Approximately Correct Label Distribution Learning

Weiwei Li [1] **Haitao Wu** [2 1] **Yunan Lu** [2 3] **Xiuyi Jia** [2]

Abstract

Label distribution learning (LDL) is a powerful learning paradigm that emulates label polysemy by assigning label distributions over the label space. However, existing LDL evaluation metrics struggle to capture meaningful performance differences due to their insensitivity to subtle distributional changes, and existing LDL learning objectives often exhibit biases by disproportionately emphasizing a small subset of samples with extreme predictions. As a result, the LDL metrics lose their discriminability, and the LDL objectives are also at risk of overfitting. In this paper, we propose DeltaLDL, a percentage of predictions that are approximately correct within the context of LDL, as a solution to the above problems. DeltaLDL can serve as a novel evaluation metric, which is parameter-free and reflects more on real performance improvements. DeltaLDL can also serve as a novel learning objective, which is differentiable and encourages most samples to be predicted as approximately correct, thereby mitigating overfitting. Our theoretical analysis and empirical results demonstrate the effectiveness of the proposed solution.

1. Introduction

Label polysemy is a common phenomenon in real-world scenarios, traditionally emulated by multi-label learning (Zhang & Zhou, 2013), which treats labels in a binary fashion. In contrast, label distribution learning (LDL) (Geng, 2016) offers a more refined framework by answering the question: "*To what extent does each label y describe the instance $\boldsymbol{x}$?*" This is achieved through the concept of a *label distribution* $\boldsymbol{d}$, represented as a probability simplex, where each label is associated with a real-valued *description degree* $d_{\boldsymbol{x}}^{y}$. By quantifying label polysemy, LDL enables a broader spectrum of applications, including tasks like object detection (Xu et al., 2023), age estimation (Wen et al., 2023), sentiment analysis (Chen et al., 2020), and diagnostic assistance (Li et al., 2023), etc.

For years, there are some deep-rooted problems in the field of LDL: 1) existing evaluation metrics for LDL are based on some distance/similarity metrics, e.g., Kullback-Leibler divergence (KLD), which have very small values and thus have poor discriminability between superior and inferior models, since the label distribution is subject to two constraints, non-negativity (i.e., $d_{\boldsymbol{x}}^{y} \geq 0$) and sum-to-one (i.e., $\sum_{y \in \mathcal{Y}} d_{\boldsymbol{x}}^{y} = 1$); 2) existing LDL algorithms mainly focus on minimizing the average of some kind of measurement, usually the KLD as well, which may lead to overfitting due to excessive preference for some samples, resulting even worse performance than that calculated from uniform vectors.

To kill two birds with one stone, the idea of approximately correct prediction can be introduced. On the one hand, it enables the design of percentage-based metrics, quantifying what proportion of the samples are predicted approximately correctly, which is intuitive and semantically meaningful; on the other hand, it enables the design of noise-robust objectives, allowing the model to sacrifice samples that are particularly difficult to learn, ensuring that most samples are predicted approximately correctly, thereby avoiding overfitting to a specific subset of data.

Contributions and organizational structure In this paper: 1) we conduct a theoretical analysis of the KLD to demonstrate its unsuitability as an evaluation metric/learning objective for LDL (Section 2.1); 2) we propose DeltaLDL, a function that calculates the percentage of label distributions that are approximately correctly predicted, exhibiting desirable properties, making it a promising candidate for both metrics and objectives (Section 2.2); 3) based on DeltaLDL, we propose a novel evaluation metric, the μ metric, which possesses superior discriminative power according to our analysis (Section 3); 4) based on DeltaLDL, we propose a novel learning objective and encapsulate it in a new LDL algorithm, named δ-LDL, which can achieve highly compet-

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China [2]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China [3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. Correspondence to: Xiuyi Jia <jiaxy@njust.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

itive performance according to our experiments (Sections 4 and 5). Related work and limitations are discussed in Sections 6 and 7, respectively.

2. Theoretical analysis

In this section, we conduct a theoretical analysis to argue that the KLD is not suitable as an evaluation metric/learning objective for LDL. Then, we propose our new solution, named DeltaLDL.

Preliminaries Let $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^q$ denote the feature of the instance and $\boldsymbol{d} = (d_{\boldsymbol{x}}^{y_j})_{j=1}^c \in \Delta^{c-1}$ denote the label distribution, where c is the number of the labels, y_j is the j-th label in the label space $\mathcal{Y} = \{y_j\}_{j=1}^c$, and

$$\Delta^{k-1} \triangleq \{\boldsymbol{v} \in \mathbb{R}^k \mid \mathbf{1}\boldsymbol{v}^\top = 1,\, \boldsymbol{v} \geq 0\} \tag{1}$$

is the $(k-1)$-dimensional probability simplex. LDL's goal is to find a mapping $f : \mathcal{X} \mapsto \Delta^{c-1}$. More specifically, given a training set $\{\boldsymbol{x}_i,\, \boldsymbol{d}_i\}_{i=1}^m$ and an LDL model $f(\cdot;\, \boldsymbol{\Theta})$, the loss function can be defined as

$$\ell(\boldsymbol{\Theta}) = \frac{1}{m}\sum_{i=1}^{m} \text{dist}(f(\boldsymbol{x}_i;\, \boldsymbol{\Theta}),\, \boldsymbol{d}_i), \tag{2}$$

where $\boldsymbol{\Theta}$ is the model parameters and $\text{dist}(\cdot,\, \cdot)$ is a distance metric, usually the KLD. The optimal parameters $\boldsymbol{\Theta}^*$ can be obtained by minimizing Equation (2), i.e.,

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \ell(\boldsymbol{\Theta}). \tag{3}$$

2.1. Issues of existing metrics: KLD as an example

KLD is defined as

$$\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v}) = \sum_{j=1}^{c} u_j \log \frac{u_j}{v_j}, \tag{4}$$

which exposes shortcomings in two aspects. First, KLD is *not bounded and is only sensitive to small probability values of the prediction*, i.e., $\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v}) \in [0, +\infty)$ and $\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v}) \to \infty$ if $v_j \to 0^+, \exists j \in [c]$. Most existing LDL methods are based on maximum entropy models or neural networks with softmax as the final mathematical processing, hence it is difficult to trigger its sensitivity. Second, KLD *does not adequately capture the changing nature of probability simplexes*, i.e., $\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v})$ lacks the significance of changes to tell the difference between superior and inferior models when $\boldsymbol{u},\, \boldsymbol{v} \in \Delta^{c-1}$. To illustrate the second aspect, we introduce the following assumption and proposition.

Assumption 2.1. The ground-truth label distribution $\boldsymbol{d}$ follows a Dirichlet distribution, i.e., the probability density function of $\boldsymbol{d}$ is given by

$$p(\boldsymbol{d}_i;\, \boldsymbol{\alpha}) = \frac{1}{\text{B}(\boldsymbol{\alpha})}\prod_{j=1}^{c}(d_{\boldsymbol{x}_i}^{y_j})^{\alpha_j - 1}, \quad \text{B}(\boldsymbol{\alpha}) = \frac{\prod_{j=1}^{c}\Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{c}\alpha_j)}, \tag{5}$$

Table 1. Case analysis of two LDL datasets

Case	Dataset	$\mathbb{E}(\text{KL}(\boldsymbol{u} \,\|\, \boldsymbol{v}))$	SA-BFGS	LRR
1)	SBU_3DFE	.0707	.0627	.0571
2)	M^{2}B	.6335	.6878	.6664

where $\Gamma(\cdot)$ is the gamma function. Assumption 2.1 is currently the most common assumption in the literature of LDL (Lu & Jia, 2022; He et al., 2024). Based on Equation (5), we have the following proposition.

Proposition 2.2. *The expectation of the KLD between a Dirichlet-distributed random vector $\boldsymbol{u}$ and a uniform vector*

$$\boldsymbol{v} = \underbrace{\left(\frac{1}{c},\, \frac{1}{c},\, \cdots,\, \frac{1}{c}\right)}_{c} \tag{6}$$

is given by

$$\mathbb{E}\left(\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v})\right) = \sum_{j=1}^{c}\left(\frac{\alpha_j}{\sum_{k=1}^{c}\alpha_k}\right.$$
$$\left.\left(\psi(\alpha_j + 1) - \psi\left(\sum_{k=1}^{c}\alpha_k + 1\right)\right) + \frac{\alpha_j}{\sum_{k=1}^{c}\alpha_k}\log c\right), \tag{7}$$

where $\psi(\cdot)$ is the digamma function. $\mathbb{E}(\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v}))$ can be regarded as the theoretical stochastic KLD performance of an LDL model.

Proof. See Appendix A for details. □

Remark 2.3. To estimate $\boldsymbol{\alpha}$ from given data $(\boldsymbol{d}_i)_{i=1}^m$, one can employ maximum likelihood estimation (MLE), i.e., minimizing:

$$\begin{aligned}\ell_{\text{MLE}}(\boldsymbol{\alpha}) = &-\sum_{i=1}^{m}\log p(\boldsymbol{d}_i;\, \boldsymbol{\alpha}) = \\ &-m\left(\ln\Gamma\left(\sum_{j=1}^{c}\alpha_j\right) - \sum_{j=1}^{c}\ln\Gamma(\alpha_j)\right) \\ &-\sum_{i=1}^{m}\sum_{j=1}^{c}(\alpha_j - 1)\ln d_{\boldsymbol{x}_i}^{y_j},\end{aligned} \tag{8}$$

the closed-form solution of which does not exist. Thus, one can employ numerical optimization methods, e.g., L-BFGS, to solve the optimization problem.

