Making Reliable and Flexible Decisions in Long-tailed Classification

Bolian Li *li4468@purdue.edu*
Ruqi Zhang *ruqiz@purdue.edu*
Department of Computer Science, Purdue University
West Lafayette, IN 47907, USA

Reviewed on OpenReview: *https://openreview.net/forum?id=hMO8sT9qaD*

Abstract

Long-tailed classification is challenging due to its heavy imbalance in class probabilities. While existing methods often focus on overall accuracy or accuracy for tail classes, they overlook a critical aspect: certain types of errors can carry greater risks than others in real-world long-tailed problems. For example, misclassifying patients (a tail class) as healthy individuals (a head class) entails far more serious consequences than the reverse scenario. To address this critical issue, we introduce Making **R**eliable and **F**lexible **D**ecisions in **L**ong-tailed **C**lassification (RF-DLC), a novel framework aimed at reliable predictions in long-tailed problems. Leveraging Bayesian Decision Theory, we introduce an integrated gain to seamlessly combine long-tailed data distributions and the decision-making procedure. We further propose an efficient variational optimization strategy for the decision risk objective. Our method adapts readily to diverse utility matrices, which can be designed for specific tasks, ensuring its flexibility for different problem settings. In empirical evaluation, we design a new metric, False Head Rate, to quantify tail-sensitivity risk, along with comprehensive experiments on multiple real-world tasks, including large-scale image classification and uncertainty quantification, to demonstrate the reliability and flexibility of our method.[1]

1 Introduction

Real-world categorical data is often long-tailed distributed, where the data distributions are biased towards a few "head" classes and the "tail" classes have much fewer samples (Reed, 2001; Lin et al., 2014; Van Horn & Perona, 2017; Krishna et al., 2017; Liu et al., 2019b; Wang et al., 2020; Li et al., 2022). The long-tailed problem primarily stems from the inherent biases in the data collection process, which are challenging to avoid. Models trained on long-tailed data using conventional methods often exhibit notable performance drops compared to those trained on balanced datasets (Wang et al., 2022).

Existing approaches to long-tailed classification, such as re-weighting loss (Lin et al., 2017; Cao et al., 2019; Cui et al., 2019; Wu et al., 2020), logit adjustment (Menon et al., 2020; Ren et al., 2020; Hong et al., 2021), and knowledge transfer (Liu et al., 2019b; Xiang et al., 2020), usually focus on improving overall evaluation metrics or the metrics for tail classes (e.g., accuracy, AUC, F1-score, ... (Menon et al., 2013)). However, in real-world long-tailed data, the misprediction penalty between different classes often depends on their semantic meaning (He et al., 2024), and the ultimate objective is making optimal decisions rather than optimizing certain metrics. The probability of misclassifying tailed samples as head samples is very high in existing methods. For example, in disease detection where healthy individuals belong to head classes and patients belong to tail classes, the risk of classifying patients as healthy individuals is significantly larger than the reverse scenario (Yang et al., 2022). In autonomous driving, mispredicting tail classes like pedestrians or cyclists as head classes like vehicles will significantly increase the risk of accidents and injuries (Carranza-García et al., 2021).

[1] https://github.com/lblaoke/RF-DLC.

To enable reliable long-tailed classification and simultaneously integrate decision-making, we propose Making **R**eliable and **F**lexible **D**ecisions in **L**ong-tailed **C**lassification (RF-DLC), a general learning framework aimed at reliable predictions[2] on diverse realistic long-tailed problems. Specifically, we introduce the integrated gain from *Bayesian Decision Theory* (Robert et al., 2007; Berger, 2013), an important branch of cost-sensitive learning, which allows us to seamlessly incorporate decision risk and long-tailed data distributions in a single objective. While cost-sensitive learning has been mainly applied to standard classification (Elkan, 2001; He & Garcia, 2009; Thai-Nghe et al., 2010; Chung et al., 2016; Shu et al., 2019), its application to long-tailed classification remains under-explored. We also propose a variational optimization strategy to efficiently maximize the integrated gain w.r.t. model parameters. With "utility matrix" as a part of the objective, our method is flexible to many real-world tasks with different types of risks. In our empirical evaluations, we design a new metric, False Head Rate (FHR), to quantity the mispredictions from tail classes to head classes–a common source of high risk in long-tailed classification (Sengupta et al., 2016; Rahman et al., 2021; Yang et al., 2022).

The main contributions of this paper are summarized as follows:

- RF-DLC is the first to consider decision-making in long-tailed classification. Built upon Bayesian Decision Theory, RF-DLC enables optimal decision-making on long-tailed data.
- RF-DLC introduces a new objective called integrated gain, an efficient variational optimization strategy, and several utility matrices tailored for different long-tailed scenarios. All of these are directly derived from Bayesian Decision Theory, ensuring that they are integral to the model's function.
- RF-DLC is flexible and can be adapted to various tasks with diverse metrics. Users can adopt RF-DLC to many specific fields by re-designing utility matrices.
- We conduct comprehensive experiments to demonstrate that RF-DLC significantly improves decision-making while maintaining or improving traditional metrics such as accuracy and calibration.

2 Related Works

Long-tailed Classification. Previous methods mainly tackle long-tailed classification from the following aspects: i) adjusting data distributions to obtain balanced datasets, including over-sampling (Han et al., 2005), under-sampling (Liu et al., 2008), and data augmentation (Chu et al., 2020; Kim et al., 2020; Liu et al., 2020)[3]; ii) re-balancing the importance of different classes in loss functions, including re-weighting loss (Lin et al., 2017; Mahajan et al., 2018; Cao et al., 2019; Cui et al., 2019; Menon et al., 2020; Wu et al., 2020) and logit adjustment (Menon et al., 2020; Ren et al., 2020; Hong et al., 2021); iii) applying heterogeneous model architectures to handle head and tail samples in different ways, including OLTR (Liu et al., 2019b), LFME (Xiang et al., 2020), RIDE (Wang et al., 2020), TLC (Li et al., 2022), and SRepr (Nam et al., 2023). Other attempts explore broader long-tailed classifications, including non-uniform testing distributions (Zhang et al., 2022), outlier samples (Wang et al., 2022; Bai et al., 2022), and partial-labeled datasets (Hong et al., 2022). Our method is significantly different from existing methods, targeting at decision-making and asymmetric misprediction risks, which are important problems in realistic long-tailed data.

Bayesian Decision Theory and Cost-sensitive Learning. Robert et al. (2007); Berger (2013) have comprehensively introduced Bayesian Decision Theory, which enables optimal decisions under diverse problem settings (via the choices of utility functions). For its adaptation to deep neural networks, multiple ways have been explored: i) loss-calibrated variational inference, including Loss-calibrated EM (Lacoste-Julien et al., 2011), LCVB (Jaiswal et al., 2020), and variational inference on continuous utilities (Kuśmierczyk et al., 2019); ii) loss-calibrated expectation propagation (Morais & Pillow, 2022). Bayesian Decision Theory is also one of the main methodologies to solve cost-sensitive learning (Elkan, 2001; Ling & Sheng, 2008; Chung et al., 2016), which focuses on decision-making under heterogeneous misprediction costs. Standard

[2]The words "prediction" and "decision" are used interchangeably throughout this paper.

[3]These methods fall behind the SOTA of long-tailed classification significantly, and thus are not compared in the experiments.

cost-sensitive learning under expected loss minimization (Pires et al., 2013) can overlook the fact that long-tailed training data distribution may be distributed differently from the testing distribution. Our method is built upon Bayesian Decision Theory and focuses on the specific problem of long-tailed classification, which has not been explored in previous works. We include a detailed comparison with cost-sensitive learning in Appendix F.3.

3 Background

This section provides the necessary background knowledge on long-tailed classification and Bayesian Decision Theory for understanding our method.

Long-tailed Distributions. In long-tailed categorical data, the training and testing sets follow different distributions (Moreno-Torres et al., 2012; Cui et al., 2019; Liu et al., 2019b). Specifically, the training data is distributed in a descending manner over categories in terms of class probability:

$$p(y_1 = k_1) \geq p(y_2 = k_2), \quad \text{if } k_1 \leq k_2, \tag{1}$$

while most existing works (Cui et al., 2019; Cao et al., 2019; Wang et al., 2020; Li et al., 2022) assume that the testing data distribution is uniform over categories: $p(y = k_1) = p(y = k_2)$ for all pairs of (k_1, k_2). The distributional shift between training and testing data makes long-tailed problems extremely challenging.

Bayesian Decision Theory. To make optimal decisions in diverse problem settings, Bayesian Decision Theory provides a principled framework (Robert et al., 2007; Berger, 2013). In the supervised setting, for a dataset $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{Y}\} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ and a $\boldsymbol{\theta}$-parameterized model, we denote the likelihood and prior to be $\prod_{i=1}^N p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$ respectively. The posterior distribution is then $p(\boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\theta}) \prod_{i=1}^N p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) / \prod_{i=1}^N p(y_i|\boldsymbol{x}_i)$. We further assume a *decision gain* $g(d|\boldsymbol{x}, \theta)$ to quantify the utility gained by choosing decision d for input $\boldsymbol{x}$ when the model with parameters $\boldsymbol{\theta}$ controls the mapping from $\boldsymbol{x}$ to y. Subsequently, the *posterior expected gain* for an input $\boldsymbol{x}$ is defined as

$$G(d|\boldsymbol{x}, \mathcal{D}) := \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D})} g(d|\boldsymbol{x}, \boldsymbol{\theta}), \tag{2}$$

where we average over all possible models weighted by their posterior probabilities. Previous works (Lacoste-Julien et al., 2011; Cobb et al., 2018) naively maximize Eq. 2 w.r.t. $\boldsymbol{\theta}$ to obtain a decision-calibrated posterior $q(\boldsymbol{\theta})$, which does not consider non-uniform data distribtions.

