distributions, when the following conditions are satisfied: 1) $\mathcal{G}$ is connected; 2) $\mathcal{G}$ covers all labels in the label space, and 3) corresponding description degrees of all cut vertices of $\mathcal{G}$ are not zero.*

[3]These three cases correspond to 1) masks that are exactly the same as the primary task; 2) masks that fail to form label distributions; and 3) duplicate masks among the remaining masks, respectively.

Proof. We solely discuss the extreme case where two subtask label spaces overlap with just one label. Further specialized cases can be deduced by the reader via induction. With a little bit of symbol abuse, let the general normalization function be defined as $\mathcal{N}(\boldsymbol{v}) \triangleq {}^{p(\boldsymbol{v})}/_{q(\boldsymbol{v})}$. Assume that there is a label distribution $\boldsymbol{d} = (d_1, \cdots, d_L)$ and its corresponding label space is $\mathcal{Y} = \{y_1, \cdots, y_L\}$. The two decompositions of $\mathcal{Y}$ are $\mathcal{Y}_{\boldsymbol{a}} = \{y_1, \cdots, y_k\}$ and $\mathcal{Y}_{\boldsymbol{b}} = \{y_k, \cdots, y_L\}$, respectively. It is obvious that $\mathcal{Y}_{\boldsymbol{a}} \cup \mathcal{Y}_{\boldsymbol{b}} = \mathcal{Y}$ and $\mathcal{Y}_{\boldsymbol{a}} \cap \mathcal{Y}_{\boldsymbol{b}} = \{y_k\}$. Let the subspace label distribution corresponding to these two decompositions be $\boldsymbol{a} = (a_1, \cdots, a_k)$ and $\boldsymbol{b} = (b_k, \cdots, b_L)$. According to our assumptions, $d_k \neq 0$. Then, for any integer $j \in [1, k]$, we have

$$\frac{a_j}{a_k} = \frac{[\mathcal{N}(\boldsymbol{d})]_j}{[\mathcal{N}(\boldsymbol{d})]_k} = \frac{[p(\boldsymbol{d})]_j}{[q(\boldsymbol{d})]_j} \frac{[q(\boldsymbol{d})]_k}{[p(\boldsymbol{d})]_k}. \tag{3}$$

Typically, for most normalization functions, $q(\cdot)$ is a normalizing constant, i.e., $[q(\boldsymbol{d})]_j = [q(\boldsymbol{d})]_k$. Thus Eq. (3) can be rewritten into $a_j[p(\boldsymbol{d})]_k = a_k[p(\boldsymbol{d})]_j$. Plug it into $\sum_{j=1}^{k} a_j = 1$, and do the same for $\boldsymbol{b}$ as well, and get

$$\frac{a_k \sum_{j=1}^{k}[p(\boldsymbol{d})]_j}{[p(\boldsymbol{d})]_k} = 1, \quad \frac{b_k \sum_{j=k}^{L}[p(\boldsymbol{d})]_j}{[p(\boldsymbol{d})]_k} = 1. \tag{4}$$

Add these two equations together, we have

$$[p(\boldsymbol{d})]_k + \sum_{j=1}^{L}[p(\boldsymbol{d})]_j = \frac{[p(\boldsymbol{d})]_k}{a_k} + \frac{[p(\boldsymbol{d})]_k}{b_k}. \tag{5}$$

Eq. (5) implies that $\sum_{j=1}^{L}[p(\boldsymbol{d})]_j$ must be given, and $[p(\boldsymbol{d})]_k$ is related to d_k, and only d_k. To make it possible, the only thing we can exploit is the sum-to-one constraint of $\boldsymbol{d}$, i.e., $\sum_{j=1}^{L} d_j = 1$. Therefore $[p(\boldsymbol{v})]_j = v_j$. Since $\sum_{i}^{|\boldsymbol{v}|}[\mathcal{N}(\boldsymbol{v})]_i = 1$, we have $q(\boldsymbol{v}) = \sum_{i}^{|\boldsymbol{v}|} v_i$, i.e., the finally deduced normalization function is Eq. (2). In this case, for any integer $j \in [1, L]$, we have

$$d_j = \begin{cases} \dfrac{a_j b_k}{a_k + b_k - a_k b_k}, & j = 1, \cdots, k \\ \dfrac{a_k b_j}{a_k + b_k - a_k b_k}, & j = k+1, \cdots, L \end{cases}, \tag{6}$$

which illustrates that the original label distribution $\boldsymbol{d}$ can be reconstructed by subtask label distributions $\boldsymbol{a}$ and $\boldsymbol{b}$. This is possible thanks to the use of $\mathcal{N}_{\text{SUM}}$. □

Theorem 1 also states that it is not appropriate to replace Eq. (2) with the min-max or softmax function because doing so destroys the reconstruction information.

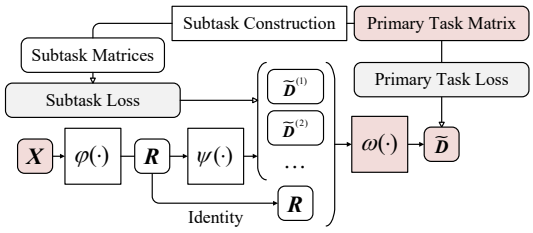


Figure 3: The overview of $\mathcal{S}$-LDL (deep regime). White, red and gray highlight our proposed, existing methods, and loss functions, respectively.

4.3 Complexity analysis

The overall time cost of SC is primarily influenced by the calculation of $\boldsymbol{M}$ and the normalization process. The time complexity of computing and updating $\boldsymbol{M}$ are $\mathcal{O}(L(TN + T^2))$ and $\mathcal{O}(LT)$, respectively. The time complexity of the normalization process is $\mathcal{O}(LTN)$. The overall time complexity of each iteration of SC is $\mathcal{O}(L(TN + T^2))$, which is linear with respect to the number of instances and labels. Therefore, it is clear that SC can be applied to large-scale datasets.

5 $\mathcal{S}$-LDL of the deep regime

The aforementioned analysis has exposed the problems of the shallow regime: 1) shallow methods as base estimators have low potential in themselves; 2) there is a training gap between the primary task and subtasks, i.e., no representation learning is involved. Therefore, it is necessary to introduce our proposed $\mathcal{S}$-LDL of the deep regime, the overview of which is illustrated in Fig. 3. We illustrate our framework by introducing the learnable parts one by one.

Table 2: Modifications of different task adaptations

Type	Subtask construction	ℓ_{PRI}	ℓ_{SUB}
Vanilla LDL	$(\boldsymbol{D}^{(1)}, \cdots) \leftarrow \texttt{SC}(\boldsymbol{D})$	$\ell(\boldsymbol{D}, \tilde{\boldsymbol{D}}) \in \mathcal{L}_{\text{LDL}}$	$\ell_{\text{SUB}}(\boldsymbol{D}^{(1)}, \cdots; \tilde{\boldsymbol{D}}^{(1)}, \cdots)$
LDL4C	$(\boldsymbol{D}^{(1)}, \cdots) \leftarrow \texttt{SC}(\boldsymbol{D})$	$\ell(\boldsymbol{D}, \bar{\boldsymbol{D}}, \tilde{\boldsymbol{D}}) \in \mathcal{L}_{\text{LDL4C}}$	$\ell_{\text{SUB}}(\boldsymbol{D}^{(1)}, \cdots; \tilde{\boldsymbol{D}}^{(1)}, \cdots)$
IncomLDL	$(\boldsymbol{D}_{\Omega}^{(1)}, \cdots) \leftarrow \texttt{SC}(\mathcal{R}_{\Omega}(\boldsymbol{D}))$	$\ell(\mathcal{R}_{\Omega}(\boldsymbol{D}), \mathcal{R}_{\Omega}(\tilde{\boldsymbol{D}})) \in \mathcal{L}_{\text{IncomLDL}}$	$\ell_{\text{SUB}}(\boldsymbol{D}_{\Omega}^{(1)}, \cdots; \mathcal{R}_{\Omega}(\tilde{\boldsymbol{D}}^{(1)}), \cdots)$
LE	$(\boldsymbol{L}^{(1)}, \cdots) \leftarrow \texttt{SC}(\boldsymbol{L})$	$\ell(\boldsymbol{L}, \tilde{\boldsymbol{D}}) \in \mathcal{L}_{\text{LE}}$	$\ell_{\text{SUB}}(\boldsymbol{L}^{(1)}, \cdots; \tilde{\boldsymbol{D}}^{(1)}, \cdots)$

- $\varphi(\cdot)$ is guided by subtasks to learn a powerful representation, i.e., $\boldsymbol{R} = \varphi(\boldsymbol{X})$.
- $\psi(\cdot)$ is responsible for predicting subtask label distributions, i.e., $(\tilde{\boldsymbol{D}}^{(1)}, \cdots) = \psi(\boldsymbol{R})$. To ensure the precise prediction of subtask label distributions for reconstruction, we employ the mean absolute error function for subtask learning. The loss is weighted by the summation of the description degrees corresponding to the primary tasks, allowing more reliable label spaces to receive more attention. The subtask learning loss has the following form:

$$\ell_{\text{SUB}}\left(\mathcal{D}^{\circ}; \tilde{\mathcal{D}}^{\circ}\right) = \frac{1}{N\left|\mathcal{Y}^{\circ}\right|} \sum_{\mathcal{Y}^{(t)} \in \mathcal{Y}^{\circ}} \sum_{i=1}^{N} \left(\sum_{y_k \in \mathcal{Y}^{(t)}} d_{\boldsymbol{x}_i}^{y_k} \right) \sum_{j=1}^{\left|\mathcal{Y}^{(t)}\right|} \left| d_{\boldsymbol{x}_i}^{(t)y_j} - \tilde{d}_{\boldsymbol{x}_i}^{(t)y_j} \right|. \tag{7}$$

