5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–21

Long-tailed Classification from a Bayesian-decision-theory Perspective

Bolian Li LI4468@PURDUE.EDU
Ruqi Zhang RUQIZ@PURDUE.EDU
Department of Computer Science, Purdue University, USA

Abstract

Long-tailed classification poses a challenge due to its heavy imbalance in class probabilities and tail-sensitivity risks with asymmetric misprediction costs. Recent attempts have used re-balancing loss and ensemble methods, but they are largely heuristic and depend heavily on empirical results, lacking theoretical explanation. Furthermore, existing methods overlook the decision loss, which characterizes different costs associated with tailed classes. This paper presents a general and principled framework from a Bayesian-decision-theory perspective, which unifies existing techniques including re-balancing and ensemble methods, and provides theoretical justifications for their effectiveness. From this perspective, we derive a novel objective based on the integrated risk and a Bayesian deep-ensemble approach to improve the accuracy of all classes, especially the "tail". Besides, our framework allows for task-adaptive decision loss which provides provably optimal decisions in varying task scenarios, along with the capability to quantify uncertainty. Finally, We conduct comprehensive experiments, including standard classification, tail-sensitive classification with a new False Head Rate metric, calibration, and ablation studies. Our framework significantly improves the current SOTA even on large-scale real-world datasets like ImageNet.

1. Introduction

Machine learning methods usually assume that training and testing data are both i.i.d. sampled from the same data distribution. However, this is not always true for real-world scenarios (Hand, 2006). One example is *long-tailed classification* (Liu et al., 2019b; Li et al., 2022), where the training data is biased towards a few "head" classes, while the "tailed" classes have fewer samples, resulting in a "long-tailed" distribution of class probabilities. The long-tailed problem is mainly due to the process of collecting data, which is unavoidably biased. Conventional models trained on long-tailed data often report significant performance drops compared with the results obtained on balanced training data (Wang et al., 2022). Besides, for some real-world applications, the risk of classifying tailed samples as head (more common) is obviously more severe than that of classifying head samples as tail (less common) (Rahman et al., 2021; Yang et al., 2022).

Existing works usually re-balance the loss function to promote the accuracy of tail classes (Cao et al., 2019; Cui et al., 2019). Their re-weighting strategy compensates for the lack of training samples in tailed classes, but suffers from sub-optimal head class accuracies. Other attempts on ensemble models try to reduce the model variance to promote the head and tail accuracies at the same time (Wang et al., 2020). Despite the effectiveness of existing works, they suffer from significant limitations: **i)** they are largely based on empirical results without adequate theoretical explanation; **ii)** they ignore the decision loss, which represents the application-related risks (e.g., the tail-sensitivity risk) in the real world and thus their models are not applicable to tasks with different metrics other than standard classification task; **iii)** most methods do not quantify uncertainty, which reduces their reliability.

In this paper, we propose a unified framework for long-tailed classification, rooted in *Bayesian Decision Theory* (Berger, 1985; Robert et al., 2007). Our framework unifies existing methods and provides theoretical justifications for their effectiveness, including re-balancing loss and ensemble methods, which have been shown to achieve promising results. To derive our framework, we first introduce a new objective based on the *integrated risk* which unifies three crucial components in long-tailed problems: data distribution, decision loss, and posterior inference. To minimize this objective, we then derive a tractable lower bound based on variational EM (Lacoste-Julien et al., 2011) and approximate the posterior by a particle-based ensemble model (D'Angelo and Fortuin, 2021). Furthermore, we design two kinds of *utility functions* for the standard and tail-sensitive classifications respectively, which enables real-world applications with tail-sensitivity risks. Finally, we conduct comprehensive experiments to demonstrate the superiority of our method in general settings.

We summarize our contributions as follows: **i)** *Long-tailed Bayesian Decision* (LBD) is the first to formulate long-tailed classification under Bayesian Decision Theory; **ii)** for real-world applications, we take the decision loss into account, extending our method to more realistic long-tailed problems where the risk of wrong predictions varies and depends on the type of classes (e.g., head or tail); **iii)** we conduct comprehensive experiments including a newly designed False Head Rate (FHR) to show the effectiveness of our method.

2. Background

2.1. Long-tailed Distribution

Long-tailed distributed data is a special case of *dataset shift* (Quinonero-Candela et al., 2008), in which the common assumption is violated that the training and testing data follow the same distribution (Moreno-Torres et al., 2012). For the long-tailed scenario studied in this paper, the training data $\mathcal{D}_{train}$ is distributed in a descending manner over categories in terms of class probability: $p(\boldsymbol{x}_1, y_1 = k_1) \geq p(\boldsymbol{x_2}, y_2 = k_2)$, if $k_1 \leq k_2$ for all $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2) \in \mathcal{D}_{train}$. While the testing data $\mathcal{D}_{test}$ is assumed to be distributed uniformly over categories: $p(\boldsymbol{x}_1, y_1 = k_1) = p(\boldsymbol{x_2}, y_2 = k_2)$ for all $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2) \in \mathcal{D}_{test}$. One important feature of long-tailed distribution is that both training and testing data are semantically identical, and the only difference lies in class probabilities.

2.2. Bayesian Decision Theory

Bayesian Decision Theory is a general statistical approach and can be applied to the task of pattern classification (Berger, 1985; Robert et al., 2007). The Bayesian Decision Theory considers the utility of making different decisions and the data distribution, which bridges posterior inference, data distribution and decision-making in a unified manner. For example, *posterior risk* is defined by the decision losses averaged over the posterior, and *integrated risk* further considers the data distribution. Bayesian Decision Theory has theoretical guarantees on the results and is provable to provide a desirable decision. Models following Bayesian Decision Theory are expected to have smaller risks than models trained in other ways.

