# Label Distribution Learning using the Squared Neural Family on the Probability Simplex

**Daokun Zhang**[1,2] **Russell Tsuchida**[2] **Dino Sejdinovic**[3]

[1]School of Computer Science, University of Nottingham Ningbo China
[2]Department of Data Science & AI, Monash University
[3]School of Computer and Mathematical Sciences, The University of Adelaide

## Abstract

Label distribution learning (LDL) provides a framework wherein a distribution over categories rather than a single category is predicted, with the aim of addressing ambiguity in labeled data. Existing research on LDL mainly focuses on the task of point estimation, i.e., finding an optimal distribution in the probability simplex conditioned on the given sample. In this paper, we propose a novel label distribution learning model SNEFY-LDL, which estimates a probability distribution of all possible label distributions over the simplex, by unleashing the expressive power of the recently introduced Squared Neural Family (SNEFY), a new class of tractable probability models. As a way to summarize the fitted model, we derive the closed-form label distribution mean, variance and covariance conditioned on the given sample, which can be used to predict the ground-truth label distributions, construct label distribution confidence intervals, and measure the correlations between different labels. Moreover, more information about the label distribution prediction uncertainties can be acquired from the modeled probability density function. Extensive experiments on conformal prediction, active learning and ensemble learning are conducted, verifying SNEFY-LDL's great effectiveness in LDL uncertainty quantification. The source code of this paper is available at `https://github.com/daokunzhang/SNEFY-LDL`.

## 1 INTRODUCTION

Label distribution learning (LDL) is a technique which handles ambiguity in multi-class classification, by utilizing simplex-valued rather than categorical-valued labels in training data. Unlike traditional multi-class and multi-label learning paradigms, which assign a deterministic label prediction to instances, LDL corresponds to the question "*How well does each of the labels describe an instance?*", by using a discrete probability distribution to characterize each label's composition ratio in jointly describing the given instance. For example, when we predict the functionality of a district in a city, we might predict a result such as: the district has 20% functionality for business, 40% functionality for entertainment, and 40% functionality for education.

Many LDL algorithms have been proposed to directly predict label distribution vectors from instance features, by adapting machine learning algorithms designed for "hard" label prediction to the "soft" label prediction setting. Though a discrete distribution among candidate labels is predicted, existing LDL algorithms still operate at the level of point estimation, i.e., they search for a single point on a probability simplex (the set of all possible label distributions) for each given instance. The point estimation paradigm is particularly susceptible to data uncertainty and inexact mappings between instances and labels, due to the inherent complexity of the data collection and generation processes. Therefore, modeling the probability distribution of label distribution vectors, i.e., the probability distribution supported on the probability simplex, is an important step towards trustworthy LDL. An additional bonus of the distribution modeling is the ability to quantify the prediction reliability and uncertainties, which not only facilitates reliable model deployment in real-world safety critical applications, but is also essential to various reliability/uncertainty-aware tasks, like pseudo labeling, active learning, and ensemble learning.

**Contributions.** In this paper, we propose a novel LDL framework, SNEFY-LDL, by unleashing the probability modeling power of the recently introduced Squared Neural Family (SNEFY) [Tsuchida et al., 2023], a new class of tractable probability models. By restricting the support set of SNEFY to a probability simplex, SNEFY-LDL constructs an expressive multimodal distribution modeling of the label distribution vector conditioned the given sample. The conditional distribution model has a closed-form normal-

izing constant, guaranteeing computational tractability. In this way, model parameters can be learned efficiently by maximizing the conditional likelihoods of training samples with stochastic gradient descent. As a way to summarize the fitted model, we derive the closed-form label distribution mean, variance and covariance conditioned on the given sample, which can be used to predict the ground-truth label distribution, construct label distribution confidence intervals, and measure the correlations between different labels. However, the fitted model is not limited to these usecases, and the probability density values can be used to directly evaluate the reliability of label distribution predictions.

We conduct extensive experiments on label distribution conformal prediction, active learning and ensemble learning to verify the efficacy of SNEFY-LDL in quantifying prediction uncertainties. For the task of conformal prediction, we use the SNEFY-LDL's closed-form conditional mean and variance of label distribution predictions to construct a confidence interval for each label's composition ratio in describing the given instance and calibrate the confidence interval through conformal prediction [Angelopoulos and Bates, 2021]. Experimental results show that the confidence intervals constructed by the SNEFY-LDL model have greater adaptivity than the confidence intervals constructed by the naive Dirichlet distribution. The max-entropy principle is used to achieve active learning with the estimated SNEFY-LDL entropy, i.e., select the most informative unlabeled samples with the largest entropy values, query their labels and augment training samples, to attain the largest performance gain of the re-trained LDL model. Experimental results show that the max-entropy principle achieves significantly better active learning performance than the representativeness based active learning baselines. The experiments on ensemble learning demonstrate that SNEFY-LDL gives a further usecase for the fitted probabilistic model, as it provides an intelligent mechanism for weighting base learners, significantly outperforming the uniform weighting strategy.

## 2 RELATED WORK

LDL is first proposed by Geng et al. [2013] to solve the facial age estimation problem. Since then, a series of LDL algorithms have been developed, which are mainly in three categories: Problem Transformation (PT), Algorithm Adaptation (AA) and Specialized Algorithms (SA) [Geng, 2016].

**Problem Transformation.** PT [Geng, 2016] transforms the LDL problem into the single-label classification problem, by decomposing each training sample assigned with a label distribution into a set of duplicate training samples. Each of them is assigned with a different label and accounts for a partial sample in proportion to the label probability value, and is then used to train the single-label classifiers. The label likelihoods predicted by the single-label classifiers are then aggregated to form the final prediction of label distributions. PT-Bayes [Geng, 2016] and PT-SVM [Geng and Hou, 2015] transform the LDL problem into the single-label multi-class classification problem and respectively employ Bayes and SVM as the single-label classifiers. DF-LDL [González et al., 2021a] decomposes the label distribution prediction task into a number of one-versus-one binary classification tasks, and fuses the binary classification likelihoods to form the final label distribution predictions.

**Algorithm Adaptation.** AA [Geng, 2016] adapts traditional single-label classification models into the LDL setting, by leveraging the models' compatibility in outputting a soft label distribution vector. Derived from the K Nearest Neighbor (KNN) algorithm [Wu et al., 2008], AA-KNN [Geng, 2016] predicts samples' label distributions by averaging the label distributions of their $k$ nearest neighbors in feature space. AA-BP [Geng, 2016] constructs a three-layer neural network and adopts the softmax function as the activation of the output layer, making the neural network naturally produce a label distribution for each example. The neural network is trained by minimizing the sum of squared errors between the model-output label distributions and the ground-truth label distributions.

**Specialized Algorithms.** SA [Geng, 2016] designs algorithms from scratch to directly solve the LDL problem. SA-IIS [Geng et al., 2013] and SA-BFGS [Geng, 2016] use the maximum entropy model to parameterize label distributions. They are trained by minimizing the Kullback-Leibler (KL) divergence between the model-output and ground-truth label distributions, where Improved Iterative Scaling (IIS) [Malouf, 2002] and BFGS [Nocedal and Wright, 1999] are respectively leveraged by SA-IIS and SA-BFGS as optimizers. CPNN [Geng et al., 2013] uses a neural network to parameterize the joint probability distribution between sample features and labels following Modha's probability distribution formulation [Modha and Fainman, 1994]. BCPNN [Yang et al., 2017] and ACPNN [Yang et al., 2017] then improve on CPNN through leveraging binary label encoding and augmenting training samples respectively. LDLF [Shen et al., 2017] employs differentiable decision trees [Kontschieder et al., 2015] to model label distributions and KL divergence is used to design the learning objective. LDL-SCL [Jia et al., 2021] forces the label distributions of samples located closely in feature space to be similar to each other. LDL-LRR [Jia et al., 2023b] and LDL-DPA [Jia et al., 2023a] maintain the relative importance ranking between different labels in label distribution modeling, by penalizing a label importance ranking loss in their learning objectives.

**Extensions.** In addition, LDL has been extended to other tasks, like label enhancement [Xu et al., 2019b, 2020, Zheng et al., 2023], multi-class classification [Wang and Geng, 2019, 2021b,a], learning with incomplete supervision [Xu and Zhou, 2017], oversampling [González et al., 2021b], ordinal LDL [Wen et al., 2023], semi-supervised learning [Xie et al., 2023], and label calibration [He et al., 2024]. Further-

more, LDL has been applied to solve numerous real-world problems, including facial age estimation [Geng et al., 2013], facial emotion recognition [Chen et al., 2020], head pose estimation [Xu et al., 2019a], crowd opinion prediction [Geng and Hou, 2015], emphasis selection [Shirani et al., 2019], lesion counting [Wu et al., 2019], and urban functionality prediction [Huang et al., 2023]. It is worth noting that there is a related topic in the classical statistics literature, termed *compositional data analysis* [Greenacre, 2021], with a broad range of applications including geochemistry (e.g. labels correspond to mineral compositions) and ecology (e.g. labels correspond to relative abundance of species).

However, existing LDL algorithms mainly fall into the regime of point estimation. They discover an optimal discrete label distribution in the probability simplex with regard to a predefined learning objective, and do not provide the information about how prominent the optimal label distribution is, compared with the remaining distribution candidates. In this paper, we aim to model the distribution of label distributions, with the expectation that we can provide a promising mechanism for LDL uncertainty quantification.

## 3 PROBLEM DEFINITION

Assume we are given a set of $N$ i.i.d training samples $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ with each sample $\boldsymbol{x} \in \mathcal{X}$ located in the $d$-dimensional Euclidean space $\mathbb{R}^d$. In addition, each sample $\boldsymbol{x} \in \mathcal{X}$ is described by a $L$-dimensional label distribution vector $\boldsymbol{\ell}_{\boldsymbol{x}} \in \mathbb{R}^L$ that takes values in the $(L-1)$-simplex $\Delta^{L-1}$, corresponding to a set of $L$ given labels $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\}$. The $l$-th entry $\boldsymbol{\ell}_{\boldsymbol{x}}^{y_l}$ of the label distribution vector $\boldsymbol{\ell}_{\boldsymbol{x}}$ corresponds to the composition ratio of the $l$-th label $y_l$ in describing $\boldsymbol{x}$, satisfying the constraint that $\sum_{l=1}^{L} \boldsymbol{\ell}_{\boldsymbol{x}}^{y_l} = 1$.

With the label distribution observations of training samples, our objective is to model the probability distribution of the label distribution vector $\boldsymbol{\ell} \in \Delta^{L-1}$ conditioned on any input sample $\boldsymbol{x} \in \mathbb{R}^d$, $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x})$.

## 4 PRELIMINARIES ON SNEFY

Given a measure space $(\Omega, \mathcal{F}, \mu)$ with set $\Omega$, sigma algebra $\mathcal{F}$, and nonnegative measure $\mu$, SNEFY defines a probability distribution P on some support $\mathcal{Z} \in \mathcal{F}$ to be proportional to the evaluation of the squared 2-norm of a neural network $\boldsymbol{f}$:

$$\begin{aligned} \mathrm{P}(d\boldsymbol{z}; \boldsymbol{V}, \boldsymbol{\Theta}) &\triangleq \frac{\|\boldsymbol{f}(\boldsymbol{t}(\boldsymbol{z}); \boldsymbol{V}, \boldsymbol{\Theta})\|_2^2 \mu(d\boldsymbol{z})}{\int_{\mathcal{Z}} \|\boldsymbol{f}(\boldsymbol{t}(\boldsymbol{z}); \boldsymbol{V}, \boldsymbol{\Theta})\|_2^2 \mu(d\boldsymbol{z})}, \\ \boldsymbol{f}(\boldsymbol{t}(\boldsymbol{z}); \boldsymbol{V}, \boldsymbol{\Theta}) &= \boldsymbol{V}\sigma(\boldsymbol{W}\boldsymbol{t}(\boldsymbol{z}) + \boldsymbol{b}),\ \boldsymbol{\Theta} = (\boldsymbol{W}, \boldsymbol{b}), \end{aligned} \tag{1}$$

where $\boldsymbol{t}(\cdot) : \mathcal{Z} \to \mathbb{R}^D$ is the sufficient statistic, $\sigma$ is the activation function, $\boldsymbol{W} \in \mathbb{R}^{n \times D}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are respectively the weight matrix and bias vector at the hidden layer of neural network $\boldsymbol{f}$, and $\boldsymbol{V} \in \mathbb{R}^{m \times n}$ are $\boldsymbol{f}$'s readout parameters. $\boldsymbol{\Theta} = (\boldsymbol{W}, \boldsymbol{b}) \in \mathbb{R}^{n \times (D+1)}$ is the concatenation of $\boldsymbol{W}$ and $\boldsymbol{b}$ and its $i$th row is denoted as $\boldsymbol{\theta}_i = (\boldsymbol{w}_i, b_i) \in \mathbb{R}^{D+1}$, where $\boldsymbol{w}_i \in \mathbb{R}^D$ is the $i$th row of $\boldsymbol{W}$ and $b_i$ is the $i$th element of $\boldsymbol{b}$.