4 Methodology

In this section, we introduce RF-DLC, a framework aimed at optimal decision-making for long-tailed data, grounded in Bayesian Decision Theory. The conventional posterior expected gain in Eq. 2 implicitly assumes that both training and testing data share the same distribution, which is violated in long-tailed problems. To address this challenge, we adopt the *integrated gain* from Bayesian Decision Theory, which incorporates data distributions into the posterior expected gain:

$$G(\boldsymbol{d}) := \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \sim p(\boldsymbol{x}, y)} G(\boldsymbol{d}|\boldsymbol{X}, \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \sim p(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D})} G(\boldsymbol{d}|\boldsymbol{X}, \boldsymbol{\theta}), \tag{3}$$

where $G(\boldsymbol{d}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{i=1}^N g(d_i|\boldsymbol{x}_i, \boldsymbol{\theta})$ is the decision gain over the entire dataset and $\boldsymbol{d} = [d_1, ..., d_N]^T$ is the decision vector. In the long-tailed setting, we naturally want the model to fit the testing distribution. Therefore, the integrated gain used in our method is defined as:[4]

$$G(\boldsymbol{d}) := \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \sim p_{\text{test}}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D})} G(\boldsymbol{d}|\boldsymbol{X}, \boldsymbol{\theta}). \tag{4}$$

Eq. 4 is our training objective, integrating the data distributions and the decision gain into a single function. To use this objective in long-tailed data, there are four main challenges left to address: i) Section 4.1: how to compute the decision gain $g(d|\boldsymbol{x}, \boldsymbol{\theta})$? ii) Section 4.2: how to fit the long-tailed training distributions? iii) Section 4.3: how to learn the intractable posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$? iv) Section 4.4: how to make optimal decisions during testing?

[4]The notion $p_{\text{test}}(\boldsymbol{x}, y)$ is the testing distribution and will be discussed in Section 4.2.

4.1 How to Design the Decision Gain?

To quantify the utility of making a decision, we define the decision gain as:

$$g(d|\boldsymbol{x}, \boldsymbol{\theta}) := \prod_{y'} p(y'|\boldsymbol{x}, \boldsymbol{\theta})^{u(y',d)}. \tag{5}$$

Here, $u(y', d)$ refers to the *utility function* which scores the decision d when the true label is y'. The utility values play a critical role in re-weighting the likelihoods of different decisions to prioritize particular mispredictions and allow us to encode human knowledge and expertise tailored to specific tasks.

The theoretical foundation of utility function has been comprehensively studied in Robert et al. (2007); Berger (2013). For example, Chapter 2.2 of Robert et al. (2007) guarantees the existence of utility functions with rational decision-makers. Note that our design of the decision gain differs from previous work like Cobb et al. (2018), which uses $g'(d|\boldsymbol{x}, \boldsymbol{\theta}) := \sum_{y'} U_{y',d} \cdot p(y'|\boldsymbol{x}, \boldsymbol{\theta})$. Both definitions achieve the goal of averaging the predictive probability $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ weighted by the utility values. However, Eq. 5 has two advantages: i) Stability: Eq. 5 is more stable for training. After taking the logarithm, Eq. 5 becomes $\sum_{y'} U_{y',d(\boldsymbol{x})} \cdot \log p(y'|\boldsymbol{x}, \boldsymbol{\theta})$ which is a weighted average of the log probabilities, while Cobb et al. (2018) becomes $\log \sum_{y'} U_{y',d(\boldsymbol{x})} \cdot p(y'|\boldsymbol{x}, \boldsymbol{\theta})$, which is a log of a weighted average of the probabilities and not commonly used in classification problems. ii) Flexibility: Eq. 5 allows for more general and flexible utility values whereas Cobb et al. (2018) requires the utility matrix $\boldsymbol{U}$ to be positive definite (otherwise we may not be able to compute the logarithm). Due to these reasons, we use Eq. 5.

Once we have defined the decision gain $g(d|\boldsymbol{x}, \boldsymbol{\theta})$, the next step is designing the utility function $u(y', d)$. For classification tasks, we can employ a utility matrix $\boldsymbol{U}$:

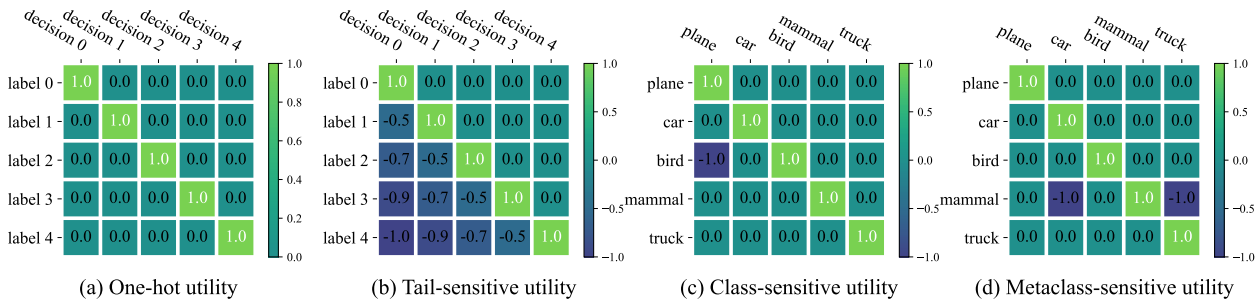
$$\boldsymbol{U} := \begin{bmatrix} u(0,0) & u(0,1) & \cdots & u(0,|\mathcal{Y}|) \\ u(1,0) & u(1,1) & \cdots & u(1,|\mathcal{Y}|) \\ \vdots & \vdots & \ddots & \vdots \\ u(|\mathcal{Y}|,0) & u(|\mathcal{Y}|,1) & \cdots & u(|\mathcal{Y}|,|\mathcal{Y}|) \end{bmatrix}, \tag{6}$$

where $|\mathcal{Y}|$ is the number of classes and $U_{ij} = u(y = i, d = j)$ is the utility score assigned to the case of predicting class i as j. The utility matrix serves as the task-specific knowledge and is pre-defined before training. Users can assign negative values to discourage certain mispredictions and positive values to encourage the desired ones, which reflects their knowledge about specific tasks. To illustrate how to design utility matrices for different long-tailed problem settings, we provide practical examples in Fig. 1. For an in-depth discussion on utility designing, please refer to Chapter 2 of Robert et al. (2007).

Standard Classification. Standard classification on long-tailed data can be regarded as a special case in our framework, where the overall accuracy is the most decisive metric in evaluation. In this case, the focus lies solely on determining whether the decision aligns with the ground truth (i.e., $y = d$). As shown in Fig. 1(a), a simple one-hot utility can be defined by $u(y', d) = \mathbb{1}\{y' = d\}$, which corresponds to the standard accuracy metric.

Tail-sensitive Classification. Tail-class samples often have high importance due to their scarcity. Mispredictions from tail classes to head classes usually induce severe consequences in real-world tasks, such as activity recognition (Rahman et al., 2021) and medical images (Yang et al., 2022). In these domains, dangerous actions and illnesses are often scarce yet profoundly harmful if neglected. Besides, the lack of training samples in tail classes has been empirically proved to be the bottleneck of classification performance (Li et al., 2022). Therefore, the ratio of false head samples in evaluation can often reflect a model's real-world potential. To this end, a tail-sensitive utility matrix can be defined by adding extra penalties on false head mispredictions, as shown in Fig. 1(b). The tail-sensitive utility matrix encourages the model to predict uncertain samples as tail rather than head, without affecting correct predictions made with confidence.

Class/Meta-class-sensitive Classification. In certain applications where semantic differences exist between different categories, preventing mispredictions between specific (meta) classes becomes crucial, regardless of their class probabilities within the long-tailed distribution. For example, object detection systems at



(a) One-hot utility (b) Tail-sensitive utility (c) Class-sensitive utility (d) Metaclass-sensitive utility

Figure 1: Examples of utility matrices, designed for (a) standard and (b) tail-sensitive classifications, along with (c) class-sensitive classification for bird-plane misprediction and (d) meta-class-sensitive classification for mammal-vehicle misprediction. The entries in the matrix reflect the risk levels (-1 is the most risky) and can be flexibly assigned based on task requirements.

airports must avoid misclassifying birds as planes to ensure flight safety (Shi et al., 2021), and autonomous driving systems must prevent misclassifying mammals on the road as vehicles (both are meta-classes including various animal and vehicle types) to ensure driving safety (Yudin et al., 2019). In these contexts, we provide two examples in Fig. 1(c) and (d) to illustrate how utility values can be assigned to prevent the specific mispredictions that are of primary concern.

These utility matrices enable us to customize decision-making based on the particular objectives and challenges in different long-tailed classification scenarios, improving the reliability and flexibility in real-world applications.

4.2 How to handle the Distribution Shift?

In long-tailed classification, there exists a distribution shift between the training (long-tailed) and testing (uniform) sets. We denote the data distributions of the training and testing sets as $p_{\text{train}}(\boldsymbol{x}, y)$ and $p_{\text{test}}(\boldsymbol{x}, y)$ respectively. Since all models are aimed to perform well on the testing set, the data distribution in Eq. 4 should be the testing distribution $p_{\text{test}}(\boldsymbol{x}, y)$. To address the discrepancy between the training and testing distributions, importance sampling (Kloek & Van Dijk, 1978) is adopted:

$$\begin{aligned}\mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{test}}(\boldsymbol{x},y)}\Psi(\boldsymbol{x},y) = \int p_{\text{test}}(\boldsymbol{x},y)\Psi(\boldsymbol{x},y)d(\boldsymbol{x},y) &= \int p_{\text{train}}(\boldsymbol{x},y)\frac{p_{\text{test}}(\boldsymbol{x},y)}{p_{\text{train}}(\boldsymbol{x},y)}\Psi(\boldsymbol{x},y)d(\boldsymbol{x},y) \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim p_{\text{train}}(\boldsymbol{x},y)}\frac{p_{\text{test}}(\boldsymbol{x},y)}{p_{\text{train}}(\boldsymbol{x},y)}\Psi(\boldsymbol{x},y),\end{aligned} \tag{7}$$

where $\Psi(\boldsymbol{x}, y)$ denote any possible function, which will be specified in Section 4.3. The ratio $p_{\text{test}}(\boldsymbol{x}, y)/p_{\text{train}}(\boldsymbol{x}, y)$ explicitly shows the discrepancy between training and testing data. One common assumption in long-tailed data is that distributional differences only exist between classes (Hong et al., 2021). Formally, this assumption can be expressed as follows:

Assumption 1 (Intra-class Consistency)**.** *The distributional differences only exist between classes. Given a fixed class label, the data distributions are the same for training and testing data:* $p_{train}(\boldsymbol{x}|y) = p_{test}(\boldsymbol{x}|y)$.[5]

Based on Assumption 1, the discrepancy ratio can be further simplified:

$$\frac{p_{\text{test}}(\boldsymbol{x},y)}{p_{\text{train}}(\boldsymbol{x},y)} = \frac{p_{\text{test}}(y)p_{\text{test}}(\boldsymbol{x}|y)}{p_{\text{train}}(y)p_{\text{train}}(\boldsymbol{x}|y)} = \frac{p_{\text{test}}(y)}{p_{\text{train}}(y)}, \tag{8}$$

which only depends on the class probabilities of training and testing data. Given that the testing set is assumed to be uniform, the probability $p_{\text{test}}(y)$ would be a constant, and thus the ratio is equivalent to:

$$\frac{p_{\text{test}}(y)}{p_{\text{train}}(y)} \propto \frac{1}{p_{\text{train}}(y)} \propto \frac{1}{f(n_y)}, \tag{9}$$

[5]This assumption is widely adopted in previous works (Hong et al., 2021; Ren et al., 2022).

where $f(\cdot)$ is an increasing function and n_y refers to the number of samples in the class y. We introduce the notion of $f(n_y)$ because the class probability only depends on the number of samples in this class. Many existing re-weighting methods in long-tailed classification can be regarded as special instances of $f(\cdot)$. For example, $f(n_y) = n_y^\gamma$ is the most conventional choice with a sensitivity factor γ to control the importance of head classes (Huang et al., 2016; Wang et al., 2017; Pan et al., 2021); $f(n_y) = (1-\beta^{n_y})/(1-\beta)$ is the effective number which considers data overlap (Cui et al., 2019). A detailed analysis of the choice of $f(\cdot)$ in RF-DLC is conducted in Section 5.6.

4.3 How to Learn the Posterior Distribution?

The posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ in the integrated gain (Eq. 4) is generally intractable. To learn the posterior, we use a variational distribution $q(\boldsymbol{\theta})$ to approximate the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$. Specifically, we find a tractable lower bound for the logarithm of the integrated gain:

$$\begin{aligned}
&\log G(\boldsymbol{d}=\boldsymbol{Y}) = \log \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})} G(\boldsymbol{d}=\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) \\
&\geq \sum_{i=1}^{N} \mathbb{E}_{(\boldsymbol{x}_i,y_i)\sim p_{\text{test}}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \left[\sum_{y'} U_{y',y_i} \cdot \log p(y'|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) + C \\
&\approx \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{1}{f(n_{y_i})} \left[\sum_{y'} U_{y',y_i} \cdot \log p(y'|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) + C \\
&:= L(q, \boldsymbol{d}=\boldsymbol{Y}),
\end{aligned} \tag{10}$$

where $C = -\log p(\boldsymbol{Y}|\boldsymbol{X})$ is a constant. Here, the second equation is obtained using Jensen's inequality and the third equation is obtained using the importance sampling discussed in Section 4.2 as well as one-sample Monte Carlo (MC) approximation of the expectation over data points $(\boldsymbol{x}_i, y_i)$. The detailed derivation of this lower bound is similar to previous works (Lacoste-Julien et al., 2011; Cobb et al., 2018) and is proved in Appendix C. In the lower bound, we set $d_i = y_i$ since the true label is considered to be the optimal decision at the training stage. By maximizing $L(q, \boldsymbol{d}=\boldsymbol{Y})$ w.r.t. q, we obtain a good approximation for the posterior $p(\boldsymbol{\theta}|\mathcal{D})$.

Relationship with Standard Variational Inference. On the relationship between our method and standard variational inference (Jordan et al., 1999), a clear similarity is the notion of variational distribution $q(\boldsymbol{\theta})$, which is expected to approximate a posterior distribution. However, due to the distributional shift in long-tailed data, we should also specify whether the posterior distribution is based on training or testing distribution, which are denoted as $p_{\text{train}}(\boldsymbol{\theta}|\mathcal{D})$ and $p_{\text{test}}(\boldsymbol{\theta}|\mathcal{D})$ respectively. We show below that the lower bound L contains the objective for standard variational inference:

$$L(q, \boldsymbol{d}=\boldsymbol{Y}) \approx \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log G(\boldsymbol{d}=\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) - \mathbf{KL}(q(\boldsymbol{\theta})||p_{\text{test}}(\boldsymbol{\theta}|\mathcal{D})), \tag{11}$$

where the approximation is due to the use of MC approximation. If further assuming the utility matrix to be one-hot (i.e., $U_{y,d} = \mathbb{1}\{y=d\}$), the equation can be further simplified as:

$$L(q, \boldsymbol{d}=\boldsymbol{Y}) \approx -\mathbf{KL}(q(\boldsymbol{\theta})||p_{\text{test}}(\boldsymbol{\theta}|\mathcal{D})). \tag{12}$$

The proof is provided in Appendix D.

Particle-based Variational Distribution. To pursue the efficiency of posterior inference, we construct the variational distribution using particles (Liu & Wang, 2016; D'Angelo & Fortuin, 2021):

$$q(\boldsymbol{\theta}) = \sum_{j=1}^{M} w_j \cdot \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_j), \tag{13}$$

where $\{w_j\}_{j=1}^M$ are normalized weights which hold $\sum_{j=1}^M w_j = 1$, and $\delta(\cdot)$ is the Dirac delta function. The "particles" $\{\boldsymbol{\theta}_j\}_{j=1}^M$ are implemented through ensemble models. Empirical studies have previously explored

ensemble methods in the context of long-tailed data (Wang et al., 2020; Li et al., 2022). Our framework provides a theoretical foundation for these approaches in long-tailed problems: due to the scarcity of tailed data, there is not enough evidence to support a single solution, leading to many equally good solutions (which give complementary predictions) in the loss landscape. Therefore, estimating the full posterior distribution is essential to get a comprehensive characterization of the solution space. Particle optimization reduces the cost of Bayesian inference and is more efficient than variational inference (Blundell et al., 2015) and Markov chain Monte Carlo (MCMC) (Brooks et al., 2011), especially on the high-dimensional and multimodal deep neural network posteriors.

Repulsive Regularization. The integrated gain optimization in Eq. 10 includes a regularization term $\mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}))$, which keeps q to be close to the prior $p(\boldsymbol{\theta})$. If we assume the prior $p(\boldsymbol{\theta})$ to be a Gaussian distribution, the regularization can be extended to:

$$\mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) = \lambda \int_{\Theta} ||\boldsymbol{\theta}||^2 \cdot q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \int_{\Theta} q(\boldsymbol{\theta}) \log q(\boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{\lambda}{M} \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||^2 - H(\boldsymbol{\theta}), \tag{14}$$

where λ is a constant, Θ is the parameter space and $H(\boldsymbol{\theta})$ is the entropy of $\boldsymbol{\theta}$. The first term (L_2-regularization) prevents the model from over-fitting and the second term (entropy) applies a *repulsive force* to individual models to promote their diversity, pushing the particles away from each other (Liu & Wang, 2016; D'Angelo & Fortuin, 2021). To make the entropy term computable, we introduce a simple approximation: $H(\boldsymbol{\theta}) \approx \frac{1}{2} \log |\hat{\Sigma}_{\boldsymbol{\theta}}|$, where $\hat{\Sigma}_{\boldsymbol{\theta}}$ is the covariance matrix estimated by particles. Other entropy approximations can also be used. By the technique of SWAG-diagonal covariance (Maddox et al., 2019), the covariance matrix can then be directly computed by: $\hat{\Sigma}_{\boldsymbol{\theta}} = diag(\overline{\boldsymbol{\theta}^2} - \overline{\boldsymbol{\theta}}^2)$. Overall, the regularization term is a combination of L_2 weight decay and the repulsive force and the final form of this regularization term is:

$$\mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) \approx \frac{\lambda}{M} \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||^2 - \frac{1}{2} \sum_{k} \log (\overline{\boldsymbol{\theta}^2} - \overline{\boldsymbol{\theta}}^2)_k, \tag{15}$$

which is different from existing diversity regularization used in long-tailed classification (Wang et al., 2020; Li et al., 2022). The repulsive regularization in our method is naturally derived from the integrated gain optimization, with strong theoretical motivation.

4.4 How to make optimal decisions during testing?

Following the standard Bayesian Decision Theory, we maximize the logarithm of the posterior expected gain $G(d^\star|\boldsymbol{x}^*, \mathcal{D})$ in Eq.(2) for each testing input $\boldsymbol{x}^*$ to obtain optimal decisions:

$$d^\star = \arg\max_d \log G(d|\boldsymbol{x}^*, \mathcal{D}) \approx \arg\max_d \sum_{j=1}^{M} \sum_{y'} U_{y',d} \cdot \log p(y'|\boldsymbol{x}^*, \boldsymbol{\theta}_j). \tag{16}$$

where the approximation is due to $q(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta}|\mathcal{D})$. Importantly, for symmetric utility matrices (e.g., one-hot utility in Fig. 1(a)), Eq. 16 can be further simplified to $d^\star \approx \arg\max_d \sum_{j=1}^{M} \log p(d|\boldsymbol{x}, \boldsymbol{\theta}_j)$, which aligns with the standard ensemble models.

In summary, the integrated gain enables our method to simultaneously consider the posterior distribution, decision-making (utility matrix), and data distribution. It provides a principled way to address more realistic problems on the long-tailed data. The proposed RF-DLC is summarized in Algorithm 1.

5 Experiments

In this section, we show the experimental results to demonstrate the effectiveness of our method. We use CIFAR10/100-LT (Cui et al., 2019), ImageNet-LT (Liu et al., 2019b), and iNaturalist (Van Horn et al., 2018) as the long-tailed datasets, and compare our method with multiple long-tailed baselines. Detailed implementation is summarized in Appendix E.