3. Long-tailed Bayesian Decision

For conventional long-tailed classification, **inference** (how to infer model parameters), **decision** (model's actions in the presence of application-related risks), and **data distribution** (long-tailed distribution) are independent from each other in the training phase (Lacoste-Julien et al., 2011). To the best of our knowledge, none of previous methods can simultaneously consider these three aspects. In order to address this drawback, we introduce the *integrated risk* from Bayesian Decision Theory, which is computed over the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ and the data distribution $p(\boldsymbol{x}, y)$:

$$R(d) := \mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim p(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D})} l(\boldsymbol{\theta}, d(\boldsymbol{x}_i)), \tag{1}$$

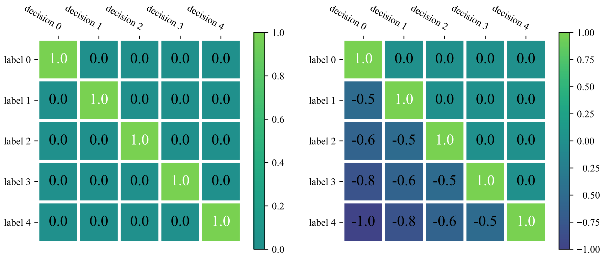
where $l(\boldsymbol{\theta}, d(\boldsymbol{x}_i))$ is the loss of making decision $d(\boldsymbol{x}_i)$ for $\boldsymbol{x}_i$ when the environment is $\boldsymbol{\theta}$ (model's parameters). The decision estimator d that minimizes the integrated risk is proved to give the optimal decisions in terms of the decision loss (Robert et al., 2007).

In order to exploit Eq. 1 as the objective, we need to determine the posterior and the optimal decision at the same time, which is notoriously hard because they depend on each other. Inspired by the EM algorithm (Lacoste-Julien et al., 2011), which alternately conducts the integration and optimization steps, we propose a long-tailed version of variational EM algorithm to alternately update a variational distribution and a classification decision on long-tailed data. To use EM, we convert the minimization problem to a maximization problem. Specifically, we define the *decision gain*: $g(\boldsymbol{\theta}, d(\boldsymbol{x}_i)) \propto -l(\boldsymbol{\theta}, d(\boldsymbol{x}_i)) = \prod_{y'} p(y'|\boldsymbol{x}_i, \boldsymbol{\theta})^{u(y', d(\boldsymbol{x}_i))}$ to represent what we gain from making decision $d(\boldsymbol{x}_i)$ given the environment $\boldsymbol{\theta}$ (Appendix A). Here, $u(y', d)$ is a fixed utility function that gives the utility of making decision d when the true label is y'. Then our goal becomes maximizing the *integrated gain*:

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

3.1. Task-adaptive Utility Functions

We first discuss the design of the utility function: $u(y, d)$, where y is the ground truth and d is the decision. The utility function defines the gain of making different decisions and can encode our preference for specific metrics in various tasks. The utility function is a standard component in Decision Theory and its design has been comprehensively studied in the literature. For example, Chapter 2.2 of Robert et al. (2007) guarantees the existence of utility functions with rational decision-makers. Generally, the values of the utility function over all class labels are stored in a form of utility matrix $\boldsymbol{U}$, where $U_{ij} = u(y = i, d = j)$.



(a) One-hot utility

(b) Tail-sensitive utility

Figure 1: Two examples of utility matrices, designed for (a) standard and (b) tail-sensitive classifications respectively.

In a standard classification setting, the overall accuracy is the most decisive metric in evaluation. It only matters whether the decision is consistent with the ground truth (i.e., $y = d$). Therefore, as shown in Fig. 1(a), a simple one-hot utility can be defined by $u(y, d) = \mathbb{1}\{y = d\}$, which is corresponding to the standard accuracy metric.

In modern applications of long-tailed classification, the semantic importance of "tailed" data often implies more penalty in the circumstance of predicting tailed samples as head (Sengupta et al., 2016; Yang et al., 2022). Besides, the lack of training samples in tailed 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 would reflect the potential of a model in real-world applications (Section 4.3). To this end, a tail-sensitive utility can be defined by adding an extra penalty on those false head samples, as shown in Fig. 1(b). The tail-sensitive utility encourages the model to predict any uncertain sample as tail rather than head, while not affecting the predictions of the true class when the model is confident.

3.2. Inference Step

Due to the discrepancy between training (long-tailed) and testing (uniform) data distributions, we propose to compute the integrated gain with the posterior of *testing* data $p(\boldsymbol{\theta}|\mathcal{D}_{test})$ to target at evaluation, where $\mathcal{D}_{test} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ with $(\boldsymbol{x}_i, y_i) \sim p_{test}(\boldsymbol{x}, y)$. To infer the posterior $p(\boldsymbol{\theta}|\mathcal{D}_{test})$, we use the variational method where a variational distribution $q(\boldsymbol{\theta})$ is introduced to the lower bound of the integrated gain in Eq. 2:

$$
\begin{aligned}
L(q, d) &:= \log G(d) = \log \mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim p_{test}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D}_{test})} g(\boldsymbol{\theta}, d(\boldsymbol{x}_i)) \\
&\geq \mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim p_{train}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta})} \frac{p_{test}(\boldsymbol{x}_i, y_i)}{p_{train}(\boldsymbol{x}_i, y_i)} \left[\log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + \sum_{y'} u(y', d) \log p(y'|\boldsymbol{x}_i, \boldsymbol{\theta}) \right] \\
&\quad - KL(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta})) + C,
\end{aligned}
\tag{3}
$$

where $(\boldsymbol{x}_i, y_i)$ is training data but we forcefully compute its probability on testing distribution, and C is a constant. Eq. 3 is proved in Appendix C. The lower bound $L(q, d)$ is our training objective and it provides a cross-entropy-like way to update the variational distribution $q(\boldsymbol{\theta})$, and most importantly, converts the data distribution from $p_{test}(\boldsymbol{x}, y)$ (uniform) to $p_{train}(\boldsymbol{x}, y)$ (long-tailed) to make the computation during training possible.

Moreover, the variational distribution $q(\boldsymbol{\theta})$ is guaranteed to be an approximation of the posterior $p(\boldsymbol{\theta}|\mathcal{D}_{test})$, because Eq. 3 contains Bayesian inference on the posterior of testing data. To support this, we look into the KL divergence between $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathcal{D}_{test})$:

$$
\begin{aligned}
&KL(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\mathcal{D}_{test})) \\
&= -\mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim p_{train}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta} \sim q(\boldsymbol{\theta})} \frac{p_{test}(\boldsymbol{x}_i, y_i)}{p_{train}(\boldsymbol{x}_i, y_i)} \cdot \log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + KL(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta})) - C,
\end{aligned}
\tag{4}
$$

where C is a constant. Eq. 4 is proved in Appendix D. Comparing Eq. 3 and Eq. 4, it is clear that part of the objective at inference step is Bayesian inference on the posterior of testing data, with $q(\boldsymbol{\theta})$ approaching $p(\boldsymbol{\theta}|\mathcal{D}_{test})$.

