Towards Inter-class and Intra-class Imbalance in Class-imbalanced Learning

1st Zhining Liu
Jilin University
Changchun, China
liuzn19@mails.jlu.edu.cn

2nd Pengfei Wei
ByteDance
Singapore
pengfei.wei@bytedance.com

3rd Zhepei Wei
Jilin University
Changchun, China
weizp19@mails.jlu.edu.cn

4th Boyang Yu
Jilin University
Changchun, China
yuby19@mails.jlu.edu.cn

5th Jing Jiang
University of Technology Sydney
Sydney, Australia
jing.jiang@uts.edu.au

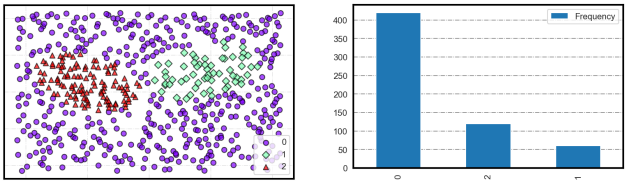
6th Wei Cao
Microsoft Research
Beijing, China
weicao@microsoft.com

7th Jiang Bian
Microsoft Research
Beijing, China
jiang.bian@microsoft.com

8th Yi Chang
Jilin University
Changchun, China
yichang@jlu.edu.cn

***Abstract*—Imbalanced Learning (IL) is an important problem that widely exists in data mining applications. Typical IL methods utilize intuitive class-wise resampling or reweighting to directly balance the training set. However, some recent research efforts in specific domains show that class-imbalanced learning can be achieved without class-wise manipulation. This prompts us to think about the relationship between the two different IL strategies and the nature of the class imbalance. Fundamentally, they correspond to two essential imbalances that exist in IL: the difference in quantity** *between examples from different classes* **as well as** *between easy and hard examples within a single class*, **i.e., *inter-class* and *intra-class* imbalance. Existing works fail to explicitly take both imbalances into account and thus suffer from suboptimal performance. In light of this, we present Duple-Balanced Ensemble, namely DUBE , a versatile ensemble learning framework. Unlike prevailing methods, DUBE directly performs inter-class and intra-class balancing without relying on heavy distance-based computation, which allows it to achieve competitive performance while being computationally efficient. We also present a detailed discussion and analysis about the pros and cons of different inter/intra-class balancing strategies based on DUBE . Extensive experiments validate the effectiveness of the proposed method. Code and examples are available at Github[1] and ReadtheDocs[2].**

***Index Terms*—class-imbalance, imbalanced learning, imbalanced classification, ensemble learning, data resampling.**



(a) Class distribution (b) Inter-class imbalance

(c) Error distribution (d) Intra-class imbalance

Fig. 1. An illustrative example of inter-class and intra-class imbalance, best viewed in color. Fig. (a): an example imbalanced toy dataset containing 3 classes. Fig. (b): the inter-class imbalanced distribution (of the number of class samples). Fig. (c): the dataset resized by the prediction error of a classifier. Fig. (d): the intra-class imbalanced distribution (of the prediction error).

I. INTRODUCTION.

Most of well-known machine learning algorithms work well under the balanced sample assumption where the training samples are approximately evenly distributed over classes [1]. However, this assumption does not always hold in practice. Due to the naturally-skewed class distributions, class-imbalance has been widely observed in many real-world application domains including computer vision, fraud detection, intrusion detection, medical diagnosis, etc [2]–[4]. Facing the class imbalance problem, most of canonical classification algorithms, like decision tree, support vector machine, and neural networks, suffer from a "majority bias" issue. More concretely, since the learning process is oriented by the global accuracy, these algorithms tend to induce a bias towards the majority classes, which may lead to a seemingly good performance in terms of the global accuracy but a poor one in terms of the accuracy on minority classes. Such a result is suboptimal as minority classes are typically of primary interest [3].

To alleviate the "majority bias" issue, an emerging research field, known as Imbalanced Learning (IL), has attracted increasing research efforts in recent years. Most of the existing typical IL solutions are developed under an intuitively appealing strategy, which is to *reduce class-imbalance by class-wise resampling or reweighting*. This can be achieved by over-sampling minority classes (e.g., [5]–[7]), under-sampling majority classes (e.g., [8]–[11]), or assigning different mis-classification costs to different classes (e.g., [12]–[15]). Nevertheless, we further observe that some recent works arise under a

Yi Chang is the corresponding author.

[1] https://github.com/ICDE2022Sub/duplebalance

[2] https://duplebalance.readthedocs.io

new strategy without any class-level operation. These methods *implicitly facilitate class balance through hard example mining* and have achieved notable success in handling IL (also known as the long-tail problem) in certain research areas (e.g., [16]–[21]). The fact that class imbalanced learning can be achieved without class-wise manipulation prompts us to think about some interesting questions:

- How do the two different strategies, i.e., **1st** class-wise manipulation and **2nd** hard example mining, help with IL? And how are the existing IL solutions raised under them connected to the nature of the class imbalance problem?
- On the basis of the above questions, what is the key to achieving better imbalanced learning?

In this paper, we answer the above questions through a comprehensive discussion on the two kinds of imbalance presented in IL, i.e., *inter-class imbalance* and *intra-class imbalance*. Fig. 1 gives illustrative examples of the two types of imbalance. We can observe that the two types of imbalance co-exist in a class-imbalanced dataset. The inter-class imbalance describes the uneven sample distribution among different classes as shown in the Fig. 1(a) and 1(b). Such imbalance can be readily observed and it is the target that most existing IL solutions under the first strategy focus on. The intra-class imbalance depicts the quantity variance of easy and hard examples as shown in the Fig. 1(c) and 1(d). We note that such imbalance is an important indicator that reflects data and task complexity. Ignoring it will result in a failure to exploit more informative supervised signals and lead to suboptimal performance. Hard example mining methods developed under the second strategy alleviate intra-class imbalance as they assume hard-to-learn samples are more informative and emphasize their importance. To this point, we can answer the first question: *the class-wise manipulation alleviates the inter-class imbalance, while hard example mining alleviates the intra-class imbalance.*

We note that some IL methods [7], [8], [22], which focus on "better" samples while balancing the data with different heuristics, can be seen as the indirect combination of the above two strategies, although none of them realizes the two types of imbalances and explicitly works on them. These methods basically outperform the naïve solutions [3]. However, most of these methods rely heavily on neighborhood computing (e.g., [6], [7], [23], [24]), which requires well-defined distance metrics and introduces extra computational overhead. Some other methods constantly emphasize misclassified samples during model training (e.g., [22]) and are thus susceptible to noise and outliers in the data. Moreover, recent works under the second strategy are mostly designed for training deep neural networks (e.g., [16], [17], [20]), thus lack versatility to other learning models and tasks. With all these factors in mind, we believe that *the key to achieving better IL is the ability to explicitly handle both types of imbalances, while avoiding the shortcomings of existing approaches*.

In light of this, we propose DUBE (Duple-Balanced Ensemble). It is a versatile ensemble learning framework for IL, with a direct focus on handling inter-class and intra-class imbalance. For inter-class balancing, DUBE corporates with the commonly used under-/over-sampling strategy. We demonstrate their pros and cons respectively with a detailed analysis. Additionally, we discuss the potential of hybrid-sampling in IL, especially in DUBE . For intra-class balancing, we show that the direct hard example mining (HEM) is not the optimal solution due to its vulnerability to outliers. In response to this, we design a robust intra-class balancing mechanism by considering the error density distribution. Specifically, this mechanism works in a similar way as HEM when the data is clean, but penalizes the importance of high-error samples (that are likely to be outliers) when the data is noisy. Finally, we introduce a simple yet effective way of data augmentation thus to **(i)** prevent the risk of overfitting caused by replication-based resampling, and **(ii)** further diversify the ensemble to achieve better generalization. Note that, different from many of the prevailing methods, DUBE involves no distance-based computing.

To sum up, this paper makes the following contributions:

- We carry out a preliminary explicit exploration of the inter-class and intra-class imbalance in IL problem and discuss the relations between the two kinds of class-imbalance and the two popular strategies of existing IL methods.
- Based on the discussion of inter-class and intra-class balancing, we propose DUBE , a simple, generic, and efficient ensemble learning framework for IL. Extensive experiments demonstrate the effectiveness of DUBE.
- We also present detailed discussions and analyses on the pros and cons of different inter/intra-class balancing strategies. Our findings may shed some light on developing better algorithms for class-imbalanced machine learning.

II. Related Works.

He et al. [1], [3], Guo et al. [2] and Krawczyk et al. [25] provided systematic surveys of algorithms and applications of class-imbalanced learning. In this section, we review the research topics that are closely related to this paper.

Inter-class Balancing Solutions. As mentioned earlier, the vast majority of existing IL solutions fall into this category. They can be divided into two regimes: resampling and reweighting. *Resampling* methods focus on directly modifying the training set to balance the class distribution (e.g., over-sampling [5]–[7] and under-sampling [8]–[11]). Beyond the naïve solutions, previous efforts have adopted different heuristics to guide their resampling process. For example, [7], [8] aim to generate or keep instances that are close to the borderline/overlap area. But on the contrary, [10], [23], [24] consider examples located in the overlap zone to be detrimental to learning, and therefore discard them from the trainig set. Such methods often rely on exploring distance-based neighborhood information, e.g., SMOTE over-sampling [5] and its variants [6], [7], [26], many denoising under-sampling techniques [9]–[11], as well as their hybrid usages [23], [24], [27]. On the other hand, *Reweighting* approaches assign different misclassification costs to different classes (e.g., cost-sensitive learning [14], [15]), thus forcing the model to focus on the pattern of minority classes.

However, these methods do not explicitly account for inter-class and intra-class imbalances in their design, and suffer from unsatisfactory performance, high computational cost, and poor applicability. Distance-based resampling requires a well-defined distance metric, which may not be available in practice since the data may contain categorical features and missing values [19]. Moreover, some of these algorithms run extremely slow on large-scale datasets as the cost of calculating the distance between each instances grows quadratically with the size of the dataset. Reweighting, i.e., cost-sensitive learning, often requires targeted modifications to the learning algorithms [2]. More importantly, in most cases, it is difficult to obtain an appropriate cost matrix given by domain experts [28].

Intra-class Balancing Solutions. Most of the works in this group are raised in recent years with the boom in deep learning approaches, especially in computer vision applications such as image classification and object detection [29], [30]. They (implicitly) reweight the gradient update of samples based on their difficulties or losses, i.e., down-weight the well-classified examples and assign more weights to hard examples. Such hard example mining (HEM) methods have achieved notable success in many areas. For example, *Qi et al.* [20] propose Class Rectification Hard Mining to handle imbalanced image classification. To deal with the background-foreground imbalance in object detection, Online HEM [16] only back-propagates gradients for hard examples in the later phase of training. *Lin et al.* [17] further propose a uniform HEM loss function FocalLoss for dense object detection.