The distribution $\mathrm{P}(d\boldsymbol{z}; \boldsymbol{V}, \boldsymbol{\Theta})$ in Eq. (1) admits a more concise formulation

$$\begin{aligned} \mathrm{P}(d\boldsymbol{z}; \boldsymbol{V}, \boldsymbol{\Theta}) &= \frac{\mathrm{Tr}[\boldsymbol{V}^\top \boldsymbol{V} \widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{z})]}{\mathrm{Tr}[\boldsymbol{V}^\top \boldsymbol{V} \boldsymbol{K}_{\boldsymbol{\Theta}}]} \mu(d\boldsymbol{z}), \\ &= \frac{\mathrm{vec}(\boldsymbol{V}^\top \boldsymbol{V})^\top \mathrm{vec}(\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{z}))}{\mathrm{vec}(\boldsymbol{V}^\top \boldsymbol{V})^\top \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}})} \mu(d\boldsymbol{z}), \end{aligned} \tag{2}$$

where $\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{z}) \in \mathbb{R}^{n \times n}$ is a positive semidefinite (PSD) matrix, whose $ij$th element is a kernel function of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$:

$$\tilde{\boldsymbol{k}}_{\sigma,\boldsymbol{t}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{z}) = \sigma(\boldsymbol{w}_i^\top \boldsymbol{t}(\boldsymbol{z}) + b_i)\sigma(\boldsymbol{w}_j^\top \boldsymbol{t}(\boldsymbol{z}) + b_j), \tag{3}$$

while $\boldsymbol{K}_{\boldsymbol{\Theta}}$ is the elementwise integral of $\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{z})$, preserving the PSD property, with its $ij$th entry formulated as another kernel function of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$:

$$\boldsymbol{k}_{\sigma,\boldsymbol{t},\mu}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \int_{\mathcal{Z}} \tilde{\boldsymbol{k}}_{\sigma,\boldsymbol{t}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{z}) \mu(d\boldsymbol{z}). \tag{4}$$

Under varying choices of the activation function $\sigma$, sufficient statistic $\boldsymbol{t}$, and the base measure $\mu$, $\boldsymbol{k}_{\sigma,\boldsymbol{t},\mu}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ is able to be computed in closed form (see Table 1 of [Tsuchida et al., 2023]) in $\mathcal{O}(D)$. This makes SNEFY a tractable probability distribution model, with great expressivity and computational efficiency.

SNEFY also enjoys a closed-form formulation for conditional distributions, under mild conditions.

**Theorem 1.** *[Tsuchida et al., 2023] Let* $\mathtt{z} = (\mathtt{z}_1, \mathtt{z}_2)$ *jointly follow a SNEFY distribution, with support set* $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2$*, sufficient statistic* $\boldsymbol{t}$*, activation function* $\sigma$*, base measure* $\mu$*, as well as parameters* $\boldsymbol{V}$ *and* $\boldsymbol{\Theta} = ([\boldsymbol{W}_1, \boldsymbol{W}_2], \boldsymbol{b})$*. Assume that* $\mu(d\boldsymbol{z}) = \mu_1(d\boldsymbol{z}_1)\mu_2(d\boldsymbol{z}_2)$ *and* $\boldsymbol{t}(\boldsymbol{z}) = (\boldsymbol{t}_1(\boldsymbol{z}_1), \boldsymbol{t}_2(\boldsymbol{z}_2))$*. Then the conditional distribution of* $\mathtt{z}_1$ *given* $\mathtt{z}_2 = \boldsymbol{z}_2$ *is also a SNEFY distribution with support set* $\mathcal{Z}_1$*, sufficient statistic* $\boldsymbol{t}_1$*, activation function* $\sigma$*, base measure* $\mu_1$*, as well as parameters* $\boldsymbol{V}$ *and* $\boldsymbol{\Theta}_{1|2} = (\boldsymbol{W}_1, \boldsymbol{W}_2\boldsymbol{t}_2(\boldsymbol{z}_2) + \boldsymbol{b})$*.*

## 5 SNEFY-LDL

SNEFY provides an effective way to model the conditional distribution of the label distribution vector $\boldsymbol{\ell} \in \Delta^{L-1}$. We can assume that the concatenation of label distribution vector $\boldsymbol{\ell} \in \Delta^{L-1}$ and its conditioning sample $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{z} = (\boldsymbol{\ell}, \boldsymbol{x})$, follows a joint SNEFY distribution, with support set $\mathcal{Z} = \Delta^{L-1} \times \mathbb{R}^d$, sufficient statistic $\boldsymbol{t}(\boldsymbol{z}) = (\boldsymbol{t}_1(\boldsymbol{\ell}), \boldsymbol{t}_2(\boldsymbol{x})) : \mathcal{Z} \to \mathbb{R}^{D_1+D_2}$ composed of $\boldsymbol{t}_1(\cdot) : \Delta^{L-1} \to \mathbb{R}^{D_1}$ and $\boldsymbol{t}_2(\cdot) : \mathbb{R}^d \to \mathbb{R}^{D_2}$, activation function $\sigma$, base measure $\mu(\boldsymbol{z}) = \mu_1(\boldsymbol{\ell})\mu_2(\boldsymbol{x})$, as well as parameters $\boldsymbol{V} \in \mathbb{R}^{m \times n}$ and

$\boldsymbol{\Theta} = ([\boldsymbol{W}_1, \boldsymbol{W}_2], \boldsymbol{b}) \in \mathbb{R}^{n\times(D_1+D_2+1)}$. Following **Theorem 1**, given sample $\boldsymbol{x}$, the conditional distribution of its label distribution vector $\boldsymbol{\ell}$ is a SNEFY distribution with support set $\Delta^{L-1}$, sufficient statistic $\boldsymbol{t}_1$, activation function $\sigma$, base measure $\mu_1$, as well as parameters $\boldsymbol{V}$ and $\boldsymbol{\Theta}_{1|2} = (\boldsymbol{W}_1, \boldsymbol{W}_2\boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{b})$. The conditional distribution is

$$\begin{aligned}\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta}) &\triangleq \frac{\|\boldsymbol{f}(\boldsymbol{t}_1(\boldsymbol{\ell}),\boldsymbol{t}_2(\boldsymbol{x});\boldsymbol{V},\boldsymbol{\Theta})\|_2^2\mu_1(d\boldsymbol{\ell})}{\int_{\Delta^{L-1}}\|\boldsymbol{f}(\boldsymbol{t}_1(\boldsymbol{\ell}),\boldsymbol{t}_2(\boldsymbol{x});\boldsymbol{V},\boldsymbol{\Theta})\|_2^2\mu_1(d\boldsymbol{\ell})},\\ \boldsymbol{f}\big(\boldsymbol{t}_1(\boldsymbol{\ell}),\boldsymbol{t}_2(\boldsymbol{x});\boldsymbol{V},\boldsymbol{\Theta}\big) &= \boldsymbol{V}\sigma\big(\boldsymbol{W}_1\boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{W}_2\boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{b}\big).\end{aligned} \tag{5}$$

Following Eq. (2), the distribution can be reformulated as

$$\begin{aligned}\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta}) &= \frac{\mathrm{Tr}[\boldsymbol{V}^\top\boldsymbol{V}\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{\ell},\boldsymbol{x})]}{\mathrm{Tr}[\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})]}\mu_1(d\boldsymbol{\ell}),\\ &= \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{\ell},\boldsymbol{x}))}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}\mu_1(d\boldsymbol{\ell}),\end{aligned} \tag{6}$$

where $\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{\ell},\boldsymbol{x}) \in \mathbb{R}^{n\times n}$ is a PSD matrix, with its $ij$th element being a kernel function of $\boldsymbol{\theta}_i = (\boldsymbol{w}_{1i}, \boldsymbol{w}_{2i}, b_i) \in \mathbb{R}^{D_1+D_2+1}$ and $\boldsymbol{\theta}_j = (\boldsymbol{w}_{1j}, \boldsymbol{w}_{2j}, b_j) \in \mathbb{R}^{D_1+D_2+1}$:

$$\begin{aligned}\tilde{\boldsymbol{k}}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{\ell},\boldsymbol{x}) =& \sigma(\boldsymbol{w}_{1i}^\top\boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{w}_{2i}^\top\boldsymbol{t}_2(\boldsymbol{x}) + b_i)\cdot\\ & \sigma(\boldsymbol{w}_{1j}^\top\boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{w}_{2j}^\top\boldsymbol{t}_2(\boldsymbol{x}) + b_j),\end{aligned} \tag{7}$$

where $\boldsymbol{w}_{1i} \in \mathbb{R}^{D_1}$ and $\boldsymbol{w}_{2i} \in \mathbb{R}^{D_2}$ are respectively the $i$th row of matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, and $b_i$ is the $i$th element of the bias vector $\boldsymbol{b}$. Then $\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})$ is the elementwise integral of $\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{\ell},\boldsymbol{x})$, preserving the PSD property, with its $ij$th entry formulated as another kernel function of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$:

$$\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x}) = \int_{\Delta^{L-1}}\tilde{\boldsymbol{k}}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{\ell},\boldsymbol{x})\mu_1(d\boldsymbol{\ell}). \tag{8}$$

By choosing the activation function $\sigma$, sufficient statistic $\boldsymbol{t}_1$ and the base measure $\mu_1$ carefully, the kernel function $\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x})$ admits a closed form, which guarantees that the conditional distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ is tractable. In particular, we have the following theorem:

**Theorem 2.** *Let $\boldsymbol{t}_1(\boldsymbol{\ell}) = (\log\boldsymbol{\ell}^{y_1}, \log\boldsymbol{\ell}^{y_2}, \cdots, \log\boldsymbol{\ell}^{y_L}) : \Delta^{L-1} \to \mathbb{R}^L$ by setting $D_1 = L$, the activation function $\sigma$ be the exponential function* exp*, the base measure $\mu_1(d\boldsymbol{\ell}) = d\boldsymbol{\ell}$ be the Lebesgue measure. Under the condition that $\boldsymbol{W}_1 > -1/2$ elementwise, the kernel function $\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x})$ admits a closed form:*

$$\begin{aligned}\boldsymbol{k}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x}) =& \exp(\boldsymbol{w}_{2i}^\top\boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{w}_{2j}^\top\boldsymbol{t}_2(\boldsymbol{x}) + b_i + b_j)\cdot\\ & \frac{\prod_{l=1}^L\Gamma(1+w_{1il}+w_{1jl})}{\Gamma\big(L+\sum_{l=1}^L(w_{1il}+w_{1jl})\big)},\end{aligned} \tag{9}$$

*where $w_{1il}$ is the $il$-th element of matrix $\boldsymbol{W}_1$ and $\Gamma(\cdot)$ is the gamma function.*

The proof is provided in the Appendix. With the closed-form kernel function in Eq. (9), we can construct the conditional SNEFY distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ in the form of Eq. (6). This model provides us with the freedom to choose any sufficient statistic $\boldsymbol{t}_2(\cdot)$ that is used to transform the input sample $\boldsymbol{x}$ from the original $d$-dimensional space to the latent $D_2$-dimensional space. To capture the non-linearity between input samples and their label distributions, deep neural networks can be leveraged to construct $\boldsymbol{t}_2(\cdot)$. The input can also be extended beyond the vector-format samples to data with special structures, like images, texts and graphs, where we can respectively leverage Convolutional Neural Networks (CNNs) [Venkatesan and Li, 2017], Transformers [Vaswani et al., 2017] and Graph Neural Networks (GNNs) [Kipf and Welling, 2017] to construct $\boldsymbol{t}_2(\cdot)$ for end-to-end learning.