- $\omega(\cdot)$ can be any existing method that can be expressed as a network structure theoretically. Since the concatenation of the representation and subtask label distributions, we have $\boldsymbol{Z} = (\boldsymbol{R}, \psi(\boldsymbol{R}))$ and $\tilde{\boldsymbol{D}} = \omega(\boldsymbol{Z})$. In the case of the primary task being vanilla LDL, the primary task loss ℓ_{PRI} can be

$$\ell_{\text{KL}}\left(\boldsymbol{D}, \tilde{\boldsymbol{D}}\right) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{L} d_{\boldsymbol{x}_i}^{y_j} \ln \frac{d_{\boldsymbol{x}_i}^{y_j}}{\tilde{d}_{\boldsymbol{x}_i}^{y_j}}, \quad \ell_{\text{KL}} \in \mathcal{L}_{\text{LDL}}. \tag{8}$$

Note that ℓ_{PRI} changes as the primary task changes. Finally, we can learn the model parameters $\boldsymbol{\Theta}$ by

$$\boldsymbol{\Theta}^{*} = \arg\min_{\boldsymbol{\Theta}} (\ell_{\text{PRI}} + \alpha \ell_{\text{SUB}}), \tag{9}$$

where α is a trade-off parameter. Compared with the shallow regime, $\mathcal{S}$-LDL of the deep regime has the following advantages: 1) There is no two-stage training gap, which makes the representation contain insights from both the primary task and the subtasks; 2) the framework not only serves LDL, but can also be directly applied to derivative tasks of LDL, e.g., LDL for classification (LDL4C) (Wang, Jing and Geng, Xin, 2019), incomplete LDL (IncomLDL) (Xu and Zhou, 2017), label enhancement (LE) (Xu et al., 2019). The modifications involved are shown in Table 2, where $\mathcal{L}_{\text{X}}$ indicates the set of losses for adaptable methods in the task of type "X". Special mathematical procedures of LDL4C and IncomLDL are defined as

$$\left[\bar{\boldsymbol{D}}^{(t)}\right]_{ij} \triangleq \begin{cases} 1, & \text{if } y_j = \arg\max_{\bar{y} \in \mathcal{Y}^{(t)}} d_{\boldsymbol{x}_i}^{\bar{y}} \\ 0, & \text{otherwise} \end{cases}, \quad \left[\mathcal{R}_{\Omega}(\boldsymbol{D})\right]_{ij} \triangleq \begin{cases} [\boldsymbol{D}]_{ij}, & \text{if } (i, y_j) \in \Omega \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

respectively, where $[\cdot]_{ij}$ represents the element in i-th row of the matrix corresponding to label y_j, and Ω represents observed elements sampled uniformly at random from $\boldsymbol{D}$ in IncomLDL. Such modifications are rational since: 1) targets of LDL4C and IncomLDL, i.e., $\bar{\boldsymbol{D}}$ and $\mathcal{R}_{\Omega}(\boldsymbol{D})$, are essentially different forms of degradation of the label distribution matrix; and 2) the target of LE is a logical label matrix $\boldsymbol{L}$, the same as the target of MLL, which is actually a special case of LDL.

6 Experiments

In this section, we evaluate $\mathcal{S}$-LDL of the deep regime. Due to page limitations, datasets, comparison methods, and their parameter settings are introduced in the appendix.

Metrics For LDL, we use the same metrics suggested by Jia et al. (2023). Due to page limitations, we only present results on `Cheby.` ↓ (Chebyshev distance), `Clark` ↓ (Clark distance), `Cosine` ↑ (cosine similarity), and `Spear.` ↑ (Spearman's coefficient) in the main paper, where ↓ (↑) indicates "the lower (higher) the better". Note that these metrics are *not* as intuitive as accuracy or error rate, i.e., *small changes can mean large performance differences.* For LDL4C, objective of which is different from LDL, we use `0/1 loss` ↓ (zero one loss) and `Err. prob.` ↓ (error probability) as metrics (Wang, Jing and Geng, Xin, 2019).

Table 3: Experimental results of LDL on JAFFE and Yeast_diau formatted as (mean ± std(rank))

Algorithms	JAFFE (Lyons et al., 1998)		Algorithms	Yeast_diau (Geng, 2016)	
	Clark ↓	Cosine ↑		Cheby. ↓	Spear. ↑
LDSVR (Geng and Hou, 2015)	.3280 ±.027 (6)	.9549 ±.010 (7)	CPNN (Geng et al., 2013)	.0385 ±.001 (9)	.2962 ±.034 (10)
AA-kNN (Geng, 2016)	.3483 ±.032 (8)	.9497 ±.010 (9)	AA-kNN	.0385 ±.001 (9)	.3674 ±.029 (9)
LDLFs (Shen et al., 2017)	.3637 ±.032 (10)	.9494 ±.009 (10)	LDLFs	.0371 ±.001 (8)	.4088 ±.021 (8)
DF-BFGS (González et al., 2021a)	.3062 ±.025 (3)	.9633 ±.007 (2)	DF-BFGS	.0368 ±.001 (5)	.4161 ±.027 (5)
KLD (Geng, 2016) •	.3608 ±.031 (9)	.9538 ±.008 (8)	LRR •	.0370 ±.001 (7)	.4154 ±.023 (6)
$\mathcal{S}$-KLD	.3007 ±.032 (2)	.9625 ±.009 (3)	$\mathcal{S}$-LRR	**.0366** ±.001 (1)	.4198 ±.023 (2)
SCL (Jia et al., 2019) •	.3358 ±.024 (7)	.9592 ±.006 (6)	QFD^2 (Wen et al., 2023) •	.0369 ±.001 (6)	.4118 ±.025 (7)
$\mathcal{S}$-SCL	.3184 ±.025 (4)	.9604 ±.008 (5)	$\mathcal{S}$-QFD^2	**.0366** ±.001 (1)	**.4203** ±.021 (1)
LRR (Jia et al., 2023) •	.3230 ±.027 (5)	.9616 ±.008 (4)	CJS (Wen et al., 2023)	.0367 ±.001 (4)	.4164 ±.025 (4)
$\mathcal{S}$-LRR	**.2934** ±.028 (1)	**.9635** ±.008 (1)	$\mathcal{S}$-CJS	**.0366** ±.001 (1)	.4198 ±.024 (2)

Table 4: Experimental results of LDL4C on sBU_3DFE and Flickr formatted as (mean±std(rank))

Algorithms	sBU_3DFE (Geng, 2016)		Algorithms	Flickr (Yang et al., 2017b)	
	0/1 loss ↓	Err. prob. ↓		0/1 loss ↓	Err. prob. ↓
LDL4C (Wang, Jing and Geng, Xin, 2019)	.5578 ±.028 (6)	.7671 ±.007 (5)	LDL4C	.8971 ±.008 (6)	.8884 ±.004 (6)
$\mathcal{S}$-LDL4C	.5526 ±.025 (5)	.7686 ±.006 (6)	$\mathcal{S}$-LDL4C	.8705 ±.138 (5)	.8702 ±.100 (5)
HR (Wang and Geng, 2021a) •	.5167 ±.027 (3)	.7596 ±.006 (2)	HR •	.4513 ±.015 (4)	.5823 ±.007 (4)
$\mathcal{S}$-HR	.5069 ±.025 (2)	.7598 ±.006 (3)	$\mathcal{S}$-HR	**.4219** ±.015 (1)	**.5639** ±.007 (1)
LDLM (Wang and Geng, 2021b) •	.5258 ±.034 (4)	.7619 ±.009 (4)	LDLM •	.4384 ±.014 (3)	.5740 ±.007 (3)
$\mathcal{S}$-LDLM	**.4809** ±.024 (1)	**.7524** ±.005 (1)	$\mathcal{S}$-LDLM	.4321 ±.016 (2)	.5667 ±.007 (2)

Results and discussion We apply $\mathcal{S}$-LDL to existing methods to demonstrate performance improvements. For each dataset we conduct ten-fold experiments repeated 10 times, and the average performance is recorded. Tables 3 to 4 show representative results and the remainder are in the appendix, where • (○) indicates that more than half of the metrics support that "$\mathcal{S}$-X" is statistically superior (inferior) to the corresponding methods "X" (pairwise t-test at 0.05 significance level); there is no significant if neither • nor ○ is shown. LRR focuses on the label ranking relationship, which is also emphasized by each subtask. We believe this is why $\mathcal{S}$-LDL and LRR fit so well. Note that our method has the least improvement in SCL, which may be attributed to its reliance on shallow regime methods in the prediction phase. It is also worth noting that the improvement in vanilla KLD is considerable, which just illustrates the limitations of loss function engineering that considers label correlation one-sidedly. QFD^2 and CJS are tailored for ordinary LDL, and may have better results than LRR on this regard. Powered by $\mathcal{S}$-LDL, these methods can all achieve better level. For LDL4C, $\mathcal{S}$-LDL significantly improves both HR and LDLM. However, it can be observed that $\mathcal{S}$-LDL4C is unstable on the Flickr dataset, which is not surprising since LDL4C itself fails on it. We believe this is caused by the combined effect of the sparsity of the dataset and the information entropy operation involved in LDL4C.

