Exploring the Learning Difficulty of Data: Theory and Measure

WEIYAO ZHU, OU WU*, FENGGUANG SU, and YINGJUN DENG, Center for Applied Mathematics,Tianjin University, China

"Easy/hard sample" is a popular parlance in machine learning. Learning difficulty of samples refers to how easy/hard a sample is during a learning procedure. An increasing need of measuring learning difficulty demonstrates its importance in machine learning (e.g., difficulty-based weighting learning strategies). Previous literature has proposed a number of learning difficulty measures. However, no comprehensive investigation for learning difficulty is available to date, resulting in that nearly all existing measures are heuristically defined without a rigorous theoretical foundation. This study attempts to conduct a pilot theoretical study for learning difficulty of samples. First, influential factors for learning difficulty are summarized. Under various situations conducted by summarized influential factors, correlations between learning difficulty and two vital criteria of machine learning, namely generalization error and model complexity, are revealed. Second, a theoretical definition of learning difficulty is proposed on the basis of these two criteria. A practical measure of learning difficulty is proposed under the direction of the theoretical definition by importing the bias-variance trade-off theory. Subsequently, the rationality of theoretical definition and the practical measure is respectively demonstrated by analysis of several classical weighting methods and abundant experiments realized under all situations conducted by summarized influential factors. The mentioned weighting methods can be reasonable explained under proposed theoretical definition and concerned propositions. The comparison in these experiments indicates that the proposed measure significantly outperforms the other measures throughout the experiments.

CCS Concepts: • **Computing methodologies** → **Instance-based learning**.

Additional Key Words and Phrases: Learning difficulty, generalization error, bias-variance trade-off, model complexity.

ACM Reference Format:
Weiyao Zhu, Ou Wu, Fengguang Su, and Yingjun Deng. 2022. Exploring the Learning Difficulty of Data: Theory and Measure. *ACM Trans. Knowl. Discov. Data.* 1, 1 (November 2022), 36 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The learning difficulty of a sample investigated in this study refers to how easy or hard it is to correctly learn the sample in a given learning task. For example, samples containing label noise or feature noise are less likely to be correctly classified, therefore they are tagged as hard samples in many works; model can collect more information from head categories (categories that have much more samples than others) in learning tasks with imbalance-distributed training data, hence samples from head categories are believed to be easier learned [48]. As an essence of data, learning difficulty of samples has earned great attention and is widely applied in various learning strategies. Among which, the partition of training data into different subsets according to their learning difficulties and adoption of separate learning

*Corresponding authors.

Authors' address: Weiyao Zhu, weiyaozhu@tju.edu.cn; Ou Wu, wuou@tju.edu.cn; Fengguang Su, fengguangsu@tju.edu.cn; Yingjun Deng, yingjun.deng@tju.edu.cn, Center for Applied Mathematics,Tianjin University, P.O. Box 92, Tianjin, China, 300110.

schemes are proven to be useful in many learning tasks [24, 48, 51, 74]. Although learning difficulty has no formal and consensus definition, it has been widely discussed and utilized in previous machine learning literature, including noise-aware, curriculum, and metric learning.

Numerous methods are proposed to measure the learning difficulty of a training sample fitting various learning tasks. The most common practice is to leverage the training output (e.g., loss and the predicted value on the true category) of a sample to construct the measurements. In Self-paced Learning (SPL) [29, 74], the training loss is used to determine whether a sample is easy or not, and easy samples are first learned. We assume that p_{i,y_i} is the prediction on the ground-truth category for a training sample x_i. In object detection, the value of $(1 - p_{i,y_i})$ is used to indicate the learning difficulty for x_i [48] . Given that the training output in an epoch may be unreliable, some methods utilize the average training output of a sample during the training to measure the difficulty. Huang et al. [37] designed a cyclic training procedure, and the model is trained from under-fitting to over-fitting in one cycle. The average training loss in the whole cyclic procedure is used as the noisy indicator for a training sample. Feng et al. [23] utilized the magnitude of the loss gradient to measure the learning difficulty of a training sample. A large gradient magnitude indicates a high degree of difficulty. Several existing methods focus on measuring learning difficulty by considering either bias or variance alone. For instance, several previous works [18, 37] use the averaged loss to represent the bias of a model's prediction, serving as a measure of learning difficulty. On the other hand, other studies [2, 39] explore the variation of loss to characterize different samples. Additionally, VoG [2] estimates the learning difficulty of samples by utilizing the variance of the gradient. However, it has been acknowledged and discussed in recent research that the bias term alone fails to fully capture the characteristics of a sample. To the best of our knowledge, none of the existing works have explored the use of both bias and variance as measures to evaluate the learning difficulty of samples.

Due to lack of a theoretical basis, different learning difficulty measures are based on different heuristic cues or empirical observations, resulting that each measure usually only suits specific application scenarios. A clearer understanding of the essence of a sample's learning difficulty can facilitate designing more effective learning difficulty measures. However, we are still far from concluding that we have a comprehensive understanding of learning difficulty:

(1) There is no summary of factors which directly affect the learning difficulty of samples. Current understandings of learning difficulty fail to fully cover all scenarios.
(2) There is no formal definition of the learning difficulty of a sample. Different studies exhibit different understandings of learning difficulty. An one-sided understanding usually results in a biased measure.
(3) There is no formal definition of the easy and hard samples. In most existing studies, easy and hard samples are heuristically judged. Consequently, it is nearly impossible to conduct a theoretical analysis for difficulty-based strategies with existing heuristic considerations.
(4) There has been few experimental studies particularly on the learning difficulty measure. Most studies only refer to the noisy learning or uncertainty settings. An extensive empirical evaluation under different settings is useful for the understanding the learning difficulty.

This study attempts to establish a preliminary theoretical definition for learning difficulty from the angle of the generalization error and the model complexity. The definitions of easy, medium and hard samples are subsequently proposed based on our theoretical definition. Based on the theoretical definition, a practical measure of learning difficulty is given by introducing the basic machine learning theory, namely bias-variance trade-off theory. The theoretical definition is supported by analysis under difficulty-based weighting learning methods including SPL and Focal loss. The proposed measure is empirically supported by the results of the extensive experiments.

Our contributions are summarized as follows:

- A summary of influential factors for learning difficulty of samples is provided. So far, this is the first summary while several factors have been mentioned separately in previous works.
- An attempt in theoretical definition of learning difficulty is made based on the generalization error and the model complexity. Formal definitions of easy and hard samples are established. Theoretical analysis of definition under weighting strategies is provided to support the reasonableness of theoretical definitions. As far as we are aware, this is the first attempt on this formalization.
- A practical measure of learning difficulty, which incorporates both bias and variance for training data, is proposed on the basis of theoretical definition by importing bias-variance trade-off theory. Extensive experiments are realized and our proposed measure significantly exceeds other measures under all scenarios.

2 RELATED WORK

2.1 Learning Difficulty Measurement

Learning difficulty is considered as an intrinsic property of data in machine learning [51, 81]. Existing measurements are usually based on heuristic cues or inspirations, and they can be divided into the following main categories:

- Loss-based measurement. This category directly uses the loss as the measure. Most measures fall into this category because it is simple yet effective in various learning tasks. Some methods [74] directly utilize the loss in one epoch as the degree of difficulty. Accordingly, the degrees for the same samples vary in different epochs. Some others utilize the average loss [50] during the partial or whole training procedure for measurement.
- Cross-validation-based measurement. This category adopts a cross-validation strategy [71]. For example, five-fold cross-validation is performed, and the whole cross-validation is repeated ten times. Consequently, each training sample receives ten predictions. The value of error predictions is used as the indicator of difficulty.
- Uncertainty-based measurement. This category uses the (model) uncertainty of a sample to measure the difficulty. D'souza et al. [18] firstly propose a framework that models both the level and source of uncertainty in samples to identify atypical and noisy samples, which insight us the relationship between uncertainty and label noise. Aguilar et al. [3] identify hard samples based on the model uncertainty and leveraged the Bayesian Neural Network [73] to infer the uncertainty.
- Margin-based measurement. This category uses the margin (distance) of a sample to the underlying decision surface as the measurement. The rationale is that a small margin denotes a large difficulty [47, 74].
- Gradient-based measurement. This category uses the loss gradient of a sample to measure the difficulty. Agarwal et al. [2] proposed the variance of gradients (VoG) across different epochs to rank data from difficult to easy. They considered that samples with high VoG values are far more difficult for the model to learn. Santiago et al. [59] applied the norm of the gradients to measure the difficulty, and high norms indicate large difficulty for learning.

The above-mentioned categories are highly correlated. For example, margin-based measurement is indeed a loss-based one when margin-based loss (e.g., hinge loss) is used.

2.2 Noisy-label Learning

Noisy labels are inevitable even in benchmark data sets [31, 37, 38, 45]. Various methods are explored to detect noisy labels. Existing noise detection methods are usually based on the information used for learning difficulty measurements,

such as loss and gradient, because samples with noisy labels are usually considered as quite hard samples. Some studies model the generation process of the noisy labels to detect them [21, 43]. Forgetting [39] is a recently recognized phenomenon that can be used to identify noisy samples. It has been observed that noisy samples are more likely to be forgotten by the model. The concept of forgetting describes the variation in the model's prediction for a given sample and utilizes this variation to characterize noisy samples. VoG [2] uses the variance of the gradient to characterize noisy samples. A recent survey can be referred to [30].

2.3 Curriculum Learning

Curriculum learning [10] draws lessons from the human learning process, which begins with the simplest and progresses to more difficult courses. Easier samples should be learned at the beginning of a learning process and gradually advance towards harder samples. SPL pertains to curriculum learning and the difficulty is measured by loss.

2.4 Uncertainty-aware Learning

Uncertainty in learning mainly refers to aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty is also called data uncertainty. A related work [4] wisely estimated uncertainty in labeling, which is also recognized as aleatoric uncertainty, since accurate and consistent labeling has high unsureness in real-world. To reduce the influence of high-uncertainty samples, they lower the weights of samples with high density and label entropy. Epistemic uncertainty is also called model uncertainty. It occurs when there is no fixed annotation for a given training sample in some learning tasks. The predictive entropy [40] and Bayesian Neural Network [73] have been used to measure epistemic uncertainty.

2.5 Model Complexity

The model complexity [65] discussed in this work describes how intricacy a neural network is. For a given learning task, an optimal model complexity consequentially exists. An oversimplified model is recognized as under-fitted, and a sophisticated model is recognized as over-fitted. The construction of model complexity varies from task to task, which leads to an implicit form of model complexity in a general way. Several researches [20, 22, 46] calculate approximately the model complexity according to their need and learning tasks and do comparisons without fixing the structure of basic model, which means that they study the complexity of models under different structures. However, there is few works that explicitly study complexity of a fixed model during a whole learning procedure under general settings.

3 THEORETICAL DEFINITION OF LEARNING DIFFICULTY

Existing learning difficulty measurements mentioned in Section 2.1 are empirically utilized in diverse situations. Despite the effect emerges under difficulty-based learning schemes, there is still a lot of room of improvement. Because existing measurements are proposed heuristically without a theoretical basis. Moreover, incentives of difference between learning difficulty of samples are multifarious. Existing measurements are mainly of unilateral considerations, which could not cover the majority incentives.

In this section, we summarize influential factors for learning difficulty of samples at first. Accordingly, by considering summarized influential factors, correlations between learning difficulty and two vital criteria of machine learning, namely generalization error and model complexity, are respectively disclosed. Subsequently, a theoretical definition of learning difficulty is given under consideration of above-mentioned correlations. In order to generate a practical measure based on the theoretical definition, we introduce bias-variance trade-off theory. Finally, a feasible measure is proposed.

3.1 Influential Factors for Learning Difficulty

So far, there has been no study that comprehensively summarizes the factors that determine the learning difficulty of a sample. In machine learning, differences among samples mainly lie in noise level, spacial location, neighbourhood, and overall distribution. Illuminated by these differences and the heuristic inspirations considered in previous measures for learning difficulty, the main influential factors are roughly summarized as follows:

- Data quality. Both feature and label noises affect the learning difficulty of samples. Young et al. [77] found that high signal-to-noise ratio signifies high feature noise level and generates low data quality, which hinders the optimization of the learning task and is harder to be well learnt. Su et al. [63] revealed that mislabeled images are low-quality data for the learning task and are of high difficulty.
- Sample margin. The sample margin is defined as the distance between the sample and the true decision boundary. Huang and Yang [36] considered that samples with small margins are hard to learn.
- Uncertainty. The (model) uncertainty of a sample is usually measured by the information entropy of its prediction [22]. The higher the information entropy is, the ampler the information is contained by the sample. Samples with higher uncertainty are more difficult to adequately learn [75].
- Category distribution. Oinar et al. [56] showed that category with fewer samples, which is also called tail category, is usually more challenging than category with more samples, which is known as head category.

3.2 Generalization Error and Model Complexity

A well-established theoretical definition should at least cover the majority incentives. The manifestation and variation of generalization error and model complexity directly correlates to the learning difficulty of samples and basically reflects the learning difficulty under majority situations.

3.2.1 *Notations.*

The features and the label of a sample are seen as two random variables, and are denoted as X and Y respectively. Realizations of X and Y are denoted as $\mathbf{x}_i$ and y_i respectively. Let $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a random training set and a sample $(\mathbf{x}_i, y_i)$ is denoted as x_i for convenience. We assume that X and Y conform to the joint distribution $P(X, Y)$, where $(X, Y) \in \Omega$ with $\Omega = \Omega_X \times \Omega_Y = \{(X, Y)|X \in \Omega_X, Y \in \Omega_Y\}$. Let $\lambda_h \in R_\lambda$ be the hyper-parameter(s), where R_λ is the feasible region[1]. Given a basic learner f trained on T and a fixed value of λ_h, the prediction of sample x_i is denoted by $f(x_i; T, \lambda_h)$[2].

3.2.2 *Generalization Error.*

The generalization error [48] (also known as the expected risk) of all training samples sampled from Ω is initially defined in regression tasks in following form:

$$Err(\lambda_h) = \mathbb{E}_{x_i \in \Omega} \mathbb{E}_T[\|y_i - f(x_i; T, \lambda_h)\|_2^2]. \tag{1}$$

Accordingly, the generalization error of a given sample x_i in regression tasks is given by:

$$Err(x_i, \lambda_h) = \mathbb{E}_T[\|y_i - f(x_i; T, \lambda_h)\|_2^2].$$

[1]Note that the hyper-parameter should locate in a feasible region. For example, if λ_h is the learning rate, then $\lambda_h < 0$ is meaningless.
[2]Once the model's structure is fixed, the parameters of the model trained on the given T and λ_h are fixed. Therefore, different from the common view which denotes the model as $f(\cdot; \theta)$, we use $f(x_i; T, \lambda_h)$ instead.