With Proposition 2.2 and Remark 2.3, one can calculate $\mathbb{E}(\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v}))$ for each LDL dataset. Here we conduct a case analysis of two LDL datasets, SBU_3DFE and M^{2}B, and compare the KLD performance of two learned LDL methods, SA-BFGS (Geng, 2016) and LRR (Jia et al., 2023b), both of which exploit average KLD as the learning objective.

The results are shown in Table 1. In case 1), since label distributions in SBU_3DFE are non-sparse, the calculated values are very small, lacking discriminability, i.e., considerable performance improvements are only reflected via these nuanced changes, which is counterintuitive; in case 2), results of SA-BFGS and LRR are worse than the stochastic performance, indicating that minimizing average KLD sometimes does not work as expected, which may be attributed to the negative contribution of some extremely predicted samples.

2.2. The proposed solution: DeltaLDL

On the one hand, Proposition 2.2 fully illustrates the non-significance of KLD as an evaluation metric/learning objective for LDL; on the other hand, it provides a baseline of performance improvements to get a glimpse of *real* performance improvements, which will be discussed in Section 3. Taking inspiration from the analysis of Table 1, we define DeltaLDL as our new solution.

Definition 2.4 (Approximately correct LDL; DeltaLDL)**.** Suppose that an LDL model $f(\cdot; \boldsymbol{\Theta})$ is evaluated on $\{\boldsymbol{x}_i, \boldsymbol{d}_i\}_{i=1}^n$; DeltaLDL of a distance metric $\text{dist}(\cdot, \cdot)$ is defined as the following function:

$$\mathfrak{D}(\text{dist}, \delta; f) \triangleq \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\text{dist}(\boldsymbol{d}_i, f(\boldsymbol{x}_i; \boldsymbol{\Theta})) \leq \delta), \quad (9)$$

where $\mathbb{I}(\cdot)$ is the indicator function to count the number of approximately correct predictions. DeltaLDL of a similarity metric $\text{sim}(\cdot, \cdot)$ is defined as the following function:

$$\mathfrak{D}(\text{sim}, \delta; f) \triangleq \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\text{sim}(\boldsymbol{d}_i, f(\boldsymbol{x}_i; \boldsymbol{\Theta})) \geq \delta). \quad (10)$$

Remark 2.5. The following properties hold for $\mathfrak{D}$:

- $\mathfrak{D}(\cdot, \cdot; \cdot) \in [0, 1]_{\mathbb{R}}$;
- $\mathfrak{D}(\text{dist}, \sup_{\boldsymbol{u},\boldsymbol{v}\in\Delta^{c-1}} \text{dist}(\boldsymbol{u}, \boldsymbol{v}); \cdot) = 1$;
- $\mathfrak{D}(\text{sim}, \inf_{\boldsymbol{u},\boldsymbol{v}\in\Delta^{c-1}} \text{sim}(\boldsymbol{u}, \boldsymbol{v}); \cdot) = 1$;
- $\mathfrak{D}(\text{dist}, \delta; \cdot) \leq \mathfrak{D}(\text{dist}, \delta'; \cdot)$, if $\delta \leq \delta'$;
- $\mathfrak{D}(\text{sim}, \delta; \cdot) \geq \mathfrak{D}(\text{sim}, \delta'; \cdot)$, if $\delta \leq \delta'$;
- $\text{dist}(\boldsymbol{d}_i, f(\boldsymbol{x}_i)) = \inf_{\boldsymbol{u},\boldsymbol{v}\in\Delta^{c-1}} \text{dist}(\boldsymbol{u}, \boldsymbol{v})$, $\forall i \in [n]$, if and only if $\mathfrak{D}(\text{dist}, \inf_{\boldsymbol{u},\boldsymbol{v}\in\Delta^{c-1}} \text{dist}(\boldsymbol{u}, \boldsymbol{v}); f) = 1$;
- $\text{sim}(\boldsymbol{d}_i, f(\boldsymbol{x}_i)) = \sup_{\boldsymbol{u},\boldsymbol{v}\in\Delta^{c-1}} \text{sim}(\boldsymbol{u}, \boldsymbol{v})$, $\forall i \in [n]$, if and only if $\mathfrak{D}(\text{sim}, \sup_{\boldsymbol{u},\boldsymbol{v}\in\Delta^{c-1}} \text{sim}(\boldsymbol{u}, \boldsymbol{v}); f) = 1$.

As stated in Definition 2.4, any distance/similarity metric can be transformed into a percentage of predictions that are approximately correct within the context of LDL. Furthermore, Remark 2.5 shows some good properties of DeltaLDL, which are intuitive and easy to proof by readers. These properties indicate that DeltaLDL has excellent potential as an evaluation metric/learning objective, and also highlight the relationship between DeltaLDL and the distance/similarity metric employed. In this paper, we focus primarily on $\mathfrak{D}(\text{KL}, \cdot; \cdot)$, since the KLD is the most widely used evaluation metric in the field of LDL. Following the definition of DeltaLDL, one may wonder how it can be utilized as an evaluation metric/learning objective.

3. DeltaLDL as an evaluation metric

To transform $\mathfrak{D}(\text{KL}, \cdot; \cdot)$ into an evaluation metric, the following challenges must be addressed: 1) the metric should be parameter-free, as δ varies with different label distributions and is difficult to determine; 2) the metric should effectively reflect real performance improvements, enabling significant differentiation between superior and inferior models. To this end, we propose the following definition.

Definition 3.1 (The discriminative power μ)**.** Suppose that an LDL model is evaluated on the test set, where the ground-truth label distribution matrix is $\boldsymbol{D} = (\boldsymbol{d}_i \overset{\text{i.i.d.}}{\sim} \mathcal{D})_{i=1}^n$, and the predicted label distribution matrix is $\tilde{\boldsymbol{D}} = (\tilde{\boldsymbol{d}}_i)_{i=1}^n$, which is given by the LDL model f; our new evaluation metric based on $\mathfrak{D}(\text{KL}, \delta; f)$, denoted as μ, is defined as

$$\begin{aligned}\mu(\boldsymbol{D}, \tilde{\boldsymbol{D}}) &\triangleq \frac{1}{\delta_0}\int_0^{\delta_0} \mathfrak{D}(\text{KL}, \delta; f)\text{d}\delta \\ &= \frac{1}{\delta_0}\int_0^{\delta_0} \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\text{KL}(\tilde{\boldsymbol{d}}_i \,||\, \boldsymbol{d}_i) \leq \delta)\text{d}\delta \\ &\in [0, 1]_{\mathbb{R}},\end{aligned} \quad (11)$$

where

$$\delta_0 = \mathbb{E}\left(\text{KL}(\boldsymbol{u} \,||\, \boldsymbol{v})\right), \quad \boldsymbol{u} \sim \mathcal{D}, \quad (12)$$

and $\boldsymbol{v}$ is a uniform vector (the same as Equation (6)).

Since the test set is available in the evaluation phase, Equation (12) can be calculated as:

$$\delta_0 \approx \delta_0' = \frac{1}{n}\sum_{i=1}^{n}\text{KL}(\boldsymbol{d}_i \,||\, \boldsymbol{v}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} d_{\boldsymbol{x}_i}^{y_j} \ln d_{\boldsymbol{x}_i}^{y_j} c. \quad (13)$$

According to Definition 3.1, μ's semantic essence lies in the overall performance of the predictions being approximately correct across different values of δ; mathematically, μ is defined as the *ratio* of the area enclosed by the function $\mathfrak{D}(\text{KL}, \delta; \cdot)$ w.r.t. δ and the coordinate axes, where δ_0^{-1} serves as the scaling factor. By adopting an integral form, μ eliminates dependency on specific δ, addressing the aforementioned challenge 1; the introduction of δ_0 establishes a baseline via the value based on uniform vectors, exclusively capturing performance improvements above this baseline, thus effectively resolving the challenge 2.

Discriminability analysis For better comparison, we define the *real* performance improvement based on the KLD/$\mu(\cdot, \cdot)$, with a little symbol abuse.

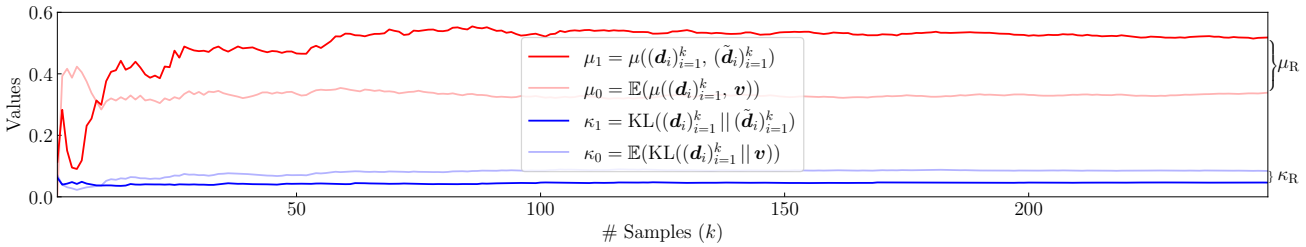


Figure 1. κ_1, κ_0, μ_1, and μ_0 of DF-LDL w.r.t. the number of samples in the test set of SBU_3DFE.