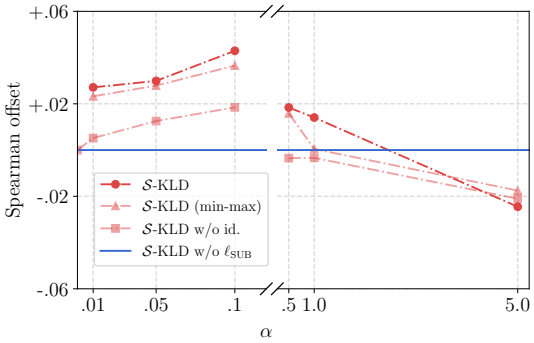


Figure 4: Visualized results of the ablation study and the parameter sensitivity analysis, which is on the Natural_Scene (Geng, 2016) dataset.

Parameter sensitivity We check the sensitivity of the trade-off parameter α on the LDL task with the Natural_Scene dataset by varying the parameter in $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$. Results are shown in Fig. 4. Spearman's coefficient of $\mathcal{S}$-LDL first increases and then decreases as α varies, demonstrating a desirable bell-shaped curve. This justifies our motivation of jointly learning the primary task and subtasks, as a good trade-off between them can enhance the performance.

Ablation study Here we are interested in the importance of each part of $\mathcal{S}$-LDL, thus an ablation study is performed with $\mathcal{S}$-KLD: 1) we replace $\mathcal{N}_{\text{SUM}}$ in SC with the min-max function to examine the importance of the subtask distribution reconstruction, and this model is denoted as $\mathcal{S}$-KLD (min-max); 2) we remove the identity mapping in Fig. 3 to examine the importance of the prediction via subtask representation, and this model is denoted as $\mathcal{S}$-KLD w/o id.; 3) we train without the term of ℓ_{SUB} (i.e.,

setting $\alpha = 0$) to examine the importance of subtask learning, and this model is denoted as $\mathcal{S}$-KLD w/o ℓ_{SUB}. Results are also shown in Fig. 4, which confirms that each part of $\mathcal{S}$-LDL contributes as long as there is a good trade-off.

7 LIMITATIONS AND CONCLUSION

Limitations First, $\mathcal{S}$-LDL of the shallow regime is proposed out of intuition, and in Section 5, we have discussed its limitations, which are addressed via the designing of $\mathcal{S}$-LDL of the deep regime. Second, when the label space is large, especially when labels are continuous and result in unimodal label distributions (e.g., age estimation), our proposed cannot be rationally applied. Fortunately, one possible workaround is to use a binning tricks for preprocessing, and then construct subtasks.

Conclusion We propose $\mathcal{S}$-LDL, a subtask learning framework nested into LDL. $\mathcal{S}$-LDL is generic: it generates pseudo-supervised information via subtask construction without any extra knowledge; $\mathcal{S}$-LDL is minimalist: it can be attached to existing methods and handle derivative tasks; $\mathcal{S}$-LDL is efficient: it captures a wide variety of label correlations. The analysis shows the validity and reconstructability of subtasks, and experiments show the superiority of our framework.

REFERENCES

Shikai Chen, Jianfeng Wang, Yuedong Chen, Zhongchao Shi, Xin Geng, and Yong Rui. Label distribution learning on auxiliary label space graphs for facial expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13984–13993, 2020.

Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017.

Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28 (7):1734–1748, 2016.

Xin Geng and Peng Hou. Pre-release prediction of crowd opinion on movies by label distribution learning. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 3511–3517, 2015.

Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.

Manuel González, Germán González-Almagro, Isaac Triguero, José-Ramón Cano, and Salvador García. Decomposition-fusion for label distribution learning. *Information Fusion*, 66:64–75, 2021a.

Manuel González, Julián Luengo, José-Ramón Cano, and Salvador García. Synthetic sample generation for label distribution learning. *Information Sciences*, 544:197–213, 2021b.

Xiuyi Jia, Zechao Li, Xiang Zheng, Weiwei Li, and Sheng-Jun Huang. Label distribution learning with label correlations on local samples. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1619–1631, 2019.

Xiuyi Jia, Xiaoxia Shen, Weiwei Li, Yunan Lu, and Jihua Zhu. Label distribution learning by maintaining label ranking relation. *IEEE Transactions on Knowledge and Data Engineering*, 35 (02):1695–1707, 2023.

Yufei Jin, Richard Gao, Yi He, and Xingquan Zhu. Gldl: Graph label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12965–12974, 2024.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Nhat Le, Khanh Nguyen, Quang Tran, Erman Tjiputra, Bac Le, and Anh Nguyen. Uncertainty-aware label distribution learning for facial expression recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6088–6097, 2023.

Xiangyu Li, Xinjie Liang, Gongning Luo, Wei Wang, Kuanquan Wang, and Shuo Li. Ambiguity-aware breast tumor cellularity estimation via self-ensemble label distribution learning. *Medical Image Analysis*, 90:102944, 2023a.

Xiangyu Li, Gongning Luo, Wei Wang, Kuanquan Wang, and Shuo Li. Curriculum label distribution learning for imbalanced medical image segmentation. *Medical Image Analysis*, 89:102911, 2023b.

Lingyu Liang, Luojun Lin, Lianwen Jin, Duorui Xie, and Mengru Li. Scut-fbp5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In *Proceedings of the 24th International Conference on Pattern Recognition*, pages 1598–1603, 2018.

Yunan Lu and Xiuyi Jia. Predicting label distribution from multi-label ranking. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, pages 36931–36943, 2022.

Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. In *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, 1998.

Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, pages 993–1002, 2018.

Jesse Read, Concha Bielza, and Pedro Larrañaga. Multi-dimensional classification with super-classes. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1720–1733, 2013.

Jesse Read, Antti Puurula, and Albert Bifet. Multi-label classification with meta-labels. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, pages 941–946, 2014.

Tingting Ren, Xiuyi Jia, Weiwei Li, Lei Chen, and Zechao Li. Label distribution learning with label-specific features. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3318–3324, 2019.

Wei Shen, Kai Zhao, Yilu Guo, and Alan Yuille. Label distribution learning forests. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 834–843, 2017.

Amirreza Shirani, Franck Dernoncourt, Paul Asente, Nedim Lipka, Seokhwan Kim, Jose Echevarria, and Thamar Solorio. Learning emphasis selection for written text in visual media from crowd-sourced label distributions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1167–1172, 2019.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, volume 21, pages 53–59, 2008.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, 2010.

Jing Wang and Xin Geng. Theoretical analysis of label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5256–5263, 2019.

Jing Wang and Xin Geng. Learn the highest label and rest label description degrees. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 3097–3103, 2021a.

Jing Wang and Xin Geng. Label distribution learning machine. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10749–10759. PMLR, 2021b.

Ke Wang, Ning Xu, Miaogen Ling, and Xin Geng. Fast label enhancement for label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(02):1502–1514, 2023.

Ran Wang, Sam Kwong, Xu Wang, and Yuheng Jia. Active k-labelsets ensemble for multi-label classification. *Pattern Recognition*, 109:107583, 2021.

Wang, Jing and Geng, Xin. Classification with label distribution learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3712–3718, 2019.

Changsong Wen, Xin Zhang, Xingxu Yao, and Jufeng Yang. Ordinal label distribution learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23481–23491, 2023.

Xiaoping Wu, Ni Wen, Jie Liang, Yukun Lai, Dongyu She, Mingming Cheng, and Jufeng Yang. Joint acne image grading and counting via label distribution learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10642–10651, 2019.

Miao Xu and Zhi-Hua Zhou. Incomplete label distribution learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3175–3181, 2017.

Ning Xu, Yunpeng Liu, and Xin Geng. Label enhancement for label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1632–1643, 2019.

Ning Xu, Jun Shu, Renyi Zheng, Xin Geng, Deyu Meng, and Min-Ling Zhang. Variational label enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6537–6551, 2023.

Jufeng Yang, Dongyu She, and Ming Sun. Joint image emotion classification and distribution learning via deep convolutional neural network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3266–3272, 2017a.

Jufeng Yang, Ming Sun, and Xiaoxiao Sun. Learning visual sentiment distributions via augmented conditional probability neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 224–230, 2017b.

Yansheng Zhai, Jianhua Dai, and Hong Shi. Label distribution learning based on ensemble neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing*, pages 593–602, 2018.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.

Xingyu Zhao, Yuexuan An, Ning Xu, Jing Wang, and Xin Geng. Imbalanced label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11336–11344, 2023a.

Xingyu Zhao, Lei Qi, Yuexuan An, and Xin Geng. Generalizable label distribution learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8932–8941, 2023b.

Qinghai Zheng, Jihua Zhu, and Haoyu Tang. Label information bottleneck for label enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7497–7506, 2023.

A Appendix: Details of Experiments

Here we try our best to provide as much information as possible for reproducible research.