The conditional distribution formulation $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ also provides a closed form of mean, variance and covariance of each label's composition ratio in describing the conditioning sample $\boldsymbol{x}$. About this, we have the following theorem:

**Theorem 3.** *Assuming the label distribution vector $\boldsymbol{\ell}$ follows the SNEFY conditional distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ in Eq. (6) with the kernel function $\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x})$ given in Eq. (9), under the setting that $\boldsymbol{t}_1(\boldsymbol{\ell}) = (\log\boldsymbol{\ell}^{y_1}, \log\boldsymbol{\ell}^{y_2}, \cdots, \log\boldsymbol{\ell}^{y_L})$, $\sigma = \exp$, and $\mu_1(d\boldsymbol{\ell}) = d\boldsymbol{\ell}$, as well as the constraint that $\boldsymbol{W}_1 > -1/2$ elementwise, for the $r$th label's composition ratio, $\boldsymbol{\ell}^{y_r}$, we have its conditional mean $\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ as*

$$\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] = \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})\circ\boldsymbol{F}^{y_r})}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}, \tag{10}$$

*where $\circ$ denotes Hadamard product, and $\boldsymbol{F}^{y_r}$ is a $n\times n$ matrix, whose $ij$th entry is*

$$F_{ij}^{y_r} = \frac{1+w_{1ir}+w_{1jr}}{L+\sum_{l=1}^L(w_{1il}+w_{1jl})}. \tag{11}$$

*The conditional variance of $\boldsymbol{\ell}^{y_r}$, $\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$, is*

$$\begin{aligned}\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] =& \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})\circ\boldsymbol{G}^{y_r})}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}\\ & - \mathrm{E}^2[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}],\end{aligned} \tag{12}$$

*where $\boldsymbol{G}^{y_r}$ is a $n\times n$ matrix, with its $ij$th element being*

$$G_{ij}^{y_r} = \frac{(1+w_{1ir}+w_{1jr})(2+w_{1ir}+w_{1jr})}{[L+\sum_{l=1}^L(w_{1il}+w_{1jl})][1+L+\sum_{l=1}^L(w_{1il}+w_{1jl})]}. \tag{13}$$

*For two different labels $y_r$ and $y_s$, with $y_r \neq y_s$, the conditional covariance of $\boldsymbol{\ell}^{y_r}$ and $\boldsymbol{\ell}^{y_s}$, $\mathrm{Cov}[\boldsymbol{\ell}^{y_r}, \boldsymbol{\ell}^{y_s}|\boldsymbol{x}]$, is*

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{\ell}^{y_r}, \boldsymbol{\ell}^{y_s}|\boldsymbol{x}] =& \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})\circ\boldsymbol{H}^{y_r,y_s})}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}\\ & - \mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]\cdot\mathrm{E}[\boldsymbol{\ell}^{y_s}|\boldsymbol{x}],\end{aligned} \tag{14}$$

*where $\boldsymbol{H}^{y_r,y_s}$ is a $n \times n$ matrix , with its $ij$th element being*

$$H_{ij}^{y_r,y_s} = \frac{(1+w_{1ir}+w_{1jr})(1+w_{1is}+w_{1js})}{[L+\sum_{l=1}^{L}(w_{1il}+w_{1jl})][1+L+\sum_{l=1}^{L}(w_{1il}+w_{1jl})]}. \tag{15}$$

The proof is provided in the Appendix. Given the fitted distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$, the mean $\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ can be used to predict the unknown label distribution as the expectation over all values in the simplex. We can use the variance $\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ to quantify label distribution prediction uncertainties. We can also use $\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ with $\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ to construct confidence intervals for label distribution predictions by applying Chebyshev's inequality [Grimmett and Stirzaker, 2020]. The covariance $\mathrm{Cov}[\boldsymbol{\ell}^{y_r},\boldsymbol{\ell}^{y_s}|\boldsymbol{x}]$ is helpful for us to understand the correlations between two different labels. More importantly, all the statistics are conditioned on the given sample $\boldsymbol{x}$, guiding us to make instance-wise decisions.

The conditional distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ relies on the parameters $\boldsymbol{V}$ and $\boldsymbol{\Theta}$, as well as the neural network parameters for constructing $\boldsymbol{t}_2$ (we also use $\boldsymbol{t}_2$ to denote the parameters without confusion). We train the model with maximum likelihood estimation (MLE), by minimizing the negative log conditional likelihoods on training samples:

$$\min_{\boldsymbol{V},\boldsymbol{\Theta},\boldsymbol{t}_2} -\sum_{\boldsymbol{x}'\in\mathcal{X}} \log \frac{\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})}{d\boldsymbol{\ell}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}',\boldsymbol{\ell}=\boldsymbol{\ell}_{\boldsymbol{x}'}}. \tag{16}$$

There are numerous metrics to measure the consistency between two label distributions, like Chebyshev distance, Kullback-Leibler divergence and Cosine coefficient [Geng, 2016]. Instead of optimizing these metrics, the MLE based learning objective in Eq. (16) provides an alternative way to train the LDL model. The fitted distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ can be applied to various downstream tasks for quantifying the uncertainty of label distribution predictions.

**Algorithm Description and Time Complexity.** We train the SNEFY-LDL model with stochastic gradient descent. The training procedure is shown in Algorithm 1. The model parameters $\boldsymbol{V}$, $\boldsymbol{\Theta}$ and $\boldsymbol{t}_2$ are first initialized by random numbers. We then iteratively select a batch of training samples, calculate the batched likelihoods with Eq. (6), and update parameters $\boldsymbol{V}$, $\boldsymbol{\Theta}$ and $\boldsymbol{t}_2$ by descending them along the gradient of batched negative log likelihoods. Taking the epoch number as a constant and assuming the latent layers of $\boldsymbol{t}_2$ have neurons in the same scale as the neuron number in the last layer $D_2$, the time complexity of Algorithm 1 is $\mathcal{O}(N(mn^2+Ln^2+dD_2+D_2^2))$, which is linear to the number of training samples $N$, making the algorithm able to scale to large datasets. For any sample $\boldsymbol{x}$ with an unknown label distribution, its label distribution mean and variance can be computed in time complexity $\mathcal{O}(mn^2+Ln^2+dD_2+D_2^2)$ using the closed-form formulations in Eq. (10) and Eq. (12).

**Algorithm 1** Training SNEFY-LDL

**Input**: Training set $\{(\boldsymbol{x}_1,\boldsymbol{\ell}_{\boldsymbol{x}_1}),(\boldsymbol{x}_2,\boldsymbol{\ell}_{\boldsymbol{x}_2}),\cdots,(\boldsymbol{x}_N,\boldsymbol{\ell}_{\boldsymbol{x}_N})\}$.
**Parameter**: $(\boldsymbol{V},\boldsymbol{\Theta},\boldsymbol{t}_2)$.
**Output**: Optimized $(\boldsymbol{V},\boldsymbol{\Theta},\boldsymbol{t}_2)$.

```
(V, Θ, t_2) ← random initialization;
while epoch number does not expire do
    B ← randomly split training set into batches;
    for each batch in B do
        Calculate batched K_Θ(x) with Eq. (9);
        Calculate batched likelihoods with Eq. (6);
        (V, Θ, t_2) ← update by descending along the gradient of batched negative log likelihoods;
    end for
end while
return optimized (V, Θ, t_2).
```

| Dataset | #Examples | #Features | #Labels |
|---|---|---|---|
| Movie | 7,755 | 1,869 | 5 |
| Natural Scene | 2,000 | 294 | 9 |
| SBU_3DFE | 2,500 | 243 | 6 |
| SJAFFE | 213 | 243 | 6 |

Table 1: Summary of the four benchmark datasets.

# 6 EXPERIMENTS

We conduct extensive experiments on conformal prediction, active learning and ensemble learning to verify SNEFY-LDL's ability in LDL uncertainty quantification.

**Benchmark Datasets.** We use four datasets [Geng, 2016] to benchmark our experiments, including the *Movie* dataset containing label distributions on five movie rating scales, the *Natural Scene* dataset with label distributions constructed by inconsistent multi-label ranking on natural scene images, the facial expression datasets *SBU_3DFE* and *SJAFFE* with label distributions on six emotions. The statistics of the four benchmark datasets are summarized in Table 1.

**Implementation Details.** When implementing SNEFY-LDL, $n$ and $m$ are respectively set to 64 and 32, $D_2$ is set as equal to $n$ and a one-layer neural network with ReLU activation is used to construct $\boldsymbol{t}_2$. The model is trained for 100 epochs with batch size 64 for conformal prediction and batch size 16 for active learning and ensemble learning. Weight clipping [Arjovsky et al., 2017] is used to control $\boldsymbol{W}_1 > -1/2$ elementwise after each parameter update.

## 6.1 CONFORMAL PREDICTION

As shown in **Theorem 3**, with the trained SNEFY-LDL model, given a new sample $\boldsymbol{x}$, we can get the closed-form conditional mean for the $r$th label's composition ratio as $\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ in Eq. (10) and the closed-form conditional vari-

| Class Id | bin size = 2 | | bin size = 4 | | bin size = 8 | |
|---|---|---|---|---|---|---|
| | Dirichlet | SNEFY-LDL | Dirichlet | SNEFY-LDL | Dirichlet | SNEFY-LDL |
| 1 | 0.8577±0.0310 | **0.8747±0.0243** | 0.6987±0.2685 | **0.8524±0.0298** | 0.6476±0.2505 | **0.8335±0.0318** |
| 2 | 0.8512±0.0333 | **0.8867±0.0300** | 0.4878±0.3210 | **0.8723±0.0361** | 0.4581±0.2952 | **0.8277±0.0472** |
| 3 | 0.8921±0.0208 | **0.8924±0.0183** | 0.1644±0.3396 | **0.8104±0.0576** | 0.1395±0.2904 | **0.7537±0.0788** |
| 4 | **0.8964±0.0176** | 0.8804±0.0256 | 0.5403±0.3666 | **0.8801±0.0257** | 0.4897±0.3246 | **0.8188±0.0412** |
| 5 | 0.8492±0.0300 | **0.8690±0.0344** | **0.7561±0.2065** | 0.5276±0.3452 | **0.6676±0.1905** | 0.4883±0.3184 |
| 6 | 0.8806±0.0248 | **0.8890±0.0230** | 0.2005±0.3537 | **0.8323±0.1615** | 0.1817±0.3193 | **0.8149±0.1582** |
| 7 | 0.8447±0.0335 | **0.8662±0.0303** | 0.5131±0.3417 | **0.8499±0.0340** | 0.4827±0.3210 | **0.7697±0.0754** |
| 8 | 0.8427±0.0329 | **0.8897±0.0218** | 0.5107±0.3399 | **0.8893±0.0219** | 0.4840±0.3175 | **0.8753±0.0245** |
| 9 | **0.8107±0.0548** | 0.7681±0.0472 | 0.1443±0.2992 | **0.4513±0.2982** | 0.1242±0.2609 | **0.4096±0.2741** |

Table 2: The conformal prediction performance measured by the FSC metric on the *Natural_Scene* dataset.

| Class Id | bin size = 2 | | bin size = 4 | | bin size = 8 | |
|---|---|---|---|---|---|---|
| | Dirichlet | SNEFY-LDL | Dirichlet | SNEFY-LDL | Dirichlet | SNEFY-LDL |
| 1 | 0.8913±0.0205 | **0.8948±0.0175** | 0.8407±0.0511 | **0.8836±0.0189** | 0.5391±0.3022 | **0.8277±0.0675** |
| 2 | **0.8816±0.0248** | 0.8658±0.0297 | **0.8703±0.0266** | 0.8472±0.0364 | **0.8475±0.0326** | 0.8210±0.0435 |
| 3 | 0.8737±0.0278 | **0.8963±0.0175** | 0.8552±0.0292 | **0.8785±0.0258** | 0.8290±0.0349 | **0.8505±0.0297** |
| 4 | 0.8848±0.0238 | **0.8879±0.0222** | 0.8274±0.0367 | **0.8725±0.0270** | 0.7193±0.1287 | **0.7261±0.1500** |
| 5 | 0.8937±0.0177 | **0.8955±0.0184** | 0.8150±0.0462 | **0.8381±0.0403** | 0.0793±0.2388 | **0.4199±0.3376** |
| 6 | 0.8911±0.0191 | **0.8920±0.0205** | **0.8625±0.0366** | 0.8334±0.0453 | **0.6025±0.2969** | 0.5048±0.3136 |