Notation	Description
X	the data feature
Y	the true label
$\mathbf{x}_i$,y_i	realizations of X and Y
$(\mathbf{x}_i, y_i)$	a sample x_i
N	number of samples
$T = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$	a training set
$P(X, Y)$	the joint distribution of X and Y
Ω_X	the feature space
Ω_Y	the label space
Ω	the whole space
λ_h	the hyper-parameter(s)
R_λ	the feasible region of λ_h
$f(\cdot; T, \lambda_h)$	the model trained on T under λ_h
$f(x_i; T, \lambda_h)$	the prediction of sample x_i given by $f(\cdot; T, \lambda_h)$
f_i^t	the prediction of sample x_i given by $f(\cdot; T, \lambda_h)$ on t-th epoch
c	the expectation of model complexity on Ω
Err	the generalization error
$Err(x_i, \lambda_h)$	the generalization error of sample x_i under λ_h on Ω
$g(\cdot)$	the mapping from λ_h to c
c^*	the optimal model complexity on Ω
λ_h^*	the optimal hyper-parameter
$Err(\mathcal{S}, \lambda_h)$	the generalization error of the set $\mathcal{S}$
$\mathcal{LD}(x)$	the theoretical learning difficulty for a sample x_i
$\mathcal{LD}(\mathcal{S})$	the theoretical learning difficulty for a set of samples $\mathcal{S}$
$\mathcal{LDC}(x_i)$	the learning difficulty coefficient
$BiasT(\lambda_h)$	the learning bias
$VarT(\lambda_h)$	the variance term
δ_e	the irreducible noise
$l(\cdot, \cdot)$ or ℓ	the loss

Table 1. Summary of the Notations.

Previous studies[3] reveal correlations between learning difficulty of samples and generalization error. In terms of data quality, Wang et al. [68] revealed that negative impacts of noisy implicit feedback occurs to the minimization of generalization error. Castells et al. [14] concluded that noisy samples tend to be harder and injure the model generalization. In terms of sample margin, Zhang et al. [79] concluded that samples close to boundary, are further from reaching near-zero generalization error than samples away from boundary. In terms of uncertainty, Pagliardini et al. [57] improved the model's generalization by estimating uncertainty quantification and perturbing high uncertain samples. In terms of category distribution, Gautheron et al. [26] derived a bound of generalization error in metric learning by involving the proportion of minority examples who throw higher generalization error values.

3.2.3 *Model Complexity.*

As is mentioned in section 2.5, there is few works directly and precisely give an explicit form for the complexity of a model under fixed structure during a whole learning procedure. However, compared with studies of models under

[3]In fact, existing studies focus on the generalization error over the whole data space rather than a local region or a single sample. Nevertheless, the positive correlation between learning difficulty and Err_x^* of a single sample is theoretically verified in our continuous study. The theoretical proofs are uploaded to Github source repository.

different structures [20, 22, 46], it is also of great importance to comprehend the changing process of a given model during training. Hence, we give a general form for the complexity of model under fixed structure.

Given a model $f(\cdot\,; T, \lambda_h)$ trained on a sampled training set T with λ_h. According, when given a sample x_i, the model $f(\cdot\,; T, \lambda_h)$ gives its prediction $f(x_i; T, \lambda_h)$. Model complexity $m(f(\cdot\,; T, \lambda_h))$ is a function depends on the given model $f(\cdot\,; T, \lambda_h)$[4] and $m(\cdot)$ may vary according to various learning tasks. Let c be the expectation of the model complexities $m(f(\cdot\,; T, \lambda_h))$ over different T given λ_h. c depends on λ_h and the distribution of T If the distribution of T, the structure of the base network f, and the function m are given, then c only depends on λ_h, i.e.,

$$c = g(\lambda_h) = \mathbb{E}_{x_i \in \Omega} \mathbb{E}_T [m(f(x_i\,; T, \lambda_h))], \quad \lambda_h \in R_\lambda, \tag{2}$$

where the function $g(\cdot)$ is defined as the mapping from λ_h to c. In the rest of the paper, the model complexity expectation is briefly termed as "model complexity". c depends on the base network (e.g., AlexNet, Transformer, and ResNet-34), λ_h (e.g., learning rate and the maximum learning epoch), and the distribution of T.

Accordingly, the model complexity of sample x_i is given by

$$c_{x_i} = \mathbb{E}_T [m(f(x_i\,; T, \lambda_h))], \quad \lambda_h \in R_\lambda. \tag{3}$$

Likewise, there is also a mapping relationship from λ_h to c_{x_i}.

Likewise, the correlation between learning difficulty and model complexity is also discussed in previous literature. Arpit et al. [6] gave a descriptive definition for easy (as well as hard) samples that "easier examples are explained by some simple patterns, which are reliably learned within the first epoch of training". This definition implies that easy samples can be modeled by simple models, which motivates us to build a theoretical description with model complexity. Noisy samples, which are generally considered as hard samples, result in a higher model complexity if these samples are well-learned [58, 61]. To correctly learn a sample in tail categories, strategy such as over-sampling is usually introduced and leads to a higher model complexity [1, 52]. To well learn the sorts of small-margin samples, which are close to the true boundary and are recognized as hard samples, the learned boundary turns to be more complex [53, 78]. A more uncertain sample has a larger variance of prediction. A more complex model is needed to reduce its variance, which is to correctly classify [4, 5, 20].

3.2.4 *Relationship Between Generalization Error and Model Complexity.*

The investigation of theoretical definition for learning difficulty of samples starts with revealing the relationship between generalization error and model complexity.

In this study, the base network f is assumed to be fixed. Therefore, both Err and $c = g(\lambda_h)$ are the functions of λ_h when $m(\cdot)$ and the distribution of T are given. Accordingly, if $g(\lambda_h)$ is reversible, Err can be seen as the function of c according to Eqs. (1) and (2), i.e., $Err(c) = \mathbb{E}_{x_i \in \Omega} \mathbb{E}_T [\|y_i - f(x_i; T, g^{-1}(c)\|_2^2]$.

The minimum generalization error is achieved when the partial derivatives of generalization error with respect to c equal to zero. The optimal hyper-parameter λ_h^* and the optimal model complexity on the whole space c^* are obtained with

$$\lambda_h^* = arg \min_{\lambda_h} Err(\lambda_h) \tag{4}$$

$$c^* = g(\lambda_{hyper}^*). \tag{5}$$

[4]Once the model's structure is fixed, the model complexity depends on both the training set T and the hyper-parameters λ_h. Therefore, the learned model's complexity is influenced by T and λ_h. The expectation of the model complexity is determined by the distribution of T and λ_h.

3.3 Definition of Learning Difficulty

Eq. (1) is defined on the whole space Ω. $P(x, y)$ is often unknown. The mean square error (MSE) used in Eq. (1) is for regression tasks and is not suitable for classification tasks. Generally, according to Eq. (1), we define the generalization error of a sample as follows:

Definition 1. *Generalization error (or expected risk) of a sample x_i is in form of*

$$Err(x_i, \lambda_h) = \mathbb{E}_T[l(y_i, f(x_i; T, \lambda_h))], \tag{6}$$

where $l(\cdot, \cdot)$ measures the error between the label y_i and a prediction $f(x_i; T, \lambda_h)$.

Accordingly, we define the generalization error of a set of samples as follows:

Definition 2. *Generalization error (or expected risk) of a set of samples, noted as $\mathcal{S}$, is in form of*

$$Err(\mathcal{S}, \lambda_h) = \mathbb{E}_{x_i \in \mathcal{S}} \mathbb{E}_T[l(y_i, f(x_i; T, \lambda_h))]. \tag{7}$$

We define a theoretical definition for the learning difficulty of a sample.

Definition 3. *Given a fixed basic learner[5] f, the theoretical learning difficulty for a sample x_i is*

$$\begin{aligned} \mathcal{LD}(x) &= c^*_{x_i} = g(\lambda^*_{h,i}) \\ s.t., \lambda^*_{h,i} &= arg\min_{\lambda_h} Err(x_i, \lambda_h). \end{aligned} \tag{8}$$

Accordingly, the learning difficulty for a set of samples can be defined.

Definition 4. *Given a fixed basic learner, the theoretical learning difficulty for a set of samples $\mathcal{S}$ is*

$$\begin{aligned} \mathcal{LD}(\mathcal{S}) &= c^*_{\mathcal{S}} = g(\lambda^*_h) \\ s.t., \lambda^*_h &= arg\min_{\lambda_h} Err(\mathcal{S}, \lambda_h). \end{aligned} \tag{9}$$

Our definition for learning difficulty of samples is consistent with the descriptive definition given by some other studies. Charrerjee and Zielinskix [15] observe that easy ImageNet samples are learned earlier and hard ImageNet samples are learned later. Arpit et al. [6] state that the model complexity will increase with the increasement of training epoch gradually. Scilicet, an easier sample corresponds to a smaller optimal model complexity, and a harder sample corresponds to a larger one.

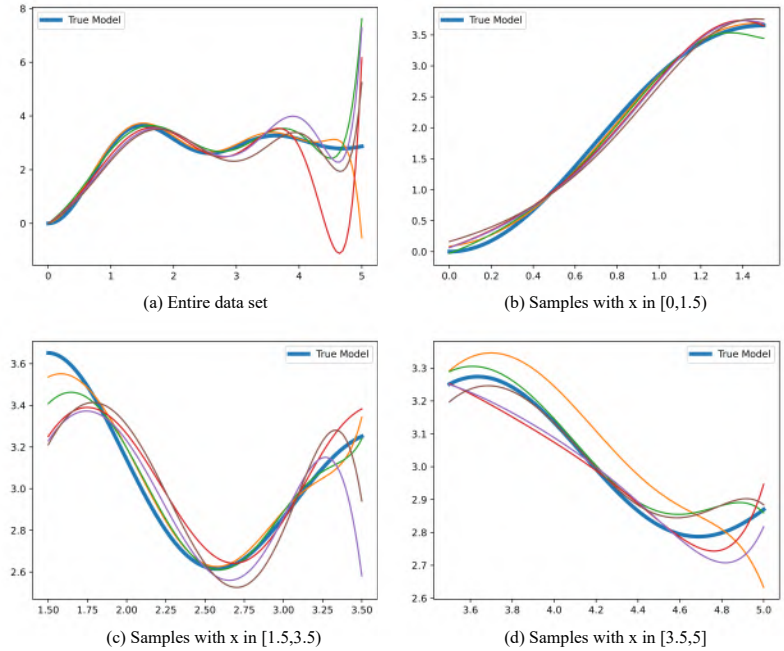
The relative learning difficulty between two samples x_1 and x_2 is obtained according to Definition 3. If $\mathcal{LD}(x_1) > \mathcal{LD}(x_2)$, then x_1 is more difficult than x_2, and vice versa.

An example of non-linear regression learning is utilized to empirically explain the definition for learning difficulty of samples.

Example 1. 4000 realizations of random variable x are sampled uniformly from $[0, 5]$. The true target value y of a sample x is given by the target model $f(x) = 3 - sin(3x)/x$. The target value is then perturbed by Gaussian noise, i.e., $\hat{y} = y + \epsilon = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1.2)$.

A 10-degree polynomial function is used and trained by ridge regression, i.e., $\hat{f}(x) \sim \mathcal{O}(10)$. The hyper-parameter λ in ridge regression is searched in $\{e^{-7}, e^{-6}, e^{-5}, e^{-4}, e^{-3}, e^{-2}, e^{-1}, e^{0}, e^{1}\}$. Under different values of λ, the complexities

[5] Essentially, when $Err(\lambda_h) = \mathbb{E}_f \mathbb{E}_{(x_i, y) \in \Omega} \mathbb{E}_T[\| y_i - f(x_i; T, \lambda_h) \|_2^2]$ is used, the learning difficulty is independent of the basic learner f.



(a) Entire data set (b) Samples with x in [0,1.5)

(c) Samples with x in [1.5,3.5) (d) Samples with x in [3.5,5]

Fig. 1. Illustrations of comparisons between learned and true models under imbalance sampling.

of fitting models differ accordingly. For each value of λ, 40 fitting models are learned using different random training sets.

Imbalance sampling is applied and each training set is consists of 100 random samples with $x \in [0, 1.5)$, 50 random samples with $x \in [1.5, 3.5)$, and 25 random samples with $x \in [3.5, 5]$. The samples in the additional data set are sampled with the same imbalance strategy. Under each value of λ, 40 models are learned using different training sets sampling from the initial 4000 realizations. ■

Let $\hat{\mathbf{w}}_t = (\hat{w}_{t,1}, \cdots, \hat{w}_{t,10})^T$ be the model parameter learnt on a training set T_t. The model complexity of a learnt model $f_{\hat{\mathbf{w}}_t}$ parameterized by $\hat{\mathbf{w}}_t$ is calculated as follows:

$$m(f_{\hat{\mathbf{w}}_t}) = \sum_{i=1}^{10} (\frac{i}{10} \hat{w}_{t,i})^2,$$

and the model complexity expectation is calculated as:

$$c(\hat{\mathbf{w}}(\lambda)) = \frac{1}{40} \sum_{t=1}^{40} m(f_{\hat{\mathbf{w}}_t}) = \frac{1}{40} \sum_{t=1}^{40} [\sum_{i=1}^{10} (\frac{i}{10} \hat{w}_{t,i})^2] \tag{10}$$

Details of the calculation are presented in Appendix A.

The imbalance sampling aims to generate three regions comprising easy, medium, and hard samples, respectively. Samples with $x \in [0, 1.5)$ are relatively easy and those with $x \in [3.5, 5]$ are relatively hard. The fitting curves are shown in Fig. 1. In Fig. 1(a), fitting curves differ distinctly from the true model in hard region (i.e., $x \in [3.5, 5]$) but precisely fit the true model in easy region (i.e., $x \in [0, 1.5)$), which shows that error of samples in easy region is much smaller than

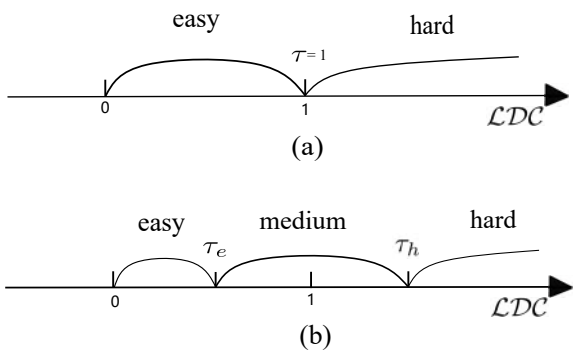


Fig. 2. Illustrations for dichotomy (a) and trichotomy (b) of samples.

that of hard region. Intuitively, fitting curves in easy region (Fig. 1(b)) are much less complex than those in hard region (Fig. 1(d)), which means that hard samples have higher optimal model complexity than easy samples.

We also define a learning difficulty coefficient as follows:

Definition 5. *Given the optimal model complexity c^* on the whole space Ω and the learning difficulty of the sample x_i, the learning difficulty coefficient ($\mathcal{LDC}$) is defined as*

$$\mathcal{LDC}(x_i) = \frac{\mathcal{LD}(x_i)}{c^*} = \frac{c^*_{x_i}}{c^*} \tag{11}$$

The larger the value of $\mathcal{LDC}$ is, the more difficult the sample x will be. The succeeding subsection will define the easy and hard samples based on $\mathcal{LDC}$.

3.4 Definitions of Easy and Hard Samples

Many existing studies are based on the two or three splits for training samples, namely easy/hard and easy/medium/hard, respectively. With $\mathcal{LDC}$, the dichotomy is defined as follows:

Definition 6. *Given a sample x_i and its learning difficulty coefficient $\mathcal{LDC}$, if $\mathcal{LDC}(x_i) \leq 1$, then x_i is an easy sample; if $\mathcal{LDC}(x_i) > 1$, then x is a hard sample.*

Definition 6 is flexible because the threshold can be a parameter instead of a fixed value. Let τ be the threshold. If $\mathcal{LDC}(x) \leq \tau$, then x is an easy sample; if $\mathcal{LDC}(x) > \tau$, then x is a hard sample.