Algorithm 1: RF-DLC

Inputs: Dataset $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{Y}\} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, initial particles $\{\boldsymbol{\theta}_j\}_{j=1}^M$, utility matrix $\boldsymbol{U}$, the step size η;
Results: Final particles $\{\boldsymbol{\theta}_j^*\}_{j=1}^M$;

for *each iteration* **do**
  $\boldsymbol{X}_\Xi, \boldsymbol{Y}_\Xi \leftarrow$ A mini-batch sampled from $\mathcal{D}$;
  $L_\Xi(q, \boldsymbol{d} = \boldsymbol{Y}_\Xi) \leftarrow$ Loss computed by Eq. 10;
  for $j = 1, ..., M$ **do**
    $\boldsymbol{\theta}_j \leftarrow \boldsymbol{\theta}_j - \eta \cdot \nabla_{\boldsymbol{\theta}_j} L_\Xi$; `/* Update` $q(\boldsymbol{\theta})$ `*/`
  end
end

Table 1: False Head Rate evaluation for tail-sensitive long-tailed classification (%). Three datasets and three tail region settings are considered. Our method consistently outperforms all baselines across all settings.

Dataset	Tail Ratio	CE	CB Loss (Cui et al., 2019)	LDAM (Cao et al., 2019)	RIDE (Wang et al., 2020)	TLC (Li et al., 2022)	RF-DLC
CIFAR10-LT	25%	21.10 ± 0.43	14.84 ± 0.93	10.05 ± 1.01	8.94 ± 0.66	10.42 ± 0.64	$\mathbf{4.99 \pm 0.32}$
	50%	37.87 ± 0.57	27.98 ± 1.44	19.64 ± 1.66	17.80 ± 1.39	20.27 ± 0.77	$\mathbf{11.76 \pm 0.29}$
	75%	48.75 ± 1.39	33.93 ± 1.60	21.37 ± 2.10	19.77 ± 3.20	22.24 ± 1.53	$\mathbf{11.01 \pm 1.28}$
	average	35.91 ± 0.54	25.58 ± 1.27	17.02 ± 1.56	15.50 ± 1.68	17.64 ± 0.93	$\mathbf{9.25 \pm 0.49}$
CIFAR100-LT	25%	45.53 ± 1.54	24.88 ± 0.34	21.22 ± 0.99	18.83 ± 0.70	21.18 ± 0.54	$\mathbf{15.39 \pm 0.57}$
	50%	73.03 ± 1.59	48.41 ± 1.24	43.04 ± 1.18	39.50 ± 1.53	41.15 ± 0.55	$\mathbf{31.34 \pm 0.55}$
	75%	91.30 ± 1.24	74.38 ± 1.47	65.62 ± 1.31	62.01 ± 2.70	61.34 ± 1.03	$\mathbf{49.51 \pm 1.45}$
	average	69.95 ± 1.40	49.22 ± 0.83	43.29 ± 1.04	40.11 ± 1.62	41.22 ± 0.55	$\mathbf{32.08 \pm 0.78}$
ImageNet-LT	25%	3.99 ± 0.08	3.66 ± 0.17	4.17 ± 0.19	3.62 ± 0.18	3.47 ± 0.13	$\mathbf{2.70 \pm 0.09}$
	50%	12.77 ± 0.29	11.80 ± 0.12	12.73 ± 0.28	11.42 ± 0.27	11.49 ± 0.13	$\mathbf{9.68 \pm 0.25}$
	75%	30.99 ± 0.40	29.39 ± 0.28	29.90 ± 0.41	26.92 ± 0.33	27.12 ± 0.13	$\mathbf{24.42 \pm 0.19}$
	average	15.92 ± 0.17	14.95 ± 0.15	15.60 ± 0.21	13.99 ± 0.24	14.03 ± 0.09	$\mathbf{12.27 \pm 0.08}$

5.1 Tail-sensitive Long-tailed Classification with False Head Rate

As discussed in Section 4.1, mispredictions from tail classes to head classes generally pose higher risks and quantifying the likelihood of such occurrences is crucial. Inspired by the false positive rate, we define the *False Head Rate* (FHR) as follows:

$$FHR = \frac{|\mathcal{P}_{\text{head}} \cap \mathcal{G}_{\text{tail}}|}{|\mathcal{G}_{\text{tail}}|}, \tag{17}$$

where $\mathcal{G}_{\text{tail}}$ is the set of samples that are labeled as tail classes and $\mathcal{P}_{\text{head}}$ is the set of samples that are predicted as head classes. To consider different tail regions, we select the last 25%, 50%, and 75% classes as tail classes, respectively. We apply the tail-sensitive utility in Fig. 1(b) to our method. From Table 1, we observe substantial improvements over all baselines across all settings, especially on the relatively small CIFAR datasets, which means that the False Head Rate is more challenging on small datasets.

5.2 Class-sensitive Long-tailed Classification

We further try the (meta) class sensitive cases on CIFAR10-LT, including bird-plane detection (Shi et al., 2021) and vehicle-mammal detection (Yudin et al., 2019), as discussed in Section 4.1. The utility matrices used for these two tasks are Fig. 1 (c) and (d) respectively, and their full formats are shown in Appendix B. We also design two corresponding metrics for accurate evaluation, and list the results in Table 2. We again observe that our method improves significantly on the metrics with negligible drops in standard accuracy. These real-world tasks show the importance of taking the decision loss into account and also demonstrate the flexibility of our method which is compatible with different utilities, leading to better performance for different types of tasks.

Table 2: Class-sensitive long-tailed classification on CIFAR10-LT, comparing the baseline one-hot utility with class/metaclass-sensitive utility. The specifically designed utilities can effectively improve tailored metrics with negligible drops in standard overall accuracy.

Task	**Metric**	**Evaluation Score (%)** ↓	**ACC (%)** ↑
bird-plane detection	$R_{\text{plane}} = \frac{\lvert\mathcal{P}_{\text{plane}} \cap \mathcal{G}_{\text{bird}}\rvert}{\lvert\mathcal{G}_{\text{bird}}\rvert}$	5.10 → 3.80 (−25.5%)	84.11 → 83.84 (−0.3%)
vehicle-mammal detection	$R_{\text{mammal}} = \frac{\lvert\mathcal{P}_{\text{vehicle}} \cap \mathcal{G}_{\text{mammal}}\rvert}{\lvert\mathcal{G}_{\text{vehicle}}\rvert}$	1.40 → 0.35 (−75.5%)	84.11 → 83.83 (−0.3%)

Table 3: Top-1 overall accuracy and tail-class accuracy on standard long-tailed classification (%). Our method outperforms all baselines on both metrics.

Method	**CIFAR10-LT**		**CIFAR100-LT**		**ImageNet-LT**	
	All	Tail	All	Tail	All	Tail
LA† (Menon et al., 2020)	77.67	-	43.89	-	55.11	-
ACE† (Cai et al., 2021)	81.2	-	49.4	23.5	54.7	-
SRepr† (Nam et al., 2023)	82.06 ± 0.01	-	47.81 ± 0.02	23.31 ± 0.11	52.12 ± 0.06	32.14 ± 0.41*
CE	73.65 ± 0.39	58.51 ± 0.62	38.82 ± 0.52	10.62 ± 1.23	47.80 ± 0.15	44.03 ± 0.24
CB Loss (Cui et al., 2019)	77.62 ± 0.69	68.73 ± 1.52	42.24 ± 0.41	20.50 ± 0.51	51.70 ± 0.25	48.29 ± 0.41
LDAM (Cao et al., 2019)	80.63 ± 0.69	77.14 ± 1.61	43.13 ± 0.67	23.50 ± 1.28	51.04 ± 0.21	47.21 ± 0.22
RIDE (Wang et al., 2020)	83.11 ± 0.52	79.62 ± 1.56	48.99 ± 0.44	28.78 ± 1.52	54.32 ± 0.54	50.74 ± 0.62
TLC (Li et al., 2022)	79.70 ± 0.65	76.39 ± 0.98	48.75 ± 0.16	28.40 ± 0.72	55.03 ± 0.34	51.56 ± 0.35
RF-DLC (ours)	$\mathbf{83.75 \pm 0.17}$	$\mathbf{82.33 \pm 1.16}$	$\mathbf{50.24 \pm 0.70}$	$\mathbf{30.34 \pm 1.49}$	$\mathbf{55.73 \pm 0.17}$	$\mathbf{51.98 \pm 0.40}$

5.3 Standard Long-tailed Classification

We evaluate the overall accuracy and tail-class accuracy. The results are shown in Table 3, where † means the results are directly copied from the original papers and $*$ is due to a different setting of tail classes[6]. We apply the one-hot utility in our method. In particular, Our method significantly outperforms other baselines on the crucial tailed data. The results on iNaturalist are summarized in appendix F. These results demonstrate the reliability of our method in standard long-tailed classification.

5.4 Uncertainty Quantification

In our method, the predictive uncertainty can be naturally obtained by the entropy of predictive distribution (Malinin & Gales, 2018). For the compared uncertainty estimation algorithms, MCP is a trivial baseline that obtains uncertainty scores from the maximum value of softmax distribution (Hendrycks & Gimpel, 2017); evidential uncertainty is rooted in the subjective logic (Jsang, 2018), and is introduced to long-tailed classification by Li et al. (2022). We evaluate the three uncertainty algorithms in Table 4. Our Bayesian predictive uncertainty outperforms the other two and has a remarkable advantage on the ECE metric, demonstrating the superiority of using principled Bayesian uncertainty quantification. The uncertainty results separated by class regions are in Appendix F.2, where the superiority of Bayesian predictive uncertainty persist on all class regions.

5.5 Medical Image Classification

Our method targets at real-world applications where long-tail problems exists. We also conduct a long-tailed medical image classification experiments in Table 5, where our RF-DLC successfully recognize different disease types and outperforms compared baselines. The experiments are based on ResNet32, and the number of particles is set to 3. The DermaMNIST dataset (Yang et al., 2023) is originally an imbalanced classification dataset with the imbalance ratio to be around 60. The images are resized to 32×32, and other hyperparameters are the same as our CIFAR experiments.

5.6 Ablation Studies

Effect of utility. The effect of tail-sensitive utility is shown in Table 6. We compare the one-hot and tail-sensitive utilities in terms of False Head Rate and standard overall accuracy. By applying the tail-sensitive utility, the performances on FHR can be significantly improved (18.00%) with a negligible drop in standard

[6]SRepr (Nam et al., 2023) has different ImageNet settings from other baselines, with fewer samples in the tailed classes. Therefore, their tail-class accuracy is much lower.