Table 3: The conformal prediction performance measured by the FSC metric on the *SBU_3DFE* dataset.

ance $\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ in Eq. (12). By applying the Chebyshev's inequality [Grimmett and Stirzaker, 2020], we can construct a confidence interval for the $r$th label's composition ratio in describing $\boldsymbol{x}$, $\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r}$, which can be formally stated as

$$\mathrm{P}(|\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r} - \mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]| \leq k\sqrt{\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]}) \geq 1 - \frac{1}{k^2}, \quad (17)$$

i.e., the confidence interval of $\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r}$ at the $1 - 1/k^2$ level is

$$\left[\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] - k\sqrt{\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]}, \mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] + k\sqrt{\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]}\right],$$

which can be further calibrated as a conformal prediction task for the one-dimensional uncertainty estimate $\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r}$. According to Angelopoulos and Bates [2021], on a calibration set with $N_{\text{cal}}$ samples, we can define a calibration score function as

$$s(\boldsymbol{x}, \boldsymbol{\ell}_{\boldsymbol{x}}^{y_r}) = \frac{|\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r} - \mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]|}{k\sqrt{\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]}}, \quad (18)$$

where $\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r}$ is the ground-truth value of the $r$th label's composition ratio in describing the calibration sample $\boldsymbol{x}$. By computing the calibration scores of all calibration samples, we can get the $\lceil(1 - 1/k^2)(N_{\text{cal}} + 1)\rceil/N_{\text{cal}}$ quantile of the calibration scores as $\hat{q}_{y_r}$. For a new sample $\boldsymbol{x}$, the calibrated confidence interval for $\boldsymbol{\ell}_{\boldsymbol{x}}^{y_r}$ at the $1 - 1/k^2$ level is

$$\begin{aligned}\mathcal{C}(\boldsymbol{x}, \boldsymbol{\ell}^{y_r}) = \Big[&\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] - k \cdot \hat{q}_{y_r}\sqrt{\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]}, \\ &\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] + k \cdot \hat{q}_{y_r}\sqrt{\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]}\Big].\end{aligned} \quad (19)$$

As a baseline, we extend the competitive SA-BFGS [Geng, 2016] algorithm by modeling the distribution of (rather than point-estimating) the label distribution vectors. We model the distribution using the Dirichlet distribution centered at the point prediction. Following the same routine of conformal prediction, we can also construct calibrated confidence intervals for label affiliation probabilities given new samples. Following Angelopoulos and Bates [2021], we use the Feature-Stratified Coverage (FSC) metric to evaluate the adaptivity of the constructed confidence intervals in Eq.(19), which categorizes the test samples into different groups by dividing the numeric values at the first feature dimension into different bins, then compute the coverage rate of the confidence intervals in each group, and picks up the lowest group-level coverage rate as the final metric value. We select the *Natural_Scene* and *SBU_3DFE* datasets with a fair number of examples and labels to test the performance of conformal prediction. We randomly split each dataset into the training, calibration and test sets according to the ratio of 50%/25%/25% for 100 times and report the average FSC scores with bin size equal to 2, 4 and 8. In the experiment, we aim to construct 90% level confidence intervals by setting $1/k^2 = 0.1$, which means that FSC scores closer to 0.9 indicates better conformal prediction performance.

Tables 2-3 compare the conformal prediction performance measured by the FSC metric on the *Natural_Scene* and

| Method | Cheby ↓ | Clark ↓ | Canb ↓ | KL ↓ | Cos ↑ | Inter ↑ |
|---|---|---|---|---|---|---|
| Random | 0.1490±0.0098 | 0.6359±0.0247 | 1.1999±0.0448 | 0.1409±0.0136 | 0.9054±0.0093 | 0.7950±0.0089 |
| Kmeans | 0.1456±0.0076 | 0.6242±0.0176 | 1.1815±0.0320 | 0.1361±0.0072 | 0.9079±0.0068 | 0.7981±0.0077 |
| CoreSet | 0.1484±0.0164 | 0.6322±0.0383 | 1.1962±0.0720 | 0.1399±0.0179 | 0.9042±0.0158 | 0.7941±0.0171 |
| Graph Density | 0.1428±0.0059 | 0.6175±0.0197 | 1.1671±0.0346 | 0.1326±0.0068 | 0.9108±0.0057 | 0.8013±0.0076 |
| Dirichlet | 0.1472±0.0090 | 0.6282±0.0245 | 1.1899±0.0470 | 0.1376±0.0116 | 0.9064±0.0085 | 0.7959±0.0099 |
| SNEFY-LDL | **0.1350±0.0030** | **0.5981±0.0134** | **1.1283±0.0254** | **0.1209±0.0049** | **0.9191±0.0030** | **0.8098±0.0042** |

Table 4: The label distribution active learning performance on the *Motive* dataset.

| Method | Cheby ↓ | Clark ↓ | Canb ↓ | KL ↓ | Cos ↑ | Inter ↑ |
|---|---|---|---|---|---|---|
| Random | 0.3591±0.0197 | 2.4871±0.0280 | 6.9131±0.1456 | 0.9835±0.0766 | 0.6554±0.0411 | 0.4528±0.0336 |
| Kmeans | 0.3690±0.0163 | 2.4929±0.0200 | 6.9407±0.0808 | 1.0498± 0.1078 | 0.6318±0.0375 | 0.4361±0.0228 |
| CoreSet | 0.3629±0.0193 | 2.4842±0.0233 | 6.8882±0.1089 | 1.0102±0.1078 | 0.6442±0.0489 | 0.4485±0.0340 |
| Graph Density | 0.3700±0.0242 | 2.4958±0.0277 | 6.9643±0.1394 | 1.0471±0.0991 | 0.6230±0.0453 | 0.4294±0.0321 |
| Dirichlet | 0.3720±0.0213 | 2.4951±0.0226 | 6.9502±0.0977 | 1.0628±0.0909 | 0.6249±0.0336 | 0.4326±0.0192 |
| SNEFY-LDL | **0.3474±0.0160** | **2.4807±0.0194** | **6.8755±0.0845** | **0.9244±0.0648** | **0.6819±0.0285** | **0.4718±0.0203** |

Table 5: The label distribution active learning performance on the *Natural_Scene* dataset.

*SBU_3DFE* datasets. For each comparison between SNEFY-LDL and Dirichlet, the better performer is highlighted by **boldface**. From the tables, we can find that the proposed SNEFY-LDL model outperforms the Dirichlet baseline in most cases. SNEFY-LDL constructs a multimodal distribution to model the distribution of label distribution vectors on the probability simplex space, which is more flexible than the unimodal Dirichlet, and contributes to the label distribution confidence intervals with greater adaptivity.

### 6.2 ACTIVE LEARNING

To further evaluate SNEFY-LDL's performance in uncertainty quantification, we choose the *Movie* and *Natural_Scene* datasets to conduct the active learning experiments. We first randomly split the two datasets into the training and test sets according to the ratio of 90%/10% for ten times. For each training-test set split, we randomly select 400 labeled training samples to form the initial labeled pool and take the remaining training samples as the unlabeled pool. We first train a SNEFY-LDL model with the labeled samples in the initial labeled pool. We then use different active learning strategies to pick up 100 informative samples from the unlabeled pool and query their labels. After augmenting the initial labeled pool with the 100 queried samples, we re-train the SNEFY-LDL model and evaluate the performance of its label distribution predictions produced by the closed-form conditional mean in Eq. (10). Following Geng [2016], the label distribution prediction performance is evaluated by the following six metrics: Chebyshev distance (Cheby), Clark distance (Clark), Canberra metric (Canb), Kullback-Leibler divergence (KL), Cosine coefficient (Cos) and Intersection (Inter). The average scores on the ten random training-test splits are reported. Six different active learning strategies are compared:

- **Random** [Zhan et al., 2022] randomly selects 100 samples from the unlabeled pool.
- **Kmeans** [Zhdanov, 2019] selects samples close to the cluster centroids generated by the Kmeans clustering [MacKay, 2003] in feature space.
- **CoreSet** [Sener and Savarese, 2018] selects the $k$-center samples [Har-Peled, 2011] as representative unlabeled samples, which is a variant of Kmeans.
- **Graph Density** [Ebert et al., 2012] selects highly connected samples in the constructed KNN graph [Preparata and Shamos, 2012].
- **Dirichlet** models the distribution of label distributions using a Dirichlet distribution [Ng et al., 2011] centered at predicted label distributions and selects samples with the largest differential entropy scores [Cover, 1999].
- **SNEFY-LDL** uses importance sampling [Kloek and Van Dijk, 1978] to estimate the differential entropy values of the conditional distributions modeled by SNEFY-LDL and picks up samples that have the largest differential entropy scores.

Tables 4-5 compare the performance of different active learning strategies, where the best performer is highlighted by **boldface**. As is shown in the tables, SNEFY-LDL consistently achieves the best performance in terms of all metrics. By accurately evaluating label distribution prediction uncertainties, SNEFY-LDL can pick up more informative unla-

| Base Learner | Bagging | Cheby ↓ | Clark ↓ | Canb ↓ | KL ↓ | Cos ↑ | Inter ↑ |
|---|---|---|---|---|---|---|---|
| SA-BFGS | Average | 0.1178±0.0020 | 0.3743±0.0068 | 0.7948±0.0163 | 0.0641±0.0022 | 0.9370±0.0020 | 0.8575±0.0028 |
| | Weighted | **0.1137±0.0023** | **0.3625±0.0062** | **0.7686±0.0149** | **0.0604±0.0021** | **0.9406±0.0020** | **0.8624±0.0026** |
| DF-LDL | Average | 0.1203±0.0017 | 0.3762±0.0059 | 0.8040±0.0146 | 0.0657±0.0019 | 0.9353±0.0018 | 0.8557±0.0025 |
| | Weighted | **0.1152±0.0024** | **0.3617±0.0070** | **0.7715±0.0168** | **0.0609±0.0024** | **0.9399±0.0023** | **0.8617±0.0030** |
| LDL-SCL | Average | 0.1256±0.0020 | 0.3828±0.0047 | 0.8260±0.0118 | 0.0699±0.0021 | 0.9315±0.0018 | 0.8519±0.0020 |
| | Weighted | **0.1246±0.0023** | **0.3772±0.0048** | **0.8147±0.0114** | **0.0684±0.0022** | **0.9330±0.0019** | **0.8540±0.0020** |
| LDL-LRR | Average | 0.1269±0.0021 | 0.3966±0.0052 | 0.8478±0.0131 | 0.0730±0.0022 | 0.9285±0.0019 | 0.8478±0.0023 |
| | Weighted | **0.1250±0.0020** | **0.3916±0.0044** | **0.8373±0.0106** | **0.0710±0.0020** | **0.9305±0.0018** | **0.8498±0.0019** |

Table 6: The label distribution ensemble learning performance on the *SBU_3DFE* dataset.

| Base Learner | Bagging | Cheby ↓ | Clarky ↓ | Canb ↓ | KL ↓ | Cos ↑ | Inter ↑ |
|---|---|---|---|---|---|---|---|
| SA-BFGS | Average | 0.0889±0.0085 | 0.3180±0.0197 | 0.6529±0.0461 | 0.0406±0.0058 | 0.9613±0.0057 | 0.8890±0.0086 |
| | Weighted | **0.0842±0.0084** | **0.3118±0.0200** | **0.6390±0.0466** | **0.0385±0.0056** | **0.9636±0.0056** | **0.8923±0.0089** |
| DF-LDL | Average | 0.0951±0.0093 | 0.3385±0.0253 | 0.6958±0.0611 | 0.0456±0.0078 | 0.9566±0.0072 | 0.8818±0.0110 |
| | Weighted | **0.0881±0.0092** | **0.3189±0.0258** | **0.6525±0.0617** | **0.0408±0.0074** | **0.9612±0.0069** | **0.8895±0.0110** |
| LDL-SCL | Average | 0.0911±0.0090 | 0.3249±0.0219 | 0.6746±0.0493 | 0.0424±0.0063 | 0.9596±0.0062 | 0.8854±0.0092 |
| | Weighted | **0.0865±0.0095** | **0.3153±0.0231** | **0.6509±0.0540** | **0.0400±0.0064** | **0.9621±0.0064** | **0.8898±0.0100** |
| LDL-LRR | Average | 0.0888±0.0090 | 0.3189±0.0209 | 0.6538±0.0473 | 0.0408±0.0065 | 0.9612±0.0063 | 0.8890±0.0090 |
| | Weighted | **0.0846±0.0089** | **0.3124±0.0197** | **0.6392±0.0444** | **0.0387±0.0062** | **0.9634±0.0063** | **0.8922±0.0087** |

Table 7: The label distribution ensemble learning performance on the *SJAFFE* dataset.

beled samples than the naive uncertainty quantification strategy, Dirichlet, as well as the representativeness based active learning strategies, Kmeans, CoreSet and Graph Density, which are even sometimes inferior to the Random strategy.