In summary, $L(q, d)$ enables the framework to simultaneously consider inference, decision (utility), and data distribution ($p_{test}(\boldsymbol{x}, y)/p_{train}(\boldsymbol{x}, y)$, further discussed in Section B.1). The detailed computation process of $L(q, d)$ is discussed in Appendix B.

3.3. Decision Step

To optimize $L(q, d)$ w.r.t the decision d, one way is to select the decision $d^\star$ that maximizes the gain respectively for each input $\boldsymbol{x}_i$ given the current variational distribution:

$$d^\star = \arg\max_d \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \sum_{y'} u(y', d) \log p(y'|\boldsymbol{x}_i, \boldsymbol{\theta}). \tag{5}$$

Notably, for symmetric utility functions (e.g., one-hot utility), Eq. 5 can be further simplified: $d^\star = \arg\max_d \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log p(d|\boldsymbol{x}, \boldsymbol{\theta})$, equivalent to the maximum of the predictive distribution.

However, during training, we essentially know that the optimal decisions for training data are their true labels. Therefore, we can utilize this knowledge and simply set $d(x_i) = y_i$. We can also view this as selecting the optimal decisions under a well-estimated $q(\boldsymbol{\theta})$ in Eq. 5 instead of the current distribution, since we expect $d^\star$ approach the true labels as $q(\boldsymbol{\theta})$ keeps updating. Then the objective can be further simplified to be:

$$\begin{aligned} &L(q) := L(q, d = y) \\ &= \mathbb{E}_{(\boldsymbol{x}_i, y_i)\sim p_{train}(\boldsymbol{x}, y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{p_{test}(\boldsymbol{x}_i, y_i)}{p_{train}(\boldsymbol{x}_i, y_i)} \left[\log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + \sum_{y'} u(y', y_i) \log p(y'|\boldsymbol{x}_i, \boldsymbol{\theta}) \right] \\ &\quad - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) + C. \end{aligned} \tag{6}$$

During testing, we use Eq. 5 to select the decision for testing data $\boldsymbol{x}_i$.

4. Experiments

4.1. Experimental Settings

Datasets. We use three long-tailed image datasets. CIFAR-10-LT and CIFAR-100-LT (Cui et al., 2019) are sampled from the original CIFAR dataset (Krizhevsky and Hinton, 2009). ImageNet-LT (Liu et al., 2019b) is sampled from the the dataset of ILSVRC 2012 competition (Deng et al., 2009), and contains 115.8K images in 1,000 classes.

Evaluation. The evaluation protocol consists of standard classification accuracy, a newly designed experiment on the False Head Rate (FHR), and calibration with predictive uncertainty. 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.

Compared Baselines. We compare our method (LBD) with cross entropy baseline, rebalancing methods (CB Loss (Cui et al., 2019) and LDAM (Cao et al., 2019)), and ensemble methods (RIDE (Wang et al., 2020) and TLC (Li et al., 2022)). The numbers of classifiers in all ensemble models are set to be 3. We also compare the Bayesian predictive uncertainty with other uncertainty algorithms (Appendix F.2). We use $f(n_y) = n_y$ unless otherwise specified. More implementation details are in Appendix E.

4.2. Standard Classification

Classification accuracy is the most standard benchmark for long-tailed data, where the overall accuracy (Table 1) and accuracies for three class regions (Appendix F.1) are evaluated. We apply the one-hot utility to accord with standard accuracy metric. Our method consistently outperforms all other compared methods in terms of overall accuracy. For regional accuracies, our method achieves the best performances on all class regions in most cases. In particular, our method significantly outperforms previous methods on the crucial tailed data, while being comparable or even better on med and head classes. These results demonstrate the effectiveness of taking a Bayesian-decision-theory perspective on the long-tailed problem.

4.3. Tail-sensitive Classification with False Head Rate

Classifying tailed samples into head classes would often induce negative consequences in real-world applications. Therefore, we are interested in quantifying how likely it will happen, and further evaluating the tail sensitivity of the compared methods. Inspired by the false positive rate, we define the *False Head Rate* (FHR) as: $FHR = TH/(TT + TH)$, where TH is the number of samples that are labeled as tail but predicted as head, and TT is the number of samples that are labeled and predicted as tail. We also consider different settings of tail region, and select the last 25%, 50% and 75% classes as tail. We apply the tail-sensitive utility in Fig. 1(b) to our method. From Table 3, we observe significant improvements of LBD over previous methods under all settings, especially on the relatively small CIFAR datasets, which means that the "false head risk" is more severe on smaller datasets with scarce tailed samples. This shows the importance of taking the decision loss into account and also demonstrates the flexibility of our framework which is compatible with different utilities, leading to better performance for different types of tasks.

Table 1: Quantitative results of overall classification accuracy (in percentage). Our method (LBD) performs the best on all datasets.

Method	CIFAR-10-LT	CIFAR-100-LT	ImageNet-LT
CE	73.65±0.39	38.82±0.52	47.80±0.15
CB Loss	77.62±0.69	42.24±0.41	51.70±0.25
LDAM	80.63±0.69	43.13±0.67	51.04±0.21
RIDE	83.11±0.52	48.99±0.44	54.32±0.54
TLC	79.70±0.65	48.75±0.16	55.03±0.34
LBD	**83.75±0.17**	**50.24±0.70**	**55.73±0.17**

5. Conclusion

In this paper, we propose Long-tailed Bayesian Decision (LBD), a principled framework to solve long-tailed problems, with both theoretical explanation and strong empirical performance. Based on Bayesian Decision Theory, LBD unifies data distribution, posterior inference, and decision-making and further provides theoretical justification for existing techniques such as re-balancing and ensemble. In LBD, we introduce the integrated risk as the objective, find a tractable variational lower bound to optimize this objective, and apply particle optimization to efficiently estimate the complex posterior. For the real-world scenario with tail sensitivity risk, we design a tail-sensitive utility to pursue a better False Head Rate. In experiments, our framework outperforms the current SOTA even on large-scale real-world datasets like ImageNet.