Table 4: Calibration evaluation of different uncertainty algorithms. The Bayesian predictive uncertainty adopted in our method outperforms other algorithms.

Uncertainty Algorithm	CIFAR10-LT		CIFAR100-LT		ImageNet-LT	
	AUC (%) ↑	ECE (%) ↓	AUC (%) ↑	ECE (%) ↓	AUC (%) ↑	ECE (%) ↓
MCP (Hendrycks & Gimpel, 2017)	79.98 ± 0.10	14.33 ± 0.37	80.48 ± 0.51	23.75 ± 0.51	84.02 ± 0.24	18.35 ± 0.12
Evidential (Li et al., 2022)	83.20 ± 0.59	13.24 ± 0.55	77.37 ± 0.33	21.64 ± 0.47	81.45 ± 0.13	15.29 ± 0.12
Bayesian (RF-DLC)	$\mathbf{86.83 \pm 0.68}$	$\mathbf{9.84 \pm 0.17}$	$\mathbf{81.24 \pm 0.25}$	$\mathbf{10.35 \pm 0.28}$	$\mathbf{84.45 \pm 0.09}$	$\mathbf{8.72 \pm 0.13}$

Table 5: Experimental results on long-tailed medical image classification, evaluated on DermaMNIST (Yang et al., 2023). RF-DLC outperforms other baselines on this real-world application.

Method	ACC (%) ↑				AUC (%) ↑	ECE (%) ↓	FHR (%) ↓			
	All	Head	Med	Tail			avg	25%	50%	75%
CE	69.50	**85.51**	50.17	38.51	82.30	24.31	47.00	30.68	52.71	57.61
Focal Loss (Lin et al., 2017)	71.75	85.08	51.60	40.57	79.14	16.53	40.08	25.68	46.14	48.41
CB Loss (Cui et al., 2019)	73.89	84.41	53.27	43.99	75.15	27.81	36.73	26.73	38.14	45.32
LDAM (Cao et al., 2019)	72.82	82.62	53.71	43.15	81.66	17.61	39.35	24.48	47.33	46.24
RF-DLC (our)	**77.67**	84.65	**56.17**	**44.07**	**82.90**	**13.80**	**28.46**	**19.00**	**35.59**	**30.77**

accuracy (0.04%). Besides, we also observe that the performance is relatively robust to the specific values of utility as long as the sign is correct, which is detailed in Appendix F.4.

Forms of class probability. We compare five different forms of $f(n_y)$ in terms of standard overall accuracy in Table 7 and Fig. 2a. We also analyze the weight values (i.e., $1/f(n_y)$) and their growth rates between the first and the last class across different forms. We find that as the growth rate becomes larger, ACC will be better accordingly, which suggests a high level of class imbalance in the dataset.

Fig. 2a shows similar results on the relationship between growth rate and the tail-class accuracy. As the growth rate becomes larger, the tail and med accuracy will both become significantly better despite the slight drop in head accuracy, which is consistent with the improvement in overall accuracy. Based on these results, we suggest using $f(n_y) = n_y$ in general.

Number of particles. Generally, using more individual models will induce better performances. However, we also need to balance the performance with the computational cost. We visualize accuracy under different numbers of particles in Fig. 2b. The error bars are scaled to be within two standard deviations. The accuracy curves are all logarithm-like and the accuracy improvement is hardly noticeable for more than 6 particles. However, the computational cost is increasing at a linear speed. Therefore, we recommend using no more than six particles in practice for a desirable performance-cost trade-off.

Repulsive force. We also evaluate the effectiveness of the repulsive force. The repulsive force effectively pushes the particle-based variational distribution to the target posterior and avoids collapsing into the same solution. Therefore, with the repulsive force, better predictive distributions can be learned, and thus better predictive uncertainty can be obtained. Besides, the repulsive force can also improve the overall accuracy by promoting the diversity of particles. We show the ablation study on repulsive force in Table 8. Increasing the repulsive force weight λ is generally beneficial to the uncertainty quantification, but may hurt the ACC when λ is too large.

Comparison with ensembles of single-model methods. For fair comparison, we include the results on ensembles of single-model long-tailed classification methods, as shown in Table 9. Ensembling single-model methods can reduce the performance gap between these methods and our RF-DLC, but will not outperforms RF-DLC.

6 Conclusion and Limitations

This paper proposes Making **R**eliable and **F**lexible **D**ecisions in **L**ong-tailed **C**lassification (RF-DLC) to address decision-making in realistic long-tailed problems. We focus on the problem of asymmetric mis-prediction cost in real-world long-tailed classification, and derive a novel decision-making framework from

Table 6: Comparison of different utilities in terms of False Head Rate on CIFAR100-LT. The tail-sensitive utility can effectively improve FHR with a negligible drop in standard overall accuracy.

Utility	**FHR (%)** @tail ratio ↓ 25%	50%	75%	average	Better (%)	**ACC (%)** ↑	Worse (%)
one-hot	18.55 ± 0.38	38.62 ± 0.62	60.17 ± 1.48	39.12 ± 0.72	18.00	$\mathbf{49.91 \pm 0.33}$	0.04
tail-sensitive	$\mathbf{15.39 \pm 0.57}$	$\mathbf{31.34 \pm 0.55}$	$\mathbf{49.51 \pm 1.45}$	$\mathbf{32.08 \pm 0.78}$		49.89 ± 0.19	

Table 7: Comparison of different forms of class probabilities. Top-1 standard accuracy evaluated on CIFAR100-LT. The linear $f(n_y)$ is the most suitable form in terms of standard accuracy.

Form of $f(n_y)$		**Weight Value** first class	last class	growth (%)	**ACC (%)** ↑
linear (Wang et al., 2017)	n_y	0.0020	0.1667	8250	$\mathbf{50.17 \pm 0.25}$
effective number (Cui et al., 2019)	$(1-\beta^{n_y})/(1-\beta)$	0.0023	0.1669	7297	49.90 ± 0.36
sqrt (Pan et al., 2021)	$\sqrt{n_y}$	0.0447	0.4082	814	47.03 ± 0.30
log	$\log n_y$	0.1609	0.5581	247	45.26 ± 0.51
constant	C	1.0000	1.0000	0	43.27 ± 0.30

Bayesian Decision Theory. Specifically, we introduce the integrated gain to naturally incorporate the distributional shift in long-tailed data, and leverage the utility matrix to make flexible decisions for various task settings. In empirical evaluations, we propose a new False Head Rate metric to quantify the particular type of misprediction from tail classes to head classes, along with large-scale image classification and uncertainty quantification experiments. The evaluation results demonstrate the superiority of our method on both standard and decision-critical long-tailed classifications.

However, we believe there are still some limitations for future developments. We list a few limitations below:

Long-tailed Regression. We have not explored long-tailed problems in regression, where the distribution of targets can also be highly imbalanced. However, with adjustments to the decision gain, we believe our framework can be adapted for regression tasks as well.

Dataset Shift. We have not accounted for general dataset shift scenarios, such as out-of-distribution data, where the assumption of semantically identical training and testing sets becomes invalid. Another example is the distribution of testing data. If it is no longer assumed to be uniform, the discrepancy ratio $p_{\text{test}}(\boldsymbol{x}, y)/p_{\text{train}}(\boldsymbol{x}, y)$ will not be expressed as $1/f(n_y)$, but rather in a more general form.

Broader Impact Statement

As a general framework for long-tailed classification, our method is eligible for handling imbalanced data in many real-world applications. For example, the strategy adopted in our method can promote fairness and improve performance, especially for the underrepresented demographic groups. Besides, the use of utility functions allows for considering specific groups and raising their importance. We believe that adapting our method to realistic social problems is an interesting and important direction. We will investigate this direction in our future works.

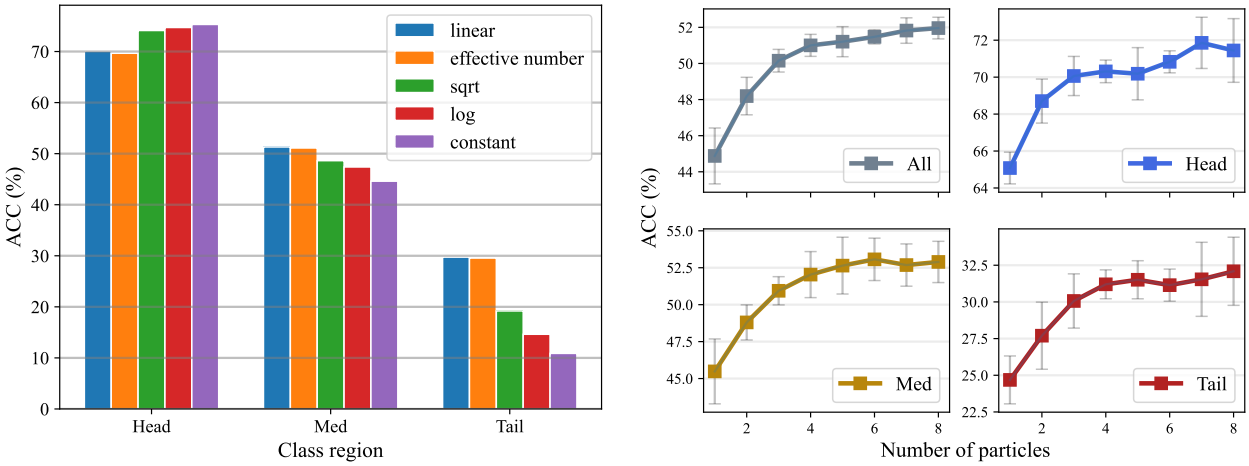
References

Jianhong Bai, Zuozhu Liu, Hualiang Wang, Jin Hao, Yang Feng, Huanpeng Chu, and Haoji Hu. On the effectiveness of out-of-distribution data in self-supervised long-tail learning. In *International Conference on Learning Representations*, 2022.

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622. PMLR, 2015.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.



(a) Different forms of $f(n_y)$ (b) Different numbers of particles

Figure 2: Ablation studies on CIFAR100-LT. Comparing (a) different forms of class probabilities and (b) varying numbers of particles in terms of overall accuracy and the accuracy of 3 class regions.