Definition 3.2 (The *real* performance improvement based on the KLD)**.** Let $\kappa_0 = \mathbb{E}(\mathrm{KL}(\boldsymbol{D} \,||\, \boldsymbol{v}))$, where $\boldsymbol{v}$ is a uniform vector; suppose that an LDL model is evaluated on the test set and the KLD is κ_1; the *real* performance improvement based on the KLD is defined as $\kappa_{\mathrm{R}} \triangleq \kappa_1 - \kappa_0$.

Definition 3.3 (The *real* performance improvement based on the μ metric)**.** Let $\mu_0 = \mathbb{E}(\mu(\boldsymbol{D}, \boldsymbol{v}))$, where $\boldsymbol{v}$ is a uniform vector; suppose that an LDL model is evaluated on the test set and the value of the μ metric is μ_1; the *real* performance improvement based on the μ metric is defined as $\mu_{\mathrm{R}} \triangleq \mu_1 - \mu_0$.[1]

Similarly, readers can get the *real* performance improvement based on other evaluation metrics. We train a DF-LDL model on the training set of the SBU_3DFE dataset, and evaluate it on the test set by calculating κ_1, κ_0, μ_1, and μ_0 with varying numbers of test samples. The results are shown in Figure 1. As can be seen, μ_{R} is much larger than κ_{R}, indicating that the μ metric is more sensitive to performance improvements than the KLD, which is consistent with our theoretical analysis. One limitation worth noting is that, when there are few samples for testing, the μ metric is unstable, which is the nature characteristic of percentage-based metrics and can be eliminated by increasing the number of samples. Since DeltaLDL has been successfully transformed into an evaluation metric, the μ metric, one may wonder how it can be utilized as a learning objective.

4. DeltaLDL as a learning objective

To transform $\mathfrak{D}(\mathrm{KL}, \cdot; \cdot)$ into a learning objective, the following challenges must be addressed: 1) the objective should not contain the indicator function $\mathbb{I}(\cdot)$, as it is difficult to optimize; 2) the objective should be to sacrifice a small number of samples that are difficult to learn and ensure that most samples can be predicted as approximately correct. To solve the challenge 1, we introduce a smoothing function $\sigma(\cdot)$ to replace $\mathbb{I}(\cdot)$. Suppose that an LDL model is training on the training set, where the ground-truth label distribution matrix is $\boldsymbol{D} = (\boldsymbol{d}_i \overset{\text{i.i.d.}}{\sim} \mathcal{D})_{i=1}^m$, and the predicted label distribution matrix is $\tilde{\boldsymbol{D}} = (\tilde{\boldsymbol{d}}_i)_{i=1}^m$, which is given by the LDL model f; then $\mathfrak{D}(\mathrm{KL}, \delta; f)$ can be transformed into the following basic loss function:

$$\ell_0(\delta; \boldsymbol{D}, \tilde{\boldsymbol{D}}) = \frac{1}{m}\sum_{i=1}^{m} \sigma\left(\left(\sum_{j=1}^{c} d_{\boldsymbol{x}_i}^{y_j} \ln \frac{d_{\boldsymbol{x}_i}^{y_j}}{\tilde{d}_{\boldsymbol{x}_i}^{y_j}}\right) - \delta\right), \tag{14}$$

which is differentiable if $\sigma(\cdot)$ is appropriate. The above challenge 2 lies in ensuring that $\mathfrak{D}(\mathrm{KL}, \delta; \cdot)$ should be as large as possible while keeping δ as small as possible. (Note that δ is difficult to determine, as mentioned in Section 3.) This requirement can be understood geometrically as minimizing the area enclosed by the function $\ell_0(\delta; \cdot, \cdot)$ w.r.t. δ and the coordinate axes. We can use adaptive Simpson's rule (ASR) (McKeeman, 1962), a numerical integration method, for loss function calculation, to solve this challenge.

Remark 4.1. From Simpson's rule, we have

$$\begin{aligned}\int_a^b \ell_0(\delta; \cdot, \cdot)\mathrm{d}\delta &\approx \varphi_0(a, b) \\ &= \frac{b-a}{6}\left(\ell_0(a) + 4\ell_0\left(\frac{a+b}{2}\right) + \ell_0(b)\right).\end{aligned} \tag{15}$$

In adaptive Simpson's rule, the term $\upsilon_{a,b}$ quantifies the error of the basic Simpson's approximation by considering the subdivision at the midpoint. It is defined as:

$$\upsilon_{a,b} = \varphi_0(a, c) + \varphi_0(c, b) - \varphi_0(a, b), \quad c = \frac{a+b}{2}. \tag{16}$$

The adaptive Simpson's rule refines the integration by recursively subdividing the interval until the error is below a specified tolerance ε.

Remark 4.2. From the adaptive Simpson's rule, we have

$$\begin{aligned}&\int_a^b \ell_0(\delta; \cdot, \cdot)\mathrm{d}\delta \approx \varphi(a, b, \varepsilon) = \\ &\quad\begin{cases}\varphi_0(a, c) + \varphi_0(c, b) + \upsilon_{a,b}/15, & |\upsilon_{a,b}| \le 15\varepsilon, \\ \varphi(a, c, \varepsilon/2) + \varphi(c, b, \varepsilon/2), & \text{o/w.}\end{cases}\end{aligned} \tag{17}$$

[1] For readers who find these definitions obscure, we hereby provide a metaphor: κ_0 and μ_0 are like the accuracy of random selection in multi-classification problems.

Algorithm 1 Adaptive Simpson's rule: ASR

Input: Interval bounds a, b; error tolerance ε; current integral estimate s; maximum recursion depth ξ.
Output: Integral estimate s'.

$c \leftarrow {}^{(a+b)}/_{2}$;
$l \leftarrow \varphi_0(a, c)$; ▷ Equation (15)
$r \leftarrow \varphi_0(c, b)$;
if $|l + r - s| \leq 15\varepsilon$ **or** $t \leq 0$ **then**
  return $l + r + {}^{(l+r-s)}/_{15}$; ▷ Equation (17)
end if
return $\mathrm{ASR}(a, c, {}^{\varepsilon}/_{2}, l, \xi - 1) + \mathrm{ASR}(c, b, {}^{\varepsilon}/_{2}, r, \xi - 1)$;

Algorithm 2 Our proposed algorithm: δ-LDL

Input: Training set $\{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^{m}$, maximum recursion depth ξ, test sample $\boldsymbol{x}'$.
Output: Label distribution $\boldsymbol{d}'$.

EPS $\leftarrow 10^{-7}$;
$\boldsymbol{\Theta} \leftarrow$ random initialization;
$\delta_0' \leftarrow {}^{1}/_{m} \sum_{i=1}^{m} \sum_{j=1}^{c} d_{\boldsymbol{x}_i}^{y_j} \ln d_{\boldsymbol{x}_i}^{y_j} c$; ▷ Equation (19)
for each epoch **to** anticipated epoch number **do**
  $\mathcal{B} \leftarrow$ randomly split $\{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^{m}$ into batches;
  for each batch **in** $\mathcal{B}$ **do**
    $\tilde{\boldsymbol{D}} \leftarrow f(\boldsymbol{x}; \boldsymbol{\Theta}), \forall \boldsymbol{x} \in$ the batch; ▷ Equation (20)
    $s_0 \leftarrow \varphi_0(0, \delta_0')$; ▷ Equation (15)
    $\ell \leftarrow \mathrm{ASR}(0, \delta_0', \mathrm{EPS}, s_0, \xi)$; ▷ Algorithm 1
    $\boldsymbol{\Theta} \leftarrow$ update by Adam, tracing ℓ; ▷ Equation (21)
  end for
end for
$\boldsymbol{d}' \leftarrow f(\boldsymbol{x}'; \boldsymbol{\Theta})$; ▷ Equation (20)
return $\boldsymbol{d}'$;

The rule is summarized in Algorithm 1. Finally, the loss function can be defined as:

$$\ell(\boldsymbol{D}, \tilde{\boldsymbol{D}}) = \varphi(0, \delta_0', \mathrm{EPS}). \tag{18}$$

Implementation details In Equation (14), $\sigma(\cdot)$ is implemented as the ReLU function; in Algorithm 2, ξ is set to 5; in Equation (18),

$$\delta_0' = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{c} d_{\boldsymbol{x}_i}^{y_j} \ln d_{\boldsymbol{x}_i}^{y_j} c, \tag{19}$$

serving as a performance baseline, and EPS can be a small positive number, e.g., 10^{-7}. The above loss function can be optimized by gradient descent methods, such as Adam (Kingma & Ba, 2015). The LDL model f is implemented by a naive network with the parameter matrix $\boldsymbol{\Theta} \in \mathbb{R}^{q \times c}$:

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) = \varsigma(\boldsymbol{x}\boldsymbol{\Theta}), \tag{20}$$

where $\varsigma(\cdot)$ is the softmax function as the final mathematical processing. The overall algorithm is summarized in Algorithm 2, denoted as δ-LDL.

