Batch Selection for Multi-Label Classification Guided by Uncertainty and Dynamic Label Correlations

Ao Zhou[1], Bin Liu [1 *], Jin Wang[1], Grigorios Tsoumakas[2]

[1] Key Laboratory of Data Engineering and Visual Computing, School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China
[2] School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
zacqupt@gmail.com, {liubin, wangjin}@cqupt.edu.cn, greg@csd.auth.gr

Abstract

The accuracy of deep neural networks is significantly influenced by the effectiveness of mini-batch construction during training. In single-label scenarios, such as binary and multi-class classification tasks, it has been demonstrated that batch selection algorithms preferring samples with higher uncertainty achieve better performance than difficulty-based methods. Although there are two batch selection methods tailored for multi-label data, none of them leverage important uncertainty information. Adapting the concept of uncertainty to multi-label data is not a trivial task, since there are two issues that should be tackled. First, traditional variance or entropy-based uncertainty measures ignore fluctuations of predictions within sliding windows and the importance of the current model state. Second, existing multi-label methods do not explicitly exploit the label correlations, particularly the uncertainty-based label correlations that evolve during the training process. In this paper, we propose an uncertainty-based multi-label batch selection algorithm. It assesses uncertainty for each label by considering differences between successive predictions and the confidence of current outputs, and further leverages dynamic uncertainty-based label correlations to emphasize instances whose uncertainty is synergistically expressed across multiple labels. Empirical studies demonstrate the effectiveness of our method in improving the performance and accelerating the convergence of various multi-label deep learning models.

Introduction

Multi-label classification (MLC) involves learning from instances associated with multiple labels simultaneously. Its goal is to derive a model capable of assigning a relevant set of labels to unseen instances. For example, a news document might cover various topics in text categorization (Chai et al. 2024; Jiang et al. 2021); an image could contain annotations for different scenes (Zhou, Huang, and Xing 2021; Nguyen, Vu, and Le 2021), and a video may consist of multiple different clips (Gupta et al. 2023; You et al. 2020). For classifying such complex scenarios, multi-label learning approaches are seen as viable solutions for handling data with multiple labels.

*Corresponding author

Deep learning has recently proven successful in learning from multi-label data (Liu et al. 2021). By forming appropriate latent embedding spaces, deep neural networks manage to unravel the complex dependencies between features and labels in multi-label data (Yeh et al. 2017; Bai, Kong, and Gomes 2021). Moreover, deep learning models can successfully dissect and analyze label correlations (Hang and Zhang 2021; Zhao et al. 2021). In addition, their inherent strength in representation learning allows them to naturally model label-specific features (Hang et al. 2022).

Recent studies highlight the critical role of mini-batch sample selection in the performance of deep neural networks (DNNs). Training with simple examples (Kumar, Packer, and Koller 2010) can enhance robustness against outliers and noisy labels, but their smaller loss and gradients lead to slower model convergence. Conversely, focusing on instances that are difficult to predict correctly (Liu et al. 2023; Huang et al. 2020; Shrivastava, Gupta, and Girshick 2016) accelerates training, but overemphasizing the losses of hard examples may lead to overfitting on noisy data. Uncertainty-based batch selection, exemplified by *Active Bias* (Chang, Learned-Miller, and McCallum 2017), is a compromise solution that prioritizes uncertain samples—those with unstable predictions, whether correct or incorrect, during the training process—thereby expediting model convergence while mitigating the risk of overfitting. *Recency Bias* (Song et al. 2020a) evaluates instance uncertainty within recent observations rather than the entire history, offering a more dynamic assessment.

While batch selection methods are well-studied in single-label tasks (binary or multi-class classification), their effectiveness in multi-label data is less explored. *Balanced* (Hand, Castillo, and Chellappa 2018) is a multi-label batch selection method that maintains label distributions in each batch consistent with the whole dataset via a re-weighting strategy. However, it only relies on the prior label distribution, neglecting losses or predictions of instances within each step of the training procedure. *Hard Imbalance* (Zhou et al. 2024) prioritizes hard samples associated with more highly imbalanced (low-frequency) labels during multi-label batch selection, but it also suffers the risk of overfitting due to overemphasizing difficult instances.

This paper adapts the uncertainty-based batch selection strategy to multi-label data. To achieve this, there are two is-

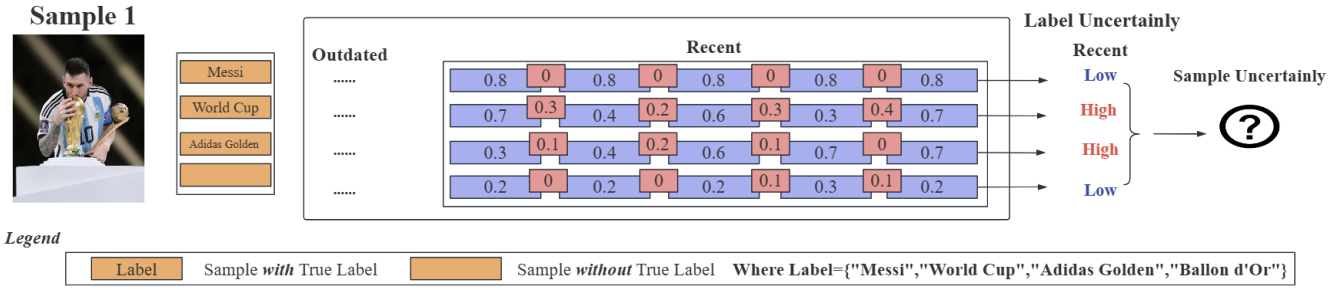


Figure 1: Sample 1 contains "Messi" "World Cup" and "Adidas Golden Ball".