References

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

James O Berger. *Statistical decision theory and Bayesian analysis.* Springer Science & Business Media, 1985.

Léon Bottou. Online algorithms and stochastic approxima-p tions. *Online learning and neural networks*, 1998.

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.

Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. *stat*, 1050:10, 2018.

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*, pages 694–710. Springer, 2020.

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*, pages 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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

Mudasir A Ganaie, Minghui Hu, et al. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*, 2021.

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*, pages 878–887. Springer, 2005.

David J Hand. Classifier technology and the illusion of progress. *Statistical science*, 21(1): 1–14, 2006.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

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

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*, pages 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*, pages 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.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pages 5436–5446. PMLR, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

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*, pages 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*, pages 6970–6979, 2022.

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*, pages 2980–2988, 2017.

Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, Jun Zhu, and Lawrence Carin. Accelerated first-order methods on the wasserstein space for bayesian inference. *stat*, 1050: 4, 2018.

Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pages 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*, pages 2970–2979, 2020.

Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

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*, pages 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)*, pages 181–196, 2018.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

Donna Katzman McClish. Analyzing a portion of the roc curve. *Medical decision making*, 9 (3):190–195, 1989.

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.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Giung Nam, Sunguk Jang, and Juho Lee. Decoupled training for long-tailed classification with stochastic representations. In *The Eleventh 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.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

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*, pages 440–445, 2021.

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

Mark J Schervish. *Theory of statistics*. Springer Science & Business Media, 2012.

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)*, pages 1–9. IEEE, 2016.

Meet P Vadera, Soumya Ghosh, Kenney Ng, and Benjamin M Marlin. Post-hoc loss-calibration for bayesian neural networks. In *Uncertainty in Artificial Intelligence*, pages 1403–1412. PMLR, 2021.

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*, pages 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.

Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for bayesian neural networks. In *International Conference on Learning Representations*, 2018.

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*, pages 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*, pages 247–263. Springer, 2020.

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*, pages 173–182. Springer, 2022.

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.

Appendix A. Decision Gain

As mentioned in the paper, we design the decision gain to be the following:

$$g(\boldsymbol{\theta}, d(\boldsymbol{x})) \propto -l(\boldsymbol{\theta}, d(\boldsymbol{x})) := \prod_{y'} p(y'|\boldsymbol{x}, \boldsymbol{\theta})^{u(y', d(\boldsymbol{x}))}. \tag{7}$$

Our design is different from previous work (Cobb et al., 2018), which uses

$$g(\boldsymbol{\theta}, d(\boldsymbol{x})) := \sum_{y'} p(y'|\boldsymbol{x}, \boldsymbol{\theta}) u(y', d(\boldsymbol{x})). \tag{8}$$

Both definitions achieve the goal of averaging the utility over the label distribution $p(y|\boldsymbol{x}, \boldsymbol{\theta})$. However, our design has two advantages: i) Eq. 7 is more stable for training. After taking the log, Eq. 7 becomes $\sum_{y'} u(y', d(\boldsymbol{x})) \log p(y'|\boldsymbol{x}, \boldsymbol{\theta})$ which is a weighted average of the logarithm of probabilities, while Eq. 8 becomes $\log \sum_{y'} u(y', d(\boldsymbol{x})) p(y'|\boldsymbol{x}, \boldsymbol{\theta})$, which is a weighted average of the probabilities. ii) Eq. 7 allows for more general and flexible utility functions whereas Eq. 8 requires utility u to be positive (otherwise we may not be able to compute the logarithm of Eq. 8). Due to these reasons, we use Eq. 7 in this paper.

Appendix B. On Computation of Inference Step

B.1. Train-test Discrepancy

At the inference step, we exploit the importance sampling to convert $p(\mathcal{D}_{test})$ to $p(\mathcal{D}_{train})$ and obtain a discrepancy ratio $p_{test}(\boldsymbol{x}, y)/p_{train}(\boldsymbol{x}, y)$. Recall that in long-tailed distribution, the training and testing data are semantically identical, and thus the model prediction must be the same for an input regardless of being in the training or testing set (i.e., $p_{train}(y|\boldsymbol{x}, \boldsymbol{\theta}) = p_{test}(y|\boldsymbol{x}, \boldsymbol{\theta})$). Therefore, the discrepancy ratio can be further simplified by:

$$\frac{p_{test}(\boldsymbol{x}, y)}{p_{train}(\boldsymbol{x}, y)} = \frac{p_{test}(y) p_{test}(\boldsymbol{x}|y)}{p_{train}(y) p_{train}(\boldsymbol{x}|y)} = \frac{p_{test}(y)}{p_{train}(y)}, \tag{9}$$

which only depends on the class probabilities of training and testing data. Since we assume a uniform distribution for the testing set in long-tailed data, the probability $p_{test}(\boldsymbol{y})$ would be a constant for all $\boldsymbol{x}$, and thus the discrepancy ratio is equivalent to:

$$\frac{p_{test}(y)}{p_{train}(y)} \propto \frac{1}{p_{train}(y)} \propto \frac{1}{f(n_y)}, \tag{10}$$

where f is an increasing function and n_y refers to the number of samples in the class y. We introduce the notation of $f(n_y)$ because the class probability only depend on the number of samples in this class.

The choices of f can determine different strategies used by previous re-balancing methods in long-tailed classification. 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 on the choice of discrepancy ratios will be conducted in Section F.3. Notably, our framework is compatible with all previous re-balancing methods as long as they can be expressed in the form of $1/f(n_y)$.