In the trichotomy, distinguishing between easy and medium or medium and hard is difficult. Accordingly, we propose the following definition for these partitions:

Definition 7. *Given a sample x_i and its learning difficulty coefficient $\mathcal{LDC}(x_i)$, let τ_e and τ_h be two positive parameters and $0 < \tau_e < 1 < \tau_h$. If $\mathcal{LDC}(x_i) \leq \tau_e$, then x_i is an easy sample; if $\tau_e < \mathcal{LDC}(x_i) \leq \tau_h$, then x_i is a medium sample; if $\mathcal{LDC}(x_i) > \tau_h$, then x_i is a hard sample.*

The two parameters depend on the concrete application tasks and data characteristics. The above two definitions describe the dichotomy and trichotomy for samples as shown in Fig. 2. Some samples are quite hard and are harmful to learning process. We can also define quite-hard samples if $\mathcal{LDC}(x) > \tau_q$ and $\tau_q > \tau_h$.

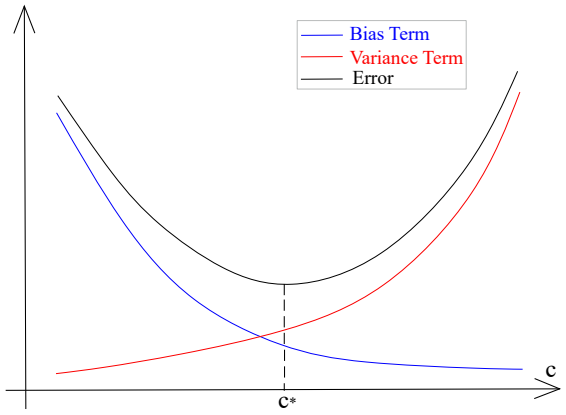


Fig. 3. The bias-variance trade-off curve.

3.5 Bias-Variance Trade-Off for Generalization Error

Although the theoretical definition of learning difficulty is solid and comprehensive, it is not ready for algorithmic implementation due to lack of a consensus calculation for model complexity (the function $m(\cdot)$ in Eq. 2). However, owing to the generalized knowledge and previous works mentioned in section 3.1, the following three positive correlations can be revealed:

- between the learning difficulty and the minimum generalization error;
- between the learning difficulty and the optimal model complexity;
- between the minimum generalization and the optimal model complexity.

Although the model complexity has no explicit form, generalization error can be explicitly written and approximately estimated. In order to practically investigate the learning difficulty defined by the optimal model complexity, the minimum generalization error is used because of above-mentioned positive correlations. A typical theory of machine learning containing both generalization error and model complexity, namely the bias-variance trade-off theory, is imported. With the help of the imported theory and the above-mentioned correlations, we study precisely the variation of generalization error with respect to the model complexity. More precisely, the variations of bias and variance with respect to the model complexity are also revealed. In fact, most existing measurement methods utilize the training loss (or loss variance) which can be considered as an approximation of the bias (or the variance) term of generalization error as a measurement of learning difficulty. Nevertheless, no study considers the bias and the variance terms simultaneously. Therefore, it is logical to consider both the bias term and the variance term, which leads us to bias-variance trade-off theory.

Bias-variance trade-off is initially constructed on regression and mean square error (MSE) is used [34]. Eq. (1) can be factorized into

$$Err(\lambda_h) = \mathbb{E}_{x\in\Omega}[\|y - \overline{f}(x;\lambda_h)\|_2^2] + \mathbb{E}_{x\in\Omega}\mathbb{E}_T[\|\overline{f}(x;\lambda_h) - f(x;T,\lambda_h)\|_2^2] + \delta_e\ , \tag{12}$$

where $\bar{f}(x;\lambda_h) = \mathbb{E}_T[f(x;T,\lambda_h)]$, and δ_e is known as the irreducible noise, and is independent of the basic learner and λ_h. The first and the second terms of the right side of Eq. (12) are the learning bias and variance terms, respectively, shown as follows:

$$Bias^2(\lambda_h) = \mathbb{E}_{x\in\Omega}[\|y - \overline{f}(x;\lambda_h)\|_2^2]. \tag{13}$$

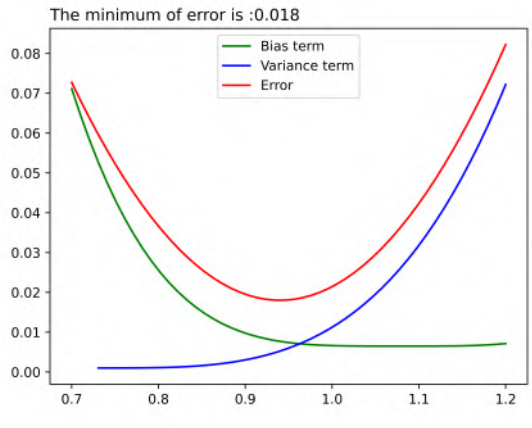


Fig. 4. Illustration of bias-variance trade-off under non-linear regression.

$$Var(\lambda_h) = \mathbb{E}_{x\in\Omega}\mathbb{E}_T[\|\overline{f}(x;\lambda_h) - f(x;T,\lambda_h)\|_2^2]. \tag{14}$$

In classification, the above derivation becomes complex [79]. Nevertheless, the following expression holds, with $BiasT$ and $VarT$ denoting the bias and the variance terms respectively:

$$Err(\lambda_h) = BiasT(\lambda_h) + VarT(\lambda_h) + \delta_e. \tag{15}$$

Variable y is categorical in classification. We suppose that x is continuous in order to consider in the total space, and facilitate the inference for classification. Therefore, the generalization error for a region $\Omega^r \subset \Omega$ s.t. $\Omega^r = \Omega_X^r \times \Omega_Y = \{(x,y)|x \in \Omega_X^r, y \in \Omega_Y\}$ is defined as

$$Err(\Omega^r, \lambda_h) = \sum_{y\in\Omega_Y} P(y) \int_{x\in\Omega_X^r} \mathbb{E}_T[l(y, f(x;T,\lambda_h))]p(x|y)\mathrm{d}x, \tag{16}$$

where $l(\cdot,\cdot)$ measures the error between the label and a prediction. $P(y)$ signifies the probability when the label equals to y and $p(x|y)$ is the conditional probability density function of x when the label equals to y.

The following widely accepted assumption[6] holds for both regression and classification.

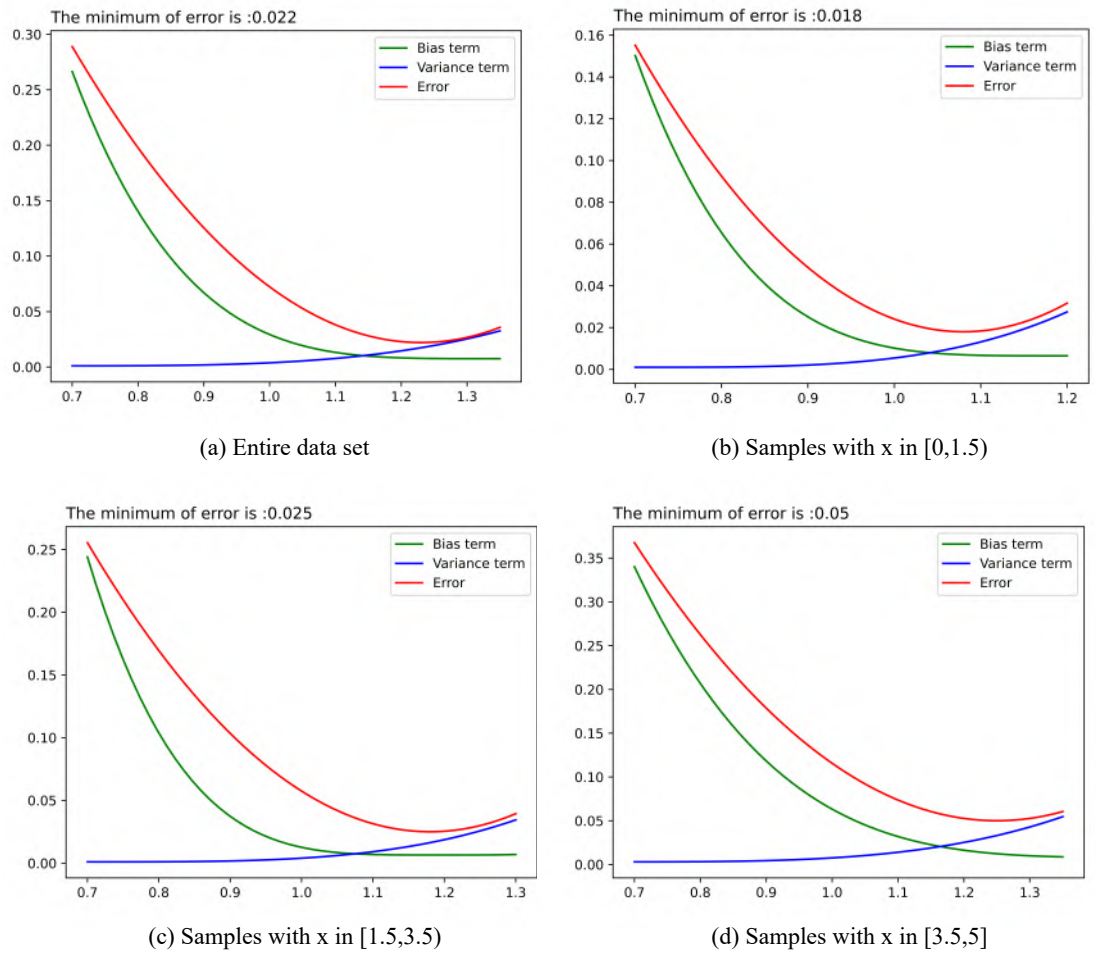
Assumption 1. *The bias term is a decreasing function of the expectation of model complexity c, whereas the variance term is an increasing function of c when the basic learner is fixed. The generalization error decreases first and then increases.*

An example of non-linear regression learning is still utilized to empirically support Assumption 1.

Example 2. The target model and the calculation of model complexity in Example 1 are still used. Each training set is composed by 200 samples randomly sampled from the 4000 realizations. An additional test set is constructed in the same way as training sets and is of the same size. ■

The bias, variance, and generalization error curves of Example 2 are given in Fig. 4. Assumption 1 holds with regard to this example. A clear bias-variance trade-off is presented. The bias curve decreases with respect to the employed

[6]Most professional books and papers explicitly or implicitly apply this assumption without giving a strict proof. Some recent studies point out that the variance curve is not increasing any more in some cases [48]. However, the structures of base models in these studies are also varied. Meanwhile, the structures of base models in this study are assumed to be fixed.



(a) Entire data set

(b) Samples with x in [0,1.5)

(c) Samples with x in [1.5,3.5)

(d) Samples with x in [3.5,5]

Fig. 5. Illustrations of bias-variance trade-off under imbalance sampling.

model complexity while the variance term increases. The generalization error first decreases to its minimum, and then increases. The minimum of the average generalization error over all samples is 0.018 and is attained around the intersection of the bias curve and the variance curve.

According to Assumption 1, the minimum generalization error is achieved when the partial derivatives of generalization error with respect to c equals to zero, i.e., the sum of the partial derivatives of its bias term and the corresponding variance term with respect to c equals to zero. A diagrammatic drawing of the bias-variance curve is shown in Fig. 3 [28, 32].

Similar to Assumption 1, we have the following assumption:

Assumption 2. *Assume that the basic model is given and fixed. The bias term for x is a decreasing function of c, whereas the variance term for x is an increasing function of c. The generalization error for x decreases first and then increases.*

Example 1 is reused and analysed to empirically support Assumption 2 by illustrating the bias, variance, and generalization error curves in different regions.

The imbalance sampling aims to generate three regions comprising easy, medium, and hard samples, respectively. Samples with $x \in [0, 1.5)$ are relatively easy and those with $x \in [3.5, 5]$ are relatively hard. The learning curves are shown in Figs. 5 and 1. Fig. 5 shows the bias-variance trade-off curves for the entire data set (Fig. 5(a)), the samples from $[0, 1.5)$ (Fig. 5(b)), the samples from $[1.5, 3.5)$ (Fig. 5(c)), and the samples from $[3.5, 5]$ (Fig. 5(d)), respectively. Under all cases, the bias term decreases with respect to the employed model complexity, while the variance term increases. The generalization error firstly decreases to its minimum, and then increases. The minimum values of generalization error vary: 0.022 for the entire data set (Fig. 5(a)), 0.018 for the majority sampling part ($[0, 1.5)$) (Fig. 5(b)), 0.025 for the medium sampling part ($[1.5, 3.5)$) (Fig. 5(c)), and 0.05 for the minority sampling part ($[3.5, 5]$) (Fig. 5(d)). Alternatively, both Assumptions 1 and 2 hold in this example.

According to Assumption 2, the minimum of $Err(x, \lambda_h)$, denoted as Err_x^*, is also attained when the partial derivatives of generalization error for x on c equals to zero, i.e., the sum of the partial deviations of the bias term and the variance term for x on c is zero.

3.6 The Proposed Measure for Learning Difficulty

An approach is proposed based on Err_x^*, and is utilized as the practical measure of learning difficulty.

Considering that it is also infeasible to calculate Err_x^* traversing all values of λ_h, we only calculate the generalization errors $Err(x, \lambda_h)$ for each sample with the same reasonable[7] λ_h to approximate the learning difficulty. As widely accepted in existing literature [19, 35], reasonable hyper-parameters that optimize the performance of the algorithm are typically obtained through techniques such as cross-validation, selection using a validation set, and meta-learning. In our study, we use the validation set to determine the values of λ_h that optimize the algorithm's performance. Specifically, the proposed approach adopts the cross-validation strategy to calculate the average learning errors for each sample. First, the whole training set is divided into M folds. $M-1$ folds are alternatively used for training, and the trained model is used to predict the label of all training samples. This cross-validation process is repeated for K times. Each sample receives $K * M$ predictions, with which can we calculate the average prediction of each sample. Second, average losses and variance of losses for each training sample are calculated using corresponding average predictions.

Let $p_{i,m}^k$ be the prediction of x_i in the m^{th} cross-validation of the k^{th} repeat. Then, according to [76], we calculate:

$$\overline{p}_i = exp\{\frac{1}{M*K}\sum_{m,k} log(p_{i,m}^k)\}. \tag{17}$$

Subsequently, the bias and the variance terms are calculated as follows

$$Bias_i \approx l_{CE}(y_i, \overline{p}_i), \tag{18}$$

$$Var_i \approx \frac{1}{M*K}\sum_{m,k} l_{CE}(\overline{p}_i, p_{i,m}^k), \tag{19}$$

where l_{CE} is the standard cross-entropy loss. The actual value of learning difficulty of x_i is

$$Err(x_i, \lambda_h) \approx Bias_i + \mu Var_i\ , \tag{20}$$

[7] In our future work, we plan to explore dividing the entire training set into subsets, where each subset will adopt the same λ_h. This strategy has the potential to better balance accuracy and efficiency compared to our current approach, where the same λ_h is used for all samples.

