Which Samples Should be Learned First: Easy or Hard?

Xiaoling Zhou and Ou Wu

***Abstract*—Treating each training sample unequally is prevalent in many machine learning tasks. Numerous weighting schemes have been proposed. Some schemes take the easy-first mode, whereas some others take the hard-first one. Naturally, an interesting yet realistic question is raised. Given a new learning task, which samples should be learned first, easy or hard? To answer this question, both theoretical analysis and experimental verification are conducted. First, a general objective function is proposed and the optimal weight can be derived from it, which reveals the relationship between the difficulty distribution of the training set and the priority mode. Two novel findings are subsequently obtained: besides the easy-first and hard-first modes, there are two other typical modes, namely, medium-first and two-ends-first; the priority mode may be varied if the difficulty distribution of the training set changes greatly. Second, inspired by the findings, a flexible weighting scheme (FlexW) is proposed for selecting the optimal priority mode when there is no prior knowledge or theoretical clues. The four priority modes can be flexibly switched in the proposed solution, thus suitable for various scenarios. Third, a wide range of experiments is conducted to verify the effectiveness of our proposed FlexW and further compare the weighting schemes in different modes under various learning scenarios. On the basis of these works, reasonable and comprehensive answers are obtained for the easy-or-hard question.**

***Index Terms*—Priority mode, learning difficulty, weighting strategy, easy-first, hard-first, bias-variance trade-off.**

I. Introduction

MANY machine learning models, in particular neural networks, are sensitive to the weights of training samples. Treating each training sample unequally can improve the learning performance of these models [1]–[4]. The cues and inspirations for the design of the weighting methods are usually derived from the following two aspects:

- Context-inspired weighting methods. Tasks such as fraud detection [5] and medical diagnosis [6] are cost-sensitive. Different samples have unequal importance according to their gains or costs. Therefore, samples with high gains/costs should be assigned with high weights.
- Characteristics-inspired weighting methods. Training samples are different from each other in characteristics, such as data quality [7]–[9], sample neighbors [10], margin [11], and category distribution [2], [12], [13]. In some tasks, samples in the minority categories are generally more difficult to learn well, so these samples should be assigned with high weights. In some other tasks, some labels of samples are of low confidence or with high noise, so these samples should be assigned with low weights.

Context-inspired weighting methods are usually defined in a heuristic manner and are only employed in particular applications. In contrast, characteristics-inspired weighting strategies have received increasing attention in recent years due to their effectiveness and universality. Data characteristics are related to an intrinsic property of training samples, namely, learning difficulty. Most related studies split training samples into easy/hard or easy/medium/hard according to their learning difficulty [14], [15]. In some schemes, hard samples are assigned with high weights, which is called the hard-first mode. For example, Lin et al. [3] proposed Focal loss in object detection, which significantly improves the detection performance. In some other schemes, easy samples have higher weights than hard ones, which is called the easy-first mode. For example, Kumar et al. [16] proposed Self-paced Learning (SPL), which sets the weights of hard samples to zero with a threshold. The threshold is gradually increased to ensure that more hard samples can participate in the next training epochs. These two priority modes, namely, easy-first and hard-first, appear to contradict each other, yet both demonstrate effectiveness in specific learning tasks. Consequently, a natural question is raised. Which samples should be learned first facing a new learning task, easy or hard? Indeed, several studies have proposed similar concerns. For example, Wang et al. [15] raised a similar question about “easy-first versus hard-first” under the context of Curriculum Learning (CL).

To answer the question (called the “easy-or-hard” question for brevity), partial observations and conclusions in existing studies are summarized, and preliminary answers are obtained, which are as follows:

- The weights for noisy samples should be decreased, making the model less disturbed by noise [17], [18]. In other words, the easy-first mode will be more effective in training data with heavy noise.
- If easy samples are excessive, the hard-first mode is preferred. For example, in the application of Focal loss [3], hard samples are assigned with high weights in object detection tasks. In imbalanced learning, head categories (i.e., categories with a relatively large proportion of samples) are relatively easy in general, and thus easy samples are excessive when a high imbalance exists [19].
- According to the human learning mechanism, easy sam-

Manuscript received x xx, xxxx; revised x xx, xxxx. This work was supported in part by NSFC under Grant 62076178, in part by TJF under Grant 22ZYYYJC00020, and in part by ZJF under Grant 2019KB0AB03.

Xiaoling Zhou and Ou Wu are in the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China. Ou Wu is the corresponding author. Email addresses: xiaolingzhou@tju.edu.cn (Xiaoling Zhou), wuou@tju.edu.cn (Ou Wu).

TABLE I
SEVERAL TYPICAL WEIGHTING METHODS.

Method	Weighting scheme	Domain	Scenario	Measure	Priority mode	Granularity
SPL-Binary [16]	$\min_{w\in[0,1]^n} \mathcal{L}(w,\lambda,l)=\sum_{i=1}^n w_i l_i-\lambda\sum_{i=1}^n w_i$	NLP CV	Noun Phrase Coreference Image Classification Object Localization (Standard)	Loss	Easy-first	Sample
SPL-Log [20]	$\min_{w\in[0,1]^n} \mathcal{L}(w,\lambda,l)=\sum_{i=1}^n w_i l_i+\sum_{i=1}^n\left(\xi w_i-\xi^{w_i}/\log\xi\right),\xi=1-\lambda$	CV	Multimedia Event Detection (Standard)	Loss	Easy-first	Sample
SPLD [21]	$\min_{\mathbf{w},\mathbf{v}} \mathbb{E}(\mathbf{w},\mathbf{v};\lambda,\gamma)=\sum_{i=1}^n v_i L\left(y_i,f\left(\mathbf{x}_i,\mathbf{w}\right)\right)-\lambda\sum_{i=1}^n v_i-\gamma\lVert\mathbf{v}\rVert_{2,1}$, s.t. $\mathbf{v}\in[0,1]^n$	CV	Multimedia Event Detection Action Recognition (Standard)	Loss	Easy-first	Sample
Focal loss [3]	$\mathcal{L}(\gamma)=-(1-p)^\gamma\log(p)$	CV	Dense Object Detection (Imbalanced)	Predicted probability	Hard-first	Sample
QFL [22]	$\mathcal{L}(\sigma,\beta)=\sum_{i=1}^N\left(-\lvert y_i-\sigma\rvert^\beta\left((1-y_i)\log(1-\sigma)+y_i\log(\sigma)\right)\right)$	CV	Dense Object Detection (Imbalanced)	Predicted probability	Hard-first	Sample
AdaBoost [23]	$w_i^{m+1}=w_i^m\exp\left(\alpha_m\right)$	CV	Handwritten Digit Recognition (Standard)	Error	Hard-first	Sample
G-RW [24]	$w^c=(1/r_c)^\rho/\sum_{k=1}^C(1/r_k)^\rho$	CV	Image Classification Object Detection (Imbalanced)	Empirical category frequency	Hard-first	Category
Curriculum Learning [25]	$w_i<w_j,\forall\gamma\left(x_i\right)<\gamma\left(x_j\right)$	NLP CV	Language Modeling Shape Recognition (Standard)	Prior knowledge	Easy-first	Sample
Self-paced Curriculum Learning [26]	$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v},\lambda,\Psi)=\sum_{i=1}^n v_i L\left(y_i,g\left(\mathbf{x}_i,\mathbf{w}\right)\right)+f(\mathbf{v};\lambda)$ s.t. $\mathbf{v}\in\Psi$	CV	Matrix Factorization, Multimedia Event Detection (Standard)	Curriculum and Loss	Easy-first	Sample
Balanced Curriculum Learning [27]	$w\left(x_i,t\right)=[1-\alpha\cdot\left(\text{diff}\left(x_i\right)\cdot e^{-\gamma\cdot t}\right)$ $-(1-\alpha)\cdot\left(\text{imgVisited}\left(x_i\right)\cdot e^{-\gamma\cdot t}\right)]^k$	CV	Instance Segmentation, Object Detection (Standard)	Human response time and Diversity	Mixed mode	Sample
GAIRAT [11]	$w_i=\left(1+\tanh\left(\lambda+5\times\left(1-2\times k\left(x_i,y_i\right)/K\right)\right)\right)/2$	CV	Image Classification (Standard)	Margin	Hard-first	Sample
Class-balanced [2]	$w^c=(1-\beta)/\left(1-\beta^{N_c}\right),\beta\in[0,1)$	CV	Image Classification (Imbalanced)	Category proportion	Hard-first	Category
Truncated loss [17]	$l_i=\begin{cases}0, & l_i^{CE}>0\wedge y_i=1\\ l_i^{CE}, & \text{otherwise}\end{cases}$	Data mining	Recommendation (Noisy)	Loss	Easy-first	Mixture
R-CE [17]	$\mathcal{L}=-p^\beta\log(p)$	Data mining	Recommendation (Noisy)	Predicted probability	Easy-first	Sample
LOW [19]	$R(w;\lambda)=-w^T\nabla_{\theta_t}+\lambda\lVert w-1\rVert^2$	CV	Image Classification (Imbalanced)	Gradient norm	Hard-first	Sample
JTT [28]	$\mathcal{L}(l,E)=\left(\lambda_{up}\sum_{(x_i,y_j)\in E}l_i+\sum_{(x_j,y_j)\notin E}l_j\right)$	NLP CV	Image Classification Sentiment Analysis (Standard)	Loss	Hard-first	Partial data
SuperLoss [18]	$\mathcal{L}\left(l_i,\sigma_i\right)=\left(l_i-\tau\right)w_i+\lambda\left(\log w_i\right)^2$	CV	Object Detection Image Retrieval (Noisy)	Loss	Easy-first	Sample
DWB [13]	$L_{\text{DWB}}=-\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^c w_j^{(1-p_{ij})}y_{ij}\log\left(p_{ij}\right)-p_{ij}\left(1-p_{ij}\right)$	CV	Cyber Intrusion Detection Skin Lesion Diagnosis (Imbalanced)	Predicted probability and Category proportion	Hard-first	Sample

* SPL means Self-paced Learning. SPLD means Self-paced Learning with Diversity. QFL means Quality Focal loss. G-RW means Generalized Re -weight. GAIRAT means Geometry-aware Instance-reweighted Adversarial Training. R-CE means Reweighted Cross-Entropy loss. LOW means Learning Optimal samples Weights. JTT means Just Train Twice. DWB means Dynamically Weighted Balanced loss.

ples should be learned first.

The above answers are far from satisfactory for the "easy-or-hard" question because there are still some deep concerns:

(i) Are there any other possible priority modes besides the easy-first and hard-first modes?

(ii) The second answer listed above refers to the difficulty distribution of training samples. What is the relationship between the difficulty distribution and the priority mode?

(iii) The priority mode is fixed during the training procedure in nearly all existing studies. Can the priority mode be changed during the entire training process?

(iv) When there is no prior knowledge or theoretical clues such as noise and difficulty distribution discussed in the preliminary answers, is there an effective and universal solution to assign weights on training samples?

To solve the four subproblems listed above, both theoretical and empirical investigations are conducted and comprehensive answers are obtained. Our contributions are summarized as follows:

- Preliminary observations and conclusions from existing studies are summarized. Four unsolved subproblems are raised according to the gap between the question and the summarized preliminary answers.
- To theoretically explore the "easy-or-hard" question, a general objective function is constructed for the difficulty-based sample weights. It reveals the relationship between the learning difficulty distribution and the priority mode. Theoretical analysis based on the bias-variance trade-off theory is then carried out.
- An effective and universal weighting solution is proposed for selecting the optimal priority mode when there is no prior knowledge or theoretical clues. Our proposed solution can be flexibly switched among the four modes (i.e., easy-first, medium-first, hard-first, and two-ends-first) only by changing its hyperparameters, while existing weighting methods can only achieve partial modes.
- Extensive experiments on various learning tasks under different scenarios are conducted on benchmark datasets. The empirical observations further support our main theoretical conclusions. Thus, an in-depth and comprehensive answer to the easy-or-hard question is obtained. In addition, our proposed weighting method achieves competitive results in all the above learning scenarios.