### 6.3 ENSEMBLE LEARNING

We also conduct experiments on ensemble learning to further verify SNEFY-LDL's ability in uncertainty quantification, with the expectation that reliable base learners can be identified by the SNEFY-LDL probability modeling. Bagging [Breiman, 1996] is adopted as an exemplary ensemble learning paradigm. We choose the *SEU_3DFE* and *SJAFFE* datasets, and randomly split them into training and test sets according to the ratio of 90%/10%. For each training-test set split, we randomly select 50 samples from the training set for 25 rounds, train 25 base LDL learners with the selected samples, and evaluate the label distribution prediction performance of the ensembled LDL model on the test set. Four competitive LDL models are employed as base learners: SA-BFGS [Geng, 2016], DF-LDL [González et al., 2021a], LDL-SCL [Jia et al., 2021] and LDL-LRR [Jia et al., 2023a], and two strategies are adopted to ensemble base learner predictions: 1) **Average**: aggregate the 25 base learner predictions with the uniform weight $1/25$, and 2) **Weighted**: weight each base learner prediction in proportion to its corresponding SNEFY-LDL probability density conditioned on each test sample in an instance-wise manner.

Tables 6-7 compare the two different ensemble learning strategies, where the best strategy is highlighted by **boldface**. From the tables, we find that the **Weighted** strategy is significantly better than **Average** in terms of all metrics. This implies that SNEFY-LDL provides an effective mechanism to quantify the reliability of base learners' label distribution predictions so that the reliable base learners are highlighted to contribute to a better ensemble learning performance.

## 7 CONCLUSION

We propose a novel LDL paradigm: estimate the distribution of label distribution vectors on the probability simplex, which brings a bird's-eye view on the relative significance of all possible label distributions. By uncovering the underlying relationship between SNEFY and LDL, we develop the SNEFY-LDL model that can provide a tractable formulation of the conditional distribution of label distribution vectors, enjoying great expressivity and high computational efficiency. SNEFY-LDL admits closed-form expressions for the distribution's mean, variance and covariance, making SNEFY-LDL able to provide real-time responses in real-world applications. Experiments on conformal prediction, active learning and ensemble learning demonstrate the great utility of SNEFY-LDL for uncertainty-aware applications.

## References


Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

Leo Breiman. Bagging predictors. *Machine Learning*, 24: 123–140, 1996.

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.

Thomas M Cover. *Elements of Information Theory*. John Wiley & Sons, 1999.

Sandra Ebert, Mario Fritz, and Bernt Schiele. RALF: A reinforced active learning formulation for object class recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633. IEEE, 2012.

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 *International Joint Conference on Artificial Intelligence*, pages 3511–3517. Citeseer, 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.

Michael Greenacre. Compositional data analysis. *Annual Review of Statistics and its Application*, 8(1):271–299, 2021.

Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford university press, 2020.

Sariel Har-Peled. *Geometric Approximation Algorithms*. Number 173. American Mathematical Soc., 2011.

Liang He, Yunan Lu, Weiwei Li, and Xiuyi Jia. Generative calibration of inaccurate annotation for label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12394–12401, 2024.

Weiming Huang, Daokun Zhang, Gengchen Mai, Xu Guo, and Lizhen Cui. Learning urban region representations with pois and hierarchical graph infomax. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:134–145, 2023.

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, 2021.

Xiuyi Jia, Tian Qin, Yunan Lu, and Weiwei Li. Adaptive weighted ranking-oriented label distribution learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023a.

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(2):1695–1707, 2023b.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

Teun Kloek and Herman K Van Dijk. Bayesian estimates of equation system parameters: an application of integration by monte carlo. *Econometrica: Journal of the Econometric Society*, pages 1–19, 1978.

Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1467–1475, 2015.

David JC MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge university press, 2003.

Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *International Conference on Computational Linguistics*, 2002.

Dharmendra S Modha and Yeshaiahu Fainman. A learning law for density estimation. *IEEE Transactions on Neural Networks*, 5(3):519–523, 1994.

Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. *Dirichlet and Related Distributions: Theory, Methods and Applications*. John Wiley & Sons, 2011.

Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer, 1999.

Franco P Preparata and Michael I Shamos. *Computational Geometry: An Introduction*. Springer Science & Business Media, 2012.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Wei Shen, Kai Zhao, Yilu Guo, and Alan L Yuille. Label distribution learning forests. *Advances in Neural Information Processing Systems*, 30, 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 Annual Meeting of the Association for Computational Linguistics*, pages 1167–1172, 2019.

Russell Tsuchida, Cheng Soon Ong, and Dino Sejdinovic. Squared neural families: a new class of tractable density models. *Advances in Neural Information Processing Systems*, 36, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Ragav Venkatesan and Baoxin Li. *Convolutional Neural Networks in Visual Computing: A Concise Guide*. CRC Press, 2017.

Jing Wang and Xin Geng. Classification with label distribution learning. In *International Joint Conference on Artificial Intelligence*, volume 1, page 2, 2019.

Jing Wang and Xin Geng. Label distribution learning machine. In *International Conference on Machine Learning*, pages 10749–10759. PMLR, 2021a.

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

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, Yu-Kun Lai, Dongyu She, Ming-Ming 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.

Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, Philip S Yu, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14: 1–37, 2008.

Kouzhiqiang Yucheng Xie, Jing Wang, Yuheng Jia, Boyu Shi, and Xin Geng. RankMatch: A novel approach to semi-supervised label distribution learning leveraging inter-label correlations. *arXiv preprint arXiv:2312.06343*, 2023.

Luhui Xu, Jingying Chen, and Yanling Gan. Head pose estimation using improved label distribution learning with fewer annotations. *Multimedia Tools and Applications*, 78:19141–19162, 2019a.

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

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

Ning Xu, Jun Shu, Yun-Peng Liu, and Xin Geng. Variational label enhancement. In *International Conference on Machine Learning*, pages 10597–10606. PMLR, 2020.

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*, volume 31, 2017.

Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*, 2022.

Fedor Zhdanov. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

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 THEOREM PROOFS

**Theorem 2.** *Let $\boldsymbol{t}_1(\boldsymbol{\ell}) = (\log \boldsymbol{\ell}^{y_1}, \log \boldsymbol{\ell}^{y_2}, \cdots, \log \boldsymbol{\ell}^{y_L}) : \Delta^{L-1} \to \mathbb{R}^L$ by setting $D_1 = L$, the activation function $\sigma$ be the exponential function* $\exp$*, the base measure $\mu_1(d\boldsymbol{\ell}) = d\boldsymbol{\ell}$ be the Lebesgue measure. Under the condition that $\boldsymbol{W}_1 > -1/2$ elementwise, the kernel function $\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x})$ admits a closed form:*

$$\boldsymbol{k}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x}) = \exp(\boldsymbol{w}_{2i}^\top \boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{w}_{2j}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_i + b_j) \cdot \frac{\prod_{l=1}^L \Gamma(1 + w_{1il} + w_{1jl})}{\Gamma\big(L + \sum_{l=1}^L (w_{1il} + w_{1jl})\big)}, \tag{9}$$

*where $w_{1il}$ is the $il$-th element of matrix $\boldsymbol{W}_1$ and $\Gamma(\cdot)$ is the gamma function.*

*Proof.* According to Eq. (8),

$$\begin{aligned}
\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x}) &= \int_{\Delta^{L-1}} \tilde{\boldsymbol{k}}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{\ell}, \boldsymbol{x}) \mu_1(d\boldsymbol{\ell}) \\
&= \int_{\Delta^{L-1}} \sigma(\boldsymbol{w}_{1i}^\top \boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{w}_{2i}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_i) \cdot \sigma(\boldsymbol{w}_{1j}^\top \boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{w}_{2j}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_j) \mu_1(d\boldsymbol{\ell}).
\end{aligned}$$

Given the setting $\boldsymbol{t}_1(\boldsymbol{\ell}) = (\log \boldsymbol{\ell}^{y_1}, \log \boldsymbol{\ell}^{y_2}, \cdots, \log \boldsymbol{\ell}^{y_L})$, $\sigma = \exp$ and $\mu_1(d\boldsymbol{\ell}) = d\boldsymbol{\ell}$, $\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}$ can be written as

$$\begin{aligned}
\boldsymbol{k}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x}) &= \int_{\Delta^{L-1}} \exp(\boldsymbol{w}_{1i}^\top \boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{w}_{2i}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_i) \cdot \exp(\boldsymbol{w}_{1j}^\top \boldsymbol{t}_1(\boldsymbol{\ell}) + \boldsymbol{w}_{2j}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_j) d\boldsymbol{\ell} \\
&= \exp(\boldsymbol{w}_{2i}^\top \boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{w}_{2j}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_i + b_j) \cdot \int_{\Delta^{L-1}} \prod_{l=1}^L (\boldsymbol{\ell}^{y_l})^{w_{1il} + w_{1jl}} d\boldsymbol{\ell}.
\end{aligned}$$

As $\boldsymbol{W}_1 > -1/2$ elementwise, $w_{1il} + w_{1jl} + 1 > 0$. Assuming $\boldsymbol{\ell}$ follows a Dirichlet distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_L)$, where $\alpha_l = w_{1il} + w_{1jl} + 1 > 0$, its probability density, $\mathrm{P}_{\mathrm{Dir}}(d\boldsymbol{\ell})/d\boldsymbol{\ell}$, is in the form:

$$\frac{\mathrm{P}_{\mathrm{Dir}}(d\boldsymbol{\ell})}{d\boldsymbol{\ell}} = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{l=1}^L (\boldsymbol{\ell}^{y_l})^{\alpha_l - 1},$$

where $\mathrm{B}(\cdot)$ is the beta function. Considering the fact that $\int_{\Delta^{L-1}} \mathrm{P}_{\mathrm{Dir}}(d\boldsymbol{\ell}) = 1$,

$$\int_{\Delta^{L-1}} \prod_{l=1}^L (\boldsymbol{\ell}^{y_l})^{\alpha_l - 1} d\boldsymbol{\ell} = \mathrm{B}(\boldsymbol{\alpha}).$$

That is to say

$$\int_{\Delta^{L-1}} \prod_{l=1}^L (\boldsymbol{\ell}^{y_l})^{w_{1il} + w_{1jl}} d\boldsymbol{\ell} = \frac{\prod_{l=1}^L \Gamma(1 + w_{1il} + w_{1jl})}{\Gamma\big(L + \sum_{l=1}^L (w_{1il} + w_{1jl})\big)}.$$

Therefore,

$$\boldsymbol{k}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x}) = \exp(\boldsymbol{w}_{2i}^\top \boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{w}_{2j}^\top \boldsymbol{t}_2(\boldsymbol{x}) + b_i + b_j) \cdot \frac{\prod_{l=1}^L \Gamma(1 + w_{1il} + w_{1jl})}{\Gamma\big(L + \sum_{l=1}^L (w_{1il} + w_{1jl})\big)}.$$