We note that there are some inter-class balancing solutions that that also incorporate the idea of HEM. For instance, *Liu et al.* [22] iteratively discard well-classified majority samples when perform under-sampling during ensemble training. The IL methods (e.g., [31]–[33]) that based on adaptive boosting (AdaBoost) can also be considered as incorporating HEM, as AdaBoost emphasizes those instances misclassified by previous classifiers. Nevertheless, this does not mean that HEM is the optimal solution for intra-class balancing. It may work well when the data is clean with no presence of noise. But on a noisy dataset, most of hard examples are likely to be outliers. They will be wrongly reinforced by direct HEM, which degrades the generalization performance. Therefore, an adaptive approach is needed to achieve more robust intra-class balancing.

Ensemble Imbalanced Learning. By merging the outputs of multiple classifiers, ensemble imbalanced learning (EIL) is known to effectively improve typical IL solutions (e.g., [19], [22], [31], [32], [34]). These EIL solutions are gaining increasing popularity [2] and have demonstrated competitive performance in many IL tasks [25]. But most of them are direct combinations of a resampling/reweighting scheme and an ensemble learning framework, e.g., SMOTE [5]+ADABOOST [35]=SMOTEBOOST [31]. It means that these methods inherit the same shortcomings of existing inter-class balancing strategies, such as the reliance on neighborhood relationships based on distance computing. Consequently, albeit EIL techniques effectively lower the variance introduced by resampling/reweighting, there is still a lot of room for improvement. Note that the DuBE proposed in this paper is also a generalized iterative ensemble learning framework.

Data Complexity Factors. Finally, several research efforts have discussed the data complexity factors present in class-imbalanced learning. These factors are believed to be widely existed in imbalanced datasets and contribute to the poor classification performance. Specifically, *Garcia et al.* [36] and *Koziarski et al.* [37] discussed the influence of noise examples in imbalanced learning. Besides, [38]–[40] suggest that class overlap/separability is the major problem responsible for degradation of classifier's performance. Some other works focus on the problem of small disjuncts [41], [42], i.e., presence of small-sized clusters (called subconcepts) containing examples from one class and located in the region of another class. We note, however, that these data complexity factors are closely related to each other: severe class overlap can yields more outliers who play a similar role to noise examples [43], and an increase in the amount of noise could also induce more small disjuncts [44], and vice versa. Compare with them, the intra-class imbalance introduced in this paper is a higher-level indicator of *task complexity*. It considers the data distribution in terms of learning difficulty (with respect to a given model), which reflects the ultimate effect of these data complexities on the classifier learning process.

III. PRELIMINARIES.

Before getting into the DUBE framework, we first briefly introduce the notations used in this paper and formally define the class-imbalanced learning problem in this section.

TABLE I
DEFINITIONS OF BASIC NOTATIONS.

Notation	Definition
d	Input dimensionality.
m	Number of classes.
N	Number of data instances.
$\mathcal{X} : \mathbb{R}^d$	Input feature space.
$\mathcal{Y} : \{c_1, c_2, ..., c_m\}$	Output label space.
$s : (x, y)$	Data instance, where $x \in \mathcal{X}, y \in \mathcal{Y}$.
$D : \{s_i\}_{i=1,2,...,N}$	Dataset with N examples.
$D_c : \{(x, y)\|y = c\}$	Subset of data examples from class c.
$F : \mathcal{X} \to \mathcal{Y}$	A learning-based classifier.
$f : \mathcal{X} \to \mathcal{Y}$	Base learner of an ensemble classifier.
$F_k : \mathcal{X} \to \mathcal{Y}$	An ensemble classifier with k base learners.

Notations. Formally, let N denote the number of instances in the dataset and d be the input dimensionality, i.e., the number of input features. We can define the feature space $\mathcal{X} : \mathbb{R}^d$ and label space $\mathcal{Y} : \{c_1, c_2, ..., c_m\}$, where m is the number of classes. Then a data sample is $s : (x, y)$ with $x \in \mathcal{X}, y \in \mathcal{Y}$. Likewise, a dataset D can be represented as $D : \{s_i\}_{i=1,2,...,N}$. A classifier is represented by $F(\cdot)$, which is a projection from the feature space to the label space, i.e., $F : \mathcal{X} \to \mathcal{Y}$. Moreover, we use $f(\cdot)$ to denote a base learner of an ensemble classifier, and F_k as an ensemble classifier with k base learners. Table I summarizes the main notation definitions in this paper.

Problem definition. Note that the most of the existing works are conducted based on the binary case of imbalanced

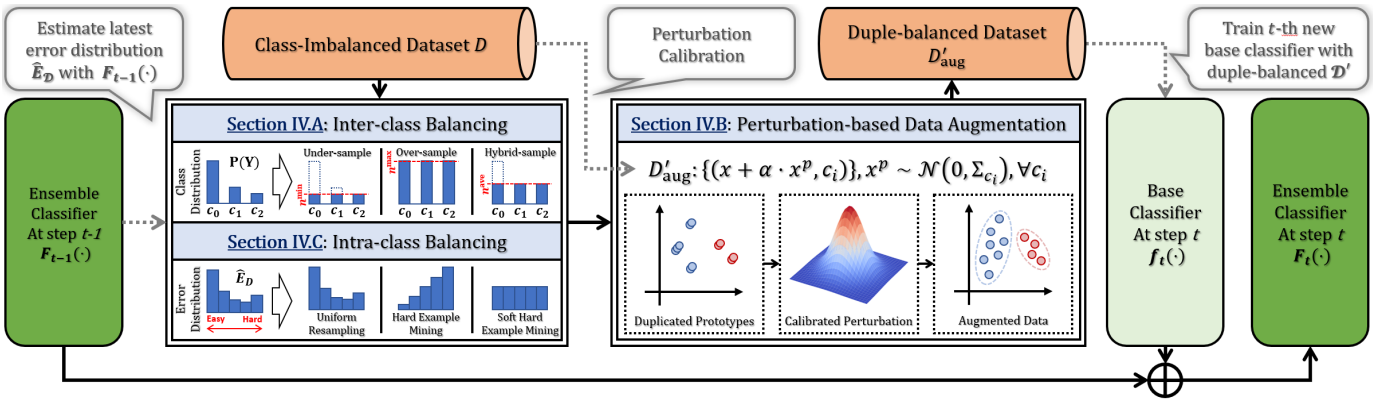


Fig. 2. Overview of the proposed DUBE Framework. Best viewed in color.

learning, i.e., $m = 2$ [1]–[3]. Without loss of generality, we describe the binary IL problem in this section, and extension to the multi-class scenario is straightforward. In this case, we consider the majority class c_{maj} and the minority class c_{min}. Let's denote the set of data examples from class c as $D_c : \{(x, y)|y = c\}$. Then the class-imbalance refers to the fact that $|D_{c_{\text{maj}}}| \gg |D_{c_{\text{min}}}|$[3]. That is, the examples are not evenly distributed over different classes, i.e., the underlying class (marginal) distribution $\mathbf{P}(\mathrm{Y})$ is skewed. Under such class-imbalance, the learning process of canonical machine learning methods are likely to be dominant by the majority class examples due to the accuracy-oriented learning objectives [1]. This usually leads to poor prediction performance for minority classes. Therefore, *the goal of imbalanced learning (IL) is to learn an unbiased classifier $F : \mathcal{X} \to \mathcal{Y}$ from a skewed dataset D*. With the above notations and definitions, we now present our DUBE framework and the associated analysis.

IV. METHODOLOGY.

In this section, we describe the proposed Duple-Balanced Ensemble (abbreviate as DUBE) framework for imbalanced learning. As previously described, DUBE focuses on directly addressing the *inter-class* and *intra-class* imbalance simultaneously. This is achieved by performing duple-balanced resampling within iterative ensemble training. Specifically, in t-th iteration, inter-class balancing is achieved by controlling the target class distribution of resampling, and intra-class balancing is achieved by assigning instance-wise sampling probabilities according to the prediction of the current ensemble model $F_{t-1}(\cdot)$. A new base classifier $f_t(\cdot)$ is then fitted to the resampled dataset D' and added to the ensemble to form $F_t(\cdot)$.

Correspondingly, DuBE consists of several mechanisms to achieve these two kinds of balancing. For inter-class balancing, we implement the commonly used under-sampling and over-sampling, as well as a hybrid-sampling that combines them. A simple perturbation-based data augmentation technique is introduced to prevent the risk of overfitting brought about by replication-based over-sampling, and further diversify the ensemble to achieve better generalization. As for intra-class balancing, in addition to naive direct hard example mining (HEM), we propose a robust approach that performs soft HEM by re-balancing the error density distribution. We discuss the advantages and disadvantages of all these solutions in details with intuitive examples. Fig. 2 shows an overview of DUBE. We will cover the technical details in the rest of this section.

A. Inter-class Balancing (InterCB).

We first introduce the three InterCB strategies considered in DUBE, i.e., under-sampling, over-sampling, and hybrid-sampling, and then discuss why they are effective for InterCB.

(1) *Under-sampling*: The class containing the fewest samples is considered the minority class c_{min}, and all others are treated as majority ones. The majority classes are then under-sampled until they are of the same size as c_{min}, i.e., $\forall c \in \mathcal{Y}, |D'_c| = |D_{c_{\text{min}}}|$, where D' represents the resampled dataset. **(2)** *Over-sampling*: Conversely, the class containing the largest number of samples is considered as the majority class c_{maj}. Other classes are over-sampled via instance duplication until they are of the same size as c_{maj}, i.e., $\forall c \in \mathcal{Y}, |D'_c| = |D_{c_{\text{maj}}}|$. **(3)** *Hybrid-sampling*: We further consider a combination of the aforementioned strategies, namely hybrid-sampling. Majority and minority classes are distinguished by whether they contain more than average number of samples. Then, we under/over-sample the majority/minority classes to bring them to the same size, i.e., $\forall c \in \mathcal{Y}, |D'_c| = \Sigma_{i=1}^{m} |D_{c_i}|/m = \frac{|D|}{m}$.