A.1 Metrics

For LDL, IncomLDL, and LE, we use the same metrics suggested by Geng (2016), which are `Cheby.` $\downarrow$ (Chebyshev distance), `Clark` $\downarrow$ (Clark distance), `Can.` $\downarrow$ (Canberra distance), `KLD` $\downarrow$ (Kullback-Leibler divergence), `Cosine` $\uparrow$ (cosine similarity), and `Int.` $\uparrow$ (intersection similarity), respectively. Here $\downarrow$ ($\uparrow$) indicates "the lower (higher) the better". For LDL and IncomLDL, we additionally use two ranking metrics: `Spear.` $\uparrow$ (Spearman's coefficient) and `Ken.` $\uparrow$ (Kendall's coefficient) (Jia et al., 2023). Note that these metrics are *not* as intuitive as accuracy or error rate, i.e., *small changes can mean large performance differences.* For LDL4C, objective of which is different from LDL, we use `0/1 loss` $\downarrow$ (zero one loss) and `Err. prob.` $\downarrow$ (error probability) as metrics (Wang, Jing and Geng, Xin, 2019). Let the real distribution be denoted by $\boldsymbol{u} = \{u_j\}_{j=1}^{L}$, and the predicted distribution be denoted by $\boldsymbol{v} = \{v_j\}_{j=1}^{L}$, then the above metrics can be summarized in Table 5, where $\rho(\cdot)$ and $\delta(\cdot, \cdot)$ are the ranking function and the Kronecker delta function, respectively.

Table 5: Summary of the metrics

Name	Formula	Name	Formula
`Cheby.` $\downarrow$	$\mathrm{Dis}_1(\boldsymbol{u}, \boldsymbol{v}) = \max_j \lvert u_j - v_j \rvert$	`Cosine` $\uparrow$	$\mathrm{Sim}_1(\boldsymbol{u}, \boldsymbol{v}) = \frac{\sum_{j=1}^{L} u_j v_j}{\sqrt{\sum_{j=1}^{L} u_j^2}\sqrt{\sum_{j=1}^{L} v_j^2}}$
`Clark` $\downarrow$	$\mathrm{Dis}_2(\boldsymbol{u}, \boldsymbol{v}) = \sqrt{\sum_{j=1}^{L} \frac{(u_j - v_j)^2}{(u_j + v_j)^2}}$	`Int.` $\uparrow$	$\mathrm{Sim}_2(\boldsymbol{u}, \boldsymbol{v}) = \sum_{j=1}^{L} \min(u_j, v_j)$
`Can.` $\downarrow$	$\mathrm{Dis}_3(\boldsymbol{u}, \boldsymbol{v}) = \sum_{j=1}^{L} \frac{\lvert u_j - v_j \rvert}{u_j + v_j}$	`Spear.` $\uparrow$	$\mathrm{Rnk}_1(\boldsymbol{u}, \boldsymbol{v}) = 1 - \frac{6\sum_{j=1}^{L}(\rho(u_j) - \rho(v_j))^2}{L(L^2-1)}$
`KLD` $\downarrow$	$\mathrm{Dis}_4(\boldsymbol{u}, \boldsymbol{v}) = \sum_{j=1}^{L} u_j \ln \frac{u_j}{v_j}$	`Ken.` $\uparrow$	$\mathrm{Rnk}_2(\boldsymbol{u}, \boldsymbol{v}) = \frac{2\sum_{j<k} \mathrm{sgn}(u_j - u_k)\mathrm{sgn}(v_j - v_k)}{L(L-1)}$
`0/1 loss` $\downarrow$	$\mathrm{C}_1(\boldsymbol{u}, \boldsymbol{v}) = \delta(\arg\max(\boldsymbol{u}), \arg\max(\boldsymbol{v}))$	`Err. prob.` $\downarrow$	$\mathrm{C}_2(\boldsymbol{u}, \boldsymbol{v}) = 1 - u_{\arg\max(\boldsymbol{v})}$

A.2 Datasets

We adopt several widely used label distribution datasets, including: `JAFFE` (Lyons et al., 1998);[4] `fbp5500` (Liang et al., 2018);[5] `sBU_3DFE`, `Movie`, `Natural_Scene`, `Yeast_heat`, `Yeast_diau`, `Yeast_cold`, and `Yeast_dtt` provided by Geng (2016);[6] `emotion6`, `Twitter`, and `Flickr` provided by Yang et al. (2017b).[7] The information of these datasets are summarized in Table 6.

A.3 Comparison methods

Table 6: Summary of datasets

Dataset	# Instances N	# Features P	# Labels L
`JAFFE`	213	243	6
`sBU_3DFE`	2500	243	6
`Movie`	7755	1869	5
`Nature_Scene`	2000	294	9
`fbp5500`	5500	512	5
`Yeast_heat`	2465	24	6
`Yeast_diau`	2465	24	7
`Yeast_cold`	2465	24	4
`Yeast_dtt`	2465	24	4
`emotion6`	1980	168	7
`Twitter`	10045	168	8
`Flickr`	11150	168	8

On the one hand, we apply our proposed $\mathcal{S}$-LDL to existing methods to demonstrate performance improvements in the LDL task (denoted by the "$\mathcal{S}$-" prefix). These methods are BFGS-LLD (KLD) (Geng, 2016), SCL (Jia et al., 2019), LRR (Jia et al., 2023), QFD^2 (Wen et al., 2023), and CJS (Wen et al., 2023) (the losses of these methods constitute the set $\mathcal{L}_{\mathrm{LDL}}$). On the other hand, we compare $\mathcal{S}$-LDL with methods that have specialized structure, which our proposed cannot directly adapt to. These methods are CPNN (Geng et al., 2013), LDSVR (Geng and Hou, 2015), AA-kNN (Geng, 2016), LDLFs (Shen et al., 2017), and DF-LDL (denoted by DF-BFGS since we use BFGS-LLDs as base estimators) (González et al., 2021a). Moreover, we apply our proposed to derivative tasks of LDL (i.e., LDL4C, IncomLDL, and LE) and the comparison methods involved are LDL4C (Wang, Jing and Geng, Xin, 2019), HR (Wang and Geng, 2021a), LDLM (Wang and Geng, 2021b), IncomLDL (Xu and Zhou, 2017), LP (Xu et al., 2019), GLLE (Xu et al., 2019), LEVI (Xu et al., 2023), and LIBLE (Zheng et al., 2023).

A.4 Parameter settings and experimental environment

The parameter settings of the proposed $\mathcal{S}$-LDL and comparison algorithms are summarized in Table 7. Note that DF-LDL is parameter-free, and we use BFGS-LLDs as its base estimators, parameter settings of which are the same as BFGS-LLD as the comparison algorithm. We use Adam (Kingma and Ba, 2015) for the optimization of $\mathcal{S}$-LDL. For all methods of the deep regime, the learning rate is chosen among $\{1, 2, 5\} \times 10^{\{-4, -3, -2\}}$, and the selection of the number of epochs is nested into

[4] `https://zenodo.org/records/3451524`

[5] `https://github.com/HCIILAB/SCUT-FBP5500-Database-Release`

[6] `https://palm.seu.edu.cn/xgeng/LDL/download.htm`

[7] `https://cv.nankai.edu.cn/projects`

Table 7: Summary of algorithms and parameter settings

Algorithms	Parameter	Value (Range)
AA-kNN	k: # Neighbors	5
LDLFs	# Estimators (trees)	5
	Depth	6
	Latent units (leaves)	64
BFGS-LLD	ε: Convergence criterion	10^{-6}
	Max iteration	600
SCL	m: # Clusters	5
	λ_1, λ_2, λ_3: Trade-off	10^{-3}, 10^{-3}, 0.1
LRR	λ: Trade-off (ranking loss)	$10^{\{-5,-4,-3,-2,-1\}}$
	β: Trade-off (regularization)	$10^{\{-3,-2,-1,0,1,2\}}$
LDL4C	C_1, C_2: Balance coefficients	10^{-2}, 10^{-6}
	ρ: Margin	10^{-2}
HR	λ_1, λ_2, λ_3: Trade-off	10^{-2}, 10^{-6}
	ρ: Margin	10^{-2}
LDLM	λ_1, λ_2, λ_3: Trade-off	10^{-6}, $10^{\{-3,-2,-1\}}$, $10^{\{-3,-2,-1\}}$
	ρ: Margin	10^{-2}
IncomLDL	ε: Convergence criterion	10^{-6}
	γ: Factor of Lipschitz constant	2
	λ: Trade-off	1
LP	α: Balance coefficient	0.5
GLLE	λ_1, λ_2: Trade-off	10^{-2}, 10^{-4}
	σ: Width parameter for similarity calculation	10
LEVI	λ: Trade-off	1
LIBLE	α, β: Trade-off	$10^{\{-3,-2,-1,0,1,2\}}$
$\mathcal{S}$-LDL	α, λ, T	0.1, 0.2, 10

a ten-fold cross validation. All the results are obtained on a Linux workstation with Intel Core i9 (3.70GHz), NVIDIA GeForce RTX 3090 (24GB), and 32GB memory.

A.5 Full experimental results

Here we provide complete results of all conducted experiments. Tables 8 to 19 are results on the LDL task with different datasets. For IncomLDL, we follow *the incomplete settings* (Xu and Zhou, 2017) and vary the observed rate $\omega\%$ from 20% to 40%. Tables 20 to 21 are results on the IncomLDL task. Tables 22 to 23 are on the LDL4C task. For LE, we follow *the settings of the recovery experiment* (Xu et al., 2019). Tables 24 to 25 show results on the LE task.