II. RELATED WORK

A. Description of Symbols

We define the symbols including the main symbols in Table I. Let $T=\{(x_i,y_i)\}_{i=1}^N$ be a set of N training samples, where x_i is the input feature and y_i is the associated label. Let C be the number of categories and $y_i\in\{1,\ldots,C\}$. Let r_c be the empirical category frequency of the c-th category. The training loss is denoted as $\mathcal{L}$. Let w_i and l_i be the weight and loss of the i-th sample. Let $p_i\in[0,1]$ be the predicted

TABLE II
FIVE CATEGORIES OF DIFFICULTY MEASURES AND THEIR CORRESPONDING TYPICAL METHODS.

Measure	Method	Number	Scenario
Prediction	SPL-Binary (2010), SPL-Log (2014), Cost-sensitive SPL (2016), Focal loss (2017), QFL (2020), ASL (2020), SuperLoss (2020), Truncated loss (2021), JTT (2021), DWB (2021), R-CE(2021)	11	Standard, Noisy, Imbalanced
Category proportion	Class-balanced loss (2019), G-RW (2021), DWB (2021)	3	Imbalanced
Gradient	GHM(2019), LOW (2021)	2	Imbalanced
Margin	GAIRAT (2021)	1	Standard
Uncertainty	FOCI (2020)	1	Noisy

* ASL means Asymmetric loss. GHM means Gradient Harmonizing Mechanism. FOCI means Focus On Clean and Informative samples.

probability of sample x_i for the ground truth. Let d_i be the i-th sample's learning difficulty, which can be approximated by other values, such as l_i and $1 - p_i$. w^c is defined as the weight of the c-th category when the category-wise weighting strategy is utilized.

B. Existing Weighting Methods

Table I lists some typical weighting methods in previous literature. The core of a weighting scheme is its weighting function for the input samples. According to the granularity of the weighting methods, the weighting functions can be sample-wise, category-wise, or their mixtures. According to the priority mode, existing weighting schemes can be divided into easy-first and hard-first. Their application scenarios (i.e., standard, imbalanced, and noisy) are also presented. The methods in Table I are inspired by a (partial) particular view of the data characteristics, and thus each method only implements one priority mode. Although some of them can achieve more than one priority modes with some minor changes, only one mode is used in their previous applications. For example, if the hyperparameter γ in Focal loss [3], which is a typical weighting method for object detection, is negative, the easy-first mode can be achieved. Nevertheless, almost all applications of Focal loss choose the hard-first mode (i.e., $\gamma > 0$).

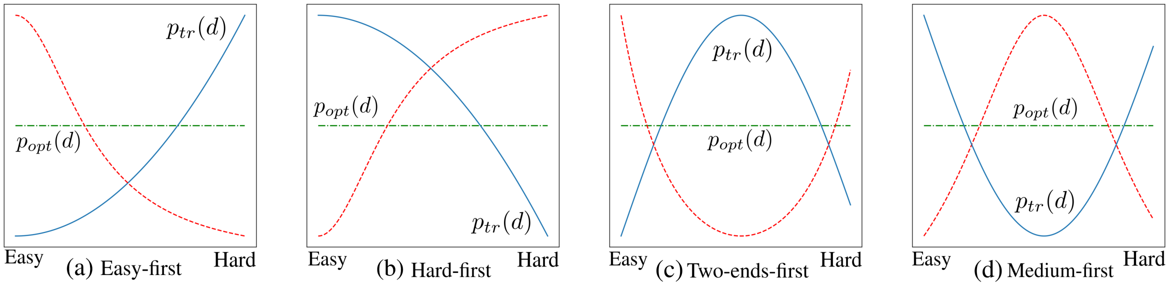
Both two priority modes are widely adopted in various learning scenarios according to the characteristics of training data. Hard-first weighting schemes are commonly used for imbalanced data. For example, Focal loss [3] is inspired by the observation that easy samples are excessive in object detection datasets relative to hard ones. Thus, it assigns relatively high weights to hard samples and relatively low weights to easy samples, which greatly improves the detection performance. Class-balanced loss [2] exerts high weights on samples in tail categories (i.e., categories with a relatively small proportion of samples), which is a typical method for imbalanced learning. Learning Optimal samples Weights (LOW) [19] forces the model to focus on less represented or more challenging samples and works well for imbalanced data. Dynamically Weighted Balanced loss (DWB) [13] sets higher weights on hard-to-train instances based on the category proportion and the predicted probability of ground truth.

Easy-first weighting methods are verified to be effective on noisy data. SPL [18] sets the weights of samples with large losses to zero with a threshold. The threshold is gradually increased to ensure that more hard samples can participate in subsequent epochs. SuperLoss [16] downweights the contribution of samples with large losses to decrease the impact of noisy samples. There are some easy-first schemes inspired by clues other than loss. CL [25], [29] is motivated by human learning that easy samples should be learned first. The observations from an empirical study [4] imply that the easy-first paradigm CL mainly takes effect in noisy scenarios. Some recent CL methods adopt more complicated priority modes [27], [30], [31]. In Balanced CL [29], on the basis of the easy-first mode, the selection of samples has to be balanced under certain constraints to ensure diversity across image regions [30] or categories [27]. Therefore, Balanced CL adopts the mixed mode. In Teacher-student CL [31], the curriculum of the student network is learned by its corresponding teacher network. Thus, its priority mode depends on the interaction between the student and teacher networks.

C. Measures for Learning Difficulty of Samples

Learning difficulty is an intrinsic property of training samples, which depends on various factors that are related to data characteristics, including data quality [7]–[9], [32], sample neighbors [10], margin [11], and category distribution [2], [19], [33]. The lower the quality of a sample is, the larger the learning difficulty of the sample will be; the more heterogeneous samples in the neighborhood of a sample are, the larger the learning difficulty of the sample will be; the smaller the margin of a sample has, the larger the learning difficulty of the sample will be; the smaller a category is, the larger the learning difficulty of samples in this category will be. Most existing difficulty measures are defined by heuristics, which can be divided into the following five categories.

- Prediction-based measures. This category mainly employs the loss [16], [18] or the predicted probability of the ground truth [3], [34] as the difficulty measure. The motivation is that a large loss (or a small predicted probability of ground truth) indicates a large difficulty.
- Category proportion-based measures. This category utilizes the category proportion [2], [13] as the difficulty measure. The intuition is that the smaller the category proportion, the harder the samples in this category.
- Gradient-based measures. This category adopts the loss gradient [19] as the difficulty measure. The motivation is that the larger the norm of the loss gradient, the harder the sample [19].
- Margin-based measures. Margin [11] means the distance from the sample to the decision boundary. The motivation is that a small margin indicates a large learning difficulty.
- Uncertainty-based measures. This category utilizes the uncertainty of a sample to measure its learning difficulty.



(a) Easy-first (b) Hard-first (c) Two-ends-first (d) Medium-first

Fig. 1. Four typical cases of the real and optimal difficulty distributions of the training data. (The green and blue lines represent $p_{opt}(d)$ and $p_{tr}(d)$ where $p_{opt}(d)$ is assumed to be uniform. The red line represents the curve of $\tau(d)$.)

Samples with large uncertainties are generally considered hard ones [35].

Table II lists the five categories of difficulty measures and their corresponding typical methods. Prediction-based measures (i.e., loss and predicted probability of ground truth) are the most widely used. Four recent methods, including Just Train Twice [28], Truncated Loss [17], Reweighted Cross-Entropy loss (R-CE) [17], and DWB [13] still use the loss as the difficulty measure. In addition, this category is simple yet effective in various learning scenarios [3], [16], [34].

III. Theoretical Investigation

A general theoretical framework is lacking to pursue the learning difficulty-based sample weights in existing studies. To conduct theoretical analyses, a general objective function is first proposed.

A. A General Objective Function

Learning difficulty-based weighted loss $\mathcal{L}_d$ is defined as:

$$\mathcal{L}_d = \frac{1}{N}\sum\nolimits_i w_i(d_i)l_i, \tag{1}$$

where w_i depends on d_i, and N is the number of samples in the training set. The sample weights can adjust the difficulty distribution of the training samples. Let $d(x)$ be the learning difficulty of x. Let $P_{opt}[d(x)]$[1] and $P_{tr}[d(x)]$ be the optimal and real difficulty distributions of the training data, and their corresponding densities are denoted as $p_{opt}[d(x)]$ and $p_{tr}[d(x)]$. Let $\widetilde{P}_{tr}[d(x)]$ be the new difficulty distribution of the training set with the weights of samples and $\widetilde{p}_{tr}[d(x)] \propto w[d(x)]p_{tr}[d(x)]$, where $\widetilde{p}_{tr}[d(x)]$ is the density of $\widetilde{P}_{tr}[d(x)]$. Theoretically, the optimal weights can be pursued with the following objective function:

$$\min_{w} KL(\widetilde{P}_{tr}[d(x)]||P_{opt}[d(x)]), \tag{2}$$

where KL is the Kullback–Leibler divergence [36] between the two distributions. According to Eq. (2), the optimal weight for a training sample x is[2]

$$w^*[d(x)] \propto \frac{p_{opt}[d(x)]}{p_{tr}[d(x)]}. \tag{3}$$

In the following discussion, x is omitted for brevity. If the optimal and real difficulty distributions of the training data are the same, then all samples are equally important.

If helpful information (e.g., partial data characteristics cues) is available and reasonable assumptions are accessible, an appropriate value for $w^*(d)$ can be obtained. The form of the optimal weight (i.e., Eq. (3)) can explain a lot of weighting strategies. For example, if the learning difficulty of samples in the same category is assumed to be equal (different categories have different learning difficulties) and $p_{opt}(d)$ is assumed to be uniform, then the reciprocal of category proportion is the optimal weight for each sample which is commonly employed in imbalanced data [2]. Gradient Harmonizing Mechanism (GHM) [8] can also be explained by Eq. (3). Its weighting function is

$$\begin{aligned} w(g) &= 1/GD(g), \\ GD(g) &= \frac{1}{l_\epsilon}\sum_{k=1}^{N}\delta_\epsilon\left(g_k, g\right), \end{aligned} \tag{4}$$

where $GD(g)$ is the gradient density which denotes the number of examples lying in the region centered at g with a length of ϵ and normalized by the length of the region l_ϵ. Thus, GHM utilizes the norm of gradient g as the difficulty measure and assumes $p_{opt}(g)$ to be uniform. Therefore, $p_{tr}(g) = GD(g)$ and the reciprocal of the gradient density is the optimal weight.

B. Four Typical Priority Modes

Based on Eq. (3), a new concept, the relative learning difficulty $\tau(d)$, is introduced, which is denoted as the likelihood ratio $\tau(d) = p_{opt}(d)/p_{tr}(d)$. It measures the insufficient degree of samples with difficulty d. If $\tau(d) < 1$, then samples with learning difficulty d are excessive in the training set. The smaller the $\tau(d)$ is, the more excessive the samples with difficulty d are. According to Eq. (3), $w^*(d) \propto \tau(d)$. Therefore, these samples should have relatively low weights. On the contrary, if $\tau(d) > 1$, then samples with learning difficulty d are insufficient in the training set. The larger the $\tau(d)$ is, the more insufficient the samples are. According to Eq. (3), high weights should be assigned to these samples. Thus, samples with large relative difficulty should be assigned with high weights.