An example problem. In this section, we use an example to illustrate the effectiveness of the different InterCB strategies described above. Please note that *the following discussion is intended to provide an intuition about how different approaches can help IL, rather than a analysis of their theoretical properties.* With this premise in mind, we consider a simple imbalanced classification setting, as shown in Fig. 3. It is a one-dimensional two-class imbalanced dataset, with 3 samples from the minority class and 15 samples from the majority class. Without loss of generality, we assume the minority class to be positive, and denote the minority/majority class as c_+/c_-. i.e., $c_{\text{min}} := c_+ = 1$ and $c_{\text{maj}} := c_- = -1$. The imbalance ratio (IR) is then $|D_{c_-}|/|D_{c_+}| = 15/3 = 5$. Each class is sampled

[3]The $|\cdot|$ operator here denotes the cardinality of the set.

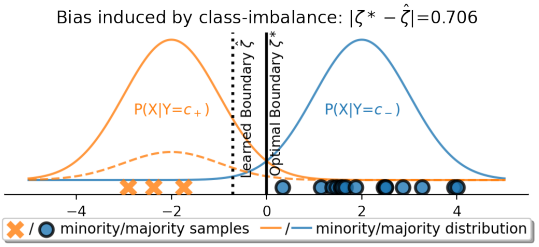


Fig. 3. A toy imbalanced classification problem, best viewed in color. It contains 3 minority samples (the "×"s) and 15 majority samples (the "○"s), i.e., imbalance ratio $|D_{c_{\text{max}}}|/|D_{c_{\text{min}}}| = 5$. The underlying distributions of both classes are also included in the figure (colored solid lines), which are one-dimensional Gaussian distributions $\mathcal{N}(\mu, \sigma^2)$ with different μ. The solid black line indicates the optimal decision boundary ζ^* with respect to the underlying distribution; and the dotted line represents the max-margin boundary $\hat{\zeta}$ on the sampled dataset. The bias induced by class-imbalance is defined as $|\zeta^* - \hat{\zeta}|$.

from a normal (Gaussian) distribution $\mathcal{N}(\mu, \sigma^2)$, with the same σ=1 but different μ, where $\mu_- - \mu_+ = 4$. It is obvious that the optimal decision boundary ζ^* with respect to the underlying distrbution $P(X|Y)$ should be $(\mu_+ + \mu_-)/2$ (the solid black line in Fig. 3). That is, the optimal classifier predicts a sample to be positive if $x < (\mu_+ + \mu_-)/2$ and negative if otherwise. However, the ζ^* cannot be learned from the sampled dataset due to the imbalanced marginal distribution $P(Y)$. Let's consider a simple hard max-margin classifier $f(x) : \text{sign}(x - b)$ that maximises the margin between the nearest points of opposite classes, which can be derived by solving:

$$\begin{aligned} \text{argmax}_{\zeta} \quad & \gamma \\ \text{s.t.} \quad & |x_i - \zeta| \geq \gamma, \quad \forall (x_i, y_i) \in D. \\ & -y_i(x_i - \zeta) \geq 0, \quad \forall (x_i, y_i) \in D. \end{aligned} \tag{1}$$

The decision boundary $\hat{\zeta}$ of the max-margin classifier learned on the toy imbalanced dataset is represented by the dotted black line in Fig. 3. It can be seen that the learned $\hat{\zeta}$ is skewed towards the minority class compared with the optimal boundary ζ^*. This is due to the lack of support vector samples that are closer to the decision boundary in the minority class. We can see that samples close to the decision boundary are rare patterns in the underlying distribution $P(X|Y)$ for both classes. However, since the class distribution $P(Y)$ is skewed, the majority class is more likely to be well represented by the data and contains rare samples, which serve as stronger support vectors in the learning process (e.g., the leftmost majority sample in Fig. 3).

Formally, for class c, let's use d_c^{max} to denote the distance between the class distribution center μ_c and the support vector (the point farthest from μ_c in the sampled dataset), i.e.,

$$d_c^{max} = \max(|x - \mu_c|), \forall (x, y) \in D_c, x \sim \mathcal{N}(\mu_c, 1). \tag{2}$$

Then the support vector of minority class can be written as $s_+^{\text{sup}} : (x_+^{\text{sup}}, c_+)$ (e.g., the rightmost c_+ instance in Fig. 3), where $x_+^{\text{sup}} = \mu_+ + d_{c_+}^{max}$. Similarly, we have the majority support vector $s_-^{\text{sup}} : (x_-^{\text{sup}}, c_-), x_-^{\text{sup}} = \mu_- - d_{c_-}^{max}$. Then the max-margin separator $\hat{\zeta} = (x_+^{\text{sup}} + x_-^{\text{sup}})/2$. Therefore, the expectation of decision bias is:

$$\begin{aligned} \mathbb{E}_{P(XY)}[|\zeta^* - \hat{\zeta}|] &= \mathbb{E}_{P(XY)}|[(x_+^{\text{sup}} + x_-^{\text{sup}})/2 - (\mu_+ + \mu_-)/2]| \\ &= |[\mathbb{E}_{P(XY)}(x_+^{\text{sup}} - \mu_+) + \mathbb{E}_{P(XY)}(x_-^{\text{sup}} - \mu_-)]/2| \\ &= \left| \frac{\mathbb{E}_{P(X|Y=c_+)}[d_{c_+}^{max}] - \mathbb{E}_{P(X|Y=c_-)}[d_{c_-}^{max}]}{2} \right| \end{aligned} \tag{3}$$

As the class distribution $P(Y)$ is skewed, we can expect the expectation of $\mathbb{E}_D[d_{c_-}^{max}]$ is larger than that of $\mathbb{E}_D[d_{c_+}^{max}]$, i.e.,

$$\mathbb{E}_D[d_{c_-}^{max}] > \mathbb{E}_D[d_{c_+}^{max}], \text{ if } |D_{c_-}| > |D_{c_+}|. \tag{4}$$

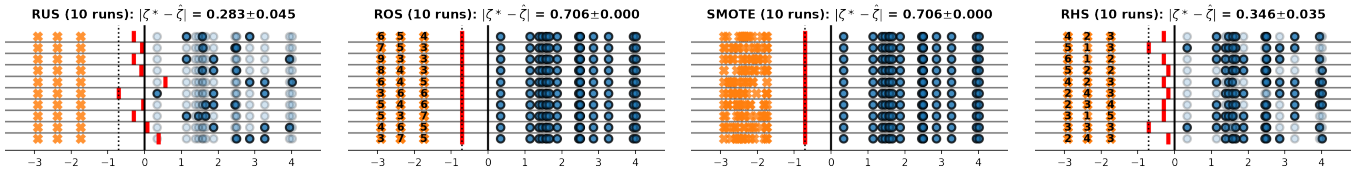
Note that $\mathbb{E}_D[d^{max}]$ has no closed-form expression, but its lower and upper bound are linearly related to $\sqrt{\log n}$ [45]. Putting Eq. (3) and Eq. (4) together, we have $|D_{c_-}| > |D_{c_+}| \Rightarrow \mathbb{E}_D[d_{c_-}^{max}] > \mathbb{E}_D[d_{c_+}^{max}] \Rightarrow \mathbb{E}_D[|\zeta^* - \hat{\zeta}|] > 0$, which shows how class-imbalance can induce bias into the classifier learning.

We then apply the above IntraCB strategies to this example dataset to intuitively demonstrate their effectiveness in mitigating the decision bias $|\zeta^* - \hat{\zeta}|$ induced by class-imbalance. The results are shown in Fig. 4. Note that to minimize the effect of randomness, for each strategy, we show the results of 10 independent runs, including the resampling results, the new decision boundaries $\hat{\zeta}$ learned on the resampled datasets (red lines), and the corresponding bias (means±variance).

Under-sampling. As shown in Fig. 4(a), random under-sampling (RUS) is very effective in terms of reducing the bias towards the minority class. On average, it greatly reduces the decision bias from 0.706 to 0.283. The reason behind is that: *randomly discarding instances from the majority class c_- until $|D_{c_-}| = |D_{c_+}|$* is equivalent to *sampling only $|D_{c_+}|$ examples from the majority underlying distribution*. Therefore, RUS can be seen as an unbiased correction to the skewed marginal distribution $P(Y)$. By reducing the $|D_{c_-}|$ to $|D'_{c_-}| = |D_{c_+}|$, RUS equalizes the $\mathbb{E}_{D'}[d_{c_-}^{max}]$ and $\mathbb{E}_{D'}[d_{c_+}^{max}]$ and thus lowers the expected bias $\mathbb{E}_{D'}[|\zeta^* - \hat{\zeta}|]$ on the resampled dataset D'.

On the downside, however, we note that the decision bias varies considerably over multiple runs (with the highest variance 0.045 among all competitors), indicating that the performance of RUS is not stable. This is caused by an obvious reason: randomly discarding most of the majority examples can introduce significant information loss [3], [25], especially when the dataset is highly imbalanced. As can be seen in Fig. 4(a), while RUS can reduce the bias towards c_+, it is sometimes too aggressive and even brings opposite bias towards c_-. Intuitively, RUS is also contradict to the common practice in data mining, which is to collect more data and exploit all available information.

Over-sampling and SMOTE. Fig. 4(b) shows the results of random over-sampling (ROS). We can observe that, despite ROS also equalizes the $|D'_{c_-}|$ and $|D'_{c_+}|$ in the resampled dataset D' like RUS, it does not help in terms of mitigating the decision bias, i.e., $|\zeta^* - \hat{\zeta}_D| = |\zeta^* - \hat{\zeta}_{D'}|$. The key difference between ROS and RUS is the *equivalence of "resampling" and "sampling from the original distribution" for the target class*.



(a) Random Under-sampling. (b) Random Over-sampling. (c) SMOTE Over-sampling [5]. (d) Random Hybrid-sampling.

Fig. 4. A comparison of different inter-class balancing strategies. Each subplot shows the resampling results of the corresponding method in 10 independent runs. The optimal boundary ζ^* (solid line), original learned $\hat{\zeta}_D$ (dotted line), and new boundaries $\hat{\zeta}_{D'}$ (red lines) learned from resampled D' are also included.

As stated above, the majority class set after RUS is equivalent to sampling from the original distribution $P(X|Y = c_-)$, except that it contains less samples. However, this is not the case for ROS. ROS performs over-sampling by replicating existing minority class instances (indicated by the numbers on minority samples in Fig. 4(b)). It means that the new synthetic instances are sampled from a uniform distribution over existing minority examples D_{c_+}, rather than from the true underlying distribution $P(X|Y = c_+)$ (unknown in practice). Consequently, ROS does not change the $\mathbb{E}_D[d^{max}_{c_+}]$ and the expected bias, albeit it enlarges the minority class size $|D_{c_+}|$, i.e., $\mathbb{E}_D[|\zeta^* - \hat{\zeta}|] = \mathbb{E}_{D'}[|\zeta^* - \hat{\zeta}|]$. Moreover, due to the duplication-based design, ROS can also cause the classifier to overfit the pattern of minority class samples [1], [3].