Table 8: Experimental results of LDL on the `JAFFE` dataset formatted as ($\texttt{mean} \pm \texttt{std}$)

Algorithms	`Cheby.` ↓	`Clark` ↓	`Can.` ↓	`KLD` ↓	`Cosine` ↑	`Int.` ↑	`Spear.` ↑	`Ken.` ↑
LDSVR	.0959 $\pm$.013	.3280 $\pm$.027	.6778 $\pm$.058	.0476 $\pm$.011	.9549 $\pm$.010	.8838 $\pm$.012	.5175 $\pm$.102	.4508 $\pm$.086
AA-kNN	.0978 $\pm$.012	.3483 $\pm$.032	.7164 $\pm$.066	.0527 $\pm$.011	.9497 $\pm$.010	.8766 $\pm$.012	.4111 $\pm$.083	.3514 $\pm$.070
LDLFs	.0940 $\pm$.010	.3637 $\pm$.032	.7355 $\pm$.066	.0550 $\pm$.009	.9494 $\pm$.009	.8766 $\pm$.011	.4364 $\pm$.108	.3749 $\pm$.093
DF-BFGS	.0827 $\pm$.009	.3062 $\pm$.025	.6239 $\pm$.052	.0388 $\pm$.007	.9633 $\pm$.007	.8944 $\pm$.010	.5244 $\pm$.087	.4493 $\pm$.077
KLD •	.0925 $\pm$.010	.3608 $\pm$.031	.7363 $\pm$.064	.0508 $\pm$.009	.9538 $\pm$.008	.8777 $\pm$.011	.4572 $\pm$.097	.3873 $\pm$.084
$\mathcal{S}$-KLD	.0818 $\pm$.011	.3007 $\pm$.032	.6132 $\pm$.067	.0395 $\pm$.010	.9625 $\pm$.009	.8960 $\pm$.012	**.5461** $\pm$.105	.4769 $\pm$.096
SCL •	.0873 $\pm$.008	.3358 $\pm$.024	.6874 $\pm$.051	.0439 $\pm$.006	.9592 $\pm$.006	.8851 $\pm$.009	.4744 $\pm$.092	.4020 $\pm$.080
$\mathcal{S}$-SCL	.0854 $\pm$.010	.3184 $\pm$.025	.6526 $\pm$.053	.0420 $\pm$.008	.9604 $\pm$.008	.8896 $\pm$.010	.5110 $\pm$.095	.4388 $\pm$.084
LRR •	.0853 $\pm$.010	.3230 $\pm$.027	.6560 $\pm$.055	.0412 $\pm$.008	.9616 $\pm$.008	.8906 $\pm$.010	.5117 $\pm$.094	.4420 $\pm$.084
$\mathcal{S}$-LRR	**.0804** $\pm$.009	**.2934** $\pm$.028	**.5989** $\pm$.059	**.0383** $\pm$.009	**.9635** $\pm$.008	**.8981** $\pm$.011	.5448 $\pm$.092	**.4819** $\pm$.084

Table 9: Experimental results of LDL on the sBU_3DFE dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
LDSVR	.1250 $\pm_{.005}$	.3710 $\pm_{.010}$	.8009 $\pm_{.021}$	.0720 $\pm_{.004}$	.9298 $\pm_{.004}$	.8559 $\pm_{.004}$	.3524 $\pm_{.031}$	.3011 $\pm_{.026}$
AA-kNN	.1272 $\pm_{.004}$	.4001 $\pm_{.009}$	.8281 $\pm_{.020}$	.0801 $\pm_{.004}$	.9217 $\pm_{.004}$	.8488 $\pm_{.004}$	.2053 $\pm_{.030}$	.1767 $\pm_{.026}$
LDLFs	.1016 $\pm_{.003}$	.3262 $\pm_{.008}$	.6841 $\pm_{.017}$	.0504 $\pm_{.003}$	.9499 $\pm_{.003}$	.8776 $\pm_{.003}$	.4212 $\pm_{.023}$	.3620 $\pm_{.019}$
DF-BFGS	.1146 $\pm_{.004}$	.3616 $\pm_{.008}$	.7627 $\pm_{.019}$	.0618 $\pm_{.003}$	.9388 $\pm_{.003}$	.8626 $\pm_{.004}$	.3026 $\pm_{.031}$	.2621 $\pm_{.026}$
KLD •	.1147 $\pm_{.004}$	.3697 $\pm_{.008}$	.7804 $\pm_{.019}$	.0624 $\pm_{.003}$	.9387 $\pm_{.003}$	.8604 $\pm_{.003}$	.3021 $\pm_{.026}$	.2643 $\pm_{.022}$
$\mathcal{S}$-KLD	.1014 $\pm_{.004}$	.3203 $\pm_{.009}$	.6736 $\pm_{.018}$	.0514 $\pm_{.003}$	.9487 $\pm_{.003}$	.8789 $\pm_{.004}$	.4334 $\pm_{.025}$	.3729 $\pm_{.022}$
SCL •	.1145 $\pm_{.004}$	.3648 $\pm_{.008}$	.7748 $\pm_{.018}$	.0605 $\pm_{.003}$	.9404 $\pm_{.003}$	.8614 $\pm_{.003}$	.3091 $\pm_{.026}$	.2701 $\pm_{.021}$
$\mathcal{S}$-SCL	.1041 $\pm_{.004}$	.3301 $\pm_{.009}$	.6936 $\pm_{.019}$	.0535 $\pm_{.003}$	.9468 $\pm_{.003}$	.8754 $\pm_{.004}$	.3956 $\pm_{.030}$	.3381 $\pm_{.027}$
LRR •	.1067 $\pm_{.003}$	.3476 $\pm_{.008}$	.7320 $\pm_{.017}$	.0543 $\pm_{.003}$	.9465 $\pm_{.003}$	.8695 $\pm_{.003}$	.3626 $\pm_{.026}$	.3123 $\pm_{.022}$
$\mathcal{S}$-LRR	**.0996** $\pm_{.004}$	**.3157** $\pm_{.008}$	**.6610** $\pm_{.017}$	**.0499** $\pm_{.003}$	**.9502** $\pm_{.003}$	**.8812** $\pm_{.003}$	**.4455** $\pm_{.026}$	**.3837** $\pm_{.023}$

Table 10: Experimental results of LDL on the Yeast_heat dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
CPNN	.0419 $\pm_{.001}$	.1818 $\pm_{.005}$	.3633 $\pm_{.009}$	.0125 $\pm_{.001}$	.9881 $\pm_{.001}$	.9404 $\pm_{.001}$	.1507 $\pm_{.034}$	.1221 $\pm_{.028}$
AA-kNN	.0441 $\pm_{.001}$	.1913 $\pm_{.005}$	.3840 $\pm_{.010}$	.0140 $\pm_{.001}$	.9867 $\pm_{.001}$	.9370 $\pm_{.002}$	.1678 $\pm_{.031}$	.1384 $\pm_{.026}$
LDLFs	.0420 $\pm_{.001}$	.1818 $\pm_{.005}$	.3627 $\pm_{.009}$	.0125 $\pm_{.001}$	.9881 $\pm_{.001}$	.9405 $\pm_{.001}$	.1731 $\pm_{.032}$	.1409 $\pm_{.026}$
DF-BFGS	.0420 $\pm_{.001}$	.1816 $\pm_{.005}$	.3624 $\pm_{.009}$	.0125 $\pm_{.001}$	.9881 $\pm_{.001}$	.9405 $\pm_{.001}$	**.1964** $\pm_{.034}$	**.1624** $\pm_{.028}$
LRR •	.0423 $\pm_{.001}$	.1828 $\pm_{.005}$	.3644 $\pm_{.009}$	.0126 $\pm_{.001}$	.9880 $\pm_{.001}$	.9402 $\pm_{.001}$	.1655 $\pm_{.033}$	.1351 $\pm_{.028}$
$\mathcal{S}$-LRR	**.0417** $\pm_{.001}$	.1806 $\pm_{.005}$	.3609 $\pm_{.009}$	**.0124** $\pm_{.001}$	**.9882** $\pm_{.001}$	.9408 $\pm_{.001}$	.1882 $\pm_{.034}$	.1548 $\pm_{.028}$
QFD^2 •	.0423 $\pm_{.001}$	.1827 $\pm_{.005}$	.3644 $\pm_{.009}$	.0126 $\pm_{.001}$	.9880 $\pm_{.001}$	.9402 $\pm_{.001}$	.1677 $\pm_{.032}$	.1351 $\pm_{.027}$
$\mathcal{S}$-QFD^2	**.0417** $\pm_{.001}$	.1808 $\pm_{.005}$	.3611 $\pm_{.009}$	**.0124** $\pm_{.001}$	**.9882** $\pm_{.001}$	.9408 $\pm_{.001}$	.1880 $\pm_{.032}$	.1544 $\pm_{.027}$
CJS •	.0423 $\pm_{.001}$	.1827 $\pm_{.005}$	.3643 $\pm_{.009}$	.0126 $\pm_{.001}$	.9880 $\pm_{.001}$	.9402 $\pm_{.001}$	.1632 $\pm_{.032}$	.1329 $\pm_{.027}$
$\mathcal{S}$-CJS	**.0417** $\pm_{.001}$	**.1804** $\pm_{.005}$	**.3603** $\pm_{.009}$	**.0124** $\pm_{.001}$	**.9882** $\pm_{.001}$	**.9409** $\pm_{.001}$	.1940 $\pm_{.030}$	.1589 $\pm_{.025}$