Algorithm 1 GELD

Input: Training set T with N samples, validation data, M, K, μ, and λ_h
Output: $Err(x_i, \lambda_h)$, $i = 1, \cdots, N$.

for k in 1 to K **do**
  Randomly split T into $T_1^{(k)}, \cdots, T_M^{(k)}$;
  for m in 1 to M **do**
    Perform the training on $T - T_m^{(k)}$. The model is selected with the validation data;
    Predict the label $p_{i,m}^k$ of x_i for each sample;
  end for
end for
Calculate $\overline{p}_i$ using Eq. (17) for each sample;
Calculate $Bias_i$ and Var_i using Eqs. (18) and (19);
Calculate $Err(x_i, \lambda_h)$ using Eq. (20).

where μ is a tuning factor for the variance. The value of $Err(x_i, \lambda_h)$ is used as the learning difficulty for x_i. This approach is called generalization error-based learning difficulty (GELD) measurement. The detailed steps of GELD are shown in Algorithm 1. The primary difference between our approach and the existing loss-based/cross-validation-based methods lies in that our approach does not discard the variance term but combines the importance of both term of generalization error. If $\mu = 0$, then GELD is similar to the conventional cross-validation-based methods. Several existing methods including O2UNet [37] also point out that hard samples have high loss variances.

3.7 Comparison with Existing Works

Existing studies that explore sample characteristics often employ measures based on either bias or variation. For instance, certain studies [18, 37] utilize the averaged loss as a measure, which primarily captures the model's prediction bias. Other works [2, 39] take a step forward and consider the variation of the loss or gradient to provide a more comprehensive understanding of the sample characteristics. In this subsection, we will theoretically analyze three classic measures, focusing on the bias or the variation.

- O2UNet: It proposes to identify noisy samples using the average loss of each sample, which is the mean of CE loss for each sample, i.e.,

$$\overline{\ell}_i = \frac{1}{\mathbf{T}} \sum_{t=1}^{\mathbf{T}} \ell_i^t, \tag{21}$$

  where $\mathbf{T}$ is the number of epochs. Considering the equation mentioned above, it is evident that the averaged loss can be regarded as the bias term in the bias-variance decomposition.
- Forgetting: It is a phenomenon referring to the degradation of previously acquired knowledge in a neural network over time. Specifically, if a sample x_i is forgotten by the model at the $t+1$-th epoch, it indicates that the model made an incorrect prediction for the sample in the $t+1$-th epoch, despite having made a correct prediction for the sample in the t-th epoch [39], i.e., $y_i = f_i^t$ and $y_i \neq f_i^{t+1}$ with f_i^t representing the prediction of sample x_i made by the model at the t-th epoch. The forgetting counts during the whole training process can be formulated as,

$$FC = \sum_{t=1}^{\mathbf{T}} \mathrm{I}(y_i = f_i^t \ \& \ y_i \neq f_i^{t+1}). \tag{22}$$

Forgetting is a sort of variation where the prediction acutely becomes incorrect at epoch $t+1$, and the forgetting count FC represents the frequency of such variations for sample x_i throughout the entire training process. As stated in [64], the forgetting phenomenon tends to occur more frequently on challenging samples, such as noisy samples. Thus, by analyzing the variations related to the forgetting phenomenon, it can be used to identify hard samples.

- VoG: It is the first framework that evaluates samples' learning difficulty using the variance of gradients, i.e.,

$$VoG_i = \mathbb{E}_{p\in\{1,\cdots,\mathbf{P}\}}[\mathbb{E}_{k\in\{1,\cdots,\mathbf{K}\}}[\mathbf{g}_{i,p}^k - \overline{\mathbf{g}}_{i,p}]], \tag{23}$$

where $\mathbf{P}$, $\mathbf{K}$, and $\mathbf{g}_{i,p}^k$ denote the number of pixels, number of checkpoints, and the gradient of the p-th pixel in sample x_i at the k-th checkpoint, respectively; $\overline{\mathbf{g}}_{i,p}$ denotes the average of $\mathbf{g}_{i,p}^k$ over $\mathbf{K}$ checkpoints.

In summary, O2UNet considers bias to capture noisy samples, while forgetting and VoG characterize samples based on their variation or variance. Both bias and variance have been proven effective in characterizing samples in existing studies. GELD combines both bias and variance as a measure of learning difficulty to characterize the samples.

4 RATIONAL ANALYSIS OF THEORETICAL DEFINITIONS

In this section, we verify theoretically the rationality of proposed definition for learning difficulty of samples by conducting an analysis under weighting strategies, given that the weighting strategies in machine learning are mainly based on learning difficulties, such as Adaboost [24], SPL [29, 74], and Focal loss [48]. Under our definition of learning difficulty, above-mentioned weighting strategies can be better rationalized and further comprehended. First, the weighted generalization error[8] is defined as follows:

$$\begin{aligned} Err^{\mathrm{w}}(\lambda_h) &= \sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X} \omega(x)Err(x,\lambda_h)p(x|y)\mathrm{d}x \\ &= \sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X} \omega(x)BiasT(x,\lambda_h)p(x|y)\mathrm{d}x \\ &+ \sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X} \omega(x)VarT(x,\lambda_h)p(x|y)\mathrm{d}x \\ &+ \delta_e^{'}, \end{aligned} \tag{24}$$

where the non-negative weighting function $\omega(x)$ is defined over the entire sample space Ω, and $\delta_e^{'}$ is the irreducible noise. Under various cases, propositions according to Eq. (24) are discussed and demonstrated. Subsequently, above-mentioned typical weighting strategies are well-explained from the angle of difficulty-based weighting in order to prove the rationality of proposed theoretical definition for learning difficulty of samples.

4.1 Propositions

We discuss influence to optimal model complexity brought by different learning schemes applied separately on samples. An underlying case is firstly discussed in Proposition 1.

Proposition 1. *If $\omega(x)$ in Eq. (24) is a constant value, then c^* remains unchanged.*

The proof is simple and omitted.

[8]When generalization error is defined over the entire sample space, regard $Err(\lambda_h)$ as $Err(\Omega,\lambda_h)$ for simplicity.

Proposition 2. *Consider a sample region $\Omega^r \subset \Omega$ in which the value of $\mathcal{LDC}$ for each sample in Ω^r is larger than one. If a constant weight ω larger than one is placed on each sample of Ω^r and the weights of other samples in Ω remain one, then the new optimal model complexity will become larger.*

The proof is contained in Section B.2. in Appendix.

This proposition is in accordance with the idea that when the weights of hard samples are increased, the learned model will become more complex than the original model. Based on this proposition, we could extend a corollary more generalized.

Corollary 1. *Consider a sample region $\Omega^r \subset \Omega$ whose learning difficulty coefficient $\mathcal{LDC}$ for each sample in Ω^r is larger than one. If a weight larger than the original weight is placed on each sample in Ω^r, and the weights on other regions remain unchanged, the new optimal complexity will become larger.*

The proof is contained in Section B.3. in Appendix.

Proposition 3. *Consider a sample region $\Omega^r \subset \Omega$ whose value of $\mathcal{LDC}$ for each sample in Ω^r is smaller than one. If a constant weight larger than one is placed on each sample of Ω^r, and the weights of other samples in Ω remain one, then the new optimal model complexity c'^* will become smaller.*

The proof is similar to that for Proposition 3 and Corollary 1 and omitted. In numerous weighting strategies, the rationale is to modify the contributions of easy, medium, and hard samples. Therefore, the following propositions are presented.

Proposition 4. *Assume that the original weight of each sample is $\omega_0(x)$. Let $\omega(x) = u(\mathcal{LD}(x))$ be a new weighting function for a sample x. If u is non-decreasing and satisfies that $0 \leq \min \omega(x) < \max \omega(x)$, then the new optimal complexity is larger than the original optimal complexity.*

The proof is contained in Section B.4. in Appendix.

Corollary 2. *Let Ω^e, Ω^m, and Ω^h be a trichotomy for the whole space Ω, and they represent the regions for easy, medium, and hard samples, respectively. Let $\omega(\cdot,\cdot)$ be a region weighting function over the three data regions, and assume that the weights in each region are identical for each sample. Note $x \in \Omega^e; x' \in \Omega^m; x'' \in \Omega^h$. If $\omega(x) \leq \omega(x') \leq \omega(x'')$ and $\omega(x) < \omega(x'')$ hold, then the new optimal complexity will become larger.*

The proof is simple and omitted.

Proposition 5. *Assume that the original weight of each sample is $\omega^0(x)$. Let $\omega(x) = u(\mathcal{LD}(x))$ be a weighting function for a sample x. If u is non-increasing and satisfies that $0 \leq \min \omega(x) < \max \omega(x)$, then the new optimal complexity is smaller than the original optimal complexity.*

Corollary 3. *Let Ω^e, Ω^m, and Ω^h be a trichotomy for the whole region Ω and they are the regions for easy, medium, and hard samples, respectively. Let $\omega(\cdot,\cdot)$ be a region weighting function over the three data regions, and assume that the weights in each region are identical for each sample. Note $x \in \Omega^e; x' \in \Omega^m; x'' \in \Omega^h$. If $\omega(x) \geq \omega(x') \geq \omega(x'')$ and $\omega(x) > \omega(x'')$ hold, then the new optimal complexity will become smaller.*

Propositions 1-5 and the associated corollaries are about the weighting on generalization errors and also the losses. They establish a theoretical framework for the analysis of the learning difficulty-aware weighting strategies in learning.

4.2 Explanations Under Classical Weighting Methods

We rationalizes several typical learning methods which assign weights on samples based on learning difficulties[9].

4.2.1 *Adaboost.*

Adaboost is a classical ensemble learning algorithm. In each epoch, it learns a new model based on the updated weights on samples defined as follows:

$$\omega_i^t = \frac{\omega_i^{t-1}}{z^{t-1}} exp(-\alpha y_i f^{t-1}(x_i)), \tag{25}$$

where ω_i^t is the weight in the t^{th} epoch, z^{t-1} is a normalized factor, f^{t-1} is the learned weak classifier in the $(t-1)^{th}$ epoch, and α is a positive weight for f^{t-1}. According to Eq. (25), if x_i is mis-predicted by f^{t-1}, then the weight of x_i will become larger in the next epoch. If x_i is correctly predicted by f^{t-1}, then the weight of x_i will become smaller in the next epoch. In essence[10], the weight in Eq. (25) can be written as follows:

$$\omega_i^t = \omega_i^{t-1} u(\mathcal{LD}(x_i)), \tag{26}$$

where

$$u(\mathcal{LD}(x_i)) = \begin{cases} \dfrac{exp(\alpha)}{z^{t-1}} & \text{if } \mathcal{LDC}(x_i) > 1 \\ \dfrac{exp(-\alpha)}{z^{t-1}} & \text{otherwise} \end{cases} \tag{27}$$

Obviously, $u(\mathcal{LD}(x_i))$ is an increasing function over learning difficulty. According to Proposition 4, the new model complexity becomes larger than the original one. Specially, the learned new classifier f^t is more complex than that in the $(t-1)^{th}$ epoch if the learner is not as simple as the decision trump. Therefore, the new classifier and the whole ensemble classifier become more complex with the increase in epoch.

Two aspects determine the complexity of the final ensemble model greatly:

- Power of the basic model. If the basic model is a strong classifier, such as SVM, the learned model will become highly complex with the increase in epoch and overfitting is inevitable. A weak learner can avoid this situation.
- Number of maximum epochs. If the maximum epoch is large, then the ensemble model in the last few epochs will become highly complex when noises exist. Accordingly, overfitting may occur.

A natural improvement is that high weights above a threshold are restricted. This condition makes the model less complex. A famous modification with solid theoretical basis is soft margin boosting [80]. The weight is calculated as follows:

$$\omega_i^t = \frac{\omega_i^{t-1}}{z^{t-1}} exp(-\alpha y_i f^{t-1}(x_i) - C\zeta_i^{t-1}|b^{t-1}|), \tag{28}$$

where $C(\geq 0)$ is a hyper-parameter, ζ_i^{t-1} is the average weight of the i^{th} sample up until the $(t-1)^{th}$ iteration, and b^{t-1} is a factor that reflects the classification performance in the $(t-1)^{th}$ iteration. If $C > 0$, then the above weight is smaller than the weight in Eq. (25) when the sample is often misclassified up until the $(t-1)^{th}$ iteration, and vice versa. Based on Corollary 1, on the contrary, the optimal complexity of the learned model based on the above weighting scheme will be smaller than that of the model based on Eq. (25).

[9]It should be noted that the difficulty measures in these methods are not equal to our proposed theoretical measure. Nevertheless, we assume that their employed measures are in accordance with ours in their contexts to facilitate further theoretical investigation.
[10]Indeed, the $\mathcal{LDC}(x_i)$ can be seen as being approximated by $exp(y_i f(x_i))$ in Adaboost.

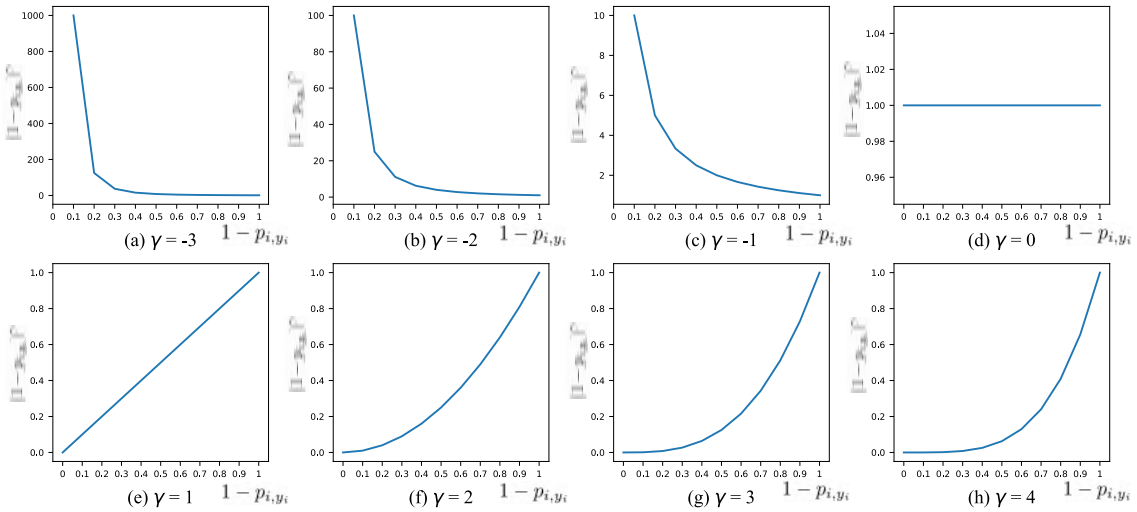


Fig. 6. Curves of Focal loss with different values of γ.

4.2.2 *SPL.*

SPL trains models from easy samples and adds hard samples with the increasing training epoch. Its objective function is as follows:

$$\min_{\Theta, v_i \in \{0,1\}} \sum_i v_i l_i - \lambda v_i, \tag{29}$$

where Θ is the model parameter, v_i is the sample weight, and $\lambda > 0$ is a hyper-parameter and increased with the epoch. Theoretically[11], the weight in SPL is defined as follows

$$\omega_i = \begin{cases} 1 & \mathcal{LD}(x_i) \leq \lambda \\ \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

In each new epoch, the weights of some hard samples are changed from zero to one with the increase in λ. According to the Corollary 1, the optimal model complexity will become larger. Alternatively, SPL obtains simple models in the initial epochs and gradually yields complex models.