The Synthetic Minority Over-sampling TechniquE [5] (SMOTE) is one of the most popular methods to prevent the risk of overfitting induced by ROS. It improves the naïve ROS by finding the k-nearest neighbors (kNNs) for all minority examples and performing linear interpolation between a seed example and one of its kNNs. SMOTE has been proved to be effective in preventing overfitting and improving the quality of minority class representations in many real-world applications [2]. However, from Fig. 4(c) we can observe that although the distribution of the over-sampled minority class is smoothed by SMOTE, the $\mathbb{E}_D[d^{max}_{c_+}]$ still remains unchanged, so does the expected bias $\mathbb{E}_D[|\zeta^* - \hat{\zeta}|]$. Therefore, SMOTE only partially solves the problems that exist in ROS.

Hybrid-sampling. Finally, we consider a simple combination of RUS and ROS called random hybrid-sampling (RHS). The results are shown in Fig. 4(d). We can see that RHS is also an effective way to mitigate the decision bias (51% reduction, 0.706→0.346) as it incorporates RUS. Compared with the strict under-sampling $|D_{c_-}| \xrightarrow{\text{RUS}} |D_{c_+}|$, the average-targeted under-sampling in RHS is less aggressive, thus preventing the introduction of the opposite bias towards c_- as in RUS. As a result, RHS yields more stable decision boundaries over multiple runs, where the variance (0.035) of RHS is lower than that of RUS (0.045). In terms of over-sampling, RHS adopts the naïve ROS, but produces fewer new samples to mitigate potential overfitting problems. The reasons for not using more advanced over-sampling techniques (e.g., SMOTE [5] and its variants [6], [7]) are: **(1)** as stated above, these techniques do not help to reduce decision bias, **(2)** they prerequisite a well-defined distance metric and introduce additional computational cost, and **(3)** with proper processing (i.e., the perturbation-based data augmentation described in § IV-B), duplication-based ROS has the potential to increase $\mathbb{E}_D[d^{max}_{c_+}]$ and thus reduces bias.

From the previous discussion, we can conclude that *the key to mitigating decision bias is the ability to equalize* $\mathbb{E}[d^{max}_{c_+}]$ *and* $\mathbb{E}[d^{max}_{c_-}]$. This can be done by scaling down $\mathbb{E}[d^{max}_{c_-}]$ and/or scaling up $\mathbb{E}[d^{max}_{c_+}]$. The former can be achieved by RUS, but with significant information loss and variance. The latter is a difficult problem as the underlying minority distribution is unknown in practice, existing methods like ROS and SMOTE fundamentally generate new examples within the existing ones, thus cannot change $\mathbb{E}[d^{max}_{c_+}]$. RHS combines RUS and ROS and achieves more robust adjustification of $\mathbb{E}[d^{max}_{c_-}]$, but new techniques are still needed to increase $\mathbb{E}[d^{max}_{c_+}]$.

B. Perturbation-based Data Augmentation (PBDA).

Motivated by the above analysis, we propose a simple way to adjust $\mathbb{E}[d^{max}_{c_+}]$, namely perturbation-based data augmentation (PBDA). In this case, we add random Gaussian noises x^p to the resampled dataset D' and get augmented data D'_{aug}. Formally,

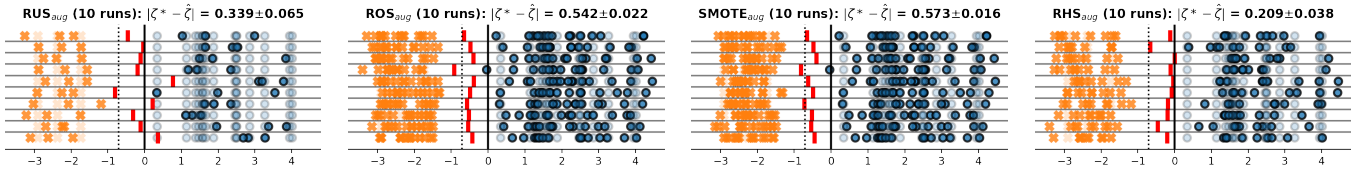
$$D'_{\text{aug}} : \{(x + x^p, y)\}, x^p \sim \mathcal{N}(0, \sigma^p), (x, y) \in D', \quad (5)$$

where σ^p is a hyper-parameter that controls the intensity of perturbation. But why does it help to increase $\mathbb{E}[d^{max}_{c_+}]$?

Why PBDA works? Recall that we define the support vector of minority class $s^{\text{sup}}_+ : (x^{\text{sup}}_+, c_+)$ and majority class $s^{\text{sup}}_- : (x^{\text{sup}}_-, c_-)$. Without loss of generality, we can always assume that the optimal decision boundary is at the origin of the coordinate axis, i.e., $\zeta^* = 0$ and $\mu_- + \mu_+ = 0$, so the bias $|\zeta^* - \hat{\zeta}| = |(x^{\text{sup}}_+ + x^{\text{sup}}_-)/2|, x^{\text{sup}}_+ < 0$. Therefore, in order to reduce the bias, we want to increase x^{sup}_+, which cannot be done by directly using the over/hybrid-sampling methods discussed before. However, note that after duplication-based over-sampling, there will be multiple replications of s^{sup}_+. Suppose the number of replications is n^{rep}, after applying PBDA, these replications can be seen as n^{rep} i.i.d. samples from $\mathcal{N}(x^{\text{sup}}_+, \sigma^p)$. Therefore, the new support vector on the augmented data $x^{\text{newsup}}_+ = max(\text{augmented replications of } x^{\text{sup}}_+)$. As proved in [45], $\mathbb{E}[x^{\text{newsup}}_+ - x^{\text{sup}}_+]$ ($\mathbb{E}[\Delta x^{\text{sup}}_+]$) is bounded by:

$$0 \leq \frac{1}{\sqrt{\pi \log 2}} \sigma^p \sqrt{\log n^{\text{rep}}} \leq \mathbb{E}[\Delta x^{\text{sup}}_+] \leq \sqrt{2} \sigma^p \sqrt{\log n^{\text{rep}}}. \quad (6)$$

This demonstrates that *duplication-based over-sampling with PBDA is effective in increasing* x^{sup}_+*, and thus yields a less biased decision boundary*. Note that the expected bias reduction



(a) Random Under-sampling. (b) Random Over-sampling. (c) SMOTE Over-sampling [5]. (d) Random Hybrid-sampling.

Fig. 5. A comparison of different inter-class balancing strategies *with data augmentation*. Each subplot shows the resampling results of the corresponding method in 10 independent runs. Same as in Fig. 4, optimal ζ^* (solid line), original $\hat{\zeta}_D$ (dotted line) and new $\hat{\zeta}_{D'_{\text{aug}}}$s (red lines) are also included.

is bounded by terms related to the hyper-parameter σ^p and the number of minority support vector replications n^{rep}.

We carry out further experiments to validate the above analysis, and the results are shown in Fig. 5. Specifically, we extend the experiments in Fig. 4 by applying PBDA on the resampled data, the σ^p is set to 0.2. First, it can be observed from Fig. 5(a) that PBDA is not helpful when used in conjunction with RUS. It is not a surprising result as $n^{\text{rep}} = 1$ for RUS, substituting into Eq. (6), we have $\mathbb{E}[\Delta x_+^{\text{sup}}] \leq \sqrt{2}\sigma^p\sqrt{\log 1} = 0$, i.e., the PBDA does not reduce the expected bias when $n^{\text{rep}} = 1$. In addition, the performance of RUS becomes even more unstable (variance increases from 0.045 to 0.065) due to the additional uncertainty introduced by PBDA. As for ROS with $n^{\text{rep}} > 1$, PBDA significantly reduces its decision bias by 0.164 (0.706→0.542, Fig. 5(b)). Similarly, the bias of SMOTE is also reduced but by a smaller margin (0.133) due to the smaller n^{rep} compared with ROS (Fig. 5(c)). Finally, we find that RHS is the best performer cooperated with PBDA (Fig. 5(d)). It yields the smallest bias (0.209 ± 0.038) since it works in both directions: with RUS to adjust $\mathbb{E}[d_{c_-}^{max}]$, and with ROS+PBDA to adjust $\mathbb{E}[d_{c_+}^{max}]$.

We note that aside of the classification errors of each base classifier, the diversity across the ensemble members is also a key factor that affects the performance of ensemble models [46], [47]. Therefore, beyond mitigating decision bias, PBDA also plays an important role in diversifying the base classifiers of DUBE. It also serves as a technique to prevent the risk of overfitting introduced by duplication-based over-sampling.

Formalization. In practice, the conditional underlying distribution $P(X|Y)$ is likely to vary in different classes, thus sample perturbation signals x^p from a single Gaussian distribution $\mathcal{N}(0, \sigma^p)$ is insufficient. In response to this, we add different perturbations for each class independently in DUBE, the intensity is controlled by a single hyper-parameter α:

$$D'_{\text{aug}} : \{(x + \alpha \cdot x^p, c_i)\}, x^p \sim \mathcal{N}(0, \Sigma_{c_i}), \forall c_i \in \mathcal{Y}, \quad (7)$$

where the Σ_{c_i} is the covariance matrix estimated from D_{c_i}:

$$\Sigma_{c_i} = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_j - \mu_i)(x_j - \mu_i)^\top. \quad (8)$$

In Eq. (8), $n_i = |D_{c_i}|$ and μ_i is the centroid of the class c_i, i.e., $\mu_i = \sum_{j=1}^{n_i}(x_j)/n_i, x_j \in D_{c_i}$. By doing so, we calibrate the perturbation using statistics of the base classes.

C. Intra-class Balancing (IntraCB).

We now discuss the influence of different IntraCB strategies in DUBE . Intra-class balancing in DUBE is achieved by assigning different sampling probability w to each instance (x, y) based on the prediction error w.r.t. a classifier f, i.e., an IntraCB strategy corresponds to a weighting function $g(\cdot) : \mathcal{X} \times \mathcal{Y} \times \mathcal{F} \to \mathbb{R}$. Here the $\mathcal{F}$ is the hypothesis (model) space.