Table 11: Experimental results of LDL on the Yeast_diau dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
CPNN	.0385 $\pm_{.001}$	.2069 $\pm_{.006}$	.4439 $\pm_{.012}$	.0138 $\pm_{.001}$	.9872 $\pm_{.001}$	.9383 $\pm_{.002}$	.2962 $\pm_{.034}$	.2427 $\pm_{.027}$
AA-kNN	.0385 $\pm_{.001}$	.2085 $\pm_{.006}$	.4487 $\pm_{.014}$	.0145 $\pm_{.001}$	.9867 $\pm_{.001}$	.9377 $\pm_{.002}$	.3674 $\pm_{.029}$	.2976 $\pm_{.024}$
LDLFs	.0371 $\pm_{.001}$	.2014 $\pm_{.006}$	.4324 $\pm_{.012}$	.0132 $\pm_{.001}$	.9879 $\pm_{.001}$	.9401 $\pm_{.002}$	.4088 $\pm_{.021}$	.3254 $\pm_{.018}$
DF-BFGS	.0368 $\pm_{.001}$	.1999 $\pm_{.006}$	.4294 $\pm_{.013}$	.0131 $\pm_{.001}$	.9879 $\pm_{.001}$	.9405 $\pm_{.002}$	.4161 $\pm_{.027}$	**.3404** $\pm_{.022}$
LRR •	.0370 $\pm_{.001}$	.2007 $\pm_{.006}$	.4307 $\pm_{.012}$	.0131 $\pm_{.001}$	.9879 $\pm_{.001}$	.9403 $\pm_{.002}$	.4154 $\pm_{.023}$	.3343 $\pm_{.020}$
$\mathcal{S}$-LRR	**.0366** $\pm_{.001}$	**.1983** $\pm_{.006}$	**.4257** $\pm_{.012}$	**.0129** $\pm_{.001}$	**.9881** $\pm_{.001}$	**.9410** $\pm_{.002}$	.4198 $\pm_{.023}$	.3389 $\pm_{.019}$
QFD^2 •	.0369 $\pm_{.001}$	.2000 $\pm_{.006}$	.4296 $\pm_{.012}$	.0131 $\pm_{.001}$	.9879 $\pm_{.001}$	.9404 $\pm_{.002}$	.4118 $\pm_{.025}$	.3326 $\pm_{.021}$
$\mathcal{S}$-QFD^2	**.0366** $\pm_{.001}$	.1985 $\pm_{.006}$	.4261 $\pm_{.012}$	**.0129** $\pm_{.001}$	**.9881** $\pm_{.001}$	.9409 $\pm_{.002}$	**.4203** $\pm_{.021}$	.3387 $\pm_{.018}$
CJS	.0367 $\pm_{.001}$	.1989 $\pm_{.006}$	.4272 $\pm_{.012}$	.0130 $\pm_{.001}$	.9880 $\pm_{.001}$	.9408 $\pm_{.002}$	.4164 $\pm_{.025}$	.3366 $\pm_{.021}$
$\mathcal{S}$-CJS	**.0366** $\pm_{.001}$	.1984 $\pm_{.006}$	.4260 $\pm_{.012}$	.0130 $\pm_{.001}$	**.9881** $\pm_{.001}$	.9409 $\pm_{.002}$	.4198 $\pm_{.024}$	.3392 $\pm_{.019}$

Table 12: Experimental results of LDL on the Yeast_cold dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
CPNN	**.0510** $\pm_{.002}$	.1392 $\pm_{.005}$	.2396 $\pm_{.008}$	**.0121** $\pm_{.001}$	**.9886** $\pm_{.001}$	**.9410** $\pm_{.002}$	**.2651** $\pm_{.036}$	**.2263** $\pm_{.032}$
AA-kNN	.0542 $\pm_{.002}$	.1476 $\pm_{.005}$	.2549 $\pm_{.008}$	.0135 $\pm_{.001}$	.9872 $\pm_{.001}$	.9371 $\pm_{.002}$	.2189 $\pm_{.035}$	.1866 $\pm_{.031}$
LDLFs	.0511 $\pm_{.002}$	.1396 $\pm_{.005}$	.2404 $\pm_{.009}$	.0122 $\pm_{.001}$	.9885 $\pm_{.001}$	.9408 $\pm_{.002}$	.2482 $\pm_{.038}$	.2112 $\pm_{.033}$
DF-BFGS	.0514 $\pm_{.002}$	.1404 $\pm_{.005}$	.2424 $\pm_{.008}$	.0123 $\pm_{.001}$	.9885 $\pm_{.001}$	.9403 $\pm_{.002}$	.2581 $\pm_{.036}$	.2190 $\pm_{.030}$
LRR	.0511 $\pm_{.002}$	.1395 $\pm_{.005}$	.2402 $\pm_{.009}$	.0122 $\pm_{.001}$	**.9886** $\pm_{.001}$	.9408 $\pm_{.002}$	.2490 $\pm_{.035}$	.2111 $\pm_{.030}$
$\mathcal{S}$-LRR	**.0510** $\pm_{.002}$	**.1391** $\pm_{.005}$	**.2395** $\pm_{.009}$	**.0121** $\pm_{.001}$	**.9886** $\pm_{.001}$	**.9410** $\pm_{.002}$	.2618 $\pm_{.037}$	.2238 $\pm_{.032}$
QFD^2	.0513 $\pm_{.002}$	.1401 $\pm_{.005}$	.2413 $\pm_{.009}$	.0123 $\pm_{.001}$	.9885 $\pm_{.001}$	.9405 $\pm_{.002}$	.2534 $\pm_{.037}$	.2158 $\pm_{.032}$
$\mathcal{S}$-QFD^2	**.0510** $\pm_{.002}$	**.1391** $\pm_{.005}$	.2396 $\pm_{.008}$	**.0121** $\pm_{.001}$	**.9886** $\pm_{.001}$	**.9410** $\pm_{.002}$	.2571 $\pm_{.039}$	.2197 $\pm_{.033}$
CJS	.0513 $\pm_{.002}$	.1401 $\pm_{.005}$	.2412 $\pm_{.008}$	.0123 $\pm_{.001}$	.9884 $\pm_{.001}$	.9406 $\pm_{.002}$	.2535 $\pm_{.038}$	.2152 $\pm_{.032}$
$\mathcal{S}$-CJS	**.0510** $\pm_{.002}$	.1392 $\pm_{.005}$	.2396 $\pm_{.009}$	**.0121** $\pm_{.001}$	**.9886** $\pm_{.001}$	**.9410** $\pm_{.002}$	.2621 $\pm_{.037}$	.2241 $\pm_{.031}$

Table 13: Experimental results of LDL on the Yeast_dtt dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
CPNN	.0361 ±.001	.0984 ±.004	.1690 ±.006	.0063 ±.001	.9941 ±.000	.9583 ±.001	.1735 ±.035	.1494 ±.030
AA-kNN	.0386 ±.001	.1047 ±.004	.1797 ±.006	.0071 ±.001	.9933 ±.000	.9556 ±.001	.1591 ±.033	.1399 ±.030
LDLFs	.0360 ±.001	.0981 ±.004	.1689 ±.006	.0063 ±.001	**.9941** ±.000	.9583 ±.001	.1986 ±.038	.1727 ±.034
DF-BFGS	.0365 ±.001	.0995 ±.004	.1712 ±.006	.0064 ±.001	.9939 ±.000	.9578 ±.001	.1804 ±.033	.1592 ±.030
LRR	.0360 ±.001	.0982 ±.004	.1690 ±.006	.0063 ±.001	**.9941** ±.000	.9583 ±.001	.2016 ±.037	.1738 ±.032
$\mathcal{S}$-LRR	**.0359** ±.001	**.0977** ±.004	**.1680** ±.006	**.0062** ±.001	**.9941** ±.000	**.9585** ±.001	.2068 ±.035	.1811 ±.031
QFD^2	.0362 ±.001	.0986 ±.004	.1696 ±.006	.0063 ±.001	.9940 ±.000	.9582 ±.001	.1917 ±.035	.1665 ±.031
$\mathcal{S}$-QFD^2	**.0359** ±.001	**.0977** ±.004	.1681 ±.006	**.0062** ±.001	**.9941** ±.000	**.9585** ±.001	**.2086** ±.036	**.1822** ±.032
CJS	.0361 ±.001	.0984 ±.004	.1692 ±.006	.0063 ±.001	**.9941** ±.000	.9582 ±.001	.1975 ±.040	.1722 ±.035
$\mathcal{S}$-CJS	**.0359** ±.001	.0978 ±.004	.1682 ±.006	**.0062** ±.001	**.9941** ±.000	**.9585** ±.001	.2080 ±.035	.1804 ±.031

Table 14: Experimental results of LDL on the emotion6 dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
LDSVR	.3152 ±.010	1.8217 ±.020	4.1452 ±.064	1.0744 ±.081	.6906 ±.015	.5773 ±.012	.3915 ±.030	.3235 ±.025
AA-kNN	.3288 ±.011	1.7116 ±.026	3.8757 ±.076	.9512 ±.115	.6632 ±.013	.5564 ±.010	.2920 ±.027	.2401 ±.022
LDLFs	.3120 ±.010	1.6625 ±.026	3.7330 ±.075	.5871 ±.024	.7143 ±.011	.5802 ±.010	.3631 ±.029	.3025 ±.024
DF-BFGS	.3026 ±.010	1.6765 ±.025	3.7675 ±.071	.5805 ±.026	.7206 ±.013	.5909 ±.010	.3940 ±.027	.3256 ±.022
KLD •	.3037 ±.010	1.6774 ±.025	3.7729 ±.074	.5863 ±.027	.7191 ±.013	.5897 ±.011	.3959 ±.028	.3259 ±.023
$\mathcal{S}$-KLD	.3024 ±.010	1.6548 ±.026	3.6984 ±.075	.5631 ±.024	.7282 ±.012	.5926 ±.010	.4063 ±.027	.3361 ±.023
SCL •	.3020 ±.010	1.6750 ±.025	3.7642 ±.073	.5803 ±.027	.7219 ±.013	.5917 ±.011	.4003 ±.028	.3299 ±.023
$\mathcal{S}$-SCL	**.3018** ±.010	1.6554 ±.026	3.6993 ±.076	.5631 ±.025	.7281 ±.012	**.5936** ±.010	**.4089** ±.027	**.3383** ±.023
LRR •	.3030 ±.010	1.6736 ±.025	3.7601 ±.073	.5804 ±.026	.7212 ±.013	.5899 ±.010	.3941 ±.027	.3243 ±.023
$\mathcal{S}$-LRR	.3028 ±.009	**1.6524** ±.026	**3.6923** ±.074	**.5607** ±.023	**.7299** ±.011	.5923 ±.010	.4078 ±.027	.3373 ±.022