Remark 4.3. Let $\ell_0(\delta; \boldsymbol{D}, f(\boldsymbol{X}; \boldsymbol{\Theta}))$ be $\ell_0(\boldsymbol{\Theta})$; assume that $\sigma(\cdot)$ is the ReLU function; $\tilde{\boldsymbol{D}} = f(\boldsymbol{X}; \boldsymbol{\Theta})$; $\ell_0(\boldsymbol{\Theta})$ is differentiable and its gradient w.r.t. $\boldsymbol{\Theta}$ is given by

$$\frac{\partial \ell_0(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = \mathbb{I}(\mathrm{KL}(\boldsymbol{D} \,||\, \tilde{\boldsymbol{D}}) > \delta) \odot (\boldsymbol{X}^{\top}(\tilde{\boldsymbol{D}} - \boldsymbol{D})), \tag{21}$$

where $\odot$ is the element-wise product.

Time complexity analysis Let $\boldsymbol{X} \in \mathbb{R}^{m \times q}$ denote the feature matrix and $\boldsymbol{D} \in \mathbb{R}^{m \times c}$ denote the ground-truth label distribution matrix; $\tilde{\boldsymbol{D}} = f(\boldsymbol{X}; \boldsymbol{\Theta})$; the overall time cost of δ-LDL (Algorithm 2) is primarily influenced by the following calculations: computing δ_0' in Equation (19) has a complexity of $\mathcal{O}(mc)$; the forward pass step in Equation (20) requires $\mathcal{O}(mqc)$; each calculation of the KLD, $\kappa = \boldsymbol{D} \odot (\log \boldsymbol{D} - \log \tilde{\boldsymbol{D}})$, has a complexity of $\mathcal{O}(mc)$; calculating the gradient, ${}^{\partial \kappa}/_{\partial \Theta} = \boldsymbol{X}^{\top}(\tilde{\boldsymbol{D}} - \boldsymbol{D})$, involves a complexity of $\mathcal{O}(mqc)$. Since the parameter ξ is fixed, ℓ_0 is computed a limited number of times and this number remains relatively small. The overall time complexity of each iteration of δ-LDL is $\mathcal{O}(kmqc)$, where k is the number of calculations of Equation (14), attributed to Algorithm 1. It is linear w.r.t. the number of samples m/labels c, clarifying that our proposed δ-LDL can be applied to large-scale datasets. Then, one may wonder how δ-LDL performs in practice.

5. Experiments

In this section, extensive experiments are conducted to illustrate the superiority of the μ metric and δ-LDL. Details of all implementations are openly accessible at GitHub.[2]

5.1. Experimental setup

Metrics & datasets We mainly evaluate LDL models' performance via the $\mu \uparrow$ metric proposed in Section 3. We also represent the same metrics suggested in (Jia et al., 2023b), which are `Cheby.` $\downarrow$ (Chebyshev distance), `Clark` $\downarrow$ (Clark distance), `Can.` $\downarrow$ (Canberra distance), `KLD` $\downarrow$ (Kullback-Leibler divergence), `Cosine` $\uparrow$ (cosine similarity), `Int.` $\uparrow$ (intersection similarity) and `Spear.` $\uparrow$ (Spearman's coefficient), respectively. Here $\downarrow$ ($\uparrow$) indicates "the lower (higher) the better". We adopt several widely used label distribution datasets, including: `M2B` (Nguyen et al., 2012), `fbp5500` (Liang et al., 2018), `RAF_ML` (Li & Deng, 2019), `SBU_3DFE` (Yin et al., 2006), `Natural_Scene` (Geng et al., 2021), `Music` (Lee et al., 2021) and `Painting` (Machajdik & Hanbury, 2010).

Comparison methods & methodology We compare δ-LDL with several LDL methods. On the one hand, methods based on the optimization of the average KLD are mainly compared, including: SA-BFGS (Geng, 2016), LDLLC (Jia et al., 2018), LDLSF (Ren et al., 2019a), LCLR (Ren et al., 2019b), SCL (Jia et al., 2019), DPA (Jia et al., 2023a) and

[2] https://github.com/SpriteMisaka/PyLDL

Table 2. Experimental results on M^{2}B, fbp5500, and SBU_3DFE, formatted as "mean ± std"

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	μ ↑
				M^{2}B				
PT-Bayes	• .5394 ±.043	• 1.7487 ±.064	• 3.6172 ±.169	• 1.7422 ±.393	• .5272 ±.053	• .4498 ±.042	• .5758 ±.052	• 25.76% ±.043
SCL	• .4788 ±.013	• 1.6358 ±.010	• 3.3495 ±.032	• .7773 ±.027	• .6842 ±.014	• .4741 ±.010	• .5752 ±.110	• 26.67% ±.012
LDLSF	• .3865 ±.018	• 1.5441 ±.031	• 3.0624 ±.083	• 2.0495 ±.497	• .7363 ±.022	• .5991 ±.019	• .6405 ±.026	• 32.95% ±.030
LDLF	• .4171 ±.024	• 1.5266 ±.048	• 3.1013 ±.114	• .5688 ±.051	• .7584 ±.020	• .5604 ±.028	.6984 ±.031	• 44.91% ±.054
LCLR	• .3867 ±.022	• 1.2516 ±.036	• 2.4208 ±.089	• .7303 ±.098	• .7152 ±.026	• .6040 ±.022	.6862 ±.026	• 52.13% ±.031
DPA	• .3847 ±.022	• 1.1734 ±.030	• 2.2681 ±.071	• .7013 ±.089	• .7187 ±.026	• .6075 ±.022	.6936 ±.024	• 52.95% ±.031
LDLLC	• .3845 ±.022	• 1.1733 ±.030	• 2.2683 ±.070	• .7003 ±.089	• .7189 ±.026	• .6077 ±.022	.6939 ±.025	• 52.99% ±.031
SA-BFGS	• .3826 ±.021	• 1.2228 ±.028	• 2.3520 ±.066	• .6878 ±.091	• .7231 ±.025	• .6080 ±.021	.6886 ±.026	• 53.06% ±.029
LRR	• .3793 ±.021	• 1.1675 ±.028	• 2.2525 ±.066	• .6664 ±.088	• .7271 ±.025	• .6127 ±.021	.6986 ±.024	• 53.90% ±.029
AA-kNN	• .3721 ±.020	○ **1.0566** ±.031	○ **1.9578** ±.067	• .6179 ±.074	• .7426 ±.023	• .6191 ±.020	.7022 ±.026	• 55.21% ±.028
DF-LDL	• .3699 ±.019	1.1011 ±.025	2.0883 ±.058	• .5743 ±.072	• .7457 ±.023	• .6219 ±.019	**.7049** ±.022	• 55.36% ±.027
δ-LDL	**.3599** ±.017	1.0966 ±.023	2.0629 ±.051	**.4882** ±.053	**.7768** ±.020	**.6333** ±.017	.6943 ±.023	**58.25%** ±.025
				fbp5500				
PT-Bayes	• .6569 ±.011	• 1.6829 ±.010	• 3.1196 ±.032	• 15.2111 ±.362	• .4779 ±.015	• .3413 ±.011	• .6987 ±.010	• .08% ±.001
SCL	• .2822 ±.005	• 1.4191 ±.007	• 2.6654 ±.022	• .4370 ±.010	• .8442 ±.006	• .6339 ±.004	• .7314 ±.053	• 38.39% ±.011
LDLSF	• .1450 ±.003	○ **1.0405** ±.016	○ **1.6992** ±.031	• .3391 ±.030	• .9515 ±.002	• .8381 ±.003	• .8876 ±.005	• 69.42% ±.013
AA-kNN	• .1515 ±.004	○ 1.0482 ±.015	○ 1.7383 ±.031	• .1819 ±.013	• .9418 ±.003	• .8315 ±.004	• .8860 ±.006	• 78.19% ±.010
LCLR	• .1386 ±.003	• 1.2978 ±.009	• 2.2253 ±.022	• .1114 ±.005	• .9523 ±.003	• .8475 ±.004	• .9009 ±.005	• 84.36% ±.007
SA-BFGS	• .1369 ±.003	• 1.2945 ±.009	• 2.2146 ±.022	• .1084 ±.005	• .9536 ±.003	• .8494 ±.004	• .9025 ±.004	• 84.77% ±.006
LDLF	• .1348 ±.004	1.2786 ±.011	2.1763 ±.028	• .1058 ±.006	• .9549 ±.003	• .8503 ±.005	• .9017 ±.005	• 85.18% ±.008
DF-LDL	• .1346 ±.003	• 1.2898 ±.010	• 2.2008 ±.023	• .1055 ±.005	• .9548 ±.002	• .8518 ±.003	.9042 ±.004	• 85.18% ±.006
DPA	• .1347 ±.003	• 1.2844 ±.010	• 2.1895 ±.024	• .1053 ±.005	• .9548 ±.003	• .8512 ±.003	.9038 ±.004	• 85.19% ±.006
LDLLC	• .1348 ±.003	• 1.2844 ±.010	• 2.1894 ±.024	• .1052 ±.005	• .9549 ±.003	• .8511 ±.003	• .9036 ±.004	• 85.20% ±.006
LRR	• .1343 ±.003	• 1.2837 ±.010	• 2.1871 ±.024	• .1046 ±.004	• .9552 ±.002	• .8516 ±.003	.9040 ±.004	• 85.29% ±.006
δ-LDL	**.1300** ±.003	1.2783 ±.010	2.1685 ±.023	**.0997** ±.005	**.9576** ±.002	**.8560** ±.003	**.9061** ±.004	**85.99%** ±.006
				SBU_3DFE				
PT-Bayes	• .1289 ±.004	• .4019 ±.009	• .8606 ±.020	• .0772 ±.004	• .9247 ±.003	• .8452 ±.004	• .1444 ±.035	• 33.40% ±.016
AA-kNN	• .1272 ±.004	• .4001 ±.009	• .8281 ±.020	• .0801 ±.004	• .9217 ±.004	• .8489 ±.004	• .2053 ±.030	• 36.18% ±.016
LDLSF	• .1091 ±.003	• .4020 ±.011	• .8224 ±.021	• .0874 ±.020	• .9402 ±.003	• .8591 ±.004	• .3358 ±.025	• 37.61% ±.020
SA-BFGS	• .1151 ±.004	• .3706 ±.009	• .7827 ±.020	• .0627 ±.003	• .9384 ±.003	• .8599 ±.004	• .3001 ±.029	• 39.68% ±.018
LCLR	• .1061 ±.004	• .3627 ±.009	• .7531 ±.020	• .0579 ±.003	• .9438 ±.003	• .8669 ±.004	• .3573 ±.025	• 43.42% ±.020
LDLLC	• .1078 ±.003	• .3581 ±.009	• .7476 ±.018	• .0573 ±.003	• .9439 ±.003	• .8671 ±.003	• .3441 ±.025	• 43.49% ±.016
DPA	• .1077 ±.003	• .3580 ±.009	• .7474 ±.019	• .0572 ±.003	• .9440 ±.003	• .8672 ±.003	• .3446 ±.027	• 43.56% ±.017
LRR	• .1078 ±.003	• .3576 ±.009	• .7470 ±.018	• .0571 ±.003	• .9440 ±.003	• .8672 ±.003	• .3421 ±.027	• 43.56% ±.017
SCL	• .1111 ±.004	• .3512 ±.008	• .7523 ±.018	• .0573 ±.003	• .9436 ±.003	• .8658 ±.003	• .2952 ±.027	• 43.61% ±.017
LDLF	• .1023 ±.003	.3310 ±.008	• .6940 ±.017	• .0512 ±.003	• .9492 ±.003	• .8760 ±.003	.4045 ±.026	• 48.82% ±.018
DF-LDL	• .1001 ±.003	.3276 ±.007	.6817 ±.016	**.0493** ±.003	.9511 ±.003	.8780 ±.003	.4273 ±.022	• 49.93% ±.018
δ-LDL	**.0986** ±.004	**.3272** ±.008	**.6794** ±.018	.0494 ±.003	**.9513** ±.003	**.8793** ±.003	**.4345** ±.029	**50.73%** ±.018