Because the two cases of excessive and insufficient samples are complementary, we only need to analyze the typical cases of excessive samples. Four typical cases, including excessive easy/medium/hard/both easy and hard samples in the training data, can be obtained and discussed as follows:

[1] If there are sufficient data in the validation set, the optimal difficulty distribution of the training set is similar to the difficulty distribution of the validation set (the whole space).

[2] The case of $p_{tr}[d(x)] = 0$ is not considered as the value of $w^*[d(x)]$ is meaningless in this case.

(1) If hard samples are excessive in the training data, then the relative difficulty $\tau(d)$ is small for hard samples. Thus, hard samples should be assigned with relatively low weights. Alternatively, the "easy-first" mode should be adopted. Fig. 1(a) illustrates this case where $p_{opt}(d)$ is assumed to be uniform and $\tau(d)$ decreases on d.

(2) If easy samples are excessive in the training data, then the relative difficulty $\tau(d)$ is small for easy samples. Thus, easy samples should be assigned with relatively low weights. Alternatively, the "hard-first" mode should be utilized. Fig. 1(b) illustrates this case and $\tau(d)$ increases on d.

(3) If medium-difficult samples are excessive in the training data, then $\tau(d)$ will be small for medium-difficult samples. Thus, samples with medium difficulty should be assigned with relatively low weights. Alternatively, the "two-ends-first" mode should be adopted. Fig. 1(c) illustrates this case and $\tau(d)$ decreases at first and then increases on d.

(4) If both easy and hard samples are excessive in the training data, then $\tau(d)$ is small for both easy and hard samples. Thus, both easy and hard samples should be assigned with relatively low weights. Alternatively, the "medium-first" mode should be taken. Fig. 1(d) illustrates this case and $\tau(d)$ increases at first and then decreases on d.

According to the above analysis, the following answers to Subproblems (i) and (ii) are obtained:

- For Subproblem (i), besides the easy-first and hard-first modes, at least two other typical priority modes exist, namely, medium-first and two-ends-first. Indeed, these two modes have been utilized in existing studies. For example, GHM [8] decreases the weights of over-hard (noise) samples on the basis of the hard-first mode. Thus, in our study, GHM is under the medium-first mode. Yang et al. [37] proposed the Self-paced Balance Learning, whose priority mode is a combination of the easy-first and hard-category-first modes. Thus, this priority mode is an approximation of the two-ends-first mode.
- For Subproblem (ii), Eq. (3) well reveals the relationship between the priority modes and the two learning difficulty distributions of the training set, namely, $P_{opt}(d)$ and $P_{tr}(d)$. In addition, samples with a large/small relative difficulty $\tau(d)$ should be assigned with high/low weights.

C. Analysis based on the Bias-variance Trade-off Theory

The bias-variance trade-off theory is a basic theory for the qualitative analysis of the generalization error for models [38]. In this section, it is used to support that when easy/hard samples are excessive, the priority mode of hard/easy-first should be adopted. It also indicates that the optimal model complexity can be changed by weighting the training samples.

Let T be a random training set and $f(x|T)$ be the trained model on T. The bias-variance trade-off is based on the following learning error [39]:

$$\begin{aligned} Err &= E_{x,y} E_T \left[\| y - f\left(x \left| T \right.\right) \|_2^2 \right] \\ &= Bias^2 + Variance + \delta_e \\ &\approx BiasT + VarT, \end{aligned} \tag{5}$$

where δ_e refers to the noise term. It indicates that the bias and variance terms will decrease and increase if the model complexity c increases [40]. Minimum learning error is achieved when the sum of the partial derivatives of the two terms with respect to the model complexity c is equal to zero. Training samples can be divided into easy, medium, and hard according to their learning difficulty. Therefore, the sample space can be divided into three corresponding regions, namely, R_{easy}, R_{medium}, and R_{hard}[3]. Thus, the generalization errors for $f(x|T)$ in the three regions can be obtained. The learning error for $f(x|T)$ in R_{easy} is as follows:

$$\begin{aligned} Err_{easy} &= E_{(x,y) \in R_{easy}} E_T \left[\| y - f\left(x \left| T \right.\right) \|_2^2 \right] \\ &\approx BiasT_{easy} + VarT_{easy}. \end{aligned} \tag{6}$$

Likewise, we can define the learning error in the R_{medium} and R_{hard} regions. According to the law of total expectation, the entire learning error can be decomposed into

$$Err = p_{easy} Err_{easy} + p_{medium} Err_{medium} + p_{hard} Err_{hard}, \tag{7}$$

where p_{easy}, p_{medium}, and p_{hard} are the possibilities of a random sample coming from R_{easy}, R_{medium}, and R_{hard}, respectively. Naturally, $p_{easy}+p_{medium}+p_{hard} = 1$. Based on the bias-variance trade-off theory on the entire sample space, we propose the following assumption:

Assumption 1: For all the three bias (e.g., $BiasT_{easy}$) and variance terms (e.g., $VarT_{easy}$) of R_{easy}, R_{medium}, and R_{hard}, the bias and variance terms are decreasing and increasing functions with respect to the model complexity c, respectively. Both the partial derivatives of the bias and variance terms with respect to c are increasing functions.

According to Assumption 1, minimum learning error for each region is achieved when the sum of the partial derivatives of its bias and variance terms with respect to c equals to zero. Let c^* be the optimal model complexity for the entire sample space when the minimum Err is attained. Likewise, let c^*_{easy} and c^*_{hard} be the optimal model complexities for R_{easy} and R_{hard}, respectively. The following assumption is proposed:

Assumption 2: $c^*_{easy} < c^* < c^*_{hard}$.

With Assumption 2, we have the following propositions.

Proposition 1: If weights higher than one are exerted on the samples in R_{hard}, and the weights for samples in other regions remain one, then the new optimal model complexity c^*_{new} over the entire space will be larger than c^*. Alternatively, the complexity of the optimal model is increased.

Proposition 2: If weights higher than one are exerted on samples in the R_{easy}, and the weights for samples in other regions remain one, the new optimal model complexity c^*_{new} over the entire space will be smaller than c^*. Alternatively, the complexity of the optimal model is decreased.

A theoretical analysis for *Proposition 1* is shown in the Appendix. *Proposition 1* supports that when easy samples are excessive in the training set, the trained model will become quite simple and underfitting. Thus, hard samples should be

[3] In our theoretical analysis, we assume that samples can be divided into easy, medium, and hard ones according to their learning difficulty. However, it is challenging and unnecessary to strictly distinguish which category a sample belongs to in real applications.

assigned with high weights to increase the complexity of the model. Alternatively, the hard-first mode should be adopted. *Proposition 2* supports that when hard samples are excessive in a training set, the easy-first mode should be utilized. These two propositions further support the two conclusions that we have discussed in Section III-B.

D. Varied Priority Mode

Subproblem (iii) concerns whether the fixed priority mode during the training procedure is always optimal. The "fixed mode" means that only one priority mode is adopted during the whole training procedure, while the "varied mode" means that different priority modes are used in different training stages. According to Eq. (3), the priority mode of the optimal weight depends on the relative difficulty $\tau(d)$. If $\tau(d)$ is fixed during the training procedure, the priority mode adopted should not be varied. For example, when "category proportion" is used as the difficulty measure [2], $\tau(d)$ will be fixed during the training process. Note that although the priority mode remains unchanged on some occasions, the weights of samples are still variable to achieve lower generalization errors of models [40]. The reason is that the learning difficulty of samples varies at each epoch as the performance of the trained model varies during training. With the improvement of the model performance, the learning difficulties of most samples will be decreased. For example, many studies leverage training loss to measure learning difficulty. The losses of most samples will decrease from one epoch to the next and thus their learning difficulties can be viewed as decreasing.

In general, $\tau(d)$ depends on both $p_{opt}(d)$ and $p_{tr}(d)$. Although $p_{opt}(d)$ is unknown, it is fixed during the training process. Therefore, the answer to Subproblem (iii) is subject to $p_{tr}(d)$. As described in Section II-C, the learning difficulty of samples is commonly measured by some heuristic factors such as loss and the norm of loss gradient in existing studies. These factors are changeable during the training procedure, so $p_{tr}(d)$ is not fixed during the training process. If the variation of $p_{tr}(d)$ is significant, then $\tau(d)$ will change greatly. Theoretically, the priority mode may be varied if $p_{tr}(d)$ changes drastically. For example, when the loss is used as the difficulty measure, there will be numerous samples with a large loss in the early training stage, as shown in Fig. 1(a). With the enhancement in the model performance, easy samples will dominate the training set if the training set has no significant deviations (e.d., heavy noise), as shown in Fig. 1(b). Thus, during the training process, the easy-first mode should be adopted in the initial training stage, and the hard-first mode should be adopted gradually.

Based on the above analysis, an answer to Subproblem (iii) is obtained. The priority mode is not absolutely fixed during the training procedure, and it should be varied if $p_{tr}(d)$ changes significantly.

E. Flexible Weighting Function

If there is no prior knowledge, theoretical clues, or empirical observations, the self-paced paradigm may be helpful as it is inspired by human learning mechanisms. However, this pattern does not work in all scenarios [4], [41]. To explore a more universal and effective solution (answer) to Subproblem (iv), a weighting strategy that can achieve all four priority modes should be proposed. Specifically, a weighting function is defined with the condition that all the four typical priority modes described in Section III-B can be achieved when different values of hyperparameters are taken. Thus, the selection of the appropriate priority mode is transformed into the problem of hyperparameter selection (optimization).

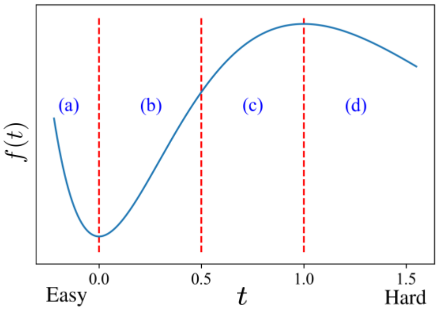


Fig. 2. The curve of $f(t)$ when $\gamma = 2$.

Let Λ be the hyperparameter(s) of a weighting function. D is defined as the domain of the learning difficulty of training samples. Ω is defined as the domain of the hyperparameter Λ. According to the four typical weight curves shown in Fig. 1, a weighting function $w(d;\Lambda)$, which can achieve the four priority modes, should satisfy the following three conditions:

(1) $w(d;\Lambda) \geq 0$, $\forall d \in D$ when $\Lambda \in \Omega$.
(2) A hyperparameter grouped with d, which can make the curve horizontally shift, is required. It ensures that any segment of the curve can be taken when $d \in D$.
(3) $w(d;\Lambda)$ should contain a local maximum and a local minimum under the same or different values of $\Lambda \in \Omega$.

Condition (3) guarantees that the medium-first and two-ends-first modes can be implemented. Specifically, if a selected segment contains a local minimum, then the two-ends-first priority mode is realized; if a selected segment contains a local maximum, then the medium-first priority mode is realized.