Table 15: Experimental results of LDL on the Twitter dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
LDSVR	.4236 ±.008	2.6722 ±.002	7.3015 ±.009	5.0018 ±.115	.7627 ±.008	.5761 ±.008	.5237 ±.008	.4246 ±.007
AA-kNN	.3172 ±.004	**2.0142** ±.012	**4.5597** ±.043	3.1429 ±.148	.7926 ±.006	.6024 ±.005	.5014 ±.009	.4432 ±.008
LDLFs	.4035 ±.014	2.5461 ±.010	6.8269 ±.040	1.6884 ±.115	.6756 ±.018	.5318 ±.013	.4164 ±.013	.3349 ±.010
DF-BFGS	.2982 ±.004	2.4025 ±.005	6.2416 ±.020	.6304 ±.012	.8250 ±.006	.6220 ±.004	.5467 ±.008	.4454 ±.007
KLD ○	.2966 ±.004	2.4059 ±.005	6.2558 ±.020	.6307 ±.013	.8243 ±.006	.6249 ±.005	.5470 ±.008	.4456 ±.007
$\mathcal{S}$-KLD	.2995 ±.005	2.4112 ±.005	6.2883 ±.020	.6491 ±.013	.8203 ±.006	.6205 ±.005	.5384 ±.009	.4385 ±.008
SCL •	.2977 ±.004	2.4028 ±.005	6.2435 ±.021	.6262 ±.013	.8256 ±.006	.6233 ±.005	.5488 ±.008	.4471 ±.007
$\mathcal{S}$-SCL	.2940 ±.005	2.4059 ±.006	6.2589 ±.023	.6203 ±.013	.8268 ±.006	.6281 ±.006	.5518 ±.008	.4497 ±.007
LRR •	.2984 ±.004	2.4046 ±.005	6.2525 ±.019	.6351 ±.012	.8232 ±.006	.6220 ±.004	.5443 ±.008	**.4636** ±.007
$\mathcal{S}$-LRR	**.2937** ±.004	2.4056 ±.005	6.2589 ±.019	**.6189** ±.013	**.8271** ±.006	**.6283** ±.005	**.5519** ±.008	.4498 ±.007

Table 16: Experimental results of LDL on the Flickr dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
LDSVR	.5174 ±.006	2.6364 ±.002	7.2094 ±.011	5.0366 ±.086	.6636 ±.008	.4683 ±.006	.4622 ±.009	.3811 ±.008
AA-kNN	.3286 ±.005	**2.0685** ±.009	**4.9363** ±.033	2.2172 ±.107	.7200 ±.006	.5582 ±.005	.4265 ±.009	.3465 ±.007
LDLFs	.4051 ±.011	2.4012 ±.012	6.3262 ±.050	1.4274 ±.077	.6073 ±.015	.4822 ±.011	.3478 ±.014	.2847 ±.012
DF-BFGS	.3007 ±.005	2.1995 ±.007	5.4900 ±.025	.6309 ±.011	.7801 ±.005	.5979 ±.004	.5102 ±.009	.4226 ±.008
KLD ○	.3015 ±.005	2.2008 ±.007	5.4969 ±.025	.6348 ±.012	.7787 ±.005	.5973 ±.004	.5113 ±.009	.4234 ±.008
$\mathcal{S}$-KLD	.3052 ±.005	2.2044 ±.007	5.5222 ±.026	.6485 ±.012	.7720 ±.005	.5926 ±.004	.5030 ±.009	.4166 ±.008
SCL •	.3280 ±.013	2.2986 ±.024	5.9247 ±.099	.8301 ±.055	.7268 ±.018	.5713 ±.015	.4566 ±.024	.3748 ±.022
$\mathcal{S}$-SCL	**.2929** ±.005	2.2045 ±.007	5.5289 ±.028	.6113 ±.012	.7862 ±.005	**.6070** ±.005	**.5265** ±.008	**.4373** ±.007
LRR •	.3057 ±.005	2.1969 ±.007	5.4763 ±.025	.6431 ±.012	.7752 ±.006	.5929 ±.004	.5047 ±.009	.4229 ±.008
$\mathcal{S}$-LRR	.2938 ±.005	2.2013 ±.006	5.5138 ±.023	**.6105** ±.012	**.7864** ±.005	.6058 ±.005	.5261 ±.009	.4369 ±.008

Table 17: Experimental results of LDL on the `Natural_Scene` dataset formatted as (`mean` ± `std`)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
LDSVR	.4899 ±.016	2.0831 ±.025	5.7724 ±.092	2.0862 ±.085	.5740 ±.017	.4430 ±.015	.4997 ±.015	.3695 ±.012
AA-kNN	.3113 ±.014	**1.9066** ±.034	**4.5413** ±.110	1.0874 ±.082	.7113 ±.015	.5636 ±.013	.4921 ±.021	.3518 ±.016
LDLFs	.2808 ±.034	2.4329 ±.024	6.6027 ±.108	**.6464** ±.118	.7679 ±.046	.5839 ±.043	.5406 ±.058	.4072 ±.045
DF-BFGS	.3074 ±.013	2.4126 ±.017	6.5896 ±.072	.7603 ±.033	.7381 ±.013	.5568 ±.011	.5110 ±.017	.3837 ±.013
KLD •	.3201 ±.013	2.4242 ±.017	6.6560 ±.070	.8285 ±.044	.7172 ±.015	.5485 ±.011	.4958 ±.016	.3715 ±.012
$\mathcal{S}$-KLD	.2743 ±.013	2.3866 ±.020	6.4733 ±.077	.6608 ±.039	**.7751** ±.014	.6133 ±.012	.5592 ±.017	.4221 ±.014
SCL •	.3379 ±.014	2.4800 ±.018	6.8659 ±.076	.8867 ±.035	.7014 ±.014	.4801 ±.014	.4109 ±.018	.3025 ±.013
$\mathcal{S}$-SCL	**.2733** ±.013	2.3734 ±.018	6.4376 ±.072	.6703 ±.043	.7744 ±.015	.6156 ±.013	.5573 ±.018	.4207 ±.014
LRR •	.3138 ±.013	2.4469 ±.018	6.7118 ±.074	.7703 ±.032	.7363 ±.013	.5456 ±.011	.5056 ±.016	.3782 ±.012
$\mathcal{S}$-LRR	.2740 ±.018	2.3461 ±.023	6.3467 ±.087	.6867 ±.070	.7715 ±.021	**.6199** ±.017	**.5595** ±.023	**.4228** ±.018

Table 18: Experimental results of LDL on the `Movie` dataset formatted as (`mean` ± `std`)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
CPNN	.1337 ±.003	.5639 ±.010	1.0746 ±.020	.1191 ±.005	.9194 ±.003	.8164 ±.004	.6610 ±.013	**.7080** ±.002
AA-kNN	.1223 ±.002	.5451 ±.009	1.0445 ±.018	.1129 ±.004	.9254 ±.003	.8250 ±.003	.6557 ±.011	.5710 ±.010
LDLFs	.1172 ±.003	.5233 ±.013	1.0134 ±.026	.1086 ±.006	.9305 ±.003	.8324 ±.004	.6929 ±.013	.6051 ±.012
DF-BFGS	.1210 ±.002	.5282 ±.009	1.0158 ±.019	.1084 ±.005	.9289 ±.003	.8301 ±.003	.6848 ±.012	.5963 ±.012
LRR •	.1135 ±.002	.5101 ±.009	.9770 ±.018	.0957 ±.004	.9369 ±.002	.8385 ±.003	.7119 ±.011	.6203 ±.011
$\mathcal{S}$-LRR	.1125 ±.002	.5086 ±.009	.9717 ±.018	**.0945** ±.004	**.9376** ±.002	.8398 ±.003	**.7126** ±.011	.6227 ±.011
QFD^2 •	.1159 ±.002	.5200 ±.009	.9920 ±.018	.0975 ±.004	.9355 ±.002	.8357 ±.003	.7075 ±.011	.6158 ±.011
$\mathcal{S}$-QFD^2	**.1123** ±.002	.5073 ±.009	.9700 ±.018	**.0945** ±.004	**.9376** ±.002	**.8401** ±.003	.7125 ±.011	.6224 ±.011
CJS •	.1153 ±.002	.5127 ±.009	.9845 ±.019	.0984 ±.004	.9352 ±.002	.8368 ±.003	.7103 ±.012	.6178 ±.011
$\mathcal{S}$-CJS	**.1123** ±.002	**.5072** ±.009	**.9699** ±.018	**.0945** ±.004	**.9376** ±.002	**.8401** ±.003	.7125 ±.011	.6223 ±.011