B.2. Particle-based Variational Distribution

To pursue the efficiency of model architecture, we use particle optimization (Liu and Wang, 2016; D'Angelo and Fortuin, 2021) to obtain the variational distribution: $q(\boldsymbol{\theta}) = \sum_{j=1}^{M} w_j \cdot \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_j)$, 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. The "particles" $\{\boldsymbol{\theta}_j\}_{j=1}^{M}$ are implemented by ensemble model, which has been empirically explored on the long-tailed data (Wang et al., 2020; Li et al., 2022). Our formulation gives theoretical justification to ensemble 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. Thus, estimating the full posterior is essential to provide a comprehensive characterization of the solution space. Particle optimization reduces the cost of Bayesian inference and is 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.3. Repulsive Regularization

In Eq. 3, the regularization term $KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}))$ guarantees the variational distribution to approach the posterior as training proceeds. If we assume the prior $p(\boldsymbol{\theta})$ to be Gaussian, the regularization can be extended to:

$$KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) = \lambda \int_{\Theta} ||\boldsymbol{\theta}||^2 \cdot q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \int_{\Theta} q(\boldsymbol{\theta}) \log q(\boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{\lambda}{M} \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||^2 - H(\boldsymbol{\theta}), \tag{11}$$

where λ is a constant, Θ is the parameter space, and $H(\boldsymbol{\theta})$ is the entropy of $\boldsymbol{\theta}$. The L_2-regularization prevents the model from over-fitting and the entropy term applies a *repulsive force* to the particles to promote their diversity, pushing the particles to the target posterior (D'Angelo and Fortuin, 2021). A simple approximation for the entropy is used in this paper:

$$H(\boldsymbol{\theta}) \propto \frac{1}{2} \log |\hat{\Sigma}_{\boldsymbol{\theta}}|, \tag{12}$$

where $\hat{\Sigma}_{\boldsymbol{\theta}}$ is the covariance matrix estimated by those 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 repulsive force, and is computed by:

$$KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) \propto \frac{\lambda}{M} \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||^2 - \frac{1}{2} \sum_{k} \log{(\overline{\boldsymbol{\theta}^2} - \overline{\boldsymbol{\theta}}^2)_k}. \tag{13}$$

Our regularization is different from existing diversity regularization (Wang et al., 2020), and is more principled and naturally derived from the integrated gain.

In summary, our method, with principled design and theoretical justification, is essentially cheap and easy to implement and can be used as a drop-in replacement for existing rebalancing and ensemble methods in general long-tailed problems.

Appendix C. Inference Step

Proof. We denote the training and testing sets as $\mathcal{D}_{train} = \{\mathcal{X}_{train}, \mathcal{Y}_{train}\}$ and $\mathcal{D}_{test} = \{\mathcal{X}_{test}, \mathcal{Y}_{test}\}$ respectively. The maximization objective would be:

$$\begin{aligned}\log G(d) &= \log \mathbb{E}_{(\boldsymbol{x},y)\sim p_{test}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}_{test}})} g(\boldsymbol{\theta}, d(\boldsymbol{x})) \\ &\overset{(a)}{\geq} \mathbb{E}_{(\boldsymbol{x},y)\sim p_{test}(\boldsymbol{x},y)} \log \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}_{test}})} g(\boldsymbol{\theta}, d(\boldsymbol{x})) \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim p_{test}(\boldsymbol{x},y)} \log \int_\Theta q(\boldsymbol{\theta}) g(\boldsymbol{\theta}, d(\boldsymbol{x})) \frac{p(\boldsymbol{\theta}|\mathcal{D}_{test})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &\overset{(b)}{\geq} \mathbb{E}_{(\boldsymbol{x},y)\sim p_{test}(\boldsymbol{x},y)} \int_\Theta q(\boldsymbol{\theta}) \log \left[g(\boldsymbol{\theta}, d(\boldsymbol{x})) \frac{p(\boldsymbol{\theta}|\mathcal{D}_{test})}{q(\boldsymbol{\theta})} \right] d\boldsymbol{\theta}.\end{aligned} \tag{14}$$

Here, (a) and (b) are by Jensen's inequality (Jensen, 1906). We will separately discuss the components in RHS of Eq. 14 below. First, by importance sampling (Kloek and Van Dijk, 1978), the outer expectation over data distribution would be:

$$\begin{aligned}\mathbb{E}_{(\boldsymbol{x},y)\sim p_{test}(\boldsymbol{x},y)} \psi(\boldsymbol{x}, y) &= \int_\mathcal{D} \psi(\boldsymbol{x}, y) p_{test}(\boldsymbol{x}, y) d(\boldsymbol{x}, y) \\ &= \int_\mathcal{D} \frac{p_{test}(\boldsymbol{x}, y)}{p_{train}(\boldsymbol{x}, y)} \psi(\boldsymbol{x}, y) p_{train}(\boldsymbol{x}, y) d(\boldsymbol{x}, y) \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim p_{train}(\boldsymbol{x},y)} \frac{p_{test}(\boldsymbol{x}, y)}{p_{train}(\boldsymbol{x}, y)} \psi(\boldsymbol{x}, y),\end{aligned} \tag{15}$$

where $\psi(\boldsymbol{x}, y)$ denotes any expression with respect to $(\boldsymbol{x}, y)$. Second, for the part inside the integral, we have:

$$\begin{aligned}&\int_\Theta q(\boldsymbol{\theta}) \log \left[g(\boldsymbol{\theta}, d(\boldsymbol{x})) \frac{p(\boldsymbol{\theta}|\mathcal{D}_{test})}{q(\boldsymbol{\theta})} \right] d\boldsymbol{\theta} \\ &= \int_\Theta q(\boldsymbol{\theta}) \log \left[g(\boldsymbol{\theta}, d(\boldsymbol{x})) \cdot \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \cdot \frac{p(\mathcal{Y}_{test}|\mathcal{X}_{test}, \boldsymbol{\theta})}{p(\mathcal{Y}_{test}|\mathcal{X}_{test})} \right] d\boldsymbol{\theta} \\ &= \int_\Theta q(\boldsymbol{\theta}) \left[\log g(\boldsymbol{\theta}, d(\boldsymbol{x})) - \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} + \log \prod_t p(y_t|\boldsymbol{x}_t, \boldsymbol{\theta}) - \log p(\mathcal{Y}_{test}|\mathcal{X}_{test}) \right] d\boldsymbol{\theta} \\ &= \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log g(\boldsymbol{\theta}, d(\boldsymbol{x})) - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) + \sum_t \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log p(y_t|\boldsymbol{x}_t, \boldsymbol{\theta}) - \log p(\mathcal{Y}_{test}|\mathcal{X}_{test}) \\ &= \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \sum_{y'} u(y', d) \log p(y'|\boldsymbol{x}, \boldsymbol{\theta}) - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) + \mathbb{E}_{(\boldsymbol{x},y)\sim p(\mathcal{D}_{test})} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) \\ &\quad - \log p(\mathcal{Y}_{test}|\mathcal{X}_{test}).\end{aligned} \tag{16}$$