LRR (Jia et al., 2023b); on the other hand, methods based on specified structures are also compared, including: PT-Bayes (Geng, 2016), AA-kNN (Geng, 2016), LDLF (Shen et al., 2017) and DF-LDL (González et al., 2021). To ensure a fair comparison, for each dataset and for each method we conduct ten-fold experiments repeated 10 times, and the average performance is recorded.

5.2. Results and discussion

Label distribution prediction Table 2 shows the representative label distribution prediction results. • (○) indicates "δ-LDL is statistically superior (inferior) to the comparing methods" (pairwise t-test at 0.05 significance level); there is no significant if neither • nor ○ is shown; the best and second-best results are highlighted in **bold** and underline, respectively. In terms of performance on M^{2}B, a considerable number of state-of-the-art methods are even inferior to AA-kNN, exposing the limitations of minimizing average KLD; in contrast, δ-LDL achieves the best performance on five metrics, significantly outperforming the other methods. Note that the KLD evaluation of δ-LDL on M^{2}B is 0.4882, which is lower than the stochastic performance, i.e., 0.6335 in Table 1, which fully demonstrates the effectiveness of our proposed δ-LDL. The results on fbp5500 and SBU_3DFE are similar, which implies that δ-LDL is effective on different datasets. Remaining results can be found in Appendix B.

Regarding the μ metric, it is always significant on t-test as our expectation. Note that the ranking derived from the μ metric may be different from that derived from the KLD. An obvious example lies in the comparison between LDLF and DF-LDL on M^{2}B, where the KLD evaluation of LDLF is 0.5688, which is lower than that of DF-LDL, i.e., 0.5743, while the μ metric shows that DF-LDL is superior to LDLF. This is because the μ metric focuses on the percentage of

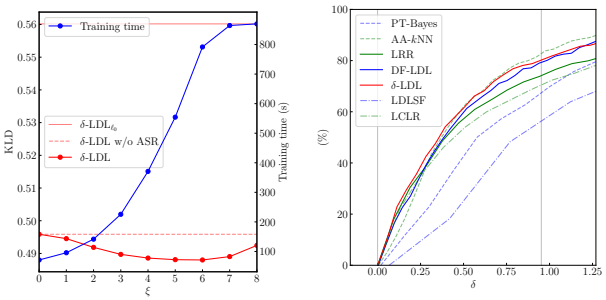
predictions that are approximately correct, which is less susceptible to extreme predictions. Compared with other metrics, the μ metric has explicit range, is more discriminable between superior and inferior models, and is more intuitive, reliable, and interpretable.

Abation study & analysis on parameters Here we are interested in the importance of each processing step in δ-LDL, thus an ablation study on M^{2}B is conducted: 1) we replace the loss function with solely $\ell_0(\delta_0'; \cdot, \cdot)$ to examine the importance of ℓ, and this model is denoted as δ-LDL$_{\ell_0}$; 2) we replace the ASR with solely Simpson's rule by setting $\xi = 0$, and this model is denoted as δ-LDL w/o ASR. The results are shown in Figure 2(a), which confirms that each processing step in δ-LDL is indispensable. We conduct an analysis on parameters on M^{2}B as well, where the KLD and training time w.r.t. ξ are also shown in Figure 2(a). The training time initially increases sharply with the growth of ξ, but gradually levels off. This is because, as ξ increases, more samples are able to meet the conditions for halting the computation of the ASR loss in advance. For the balance between training time and performance, ξ is fixed at 5 in our proposed δ-LDL.

Though δ-LDL is δ-parameter-free, we can still analyze the impact of δ on $\mathfrak{D}(\cdot, \delta; \cdot)$ with predictions of an experiment. We analyze various $\mathfrak{D}(\cdot, \delta; \cdot)$s w.r.t. δ with predictions of one experiment on M^{2}B, results of which are shown in Figure 2(b) and Appendix B. The gray lines indicate the integration interval. In terms of our proposed δ-LDL, $\mathfrak{D}(\mathrm{KL}, \delta; \cdot)$ is larger that that of other methods when δ is small, which makes the area under the curve larger. It is worth noting that $\mathfrak{D}(\mathrm{KL}, \delta; \cdot)$ is a monotonically increasing function of δ, which is consistent with our theoretical analysis.

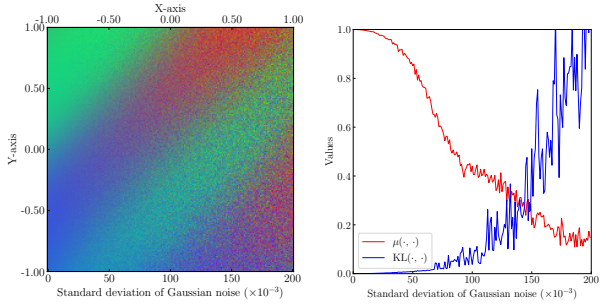
Robustness testing We now evaluate the robustness of the μ metric, exploiting the artificial dataset introduced in (Geng, 2016). First, we generate the artificial dataset with 200×200 samples. Then we continuously apply noise along a certain axis of the features, which is visualized in Figure 3(a). The KLD and μ metric w.r.t. the noise level are shown in Figure 3(b). The KLD lacks the significance of changes when the noise level is low. Then, it sharply increases and becomes highly volatile once the noise level surpasses a certain threshold. This is because when the noise increases to a certain extent, the label distribution will become sparse, triggering the sensitivity of the KLD, which is consistent with our theoretical analysis. The early tiny changes/the later substantial fluctuations in the KLD are detrimental to the discriminable/reliable evaluation of LDL models. In contrast, the μ metric remains stable and gradually decreases with the noise level. This is because the μ metric is based on the percentage of predictions that are approximately correct, which is less sensitive to noise.