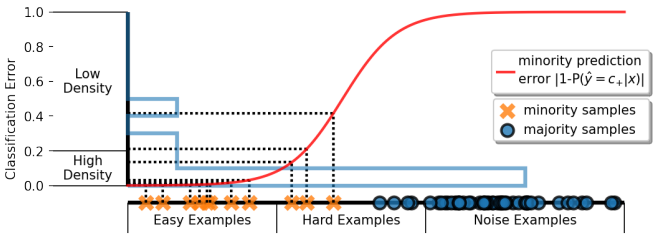
Hard example mining. Let's first introduce the hard example mining (HEM) considered in DUBE . Formally, for an input x with label space $\mathcal{Y} : \{c_1, c_2, \cdots, c_m\}$, given a classifier $f(\cdot)$, the estimated probabilities are $f(x) = \mathbf{p} = [p_1, p_2, ..., p_m]^\top$, where the i-th component $p_i \in [0, 1]$ is the estimated $P(y = c_i|x)$. For a label $y = c_i$, we define the one-hot binarized label vector $\mathbf{y} = [y_1, y_2, ..., y_m]^\top$, where $y_i = 1$ and other components are 0. Then the prediction error is defined as $\sum_{i=0}^{m} |p_i - y_i|$. For simplicity, we use $\mathbf{err}(f(x), y)$ to denote the error of (x, y) w.r.t. f. Then in the t-th iteration of DUBE, given the current ensemble classifier $F_{t-1}(\cdot)$, hard example mining corresponds to the following weighting function:

$$g_{\text{HEM}}(x, y, F_{t-1}) = \mathbf{err}(F_{t-1}(x), y), \quad (9)$$

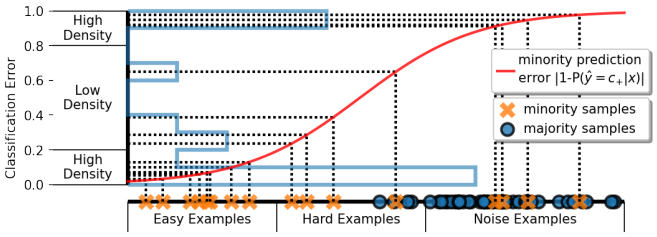
i.e., hard examples with larger prediction errors are more likely to be sampled and used in the following learning process.

When the dataset contains no noise (i.e., has better separability), we show that with duplication-based over-sampling and PBDA, HEM is effective in term of further reducing decision bias. As shown in Fig. 6(a), the minority support vector x_+^{sup} has the largest error among all minority class samples as it locates nearest to the decision boundary. Suppose that we want to expand the minority class size from n_+ to n'_+, where $n'_+ > n_+$. With ROS, the expected number of x_+^{sup} replications is $\mathbb{E}_{\text{ROS}}[n^{\text{rep}}] = n'_+/n_+$. With ROS+HEM, this expectation becomes $\mathbb{E}_{\text{ROS+HEM}}[n^{\text{rep}}] = (e_+^{\text{sup}} \cdot n'_+)/(\sum_{s_+ \in D_{c_+}} e_+)$, where e represents the error of a sample s. It is obvious that $\mathbb{E}_{\text{ROS+HEM}}[n^{\text{rep}}] > \mathbb{E}_{\text{ROS}}[n^{\text{rep}}]$ since $e_+^{\text{sup}} \geq e_+, \forall s_+ \in D_{c_+}$. Then according to Eq. (6), we can expect $\mathbb{E}_{\text{ROS+HEM}}[x_+^{\text{sup}}] > \mathbb{E}_{\text{ROS}}[x_+^{\text{sup}}]$ after PBDA, and therefore yields smaller bias.

Nevertheless, HEM can wrongly reinforce noise/outliers when working on a noisy dataset. In Fig. 6(b), we modify the data in Fig. 6(a) by adding minority class noise samples. It can be observed that these few noise samples contribute most of the prediction errors of the minority class. Consequently, these outliers will dominate the resampling process with HEM. In this case, the outliers will be duplicated multiple times, while



(a) Without data noise.



(b) With data noise.

Fig. 6. The classification error (y-axis) distributions with/without presence of minority class noise. The prediction errors (probabilities) are obtained from a Logistic Regression classifier trained on the corresponding data.

other samples including the true support vector x_+^{sup} will be ignored by HEM. This is clearly not a good sampling strategy: *the extracted data will contain mainly noisy instances, which will greatly interfere with the subsequent learning process*.

Soft hard example mining. However, we find that this can be prevented by considering the error density distribution. After taking a closer look at the distribution of classification errors (y-axis) in Fig. 6, we have some interesting findings: **(1)** hard examples are likely to be sparsely distributed in the region near the decision boundary (e.g., the "hard examples" area on the x-axis), where the estimated probability $P(\hat{Y}|X)$ also changes drastically. All these lead to an even more sparse error distribution of hard examples (e.g., the "low density" area on the y-axis); **(2)** By comparison, both easy examples and noise examples are densely distributed in terms of classification error (e.g., the "high density" areas on the y-axis). This is mainly because $P(\hat{Y}|X)$ varies only a little in both the "easy examples" and "noisy examples" regions as the classifier has a high confidence in the predictions of these samples.

Motivated by the above analysis, we propose to perform robust HEM by simply *inversing the classification error density*. By doing so, the easy and noisy examples with high error densities will be down-weighted. Intuitively, this will prevent over-weighting of noisy examples while still emphasizing the importance of mining hard examples. There are many ways to estimate the error distribution E, and herein we use a histogram to approximate E for its simplicity and efficiency. Formally, consider a histogram with b bins, then the approximated error distribution is given by a vector $\widehat{E}_D = [\widehat{E}_D^1, \widehat{E}_D^2, ..., \widehat{E}_D^b] \in \mathbb{R}^b$, where the i-th component $\widehat{E}_D^i$ denotes the proportion of the number of samples in the i-th bin (B^i) to the total, i.e.,

$$\widehat{E}_D^i = \frac{|B^i|}{|D|}; B^i = \{s_j | \frac{i-1}{b} \leq e_j < \frac{i}{b}, \forall s_j \in D\}. \quad (10)$$

Algorithm 1 `Duple-Balanced Ensemble Training`

Require: training set D, ensemble size k, perturbation hyper-parameter α, number of bins in the error histogram b

Derive covariance matrices $\Sigma_{c_1}, \Sigma_{c_2}, ..., \Sigma_{c_m}$. ▷ Eq. (8)
Train the first base classifier $f_1(\cdot)$ on D
for t=2 to k **do**
  Update current ensemble $F_{t-1}(x) = \frac{1}{t-1}\sum_{i=1}^{t-1} f_i(x)$.
  # Inter-class Balancing (RUS/ROS/RHS) ▷ Section IV-A
  Set the target class size n for resampling.
  # Intra-class Balancing (HEM/SHEM) ▷ Section IV-C
  Set the instance-wise weight w w.r.t. $F_{t-1}(\cdot)$.
  # Duple-Balanced Resampling
  Initialization: $D' \Leftarrow \emptyset$
  for i=1 to m **do**
    $D'_{c_i} \Leftarrow n$ instances sampled from D_{c_i} w.r.t. w.
    # Data Augmentation ▷ Section IV-B
    $D'_{c_i,\text{aug}} \Leftarrow \{(x + \alpha \cdot x^p, c_i)\}_{x^p \sim \mathcal{N}(0, \Sigma_{c_i})}$ ▷ Eq. (7)
    $D' \Leftarrow D' \cup D'_{c_i,\text{aug}}$
  Train a new base classifier $f_t(\cdot)$ with D'.
return Ensemble classifer $F_k(\cdot)$

Here the e_j is the error of s_j, i.e., $e_j = \textbf{err}(F_{t-1}(x_j), y_j)$. With Eq. (10), we can now formally define the Soft Hard Example Mining (SHEM) in DuBE:

$$g_{\text{SHEM}}(x, y, F_{t-1}) = 1/\widehat{E}_D^{\lceil \textbf{err}(F_{t-1}(x), y)/b \rceil}. \quad (11)$$

The $\lceil \textbf{err}(F_{t-1}(x), y)/b \rceil$ in Eq. (11) indicates the bin that (x, y) belongs to, i.e., instances are assigned sampling weights corresponding to the inverse of their classification error density. To this point, we summarize the DuBE framework in Alg. 1. Please refer to § V-B3 for ablation studies and related discussions about InterCB, IntraCB, and PBDA.

V. Experiments

To thoroughly evaluate the effectiveness of DuBE , two series of experiments are conducted. We first generate several synthetic datasets to visualize the difference between the resampling behavior of canonical methods and DuBE . After that, we extend the experiment to various real-world IL tasks to validate DuBE's performance in practical applications.

A. Experiment on Synthetic Datasets

Setup details. To intuitively demonstrate how different IL methods work in IL problems, we first show some visualizations on a series of synthetic datasets. We create three 2-dimensional classification tasks with 1,100 samples from 2 classes, the class distribution is `1000/100`, i.e., imbalance ratio (IR) = 10. Note that there exists class overlapping in the synthetic dataset, as shown in Figure 7. In this experiment, DuBE is compared with 8 representative resampling methods, 4 of which are under-sampling (i.e., RUS, TomekLink [9], NearMiss [8], Condense [48]) and the other 4 are over-sampling (i.e., ROS, the widely used Smote [5], and its variants Adasyn [6] and BorderSmote [7]). All methods

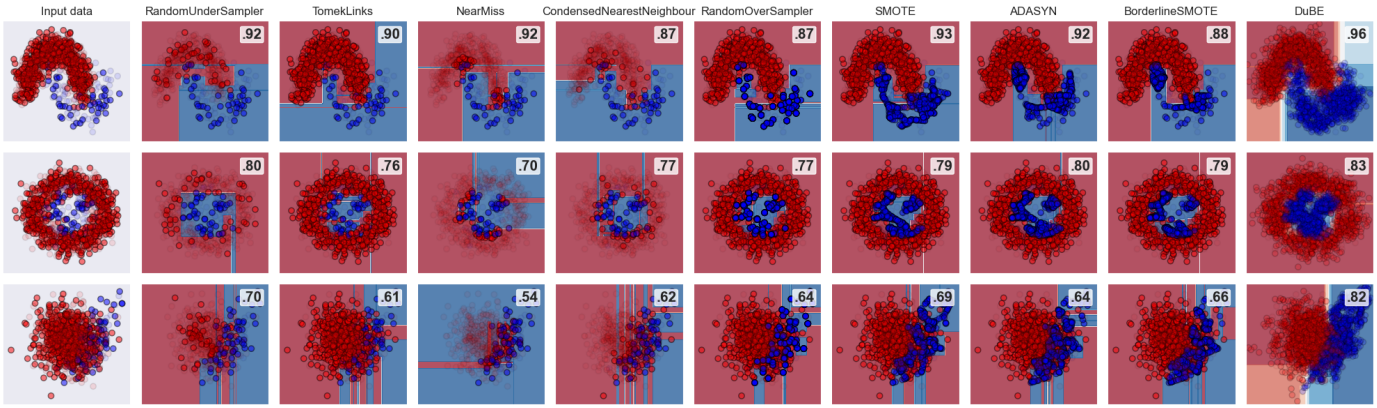


Fig. 7. Comparisons of DUBE with 8 representative resampling methods (under-sampling: RUS, TOMEKLINK [9], NEARMISS [8], CONDENSE [48], over-sampling: ROS, SMOTE [5], ADASYN [6], and BORDERSMOTE [7]) on 3 synthetic imbalanced datasets with different level of underlying class distribution overlapping (less/mid/highly overlapped in 1st/2nd/3rd row). The opaque dots represent the (resampled) training set and the translucent dots represent the test set. The number in the upper right corner of each subfigure represents the test macro AUROC score of the corresponding classifier. Best viewed in color.

included in this experiment are deployed with decision trees as base classifiers. The ensemble size of DUBE is 5. We perform 50%/50% train/test split on each dataset, and use translucent dots to represents the test set in Fig. 7. For each IL algorithm, we plot the resampled training dataset, the learned decision boundary, and the macro-averaged AUROC (a class-balanced metric) score on the test set in Fig. 7.