Table 8: The effect of repulsive force, compared by calibration on CIFAR100-LT. Applying an appropriate level of repulsive force will induce better predictive uncertainty.

Metric	$\lambda = 0$	$\lambda = 5 \times 10^{-6}$	$\lambda = 5 \times 10^{-5}$	$\lambda = 5 \times 10^{-4}$	$\lambda = 5 \times 10^{-3}$	$\lambda = 5 \times 10^{-2}$
ACC (%) ↑	50.15	50.19	50.20	**50.24**	49.96	48.18
AUC (%) ↑	75.94	78.96	79.25	81.24	**81.80**	77.51
ECE (%) ↓	13.40	13.31	12.45	10.35	**10.17**	10.79

Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 112–121, 2021.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Manuel Carranza-García, Pedro Lara-Benítez, Jorge García-Gutiérrez, and José C Riquelme. Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance. *Neurocomputing*, 449:229–244, 2021.

Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pp. 694–710. Springer, 2020.

Yu-An Chung, Hsuan-Tien Lin, and Shao-Wen Yang. Cost-aware pre-training for multiclass cost-sensitive deep learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1411–1417, 2016.

Adam D Cobb, Stephen J Roberts, and Yarin Gal. Loss-calibrated approximate inference in bayesian neural networks. *arXiv preprint arXiv:1805.03901*, 2018.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Francesco D'Angelo and Vincent Fortuin. Annealed stein variational gradient descent. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Table 9: Comparison with ensembles of single-model methods. Ensembling can only reduce the performance gap between single-model baselines and RF-DLC.

Method	ACC (%) ↑				AUC (%) ↑	ECE (%) ↓	FHR (%) ↓			
	All	Head	Med	Tail			avg	25%	50%	75%
3×CB Loss (Cui et al., 2019)	47.37	67.72	49.16	25.28	80.67	17.39	45.56	21.88	45.41	69.38
3×LDAM (Cao et al., 2019)	48.84	68.59	47.87	28.47	80.28	19.27	39.96	19.22	40.04	60.62
RF-DLC (3 particles)	**50.24**	**69.92**	**51.07**	**30.34**	**81.24**	**10.35**	**32.08**	**15.39**	**31.34**	**49.51**

Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pp. 973–978. Lawrence Erlbaum Associates Ltd, 2001.

Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2020.

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pp. 878–887. Springer, 2005.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

Yu-Cheng He, Yao-Xiang Ding, Han-Jia Ye, and Zhi-Hua Zhou. Learning only when it matters: Cost-aware long-tailed classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12411–12420, 2024.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Feng Hong, Jiangchao Yao, Zhihan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed partial label learning via dynamic rebalancing. In *International Conference on Learning Representations*, 2022.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6626–6636, 2021.

Minghui Hu, AK Malik, M Tanveer, and PN Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384, 2016.

Prateek Jaiswal, Harsha Honnappa, and Vinayak A Rao. Asymptotic consistency of loss-calibrated variational bayes. *Stat*, 9(1):e258, 2020.

Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

Audun Jsang. *Subjective Logic: A formalism for reasoning under uncertainty*. Springer Publishing Company, Incorporated, 2018.

Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13896–13905, 2020.

Tuen 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*, pp. 1–19, 1978.

Anna Korba, Adil Salim, Michael Arbel, Giulia Luise, and Arthur Gretton. A non-asymptotic analysis for stein variational gradient descent. *Advances in Neural Information Processing Systems*, 33:4672–4682, 2020.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, pp. 5436–5446. PMLR, 2020.

Tomasz Kuśmierczyk, Joseph Sakaya, and Arto Klami. Variational bayesian decision-making for continuous utilities. *Advances in Neural Information Processing Systems*, 32, 2019.

Simon Lacoste-Julien, Ferenc Huszár, and Zoubin Ghahramani. Approximate inference for the loss-calibrated bayesian. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 416–424. JMLR Workshop and Conference Proceedings, 2011.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. Trustworthy long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6970–6979, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Charles X Ling and Victor S Sheng. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of machine learning*, 2011:231–235, 2008.

Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pp. 4082–4092. PMLR, 2019a.

Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2970–2979, 2020.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019b.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 181–196, 2018.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *International Conference on Machine Learning*, pp. 603–611. PMLR, 2013.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2020.

Michael J Morais and Jonathan W Pillow. Loss-calibrated expectation propagation for approximate bayesian decision-making. *arXiv preprint arXiv:2201.03128*, 2022.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

Giung Nam, Sunguk Jang, and Juho Lee. Decoupled training for long-tailed classification with stochastic representations. In *International Conference on Learning Representations*, 2023.

Tai-Yu Pan, Cheng Zhang, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. On model calibration for long-tailed object detection and instance segmentation. *Advances in Neural Information Processing Systems*, 34:2529–2542, 2021.

Bernardo Avila Pires, Csaba Szepesvari, and Mohammad Ghavamzadeh. Cost-sensitive multiclass classification risk bounds. In *International Conference on Machine Learning*, pp. 1391–1399. PMLR, 2013.

Arafat Rahman, Iqbal Hassan, and Md Atiqur Rahman Ahad. Nurse care activity recognition: A cost-sensitive ensemble approach to handle imbalanced class problem in the wild. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, pp. 440–445, 2021.

William J Reed. The pareto, zipf and other power laws. *Economics letters*, 74(1):15–19, 2001.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in Neural Information Processing Systems*, 33:4175–4186, 2020.

Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced mse for imbalanced visual regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7926–7935, 2022.

Christian P Robert et al. *The Bayesian choice: from decision-theoretic foundations to computational implementation*, volume 2. Springer, 2007.

Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–9. IEEE, 2016.

Xiaohang Shi, Jun Hu, Xueyue Lei, and Shiyou Xu. Detection of flying birds in airport monitoring based on improved yolov5. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pp. 1446–1451. IEEE, 2021.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In *The 2010 International joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2010.

Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8769–8778, 2018.

Haotao Wang, Aston Zhang, Yi Zhu, Shuai Zheng, Mu Li, Alex J Smola, and Zhangyang Wang. Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition. In *International Conference on Machine Learning*, pp. 23446–23458. PMLR, 2022.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2020.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. *Advances in Neural Information Processing Systems*, 30, 2017.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision*, pp. 162–178. Springer, 2020.

Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*, pp. 247–263. Springer, 2020.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Zhixiong Yang, Junwen Pan, Yanzhan Yang, Xiaozhou Shi, Hong-Yu Zhou, Zhicheng Zhang, and Cheng Bian. Proco: Prototype-aware contrastive learning for long-tailed medical image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 173–182. Springer, 2022.

Dmitry Yudin, Anton Sotnikov, and Andrey Krishtopik. Detection of big animals on images with road scenes using deep learning. In *2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI)*, pp. 100–1003. IEEE, 2019.

Yifan Zhang, Bryan Hooi, HONG Lanqing, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. In *Advances in Neural Information Processing Systems*, 2022.

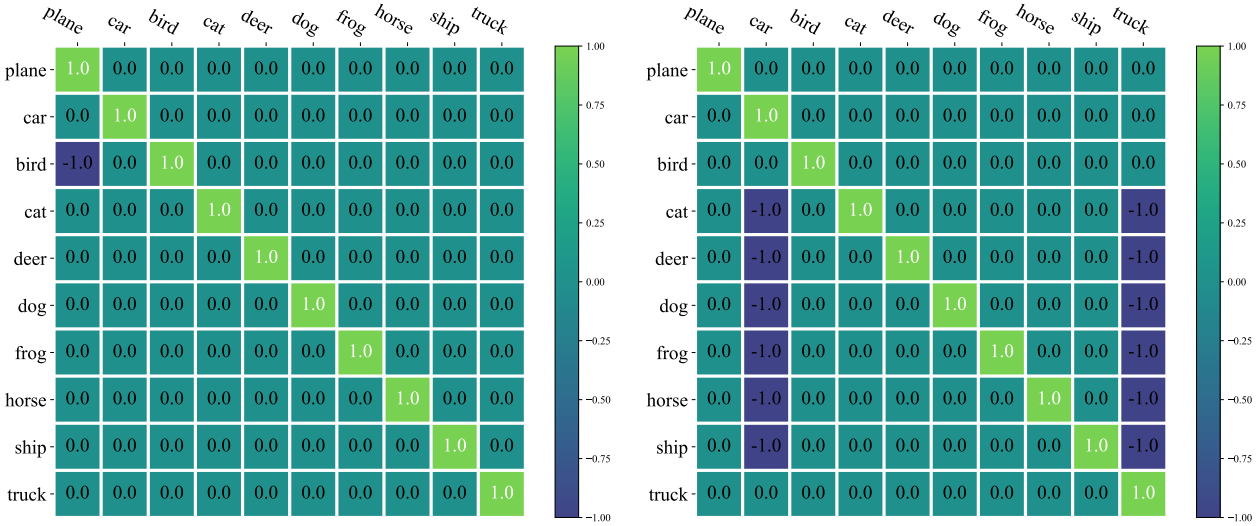
A Related Model Architectures

The model architecture of our method is an ensemble of multiple individual models. This architecture belongs to a general type of Bayesian neural networks, called particle-based BNN or particle optimization (Liu & Wang, 2016; D'Angelo & Fortuin, 2021). Generally, the particle-based variational distribution is in the form of $q(\boldsymbol{\theta}) = \sum_{j=1}^{M} w_j \cdot \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_j)$, where $\{w_j\}_{j=1}^{M}$ are normalized weights which hold $\sum_{j=1}^{M} w_j = 1$, and $\delta(\cdot)$ is the Dirac delta function. Each "particle" $\boldsymbol{\theta}_j$ is a deterministic model and provides the predictive probability $p(y|\boldsymbol{x}, \boldsymbol{\theta}_j)$. The particle-based BNN is first studied in Stein variational gradient descent (SVGD) (Liu & Wang, 2016) and then explored by Liu et al. (2019a); Korba et al. (2020); D'Angelo & Fortuin (2020). Instead of directly modeling the gradient flow, our framework optimizes the particles through stochastic gradient descent (SGD), with repulsive force induced by the integrated gain objective. Compared to existing particle optimization, our method is easy and cheap to implement, which is especially beneficial for large-scale deep learning.