4.2.3 *Focal loss.*

Focal loss assigns each sample a weight as follows:

$$\omega_i = (1 - p_{i,y_i})^\gamma, \tag{31}$$

where p_{i,y_i} is the estimated SoftMax value of x_i on the ground-truth label in the current model, and γ is positive. The motivation of Focal loss is to exert (relatively) larger weights on hard samples than simple ones. Focal loss utilizes the value of $1 - p_{i,y_i}$ as an indicator of learning difficulty. To better understand Focal loss, we first theoretically define a

[11] Indeed, the $\mathcal{LD}(x_i)$ can be seen as being approximated by l_i in SPL.

989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040

weight:

$$\omega_i = \left(\frac{\mathcal{LD}(x_i)}{\max \mathcal{LD}(x_i)} \right)^{\gamma}. \tag{32}$$

According to Proposition 4 and Corollary 2, the new optimal complexity will be increased. We further obtain the following conclusion:

Corollary 4. *The larger the value of γ, the larger the optimal complexity will be, i.e., $\forall \gamma_1 < \gamma_2, c^*(\gamma_1) < c^*(\gamma_2)$.*

The proof is similar to that for Proposition 3. Alternately, γ controls the model complexity. Consequently, if γ is quite large, then the learned model will be quite complex which affects the generalization capability of the model. If γ is smaller than zero, then the learned model will be simpler than the learned model when no weights are used (i.e., $\gamma = 0$). Fig. 6 shows the curves of Focal loss when γ is searched in $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. Corollary 4 is also supported by the empirical observations [11] shown in Fig. 7. A small(large) γ will result in under-fitting(over-fitting).

Focal loss is actually the strategy that uses the weight in Eq. (31) to approximate the weight defined in Eq. (32):

$$\left(\frac{\mathcal{LD}(x_i)}{\mathcal{LD}_{max}} \right)^{\gamma} \approx (1 - p_{i,y_i})^{\gamma}. \tag{33}$$

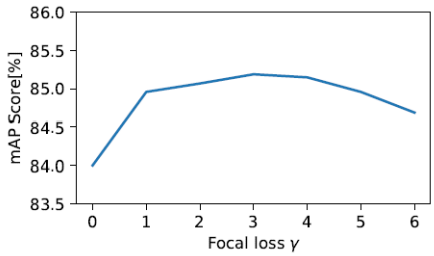


Fig. 7. Detection performance with the variations of γ [11].

Remark 1. *The proposed propositions and corollaries in Section 4.1 are in accordance with intuitions on the model complexity variations when applying weighting for learning. The explanations for the three classical methods are also reasonable and partially supported by empirical observations as shown in Fig. 7. The above analysis and explanations support the rationale of our proposed learning difficulty theory.*

5 PRACTICAL EXPERIMENTS OF PROPOSED MEASURE

This section demonstrates empirically the effectiveness of proposed practical measure under majority situations. In general, the proposed measure is compared with competitive methods in terms of precision, learning speed, and classification accuracy.

As previously mentioned, learning difficulty is heavily affected by the data quality, sample margin, uncertainty, and category distribution. Therefore, four different scenarios are designed to evaluate the precision in detection.

5.1 Measurement under Noise Detection

Two benchmark image classification data sets [41], namely, CIFAR10 and CIFAR100 are used. There are 10 classes in CIFAR10 and 100 classes in CIFAR100. On both sets, there are 50,000 images for training and 10,000 images for testing.

The test images are used as the validation data for CIFAR10 and CIFAR100. In this scenario, noises contain two types. The first type is noisy labels (y), while the second consists of noisy images (x). The competing methods are as follows:

- **Loss**. The losses for each sample in the epoch with the highest validation accuracy are used for measurement.
- **Average loss (AveLoss)**. The average values of losses of the last 100 epochs are used for measurement.
- **O2UNet [37]**. As previously introduced, this method adopts a cyclical training procedure and the average loss of each sample in the procedure is used.
- **MentorNet [38]**. This method pertaining to curriculum learning, uses the output weights of the teacher network with the highest accuracy on the validation set to present the possibility of being correct. A smaller weight indicates the sample is more difficult to learn.
- **Co-teaching [31]**. Two networks are trained. For each sample, the smaller one of the two losses given by the two networks is regarded as the learning difficulty.
- **Variance of Gradients (VoG) [2]**. This method relies on calculating the variances of the gradient norms for each sample across different training epochs. A high VoG value indicates that a sample has a higher level of difficulty.
- **Variance of Gradient for Curriculum Learning (VoG-CL) [82]**: VoG-CL, derived from VoG, incorporates the concept of curriculum learning. By utilizing multiple networks, VoG-CL calculates the averaged variance of the gradient to assess the learning difficulty of samples.
- **Our proposed method GELD**. The detailed steps are presented in Algorithm 1.

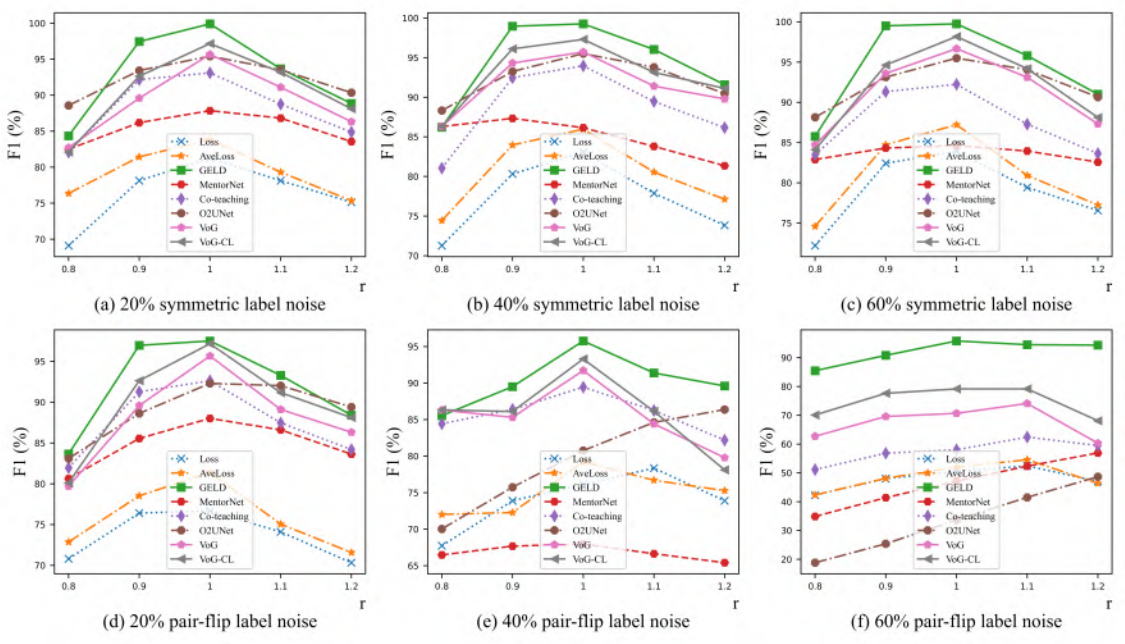


Fig. 8. The F1 scores (%) of the competing methods on CIFAR10 under different sub-types and rates of label noises.

In the methods of Loss, AveLoss, O2UNet, VoG, VoG-CL, and our GELD, ResNet-34 [33] is used as the base network. The hyper-parameters of ResNet-34 used in [33] are followed. Specifically, the batch-size is 128, the SGD optimizer

has a momentum of 0.9, and the weight decay is 1e-4. The learning rate of the first 40 epochs is 0.1 and is multiplied by 0.1 for every 40 epochs. Each model is learned for 200 epochs. The default settings of MentorNet and Co-teaching are borrowed from the corresponding papers [31, 38]. O2UNet, VoG, and VoG-CL contain specific hyper-parameters other than those of ResNet-34. These parameters follow the setting in the original paper [2, 37, 82]. In our GELD, K and M are set as 5 and 6 respectively, for GELD, except for the part E (the discussion of the impact of (K, M) value). The tuning factor μ of GELD is set as 1.

Let v be the noise rate. The result evaluation scheme used for O2UNet [37] is followed. In each method, the top-$50000 * v * r$ samples are selected as its detected noisy samples according to its estimated difficulties, where $r \in \{0.8, 0.9, 1, 1.1, 1.2\}$. Then, the whole detection is repeated three times for each method and the average F1 values on the detection results are calculated and compared. A high F1 value indicates a good performance in noisy label detection and thus the learning difficulty measurement.

5.1.1 *Noisy Label Detection.*

Two sub-types of noises are used, namely, symmetric and pair-flip. The symmetric noise describes mislabeling to each other classes of equal possibility. In pair-flip, labelers may make mistakes only within very similar class. The noise rate is set as 20%, 40%, and 60%, respectively. The detailed noise setting in [31] is followed.

Figs. 8 and 9 show the detection performances of the competing methods on CIFAR10 and CIFAR100, respectively. Our proposed approach GELD achieves the highest F1 values in most cases. Although O2UNet outperforms GELD under the symmetric noise sub-type on CIFAR100 (Figs. 9(a), (b), and (c)), its performances are quite poor under the pair-flip noises. The performance of the widely-used method Loss is poor and it achieves the worst F1 scores in several cases. Loss is not an ideal measurement for the easy and hard samples even though it does not require additional computational cost. VoG and VoG-CL exhibit superiority compared with other comparison methods, but our method GELD consistently outperforms all comparison methods. These results indicate that variance is more relevant to data characteristics than bias on these datasets, as VoG, VoG-CL, and GELD all incorporate variance for data detection. However, it is worth noting that the variance term occasionally fails to characterize data, while our method GELD, which utilizes both bias and variance, proves to be effective under all cases.

In our approach GELD, μ can be tuned. Table 2 shows the performance variations of GELD under the pair-flip noise sub-type and different values of μ in Eq. (20). When the value is larger than one, higher F1 values are achieved. These results reveal the importance of the variance term during the evaluation of the learning difficulty.

Table 2. $F1$ scores (%) of GELD under various values of μ.

	μ	0.5	0.75	1	1.25	1.5	2
CIFAR10	20%	90.48	92.01	93.67	93.99	94.83	95.06
	40%	93.95	93.99	96.03	96.00	96.00	95.39
	60%	91.60	92.47	95.80	95.67	95.52	94.99
CIFAR100	20%	89.74	90.00	90.50	90.71	91.00	91.67
	40%	90.04	91.39	91.51	92.00	92.33	92.97
	60%	90.33	91.05	91.39	91.49	92.41	93.22

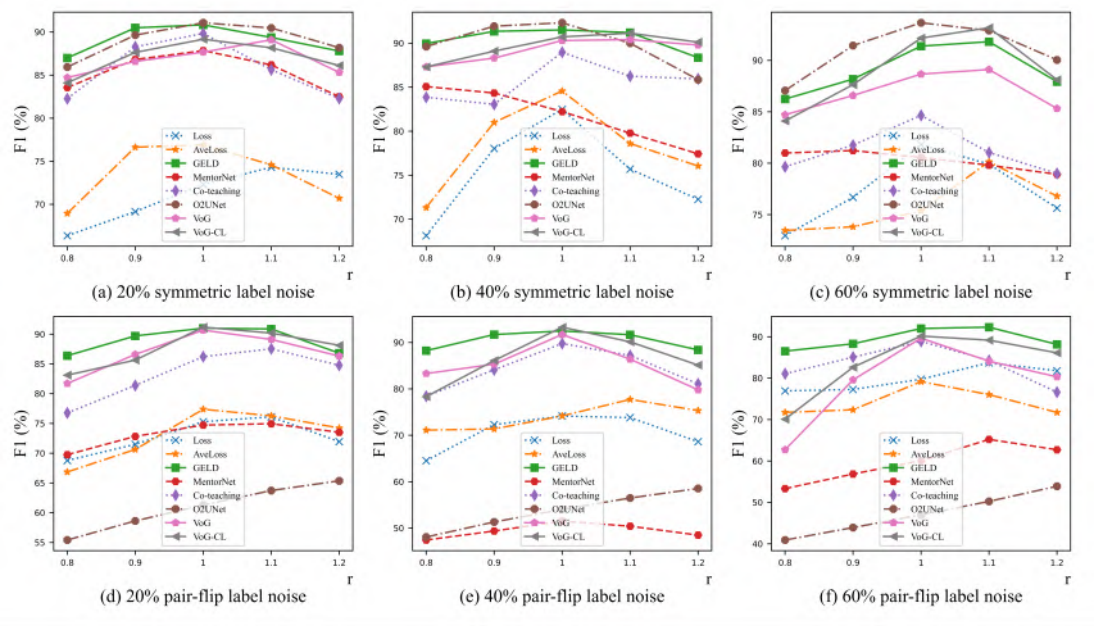


Fig. 9. The F1 scores (%) of the competing methods on CIFAR100 under different sub-types and rates of label noises.

5.1.2 *Noisy Image Detection.*

In this experiment, salt-and-pepper noises are leveraged [25]. The noise is simulated by adding white (salt) or black (pepper) noises into the original RGB images with a parameter of signal-to-noise ratio (SNR). In our experiment, the SNR is set as 0.4 for each image. Fig. 10 shows an example for nosiy images with different SNR levels. The noise rate on the whole data is set as 20% and 40%, respectively.

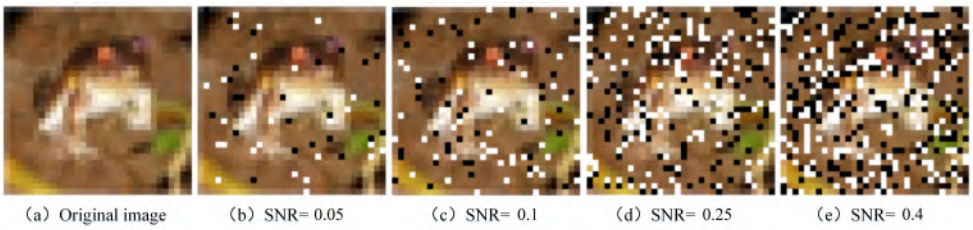


Fig. 10. Noisy images with different SNR levels.

Fig. 11 shows the performances of the competing methods on CIFAR10 and CIFAR100, respectively. The settings of competing methods remain unchanged compared with the label noise experiments. Our method GELD still achieves the highest F1 values under all the noise rates. Few noisy data were detected by methods that rely merely on the loss.

5.2 Measurement under Small-margin Data Detection

A sample with a small margin (the distance to the oracle decision boundary) is considered to be hard in learning [74]. This experiment evaluates a learning difficulty measure in terms of the detection of small-margin samples.