Let α and γ be two hyperparameters. We propose a weighting function that can achieve all four priority modes, namely, Flexible Weighting Function (FlexW), which is

$$w_i = (d_i + \alpha)^{\gamma} e^{-\gamma(d_i + \alpha)}. \tag{8}$$

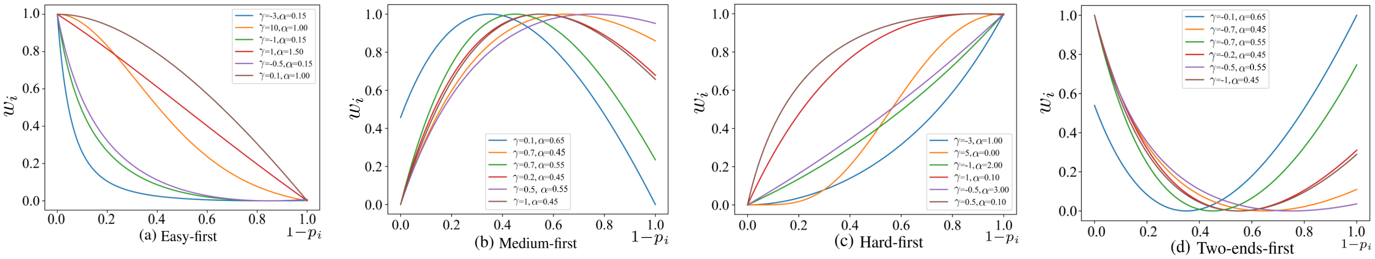
FlexW satisfies all the abovementioned conditions, and thus it can achieve all four priority modes. If γ is an even number or $\alpha + d_i > 0$, then Condition (1) holds. Condition (2) is also obviously satisfied. We analyze whether Condition (3) is satisfied. To simplify the form, $d + \alpha$ is denoted as t. Thus, the weighting function becomes

$$f(t) = t^{\gamma} e^{-\gamma t}. \tag{9}$$

Taking the derivative of Eq. (9), we obtain

$$f^{'}(t) = \gamma t^{\gamma-1} e^{-\gamma t}(1 - t). \tag{10}$$

Regardless of the value of γ, the function either has both a local maximum and a local minimum or contains only a local maximum or a local minimum. Thus, Condition (3) is satisfied.



(a) Easy-first (b) Medium-first (c) Hard-first (d) Two-ends-first

Fig. 3. The weight curves of four priority modes achieved by FlexW with different values of hyperparameters using $1-p$ as the difficulty measure.

Fig. 2 shows the curve of $f(t) = t^{\gamma}e^{-\gamma t}$ when $\gamma = 2$. If segment (a) is selected, then the "easy-first" priority mode is implemented; if segment (b) or (c) is selected, then the "hard-first" mode is implemented; if segments (a) and (b) are both selected, then the "two-ends-first" mode is implemented; if segments (c) and (d) are both selected, then the "medium-first" mode is implemented. Considering that $d \in D$, which segment(s) is/are selected depends on the horizontal shift hyperparameter α. If γ is set to zero, then the weights of all samples are equal to one, and thus CE loss with FlexW degenerates into the ordinary CE loss.

When different values of α and γ are chosen, different priority modes can be achieved by FlexW. Following Focal loss [3], $1-p$ is utilized to approximate the learning difficulty of samples. Fig. 3 shows the weight curve examples including "easy-first" (Fig. 3(a)), "medium-first" (Fig. 3(b)), "hard-first" (Fig. 3(c)) and "two-ends-first" (Fig. 3(d)), when different values of α and γ are chosen. Therefore, only the hyperparameters in FlexW are needed to be changed without changing the entire weighting function when facing different learning tasks. Our experimental evaluation of different scenarios indicates the considerable flexibility of the proposed FlexW.

Based on the above analysis, an answer to Subproblem (iv) is obtained. When there is no prior knowledge or theoretical clues for the learning task, a weighting function (e.g., FlexW) that can achieve all four priority modes should be adopted. To select the optimal priority mode for a learning task, grid search can be utilized to search hyperparameters in the hyperparameter intervals corresponding to the four priority modes. Once the optimal hyperparameters are determined, the optimal mode is subsequently obtained. Using meta-learning to learn the hyperparameters is another effective way to optimize the hyperparameters in FlexW [4], which has become a standard technique for hyperparameter selection [42], [43]. The meta-learning strategy of Shu et al. [44] can be followed. Grid search is mainly utilized in our experiments as meta-learning relies on an additional high-quality meta dataset.

IV. Experimental Investigation

Section III performs theoretical investigations of the four subproblems listed in Section I. This section conducts extensive experiments for various learning tasks under different scenarios. Our theoretical analyses are adaptable to all difficulty measures, while $1-p$ is adopted as the difficulty measure in our experiments as with multiple methods do [3], [34]. The experimental results verify the effectiveness and universality of our proposed weighting method FlexW.

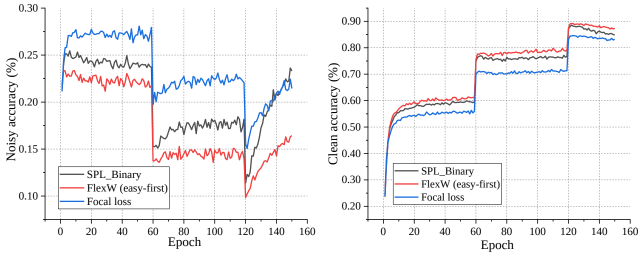


Fig. 4. Accuracy of three methods on noisy (left) and clean (right) samples of CIAFR10 with 40% flip noise.

A. Image Classification with Noisy Labels

1) Experimental settings: Two benchmark image classification datasets, namely, CIFAR10 and CIFAR100 [45], are adopted. Each sample is a 32×32 image from 1 out of 10 or 100 categories. Pair-flip and uniform label noises are simulated following the manner of Shu et al. [1]. Wide ResNet-28-10 (WRN-28-10) [46] and ResNet-32 [47] are adopted as the backbone classifier networks for the flip and uniform noises, respectively. Each experimental run is five times with different seeds for parameter initialization and label noise generation.

The comparison methods include Baseline, which directly trains the backbone network with CE loss; four easy-first methods (SPL (Binary, Log, and Poly modes) [16], [20], [48], SuperLoss [18], R-CE [17], and Information-theoretic loss ($\mathcal{L}_{DMI}$) [49]), and three hard-first methods (Focal loss [3], LOW [19], and Quality Focal loss (QFL) [22]). The weights of all samples are equal to one when the backbone network is trained with CE loss. The networks are trained using SGD with momentum 0.9, weight decay 5×10^{-4}, and initial learning rate 0.1. The batch size is set to 128. The values of γ and α in FlexW for easy/hard/medium/two-ends-first modes are searched in $\{-1, -0.5, -0.4, -0.2\} \times \{0.1, 0.2, 0.3\}$ / $\{0.2, 0.4, 0.5, 1\} \times \{0.1, 0.2, 0.3\}$ / $\{0.2, 0.4, 0.5, 1\} \times \{0.4, 0.6, 0.8\}$ / $\{-1, -0.5, -0.4, -0.2\} \times \{0.4, 0.6, 0.8\}$, respectively.

2) Results: Adding noise changes the difficulty distribution of the training set, resulting in excessive hard samples in the training set relative to the entire space. To analyze the performance of the easy-first and hard-first modes on noisy data, the specific accuracy of SPL-Binary, Focal loss, and FlexW (easy-first) on individual noisy and clean samples are analyzed, as shown in Fig. 4. From the left figure, the schemes with the easy-first mode (i.e., SPL-Binary and FlexW (easy-first)) have

[4] Grid search and meta-learning are also applicable when the prior knowledge for priority mode selection is insufficient.

TABLE III
TEST ACCURACY (%) OF RESNET-32 ON CIFAR10 AND CIFAR100 UNDER FLIP NOISE. THE BEST AND THE SECOND-BEST RESULTS ARE BOLD AND UNDERLINED. * MEANS THE SPL MANNER IS COMBINED INTO FLEXW.

Dataset	Ratio	Baseline	SPL-Binary	SPL-Log	Focal loss	SuperLoss	R-CE	SPL-Poly	LOW	$\mathcal{L}_{DMI}$	QFL	FlexW (hard-first)	FlexW (easy-first)	FlexW (easy-first*)
CIFAR10	20%	76.83	87.03	89.50	86.45	89.19	88.25	88.76	72.77	86.70	79.87	82.25	90.46	**90.96**
	40%	70.77	81.63	84.01	80.45	84.23	83.86	83.98	68.35	84.00	72.87	75.45	85.13	**85.64**
CIFAR100	20%	50.86	63.63	63.82	61.87	63.35	63.48	63.21	50.45	-	51.54	53.55	64.95	**65.48**
	40%	43.01	53.51	53.20	54.13	54.87	54.65	54.72	42.19	-	49.83	45.88	**55.87**	55.50

TABLE IV
TEST ACCURACY (%) OF WRN-28-10 ON CIFAR10 AND CIFAR100 UNDER UNIFORM NOISE.

Dataset	Ratio	Baseline	SPL-Binary	SPL-Log	Focal loss	SuperLoss	R-CE	SPL-Poly	LOW	$\mathcal{L}_{DMI}$	QFL	FlexW (hard-first)	FlexW (easy-first)	FlexW (easy-first*)
CIFAR10	40%	68.07	86.41	77.50	75.96	86.43	85.74	79.34	67.25	85.90	72.37	74.28	87.64	**88.15**
	60%	53.12	53.10	53.40	51.87	79.42	75.56	58.34	51.46	79.60	48.21	52.63	80.56	**81.87**
CIFAR100	40%	51.11	55.11	54.94	51.19	55.64	56.23	56.55	50.25	-	49.49	50.85	**58.48**	57.72
	60%	30.92	36.56	37.17	27.70	41.34	41.05	40.45	36.32	-	26.54	35.22	42.15	**42.89**

TABLE V
PERFORMANCE OF FOUR PRIORITY MODES ACHIEVED BY FLEXW ON CIFAR10 WITH 20% AND 40% FLIP NOISE.

Ratio	Easy-first	Medium-first	Hard-first	Two-ends-first
20%	**90.46**	88.87	82.25	80.13
40%	85.13	**85.81**	75.45	75.06

lower accuracy on noisy samples compared with the hard-first method (i.e., Focal loss) before 140 epochs. From the right figure, SPL-Binary and FlexW (easy-first) consistently outperform Focal loss on clean samples. Therefore, the easy-first methods make the model less disturbed by noise.

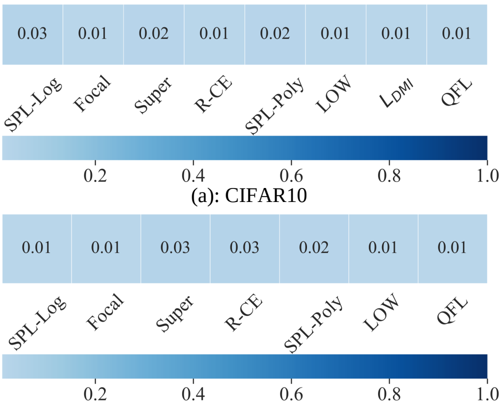
To further reduce the negative influence of quite hard samples in the early training stage, the manner for dealing with quite hard samples in SPL is combined into our FlexW. It further improves the model performance in some cases. The weighting function of FlexW with SPL manner is

$$w_i = \begin{cases} (1-p_i+\alpha)^{\gamma} e^{-\gamma(1-p_i+\alpha)}, & l_i \le \lambda \\ 0, \quad l_i > \lambda \end{cases}, \tag{11}$$

where λ is a threshold defined in SPL [16], which is gradually increased to ensure that more hard samples can participate in the next epochs of training.

Under two types of noise with varying noise ratios, we compare FlexW with some advanced weighting methods in different priority modes, as shown in Tables III and IV. The results of Focal loss are from Shu et al. [1]. FlexW under the easy-first mode achieves the best performance. In addition, the easy-first methods (e.g., SuperLoss and FlexW (easy-first)) perform better than the hard-first ones (e.g., Focal loss and LOW) on noisy data. In some cases, the performance of the hard-first methods is close to or even better than that of the Baseline, mainly due to the imperfect difficulty measure. Alternatively, using only loss to distinguish noisy samples from hard ones is not completely accurate, as discussed by Shin et al. [32], especially for asymmetric label noise. Accordingly, the hard-first priority mode will increase not only the weights of noisy samples but also those of clean hard ones. Increasing the weights of clean hard samples may have a positive influence on model training. If the positive influence is more significant than the negative influence of noisy samples, then the model performance may be improved.