Visualization & analysis. As mentioned earlier, RUS randomly removes the majority class sample, which helps mitigate decision bias but leads to severe information loss. This may cause the classifier to overfit the selected subset and generate unstable learning boundary. TOMEKLINK [9] performs under-sampling by detecting "TomekLinks", which exists if two samples of different class are the nearest neighbors of each other, i.e., it removes the majority samples that locate in the overlapping area. But we can see that only a few majority samples are discarded, which does not help with mitigating the decision bias. It is therefore outperformed by RUS, especially on highly-overlapped datasets (e.g., 3rd row). In contrast to TOMEKLINK, both NEARMISS [8] and CONDENSE [48] aim to select the majority samples that are closest to the minority ones. They assume these samples are likely to be support vectors that help classification. However, this assumption clearly does not hold when the data sets overlap: both methods end up dropping most of the majority samples that reflect the original distribution, which greatly interferes with the learning process.

Compared with under-sampling, over-sampling methods typically produce more stable decision bounds as they only add new samples and keep the original dataset untouched. ROS works by simply replicating existing minority instances, which may cause the learner over-fit the pattern of minority class samples, as shown in the [1st row, RandomOverSampler] sub-figure. Advanced over-sampling techniques prevent this by performing neighborhood-based interpolation instead of duplication. SMOTE [5] works by repeatedly selecting seed instances in the minority class, and then generating synthetic samples on the connection line between the seed and one of its nearest neighbors. This effectively smooths out the minority distribution after over-sampling. ADASYN [6] and BORDERSMOTE [7] further improve SMOTE by focusing on borderline minority examples for interpolation. However, such strategy may result in generating massive instances near minority outliers (e.g., data points at the upper left end of the "$\cup$"-shaped distribution of the 1st dataset).

Finally, we can observe that DUBE ($k = 5$, with RHS, SHEM and PBDA) achieves the best performance in all three tasks. Compared with strict under-sampling, the hybrid-sampling essentially maintains the original distribution structure of the data. On the other hand, compared with ADASYN and BORDERSMOTE, the usage of SHEM prevents DUBE from generating synthetic instances near minority outliers. Moreover, PBDA improves the coverage of minority class in the feature space, thus alleviating decision bias. It also enables data-level regularization and stabilizes the decision boundary of DUBE.

B. Experiment on Real-world Datasets

Datasets. To verify the effectiveness of DUBE in practical applications, we extend the experiments to real-world IL tasks from the UCI repository [49]. These data are collected from different application domains including bioinformatics, sociodemographics, clinical medicine, etc. To ensure a thorough assessment, these datasets vary widely in their properties. Please refer to TABLE II for detailed statistics of these datasets.

Evaluation protocol. For IL problems, classification accuracy is not a fair performance metric as it cannot reflect how the classifier works on minority classes. Hence, unbiased evaluation criterias based on the number of true/false positive/negative prediction are usually used in IL. For a comprehensive evaluation, we consider 3 unbiased metrics: macro-averaged F1-score, MCC (matthews correlation coefficient), and AUROC (area under the receiver operating characteristic curve) [50]. For each dataset, we report the result of 5-fold stratified cross-validation and execute 5 independent runs with different random seeds to eliminate the randomness.

TABLE II
CHARACTERISTICS OF THE REAL-WORLD CLASS-IMBALANCED DATASETS.

Dataset	#Samples	#Features	#Classes	Class Distribution	Imbalance Ratio	Task	Domain
ecoli	336	7	2	301/35	8.60/1.00	target: imU	Bioinformatics.
pen-digits	10,992	16	2	9,937/1,055	9.42/1.00	target: 5	Computer.
spectrometer	531	93	2	486/45	10.80/1.00	target: >=44	Astronomy.
scene	2,407	284	2	2,230/177	12.60/1.00	target: >one label	Geography.
libras-move	360	90	2	336/24	14.00/1.00	target: 1	Physics.
oil	937	49	2	896/41	21.85/1.00	target: raw	Environment.
letter-img	20,000	16	2	19,266/734	26.25/1.00	target: Z	Computer.
ozone-level	2,536	72	2	2,463/73	33.74/1.00	target: raw	Climatology.
balance-scale	625	4	3	288/288/49	5.88/5.88/1.00	target: raw	Psychology.
cmc	1,473	24	3	629/511/333	1.89/1.53/1.00	target: raw	Sociology.

TABLE III
COMPARISONS OF DUBE WITH REPRESENTATIVE RESAMPLING-BASED IL SOLUTIONS. THE MACRO-AUROC (MEAN±STD), THE NUMBER OF TRAINING SAMPLES, AND THE TIME USED FOR RESAMPLING ARE REPORTED. THE BEST RESULTS (AMONG BASELINES) ARE MARKED IN **BOLD** (UNDERLINED).

Category	Method	Base Learning Algorithm					#Training Samples	Resampling Time (ms)
		MLP	KNN	DT	BST	BAG		
No-resampling	-	0.499±0.004	0.628±0.006	0.669±0.012	0.694±0.013	0.684±0.017	2029	-
Under-sampling	RUS	0.501±0.005	0.660±0.006	0.674±0.032	0.702±0.030	0.683±0.030	116	0.72
	TOMEKLINK [9]	0.509±0.012	0.506±0.000	0.569±0.006	0.566±0.006	0.541±0.020	2,008	62.28
	NEARMISS [8]	0.505±0.009	0.444±0.000	0.472±0.011	0.489±0.013	0.487±0.011	116	4.65
	CONDENSE [48]	0.501±0.002	0.517±0.000	0.655±0.008	0.660±0.015	0.655±0.008	311	12157.64
Over-sampling	ROS	0.504±0.003	0.571±0.000	0.610±0.010	0.609±0.020	0.595±0.007	3,942	1.64
	SMOTE [5]	0.498±0.004	0.637±0.013	0.680±0.013	0.629±0.002	0.629±0.028	3,942	3.99
	ADASYN [6]	0.515±0.030	0.630±0.003	0.618±0.025	0.639±0.018	0.636±0.017	3,942	6.01
	BORDERSMOTE [7]	0.511±0.017	0.592±0.004	0.592±0.009	0.579±0.016	0.599±0.038	3,942	6.20
Over-sampling + Cleaning	SMOTEENN [24]	0.514±0.025	0.670±0.009	0.628±0.020	0.619±0.023	0.693±0.017	3,500	270.76
	SMOTETOMEK [27]	0.518±0.028	0.618±0.009	0.625±0.017	0.607±0.016	0.641±0.021	3,939	257.40
Ours	$\text{DUBE}_{10,\text{RUS}}$ a	0.610±0.076	0.723±0.005	0.794±0.014	**0.800±0.005**	0.804±0.006	116×10	1.39
	$\text{DUBE}_{10,\text{ROS}}$	0.534±0.022	0.697±0.025	0.797±0.026	0.792±0.026	**0.812±0.005**	3,942×10	1.69
	$\text{DUBE}_{10,\text{RHS}}$	**0.659±0.018**	**0.724±0.009**	**0.801±0.009**	0.798±0.016	0.802±0.007	2,029×10	1.52

1) Comparison with Resampling IL Methods: We first compare DUBE with resampling-based IL solutions. They have been widely used in practice for the preprocessing of class-imbalanced data [2]. Ten representative methods are selected from 4 major branches of resampling-based IL: under-&over-sampling and over-sampling with cleaning post-process (also referred as hybrid-sampling in some literature, we do not use this name to prevent confusion). All methods are tested on the *ozone-level* dataset, which has the highest imbalance ratio (IR=33.75), to test their efficiency and effectiveness. The ensemble size of DUBE is set to 10, with SHEM and PBDA. Five different classifiers, i.e., Multi-layer Perceptron (MLP), K-nearest neighbor (KNN), decision tree (DT), adaptive boosting (BST), and Bagging (BAG), are used to collaborate with these approaches. The number of training samples and the time used to perform resampling are also reported.

TABLE III details the experiment results. We show that by explicitly performing inter-&intra-class balancing, DUBE outperforms canonical resampling methods by a significant margin. In such a highly imbalanced dataset, minority class is likely to be poorly represented and lacks a clear structure. Thus the advanced resampling methods that rely on distance-computing and neighborhood relations between minority objects may deteriorate the classification performance, especially when working with high-capacity models (e.g., BST and BAG). The over-sampling + cleaning methods generally perform better than other baselines as they combine under-sampling (US) and over-sampling (OS). But such combination also introduces more distance computational overhead and makes the resampling time considerably high. We also notice that distance-based US is usually more costly than OS as it involves calculating the distance between the majority and minority instances (e.g., CONDENSE resampling takes 12157.64 ms), while OS only considers the distance within the minority class. In contrast, DUBE's resampling does not involve any distance calculation and is therefore computationally efficient.

2) Comparison with Ensemble IL Methods: We further compare DUBE with 8 representative ensemble IL solutions on 10 real-world imbalanced classification tasks. Baselines include 4 under-sampling-based methods (RUSBST [32], UNDERBAG [51], CASCADE [22], and SPE [19]), and 4 over-sampling-based ones (OVERBST [46], SMOTEBST [31], OVERBAG [46], and SMOTEBAG [34]). "BST"/"BAG" indicates that the method is based on Adaptive Boosting/Bootstrap Aggregating (boosting/bagging) ensemble learning framework. For a fair comparison, the ensemble size of all test methods is set to 10. DUBE is implemented with RHS, SHEM and PBDA. We use C4.5 decision tree as the base learner for all ensembles following the setting of most of the previous works [2].