Table 19: Experimental results of LDL on the `fbp5500` dataset formatted as (`mean` ± `std`)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
CPNN	.1864 ±.005	1.3367 ±.009	2.3604 ±.020	.1664 ±.005	.9281 ±.004	.7958 ±.005	.8688 ±.005	.7831 ±.007
AA-kNN	.1515 ±.004	**1.0443** ±.015	**1.7295** ±.031	.1846 ±.016	.9419 ±.004	.8317 ±.005	.8865 ±.006	.8123 ±.008
LDLFs	.1307 ±.003	1.2787 ±.010	2.1703 ±.024	.1002 ±.005	.9575 ±.003	.8552 ±.004	.9060 ±.005	.8352 ±.007
DF-BFGS	.1341 ±.003	1.2889 ±.010	2.1982 ±.023	.1050 ±.005	.9551 ±.003	.8523 ±.004	.9047 ±.005	.8337 ±.007
LRR	.1312 ±.003	1.2767 ±.010	2.1655 ±.024	.1004 ±.004	.9575 ±.002	.8547 ±.003	.9059 ±.004	.8350 ±.006
$\mathcal{S}$-LRR	**.1302** ±.003	1.2796 ±.010	2.1717 ±.024	**.0997** ±.005	**.9576** ±.002	.8558 ±.003	.9063 ±.004	.8425 ±.006
QFD^2 •	.1380 ±.003	1.2803 ±.010	2.1858 ±.024	.1084 ±.005	.9535 ±.003	.8476 ±.004	.9021 ±.004	.8297 ±.006
$\mathcal{S}$-QFD^2	.1321 ±.004	1.2811 ±.010	2.1779 ±.024	.1027 ±.006	.9561 ±.003	.8537 ±.004	.9044 ±.005	.8330 ±.007
CJS •	.1343 ±.003	1.3057 ±.010	2.2374 ±.024	.1084 ±.005	.9544 ±.003	.8527 ±.004	.9044 ±.004	.8334 ±.006
$\mathcal{S}$-CJS	**.1302** ±.003	1.2802 ±.010	2.1731 ±.024	**.0997** ±.005	.9575 ±.002	**.8559** ±.003	**.9066** ±.004	**.8429** ±.007

Table 20: Experimental results of IncomLDL on the `JAFFE` dataset formatted as (`mean` ± `std`)

Algorithms	$\omega = 20\%$							
	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
IncomLDL •	.0898 ±.010	.3304 ±.024	.6742 ±.049	**.0425** ±.007	**.9598** ±.007	.8861 ±.009	.4742 ±.094	.4017 ±.082
$\mathcal{S}$-IncomLDL	**.0863** ±.013	**.3179** ±.036	**.6525** ±.077	.0433 ±.012	.9590 ±.011	**.8893** ±.014	**.5034** ±.114	**.4401** ±.100
Algorithms	$\omega = 40\%$							
	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
IncomLDL •	.0946 ±.010	.3454 ±.026	.7073 ±.053	.0465 ±.007	.9558 ±.007	.8801 ±.010	.4231 ±.086	.3534 ±.074
$\mathcal{S}$-IncomLDL	**.0868** ±.013	**.3211** ±.038	**.6568** ±.081	**.0439** ±.014	**.9585** ±.012	**.8886** ±.015	**.5075** ±.115	**.4434** ±.098

Table 21: Experimental results of IncomLDL on the SBU_3DFE dataset formatted as (mean ± std)

Algorithms	$\omega = 20\%$							
	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
IncomLDL •	.1088 $\pm.003$	.3586 $\pm.008$	.7586 $\pm.017$	.0574 $\pm.003$	.9439 $\pm.003$	.8655 $\pm.003$	.3171 $\pm.027$	.2746 $\pm.023$
$\mathcal{S}$-IncomLDL	**.1014** $\pm.004$	**.3208** $\pm.009$	**.6727** $\pm.019$	**.0516** $\pm.003$	**.9485** $\pm.003$	**.8790** $\pm.004$	**.4255** $\pm.026$	**.3679** $\pm.023$
Algorithms	$\omega = 40\%$							
	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	Ken. ↑
IncomLDL •	.1104 $\pm.003$	.3621 $\pm.008$	.7673 $\pm.017$	.0586 $\pm.003$	.9426 $\pm.003$	.8638 $\pm.003$	.3003 $\pm.024$	.2595 $\pm.020$
$\mathcal{S}$-IncomLDL	**.1016** $\pm.004$	**.3213** $\pm.009$	**.6748** $\pm.019$	**.0516** $\pm.003$	**.9485** $\pm.003$	**.8786** $\pm.004$	**.4232** $\pm.025$	**.3659** $\pm.022$

Table 22: Experimental results of LDL4C on JAFFE and Twitter formatted as (mean ± std)

Algorithms	JAFFE		Algorithms	Twitter	
	0/1 loss ↓	Err. prob. ↓		0/1 loss ↓	Err. prob. ↓
LDL4C •	.4973 $\pm.108$	.7665 $\pm.020$	LDL4C	.9081 $\pm.009$	.8846 $\pm.005$
$\mathcal{S}$-LDL4C	**.4453** $\pm.102$	**.7600** $\pm.019$	$\mathcal{S}$-LDL4C	.8714 $\pm.207$	.8729 $\pm.156$
LDL-HR	.4786 $\pm.097$	.7676 $\pm.020$	LDL-HR •	.3656 $\pm.017$	.4928 $\pm.011$
$\mathcal{S}$-HR	.4653 $\pm.105$	.7655 $\pm.019$	$\mathcal{S}$-HR	**.2753** $\pm.013$	**.4250** $\pm.008$
LDLM	.4787 $\pm.109$	.7687 $\pm.021$	LDLM •	.2814 $\pm.013$	.4291 $\pm.008$
$\mathcal{S}$-LDLM	.4737 $\pm.097$	.7689 $\pm.019$	$\mathcal{S}$-LDLM	**.2753** $\pm.014$	**.4250** $\pm.008$

Table 23: Experimental results of LDL4C on sBU_3DFE and Flickr formatted as (mean ± std)

Algorithms	sBU_3DFE		Algorithms	Flickr	
	0/1 loss ↓	Err. prob. ↓		0/1 loss ↓	Err. prob. ↓
LDL4C	.5578 $\pm.028$	.7671 $\pm.007$	LDL4C	.8971 $\pm.008$	.8884 $\pm.004$
$\mathcal{S}$-LDL4C	.5526 $\pm.025$	.7686 $\pm.006$	$\mathcal{S}$-LDL4C	.8705 $\pm.138$	.8702 $\pm.100$
LDL-HR •	.5167 $\pm.027$	.7596 $\pm.006$	LDL-HR •	.4513 $\pm.015$	.5823 $\pm.007$
$\mathcal{S}$-HR	.5069 $\pm.025$	.7598 $\pm.006$	$\mathcal{S}$-HR	**.4219** $\pm.015$	**.5639** $\pm.007$
LDLM •	.5258 $\pm.034$	.7619 $\pm.009$	LDLM •	.4384 $\pm.014$	.5740 $\pm.007$
$\mathcal{S}$-LDLM	**.4809** $\pm.024$	**.7524** $\pm.005$	$\mathcal{S}$-LDLM	.4321 $\pm.016$	.5667 $\pm.007$

Table 24: Experimental results of LE on the JAFFE dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑
LP	.0812 $\pm.001$	.3446 $\pm.002$	.7125 $\pm.005$	.0424 $\pm.001$	.9618 $\pm.001$	.8808 $\pm.001$
GLLE	.0821 $\pm.002$	.3196 $\pm.013$	.6518 $\pm.028$	.0386 $\pm.003$	.9638 $\pm.002$	.8901 $\pm.004$
LEVI	.0787 $\pm.003$	.3316 $\pm.013$	.6864 $\pm.028$	.0391 $\pm.003$	.9649 $\pm.002$	.8860 $\pm.004$
LIBLE •	.0813 $\pm.006$	.3106 $\pm.020$	.6358 $\pm.044$	.0370 $\pm.005$	.9652 $\pm.005$	.8929 $\pm.008$
$\mathcal{S}$-LIBLE	**.0770** $\pm.003$	**.2942** $\pm.007$	**.5997** $\pm.016$	**.0332** $\pm.002$	**.9685** $\pm.002$	**.8987** $\pm.003$

Table 25: Experimental results of LE on the Yeast_heat dataset formatted as (mean ± std)

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑
LP	**.0421** $\pm.000$	.2148 $\pm.000$	.4711 $\pm.001$	.0153 $\pm.000$	.9860 $\pm.000$	.9235 $\pm.000$
GLLE	.0481 $\pm.001$	.2114 $\pm.005$	.4282 $\pm.011$	.0168 $\pm.001$	.9842 $\pm.001$	.9298 $\pm.002$
LEVI	.0494 $\pm.007$	.2125 $\pm.027$	.4307 $\pm.056$	.0169 $\pm.004$	.9838 $\pm.004$	.9289 $\pm.009$
LIBLE •	.0453 $\pm.000$	.1973 $\pm.001$	.3982 $\pm.003$	.0148 $\pm.000$	.9859 $\pm.000$	.9346 $\pm.000$
$\mathcal{S}$-LIBLE	.0445 $\pm.000$	**.1901** $\pm.002$	**.3790** $\pm.005$	**.0137** $\pm.000$	**.9869** $\pm.000$	**.9376** $\pm.001$