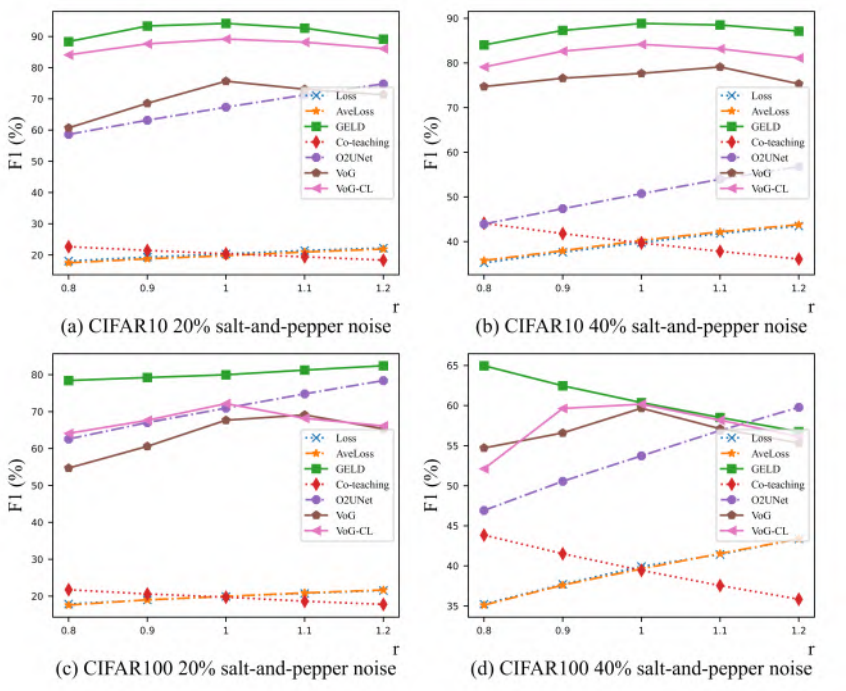


Fig. 11. The F1 scores (%) of the competing methods on CIFAR10 and CIFAR100 under different rates of salt-and-pepper noise.

In this experiment, four UCI [8] data sets are used, namely, Iris, Mammographic, Haberman, and Abalone. To better construct the ground-truth, only binary classification is considered. Only two categories are selected for both Iris (the “Setosa" and “Versicolour" categories) and Abalone (the “9" and “10" categories). Mammographic and Haberman contain only two categories. The details of used data in this experiment are presented in Table 3. The classical margin-based learning method SVM [16] is used to construct the ground truth, i.e., the small-margin samples. Specifically, the SVM with RBF kernel is used. Two parameters C and g are searched in $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ and $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$, respectively, via five-fold cross-validation. The optimal parameter setting is used and the SVM is trained on the whole training set. Constantly, the margin of each sample is calculated as the ground-truth difficulty. The margin is $yf(x)$ for the sample x, where y is the label and $f(x)$ is the output of the kernel SVM. Let N be the #Instances. The top-$N * v$ samples with small margins are selected as the ground-truth samples to detect.

As the base network ResNet-34 is inappropriate in this experiment, a three-layer perception with the Sigmoid activation function is used as the base network. The number of epoch is set as 10000. Its hyper-parameters are also pursed via five-fold cross-validation. Considering that MentorNet and Co-teaching are quite complex for this scenario, they are not compared in this experiment. The competing methods include Loss, AveLoss, O2UNet, VoG, VoG-CL, and our GLED. The evaluation criteria and the whole calculate scheme follow the setting in the previous experiments. The value of v is set as 20%, 40%, and 60%, respectively; r is set as one.

The results are shown in Fig. 12. Our approach GELD achieves the highest F1 values on all data sets. In addition, GELD is stable across different vs and different data sets. By contrast, the other methods are not stable.

Table 3. Details of the four UCI data sets.

Data set	#Dimensions	#Classes	#Instances
Iris	4	2	100
Mammographic	5	2	961
Haberman	3	2	360
Abalone	8	2	1323

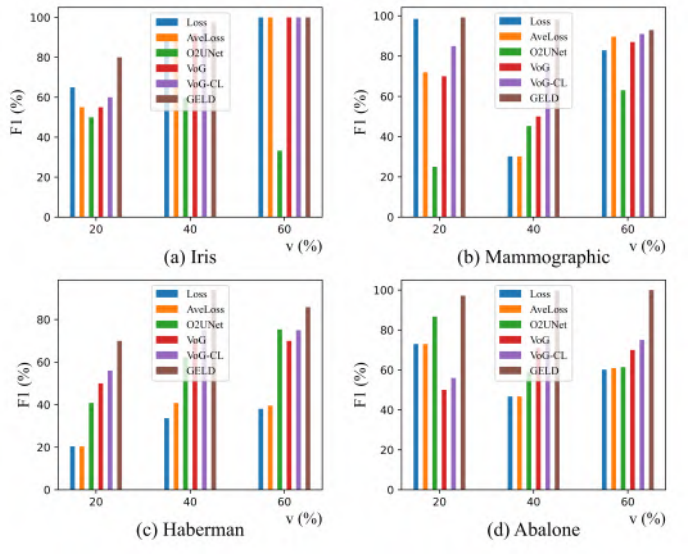


Fig. 12. Histograms of F1 scores (%) under various data sets using margin as ground-truth.

5.3 Measurement under Epistemic Uncertainty Detection

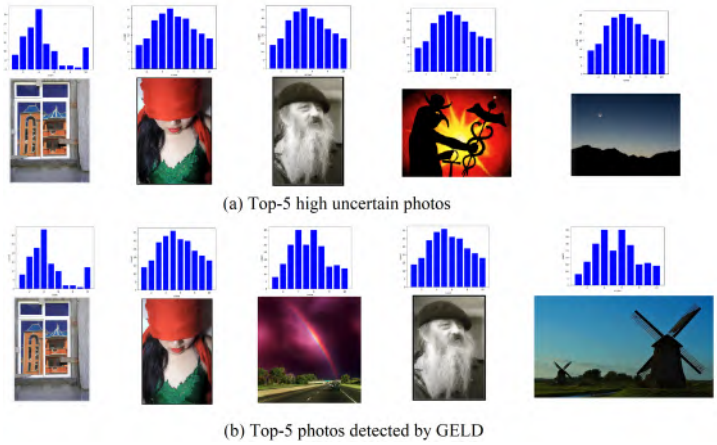
In this experiment, a data corpus containing epistemic uncertainty is required. An image aesthetic assessment data corpus, namely, AVA benchmark image aesthetic data corpus [54], is then utilized and each photo receives multiple rating scores from multiple different users. The variance of user rating scores of a photo reflects the epistemic uncertainty on the photo using each user as a gold model. A large variance indicates a large uncertainty for a photo. Specifically, the former 50,000 samples in the AVA corpus are downloaded and each sample receives 210 rates on average counting from 1 to 10. We use ResNet-101 as the base network as the image quality of AVA is much higher than CIFAR10 and CIFAR100. Given that the Bayesian Neural Network (BNN) [73] is particularly designed for modeling uncertainty, we import BNN as one of the competing methods, and its outputting confidence coefficient is used to detect uncertain photos. Therefore, the competing methods include Loss, AveLoss, O2UNet, BNN, VoG, VoG-CL, and our GELD.

The top-$50,000 * v$ photos with high uncertain scores are taken as the objective samples to detect. The v values are set as 20%, 40%, and 60%. The r value is set as one. In the BNN method, the dropout strategy described in [73] is followed. The parameter setting of ResNet-101 reported in [33] is adopted. The photos are resized into shape of (3, 192, 192) to fit in the ResNet-101.

Table 4. $F1$ scores (%) using the variance of scores as standard

Method	Loss	AveLoss	O2UNet	BNN	VoG	VoG-CL	GELD
20%	14.64	54.05	33.27	62.71	61.09	62.45	**63.96**
40%	28.38	33.33	35.67	54.30	71.99	73.08	**78.39**
60%	30.23	45.79	38.20	78.21	76.42	78.69	**79.11**

Table 4 shows the F1 scores of the competing methods in high uncertain sample detection. GELD still performs the best and slightly outperforms BNN. The rest of the three methods poorly perform in this detection task. Fig. 13 shows the top-5 high uncertain photos and the top-5 photos detected by our GELD approach. The aforementioned figure (Fig. 14) also shows the last five photos with small uncertainty and the last five photos detected by our GELD.



(a) Top-5 high uncertain photos

(b) Top-5 photos detected by GELD

Fig. 13. True highest uncertain photos and our detection results. The histograms of user ratings are also presented.

5.4 Measurement for Imbalance Data

The long-tail versions of the CIFAR10 and CIFAR100 are used in this experiment. Buda et al. [13] compiled a series of data sets under different imbalance ratios. The two data sets under the 20 : 1 ratio for CIFAR10 and CIFAR100 are used. There is no ground-truth information for the learning difficulties because the head categories can also contain difficult samples. Consequently, we only plot the histograms of the numbers of hard samples detected by the competing methods in the head and tail categories. The competing methods are Loss, AvgLoss, O2UNet, VoG, and our proposed GLED. The base network and the concerning setting in part A are followed.

The top-40% samples detected by each competing methods are regarded as their detected hard samples. Figs. 15 and 16 show the histograms of the detected hard samples by each method on CIFAR10 and CIFAR100 (the top five head categories and the last five tail categories), respectively. All the competing methods identically behave on the tail categories on both data corpora. This condition is reasonable and accords well with the primary motivation for imbalance learning that samples in tail categories are hard to learn. There are slight differences between our GELD and other competing methods on the head categories. The numbers of hard samples detected by GELD are larger than those of other methods, which is reasonable because head categories still contain hard samples.

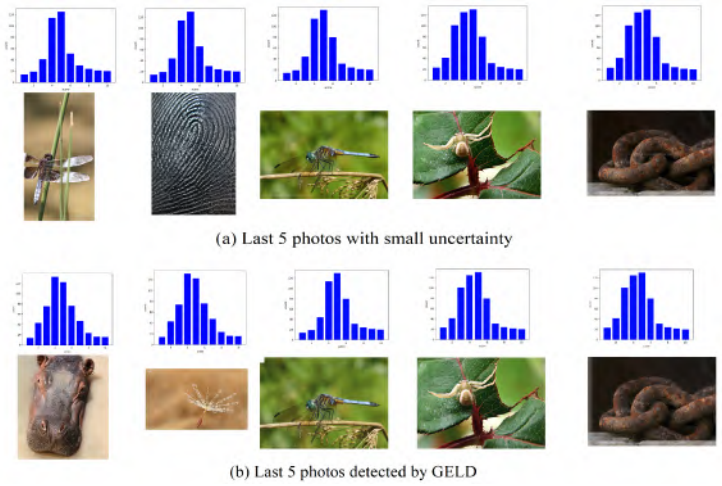


Fig. 14. True lowest uncertain photos and our detection results. The histograms of user ratings are also presented.

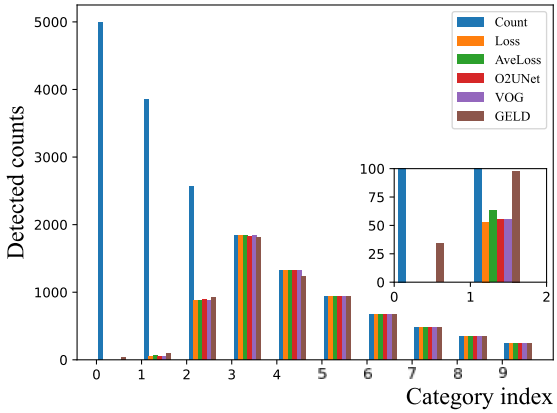


Fig. 15. Histogram of CIFAR10-LT top-40% detected hard samples.

5.5 Discussion

The above experiments on the four scenarios, namely, noise detection, small-margin sample detection, uncertain sample detection, and hard sample detection in imbalance learning, verify the superiority of the proposed GELD approach over existing classical and state-of-the-art methods. As previously introduced, the primary difference between GELD and many existing methods lies in that GELD explicitly considers variances. Fig. 17 shows the histograms of the bias values and the variance values achieved by GELD on clean samples and label noisy samples under both pair-flip and symmetric noise types on CIFAR10. The variance values between clean and noisy samples are also considerably distinct as shown in Figs. 17 (b) and (d), which demonstrates the usefulness of the variance term utilized in our GELD method. In addition, the difference consist in histograms of noisy and clean samples under the pair-flip noises is trivial, which rationalises the poor performances of the loss-based methods such as O2UNet.

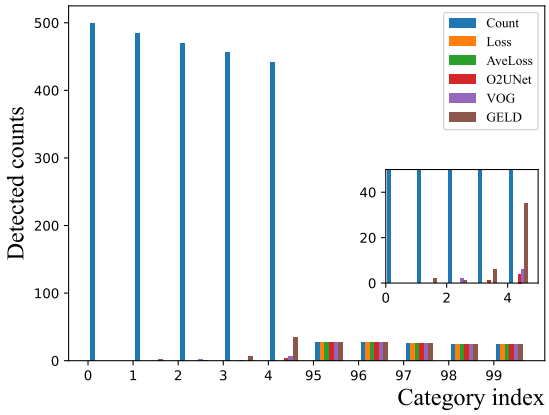


Fig. 16. Histogram of CIFAR100-LT top-40% detected hard samples.

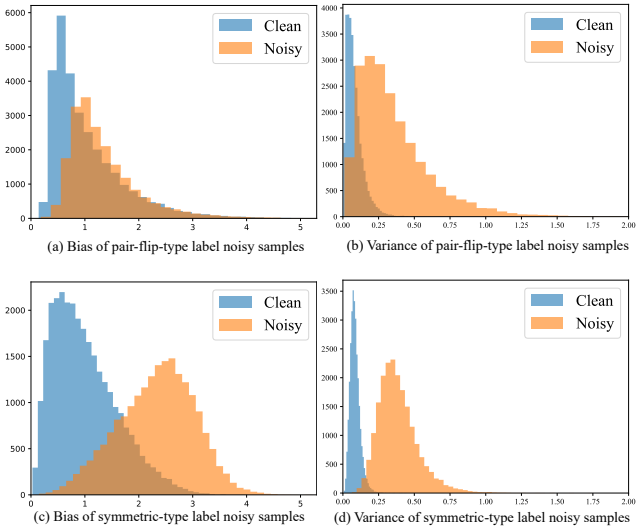


Fig. 17. Histograms of bias and variance values calculated by GELD under different types of label noise ($v = 40\%$).

Although good results are achieved, GELD is only an appropriation for theoretical difficulty in Definition 1. We evaluate the robustness of the method in terms of the variations on the two key parameters, namely, K and M. Table 5 shows the performances of pair-flip noise detection ($r = 1$) on CIFAR10 and CIFAR100. The results show that the performances of GELD are stable when the value of (K, M) is set in $\{(4, 5), (5, 6), (5, 8), (5, 10)\}$.

We are interested in whether a simple model can also obtain better results. A simple network, namely, AlexNet [42], is used as the basic leaner in pair-flip noisy label. The base network and setting of other methods follow the previous setting of the corresponding experiments in part A. The training setting of AlexNet in [55] is followed, and the output-size is modified into 100 while training CIAFR100. The results are shown in Table 6. GELD (AlexNet) is inferior to GELD (ResNet-34). However, it is comparable to Co-teaching and outperforms the rest of the methods.

Table 5. $F1$ scores (%) with various values of $K * M$.

	$K * M$	20 (4×5)	30 (5 × 6)	40 (5 × 8)	50 (5 × 10)
CIFAR10	20%	96.83	97.51	96.91	97.04
	40%	94.12	95.77	96.10	96.34
	60%	94.81	95.82	94.95	95.34
CIFAR100	20%	90.36	91.00	92.67	91.46
	40%	91.78	92.48	91.86	91.04
	60%	90.02	92.00	91.17	90.89

Table 6. $F1$ scores (%) of the competing methods plus GELD using AlexNet as base model.