More ablation studies implemented by FlexW with four priority modes are conducted on CIFAR10 with 20% and 40% flip noise. The results are shown in Table V, indicating that the medium-first mode also achieves good performance on noisy data. Therefore, the easy/medium-first modes are more suitable than the hard-first ones for noisy data. The experimental results are consistent with our theoretical analysis for Subproblem (ii), as shown in Sections III-B and III-C.



(a): CIFAR10

(b): CIFAR100

Fig. 5. (a): results of ASO significance test on CIFAR10 with 20% flip noise. (b): results of ASO test on CIFAR100 with 60% uniform noise.

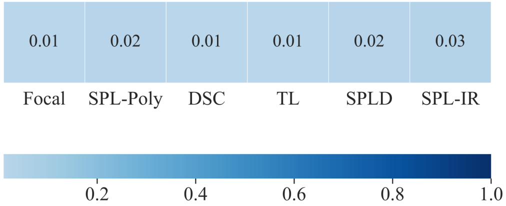


Fig. 6. Results of ASO significance test for the comparison between FlexW (easy-first) and other methods on SST-2 with 20% noise.

TABLE VI
TEST ACCURACY (%) OF COMPETING METHODS ON NOISY SST-2.

Dataset	SST-2	20%	40%	60%
Baseline	93.50	91.58	87.57	69.55
Focal loss	93.57	88.58	84.37	65.27
SPL-Poly	94.16	92.24	88.84	75.36
DSC	94.26	89.48	85.38	66.91
TL	93.65	89.36	85.43	67.24
SPLD	93.68	92.42	88.26	76.12
SPL-IR	93.85	92.58	88.64	75.87
FlexW(easy-first)	94.51	**94.01**	**89.82**	**77.29**
FlexW(medium-first)	94.80	93.14	89.23	76.02
FlexW(hard-first)	**94.91**	89.86	86.03	66.84
FlexW(two-ends-first)	94.75	89.82	84.78	58.58

TABLE VII
TEST ACCURACY (%) ON IMBALANCED CIFAR10 AND CIFAR100 WITH DIFFERENT IMBALANCE FACTORS. * MEANS THE CATEGORY-WISE HYPERPARAMETER IS COMBINED INTO FLEXW.

Dataset	Long-tailed CIFAR10					Long-tailed CIFAR100				
Imbalance factor	200	100	50	20	10	200	100	50	20	10
Baseline	65.68	70.36	74.81	82.23	86.39	34.84	38.32	43.85	51.14	55.71
Focal loss-γ=1	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
Focal loss-γ=0.5	64.00	70.33	76.72	82.89	86.81	35.00	38.69	44.12	51.10	55.70
Focal loss-γ=2	64.88	69.59	76.52	83.23	86.32	34.75	38.39	43.70	51.02	55.00
SPL-Binary	65.64	70.94	76.82	82.41	87.09	35.56	38.16	42.77	50.91	56.70
SPL-Log	62.05	70.46	75.64	82.66	86.62	33.08	38.51	41.71	49.71	54.79
SPL-Poly	64.98	70.56	75.33	82.45	87.02	34.44	38.53	42.78	50.55	56.13
SPL-Linear	62.09	64.85	72.19	80.68	86.32	33.24	37.05	42.38	49.61	55.16
Cost-sensitive SPL	65.84	70.29	74.88	82.45	86.78	35.12	38.39	44.67	51.98	55.37
Class-balanced	68.77	72.68	78.13	84.56	87.90	35.56	38.77	44.79	51.94	57.57
Class-balanced Focal	68.15	74.57	79.22	83.78	87.48	36.23	39.60	45.21	52.59	57.99
LDAM	66.75	73.55	78.83	83.89	87.32	36.53	40.60	46.16	51.59	57.29
SuperLoss	64.75	70.62	75.23	82.45	85.98	34.56	38.46	42.74	50.99	55.28
R-CE	63.98	70.13	75.66	82.23	86.55	34.21	38.05	43.54	49.88	55.35
LOW	67.23	72.78	76.42	83.46	87.55	35.21	38.88	44.64	51.93	55.42
FlexW (easy-first)	64.82	70.96	76.89	82.54	87.12	34.84	38.64	43.72	50.78	56.72
FlexW (hard-first)	69.64	**75.81**	80.13	85.35	**88.86**	**37.81**	41.72	46.98	53.41	58.82
FlexW (hard-first*)	**70.27**	75.33	**80.56**	**85.64**	88.50	37.54	**42.31**	**47.77**	**53.85**	**59.29**

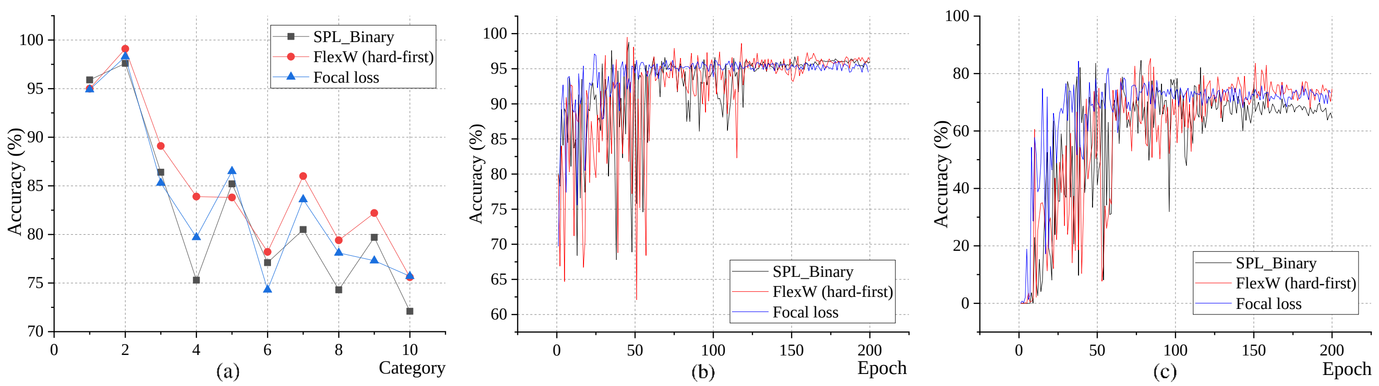
To further demonstrate the superiority of FlexW, the significance test is conducted. The Almost Stochastic Order (ASO) [50] method is used, which returns a value ϵ_{min}, expressing (an upper bound to) the amount of violation of stochastic order. If $\epsilon_{min} < \tau$ (where τ is 0.5 or less), then the corresponding algorithm can be declared as superior. ϵ_{min} can also be interpreted as a confidence score. The lower it is, the more sure the conclusion. The results on noisy CIFAR10 and CIFAR100 datasets are shown in Fig. 5, in which all the obtained ϵ_{min}s for the comparison between FlexW (easy-first) and other methods are smaller than 0.05, indicating that using ASO with a confidence level 0.05, the score distribution of FlexW (easy-first) based on three random seeds is stochastically dominant ($\epsilon_{min} = 0.05$) over other compared methods on noisy CIFAR data.

B. Text Classification with Noisy Labels

1) Experimental settings: The Stanford Sentiment Treebank (SST-2) [51] dataset is adopted. Its training set contains 67,000 movie reviews, and the test set contains 18,000 reviews. The labels of samples are randomly flipped, and the values of the noise ratio are set to 20%, 40%, and 60%, respectively. BERT-base [52] model is applied as the backbone network. The experimental configuration follows that of Devlin et al. [52], and the hyperparameter setting of FlexW is the same as that in Section IV-A. The comparison methods include Baseline, which trains the backbone network with CE loss; three hard-first methods (Focal loss [3], Tversky loss (TL) [53], and Sørensen–Dice Coefficient loss (DSC) [53]), and three easy-first approaches (SPL-Poly [48], Self-paced Learning with Diversity (SPLD) [21], and Self-paced Implicit Regularizer (SPL-IR) [54]).

2) Results: The comparing results are shown in Table VI. The easy-first methods, including SPLD, SPL-Poly, SPL-IR, and FlexW (easy-first), achieve good performance, while the hard-first ones perform poorly. From the ablation studies of FlexW, FlexW (medium-first) also obtains competitive performance. Thus, the easy-first and medium-first modes are more suitable for noisy data. Alternatively, the weights for noisy samples should be decreased.

As with Section IV-A, ASO is employed to demonstrate the significance of FlexW on noisy SST-2 data. The results are shown in Fig. 6, in which all obtained ϵ_{min}s for the comparison between FlexW (easy-first) and other methods are smaller than 0.05, indicating that using ASO with a confidence level 0.05, the score distribution of FlexW (easy-first) based on three random seeds is stochastically dominant ($\epsilon_{min} = 0.05$) over other compared methods on noisy SST-2.



(a) (b) (c)

Fig. 7. (a): the accuracy of the three methods for ten categories on CIFAR10 with an imbalance factor of 20. (b) and (c): the accuracy of the three methods for Categories 1 and 10 on CIFAR10 with an imbalance factor of 20.

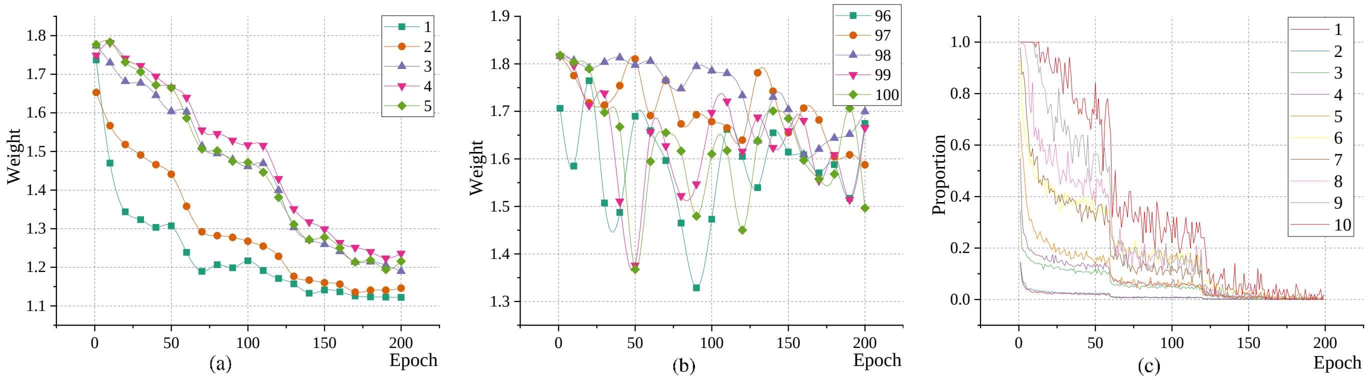


Fig. 8. (a) and (b): the average weights of the first/last five head/tail categories on CIFAR100 with an imbalance factor of 100. (c): the proportion of hard samples in each category in CIFAR10 with an imbalance factor of 100.

C. Image Classification with Imbalanced Datasets

1) Experimental settings: Long-tailed versions of CIFAR benchmarks with different imbalance factors compiled by Cui et al. [2] are used. ResNet-32 [47] is adopted as the backbone. The compared methods include Baseline, which trains the backbone network with CE loss; four hard-first methods (Focal loss [3], Label-Distribution-Aware Margin loss (LDAM) [55], Class-balanced loss [2], and LOW [19]), three easy-first methods (SPL (Binary, Log, Poly, and Linear modes) [16], [20], [48], SuperLoss [18], and R-CE [17]), and an approximated two-ends-first method (Cost-sensitive SPL [37]). Other experimental settings are the same as those in Section IV-A.