Similar architectures like the "multi-expert" models have been explored by previous long-tailed classification methods (Xiang et al., 2020; Wang et al., 2020; Li et al., 2022). They usually combine several individual classifiers with a shared encoder to obtain better generalization performances (Lakshminarayanan et al., 2017; Hu et al., 2022), which is inspired by the observation that multiple i.i.d. initializations are less likely to generate averagely "bad" models (Dietterich, 2000). The particle-based models reduce the cost of Bayesian inference and are more efficient than variational inference and Markov chain Monte Carlo (MCMC), especially on high-dimensional and multimodal distributions. Besides, the computational cost of our method can be further reduced by leveraging recent techniques, such as partially being Bayesian in model architectures (Kristiadi et al., 2020).

B Full Utility Matrices for Class/Metaclass-sensitive Utility

We show the full utility matrices of class-sensitive and metaclass-sensitive utilities in Fig. 3, which is based on the 10 categories of CIFAR10-LT (Cui et al., 2019).



(a) Full Class-sensitive Utility Matrix

(b) Full Metaclass-sensitive Utility Matrix

Figure 3: Full utility matrix configurations on CIFAR10-LT. The two matrices are designed to particularly avoid certain misprediction types (bird-place and mammal-vehicle mispredictions respectively).

C Derivation of Eq. 10

The following proof is based on the assumption that both training and testing data are semantically identical[7], and the only difference lies in class probabilities ($p_{\text{train}}(y)$ and $p_{\text{test}}(y)$).

[7]In contrast to the open-set scenario (Geng et al., 2020), where additional classes may cause testing data to be semantically irrelevant to training data.

Proof. We denote the dataset as $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{Y}\} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where each pair is drawn from the same data distribution. Then the logarithm of integrated gain would be:

$$\begin{aligned}
\log G(\boldsymbol{d}) &= \log \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})} G(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{\theta}) \\
&\overset{\text{(a)}}{\geq} \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \log \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})} G(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{\theta}) \\
&= \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \log \int_{\boldsymbol{\Theta}} q(\boldsymbol{\theta}) G(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \\
&\overset{\text{(b)}}{\geq} \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \int_{\boldsymbol{\Theta}} q(\boldsymbol{\theta}) \log \left[G(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta})} \right] d\boldsymbol{\theta} \\
&:= \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \Psi(\mathcal{D}).
\end{aligned} \tag{18}$$

Here, (a) and (b) are by Jensen's inequality (Jensen, 1906). The $\Psi(\mathcal{D})$ inside the expectation can be further extended to be:

$$\begin{aligned}
\Psi(\mathcal{D}) &= \int_{\Theta} q(\boldsymbol{\theta}) \log \left[G(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta}|\mathcal{D})}{q(\boldsymbol{\theta})} \right] d\boldsymbol{\theta} \\
&= \int_{\Theta} q(\boldsymbol{\theta}) \log \left[G(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{\theta}) \cdot \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \cdot \frac{p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Y}|\boldsymbol{X})} \right] d\boldsymbol{\theta} \\
&= \int_{\Theta} q(\boldsymbol{\theta}) \left[\log \prod_{i=1}^N g(d_i|\boldsymbol{x}_i,\boldsymbol{\theta}) - \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log \prod_{i=1}^N p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \right] d\boldsymbol{\theta} \\
&= \sum_{i=1}^N \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \left[\log g(d_i|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}).
\end{aligned} \tag{19}$$

Therefore, the ultimate expression of integrated gain is:

$$\begin{aligned}
\log G(\boldsymbol{d}) &\geq \mathbb{E}_{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\sim p_{\text{test}}(\boldsymbol{x},y)} \Psi(\mathcal{D}) \\
&= \sum_{i=1}^N \mathbb{E}_{(\boldsymbol{x}_i,y_i)\sim p_{\text{test}}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \left[\log g(d_i|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&= \sum_{i=1}^N \mathbb{E}_{(\boldsymbol{x}_i,y_i)\sim p_{\text{train}}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{p_{\text{test}}(\boldsymbol{x}_i,y_i)}{p_{\text{train}}(\boldsymbol{x}_i,y_i)} \left[\log g(d_i|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&\overset{\text{(c)}}{\approx} \sum_{i=1}^N \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{p_{\text{test}}(\boldsymbol{x}_i,y_i)}{p_{\text{train}}(\boldsymbol{x}_i,y_i)} \left[\log g(d_i|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&\propto \sum_{i=1}^N \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{1}{f(n_{y_i})} \left[\sum_{y'} U_{y',d_i} \cdot \log p(y'|\boldsymbol{x}_i,\boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) \right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&:= L(q,\boldsymbol{d}).
\end{aligned} \tag{20}$$

Here, (c) is by the one-sample MC estimates of the expectation over the data point $(\boldsymbol{x}_i, y_i)$. □

D Proof of Eq. 11 and Eq. 12

Proof. The objective in Eq. 10 can be written as:

$$
\begin{aligned}
L(q, \boldsymbol{d}) &\approx \sum_{i=1}^{N} \mathbb{E}_{(\boldsymbol{x}_i, y_i)\sim p_{\text{test}}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \left[\log g(d_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})\right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&= \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\sim p_{\text{test}}(\boldsymbol{x}, y)} \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \left[\log g(d_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + \log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})\right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&= \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\sim p_{\text{test}}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \left[\log G(\boldsymbol{d}|\boldsymbol{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})\right] - \mathbf{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \log p(\boldsymbol{Y}|\boldsymbol{X}) \\
&\overset{\text{(a)}}{\approx} \int_{\Theta} q(\boldsymbol{\theta}) \left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} - \log p_{\text{test}}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{Y}|\boldsymbol{X})\right] d\boldsymbol{\theta} + \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\sim p_{\text{test}}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log G(\boldsymbol{d}|\boldsymbol{X}, \boldsymbol{\theta}) \\
&= -\mathbf{KL}(q(\boldsymbol{\theta})||p_{\text{test}}(\boldsymbol{\theta}|\mathcal{D})) + \mathbb{E}_{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\sim p_{\text{test}}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log G(\boldsymbol{d}|\boldsymbol{X}, \boldsymbol{\theta}).
\end{aligned}
\tag{21}
$$

Here, (a) is by N independent one-sample MC approximation over all of the data points. When we choose the one-hot utility matrix $\boldsymbol{U} = \boldsymbol{I}$, the decision gain $G(\boldsymbol{d} = \boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})$ will equal the predictive probability $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})$, and thus it can be ignored. Therefore, the objective with one-hot utility matrix is:

$$
L(q, \boldsymbol{d}) \approx -\mathbf{KL}(q(\boldsymbol{\theta})||p_{\text{test}}(\boldsymbol{\theta}|\mathcal{D})). \tag{22}
$$

□

E Implementation Details

We summarize the necessary implementation details in this section for the reproducibility of our method. The hyper-parameter choices are concluded in table 10. The optimal values of those hyper-parameters are determined by grid search. We will release the code after acceptance.

Table 10: Hyper-parameter configurations.

Dataset	Base Model	Optimizer	Batch Size	Learning Rate	Training Epochs	Discrepancy Ratio	λ	τ	α
CIFAR10-LT	ResNet32	SGD	128	0.1	200	linear	5e-4	40	0.002
CIFAR100-LT	ResNet32	SGD	128	0.1	200	linear	5e-4	40	0.3
ImageNet-LT	ResNet50	SGD	256	0.1	100	linear	2e-4	20	50
iNaturalist	Slim ResNet50	SGD	512	0.2	100	linear	2e-4	20	100

E.1 Evaluation Protocol

The evaluation protocol consists of standard classification accuracy, three newly designed experiments on the False Head Rate (FHR) along with other two metrics, and calibration experiments on AUC and ECE metrics. Besides, we conduct several ablation studies to evaluate different choices of implementation and the effectiveness of components in our method. For all quantitative and visual results, we repeatedly run the experiments five times with random initialization to obtain the averaged results and standard deviations to eliminate random error. We use $f(n_y) = n_y$ unless otherwise specified.

E.2 Training Objective

We slightly modify the training objective in the code-level implementation to apply two practical tricks for better performance.

Repulsive force. We find in experiments that although applying repulsive force can promote the diversity of particles, it will certainly disturb the fine-tuning stage in training, which consequently results in sub-optimal performances by the end of training. To address this issue, we apply an annealing weight to the repulsive force to reduce

its effect as the training proceeds:

$$\exp\{-t/\tau\} \cdot \frac{1}{2} \sum_k \log\left(\overline{\boldsymbol{\theta}^2} - \overline{\boldsymbol{\theta}}^2\right)_k, \tag{23}$$

where t refers to the epoch and τ is a stride factor that controls the decay of annealing weight. With the annealing weight, the repulsive force will push particles away at the beginning of training, and gradually become negligible at the end of training.

Utility matrix. Although the utility matrices in Fig. 1 are designed to address the many realistic problem settings, they will also affect the accuracy of the classification task. Therefore, the utility term in Eq. 10 needs re-scaling so that its negative effect on the accuracy is controllable:

$$\log p(y|\boldsymbol{x}, \boldsymbol{\theta}) + \frac{1}{\alpha} \cdot \sum_{y'} U_{y',d} \cdot \log p(y'|\boldsymbol{x}, \boldsymbol{\theta}), \tag{24}$$

where α is the scaling factor. We can adjust the value of α to carefully control the effect of the utility term, which will bring us significant improvement on the False Head Rate (along with other metrics) with an acceptable accuracy drop.

E.3 Discussion on Computational Cost

We run all experiments on an NVIDIA RTX A6000 GPU (49 GB) and do not need multiple GPUs for one model. The model architecture follows RIDE (Wang et al., 2020) and TLC (Li et al., 2022), in which the first few layers in neural networks are shared among all particles. Therefore, the computational cost of our method is comparable to existing ensemble models. Besides, compared with gradient-flow-based BNN like D'Angelo & Fortuin (2021), which typically uses 20 particles, our model is far more efficient with no more than 5 particles.

We also provide a computational cost comparison with other multi-model methods in Table 11. We use 3 particles of ResNet32 as the backbone of each method, and train on CIFAR100-LT for 200 epochs at one NVIDIA RTX A6000. The results show that the efficiency of our method is comparable with baselines.