Method	CIFAR10			CIFAR100		
Label noise rate	20%	40%	60%	20%	40%	60%
GELD (ResNet-34)	97.51	95.77	95.82	91.00	92.48	92.00
O2UNet	92.31	80.76	33.69	61.29	54.03	46.98
Co-teaching	92.63	89.45	58.02	86.22	89.77	88.90
MentorNet	88.02	67.96	47.14	73.00	51.49	60.09
AveLoss	81.39	79.23	51.98	77.38	74.18	79.17
Loss	76.69	76.12	50.53	75.29	74.15	79.75
GELD (AlexNet)	82.58	88.77	83.54	82.96	86.51	84.39

Table 7. Time cost of different complex methods.

Method	Time cost (hours)
O2UNet	7.95
GELD (1 GPU)	11.11
GELD (2 GPUs)	5.67
GELD (3 GPUs)	3.71
GELD (4 GPUs)	2.83

A large real-world data set, namely, Clothing1M [41], is used to further evaluate the performance of our GELD measure in terms of image classification. There are 14 classes in Clothing1M containing 1,000,000 training images with real noisy labels, 48,000 training samples verified to be clean, and 10,000 testing images. The 48,000 clean training samples are used as the validation data for Clothing1M. Clothing1M has a noise proportion of 38% approximately. ResNet-101 is used and the settings of the network in [37] are employed. (K, M) are set as $(50, 50)$. Each model is learned for 50 epochs for GELD. The model is selected using the 48,000 clean training samples. The learning rate remains constant as $1e-6$. The batch size is set as 16 during the GELD calculation. Other hyper-parameters follow the settings in [37]. The top 10% samples with the highest $Err(x_i, \lambda_h)$ values are removed as detected noisy samples. Remaining samples are used to learn the final image classifier. The batch size is set as 128 and the maximum epoch is

Table 8. Classification accuracy (%) on Clothing1M

MentorNet	Co-teaching	O2UNet	GELD
79.30	78.52	82.38	**82.94**

set as 10 during this procedure. The settings of other methods in [37] are followed. Table 8 shows the comparison of classification accuracies (%) among the four competing methods. GELD outperforms the other three methods. Results of methods beside GELD are directly from the O2UNet study [37].

The computational cost of GELD is relatively high as $K * M$ models should be trained. However, it is still smaller than another SOTA method O2U-Net. Moreover, several ways can significantly reduce the complexity. First, the task can be performed in parallel. Four NVIDIA GeForce RTX 3090 GPUs are used in our experiments. The average time costs for our GELD using different number of GPUs on CIFAR10 using ResNet-34 as base network are shown in Table 7. The settings follow the pair-flip label noise experiments' in part A with (K, M) is set as (5, 6). The time consumption is considerably reduced when GELD is run on more GPUs in parallel. The time cost of O2UNet is also large. More over, O2UNet cannot be performed in parallel. Second, a relatively small training set instead of the entire can be used when dealing with large corpora. Third, a dropout-based strategy (like the quantifying uncertainties in BNN) can also reduce the time cost. To sum up, the time complexity of GELD does not hinder its applications based on these strategies.

Although four key factors, namely, data quality, sample margin, uncertainty, and category distribution, are summarized and the proposed method achieves quite competing performance, a directly theoretical connection between the four factors and the learning difficulty is not established in this study. We leave this theoretical investigation as our future work.

6 CONCLUSION

This study has conducted a comprehensive investigation on learning difficulty of data in machine learning. We gave a first summary of influential factors for learning difficulty. Correlations between generalization error, model complexity, and influential factors are surveyed and analyzed. Then we established a theoretical definition of learning difficulties of data based on generalization error and model complexity. The well discussed and explored concepts, easy and hard samples, are formally described based on the theoretical definition and the associated difficulty coefficients. A practical measure, namely, the generalization error-based learning difficulty (GELD) measurement, is then proposed by importing the typical bias-variance trade-off theory to calculate the learning difficulty of each training sample. Our proposed measure is the first measure which incorporates both the bias and the variance to characterize samples. Finally, the properties of the weighted learning strategy are presented and three classical methods are explained on the basis of the theoretical formalization. Extensive experiments validate the effectiveness of our proposed measure, which outperforms existing state-of-the-art methods under different scenarios considering concluded influential factors. Learning speed and classification accuracy are also explored and GELD shows its outstanding performance.

This study conducts an attempt to establish a theory for learning difficulty of samples. Our future work aims to reveal the mathematical correlations between the theoretical definition (i.e., optimal model complexity) and the measure (i.e., generalization error) for learning difficulty.

REFERENCES

[1] Abdi, L., and Hashemi, S., To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Transactions on Knowledge and Data Engineering*, *28*(1), 238-251. https://doi.org/10.1109/TKDE.2015.2458858

[2] Agarwal, C., D'souza D., and Hooker, S., (2022). Estimating example difficulty using variance of gradients. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10358-10368. https://doi: 10.1109/CVPR52688.2022.01012.

[3] Aguilar-Torres, E., Nagarajan, B., Khatun, R., Bolaños, M., and Radeva, P., (2021). Uncertainty modeling and deep learning applied to food image analysis. *Biomedical Engineering Systems and Technologies*, 3-16. https://doi.org/10.5220/0009429400090016

[4] Almeida, M., Zhuang, Y., Ding, W., and Crouter, S.E., (2021). Mitigating class-boundary label uncertainty to reduce both model bias and variance. *Journal of ACM Trans. Knowl. Discov.Data*, *15*(2), 1-18. https://doi.org/10.1145/3429447

[5] Arkesteijn, L., and Pande, S., (2013). On hydrological model complexity, its geometrical interpretations and prediction uncertainty. *Water Resources Research*. https://doi.org/10.1002/wrcr.20529.

[6] Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S, Maharaj, T., Fischer, A., Courville, A.C., Bengio, Y., and Lacoste-Julien, S., (2017). A closer look at memorization in deep network. *Proceedings of the 34th International Conference on Machine Learning, PMLR*, 70, 233-242. https://doi.org/10.48550/arXiv.1706.05394

[7] Arsomngern, P., Long, C., Suwajanakorn, S., and Nutanong, S., (2021). Self-supervised deep metric learning for pointsets. *IEEE International Conference on Data Engeneering*, 2171-2176. https://doi.org/10.1109/ICDE51399.2021.00219

[8] Astola, J., and Kuosmanen, P., (1997). *Fundamentals of nonlinear digital filtering*. CRC Press. ISBN 9780367448257

[9] Barron, A., Rissanen, J., and Yu, B., (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory* , *14*(6), 2743 - 2760. https://doi.org/10.1109/18.720554

[10] Bengio, Y., Louradour, J., Collobert, R., and Weston, J., (2009). Curriculum learning. *International Conference on Machine Learning*, 41-48.

[11] Ben-Baruch, E., Ridnik, T., Zamir, N., Noy, A., Friedman, I., Protter, M., and Zelnik-Manor, L., (2020). Asymmetric loss for multi-Label classification. *IEEE International Conference on Computer Vision*, 2020. https://doi.org/10.48550/arXiv.2009.14119

[12] Breiman, L., (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

[13] Buda, M., Maki, A., and Mazurowski, M., (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249-259. https://doi.org/10.1016/j.neunet.2018.07.011

[14] Castells, T., Weinzaepfel, and P., Revaud, J., (2020). SuperLoss: A generic loss for robust curriculum learning. *Conference on Neural Information Processing Systems*, 1–12.

[15] Chatterjee, S., and Zielinski, P., (2022). On the generalization mystery in deep learning. *arXiv*:2203.1003.

[16] Cortes, C., and Vapnik, V., (1993). Support-vector networks. *Machine Learning*. https://doi.org/10.1007/BF00994018

[17] David E. Rumelhart; James L. McClelland, "Learning Internal Representations by Error Propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations , MIT Press, 1987, pp.318-362.

[18] D'souza, D., Nussbaum, Z., Agarwal, C., and Hooker, S., (2021). A tale of two long tails. *IEEE Trans. Neural Netw. Learn. Syst.*, 1716-1721. https://doi.org/10.1109/TNNLS.2016.2546956

[19] P. M. Djuric, "Model selection by cross-validation," 1990 IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, LA, USA, 1990, pp. 2760-2763 vol.4.

[20] Ding, Y., Liu, J., Xiong, J., and Shi, Y., (2020). Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off. *IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPRW50498.2020.00010

[21] Duan, Y., and Wu, O., (2016). Learning with auxiliary less-noisy labels. *IEEE Trans. Neural Netw. Learn. Syst.*, 1716-1721. https://doi.org/10.1109/TNNLS.2016.2546956

[22] Dwivedi, R., Singh, C., Yu, B., and Wainwright, M.J., (2020). Revisiting complexity and the bias-variance tradeoff. *arXiv*2006.10189.

[23] Feng, J., Xu, P., Pu, S., Zhao, K., and Zhang, H., (2020). Robust visual tracking by embedding combination and weighted-gradient optimization. *Pattern Recognition*, *104*(107339). https://doi.org/10.1016/j.patcog.2020.107339

[24] Freund, Y., and Schapire, R.E., (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

[25] Fu, B., Zhao, X., Li, Y., Wang, X., and Reng, Y., (2018). A convolutional neural networks denoising approach for salt and pepper noise. *Multimedia Tools and Applications*, 1–18.

[26] Gautheron, L., Habrard, A., Morvant, E., and Sebban, M., (2020). Metric learning from imbalanced data with generalization guarantees. *Pattern Recognition Letters*, *133*, 298-304. https://doi.org/10.1016/j.patrec.2020.03.008.

[27] Ge, W., Huang, W., Dong, D., and Scott, M.R., (2018). Deep metric learning with hierarchical triplet loss. *European Conference on Computer Vision, Lecture Notes in Computer Science*, (11210), 272-288. https://doi.org/10.48550/arXiv.1810.06951

[28] Geman, S., Bienenstock, E., and Doursat, R., (1992). Neural networks and the bias/variance dilemma. *Neural Computation* 4(1), 1-58.

[29] Han, B., Tsang, I.W., Xiao, X., Chen, L., Fung, S.-F., and Yu, C.P., (2018). Privacy-preserving stochastic gradual learning. *IEEE Transactions on Knowledge and Data Engineering*, *33*(8), 3129-3140. https://doi.org/10.48550/arXiv.1810.00383

[30] Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I.W., Kwo, J.T., and Suigiyama, M., (2020). A survey of label-noise representation learning. *arXiv*:2011.04406, 2020.

[31] Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M., (2018). Co-teaching robust training of deep neural networks with extremely noisy labels. *International Conference on Neural Information Processing Systems*, 8536-8546. https://doi.org/10.48550/arXiv.1804.06872

[32] Hastie, T., Tibshirani, R., Friedman, J.H., (2009). The Elements of Statistical Learning. *ISBN 9780387848570*. https://doi.org/10.1007/978-0-387-21606-5

[33] He, K., Zhang, X., Ren, S., and Sun, J., (2016). Deep residual learning for image recognition. *IEEE conference on computer vision and pattern recognition*, 770–778. https://doi.org/10.1109/CVPR.2016.90

[34] Heskes, T., (1998). Bias-variance decompositions for likelihood-based estimatiors. *Neural Computation*, *10*(6), 1425-1433. https://doi.org/10.1162/089976698300017232

[35] T. Hospedales, A. Antoniou, P. Micaelli and A. Storkey, "Meta-Learning in Neural Networks: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 9, pp. 5149-5169, 1 Sept. 2022, doi: 10.1109/TPAMI.2021.3079209.

[36] Huang, H., and Yang, Q., (2020). Large scale analysis of generalization error in learning using margin based classification methods. *arXiv*:1901.08057.

[37] Huang, J., Qe, L., Jia, R., and Zhao, B., (2019). O2U-Net: A simple noisy label detection approach for deep neural networks. *IEEE International Conference on Computer Vision*, 3326-3334. https://doi.org/10.1109/ICCV.2019.00342

[38] Jiang, L., Zhou, Z., Leung, T., Li, L., and Li, F., (2018). Mentor-Net : learning data-driven curriculum for very deep neural networks with extremely noisy labels. *International Conference on Machine Learning*, 2309-2318. https://doi.org/10.48550/arXiv.1712.05055

[39] P. Kaushik, A. Gain, A. Kortylewski, and A.L. Yuille. " Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping." ArXiv, abs/2102.11343.

[40] Kendall, A., and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *International Conference on Neural Information Processing Systems*, 5580-5590. https://doi.org/10.48550/arXiv.1703.04977

[41] Krizhevsky, A., (2009). *Learning multiple layers of features from tiny images*. Technical report, University of Toronto.

[42] Krizhevsky, A., Hinton, G., and Sutskever, I., (2017). Image-net classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

[43] Krivosheev, E., Bykau, S., Casati, F., and Prabhakar, S., (2020). Detecting and preventing confused labels in crowdsourced data. *Proceedings of the VLDB Endowment*, *13*(12), 2522-253.

[44] Li, B., Liu, Y., and Wang, X., (2019). Gradient harmonized single-stage detector. *AAAI Conference on Artificial Intelligence*, 8577-8584. https://doi.org/10.48550/arXiv.1811.05181

[45] Li, P., Rao, X., Blase, J., Zhang, Y., Chu, X., and Zhang, C., (2021). CleanML: a study for evaluating the impact of data cleaning on ML classification tasks. *IEEE International Conference on Data Engeneering*, 13-24. https://doi.org/10.48550/arXiv.1904.09483

[46] Li, M., and Vitányi, P., (1997). An introduction to Kolmogorov complexity and its applications. *Springer*. https://doi.org/10.1007/978-3-030-11298-1

[47] Lin, J.Z., and Bradic, J., (2021). Learning to combat noisy labels via classification margins," *arXiv*:2102.00751.

[48] Lin, TY., Goyal, P., Girshick, R., He, K., and Dollar, P., (2017). Foacl loss for dense object detection. *IEEE International Conference on Computer Vision*, 2999–3007. https://doi.org/10.1109/ICCV.2017.324

[49] Liu, Z., Cao, W., Gao, Z., Bian, J., Chen, H., Chang, Y., and Liu, T., (2020). Self-paced ensemble for highly imbalanced massive data classification. *IEEE International Conference on Data Engeneering*, 841-852. https://doi.org/10.48550/arXiv.1909.03500

[50] Maalouf, A., Eini, G., Mussay, B., Feldman, D., and Osadchy, M., (2021). A unified approach to coreset learning. *arXiv*:2111.03044.

[51] Mangalam, K., and Prabhu, V., (2019). Do deep neural networks learn shallow learnable examples first? *International Conference on Machine Learning Workshop Deep Phenomena*, 16. https://doi.org/10.1016

[52] Mosley, and Lawrence, (2013). A balanced approach to the multi-class imbalance problem.

[53] Mou, L., Jia, R., Xu, Y., Li, G., Zhang, L., and Jin, Z., (2016). Distilling word embeddings: An encoding approach. *the 25th ACM International Conference on Information and Knowledge Management*. https://doi.org/10.1145/2983323.2983888

[54] Murray, N., Marchesotti, L., and Perronnin, F., (2012). Ava: A large-scale database for aesthetic visual analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, 2408–2415. https://doi.org/10.1109/CVPR.2012.6247954

[55] Muller, R., Kornblith, S., and Hinton, G., (2019). When does label smoothing help. *International Conference on Neural Information Processing Systems*, 4696-4705. https://doi.org/10.48550/arXiv.1906.02629

[56] Oinar, C., Le, B.M., and Woo, S.S., (2022). KappaFace: adaptive additive angular margin loss for deep face recognition. *arXiv*2201.07394.