2) Results: For imbalanced training data, the proportion of easy samples is larger than that of the entire space, as shown in Fig. 1(b). To study the performance of the hard-first and easy-first priority modes on imbalanced data, the accuracy for each category is analyzed on CIFAR10 with an imbalance factor of 20. Fig. 7(a) indicates that the methods under the hard-first mode (i.e., FlexW (hard-first) and Focal loss) increase the accuracy of most tail categories compared with those under the easy-first mode (i.e., SPL-Binary). For example, FlexW (hard-first) improves the accuracy of tail categories 6, 7, 8, 9, and 10, and Focal loss increases the accuracy of tail categories 5, 7, 8, and 10. Fig. 7(c) shows that the hard-first methods improve the accuracy of the last tail category significantly. Since the first head category contains a large number of samples, all three methods perform well, as shown in Fig. 7(b). Thus, for imbalanced data, an effective weighting scheme should focus on improving the performance of the tail categories, which can be achieved by the hard-first mode.

Inspired by the manner of dealing with the imbalance in Focal loss, a category-wise hyperparameter is introduced in FlexW. The weighting function of FlexW combined with the category-wise hyperparameter is

$$w_i = c_{y_i}(1-p_i+\alpha)^{\gamma}e^{-\gamma(1-p_i+\alpha)}, \tag{12}$$

where c_{y_i} is the category-wise hyperparameter which can further increase the weights for samples in tail categories. It can be set to two for tail categories 6-10 (71-100) and one for other categories in CIFAR10 (CIFAR100), respectively.

Table VII compares some advanced methods under various imbalance factors. The results of Focal loss-γ=1, Class-balanced, Class-balanced Focal, and LDAM are from Li et al. [56]. The performance of the hard-first methods (e.g., FlexW (hard-first(*)), Class-balanced loss, Class-balanced Focal loss) generally surpasses that of the easy-first methods (e.g., SPL and FlexW (easy-first)). In addition, FlexW under the hard-first mode achieves the best results, and the category-wise hyperparameter further improves the model performance in some cases. The performance of SPL is approaching that of Focal loss in some cases. It is because the easy-first methods can improve the accuracy of head categories. However, these methods further enlarge the gap between the head and tail categories which is not desirable.

TABLE VIII
TEST ACCURACY (%) OF COMPETING METHODS ON TWO PI DATASETS.

Dataset	MRPC	QQP
Baseline	88.05	91.42
Focal loss	88.46	91.91
SPL-Poly	86.84	90.38
DSC	88.78	92.27
TL	88.71	92.12
SPLD	85.99	90.85
SPL-IR	86.12	90.74
FlexW(easy-first)	86.78	91.25
FlexW(medium-first)	88.24	92.08
FlexW(hard-first)	**89.73**	**93.35**
FlexW(two-ends-first)	87.32	90.68

Figs. 8(a) and (b) show the average weights of samples in the first five head (a) and last five tail (b) categories of CIFAR100, reflecting the contribution of samples in each category to the model. The weights of the five head categories drop quickly, whereas those of the tail categories remain high during the entire training process. It indicates that the hard-first mode increases the influence of samples in the tail categories on the model. Fig. 8(c) shows the proportion of hard samples (with $l_i \geq$ log10) in each category. The tail categories have larger proportions of hard samples than head ones, which supports the common sense that samples in the tail categories are harder to learn than those in the head on average.

The experimental results are in accordance with our theoretical analysis for Subproblem (ii), as shown in Sections III-B and III-C. The trained model will be underfitting when easy samples are excessive, as shown in Fig. 1(b). To appropriately increase the complexity of the trained model, the hard-first mode should be adopted. Furthermore, the significance of the performance of FlexW is also verified by ASO and indicates that the score distribution of FlexW (hard-first) based on three

TABLE IX
MAPS (%) OF SIX WEIGHTING SCHEMES ON FOUR VOC DATASETS.

Dataset	FL (hard-first)	FL (easy-first)	FlexW (hard-first)	FlexW (easy-first)	FlexW (medium-first)	FlexW (two-ends-first)
VOC-e	75.21	66.96	**76.84**	71.70	73.25	66.88
VOC-h	66.62	68.30	67.67	**69.25**	67.64	65.74
VOC-m	55.74	62.36	60.14	62.71	58.76	**63.25**
VOC-b	61.75	57.72	61.43	59.54	**63.58**	58.66

TABLE X
TEST ACCURACY (%) UNDER THE HARD-FIRST AND VARIED MODES ACHIEVED BY FLEXW ON IMBALANCED CIFAR10 AND CIFAR100.

Dataset	Imbalance	200	100	50	20	10
CIFAR10	Hard-first	69.64	75.81	80.13	85.35	88.86
	Varied mode	**69.98**	**76.23**	**80.71**	84.54	88.00
CIAFR100	Hard-first	37.81	41.72	46.98	53.41	58.82
	Varied mode	37.34	**41.85**	45.25	**53.97**	57.23

random seeds is stochastically dominant ($\epsilon_{min} = 0.05$) over other compared methods on imbalanced CIFAR data using ASO with a confidence level 0.05.

D. Paraphrase Identification with Imbalanced Datasets

1) Experimental settings: Category imbalance is a common data bias in various natural language processing tasks such as Tagging and Machine Reading Comprehension. In this subsection, two Paraphrase Identification (PI) datasets are adopted, namely, Microsoft Research Paraphrase Corpus (MRPC) [57] and Quora Question Pairs (QQP) [58]. MRPC contains sentence pairs automatically extracted from online news, with human annotations of whether the sentence pairs are semantically equivalent. Its category distribution is imbalanced. There are 6,800 pairs in total. 68% of them are positive, and 32% of them are negative. QQP is a collection of question pairs from the community question-answering website Quora. Its category distribution is also imbalanced. There are over 400,000 question pairs in total. 37% of them are positive, and 63% of them are negative. We adopt BERT-base [52] as the backbone network. The experimental configuration of Li et al. [53] is followed. The comparison methods include Baseline, which trains the backbone network with CE loss; three hard-first methods (Focal loss [3], TL [53], and DSC [53]), and three easy-first methods (SPL-Poly [48], SPLD [21], and SPL-IR [54]). The hyperparameter setting of FlexW is the same as that in Section IV-A.

2) Results: The results are shown in Table VIII. The hard-first methods (e.g., DSC and FlexW (hard-first)) perform better than the easy-first ones (e.g., SPLD and FlexW (easy-first)) on imbalanced text data, which is in accordance with our analysis in Sections III.B and III.C. In addition, the results of the ASO significance test manifest that using ASO with a confidence level 0.05, the score distribution of FlexW (hard-first) based on three random seeds is stochastically dominant ($\epsilon_{min} = 0.05$) over other compared methods on QQP data.

E. Object Detection with Different Difficulty Distributions

1) Experimental settings: Dense object detection is a typical application where the distribution of easy and hard samples is imbalanced. PASCAL VOC [59], [60] is utilized whose training set consists of VOC2007 and VOC2012 train and validation sets with a total of 16,551 samples. As the training set contains excessive easy samples [3], it is denoted as VOC-easy (VOC-e). To investigate other difficulty distribution cases, we compiled three training sets based on the training set of VOC: dataset with excessive hard samples (VOC-h), dataset with excessive medium-difficult samples (VOC-m), and dataset with both excessive easy and hard samples (VOC-b). All three artificially constructed training sets contain 8,000 images. For VOC-h, 7,000 images are those with the largest loss-conf values in the original VOC training set. The remaining 1,000 images are randomly selected from the training data except for the hardest 7,000 ones. VOC-m is composed of 8,000 images with moderate loss-conf values in the range of [0.8,1.8]. VOC-b is composed of 4,000 images with the smallest loss-conf values and 4,000 images with the largest loss-conf values in the original VOC training set. VOC2007 test is adopted as the test set with a total of 4,952 samples. YOLOv4 [61] and the weight pre-trained by Darknet [62] are adopted. The optimizer we used is SGD, where the momentum and weight decay are set to 0.9 and 5×10^{-4}. The initial learning rate is 1×10^{-4}, and the final learning rate is 1×10^{-6}. We use a batch size of 4 images. The hyperparameter setting of FlexW is the same as that in Section IV-A.

In this experiment, we reveal an interesting fact that Focal loss (FL) can also implement the easy-first mode when its hyperparameter γ is negative. The performance of the six weighting schemes (FL (easy-first), FL (hard-first), FlexW (easy-first), FlexW (hard-first), FlexW (medium-first), FlexW (two-ends-first)) on the four datasets is shown in Table IX.

2) Results: The two hard-first schemes, including FlexW (hard-first) and FL (hard-first), obtain better results on VOC-e, which contains excessive easy samples. In contrast, when the dataset contains excessive hard samples, the easy-first methods (i.e., FlexW (easy-first) and FL (easy-first)) achieve better performance. For VOC-m, FlexW (two-ends-first) gets the best performance, and FlexW (easy-first) gets the second-best performance. For VOC-b, FlexW (medium-first) and FL (hard-first) obtain the best and second-best performance.

The experimental results are consistent with the four conclusions obtained in our theoretical analysis for Subproblems (i) and (ii), as shown in Section III-B. Furthermore, FlexW achieves competitive performance on object detection tasks.

F. Performance of Varied Priority Mode

As mentioned in Section III-D, if the difficulty distribution of the training set changes drastically during the training process, the corresponding priority mode should also be varied. To demonstrate the varied mode is more effective in some cases, we manually switch the priority mode during the training process. When loss is used as the difficulty measure, the difficulty distribution of the training set will change as training

TABLE XI
TEST ACCURACY (%) OF COMPETING METHODS ON STANDARD CIFAR DATA USING WRN-28-2.

Dataset	Focal loss	SPL	SuperLoss	SPLD	LOW	DIHCL	MCL	DoCL	FlexW (easy-first)	FlexW (hard-first)	FlexW (medium-first)	FlexW (two-ends-first)	FlexW (meta)
CIFAR10	93.89	93.76	94.45	94.37	94.52	94.88	94.34	95.26	94.51	95.84	94.87	93.28	**96.29**
CIFAR100	74.79	75.26	75.75	75.67	74.61	77.05	75.87	77.58	78.14	77.56	77.53	75.99	**78.73**

TABLE XII
TEST ACCURACY (%) OF COMPETING METHODS ON STANDARD CIFAR DATA USING VGG-16.

Dataset	SPL	Inverse-SPL	SPLD	LOW	Focal loss	DIHCL	MCL	DoCL	FlexW (easy-first)	FlexW (hard-first)	FlexW (medium-first)	FlexW (two-ends-first)	FlexW (meta)
CIFAR10	93.07	93.43	93.32	93.90	93.78	93.89	93.45	94.11	93.91	94.72	93.56	92.78	**95.27**
CIFAR100	71.88	71.76	72.51	72.44	71.49	72.81	72.27	73.15	73.76	72.47	72.84	71.34	**74.31**

progresses. Using FlexW, the varied mode is investigated on imbalanced CIFAR data. In the early training stage, most samples have large losses. Thus, the easy-first mode should be utilized. Alternatively, the hyperparameters in FlexW (i.e., α and γ) should be searched in the interval of the easy-first mode. $\gamma = -0.5$ and $\alpha = 0.1$ are employed in the first 100 epochs. With the model trained better, easy samples are excessive because hard samples are mostly samples in tail categories. Thus, the hard-first mode should be adopted to improve the classification performance of the tail categories in later periods. In this stage, the hyperparameters in FlexW should be searched in the intervals of the hard-first mode. $\gamma = 0.5$ and $\alpha = 0.1$ are adopted in the remaining epochs.

Table X shows the performance of FlexW with the varied priority mode and hard-first mode. The results indicate that the varied mode achieves better performance than the fixed mode in some cases, which is in accordance with our analysis in Section III-D. The priority mode does not need to remain fixed, and it should be varied if the difficulty distribution of the training set changes significantly.