Third, combining Eq. 15 and Eq. 16, we have:

$$\begin{aligned}\log G(d) \geq\ & \mathbb{E}_{(\boldsymbol{x},y)\sim p_{train}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{p_{test}(\boldsymbol{x}, y)}{p_{train}(\boldsymbol{x}, y)} \left[\log p(y|\boldsymbol{x}, \boldsymbol{\theta}) + \sum_{y'} u(y', d) \log p(y'|\boldsymbol{x}, \boldsymbol{\theta}) \right] \\ & - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) + C,\end{aligned} \tag{17}$$

where $C = -\log p(\mathcal{Y}_{test}|\mathcal{X}_{test})$, as desired.

Appendix D. Relationship between Variational Distribution and the Testing Posterior

Proof. We show the relationship between $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathcal{D}_{test})$ by computing the KL divergence between them:

$$
\begin{aligned}
KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}_{test})) &= \int_{\Theta} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathcal{X}_{test}, \mathcal{Y}_{test})} d\boldsymbol{\theta} \\
&= \int_{\Theta} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta}) p(\mathcal{Y}_{test}|\mathcal{X}_{test})}{p(\boldsymbol{\theta}) p(\mathcal{Y}_{test}|\mathcal{X}_{test}, \boldsymbol{\theta})} d\boldsymbol{\theta} \\
&= \int_{\Theta} q(\boldsymbol{\theta}) \left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} - \sum_t \log p(y_t|\boldsymbol{x}_t, \boldsymbol{\theta}) + \log p(\mathcal{Y}_{test}|\mathcal{X}_{test}) \right] d\boldsymbol{\theta} \\
&= KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \mathbb{E}_{(\boldsymbol{x},y)\sim p_{test}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) + \log p(\mathcal{Y}_{test}|\mathcal{X}_{test}) \\
&\overset{(c)}{=} KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) - \mathbb{E}_{(\boldsymbol{x},y)\sim p_{train}(\boldsymbol{x},y)} \mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta})} \frac{p_{test}(\boldsymbol{x},y)}{p_{train}(\boldsymbol{x},y)} \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) - C.
\end{aligned}
\tag{18}
$$

Here, (c) is by importance sampling (see Eq. 15) and $C = -\log p(\mathcal{Y}_{test}|\mathcal{X}_{test})$.

Appendix E. Implementation Details

Due to practical reasons, we slightly modify the training objective of our framework in experiments.

Table 2: Hyper-parameter configurations.

Dataset	Base model	Optimizer	Batch size	Learning rate	Training epochs	Discrepancy ratio	λ	τ	α
CIFAR-10-LT	ResNet32	SGD	128	0.1	200	linear	5e-4	40	0.002
CIFAR-100-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

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:

$$
\exp\{-epoch/\tau\} \cdot \frac{1}{2} \sum_j \log{(\overline{\boldsymbol{\theta}^2} - \overline{\boldsymbol{\theta}}^2)_j}, \tag{19}
$$

where τ is a stride factor which 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.

Tail-sensitive utility. Although the tail-sensitive utility matrix in Fig. 1 is designed to address the problem of classifying too many tailed samples into head classes, it will also affect the accuracy of classification task. Therefore, the utility term in Eq. 3 needs re-scaling

Table 3: Quantitative results of False Head Rates under three different tail ratios and their averaged results. LBD consistently achieves the lowest rates under different scenarios.

$\mathcal{D}$	Method	FHR (%) @tail ratio ↓			
		25%	50%	75%	avg
CIFAR-10-LT	CE	21.10±0.43	37.87±0.57	48.75±1.39	35.91±0.54
	CB Loss	14.84±0.93	27.98±1.44	33.93±1.60	25.58±1.27
	LDAM	10.05±1.01	19.64±1.66	21.37±2.10	17.02±1.56
	RIDE	8.94±0.66	17.80±1.39	19.77±3.20	15.50±1.68
	TLC	10.42±0.64	20.27±0.77	22.24±1.53	17.64±0.93
	LBD	**4.99±0.32**	**11.76±0.29**	**11.01±1.28**	**9.25±0.49**
CIFAR-100-LT	CE	45.53±1.54	73.03±1.59	91.30±1.24	69.95±1.40
	CB Loss	24.88±0.34	48.41±1.24	74.38±1.47	49.22±0.83
	LDAM	21.22±0.99	43.04±1.18	65.62±1.31	43.29±1.04
	RIDE	18.83±0.70	39.50±1.53	62.01±2.70	40.11±1.62
	TLC	21.18±0.54	41.15±0.55	61.34±1.03	41.22±0.55
	LBD	**15.39±0.57**	**31.34±0.55**	**49.51±1.45**	**32.08±0.78**
ImageNet-LT	CE	3.99±0.08	12.77±0.29	30.99±0.40	15.92±0.17
	CB Loss	3.66±0.17	11.80±0.12	29.39±0.28	14.95±0.15
	LDAM	4.17±0.19	12.73±0.28	29.90±0.41	15.60±0.21
	RIDE	3.62±0.18	11.42±0.27	26.92±0.33	13.99±0.24
	TLC	3.47±0.13	11.49±0.13	27.12±0.13	14.03±0.09
	LBD	**2.70±0.09**	**9.68±0.25**	**24.42±0.19**	**12.27±0.08**

so that its negative effect on the accuracy is controllable:

$$\log p(y|\boldsymbol{x}, \boldsymbol{\theta}) + \frac{1}{\alpha} \cdot \sum_{y'} u(y', d) \log p(y'|\boldsymbol{x}, \boldsymbol{\theta}), \tag{20}$$

where α is the scaling factor. We can adjust the value of α to carefully control the effect of the tail-sensitive utility term, which will bring to us significant improvement on the False Head Rate with acceptable accuracy drop.

Computational cost. 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 LBD is comparable to previous ensemble models. Besides, compared with gradient-flow-based BNN like (D'Angelo and Fortuin, 2021), which typically uses 20 particles, our model is far more efficient with no more than 5 particles.