The results are reported in TABLE IV. We also report

TABLE IV

COMPARISONS OF DUBE WITH REPRESENTATIVE ENSEMBLE IL SOLUTIONS. FOR EACH DATASET, WE REPORT THE GENERALIZED F1-SCORE, MCC, AND MACRO-AUROC SCORES (MEAN±STD) IN THE 1ST/2ND/3RD ROW. THE BEST AND SECOND BEST RESULTS ARE MARKED IN **BOLD** AND UNDERLINED.

Task × Metric \ Method		RUSBST	OVERBST	SMOTEBST	UNDERBAG	OVERBAG	SMOTEBAG	CASCADE	SPE	DUBE	Δ_{mean}
ecoli ×	F1-score	.712±.013	.736±.024	.705±.020	.772±.017	.754±.014	.764±.012	.715±.015	.745±.023	**.778±.004**	5.48%
	MCC	.437±.026	.472±.047	.413±.039	.571±.026	.518±.023	.530±.026	.508±.023	.508±.046	**.578±.010**	17.92%
	AUROC	.756±.015	.735±.020	.716±.018	.851±.007	.727±.022	.758±.004	.858±.013	.804±.026	**.882±.008**	14.21%
pen-digits		.938±.009	.978±.002	.976±.002	.980±.001	.982±.001	.983±.001	.986±.001	.989±.001	**.996±.000**	1.98%
		.876±.017	.956±.003	.952±.003	.959±.002	.964±.002	.966±.002	.973±.002	.979±.001	**.991±.001**	4.09%
		.953±.004	.973±.001	.978±.001	.983±.001	.972±.001	.975±.001	.988±.001	.986±.002	**.995±.000**	1.98%
spectrometer		.800±.048	.783±.028	.770±.026	.762±.003	.786±.008	.801±.045	.901±.025	.880±.016	**.915±.013**	13.30%
		.605±.097	.568±.055	.541±.052	.553±.008	.604±.025	.613±.087	.804±.048	.762±.032	**.831±.026**	34.11%
		.835±.056	.767±.029	.787±.031	.850±.011	.729±.002	.762±.048	.919±.016	.907±.013	**.927±.015**	13.87%
scene		.535±.019	.576±.000	.554±.003	.576±.010	.549±.006	.573±.010	.537±.006	.546±.006	**.601±.011**	8.16%
		.114±.031	.161±.005	.112±.009	.200±.022	.170±.004	.196±.017	.166±.013	.199±.014	**.204±.024**	29.64%
		.585±.020	.564±.003	.565±.008	.649±.016	.537±.004	.553±.007	.640±.010	.670±.013	**.678±.014**	14.59%
libras-move		.727±.040	.847±.017	.827±.022	.791±.055	.827±.028	.801±.023	.791±.036	.820±.041	**.930±.014**	15.87%
		.470±.069	.697±.035	.654±.044	.600±.100	.684±.047	.643±.038	.610±.069	.649±.079	**.860±.027**	39.19%
		.751±.026	.820±.013	.822±.022	.862±.034	.762±.032	.735±.024	.886±.039	.869±.035	**.933±.021**	15.16%
oil		.578±.033	.611±.023	.683±.018	.597±.030	.640±.024	.702±.049	.576±.014	.607±.013	**.707±.006**	13.86%
		.224±.073	.221±.046	.370±.038	.304±.045	.368±.039	**.434±.089**	.286±.018	.319±.020	.420±.013	39.82%
		.694±.062	.615±.025	.705±.028	.784±.027	.592±.018	.655±.044	.785±.017	.793±.013	**.833±.012**	19.89%
letter-img		.826±.044	.942±.002	.940±.003	.879±.008	.949±.000	.960±.004	.943±.001	.972±.001	**.979±.003**	5.99%
		.667±.072	.885±.003	.879±.005	.776±.013	.901±.001	.922±.007	.889±.001	.945±.003	**.959±.005**	13.05%
		.896±.023	.932±.001	.947±.002	.969±.002	.917±.001	.938±.005	.978±.002	.973±.003	**.987±.002**	4.64%
ozone-level		.546±.014	.593±.004	.592±.003	.586±.010	.586±.020	.596±.035	.555±.005	.580±.006	**.630±.014**	8.79%
		.182±.011	.187±.007	.201±.003	.276±.021	.249±.052	.228±.077	.230±.019	.282±.013	**.290±.026**	29.46%
		.691±.010	.586±.006	.641±.002	.781±.019	.554±.012	.566±.025	.756±.029	**.803±.017**	.801±.009	20.66%
balance-scale		.599±.021	.587±.008	.578±.010	.610±.006	.586±.004	.586±.004	.611±.008	.624±.004	**.673±.009**	12.77%
		.587±.034	.649±.017	.611±.014	.572±.008	**.665±.013**	.653±.007	.569±.009	.584±.006	.653±.015	7.13%
		.814±.016	.722±.007	.698±.014	.821±.010	.781±.011	.790±.002	.827±.006	.840±.003	**.891±.015**	13.67%
cmc		.431±.005	.452±.006	.456±.008	**.489±.006**	.476±.004	.486±.009	.472±.010	.484±.009	.482±.003	3.05%
		.152±.006	.192±.009	.196±.014	.243±.008	.224±.005	.240±.012	.218±.016	.238±.014	**.268±.005**	28.69%
		.611±.003	.627±.005	.630±.002	.678±.004	.669±.001	.668±.006	.654±.005	.663±.006	**.686±.010**	5.64%

DUBE's average performance improvement relative to all baselines (Δ_{mean}) in percentage. It can be observed that DUBE achieves competitive performance in various real-world IL tasks. It outperforms all other 8 ensemble methods in 26 out of 10×3 task-metric pairs. On average, DUBE brings significant performance improvements over existing ensemble IL baselines (1.98%/15.22%/39.82% min/ave/max Δ_{mean}) by explicitly considering inter- and intra-class imbalance. We can also see that bagging-based approaches generally perform better than boosting-based ones (e.g., on all 30 task-metric pairs, the performance of SMOTEBAG is 6.54% better than SMOTEBST). Compared with boosing that only manipulates sample weights, the bootstrap sampling in bagging introduces additional data-level diversity, which helps prevent overfitting in IL, especially on minority class(es). Note that boosting is a way to implement HEM by reweighting. On the other hand, CASCADE and SPE also incorporate the idea of HEM, but by resampling. We find that the latter two resampling-based methods also usually outperform the boosting-based ones (mean(CASCADE, SPE) is 10.87% better than mean(RUSBST, OVERBST, SMOTEBST)), which further validates the importance of data-level diversity in IL. In DuBE, in addition to reducing decision bias, RHS and PBDA also play important roles in diversifying the base classifiers and preventing overfitting.

3) Ablation Study and Discussions: We carry out further experiment to validate the contribution of different inter-class (RUS/ROS/RHS) and intra-class balancing (HEM/SHEM) as well as the perturbation-based data augmentation (PBDA).

Intra-class balancing (IntraCB). As previously discussed in § IV-C, HEM helps alleviate intra-class imbalance by focusing on informative high-error instances, but is vulnerable to noise/outliers. SHEM is thus designed for robust HEM. In this experiment, we test their robustness against noise. In practice, the training data often contains corrupted labels due to errors in the labeling process like crowdsourcing. Here we simulate this by introducing flip noise, i.e., with noise ratio r, $r \cdot n_{\text{minority}}$ minority samples will be assigned opposite labels and vice versa. The largest *letter-img* dataset is used as a representative. We test $\text{DUBE}_{10,\text{RUS}}$ with HEM/SHEM and without IntraCB (uniform) on the 0%-50% corrupted data, as shown in Fig. 8. We can observe that when the data contains no noise (r=0), both HEM and SHEM achieve significantly better performance than uniform sampling. However, HEM's performance drops rapidly as r increases, and is even worse than uniform resampling when $r \geq 0.2$. In contrast, SHEM is more robust and consistently outperforms uniform resampling at different noise levels, which validates its effectiveness.

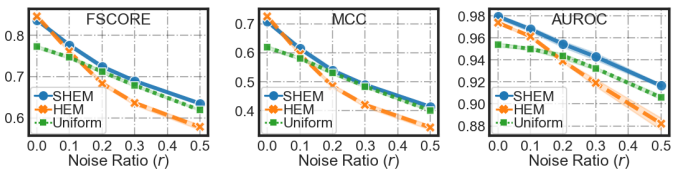


Fig. 8. Comparison of different IntraCB strategies under varying noise ratio.

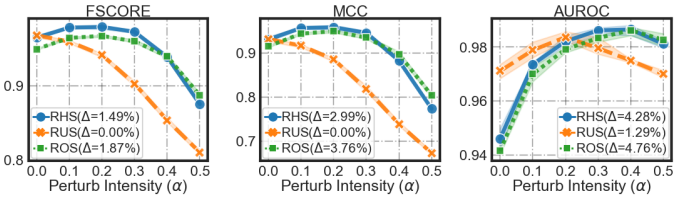


Fig. 9. Comparison of InterCB strategies under varying perturb intensity α, Δ is the relative performance gain from PBDA, i.e., $\frac{\max(\text{score}_{\alpha>0})-\text{score}_{\alpha=0}}{\text{score}_{\alpha=0}}$.

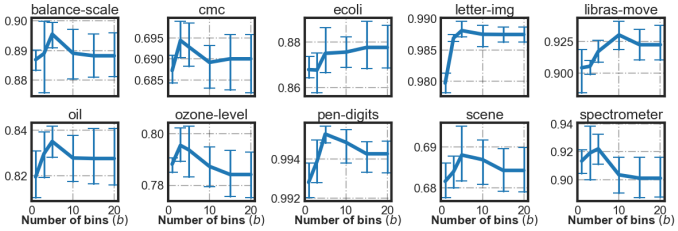


Fig. 10. The influence of the number of bins b (macro AUROC).

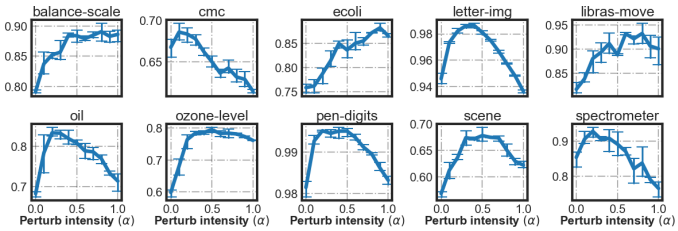


Fig. 11. The influence of the perturbation coefficient α (macro AUROC).