G. Image Classification with Standard Datasets

1) Experimental settings: Standard CIFAR10 and CIFAR100 [45] datasets are utilized in this experiment. The five hard-first methods (Focal loss [3], LOW [19], Curriculum Learning by Dynamic Instance Hardness (DIHCL) [63], Minimax curriculum learning (MCL) [64], and Curriculum Learning by Optimizing Learning Dynamics (DoCL) [65]), the three easy-first methods (SPL [16], SuperLoss [18], and SPLD [21]), our proposed FlexW (easy-first, hard-first, medium-first, and two-ends-first), and FlexW with meta-learning are compared on WRN-28-2 [46], as shown in Table XI. In addition, the two easy-first methods (SPL-Binary [16] and SPLD [21]), the six hard-first methods (Inverse-SPL [66], LOW [19], Focal loss [3], DIHCL [63], MCL [64], and DoCL [65]), FlexW (easy-first, hard-first, medium-first, and two-ends-first), and FlexW with meta-learning are compared on VGG-16 [67], as shown in Table XII. For VGG-16, we adopt a slightly modified version [67], which contains only one fully-connected layer. Experimental settings are the same as those in Section IV-A. For FlexW with meta-learning, the construction of meta data follows the manner of Shu et al. [1].

2) Results: Tables XI and XII show that there is no clear judgment among all the priority modes on standard data. For example, the hard-first and easy-first modes perform better on standard CIFAR10 and CIFAR100, respectively. Our proposed FlexW outperforms three advanced CL methods, including DIHCL, MCL, and DoCL. In addition, using meta-learning to optimize the hyperparameters in FlexW can obtain better performance. The experimental results are consistent with the answer for Subproblem (iv), which is stated in Section III.E. Weighting methods that can implement all four priority modes (e.g., FlexW) should be utilized when there is no prior knowledge or theoretical clues. Grid search and meta-learning strategies can be adopted to select the optimal hyperparameters. Once the optimal hyperparameters are determined, the optimal priority mode can be obtained.

TABLE XIII
HYPERPARAMETER INTERVALS CORRESPONDING TO FOUR PRIORITY MODES OF FLEXW WITH STABLE PERFORMANCE.

Priority mode	Easy-first	Medium-first	Hard-first	Two-ends-first
Interval	[-0.6,-0.4]×[0.1,0.3]	[0.4,0.6]×[0.4,0.6]	[0.4,0.6]×[0.1,0.3]	[-0.6,-0.4]×[0.4,0.6]

H. Selection of Hyperparameters in FlexW

For the convenience of application, we give the hyperparameter intervals with stable performance corresponding to the four priority modes, as shown in Table XIII. If the optimal priority mode can be pre-determined, grid search can be used to search hyperparameters within specific intervals.

When no prior knowledge exists, both grid search and meta-learning methods can be utilized to select the optimal hyperparameters (i.e., priority modes), as stated in Section III.E. It is worth noting that FlexW can be flexibly switched among the four priority modes only by changing its hyperparameters. Alternatively, when facing a new learning task, only the values of hyperparameters in FlexW are needed to be changed without changing the entire weighting function.

V. ANSWERS AND DISCUSSIONS

According to the aforementioned theoretical analyses and empirical observations, a comprehensive answer is obtained for our investigated “easy-or-hard” question:

- No universal fixed optimal priority mode exists for an arbitrary learning task.
- Except for the easy-first and hard-first priority modes, there are two other typical priority modes, namely, medium-first and two-ends-first.
- The weights for noisy samples should be decreased. Thus, the priority modes of easy-first and medium-first are more effective on noisy data.
- The relationship between the difficulty distribution and the priority mode can be analyzed based on Eq. (3). Samples with large/small relative difficulty should be assigned with high/low weights. Thus, four conclusions

can be obtained: (1) If there are excessive easy samples in the training data, the hard-first mode should be adopted. (2) If there are excessive hard samples, the easy-first mode should be adopted. (3) If there are excessive medium-difficult samples, the two-ends-first mode should be adopted. (4) If there are both excessive easy and hard samples, the medium-first mode should be adopted.

- The priority mode is not necessary to be fixed during training. If the difficulty distribution of the training data changes significantly during the training process, the priority mode should also be varied.
- If there is no prior knowledge or theoretical clues, the weighting schemes (e.g., FlexW) which can achieve all four priority modes should be adopted. Which mode is more appropriate depends on the results of the hyperparameter tuning.

The above answer indicates that the measurement of the learning difficulty of samples is crucial as the weight is directly determined by the difficulty distribution of the training set. In most existing studies, the learning difficulty is approximated by the loss (or the predicted probability of ground truth) [3], [16], which is shown in Table II. Although reasonable, an ideal solution should fully consider factors, including loss, the neighbor of samples, category distribution, and noise level. This study will be the focus of our future work.

Another critical issue is the judgment of whether easy/medium/hard samples are excessive in the training set. Prior knowledge of the training set, such as noise level and category proportion, can help us with this issue. If there is not any prior knowledge but the learning difficulty of each sample is accessible, the excess of easy/medium/hard samples should be judged according to the difference between the difficulty distributions of the entire sample space and the training dataset. It is impractical to utilize the distribution of the entire sample space. A feasible way is to take the difficulty distribution of the validation set as a reference, as the validation set is generally regarded as unbiased. If which kinds of samples are excessive can not be determined, weighting strategies that can achieve all four modes (e.g., FlexW) should be adopted. Both the grid search and meta-learning methods can be utilized to select the optimal hyperparameters of FlexW.

VI. Conclusions

This study focuses on an interesting and important question about the choices of priority modes for learning tasks. A deep investigation of this question facilitates understanding various existing weighting schemes and choosing an appropriate priority mode for a new learning task. First, a general objective function is proposed, which offers an explanation of the relationship between the difficulty distribution of the training set and the priority mode. This objective function provides a comprehensive view to analyze the "easy-or-hard" question theoretically. Second, several theoretical answers are obtained based on the general objective function. Two other typical priority modes, namely, medium-first and two-ends-first, are revealed. In addition, the priority mode is not necessary to be fixed during the training process. Third, an effective and universal weighting solution (i.e., FlexW) is proposed when there is no prior knowledge or theoretical clues. This solution alleviates the defects of existing schemes that only single or partial priority mode(s) can be implemented. Fourth, extensive experiments for various tasks are conducted under different data characteristics. Finally, a comprehensive answer to the "easy-or-hard" question is obtained according to the theoretical analysis and empirical evaluation.

Appendix

A. Theoretical Analysis for Propositions 1 and 2

A strict proof for Proposition 1 is challenging. We give the proof under a special case that the weights exerted on R_{easy} are identical. Without loss of generality, the weights on each sample in R_{hard} are denoted as $(1+\epsilon)$, where $\epsilon > 0$.

Let $BiasT(c)$ and $VarT(c)$ be the bias and variance terms defined in Eq. (5) in Section III-C, respectively. Minimum learning error is achieved when the sum of the partial derivatives of the two terms with respect to the model complexity c is equal to zero. Let c^* be the optimal model complexity when the minimum learning error is achieved. We yield

$$\left.\frac{\partial Err}{\partial c}\right|_{c^*} = \left.\frac{\partial BiasT(c)}{\partial c}\right|_{c^*} + \left.\frac{\partial VarT(c)}{\partial c}\right|_{c^*} = 0. \tag{13}$$

According to Assumptions 1 and 2, we have

$$\begin{aligned} \left.\frac{\partial BiasT_{easy}(c)}{\partial c}\right|_{c^*} + \left.\frac{\partial VarT_{easy}(c)}{\partial c}\right|_{c^*} > 0, \\ \left.\frac{\partial BiasT_{hard}(c)}{\partial c}\right|_{c^*} + \left.\frac{\partial VarT_{hard}(c)}{\partial c}\right|_{c^*} < 0. \end{aligned} \tag{14}$$

Let p_{easy}, p_{medium}, and p_{hard} be the probabilities that a random sample coming from R_{easy}, R_{medium}, and R_{hard}, respectively. Naturally, $p_{easy} + p_{medium} + p_{hard} = 1$. According to the law of total expectation, the error in the entire sample space can be split into three regions and calculated separately. Thus, we yield

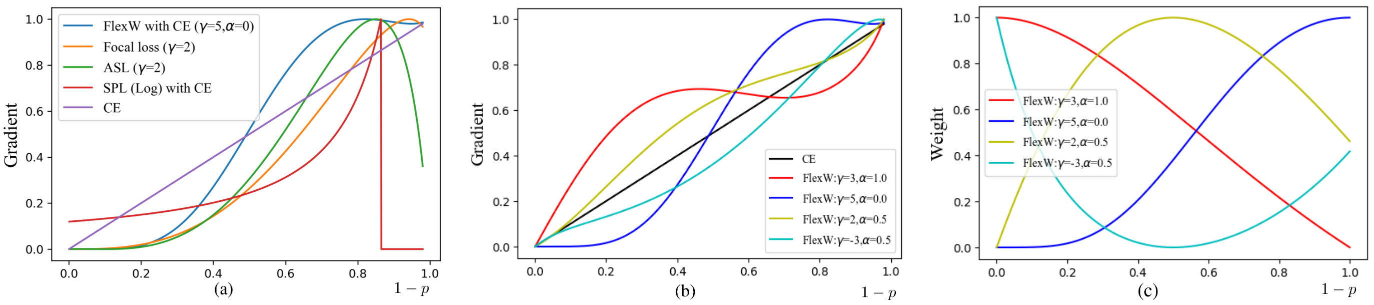
$$\begin{aligned} Err(c^*) &= p_{easy}Err_{easy}(c^*) + p_{medium}Err_{medium}(c^*) \\ &\quad + p_{hard}Err_{hard}(c^*), \\ BiasT(c^*) &= p_{easy}BiasT_{easy}(c^*) + p_{medium}BiasT_{medium}(c^*) \\ &\quad + p_{hard}BiasT_{hard}(c^*), \\ VarT(c^*) &= p_{easy}VarT_{easy}(c^*) + p_{medium}VarT_{medium}(c^*) \\ &\quad + p_{hard}VarT_{hard}(c^*). \end{aligned} \tag{15}$$

When the weights $(1+\epsilon)$ are exerted on samples in R_{hard}, then $BiasT(c^*)$ and $VarT(c^*)$ become

$$\begin{aligned} BiasT_{\epsilon}(c^*) &= p_{easy}BiasT_{easy}(c^*) + p_{medium}BiasT_{medium}(c^*) \\ &\quad + p_{hard}BiasT_{hard} + \epsilon p_{hard}BiasT_{hard}(c^*), \\ VarT_{\epsilon}(c^*) &= p_{easy}VarT_{easy}(c^*) + p_{medium}VarT_{medium}(c^*) \\ &\quad + p_{hard}VarT_{hard}(c^*) + \epsilon p_{hard}VarT_{hard}(c^*). \end{aligned} \tag{16}$$

Based on Eqs. (14) and (16), we have

$$\begin{aligned} &\left.\frac{\partial BiasT_{\epsilon}(c)}{\partial c}\right|_{c^*} + \left.\frac{\partial VarT_{\epsilon}(c)}{\partial c}\right|_{c^*} \\ &= 0 + \epsilon p_{hard}\left(\left.\frac{\partial VarT_{hard}(c)}{\partial c}\right|_{c^*} + \left.\frac{\partial BiasT_{hard}(c)}{\partial c}\right|_{c^*}\right) < 0. \end{aligned} \tag{17}$$



(a) (b) (c)

Fig. 9. (a): loss gradients of different weighting methods. (b) loss gradients of FlexW with four priority modes. (c) weights curves of FlexW with four priority modes.

Accordingly, the model complexity should be increased to attain the new balance between the bias and variance terms. Alternatively, the new optimal model complexity c^*_{new} will be larger than c^*. The proof for Proposition 2 is with a similar inference manner.