Other settings and hyper-parameters are concluded in Table 2. We use stepwise decaying learning rate and data augmentation for all compared baselines. The optimal values of those hyper-parameters are determined by grid search. The code is publicly available[1].

Table 4: Quantitative results of classification accuracies on three class regions. LBD outperforms previous methods in all class regions in most cases, especially on tailed data.

$\mathcal{D}$	Method	ACC (%) ↑		
		HEAD	MED	TAIL
CIFAR-10-LT	CE	93.22±0.26	74.27±0.42	58.51±0.62
	CB Loss	91.70±0.57	75.41±0.76	68.73±1.52
	LDAM	90.03±0.47	75.88±0.81	77.14±1.61
	RIDE	**91.49±0.40**	**79.39±0.61**	79.62±1.56
	TLC	89.47±0.33	74.33±0.96	76.39±0.98
	LBD	90.49±0.60	78.89±0.87	**82.33±1.16**
CIFAR-100-LT	CE	68.30±0.61	38.39±0.49	10.62±1.23
	CB Loss	62.53±0.44	44.36±0.96	20.50±0.51
	LDAM	63.58±0.93	42.90±1.03	23.50±1.28
	RIDE	69.11±0.54	49.70±0.59	28.78±1.52
	TLC	69.43±0.36	49.02±0.94	28.40±0.72
	LBD	**69.92±0.77**	**51.07±0.82**	**30.34±1.49**
ImageNet-LT	CE	53.46±0.36	45.92±0.19	44.03±0.24
	CB Loss	57.62±0.46	49.19±0.21	48.29±0.41
	LDAM	57.66±0.40	48.26±0.19	47.21±0.22
	RIDE	60.88±0.71	51.35±0.44	50.74±0.62
	TLC	61.19±0.53	52.35±0.31	51.56±0.35
	LBD	**62.18±0.28**	**53.06±0.22**	**51.98±0.40**

Appendix F. Additional Experimental Results

F.1. Classification on Different Class Regions

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 CIFAR-100-LT.

F.2. Calibration

In our method, the predictive uncertainty can be naturally obtained by the entropy of predictive distribution (Malinin and Gales, 2018). For the compared uncertainty algorithms, MCP is a trivial baseline which obtains uncertainty scores from the maximum value of softmax distribution, which is added to the RIDE (Wang et al., 2020) backbone; evidential uncertainty is rooted in the subjective logic (Audun, 2018), and is introduced to long-tailed classification by TLC (Li et al., 2022). We evaluate the uncertainty algorithms with AUC (McClish, 1989) and ECE (Naeini et al., 2015), which are shown in Table 5. Our Bayesian predictive uncertainty outperforms the other two counterparts and has a remarkable advantage on the ECE metric, demonstrating the superiority of using principled Bayesian uncertainty quantification.

[1] https://github.com/lblaoke/LBD

Table 5: Quantitative results of calibration of different uncertainty algorithms. LBD outperforms previous methods remarkably on both metrics and all datasets.

Dataset	Algorithm	AUC (%) ↑	ECE (%) ↓
CIFAR-10-LT	MCP	79.98±0.10	14.33±0.37
	evidential	83.20±0.59	13.24±0.55
	Bayesian	**86.83±0.68**	**9.84±0.17**
CIFAR-100-LT	MCP	80.48±0.51	23.75±0.51
	evidential	77.37±0.33	21.64±0.47
	Bayesian	**81.24±0.25**	**10.35±0.28**
ImageNet-LT	MCP	84.02±0.24	18.35±0.12
	evidential	81.45±0.13	15.29±0.12
	Bayesian	**84.45±0.09**	**8.72±0.13**

Table 6: Ablation study on the choice of utility function, compared by False Head Rates and classification accuracy on CIFAR-100-LT.

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

F.3. Ablation Studies

Utility Function. The effectiveness of tail-sensitive utility is shown in Table 6, where we compare the one-hot and tail-sensitive utilities in terms of False Head Rate and classification accuracy. By applying the tail-sensitive utility, the performances on False Head Rate can be significantly improved (18.00%) with negligible drop on the classification accuracy (0.04%).

Train-test Discrepancy. We compare five different forms of discrepancy ratio in terms of classification accuracy in Table 7 and Fig. 2. We also analyze the properties of the compared discrepancy ratios. The differences of $f(n_y)$ show up when n_y is large, and it can be measured by the growth rate of weight values (i.e., $1/f(n_y)$) between the first and

Table 7: Ablation study on the choice of discrepancy ratio, compared by classification accuracies on CIFAR-100-LT.

Discrepancy ratio			Weight value			ACC (%) ↑
			first class	last class	growth (%)	
linear (Wang et al., 2017)	$f(n_y) =$	n_y	0.0020	0.1667	**8250**	**50.17±0.25**
effective (Cui et al., 2019)		$\frac{1-\beta^{n_y}}{1-\beta}$	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
plain		constant	1.0000	1.0000	0	43.27±0.30

Table 8: Ablation study on repulsive force, compared by uncertainty calibration on CIFAR-100-LT.

Repulsive force	AUC (%) ↑	ECE (%) ↓	ACC (%) ↑
✓	**81.24±0.25**	**10.35±0.28**	**50.24±0.70**
×	75.94±0.56	13.40±0.80	50.15±0.41

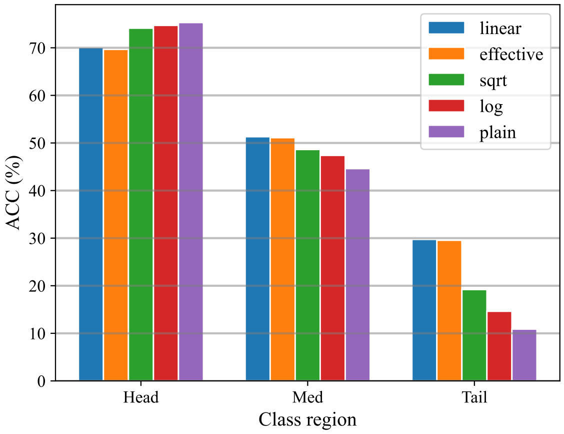


Figure 2: Visual results of classification with respect to the choice of discrepancy ratio on CIFAR-100-LT.

the last class. We find that as the growth rate becomes larger, the overall accuracy will be better accordingly, which shows the severity of class imbalance.