□

**Theorem 3.** *Assuming the label distribution vector $\boldsymbol{\ell}$ follows the SNEFY conditional distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{\Theta})$ in Eq.* (6) *with the kernel function $\boldsymbol{k}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x})$ given in Eq.* (9)*, under the setting that $\boldsymbol{t}_1(\boldsymbol{\ell}) = (\log \boldsymbol{\ell}^{y_1}, \log \boldsymbol{\ell}^{y_2}, \cdots, \log \boldsymbol{\ell}^{y_L})$, $\sigma = \exp$, and $\mu_1(d\boldsymbol{\ell}) = d\boldsymbol{\ell}$, as well as the constraint that $\boldsymbol{W}_1 > -1/2$ elementwise, for the $r$th label's composition ratio, $\boldsymbol{\ell}^{y_r}$, we have its conditional mean $\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ as*

$$\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] = \frac{\mathrm{vec}(\boldsymbol{V}^\top \boldsymbol{V})^\top \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \circ \boldsymbol{F}^{y_r})}{\mathrm{vec}(\boldsymbol{V}^\top \boldsymbol{V})^\top \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}, \tag{10}$$

*where $\circ$ denotes Hadamard product, and $\boldsymbol{F}^{y_r}$ is a $n \times n$ matrix, whose $ij$th entry is*

$$F_{ij}^{y_r} = \frac{1 + w_{1ir} + w_{1jr}}{L + \sum_{l=1}^L (w_{1il} + w_{1jl})}. \tag{11}$$

*The conditional variance of $\boldsymbol{\ell}^{y_r}$, $\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$, is*

$$\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] = \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})\circ\boldsymbol{G}^{y_r})}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))} - \mathrm{E}^2[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}], \tag{12}$$

*where $\boldsymbol{G}^{y_r}$ is a $n\times n$ matrix , with its $ij$th element being*

$$G_{ij}^{y_r} = \frac{(1+w_{1ir}+w_{1jr})(2+w_{1ir}+w_{1jr})}{[L+\sum_{l=1}^{L}(w_{1il}+w_{1jl})][1+L+\sum_{l=1}^{L}(w_{1il}+w_{1jl})]}. \tag{13}$$

*For two different labels $y_r$ and $y_s$, with $y_r\neq y_s$, the conditional covariance of $\boldsymbol{\ell}^{y_r}$ and $\boldsymbol{\ell}^{y_s}$, $\mathrm{Cov}[\boldsymbol{\ell}^{y_r},\boldsymbol{\ell}^{y_s}|\boldsymbol{x}]$, is*

$$\mathrm{Cov}[\boldsymbol{\ell}^{y_r},\boldsymbol{\ell}^{y_s}|\boldsymbol{x}] = \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x})\circ\boldsymbol{H}^{y_r,y_s})}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))} - \mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]\cdot\mathrm{E}[\boldsymbol{\ell}^{y_s}|\boldsymbol{x}], \tag{14}$$

*where $\boldsymbol{H}^{y_r,y_s}$ is a $n\times n$ matrix , with its $ij$th element being*

$$H_{ij}^{y_r,y_s} = \frac{(1+w_{1ir}+w_{1jr})(1+w_{1is}+w_{1js})}{[L+\sum_{l=1}^{L}(w_{1il}+w_{1jl})][1+L+\sum_{l=1}^{L}(w_{1il}+w_{1jl})]}. \tag{15}$$

*Proof.* For any fixed scalar function $\varphi(\boldsymbol{\ell})$ of $\boldsymbol{\ell}$ with $\varphi(\cdot):\Delta^{L-1}\to\mathbb{R}$, its expectation with regard to the conditional SNEFY distribution in Eq. (6), $\mathrm{E}[\varphi(\boldsymbol{\ell})|\boldsymbol{x}]$, can be computed as

$$\begin{aligned}\mathrm{E}[\varphi(\boldsymbol{\ell})|\boldsymbol{x}] &= \int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})\\ &= \int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{\ell},\boldsymbol{x}))}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}\mu_1(d\boldsymbol{\ell})\\ &= \frac{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{\Phi}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}{\mathrm{vec}(\boldsymbol{V}^\top\boldsymbol{V})^\top\mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))},\end{aligned}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\Theta}}(\boldsymbol{x})\in\mathbb{R}^{n\times n}$ is the elementwise integral:

$$\boldsymbol{\Phi}_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\widetilde{\boldsymbol{K}}_{\boldsymbol{\Theta}}(\boldsymbol{\ell},\boldsymbol{x})\mu_1(d\boldsymbol{\ell}),$$

whose $ij$th element is

$$\begin{aligned}\boldsymbol{\phi}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x}) &= \int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\tilde{\boldsymbol{k}}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{\ell},\boldsymbol{x})\mu_1(d\boldsymbol{\ell})\\ &= \int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\sigma(\boldsymbol{w}_{1i}^\top\boldsymbol{t}_1(\boldsymbol{\ell})+\boldsymbol{w}_{2i}^\top\boldsymbol{t}_2(\boldsymbol{x})+b_i)\cdot\sigma(\boldsymbol{w}_{1j}^\top\boldsymbol{t}_1(\boldsymbol{\ell})+\boldsymbol{w}_{2j}^\top\boldsymbol{t}_2(\boldsymbol{x})+b_j)\mu_1(d\boldsymbol{\ell}).\end{aligned}$$

By setting $\boldsymbol{t}_1(\boldsymbol{\ell}) = (\log\boldsymbol{\ell}^{y_1},\log\boldsymbol{\ell}^{y_2},\cdots,\log\boldsymbol{\ell}^{y_L})$, $\sigma=\exp$, and $\mu_1(d\boldsymbol{\ell}) = d\boldsymbol{\ell}$, $\boldsymbol{\phi}_{\sigma,\boldsymbol{t}_1,\boldsymbol{t}_2,\mu_1}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x})$ can be written as

$$\begin{aligned}\boldsymbol{\phi}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j;\boldsymbol{x}) &= \int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\exp(\boldsymbol{w}_{1i}^\top\boldsymbol{t}_1(\boldsymbol{\ell})+\boldsymbol{w}_{2i}^\top\boldsymbol{t}_2(\boldsymbol{x})+b_i)\cdot\exp(\boldsymbol{w}_{1j}^\top\boldsymbol{t}_1(\boldsymbol{\ell})+\boldsymbol{w}_{2j}^\top\boldsymbol{t}_2(\boldsymbol{x})+b_j)d\boldsymbol{\ell}\\ &= \exp(\boldsymbol{w}_{2i}^\top\boldsymbol{t}_2(\boldsymbol{x})+\boldsymbol{w}_{2j}^\top\boldsymbol{t}_2(\boldsymbol{x})+b_i+b_j)\cdot\int_{\Delta^{L-1}}\varphi(\boldsymbol{\ell})\prod_{l=1}^{L}(\boldsymbol{\ell}^{y_l})^{w_{1il}+w_{1jl}}d\boldsymbol{\ell}.\end{aligned}$$

As $\boldsymbol{W}_1 > -1/2$ elementwise, $w_{1il}+w_{1jl}+1>0$. Assuming $\boldsymbol{\ell}$ follows a Dirichlet distribution with parameters $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_L)$, where $\alpha_l = w_{1il}+w_{1jl}+1>0$, its probability density, $\mathrm{P}_{\mathrm{Dir}}(d\boldsymbol{\ell})/d\boldsymbol{\ell}$, is in the form:

$$\frac{\mathrm{P}_{\mathrm{Dir}}(d\boldsymbol{\ell})}{d\boldsymbol{\ell}} = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{l=1}^{L}(\boldsymbol{\ell}^{y_l})^{\alpha_l-1},$$

where $\mathrm{B}(\cdot)$ is the beta function. Then, we have

$$\begin{aligned}\int_{\Delta^{L-1}} \varphi(\boldsymbol{\ell}) \prod_{l=1}^{L} (\boldsymbol{\ell}^{y_l})^{w_{1il}+w_{1jl}} d\boldsymbol{\ell} &= \mathrm{B}(\boldsymbol{\alpha}) \int_{\Delta^{L-1}} \varphi(\boldsymbol{\ell}) \mathrm{P}_{\mathrm{Dir}}(d\boldsymbol{\ell}) \\ &= \mathrm{B}(\boldsymbol{\alpha}) \mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}] \\ &= \frac{\prod_{l=1}^{L} \Gamma(1 + w_{1il} + w_{1jl})}{\Gamma\big(L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})\big)} \mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}],\end{aligned}$$

where $\mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}]$ is the expectation of $\varphi(\boldsymbol{\ell})$ with regard to the Dirichlet distribution parameterized by $\boldsymbol{w}_{1i}$ and $\boldsymbol{w}_{1j}$. Therefore,

$$\begin{aligned}\boldsymbol{\phi}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x}) =& \exp(\boldsymbol{w}_{2i}^{\top} \boldsymbol{t}_2(\boldsymbol{x}) + \boldsymbol{w}_{2j}^{\top} \boldsymbol{t}_2(\boldsymbol{x}) + b_i + b_j) \cdot \frac{\prod_{l=1}^{L} \Gamma(1 + w_{1il} + w_{1jl})}{\Gamma\big(L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})\big)} \mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}] \\ =& \boldsymbol{k}_{\boldsymbol{t}_2}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j; \boldsymbol{x}) \mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}].\end{aligned}$$

By using $\boldsymbol{E}_{\varphi}$ to denote the $n \times n$ matrix whose $ij$th element is $\mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}]$, we have

$$\boldsymbol{\Phi}_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \circ \boldsymbol{E}_{\varphi},$$

$$\mathrm{E}[\varphi(\boldsymbol{\ell})|\boldsymbol{x}] = \frac{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \circ \boldsymbol{E}_{\varphi})}{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}.$$

For the Dirichlet distribution, $\mathrm{E}_{\mathrm{Dir}}[\varphi(\boldsymbol{\ell}); \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}]$ has closed forms for some moments. In particular, for $\varphi(\boldsymbol{\ell}) = \boldsymbol{\ell}^{y_r}$, $\varphi(\boldsymbol{\ell}) = (\boldsymbol{\ell}^{y_r})^2$, and $\varphi(\boldsymbol{\ell}) = \boldsymbol{\ell}^{y_r} \cdot \boldsymbol{\ell}^{y_s}$ with $y_r \neq y_s$, we respectively have

$$\mathrm{E}_{\mathrm{Dir}}[\boldsymbol{\ell}^{y_r}; \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}] = \frac{1 + w_{1ir} + w_{1jr}}{L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})},$$

$$\mathrm{E}_{\mathrm{Dir}}[(\boldsymbol{\ell}^{y_r})^2; \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}] = \frac{(1 + w_{1ir} + w_{1jr})(2 + w_{1ir} + w_{1jr})}{[L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})][1 + L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})]},$$

$$\mathrm{E}_{\mathrm{Dir}}[\boldsymbol{\ell}^{y_r} \cdot \boldsymbol{\ell}^{y_s}; \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}] = \frac{(1 + w_{1ir} + w_{1jr})(1 + w_{1is} + w_{1js})}{[L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})][1 + L + \sum_{l=1}^{L} (w_{1il} + w_{1jl})]}.$$

Finally, we have

$$\begin{aligned}\mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] &= \frac{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \circ \boldsymbol{F}^{y_r})}{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}, \\ \mathrm{E}[(\boldsymbol{\ell}^{y_r})^2|\boldsymbol{x}] &= \frac{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \circ \boldsymbol{G}^{y_r})}{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))}, \\ \mathrm{E}[\boldsymbol{\ell}^{y_r} \cdot \boldsymbol{\ell}^{y_s}|\boldsymbol{x}] &= \frac{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}) \circ \boldsymbol{H}^{y_r, y_s})}{\mathrm{vec}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\top} \mathrm{vec}(\boldsymbol{K}_{\boldsymbol{\Theta}}(\boldsymbol{x}))},\end{aligned}$$

where $\boldsymbol{F}^{y_r}$, $\boldsymbol{G}^{y_r}$ and $\boldsymbol{H}^{y_r, y_s}$ denote the $n \times n$ matrices whose $ij$th elements are $\mathrm{E}_{\mathrm{Dir}}[\boldsymbol{\ell}^{y_r}; \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}]$, $\mathrm{E}_{\mathrm{Dir}}[(\boldsymbol{\ell}^{y_r})^2; \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}]$ and $\mathrm{E}_{\mathrm{Dir}}[\boldsymbol{\ell}^{y_r} \cdot \boldsymbol{\ell}^{y_s}; \boldsymbol{w}_{1i}, \boldsymbol{w}_{1j}]$ respectively.