Table 11: Comparison of computational cost

Method	Training Time per 200 Epoch (min)	# Parameters
ResNet32 (single model)	31.85	469,904
RIDE (Wang et al., 2020)	43.13	1,408,784
TLC (Li et al., 2022)	45.27	1,408,784
RF-DLC (ours)	44.68	1,408,784

F Additional Experimental Results

F.1 Full Experimental Results on Classification

We list the full experimental results of top-1 accuracy in Table 12, including the results on iNaturalist (Van Horn et al., 2018). Classes are equally split into three class regions (head, med and tail). For example, there are 33, 33 and 34 classes respectively in the head, med and tail regions of CIFAR100-LT. The results on iNaturalist are obtained by a single run due to the large size of the dataset. We use a slim version of ResNet50 for all baselines due to GPU memory limitation. The reported results on iNaturalist are different from other papers but are still **fair comparisons**. Our method successfully outperforms all other baselines especially on the tailed classes.

F.2 Full Experimental Results on Uncertainty Quantification

Table 13 shows the full experimental results on uncertainty quantification. The results are broken down to 3 class regions, where Bayesian predictive uncertainty outperforms other uncertainty algorithms on all of them.

F.3 Comparison with Cost-sensitive Learning Approaches

The canonical approaches in cost-sensitive learning (Elkan, 2001) could not effectively solve the decision-making problem in long-tailed classification due to their lack of a holistic approach that integrates distributional shift and

Table 12: Full top-1 accuracy results (%). † means the results are directly copied from the original paper. * means the results are obtained from a slim version of ResNet50 due to GPU memory limits.

Dataset	Method	All	Head	Med	Tail
CIFAR10-LT	LA†	77.67	-	-	-
	ACE†	81.2	-	-	-
	SRepr†	82.06 ± 0.01	-	-	-
	CE	73.65 ± 0.39	$\mathbf{93.22 \pm 0.26}$	74.27 ± 0.42	58.51 ± 0.62
	CB Loss	77.62 ± 0.69	91.70 ± 0.57	75.41 ± 0.76	68.73 ± 1.52
	LDAM	80.63 ± 0.69	90.03 ± 0.47	75.88 ± 0.81	77.14 ± 1.61
	RIDE	83.11 ± 0.52	91.49 ± 0.40	$\mathbf{79.39 \pm 0.61}$	79.62 ± 1.56
	TLC	79.70 ± 0.65	89.47 ± 0.33	74.33 ± 0.96	76.39 ± 0.98
	RF-DLC	$\mathbf{83.75 \pm 0.17}$	90.49 ± 0.60	78.89 ± 0.87	$\mathbf{82.33 \pm 1.16}$
CIFAR100-LT	LA†	43.89	-	-	-
	ACE†	49.4	66.1	55.7	23.5
	SRepr†	47.81 ± 0.02	66.69 ± 0.01	49.91 ± 0.01	23.31 ± 0.11
	CE	38.82 ± 0.52	68.30 ± 0.61	38.39 ± 0.49	10.62 ± 1.23
	CB Loss	42.24 ± 0.41	62.53 ± 0.44	44.36 ± 0.96	20.50 ± 0.51
	LDAM	43.13 ± 0.67	63.58 ± 0.93	42.90 ± 1.03	23.50 ± 1.28
	RIDE	48.99 ± 0.44	69.11 ± 0.54	49.70 ± 0.59	28.78 ± 1.52
	TLC	48.75 ± 0.16	69.43 ± 0.36	49.02 ± 0.94	28.40 ± 0.72
	RF-DLC	$\mathbf{50.24 \pm 0.70}$	$\mathbf{69.92 \pm 0.77}$	$\mathbf{51.07 \pm 0.82}$	$\mathbf{30.34 \pm 1.49}$
ImageNet-LT	LA†	51.11	-	-	-
	ACE†	54.7	-	-	-
	SRepr†	52.12 ± 0.06	62.52 ± 0.26	49.44 ± 0.18	32.14 ± 0.41
	CE	47.80 ± 0.15	53.46 ± 0.36	45.92 ± 0.19	44.03 ± 0.24
	CB Loss	51.70 ± 0.25	57.62 ± 0.46	49.19 ± 0.21	48.29 ± 0.41
	LDAM	51.04 ± 0.21	57.66 ± 0.40	48.26 ± 0.19	47.21 ± 0.22
	RIDE	54.32 ± 0.54	60.88 ± 0.71	51.35 ± 0.44	50.74 ± 0.62
	TLC	55.03 ± 0.34	61.19 ± 0.53	52.35 ± 0.31	51.56 ± 0.35
	RF-DLC	$\mathbf{55.73 \pm 0.17}$	$\mathbf{62.18 \pm 0.28}$	$\mathbf{53.06 \pm 0.22}$	$\mathbf{51.98 \pm 0.40}$
iNaturalist	LA†	66.36	-	-	-
	ACE†	72.9	-	-	-
	SRepr†	70.79 ± 0.17	70.70 ± 0.31	$\mathbf{70.83 \pm 0.20}$	$\mathbf{70.79 \pm 0.17}$
	CE*	65.17	75.28	64.00	54.22
	LDAM*	67.20	74.58	65.17	60.84
	RIDE*	72.87	**76.71**	69.73	69.28
	RF-DLC*	**73.80**	**76.66**	**70.79**	**70.55**

Table 13: Full results of uncertainty quantification. Bayesian predictive uncertainty outperforms other uncertainty algorithms on all class regions.

Uncertainty Algorithm	AUC (%) ↑				ECE (%) ↓			
	All	Head	Med	Tail	All	Head	Med	Tail
MCP (Hendrycks & Gimpel, 2017)	80.15	87.56	80.78	62.15	19.39	11.52	18.30	36.84
Evidential (Li et al., 2022)	78.16	84.51	79.62	**67.63**	21.88	10.27	22.01	34.53
Bayesian (RF-DLC)	**80.62**	**88.47**	**81.82**	63.05	**10.87**	**6.37**	**11.09**	**16.71**

decision-making processes. For example, the method in Section 2 of Elkan (2001) does not consider specific error types during the training phase, and fails to incorporate the utility matrix during testing. The Bayesian method in Section 4 of Elkan (2001) learns a standard posterior without accounting for data distribution or the utility matrix, applying the latter only during the testing phase. Additionally, Elkan (2001) employs a different decision strategy (Eq. 1), which is not as flexible as ours and has been discussed in Section 4.1.

To further illustrate the advantages of our method, we empirically compare our method with two baselines: i) the Bayesian method in Elkan (2001) and ii) the naive combination of re-weighting loss and the Bayesian method in Elkan (2001). Table 14 shows that our method significantly outperforms both baselines in terms of ACC and FHR. This demonstrates the advantages of our method which concurrently addresses long-tailed distributions and decision-making during both training and testing phases in a unified way.

Table 14: Comparison with cost-sensitive learning approaches on CIFAR100-LT. Tail-sensitive utility is applied.

Method	**ACC (%) ↑**		**FHR (%) ↓**			
	All	Tail	25%	50%	75%	Avg
RF-DLC	49.92	33.74	14.92	30.22	51.80	32.31
Elkan (2001)	43.36	24.68	25.97	41.98	52.64	40.20
Reweighting Loss + Elkan (2001)	45.40	26.35	20.69	41.28	65.36	42.44

F.4 Ablation Study on the Robustness of Utility Values

We conducted the experiment on the robustness of utility values on CIFAR100-LT. The utility matrix used in this experiment is:

$$\boldsymbol{U} := \begin{bmatrix} 1 & 0 & \cdots & 0 \\ u & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u & u & \cdots & 1 \end{bmatrix}. \tag{25}$$

The results, as shown in Table 15, demonstrate that all values of u will lead to improvement in FHR and comparable or better ACC compared with the baseline one-hot matrix (i.e. $u = 0$). These findings indicate that our method's performance is relatively insensitive to variations in utility values (we simply use -1 in the paper) and tuning these values can further improve performance.

Table 15: Ablation study on the robustness of utility values on CIFAR100-LT.

Utility Value	**ACC (%) ↑**		**FHR (%) ↓**			
	All	Tail	25%	50%	75%	Avg
0	49.76	30.00	19.12	38.04	60.44	39.20
-0.1	49.70	29.88	18.12	36.56	59.32	38.00
-0.2	49.09	29.38	18.16	36.34	60.28	38.26
-0.3	49.41	29.82	17.81	35.28	58.12	37.07
-0.4	49.49	30.41	17.20	34.64	56.80	36.21
-0.5	49.90	31.47	16.71	33.52	55.72	35.32
-0.6	49.84	31.82	16.92	33.82	54.72	35.15
-0.7	49.32	30.85	16.33	32.04	55.08	34.48
-0.8	49.66	33.00	15.91	31.32	51.60	32.94
-0.9	49.27	32.12	15.53	30.68	51.24	32.48
-1	49.92	33.74	14.92	30.22	51.80	32.31

F.5 Ablation Study on the Task-specific Utility Functions

The task-specific utility function (used in Eq. 10 and Eq. 16) is one of the key contributions of this paper. It specifically avoids mispredicting some classes as other classes. To understand the effect of utility functions, we add an ablation study in Table 16, where Eq. 16 with tail-sensitive utility is added to baselines at inference time. Adding Eq. 16 improves the FHR performance and maintains the ACC, which demonstrates the effectiveness of our inference strategy. Importantly, our method is still better than baselines with utility functions at inference time.

Table 16: Ablation study on the effectiveness of task-specific utility functions (Eq. 16) at inference time. Naively applying Eq. 16 to other baselines can improve FHR, but will not outperforms RF-DLC.

Method	**ACC (%) ↑**				**FHR (%) ↓**			
	All	Head	Med	Tail	avg	25%	50%	75%
CB Loss	42.24	62.53	44.36	20.50	49.22	24.88	48.41	74.38
CB Loss + Eq. 16	42.07	60.35	43.69	23.10	47.89	21.86	49.66	72.16
LDAM	43.13	63.58	42.90	23.50	43.29	21.22	43.04	65.62
LDAM + Eq. 16	43.16	62.11	43.07	24.59	40.90	20.17	40.85	61.68
RF-DLC	**50.24**	**69.92**	**51.07**	**30.34**	**32.08**	**15.39**	**31.34**	**49.51**