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

Repulsive Force. We evaluate the effectiveness of repulsive force in Table 8. The repulsive force effectively pushes the particles 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 accuracy by promoting the diversity of particles.

Number of Particles. Generally, using more classifiers in ensembles will induce better performances. However, we also need to balance the performance with the computational cost. We visualize the classification accuracies under different numbers of particles in Fig. 3. The error bars are scaled to be 2σ, where σ is the standard deviation from repeated experiments. The accuracy curves are all logarithm-like and the accuracy improvement is hardly noticeable for more than six particles. However, the computational cost is increasing in a linear speed. Therefore, we recommend using no more than six particles in practice for a desirable performance-cost trade-off.

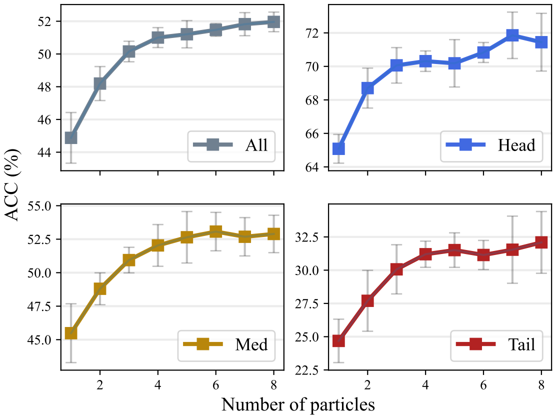


Figure 3: Visual results of classification with respect to the number of particles on CIFAR-100-LT.

Appendix G. Related Works

Long-tailed Classification. To overcome long-tailed class distributions, over-sampling (Han et al., 2005) uses generated data to compensate the tailed classes, under-sampling (Liu et al., 2008) splits the imbalanced dataset into multiple balanced subsets, and data augmentation (Chu et al., 2020; Kim et al., 2020; Liu et al., 2020) introduces random noise to promote model's robustness. Recent advances focus on improving training loss functions and model architectures. For example, re-weighting methods (Cao et al., 2019; Cui et al., 2019; Lin et al., 2017; Menon et al., 2020; Wu et al., 2020; Mahajan et al., 2018) adjust the loss function by class probabilities in the training data, OLTR (Liu et al., 2019b) transfers the knowledge learned from head classes to the learning of tailed classes, LFME (Xiang et al., 2020) uses multiple teacher models to learn relatively balanced subsets of training data, RIDE (Wang et al., 2020) develops a multi-expert framework to promote the overall performances with ensemble model, and TLC (Li et al., 2022) exploits the evidential uncertainty to optimize the multi-expert framework. Besides, SRepr (Nam et al., 2023) explores Gaussian noise in stochastic weight averaging to obtain stochastic representation, and SADE (Zhang et al., 2022) considers the case of non-uniform testing distributions in long-tailed problems. These methods are largely designed based on empirical heuristics, and thus their performances are not explainable and guaranteed. In contrast, our method is rooted in Bayesian principle and decision theory, inheriting their theoretical guarantees and explanation.

Bayesian Decision Theory. Bayesian Decision Theory is introduced in Robert et al. (2007); Berger (1985). It provides a bridge which connects posterior inference, decision, and data distribution. Loss-calibrated EM (Lacoste-Julien et al., 2011) exploits the posterior risk (Schervish, 2012) to simultaneously consider inference and decision. Cobb et al. (2018) further extends this method using dropout-based Bayesian neural networks. Loss-EP (Morais and Pillow, 2022) applies the technique of loss calibration in expectation propagation. Post-hoc Loss-calibration (Vadera et al., 2021) develop an inference-agnostic way to learn the high-utility decisions. These methods all use the notion of utility to represent their prior knowledge about the application-related risks, and exploit the posterior risk. While they show great advantages in some applications, none of them consider the data distribution, which prevents their applications on long-tailed data. Our method overcomes this limitation by introducing the integrated risk, which unifies data distributions, inference, and decision-making.

Ensemble and Particle Optimization. Ensemble models combine several individual deep models to obtain better generalization performances (Lakshminarayanan et al., 2017; Ganaie et al., 2021), which is inspired by the observation that multiple i.i.d. initializations are less likely to generate averagely "bad" models (Dietterich, 2000). Ensemble models can also be used to approximate the posterior with the technique of *particle optimization*, which is first studied in Stein variational gradient descent (SVGD, Liu and Wang (2016)) and then explored by Liu et al. (2019a); Korba et al. (2020); D'Angelo and Fortuin (2020). Liu (2017) analyzes SVGD in a gradient-flow perspective. Wang et al. (2018) performs the particle optimization directly in the function space. Chen et al. (2018); Liu et al. (2018) put the particle optimization in the 2-Wasserstein space. D'Angelo and Fortuin (2021) implements particles by introducing a repulsive force in the gradient flow. Instead of directly modeling the gradient flow, our framework optimizes the particles through stochastic gradient descent (SGD, Bottou (1998)), with repulsive force induced by the integrated risk objective. Compared to existing particle optimization, our method is easy and cheap to implement, which is especially beneficial for large deep models.

Appendix H. Future Directions

Our method is simple to use in general long-tailed problems, providing superior accuracy on all types of classes and uncertainty estimation. We believe there is considerable space for future developments that build upon our method and we list a few below:

Long-tailed Regression. Long-tail problem also exists in regression, where the distribution of targets can be heavily imbalanced. With some adjustments on the decision gain, our framework might also be adapted to regression.

Utility Function. Beyond long-tailed classification, there are other tasks which also need specific utility functions. For example, we might have to separately deal with the relationship between categories due to their semantic connections. In this case, all of the values in the utility matrix will need re-calculating.

Dataset Shift. In more general dataset shift scenarios like out-of-distribution data, the assumption about semantically identical training and testing sets will be no longer valid. Another example is about the distribution of testing data. If it is assumed to be not uniform, the discrepancy ratio $p_{test}(y)/p_{train}(y)$ will no longer be expressed in the form $1/f(n_y)$, but a more general form.