$\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ and $\mathrm{Cov}[\boldsymbol{\ell}^{y_r}, \boldsymbol{\ell}^{y_s}|\boldsymbol{x}]$ can be directly derived by using the identities that $\mathrm{Var}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] = \mathrm{E}[(\boldsymbol{\ell}^{y_r})^2|\boldsymbol{x}] - \mathrm{E}^2[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}]$ and $\mathrm{Cov}[\boldsymbol{\ell}^{y_r}, \boldsymbol{\ell}^{y_s}|\boldsymbol{x}] = \mathrm{E}[\boldsymbol{\ell}^{y_r} \cdot \boldsymbol{\ell}^{y_s}|\boldsymbol{x}] - \mathrm{E}[\boldsymbol{\ell}^{y_r}|\boldsymbol{x}] \cdot \mathrm{E}[\boldsymbol{\ell}^{y_s}|\boldsymbol{x}]$. □

# B EXPERIMENTAL DETAILS

## B.1 CONFORMAL PREDICTION

We select the *Natural Scene* and *SBU_3DFE* datasets for the conformal prediction experiments, as they have a relatively large number of labels that give more freedom in the change of label distribution compositions to warrant more uncertainties in label distribution predictions, and they also have enough samples to do the training/calibration/test set split. According

to the ratio of 50%/25%/25%, we randomly split each dataset into the training, calibration and test sets. Here, we denote the training, calibration and test sets as $\mathcal{S}^{\text{tr}} = \{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{tr}}}) : i = 1, \cdots, N_{\text{tr}}\}$, $\mathcal{S}^{\text{cal}} = \{(\boldsymbol{x}_i^{\text{cal}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{cal}}}) : i = 1, \cdots, N_{\text{cal}}\}$ and $\mathcal{S}^{\text{te}} = \{(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{te}}}) : i = 1, \cdots, N_{\text{te}}\}$ respectively, with $N_{\text{tr}}$, $N_{\text{cal}}$ and $N_{\text{te}}$ denoting the number of samples in the training, calibration and test sets respectively. For each split, a SNEFY-LDL is first trained on the training set $\mathcal{S}^{\text{tr}}$. The trained SNEFY-LDL is then leveraged to construct a 90% level confidence interval for each label's composition ratio with SNEFY-LDL's closed-form conditional mean and variance. The confidence intervals constructed by the trained SNEFY-LDL model are then calibrated with the calibration set. The adaptivity of the calibrated confidence intervals are finally evaluated on the test set, as an indicator of SNEFY-LDL's ability in quantifying the prediction uncertainty of each label's composition ratio. On the test set, the adaptivity of the calibrated confidence intervals $\mathcal{C}(\boldsymbol{x}, \ell^{y_r})$ in Eq. (19) for the $r$th label's composition ratio $\ell^{y_r}$ is measured by the Feature-Stratified Coverage (FSC) metric [Angelopoulos and Bates, 2021]. For calculating the FSC metric, we first bin the first feature of $\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathcal{S}_{\text{te}}$ into a number of categories, $1, \cdots, G$, then categorize test samples in $\mathcal{S}_{\text{te}}$ into different groups $\{\mathcal{S}_g^{\text{te}} : g = 1, \cdots, G\}$ according to the first feature's category values. Here, $G$ is termed as bin size or group number. The FSC metric for the confidence intervals of the $r$th label's composition ratio $\ell^{y_r}$ is evaluated as the minimal coverage rate among the groups $\{\mathcal{S}_g^{\text{te}} : g = 1, \cdots, G\}$:

$$\text{FSC}(\ell^{y_r}) := \min_{g \in \{1, \cdots, G\}} \frac{1}{|\mathcal{S}_g^{\text{te}}|} \sum_{\boldsymbol{x} \in \mathcal{S}_g^{\text{te}}} \mathbb{1}\left\{\ell_{\boldsymbol{x}}^{y_r} \in \mathcal{C}(\boldsymbol{x}, \ell^{y_r})\right\}. \tag{20}$$

The detailed procedure of conformal prediction with SNEFY-LDL is described by Algorithm A1.

**Algorithm A1** Conformal Prediction with SNEFY-LDL

**Input**: Training set $\mathcal{S}^{\text{tr}} = \{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{tr}}}) : i = 1, \cdots, N_{\text{tr}}\}$, calibration set $\mathcal{S}^{\text{cal}} = \{(\boldsymbol{x}_i^{\text{cal}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{cal}}}) : i = 1, \cdots, N_{\text{cal}}\}$ and test set $\mathcal{S}^{\text{te}} = \{(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{te}}}) : i = 1, \cdots, N_{\text{te}}\}$, as well as the given label set $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\}$.
**Parameter**: $1/k^2 = 0.1$ for constructing 90% level confidence intervals.
**Output**: FSC scores on the test set.

Train a SNEFY-LDL model on the training set $\mathcal{S}^{\text{tr}}$ with Algorithm 1;
**for** each sample $\boldsymbol{x} \in \mathcal{S}^{\text{cal}}$ **do**
  **for** each label $y_r \in \mathcal{Y}$ **do**
    Calculate the calibration score $s(\boldsymbol{x}, \boldsymbol{\ell}_{\boldsymbol{x}}^{y_r})$ with Eq.(18) and the trained SNEFY-LDL model;
  **end for**
**end for**
**for** each label $y_r \in \mathcal{Y}$ **do**
  Calculate the $\lceil (1 - 1/k^2)(N_{\text{cal}} + 1) \rceil / N_{\text{cal}}$ quantile as $\hat{q}_{y_r}$ among the $N_{\text{cal}}$ calibration scores $s(\boldsymbol{x}, \boldsymbol{\ell}_{\boldsymbol{x}}^{y_r})$ with $\boldsymbol{x} \in \mathcal{S}^{\text{cal}}$;
**end for**
**for** each sample $\boldsymbol{x} \in \mathcal{S}^{\text{te}}$ **do**
  **for** each label $y_r \in \mathcal{Y}$ **do**
    Construct the calibrated confidence interval $\mathcal{C}(\boldsymbol{x}, \ell^{y_r})$ with Eq. (19) and the calculated quantile $\hat{q}_{y_r}$;
  **end for**
**end for**
**for** each label $y_r \in \mathcal{Y}$ **do**
  Evaluate the $\text{FSC}(\ell^{y_r})$ score with Eq. (20) and the $N_{\text{te}}$ confidence intervals $\mathcal{C}(\boldsymbol{x}, \ell^{y_r})$ with $\boldsymbol{x} \in \mathcal{S}^{\text{te}}$;
**end for**
**return** the evaluated $\text{FSC}(\ell^{y_r})$ scores for all $y_r \in \mathcal{Y}$.

### B.2 ACTIVE LEARNING

The *Motive* and *Natural Scene* datasets are selected for active learning, as they are relatively sensitive to the label sparsity issue, more suitable to benchmark the performance change with informative samples labeled and augmented to the training data. We randomly split the two datasets into the training and test sets according to the ratio of 90%/10% for ten times. We denote the training set as $\mathcal{S}^{\text{tr}} = \{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{tr}}}) : i = 1, \cdots, N_{\text{tr}}\}$ and test set as $\mathcal{S}^{\text{te}} = \{(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{te}}}) : i = 1, \cdots, N_{\text{te}}\}$, where $N_{\text{tr}}$ and $N_{\text{te}}$ are respectively the number of samples in the training and test sets. For each training-test set split, we randomly select 400 labeled samples from the training set $\mathcal{S}^{\text{tr}}$ to form the initial labeled pool and take the remaining samples in $\mathcal{S}^{\text{tr}}$ as unlabeled samples. We first train a SNEFY-LDL model with the initial labeled pool. To achieve active learning with the

trained SNEFY-LDL model, we first evaluate the differential entropy $H(\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ for each unlabeled sample $\boldsymbol{x}$ as

$$H(\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta}) = -\int_{\Delta^{L-1}} \left\{ \log \frac{\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})}{d\boldsymbol{\ell}} \right\} \mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta}), \tag{21}$$

then select 100 most informative unlabeled samples with the largest differential entropy values. After querying the labels of the 100 selected samples, we augment them into the initial labeled pool, and re-train another SNEFY-LDL model with the augmented labeled pool. With the re-trained SNEFY-LDL model, we can predict the label distribution vectors for samples in the test set given their feature vectors $\boldsymbol{x}$ by directly using the closed-form conditional mean in Eq.(10). As a criterion of active learning, the label distribution prediction performance on the test set is evaluated using the six metrics [Geng, 2016]: Chebyshev distance (Cheby), Clark distance (Clark), Canberra metric (Canb), Kullback-Leibler divergence (KL), Cosine coefficient (Cos) and Intersection (Inter). Given the ground-truth and predicted label distribution vectors as $\boldsymbol{\ell}$ and $\hat{\boldsymbol{\ell}}$ respectively, the evaluation metrics are defined as follows:

$$\begin{aligned}
&\mathrm{Cheby}(\boldsymbol{\ell},\hat{\boldsymbol{\ell}}) = \|\boldsymbol{\ell}-\hat{\boldsymbol{\ell}}\|_\infty \downarrow, \quad \mathrm{Clark}(\boldsymbol{\ell},\hat{\boldsymbol{\ell}}) = \left\|\frac{\boldsymbol{\ell}-\hat{\boldsymbol{\ell}}}{\boldsymbol{\ell}+\hat{\boldsymbol{\ell}}}\right\|_2 \downarrow, \quad \mathrm{Canb}(\boldsymbol{\ell},\hat{\boldsymbol{\ell}}) = \left\|\frac{\boldsymbol{\ell}-\hat{\boldsymbol{\ell}}}{\boldsymbol{\ell}+\hat{\boldsymbol{\ell}}}\right\|_1 \downarrow, \\
&\mathrm{KL}(\boldsymbol{\ell},\hat{\boldsymbol{\ell}}) = \sum_{y\in\mathcal{Y}} \boldsymbol{\ell}^y \log \frac{\boldsymbol{\ell}^y}{\hat{\boldsymbol{\ell}}^y} \downarrow, \quad \mathrm{Cos}(\boldsymbol{\ell},\hat{\boldsymbol{\ell}}) = \frac{\boldsymbol{\ell}^\top\hat{\boldsymbol{\ell}}}{\|\boldsymbol{\ell}\|_2\|\hat{\boldsymbol{\ell}}\|_2} \uparrow, \quad \mathrm{Inter}(\boldsymbol{\ell},\hat{\boldsymbol{\ell}}) = \sum_{y\in\mathcal{Y}} \min(\boldsymbol{\ell}^y,\hat{\boldsymbol{\ell}}^y) \uparrow .
\end{aligned} \tag{22}$$

For each metric, $\uparrow$ ($\downarrow$) indicates that higher (lower) scores imply better label distribution prediction performance.

However, the differential entropy $H(\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ in Eq. (21) cannot be computed in a closed form. To overcome this difficulty, we adopt importance sampling [Kloek and Van Dijk, 1978] to approximately estimate the differential entropy values, where the uniform distribution is selected as the proposal distribution. The detailed procedure is provided in Algorithm A2.

**Algorithm A2** Differential Entropy Estimation with Importance Sampling

**Input**: The feature vector of an unlabeled sample $\boldsymbol{x}$ and the conditional distribution $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ modeled by the trained SNEFY-LDL model, and the label set size $L$.
**Parameter**: The number of sampling iterations $N_{\text{iter}} = 1,000$.
**Output**: The estimated differential entropy for the unlabeled sample $\boldsymbol{x}$, $\hat{H}(\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$.