Next, we evaluate the robustness of δ-LDL and other methods, exploiting the same artificial dataset mentioned above. From bottom to top, from left to right, the four types of processing in Figure 4(a) are: 1) no treatment; 2) applying Gaussian noise (He et al., 2024); 3) randomly setting description degrees to zero (Xu & Zhou, 2017); 4) randomly emphasizing description degrees (Kou et al., 2023). These treatments correspond to different possible types of noise in label distributions. The prediction results of representative methods are shown in Figures 4(b) to 4(g), and the ground truth, i.e., the original artificial data without noise, is shown in Figure 4(h) for comparison. The consistency of each prediction result reflects the robustness of each method. Although it is hard to visualize the so-called "decision boundary" in the field of LDL, Figure 4 shows that our method predicts accurately and the prediction results are not easily affected by noise.



(a) KLD/training time w.r.t. ξ (b) $\mathfrak{D}(\mathrm{KL}, \delta; \cdot)$ w.r.t. δ

Figure 2. (Best viewed in color) Visualized results of the analysis on parameters.

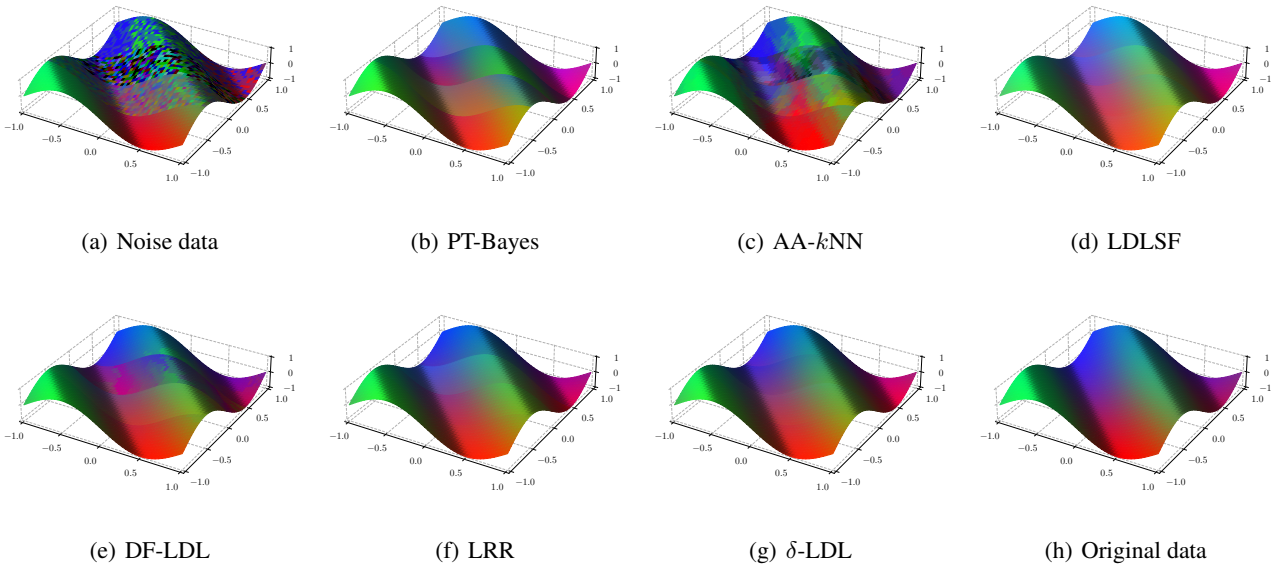


(a) Visualization of noise data (b) KLD/μ w.r.t. noise level

Figure 3. (Best viewed in color) Visualized results of the robustness testing of the μ metric.

6. Related work

Our work is mainly related to LDL. LDL was initially proposed by Geng et al. to tackle the problem of age estimation, and subsequently developed into a novel paradigm of machine learning (Geng, 2016), which is now supported by



(a) Noise data (b) PT-Bayes (c) AA-kNN (d) LDLSF

(e) DF-LDL (f) LRR (g) δ-LDL (h) Original data

Figure 4. (Best viewed in color) Visualized results of the robustness testing of δ-LDL and other methods.

theoretical underpinnings (Wang & Geng, 2019). Over the years, a large number of specialized algorithms for LDL have emerged (Geng et al., 2013; Geng & Hou, 2015; Shen et al., 2017; González et al., 2021).

Regrading evaluation metrics, the average KLD is the most widely used one and it is first considered by Geng, who advocates hierarchical clustering to select representative metrics, and demonstrates that Clark and Canberra metrics suffer from oversensitivity to small values. Subsequent studies by Xu & Zhou revealed KLD's unreliability with sparse predictions, prompting calls for alternative measures. Metrics that used in early work therefore directly borrows from existing mathematical foundations, e.g., Topsϕe (Ren & Geng, 2017), Sϕrensen (Ren et al., 2019a) and Spearman's coefficient (Jia et al., 2023b), etc. There is little work on designing new metrics until recently, e.g., the DPA proposed by Jia et al., which, however, from the empirical results, still lacks discriminability between superior and inferior models. The CAD, QFD^2 and CJS proposed by Wen et al. do not address this issue either, and only work with ordinal label distributions. Our proposed μ metric, which is based on the percentage of predictions that are approximately correct, is the first to tackle this issue.

Regrading learning objectives, the average KLD and the MSE are the most widely adopted. For example, AA-BP (Geng, 2016), a simple 3-layer network, minimizes MSE but underperforms due to overfitting. The average KLD is exploited in many LDL work based on loss function engineering via different optimization methods, including BFGS (Jia et al., 2018; 2023a), ADMM (Ren et al., 2019a;b), and gradient-based methods (Jia et al., 2019; 2023b). These methods 1) implicitly address overfitting through ad-hoc regularization, indicating the field of LDL lacks principled solutions; 2) sometimes are worse than average performance deduced by uniform vectors, which could not even be verified before our work. The essential difference between our proposed δ-LDL and these methods is that it no longer focuses on all samples, but sacrifices a part of them by using an approximately correct interval.

Furthermore, the parameter δ is potentially related to the parameter ρ in the margin theory, which is exploited in some LDL work (Wang & Geng, 2019; 2021a;b). The difference, however, is that: 1) their work focuses on the problem of objective mismatch, using margin theory to improve the performance of classification tasks, destroying the expected shape of the predicted label distributions, while our work still focuses on LDL tasks; 2) the margin ρ is still a hyperparameter that needs to be set in their methods, while our method is δ-parameter-free, which is more convenient for practical scenarios.

7. Limitations and conclusion

Limitations A notable limitation of the proposed δ-LDL algorithm is that it has not yet achieved optimal performance across all metrics, which is close to the theoretical best-case scenario. This is likely due to our current work not considering fitting different shapes of label distributions, which is an area that warrants further exploration.

Conclusion In this paper, we propose DeltaLDL, a percentage of predictions that are approximately correct in LDL. DeltaLDL can serve as a novel evaluation metric, which is parameter-free and reflects real performance improvements; DeltaLDL can also serve as a novel learning objective, which is differentiable and ensures that most samples can be predicted as approximately correct. Our theoretical analysis and empirical results demonstrate the effectiveness of the proposed solution.

Impact statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This research is partially supported by Shanghai Central Guidance Science and Technology Development Fund (YDZX20233100004008), the Natural Science Foundation of Jiangsu Province (BK20242045), and the National Natural Science Foundation of China (62176123, 62476130).

References

Chen, S., Wang, J., Chen, Y., Shi, Z., Geng, X., and Rui, Y. 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*, pp. 13984–13993, 2020.

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

Geng, X. and Hou, P. Pre-release prediction of crowd opinion on movies by label distribution learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3511–3517, 2015.

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

Geng, X., Zheng, R., Lv, J., and Zhang, Y. Multilabel ranking with inconsistent rankers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5211–5224, 2021.

González, M., González-Almagro, G., Triguero, I., Cano, J.-R., and García, S. Decomposition-fusion for label distribution learning. *Information Fusion*, 66:64–75, 2021.

He, L., Lu, Y., Li, W., and Jia, X. Generative calibration of inaccurate annotation for label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 12394–12401, 2024.

Jia, X., Li, W., Liu, J., and Zhang, Y. Label distribution learning by exploiting label correlations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3310–3317, 2018.

Jia, X., Li, Z., Zheng, X., Li, W., and Huang, S.-J. Label distribution learning with label correlations on local samples. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1619–1631, 2019.

Jia, X., Qin, T., Lu, Y., and Li, W. Adaptive weighted ranking-oriented label distribution learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35 (8):11302–11316, 2023a.

Jia, X., Shen, X., Li, W., Lu, Y., and Zhu, J. Label distribution learning by maintaining label ranking relation. *IEEE Transactions on Knowledge and Data Engineering*, 35 (02):1695–1707, 2023b.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

Kou, Z., Wang, J., Jia, Y., Liu, B., and Geng, X. Instance-dependent inaccurate label distribution learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):1425–1437, 2023.

Lee, H., Höger, F., Schoenwiesner, M., Park, M., and Jacoby, N. Cross-cultural mood perception in pop songs and its alignment with mood detection algorithms. In *Proceedings of the International Society for Music Information Retrieval Conference*, 2021.

Li, S. and Deng, W. Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning. *International Journal of Computer Vision*, 127(6-7):884–906, 2019.

Li, X., Luo, G., Wang, W., Wang, K., and Li, S. Curriculum label distribution learning for imbalanced medical image segmentation. *Medical Image Analysis*, 89:102911, 2023.