[57] Pagliardini, M., Manunza, G.o, Jaggi, M., Jordan, M.I., and Thavdarova, T., (2022). Increasing the classification margin with uncertainty driven perturbations. *arXiv*:2202.05737.

[58] Salekshahrezaee, Z., Leevy, J.L., and Khoshgoftaar, T.M., (2021). A reconstruction error-based framework for label noise detection. *J. Big Data*, *8*(57). https://doi.org/10.1186/s40537-021-00447-5

[59] Santiago, C., Barata, C., Sasdelli, M., Carneiro, G., and Nascimento, J.C., (2021). LOW: training deep neural networks by learning optimal sample weights. *Pattern Recognition*, *110*(107585). https://doi.org/10.1016/j.patcog.2020.107585

[60] Schroff, F., Kalenichenko, D., and Philbin, J., (2015). Facenet: a unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 815-823. https://doi.org/10.1109/CVPR.2015.7298682

[61] Sharma, M., Yadav, A., Soman, S., and Jayadeva, (2019). Effect of various regularizers on model complexities of neural networks in presence of input noise. *arXiv*:1901.11458.

[62] Shin, W., Ha, J., Li, S., Cho, Y., Song, H., and Kwon, S., (2020). Which strategies matter for noisy label classification? insight into loss and uncertainty. *arXiv*:2008.06218.

[63] Su, J., Wen, Z., Lin, T., and Guan, Y., (2022). Learning disentangled behaviour patterns for wearable-based human activity recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*. https://doi.org/10.1145/3517252

[64] M. Toneva, A. Sordoni, R.T. Combes, A. Trischler, Y. Bengio, and G.J. Gordon. "An Empirical Study of Example Forgetting during Deep Neural Network Learning." ArXiv, abs/1812.05159.

[65] Valle-Perez, G., Camargo, C.Q., and Louis, A.A., (2019). Deep learning generalizes because the parameter-function map is biased towards simple functions. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1805.08522

[66] Van Rooyen, B., Menon, A., and Williamson, B., (2015). Learning with symmetric label noise: The importance of being unhinged. *International Conference on Neural Information Processing Systems*, 10-18. https://doi.org/10.48550/arXiv.1505.07634

[67] Vasudeva, B., Deora, P., Bhattacharya, S., Pal, U., and Chanda, S., (2021). LoOp: Looking for optimal hard negative embeddings for deep metric learning. *IEEE International Conference on Computer Vision*, 10634-10643. https://doi.org/10.48550/arXiv.2108.09335

[68] Wang, W., Feng, F., He, X., Nie, L., and Chua, T.-S., (2021). Denoising implicit Feedback for recommendation. *ACM International Conference on Web Search and Data Mining*,373–381. https://doi.org/10.1145/3437963.3441800

[69] Wang, X., Hua, Y., Kodirov, E., and Robertson, N.M., (2021). Ranked list loss for deep metric learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5207-5216. https://doi.org/10.48550/arXiv.1903.03238

[70] Wang, Y., Gan, W., Yang, J., Wu, W., and Yan, J., (2019). Dynamic curriculum learning for imbalanced data classification. *IEEE International Conference on Computer Vision*, 5016-5025. https://doi.org/10.48550/arXiv.1901.06783

[71] Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J., and Han, J., (2019). CrossWeigh: Training named entity tagger from imperfect annotations. *International Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, 5153-5162. https://doi.org/10.48550/arXiv.1909.01441

[72] Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X., (2015). Learning from massive noisy labeled data for image classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 2691–2699. https://doi.org/10.1109/CVPR.2015.7298885

[73] Xiao, Y., and Wang, W.Y., (2019). Quantifying uncertainties in natural language processing tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(1), 7322-7329. https://doi.org/10.48550/arXiv.1811.07253

[74] Xu, D., Ye, Y., and Ruan, C., (2021) Understanding the role of importance weighting for deep learning. *International Conference on Learning Representations 2021 Review Score*. https://doi.org/10.48550/arXiv.2103.15209

[75] Yan, P., Wu, Z., Liu, M., Zeng, K., Lin, L., and Li, G., (2022). Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning. *Association for the Advancement of Artificial Intelligence*. https://doi.org/10.48550/arXiv.2202.13170

[76] Yang, Z., Yu, Y., You, C., Steinhardt, J., and Ma, Y., (2020). Rethinking bias-variance trade-off for generalization of neural network. *International Conference on Machine Learning*, 10767-10777, 2020. https://doi.org/10.48550/arXiv.2002.11328

[77] Young, S.I., Dalca, A.V., Ferrante, E., Golland, P., Fischl, B., Iglesias, J.E., (2022). SUD: Supervision by denoising for medical image segmentation. *arXiv*:2202.02952.

[78] Yu, D., Yang, J., Zhang, Y., and Yu, S., (2021). Additive DenseNet: Dense connections based on simple addition operations. *J. Intell, Fuzzy Syst.*, *44*(3), 5015-5025. https://doi.org/10.3233/JIFS-201758

[79] Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M., (2020). Geometry-aware instance-reweighted adversarial training. *International Conference on Learning Representations*, 332. https://doi.org/10.48550/arXiv.2010.01736

[80] Zhang, Y., Yao, Q., Shao, Y., and Chen, L., (2019). NSCaching: simple and efficient negative sampling for knowledge graph embedding. *IEEE International Conference on Data Engeneering*, 614-625. https://doi.org/10.48550/arXiv.1812.06410

[81] Zhou, X., and Wu, O., (2021). Which samples should be learnt first? Easy or hard? *arXiv*:2110.05481.

[82] Z. Zhou, J. Luo, D. Arefan, G. Kitamura and S. Wu, "Human Not in the Loop: Objective Sample Difficulty Measures for Curriculum Learning," 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI), Cartagena, Colombia, 2023, pp. 1-5.

A CALCULATION OF MODEL COMPLEXITY

The model complexity used in this study is on the basis of minimum description length (MDL) [9] and the Kolmogorov complexity [46]. MDL and Kolmogorov complexity are combined and used to describe the model complexity under various learning tasks in [22].

Let $\mathbf{w} = (w_1, .., w_{d-1})^T$ be the model parameter of a regression model where d signifies the dimension of inputs. Let $p(\mathbf{w})$ be the probability density of $\mathbf{w}$. According to MDL, the model complexity expectation $c(\mathbf{w})$ is the expectation of model complexity

$$m(\mathbf{w}) = -\log p(\mathbf{w}),$$

over different training set T, i.e., $c(\mathbf{w}) = \mathbb{E}_T[m(\mathbf{w})]$. Suppose that each component of $\mathbf{w}$ is independent to each other and follows the identical Gaussian distribution $w_i \sim \mathcal{N}(0, \sigma^2)$. Therefore, the model complexity defined on the basis of

MDL equals to

$$\begin{aligned} m(\mathbf{w}) &= -\log p(\mathbf{w}) \\ &= -\sum_{i=1}^{d} \log p(w_i) \\ &= -\log \prod_{i=1}^{d} p(w_i) \\ &= -\log \prod_{i=1}^{d} [\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{w_i^2}{2\sigma^2})] \\ &= \sum_{i=1}^{d} \frac{w_i^2}{2\sigma^2} + d\log(\sqrt{2\pi}\sigma) \\ &= \frac{\|\mathbf{w}\|_2^2}{2\sigma^2} + d\log(\sqrt{2\pi}\sigma) \end{aligned} \tag{A.1}$$

When the construction of the basic learner and σ^2 is fixed, the model complexity only concerns $\|\mathbf{w}\|_2^2$.

Under the ridge regression, with the input denoted as x and its label denoted as y, the objective function is

$$\mathscr{L} = \sum_{n=1}^{N} l(f(x;\mathbf{w}), y) + \lambda\|\mathbf{w}\|_2^2,$$

and the estimated parameters are given by

$$\hat{\mathbf{w}}(\lambda) = (x^T x + \lambda I)^{-1} x^T y.$$

Based on the Assumption 1, the variance term is an increasing function with respect to the model complexity, and as the loss function normally used during the training procedure is in fact the bias term in the bias-variance trade-off, the generalization error can be estimated by the following form:

$$\begin{aligned} \mathscr{L} &= \sum_{n=1}^{N} l(f(x;\mathbf{w}), y) + m(\mathbf{w}) \\ &= \sum_{n=1}^{N} l(f(x;\mathbf{w}), y) + \frac{\|\mathbf{w}\|_2^2}{2\sigma^2} + d\log(\sqrt{2\pi}\sigma) \\ &\sim \sum_{n=1}^{N} l(f(x;\mathbf{w}), y) + \lambda\|\mathbf{w}\|_2^2 \end{aligned}$$

Accordingly, an enlargement of λ leads to a reducing of $\|\mathbf{w}\|_2^2$, and moreover, a lower model complexity. The above analysis indicates that the MDL-based model complexity well explains the ridge regression. However, the former calculation is based on the identical distribution assumption for each w_i. When the model is a polynomial function, the contributions of each component of $\mathbf{w}$ to the whole model complexity are not identical. For example, when using a polynomial function $g(x) \sim \mathscr{O}(3)$ to perform ridge regression, w_3 should contributes more to the model complexity comparing to w_0. Therefore, an identical distribution for all components of $\mathbf{w}$ is unreasonable. A more reasonable

assumption is that $w_i \sim \mathcal{N}(0, \sigma_i^2)$ with the condition that $\sigma_i^2 < \sigma_j^2$ if $i > j$. From Eq. (A.1), we have

$$\begin{aligned} m(\mathbf{w}) &= -\log p(\mathbf{w}) \\ &= -\sum_{i=1}^{d} \log p(w_i) \\ &= -\log \prod_{i=1}^{d} p(w_i) \\ &= -\log \prod_{i=1}^{d} [\frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{w_i^2}{2\sigma_i^2})] \\ &= \sum_{i=1}^{d} \frac{w_i^2}{2\sigma_i^2} + d\log(\sqrt{2\pi}\sigma_i) \end{aligned} \tag{A.2}$$

In our practical calculation, let $\sigma_i^2 = (\frac{d}{i}\sigma)^2$. Denoting $\hat{\mathbf{w}}_t(\lambda)$ as parameters of model learnt on training set T_t. Ignoring the constant term, the model complexity becomes

$$m(\hat{\mathbf{w}}_t(\lambda)) = \sum_{i=1}^{d} (\frac{i}{d}\hat{w}_{t,i}(\lambda))^2.$$

Given M training sets, the model complexity expectation is

$$c(\hat{\mathbf{w}}(\lambda)) = \frac{1}{M}\sum_{i=1}^{M}\sum_{i=1}^{d} (\frac{i}{d}\hat{w}_{t,i}(\lambda))^2. \tag{A.3}$$

B PROOFS OF PROPOSITIONS

B.1 Proof of Proposition 2

Proof.

$$\begin{aligned} Err^{\mathrm{w}}(\lambda_h) &= \sum_{y\in\Omega_Y} P(y) \int_{x\in\Omega_X} \omega\, Err(x,\lambda_h) p(x|y)\mathrm{d}x \\ &= \sum_{y\in\Omega_Y} P(y) \int_{x\in\Omega_X^r} \omega\, Err(x,\lambda_h) p(x|y)\mathrm{d}x \\ &+ \sum_{y\in\Omega_Y} P(y) \int_{x\in\Omega_X/\Omega_X^r} Err(x,\lambda_h) p(x|y)\mathrm{d}x \\ &= Err(\lambda_h) + \sum_{y\in\Omega_Y} P(y) \int_{x\in\Omega_X^r} (\omega-1) Err(x,\lambda_h) p(x|y)\mathrm{d}x \end{aligned}$$

Note that $\frac{\partial Err(\lambda_h)}{\partial c}|_{c=c^*} = 0$. Given that $\mathcal{LDC}(x) > 1$, $\frac{\partial Err(x,\lambda_h)}{\partial c}|_{c=c^*} < 0,\ \forall x \in \Omega^r$. With $\omega > 1$, we have

$$\frac{\partial Err^{\mathrm{w}}(\lambda_h)}{\partial c}|_{c=c^*} < 0.$$

According to Proposition 1, the new optimal model complexity c'^* will be larger than c^*. □

B.2 Proof of Corollary 1

Proof. The optimal complexity c^* under the original weights ω can be theoretically inferred under the (original) weighted distribution $P_1 \sim \omega P$. The learning with new weights $\tilde{\omega}$ equals to the learning with the weights $\tilde{\omega}/\omega$ for each

sample in Ω^r under the distribution P_1. Because the new weights are larger than the original weights on Ω^r, $\tilde{\omega}/\omega$ is larger than one on Ω^r. According to Proposition 3, the new optimal complexity becomes larger. □

B.3 Proof of Proposition 4

Proof. Let Ω^e and Ω^h be the regions containing the easy and hard samples according to c^*, respectively.

$$\begin{aligned}\frac{\partial Err^{w_0}(\lambda_h)}{\partial c}|_{c=c^*} =& \sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X^e}\omega_0(x)\frac{\partial Err(x,\lambda_h)}{\partial c}|_{c=c^*}p(x|y)\mathrm{d}x\\ &+\sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X^h}\omega_0(x)\frac{\partial Err(x,\lambda_h)}{\partial c}|_{c=c^*}p(x|y)\mathrm{d}x\\ =&0\end{aligned}.$$

Let $\omega_e^* = \max_{x\in\Omega^e}\omega(x)$ and $\omega_h^* = \min_{x\in\Omega^h}\omega(x)$. Moreover, $\omega_e^* \le \omega_h^*$. We have

$$\begin{aligned}\frac{\partial Err^{w}(\lambda_h)}{\partial c} =& \sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X^e}\omega(x)\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x\\ &+\sum_{y\in\Omega_Y} P(y)\int_{x\in\Omega_X^h}\omega(x)\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x\end{aligned}$$

Note that

$$\begin{aligned}\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}|_{c=c^*} &> 0, \forall\, x\in\Omega_X^e\\ \frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}|_{c=c^*} &< 0, \forall\, x\in\Omega_X^h\end{aligned}.$$

Therefore,

$$\underbrace{\int_{x\in\Omega_X^e}\omega(x)\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x}_{①} \le \int_{x\in\Omega_X^e}\omega_e^*\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x,$$

and

$$\int_{x\in\Omega_X^h}\omega_h^*\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x \ge \underbrace{\int_{x\in\Omega_X^h}\omega(x)\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x}_{②}$$

$$\begin{aligned}① + ② \le& \int_{x\in\Omega_X^e}\omega_e^*\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x + \int_{x\in\Omega_X^h}\omega_h^*\frac{\partial Err'(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x\\ \le& \int_{x\in\Omega_X^e}\omega_h^*\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x + \int_{x\in\Omega_X^h}\omega_h^*\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}p(x|y)\mathrm{d}x\\ =&\omega_h^*\int_{x\in\Omega_X}\frac{\partial Err^{w_0}(x,\lambda_h)}{\partial c}|_{c=c^*}p(x|y)\mathrm{d}x\\ =&0\end{aligned}.$$

The equal relation holds if and only if

$$\min_{x\in\Omega^e}\omega(x) = \omega_e^* = \omega_h^* = \max_{x\in\Omega^h}\omega(x).$$

Note that $\min\omega(x) < \max\omega(x)$. Therefore, $\dfrac{\partial Err^{w}(\lambda_h)}{\partial c} < 0$. Accordingly, the optimal complexity becomes larger. □

C ONLINE RESOURCES

https://github.com/Weiyao619/GELD.git