1: **for** each iteration $i \in \{1,\cdots,N_{\text{iter}}\}$ **do**
2: Sample a label distribution vector $\tilde{\boldsymbol{\ell}}^{(i)}$ from the uniform distribution over the simplex $\Delta^{L-1}$ with probability density $q(\tilde{\boldsymbol{\ell}}^{(i)}) = (L-1)!$;
3: Evaluate the probability density value $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})/d\boldsymbol{\ell}$ at $\boldsymbol{\ell} = \tilde{\boldsymbol{\ell}}^{(i)}$ as $p(\tilde{\boldsymbol{\ell}}^{(i)}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ with Eq. (6) and the trained SNEFY-LDL model;
4: **end for**
5: Calculate the approximated differential entropy $\hat{H}(\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta}) = -\frac{1}{N_{\text{iter}}}\sum_{i=1}^{N_{\text{iter}}} \frac{p(\tilde{\boldsymbol{\ell}}^{(i)}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})}{q(\tilde{\boldsymbol{\ell}}^{(i)})} \log p(\tilde{\boldsymbol{\ell}}^{(i)}|\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$;
6: **return** the estimated differential entropy $\hat{H}(\boldsymbol{x};\boldsymbol{V},\boldsymbol{\Theta})$ for the given unlabeled sample $\boldsymbol{x}$.

The detailed procedure for active learning with SNEFY-LDL is described by Algorithm A3.

### B.3 ENSEMBLE LEARNING

The *SBU_3DFE* and *SJAFFE* datasets are selected for the ensemble learning experiments, as they have a relatively small number of features, more efficient to train a number of base learners and do ensemble prediction. We randomly split the two datasets into the training and test sets according to the ratio of 90%/10% for ten times. We denote the training set as $\mathcal{S}^{\text{tr}} = \{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{tr}}}) : i = 1,\cdots,N_{\text{tr}}\}$ and test set as $\mathcal{S}^{\text{te}} = \{(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{te}}}) : i = 1,\cdots,N_{\text{te}}\}$, where $N_{\text{tr}}$ and $N_{\text{te}}$ are respectively the number of samples in the training and test sets. For each training-test set split, we randomly select 50 samples from the training set for 25 rounds, train 25 base LDL learners with the selected samples, and evaluate the label distribution prediction performance of the ensembled LDL model on the test set. Four competitive LDL algorithms are employed to train base learners: SA-BFGS [Geng, 2016], DF-LDL [González et al., 2021a], LDL-SCL [Jia et al., 2021] and LDL-LRR [Jia et al., 2023a]. To achieve ensemble learning with SNEFY-LDL, we first train a SNEFY-LDL model with the training set $\mathcal{S}^{\text{tr}}$, and then do the ensemble prediction by weighting the base learners according to the SNEFY-LDL conditional probability densities measured at their predictions given each test sample $\boldsymbol{x} \in \mathcal{S}^{\text{te}}$. The detailed procedure is shown in Algorithm A4.

**Algorithm A3** Active Learning with SNEFY-LDL

**Input**: Training set $\mathcal{S}^{\text{tr}} = \{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{tr}}}) : i = 1, \cdots, N_{\text{tr}}\}$ and test set $\mathcal{S}^{\text{te}} = \{(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{te}}}) : i = 1, \cdots, N_{\text{te}}\}$.
**Parameters**: The size of the initial labeled pool $N_{\text{initial}} = 400$ and the number of queried samples $N_{\text{query}} = 100$.
**Output**: The label distribution prediction performance scores measured by Cheby, Clark, Canb, KL, Cos and Inter on the test set $\mathcal{S}^{\text{te}}$.

Randomly select $N_{\text{initial}}$ samples from the training set $\mathcal{S}^{\text{tr}}$ to form the labeled pool $\mathcal{S}^{\text{tr}}_{\text{label}}$ and use the remaining samples to form the unlabeled pool $\mathcal{S}^{\text{tr}}_{\text{unlabel}}$;
Train a SNEFY-LDL model with the initial labeled pool;
**for** each sample $\boldsymbol{x} \in \mathcal{S}^{\text{tr}}_{\text{unlabel}}$ **do**
    Estimate the differential entropy value $\hat{H}(\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{\Theta})$ for $\boldsymbol{x}$ with Algorithm A2 and the trained SNEFY-LDL model;
**end for**
Select $N_{\text{query}}$ samples from $\mathcal{S}^{\text{tr}}_{\text{unlabel}}$ with the top-$N_{\text{query}}$ differential entropy values and query their label distributions;
Augment the $N_{\text{query}}$ queried samples into the labeled pool $\mathcal{S}^{\text{tr}}_{\text{label}}$;
Re-train another SNEFY-LDL model with the augmented labeled pool $\mathcal{S}^{\text{tr}}_{\text{label}}$;
Evaluate the label distribution prediction performance of the re-trained SNEFY-LDL model on the test set $\mathcal{S}^{\text{te}}$ (as an average over all test samples) with the metrics of Cheby, Clark, Canb, KL, Cos and Inter defined in Eq. (22);
**return** the label distribution prediction performance scores measured by Cheby, Clark, Canb, KL, Cos and Inter.

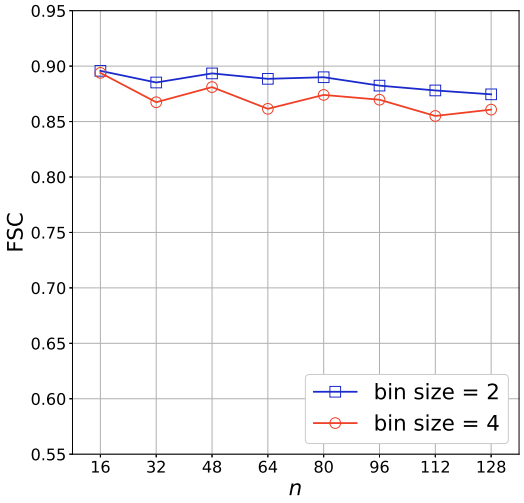
**Algorithm A4** Ensemble Learning with SNEFY-LDL

**Input**: Training set $\mathcal{S}^{\text{tr}} = \{(\boldsymbol{x}_i^{\text{tr}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{tr}}}) : i = 1, \cdots, N_{\text{tr}}\}$, test set $\mathcal{S}^{\text{te}} = \{(\boldsymbol{x}_i^{\text{te}}, \boldsymbol{\ell}_{\boldsymbol{x}_i^{\text{te}}}) : i = 1, \cdots, N_{\text{te}}\}$, and a LDL base learner training algorithm $\in \{\text{SA-BFGS}, \text{DF-LDL}, \text{LDL-SCL}, \text{LDL-LRR}\}$.
**Parameters**: The number of samples for training base learners $N_{\text{sample}} = 50$ and the number of base learners $N_{\text{base}} = 25$.
**Output**: The label distribution prediction performance scores measured by Cheby, Clark, Canb, KL, Cos and Inter on the test set $\mathcal{S}^{\text{te}}$.

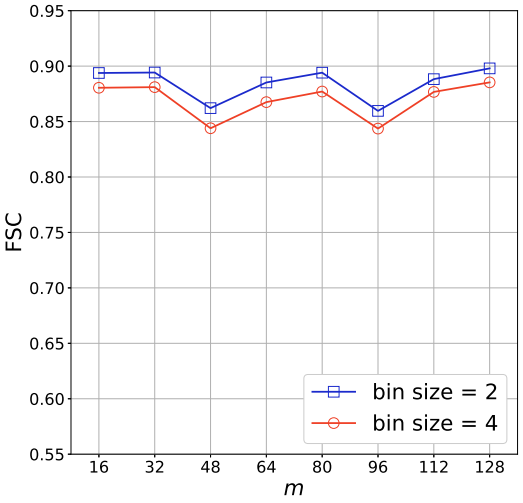
**for** each iteration $i \in \{1, \cdots, N_{\text{base}}\}$ **do**
    Randomly select $N_{\text{sample}}$ samples from the training set $\mathcal{S}^{\text{tr}}$;
    Train a LDL base learner $B_i$ with the selected samples;
**end for**
Train a SNEFY-LDL model with the training set $\mathcal{S}^{\text{tr}}$;
**for** each sample $\boldsymbol{x} \in \mathcal{S}^{\text{te}}$ **do**
    **for** each iteration $i \in \{1, \cdots, N_{\text{base}}\}$ **do**
        Predict $\boldsymbol{x}$'s label distribution vector with base learner $B_i$ as $\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}$;
        Evaluate the probability density value $\mathrm{P}(d\boldsymbol{\ell}|\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{\Theta})/d\boldsymbol{\ell}$ at $\boldsymbol{\ell} = \tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}$ as $p(\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}|\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{\Theta})$ with Eq. (6) and the trained SNEFY-LDL model;
    **end for**
    Predict $\boldsymbol{x}$'s label distribution vector $\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}$ as the weighted average of $\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}$, i.e., $\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}} = \frac{\sum_{i=1}^{N_{\text{base}}} p(\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}|\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{\Theta}) \tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}}{\sum_{i=1}^{N_{\text{base}}} p(\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}|\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{\Theta})}$;
**end for**
Evaluate the performance of the ensembled label distribution predictions $\tilde{\boldsymbol{\ell}}_{\boldsymbol{x}}^{(i)}$ on the test set $\mathcal{S}^{\text{te}}$ (as an average over all test samples) with the metrics of Cheby, Clark, Canb, KL, Cos and Inter defined in Eq. (22);
**return** the label distribution prediction performance scores measured by Cheby, Clark, Canb, KL, Cos and Inter.

# C PARAMETER SENSITIVITY STUDY

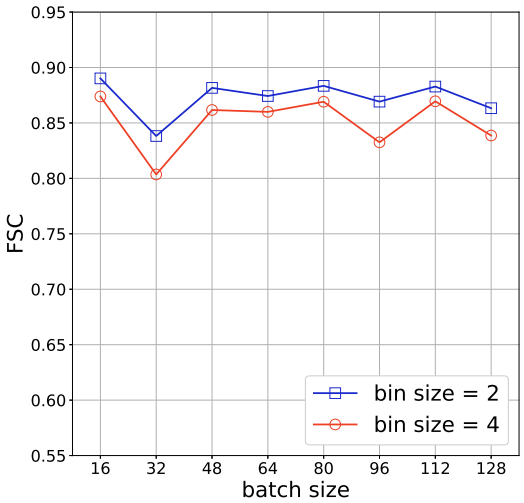
By choosing the *Natural Scene* dataset and the conformal prediction task, we take turns to study the sensitivity of SNEFY-LDL with regard to the four hyperparameters: $n$ and $m$, as well as the batch size and epoch number used for training, by varying the studied hyperparameter in a predefined range and fixing the remaining three as default values at each turn. For the conformal prediction task, the default values of $n$, $m$, batch size and epoch number are 64, 32, 64 and 100 respectively. Figure A1 plots the conformal prediction performance change of SNEFY-LDL measured by FSC with bin size equal to 2 and 4 when the values of hyperparameters vary in a range. From Figure A1, we can find that the performance of SNEFY-LDL remains relatively stable with the change of the four hyperparameters in most cases, except for the cases with $m = 48$ and 96 in Figure A1(b), as well as batch size $= 32$ and 96 in Figure A1(c), where the conformal prediction performance of SNEFY-LDL goes through a obvious drop.



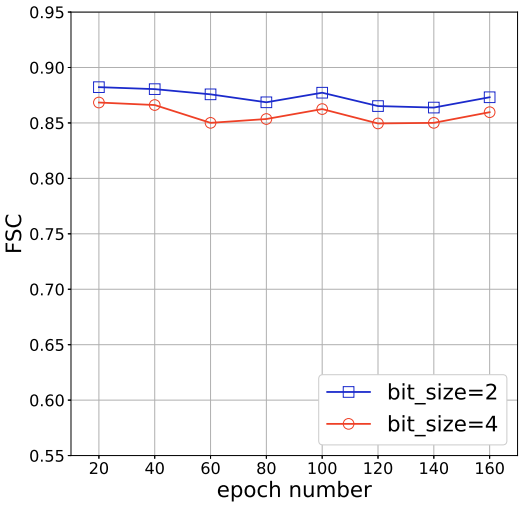

(a) $n$




(b) $m$




(c) batch size




(d) epoch number

Figure A1: The sensitivity of SNEFY-LDL with regard to the four hyperparameters: $n$, $m$, batch size and epoch number.