Liang, L., Lin, L., Jin, L., Xie, D., and Li, M. SCUT-FBP5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In *Proceedings of the International Conference on Pattern Recognition*, pp. 1598–1603, 2018.

Lu, Y. and Jia, X. Predicting label distribution from multi-label ranking. In *Proceedings of the Annual Conference*

on Neural Information Processing Systems, pp. 36931–36943, 2022.

Machajdik, J. and Hanbury, A. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the ACM International Conference on Multimedia*, pp. 83–92, 2010.

McKeeman, W. M. Algorithm 145: Adaptive numerical integration by simpson's rule. *Communications of the ACM*, 5(12):604, 1962.

Nguyen, T. V., Liu, S., Ni, B., Tan, J., Rui, Y., and Yan, S. Sense beauty via face, dressing, and/or voice. In *Proceedings of the ACM International Conference on Multimedia*, pp. 239–248, 2012.

Ren, T., Jia, X., Li, W., Chen, L., and Li, Z. Label distribution learning with label-specific features. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3318–3324, 2019a.

Ren, T., Jia, X., Li, W., and Zhao, S. Label distribution learning with label correlations via low-rank approximation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3325–3331, 2019b.

Ren, Y. and Geng, X. Sense beauty by label distribution learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2648–2654, 2017.

Shen, W., Zhao, K., Guo, Y., and Yuille, A. Label distribution learning forests. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, pp. 834–843, 2017.

Wang, J. and Geng, X. Classification with label distribution learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3712–3718, 2019.

Wang, J. and Geng, X. Label distribution learning machine. In *Proceedings of the International Conference on Machine Learning*, pp. 10749–10759, 2021a.

Wang, J. and Geng, X. Learn the highest label and rest label description degrees. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3097–3103, 2021b.

Wen, C., Zhang, X., Yao, X., and Yang, J. Ordinal label distribution learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23481–23491, 2023.

Xu, H., Liu, X., Zhao, Q., Ma, Y., Yan, C., and Dai, F. Gaussian label distribution learning for spherical image object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1033–1042, 2023.

Xu, M. and Zhou, Z.-H. Incomplete label distribution learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3175–3181, 2017.

Yin, L., Wei, X., Sun, Y., Wang, J., and Rosato, M. J. A 3D facial expression database for facial behavior research. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 211–216, 2006.

Zhang, M.-L. and Zhou, Z.-H. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.

A. Proof of Proposition 2.2

Proof. Let $\boldsymbol{u} = (u_1, u_2, \cdots, u_c)$ be a random vector distributed according to a Dirichlet distribution, i.e., $\boldsymbol{u} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, with parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_c)$. The KLD between $\boldsymbol{u}$ and $\boldsymbol{v}$ is given by $\mathrm{KL}(\boldsymbol{u} \,||\, \boldsymbol{v}) = \sum_{j=1}^{c} u_j \log(cu_j)$. Then the expectation of the KLD can be decomposed as

$$\mathbb{E}\left(\mathrm{KL}(\boldsymbol{u} \,||\, \boldsymbol{v})\right) = \mathbb{E}\left(\sum_{j=1}^{c} u_j \log(cu_j)\right) = \sum_{j=1}^{c} \mathbb{E}\left(u_j \log(cu_j)\right) = \sum_{j=1}^{c} \left(\mathbb{E}(u_j \log u_j) + \mathbb{E}(u_j) \log c\right). \tag{22}$$

From the properties of the Dirichlet distribution, the expected value of u_j is

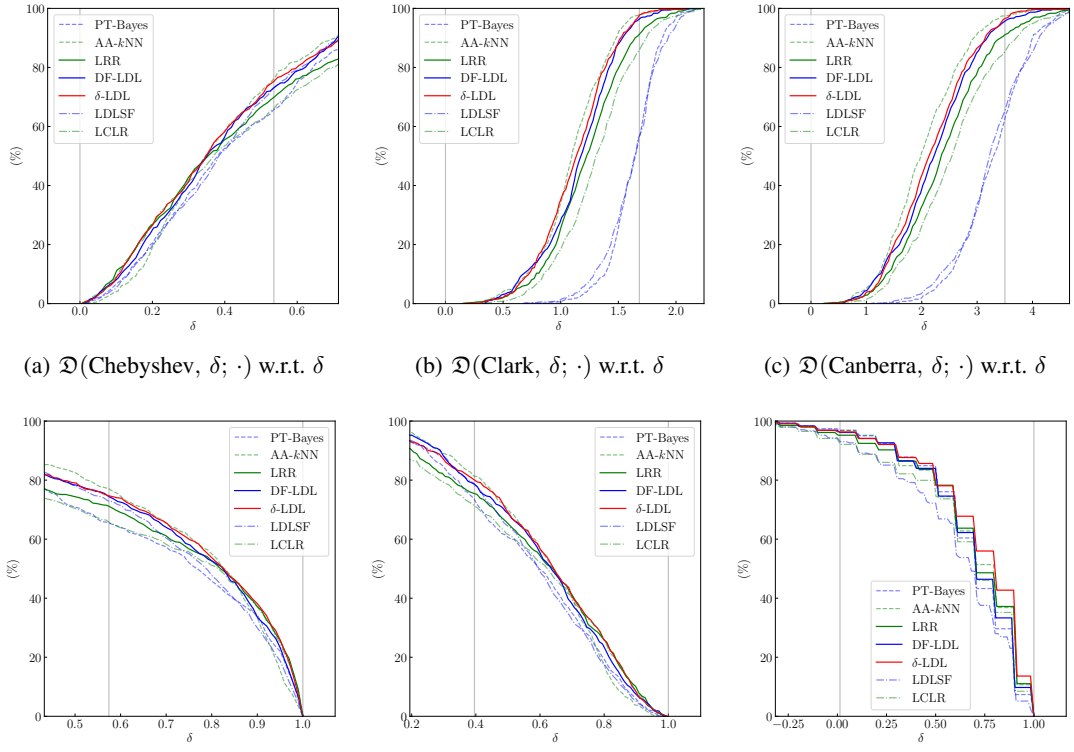
$$\mathbb{E}(u_j) = \frac{\alpha_j}{\sum_{k=1}^{c} \alpha_k}. \tag{23}$$

Since $\boldsymbol{u} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, we have $u_j \sim \mathrm{Beta}(\alpha_j, \sum_{k=1}^{c} \alpha_k - \alpha_j), \forall j \in [c]$. From the properties of the Beta distribution, we have

$$\mathbb{E}(u_j \log u_j) = \frac{\alpha_j}{\sum_{k=1}^{c} \alpha_k} \left(\psi(\alpha_j + 1) - \psi\left(\sum_{k=1}^{c} \alpha_k + 1\right)\right). \tag{24}$$

Combining Equations (22) to (24), the proof is complete. □

B. Remaining experimental results



(a) $\mathfrak{D}$(Chebyshev, δ; $\cdot$) w.r.t. δ (b) $\mathfrak{D}$(Clark, δ; $\cdot$) w.r.t. δ (c) $\mathfrak{D}$(Canberra, δ; $\cdot$) w.r.t. δ

(d) $\mathfrak{D}$(Cosine, δ; $\cdot$) w.r.t. δ (e) $\mathfrak{D}$(Intersection, δ; $\cdot$) w.r.t. δ (f) $\mathfrak{D}$(Spearman, δ; $\cdot$) w.r.t. δ

Figure 5. (Best viewed in color) Visualized results of the analysis on parameters.

Table 3. Experimental results on RAF_ML, Natural_Scene, Music, and Painting, formatted as "mean ± std"