Inter-class balancing (InterCB) and PBDA. In § IV-A, we have discussed the difference between RUS, RHS and ROS. Specifically, we find that RUS is very effective in mitigating decision bias as it can be seen as an unbiased correction to the skewed marginal distribution $P(Y)$. The replication-based over-sampling in RHS and ROS on the other hand, is not equivalent to resampling from the underlying distribution $P(X|Y = c_{\text{min}})$ and thus does not help to mitigate bias. But recall that in § IV-B, we further introduce PBDA, which could be used to improve RHS and ROS by augmenting duplicated minority instances. To confirm the effectiveness of different InterCB strategies and PBDA, we test DUBE$_{10,\text{SHEM}}$ with RUS, RHS, ROS and different perturb intensity α. Results are shown in Fig. 9. It can be observed that in the absence of PBDA (α = 0), pure RUS outperforms RHS and ROS, especially in terms of AUROC scores, because of its ability to naturally mitigate bias. However, we also notice that RUS has significantly weaker performance gains from PBDA compared with RHS and ROS, as indicated by Δs in Fig. 9. With proper data augmentation, RHS&ROS achieve better performance. These findings are consistent with our discussions in § IV-B&IV-A. We also find that metrics have different responses to PBDA, e.g., for RHS, the optimal α = 0.2 for F1-score and MCC, but 0.4 for AUROC.

Implementation details. Our implementation of DUBE is based on Python 3.8.5. We use open-source software packages *imbalanced-learn* [52] and *imbalanced-ensemble* for implementation of all baseline resampling and ensemble IL methods. The base learning models (MLP, KNN, DT, etc.) are from the *scikit-learn* [53] package. The resampling times are obtained based on the results of running on an Intel Core™ i7-10700KF CPU with 32GB RAM.

Complexity analysis. The complexity of DUBE (Alg.1) mainly comes from the intra-class balancing (line#8), which requires the latest prediction probabilities estimated by the current ensemble. Suppose the complexity for a base classifier $f(\cdot)$ to predict on dataset D is $C^{pred}_{f,D}$. Then the total resampling cost for training a k-classifier DUBE is $k^{\text{pred}} \cdot C^{pred}_{f,D}$, where k^{pred} is the number of times of making predictions with $f(\cdot)$. Normally we have $k^{\text{pred}} = \Sigma_{t=1}^{k-1} t = \frac{k(k-1)}{2}$ according to Alg.1, i.e., resampling complexity is $O(k^2 C^{pred}_{f,D})$. But as DUBE is an additive ensemble, we can reduce k^{pred} to k by buffering the predicted probabilities (a $N \times m$ matrix) for each fitted $f(\cdot)$. Together with the training cost, the final complexity of DuBE is $O(k(C^{pred}_{f,D} + C^{train}_{f,D'}))$, note that the $C^{train}_{f,D'}$ term depends on the InterCB strategy used, since $|D'_{\text{RUS}}|<|D'_{\text{RHS}}|<|D'_{\text{ROS}}|$.

Parameter analysis. DUBE has two main parameters: the number of bins in the histogram b and the perturbation coefficient α. In this section, we discuss their influence based on real-world IL tasks. First, b determines how detailed the error distribution approximation is, we show its influence in Fig. 10. We can see that setting a small b may lead to a poor performance, e.g., b=1 degrades SHEM to uniform sampling. Using a large b (e.g., ≥ 10) does not necessarily lead to better performance either, as many bins may be empty. For these reasons, we recommend setting the b to be 5, which is also the setting we used in the experiments. One can try increasing b when working on large datasets. On the other hand, α determines the intensity of the perturbation-based augmentation. Fig. 11 shows it influence in real-world tasks. We can see that with proper α, PBDA significantly helps generalization, but keep increasing α will introduce too much perturbation and degrade the performance. In our implementation of DUBE, α could be automatically tuned using a small validation subset.

VI. Conclusion & Limitations

In this paper, we discuss two types of imbalance that existed in the nature of the IL, i.e., inter- and intra-class imbalance, and how they implicitly correspond to existing IL strategies. To explicitly handle them in a unified learning framework, DUBE is proposed, along with a systematic discussion on the pros and cons of exisitng Inter/IntraCB solutions. Extensive comparative studies validate the effectiveness of DUBE. We note, however, DUBE is a simple and straightforward solution with clear room for improvement, e.g., the calibration of PBDA assumes the distributions of base classes are Gaussian. Beyond the discussions in this paper, we believe that more in-depth theoretical analysis is needed to find better solutions for inter-&intra-class balancing. To summarize, this work contributes some preliminary effort towards understanding the inter-class and intra-class imbalance in IL. We hope this work can shed some light on finding new ways to handle the IL problem.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge & Data Engineering, no. 9, pp. 1263–1284, 2008.

[2] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," Expert Systems with Applications, vol. 73, pp. 220–239, 2017.

[3] H. He and Y. Ma, Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons, 2013.

[4] F. Alberto, G. Salvador, G. Mikel, P. Ronaldo C., and K. Bartosz, Learning from Imbalanced Data Sets. Springer, 2018.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[6] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, 2008, pp. 1322–1328.

[7] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing. Springer, 2005, pp. 878–887.

[8] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in Proceedings of workshop on learning from imbalanced datasets, vol. 126, 2003.

[9] I. Tomek, "Two modifications of cnn," IEEE Trans. Systems, Man and Cybernetics, vol. 6, pp. 769–772, 1976.

[10] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," IEEE Transactions on Systems, Man, and Cybernetics, no. 3, pp. 408–421, 1972.

[11] M. Kubat, S. Matwin et al., "Addressing the curse of imbalanced training sets: one-sided selection," in Icml, vol. 97. Nashville, USA, 1997, pp. 179–186.

[12] X.-Y. Liu and Z.-H. Zhou, "The influence of class imbalance on cost-sensitive learning: An empirical study," in Sixth International Conference on Data Mining (ICDM'06). IEEE, 2006, pp. 970–974.

[13] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.

[14] C. X. Ling, Q. Yang, J. Wang, and S. Zhang, "Decision trees with minimal costs," in Proceedings of the twenty-first international conference on Machine learning. ACM, 2004, p. 69.

[15] X. Chai, L. Deng, Q. Yang, and C. X. Ling, "Test-cost sensitive naive bayes classification," in Fourth IEEE International Conference on Data Mining (ICDM'04). IEEE, 2004, pp. 51–58.

[16] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 761–769.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[18] Z. Liu, P. Wei, J. Jiang, W. Cao, J. Bian, and Y. Chang, "Mesa: Boost ensemble imbalanced learning with meta-sampler," Advances in Neural Information Processing Systems, vol. 33, 2020.

[19] Z. Liu, W. Cao, Z. Gao, J. Bian, H. Chen, Y. Chang, and T.-Y. Liu, "Self-paced ensemble for highly imbalanced massive data classification," in 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020, pp. 841–852.

[20] Q. Dong, S. Gong, and X. Zhu, "Class rectification hard mining for imbalanced deep learning," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1851–1860.

[21] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 821–830.

[22] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 2, pp. 539–550, 2009.

[23] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory," Knowledge and information systems, vol. 33, no. 2, pp. 245–265, 2012.

[24] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM SIGKDD explorations newsletter, vol. 6, no. 1, pp. 20–29, 2004.

[25] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," Progress in Artificial Intelligence, vol. 5, no. 4, pp. 221–232, 2016.

[26] G. Kovács, "Smote-variants: A python implementation of 85 minority oversampling techniques," Neurocomputing, vol. 366, pp. 352–354, 2019.

[27] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study." in WOB, 2003, pp. 10–18.

[28] B. Krawczyk, M. Woźniak, and G. Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification," Applied Soft Computing, vol. 14, pp. 554–562, 2014.

[29] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," Journal of Big Data, vol. 6, no. 1, pp. 1–54, 2019.

[30] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," IEEE transactions on pattern analysis and machine intelligence, 2020.

[31] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in European conference on principles of data mining and knowledge discovery. Springer, 2003, pp. 107–119.

[32] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 40, no. 1, pp. 185–197, 2010.

[33] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," Pattern Recognition, vol. 40, no. 12, pp. 3358–3378, 2007.

[34] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in 2009 IEEE Symposium on Computational Intelligence and Data Mining. IEEE, 2009, pp. 324–331.

[35] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of computer and system sciences, vol. 55, no. 1, pp. 119–139, 1997.

[36] L. P. Garcia, A. C. de Carvalho, and A. C. Lorena, "Effect of label noise in the complexity of classification problems," Neurocomputing, vol. 160, pp. 108–119, 2015.

[37] M. Koziarski, M. Woźniak, and B. Krawczyk, "Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise," Knowledge-Based Systems, vol. 204, p. 106223, 2020.

[38] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," Journal of Intelligent Information Systems, vol. 46, no. 3, pp. 563–597, 2016.

[39] M. Denil and T. Trappenberg, "Overlap versus imbalance," in Canadian Conference on Artificial Intelligence. Springer, 2010, pp. 220–231.

[40] V. García, J. Sánchez, and R. Mollineda, "An empirical study of the behavior of classifiers on imbalanced and overlapped data sets," in Iberoamerican Congress on Pattern Recognition. Springer, 2007, pp. 397–406.

[41] R. C. Prati, G. E. Batista, and M. C. Monard, "Learning with class skews and small disjuncts," in Brazilian Symposium on Artificial Intelligence. Springer, 2004, pp. 296–306.

[42] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," ACM Sigkdd Explorations Newsletter, vol. 6, no. 1, pp. 40–49, 2004.

[43] S. Gupta and A. Gupta, "Handling class overlapping to detect noisy instances in classification," The Knowledge Engineering Review, vol. 33, 2018.

[44] V. García, J. S. Sánchez, H. O. Domínguez, and L. Cleofas-Sánchez, "Dissimilarity-based learning from imbalanced data with small disjuncts and noise," in Iberian Conference on Pattern Recognition and Image Analysis. Springer, 2015, pp. 370–378.

[45] G. Kamath, "Bounds on the expectation of the maximum of samples from a gaussian," URL http://www. gautamkamath. com/writings/gaussian max. pdf, 2015.

[46] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 463–484, 2012.

[47] C. Zhang and Y. Ma, Ensemble machine learning: methods and applications. Springer, 2012.
[48] P. Hart, "The condensed nearest neighbor rule (corresp.)," IEEE transactions on information theory, vol. 14, no. 3, pp. 515–516, 1968.
[49] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[50] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," Machine learning, vol. 45, no. 2, pp. 171–186, 2001.
[51] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," Pattern Analysis & Applications, vol. 6, no. 3, pp. 245–256, 2003.
[52] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," Journal of Machine Learning Research, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html
[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," Journal of machine learning research, vol. 12, no. Oct, pp. 2825–2830, 2011.