B. Gradient Analysis of FlexW

The weighting strategies affect the training process by influencing the loss gradients of samples. To better understand FlexW, the gradient of the loss function with FlexW is analyzed. The loss gradient of CE loss with FlexW is:

$$\begin{aligned}\frac{d\mathcal{L}}{dz} &= \frac{\partial \mathcal{L}}{\partial p}\times\frac{\partial p}{\partial z}\\ &= p(1-p)(1-p+\alpha)^{\gamma-1}e^{-\gamma(1-p+\alpha)}\times\\ &(\gamma\log p(p-\alpha)-\frac{1-p+\alpha}{p}),\end{aligned} \tag{18}$$

where $p=\frac{1}{1+e^{-z}}$. The gradient of CE loss with FlexW is in comparison to the gradients of CE loss, Focal loss [3], CE loss with SPL-Log [20], and ASL [34].

Fig. 9(a) shows the gradients of different weighting strategies. Under CE loss, harder samples have larger gradients than easier ones. Focal loss increases the gradients of hard samples. However, it is sensitive to noise. ASL decreases the gradients of quiet-hard samples. Fig. 9(b) shows the gradients of the four variants of FlexW with CE loss. The weight curves of the four variants are shown in Fig. 9(c). When the easy/medium/hard/two-ends-first mode of FlexW is utilized, the loss gradients of easy/medium/hard/both easy and hard samples are increased compared with those under CE loss.

References

[1] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *Proc. NIPS*, pp. 1–23, 2019.

[2] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie, "Class-balanced Loss Based on Effective Number of Samples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 9260–9269, 2019.

[3] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2999–3007, 2017.

[4] X. Wu, E. Dye,r and B. Neyshabur, "When Do Curricula Work?," in *Proc. ICLR*, pp. 1–23, 2021.

[5] A.-D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, 2018.

[6] Y. Artan, D.-L. Langer, M.-A. Haider, T.-H. van der Kwast, A.-J. Evans, M.-N. Wernick, and I.-S. Yetik, "Prostate Cancer Localization With Multispectral MRI Using Cost-Sensitive Support Vector Machines and Conditional Random Fields," in *IEEE Trans. on Image Processing*, vol. 19, no. 9, pp. 2444–2455, 2010.

[7] H. Song, M. Kim, D. Park, Y. Shin, and J-G. Lee, "Learning From Noisy Labels With Deep Neural Networks: A Survey," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–19, 2022.

[8] B. Li, Y. Liu, and X. Wang, "Gradient Harmonized Single-stage Detector," in *Proc. AAAI*, pp. 8577–8584, 2019.

[9] B. Frenay, and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, 2013.

[10] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proc. AAAI*, pp. 3438–3445, 2020.

[11] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware Instance-reweighted Adversarial Training," in *Proc. ICLR*, pp. 1–29, 2021.

[12] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019.

[13] K. R. M. Fernando and C. P. Tsokos, "Dynamically Weighted Balanced Loss: Class Imbalanced Learning and Confidence Calibration of Deep Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2162–2388, 2021.

[14] D. Arpit, et al, "A closer look at memorization in deep networks," in *Proc. ICML*, pp. 350–359, 2017.

[15] X. Wang, Y. Chen, and W. Zhu, "A Survey on Curriculum Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 1, pp. 1–20, 2021.

[16] M. Pawan Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. NIPS*, pp. 1–9, 2010.

[17] W. Wang, F. Feng, X. He, L. Nie, and T. Chua, "Denoising Implicit Feedback for Recommendation," in *Proc. WSDM*, pp. 373–381, 2021.

[18] T. Castells, P. Weinzaepfel, and J. Revaud, "SuperLoss: A generic loss for robust Curriculum Learning," in *Proc. NIPS*, pp. 1–12, 2020.

[19] C. Santiagoa, C. Barataa, M. Sasdellib, G. Carneirob, and J. C.Nasciment, "LOW: Training deep neural networks by learning optimal sample weights," *Pattern Recognition*, vol. 110, no. 1, pp. 1–12, 2021.

[20] L. Jiang, D. Meng, T. Mitamural, and A. G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *Proc. MM 2014*, pp. 547–556, 2014.

[21] L. Jiang, D. Meng, S. Yu, Z. Lan, S. Shan, and A.-G. Hauptmann, "Self-Paced Learning with Diversity," in *Proc. NIPS*, pp. 2078–2086, 2014.

[22] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," *arXiv:2006.04388*, 2020.

[23] Y. Freund, and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *Proc. ICML*, pp. 1–9, 1996.

[24] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun, "Distribution Alignment: A Unified Framework for Long-tail Visual Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2361–2370, 2021.

[25] Y. Bengio, J. Louradour, R. Collobert, J. Weston, "Curriculum Learning," in *Proc. ICML*, pp. 41–48, 2009.

[26] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A.-G. Hauptmann, "Self-paced Curriculum Learning," in *Proc. AAAI*, pp. 2694–2700, 2015.

[27] P. Soviany, "Curriculum Learning with Diversity for Supervised Computer Vision Tasks," in *Proc. ECAI*, pp. 37–44, 2020.

[28] E. Z. Liu, B. Haghgoo, A. S. Chen, A. Raghunathan, P. W. Koh, S. Sagawa, P. Liang, and C. Finn, "Just Train Twice: Improving Group Robustness without Training Group Information," in *Proc. ICML*, pp. 6781–6792, 2021.

[29] P. Soviany, R.-T. Ionescu, P. Rota, and N. Sebe, "Curriculum Learning: A Survey," *arXiv:2101.10382*, 2021.

[30] D. Zhang, D. Meng, C. Li, L. Jiang, Q. Zhao, and J. Han, "A self-paced multiple-instance learning framework for co-saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 594–602, 2015.

[31] M. Zhang, Y. Luo, Z. Wang, H. Qin, W. Zhao, T. Liu, "Automatic Digital Modulation Classification Based on Curriculum Learning," *Applied Sciences*, vol. 9, no. 1, pp. 1–14, 2019.

[32] W. Shin, J. Ha, S. Li, Y. Cho, H. Song, and S. Kwon, "Which Strategies Matter for Noisy Label Classification? Insight into Loss and Uncertainty," *arXiv:2008.06218*, 2020.

[33] S. H. Khan, M. Hayat, M. Bennamoun, F. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, 2018.

[34] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric Loss For Multi-Label Classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 82–91 2021.

[35] E. Aguilar, B. Nagarajan, R. Khatun, M. Bolanos, and P. Radeva, "Uncertainty modeling and deep learning applied to food image analysis," in *Proc. BIOSTEC*, pp. 3–16, 2020.

[36] T. V. Erven, and P. Harremos, "Renyi Divergence and Kullback-Leibler Divergence," *IEEE Trans on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[37] J. Yang, X. Wu, J. Liang, X. Sun, M. Cheng, P. L. Rosin, and L. Wang, "Self-Paced Balance Learning for Clinical Skin Disease Recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2832–2846, 2020.

[38] T. Heskes, "Bias/Variance Decompositions for Likelihood-Based Estimators," *Neural Computation*, vol. 10, no. 6, pp. 1425–1433, 1998.

[39] Z. Yang, Y. Yu, C. You, J. Steinhardt, and Y. Ma, "Rethinking Bias-Variance Trade-off for Generalization of Neural Networks," in *Proc. ICML*, pp. 10698–10708, 2020.

[40] P. Domingos, "A Unified Bias-Variance Decomposition for Zero-One and Squared Loss," in *Proc. AAAI*, pp. 564–569, 2000.

[41] G. Hacohen, and D. Weinshall, "On The Power of Curriculum Learning in Training Deep Networks," in *Proc. ICML*, pp: 2535–2544, 2019.

[42] J. Snoek, H. Larochelle, and R.-P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proc. NIPS*, pp. 2951–2959, 2012.

[43] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *Proc. ICML*, pp. 2113–2122, 2015.

[44] J. Shu, D. Meng, and Z. Xu, "Meta self-paced learning," *Scientia Sinica Informationis*, vol. 50, no. 6, pp. 781–793, 2020.

[45] A. Krizhevsky, *Learning multiple layers of features from tiny images*, MIT Press, USA, 2009.

[46] S. Zagoruyko, and N. Komodakis, "Wide Residual Networks," in *Proc. BMVC*, vol. 87, pp. 1–12, 2016.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

[48] M. Gong, H. Li, D. Meng, Q. Miao, and J. Liu, "Decomposition-based evolutionary multiobjective optimization to self-paced learning," *IEEE Trans. Evol*, vol. 23, no. 2, pp. 288–302, 2018.

[49] Y. Xu, P. Cao, Y. Kong, and Y. Wang, "L-dmi: An information-theoretic noise-robust loss function," *arXiv:1909.03388*, 2019.

[50] R. Dror, S. Shlomov, and R. Reichart, "Deep Dominance - How to Properly Compare Deep Neural Models," in *Proc. ACL*, pp. 2773–2785, 2019.

[51] R. Socher, A. Perelygin, J.-Y. Wu, J. Chuang, C. Manning, A.-Y. Ng, C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, pp. 1631–1642, 2013.

[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL*, pp. 4171–4186, 2019.

[53] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced NLP tasks," in *Proc. ACL*, pp. 465–476, 2020.

[54] Y. Fan, R. He, J. Liang, and B. Hu, "Self-Paced Learning: an Implicit Regularization Perspective,", in *Proc. AAAI*, pp. 1–12, 2017.

[55] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss," in *Proc. NIPS*, pp. 1567–1578, 2019.

[56] S. Li, K. Gong, C.-H. Liu, Y. Wang, F. Qiao, and X. Cheng, "MetaSAug: Meta Semantic Augmentation for Long-Tailed Visual Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5208–5217, 2021.

[57] W.-B. Dolan, C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proc. IWP*, pp. 1–8, 2005.

[58] Z. Chen, H. Zhang, X. Zhang, L. Zhao, "Quora question pairs," in *University of Waterloo*, pp. 1–7, 2018.

[59] E. Mark, G. Luc, C.-K.-I. Williams, W. John, and Z. Andrew, "The PASCAL Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 5, no. 1, pp. 329–359, 1996.

[60] E. Mark, E.-S.-M. AliVan, G. Luc, C.-K.-I. Williams, W. John, and Z. Andrew, "The PASCAL Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[61] A. Bochkovskiy, C. Wang, and H.-M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934*, 2020.

[62] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 'You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 779–788, 2016.

[63] T. Zhou, S. Wang, and J. Bilmes, "Curriculum Learning by Dynamic Instance Hardness," in *Proc. NIPS*, pp. 8602–8613, 2020.

[64] T. Zhou, and J. Bilmes, "Minimax Curriculum Learning: Machine Teaching with Desirable Difficulties and Scheduled Diversity," in *Proc. ICLR*, pp. 1–15, 2018.

[65] T. Zhou, S. Wang, J. Bilmes, "Curriculum Learning by Optimizing Learning Dynamics," in *Proc. AISTATS*, pp. 433–441, 2021.

[66] H. Cheng, D. Lian, B. Deng, S. Gao, T. Tan, and Y. Geng, "Local to global learning: Gradually adding classes for training deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4743–4751, 2019.

[67] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets," in *Proc. ICLR*, pp. 1–12, 2016.

Xiaoling Zhou received the B.Sc. degree in Mathematics from Tiangong University, Tianjin, China, in 2016. She is currently a Postgraduate at the Center for Applied Mathematics, Tianjin University, Tianjin, China, under the supervision of Professor Ou Wu. Her research interests include data mining and deep learning.

Ou Wu received the B.Sc. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003, and the M.Sc. and Ph.D. degrees in computer science from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2006 and 2012, respectively. In 2007, he joined NLPR as an Assistant Professor. In 2017, he became a Full Professor at the Center for Applied Mathematics, Tianjin University, China. His research interests include data mining and machine learning.