Algorithms	Cheby. ↓	Clark ↓	Can. ↓	KLD ↓	Cosine ↑	Int. ↑	Spear. ↑	μ ↑
				RAF_ML				
SCL	• .3566 ±.006	• 1.5754 ±.012	• 3.4061 ±.031	• .6360 ±.011	• .7080 ±.006	• .5337 ±.004	• .3372 ±.025	• 20.95% ±.006
PT-Bayes	• .3482 ±.022	• 1.7253 ±.033	• 3.7506 ±.106	• .8545 ±.123	• .7346 ±.024	• .5983 ±.023	• .5575 ±.028	• 32.15% ±.039
LDLSF	• .1885 ±.004	• 1.5429 ±.013	• 3.0619 ±.034	• .8256 ±.069	• .9102 ±.005	• .7672 ±.005	• .7713 ±.010	• 48.58% ±.017
AA-kNN	• .2643 ±.005	• 1.4882 ±.014	• 3.0326 ±.036	• .4073 ±.014	• .8221 ±.006	• .6737 ±.006	• .6041 ±.013	• 49.37% ±.012
LDLF	• .1974 ±.008	○ **1.4077** ±.017	• 2.7417 ±.045	• .2444 ±.014	• .9071 ±.006	• .7569 ±.009	• .7494 ±.012	• 68.14% ±.017
DF-LDL	• .1546 ±.005	• 1.4196 ±.015	• 2.7375 ±.036	• .1957 ±.011	• .9273 ±.005	• .8087 ±.005	• .7817 ±.010	• 75.68% ±.010
SA_BFGS	• .1538 ±.005	• 1.4199 ±.015	• 2.7385 ±.036	• .1957 ±.011	• .9277 ±.005	• .8096 ±.005	• .7840 ±.010	• 75.77% ±.010
LCLR	• .1538 ±.005	• 1.4199 ±.015	• 2.7386 ±.036	• .1957 ±.011	• .9277 ±.005	• .8096 ±.005	• .7839 ±.010	• 75.77% ±.010
LDLLC	• .1538 ±.005	1.4178 ±.015	• 2.7335 ±.036	• .1956 ±.011	• .9276 ±.005	• .8096 ±.005	• .7842 ±.010	• 75.79% ±.010
LRR	• .1538 ±.005	• 1.4185 ±.015	• 2.7356 ±.037	• .1957 ±.011	• .9276 ±.005	• .8096 ±.005	• .7841 ±.010	• 75.79% ±.010
DPA	• .1538 ±.005	• 1.4185 ±.016	• 2.7355 ±.037	• .1957 ±.011	• .9276 ±.005	• .8096 ±.005	• .7841 ±.010	• 75.80% ±.010
δ-LDL	**.1527** ±.005	1.4159 ±.016	**2.7267** ±.038	**.1929** ±.011	**.9284** ±.005	**.8108** ±.006	**.7847** ±.010	**76.06%** ±.010
				Natural_Scene				
PT-Bayes	• .7469 ±.019	• 2.9025 ±.008	• 8.5810 ±.033	• 19.3376 ±.971	• .3446 ±.022	• .2416 ±.019	• .3467 ±.024	• 10.19% ±.017
SCL	• .3486 ±.017	• 2.4728 ±.020	• 6.8978 ±.083	• .9959 ±.028	• .6647 ±.008	• .4180 ±.010	• .3368 ±.041	• 25.11% ±.007
LDLSF	• .3185 ±.012	• 2.4948 ±.017	• 6.8440 ±.071	• 2.6764 ±.348	• .7193 ±.012	• .5493 ±.011	• .4718 ±.016	• 35.02% ±.020
LCLR	• .3590 ±.016	○ 2.3688 ±.018	○ 6.5106 ±.072	• 1.1351 ±.093	• .6681 ±.018	• .5258 ±.014	• .4697 ±.017	• 35.33% ±.024
SA_BFGS	• .3323 ±.015	○ 2.3921 ±.019	6.5725 ±.073	• .9239 ±.070	• .7016 ±.016	• .5480 ±.014	• .4898 ±.016	• 39.00% ±.025
DF-LDL	• .3067 ±.013	○ 2.3698 ±.019	○ 6.4633 ±.074	• .7828 ±.041	• .7323 ±.013	• .5670 ±.012	• .5092 ±.016	• 42.55% ±.022
AA-kNN	• .2888 ±.014	○ **1.7886** ±.038	○ **4.1708** ±.110	• 1.0107 ±.081	• .7391 ±.015	.5969 ±.014	• .5319 ±.020	• 42.71% ±.024
LDLLC	• .2994 ±.012	• 2.4214 ±.018	• 6.6168 ±.076	• .7327 ±.035	• .7504 ±.013	• .5718 ±.011	• .5251 ±.015	• 44.30% ±.018
DPA	• .2994 ±.012	• 2.4207 ±.019	• 6.6152 ±.079	• .7327 ±.035	• .7500 ±.012	• .5722 ±.011	• .5252 ±.015	• 44.31% ±.018
LRR	• .2991 ±.012	• 2.4210 ±.018	• 6.6161 ±.076	• .7322 ±.035	• .7499 ±.013	• .5722 ±.011	• .5254 ±.015	• 44.33% ±.018
LDLF	• .2842 ±.012	• 2.4188 ±.018	• 6.5960 ±.074	• .6787 ±.037	• .7706 ±.014	• .5899 ±.012	• .5473 ±.017	• 47.47% ±.021
δ-LDL	**.2760** ±.012	2.4123 ±.017	6.5501 ±.075	**.6586** ±.033	**.7807** ±.012	**.5974** ±.011	**.5553** ±.017	**48.96%** ±.018
				Music				
PT-Bayes	• .1733 ±.026	• 1.1169 ±.101	• 2.8681 ±.292	• .3290 ±.082	• .7818 ±.040	• .6930 ±.034	• .2079 ±.105	• 04.85% ±.026
LDLSF	• .0946 ±.006	• .9498 ±.072	• 2.4003 ±.194	• .6168 ±.306	• .8718 ±.018	• .7678 ±.019	• .3123 ±.072	• 23.89% ±.045
SA_BFGS	• .1020 ±.008	• .8802 ±.060	• 2.2770 ±.172	• .1912 ±.030	• .8665 ±.018	• .7673 ±.018	• .3030 ±.072	• 25.57% ±.047
DF-LDL	• .1022 ±.008	• .8489 ±.057	• 2.1694 ±.160	• .1719 ±.026	• .8726 ±.018	• .7763 ±.018	• .2688 ±.080	• 26.18% ±.045
LCLR	• .1000 ±.008	• .8644 ±.064	• 2.2295 ±.182	• .1821 ±.034	• .8738 ±.019	• .7733 ±.020	• .3673 ±.073	• 26.23% ±.049
DPA	• .0971 ±.007	• .8532 ±.059	• 2.2104 ±.166	• .1758 ±.027	• .8746 ±.017	• .7741 ±.018	• .3224 ±.072	• 27.04% ±.047
LRR	• .0969 ±.007	• .8519 ±.058	• 2.2061 ±.166	• .1751 ±.027	• .8750 ±.017	• .7745 ±.018	• .3214 ±.071	• 27.11% ±.047
LDLLC	• .0968 ±.007	• .8518 ±.057	• 2.2063 ±.160	• .1751 ±.027	• .8751 ±.017	• .7745 ±.017	• .3212 ±.070	• 27.12% ±.046
SCL	• .0776 ±.004	• .7676 ±.050	• 1.9339 ±.131	• .1157 ±.013	• .9150 ±.008	• .8065 ±.012	• .4544 ±.055	• 32.12% ±.028
LDLF	• .0775 ±.005	• .7532 ±.054	• 1.9050 ±.140	• .1134 ±.014	• .9161 ±.009	• .8096 ±.014	.4538 ±.078	• 34.01% ±.045
AA-kNN	• .0784 ±.006	• .7499 ±.061	• 1.9309 ±.172	• .1258 ±.021	• .9067 ±.015	• .8051 ±.018	• .3989 ±.081	• 36.75% ±.045
δ-LDL	**.0733** ±.005	**.7185** ±.055	**1.8221** ±.150	**.1061** ±.016	**.9218** ±.011	**.8186** ±.015	**.4942** ±.058	**39.35%** ±.036
				Painting				
PT-Bayes	• .7070 ±.056	• 2.5747 ±.056	• 7.0292 ±.221	• 7.4050 ±1.412	• .3337 ±.055	• .2275 ±.044	• .1484 ±.075	• 00.07% ±.003
LDLSF	• .2853 ±.023	• 1.8290 ±.067	• 4.3005 ±.204	• 4.1107 ±1.205	• .6565 ±.033	• .5507 ±.025	.2047 ±.067	• 11.89% ±.040
LCLR	• .2979 ±.028	• 1.8632 ±.070	• 4.5308 ±.229	• .9065 ±.215	• .6579 ±.036	• .5538 ±.028	.2454 ±.066	• 16.23% ±.045
SA_BFGS	• .2965 ±.028	• 1.8559 ±.073	• 4.5050 ±.237	• .8997 ±.233	• .6596 ±.036	• .5547 ±.028	.2363 ±.069	• 16.37% ±.044
DF-LDL	• .2714 ±.021	• 1.8051 ±.064	• 4.3184 ±.202	• .6950 ±.096	• .6772 ±.032	.5746 ±.023	.2310 ±.073	• 17.13% ±.044
SCL	• .2579 ±.019	○ **1.6915** ±.067	**3.9339** ±.193	• .5574 ±.042	• .7223 ±.017	• .5940 ±.019	• .1987 ±.078	• 19.36% ±.024
AA-kNN	.2572 ±.021	• 1.7553 ±.060	• 4.1427 ±.182	• .7100 ±.161	• .6987 ±.027	• .5888 ±.021	.2241 ±.072	• 19.76% ±.041
DPA	.2585 ±.020	• 1.7561 ±.069	• 4.1452 ±.215	• .5999 ±.067	.7116 ±.028	.5951 ±.022	.2780 ±.069	• 20.48% ±.041
LRR	.2587 ±.019	• 1.7556 ±.068	• 4.1422 ±.212	• .5988 ±.064	.7118 ±.027	.5954 ±.022	.2782 ±.068	• 20.48% ±.040
LDLLC	.2586 ±.020	• 1.7560 ±.069	• 4.1438 ±.216	• .5994 ±.066	.7120 ±.027	.5953 ±.022	.2774 ±.068	• 20.48% ±.041
LDLF	.2485 ±.019	1.7059 ±.066	3.9700 ±.200	.5343 ±.046	.7336 ±.023	.6075 ±.021	.3054 ±.088	22.54% ±.036
δ-LDL	**.2484** ±.020	1.7019 ±.067	3.9520 ±.201	**.5288** ±.048	**.7377** ±.023	**.6102** ±.022	**.3146** ±.092	**23.05%** ±.037