sues that need to be addressed. First, traditional variance and entropy-based measures (Chang, Learned-Miller, and McCallum 2017; Song et al. 2020a) for assessing label-wise uncertainty ignore fluctuations within sliding windows, i.e., the changes between successive predictions, and the confidence of the current prediction. As the example shown in Fig 1. *Active Bias* and *Recent Bias* fail to distinguish the uncertainty of the labels "World Cup" and "Adidas Golden Ball" [1]. Secondly, directly summing all label uncertainties overlooks the correlation between labels. These inter-label uncertainties contain valuable information, reflecting the model's ability to collaboratively learn and predict highly correlated labels. However, previous multi-label batch selection methods have not accounted for these dynamic, uncertainty-based label correlations (Hand, Castillo, and Chellappa 2018; Zhou et al. 2024).

To tackle the two issues, we proposed an uncertainty-based multi-label batch selection method, which considers both current confidence and fine-grained variance in sliding windows, and leverages dynamic label correlations to emphasize the importance of uncertain samples during the training. Specifically, we propose a new absolute difference-based measure to average the changes between adjacent predictions within the sliding window for each label, which reflects the reliabilities of the current prediction and fine-grained fluctuations between successive predictions. Based on individual label uncertainty, we derive the dynamic uncertainty-based label correlations in each epoch, and prioritize samples whose uncertainty is synergistically expressed in more labels for mini-batch selection.

The main contributions of this paper are as follows:

- **Label Uncertainty Estimation:** Our method provides a comprehensive assessment that integrates both present uncertainty and fine-grained changes in recent predictions to evaluate uncertainty for each label.
- **Sample Uncertainty Estimation:** Our method leverages dynamic uncertainty-based label correlations to guide the sample uncertainty assessment, emphasizing instances with higher uncertainty synergistically expressed in more labels in each epoch.
- **Effectiveness and Universality:** Our method achieves the most performance improvement compared with five competitors. In addition, the superiority of our method remains consistent across various deep multi-label learning models and datasets from different domains.

[1]Please refer to section 3.1 for an explanation of the reasons.

Related Work

Multi-Label Classification

Initially, multi-label classifiers adapt conventional machine learning techniques, such as neighborhood-based classifier (Zhang and Zhou 2007), decision tree (Wu et al. 2016), and kernel method (Chen et al. 2016), to handle multi-label data. Alternatively, another solution converts multi-label classification into multiple single-label problems, which are solved by well-studied single-label models (Zhang and Zhou 2013). Representative strategies include individual label (Boutell et al. 2004), label pair (Zhang et al. 2020), label subset (Tsoumakas, Katakis, and Vlahavas 2010), and label chain (Liu and Tsoumakas 2020)-based transformations.

Recently, deep neural networks (DNNs) have emerged as a highly successful technique for solving multi-label classification tasks. Deep embedding-based methods effectively align feature and label spaces using DNNs. *C2AE* (Yeh et al. 2017) embeds features and labels into a deep latent space with a label-correlation sensitive loss function. *MP-VAE* (Bai, Kong, and Gomes 2021) aligns probabilistic embeddings of labels and features, using a decoder to model their joint distribution. While others focus on capturing label correlations or learning label-specific features. *PACA* (Hang et al. 2022) learns label prototypes and metrics in a latent space regulated by label correlations. *HOT-VAE* (Zhao et al. 2021) uses attention to capture high-order label correlations adaptively. *CLIF* (Hang and Zhang 2021) integrates label semantics with label-specific feature extraction using a graph autoencoder, and *DELA* (Hang and Zhang 2022) employs perturbation-based techniques for stable label-specific features within a probabilistic framework. All deep multi-label classification models utilize randomly selected mini-batches to optimize the model, which fails to emphasize the crucial instances during the learning procedure.

Batch Selection

Recent research emphasizes that the performance of DNNs depends on the selection of mini-batch samples (Shrivastava, Gupta, and Girshick 2016; Katharopoulos and Fleuret 2018;

Method	Criteria	Uncertainty Measure	Label correlation	Datatype
Active Bias (Chang, Learned-Miller, and McCallum 2017)	uncertainty	variance of entire prediction history	-	single-label
Recent Bias (Song et al. 2020a)	uncertainty	entropy of recent predictions	-	single-label
Balance (Hand, Castillo, and Chellappa 2018)	imbalance	✗	✗	multi-label
Hard Imbalance (Zhou et al. 2024)	hardness & imbalance	✗	✗	multi-label
Ours	uncertainty	fine-grained fluctuations of recent windows and current prediction	✓	multi-label

Table 1: The summary of uncertainty-based batch selection methods for single-label and multi-label batch selection approaches.

Song et al. 2020b; Chang, Learned-Miller, and McCallum 2017). Batch selection has been used in various learning strategies such as reinforcement learning (Fan et al. 2016), curriculum learning (Bengio et al. 2009), and active learning (Chen et al. 2022; Chakraborty, Balasubramanian, and Panchanathan 2011), as well as in different learning tasks like classification (Loshchilov and Hutter 2016; Song et al. 2020b) and sample labeling (Chen et al. 2022).

Sample difficulty plays a crucial role in mini-batch selection. Two opposing strategies—preferring easy or hard samples—are effective in different scenarios. Prioritizing easy samples helps resist outliers and noisy labels but may slow training due to smaller gradients (Kumar, Packer, and Koller 2010; Song, Kim, and Lee 2019). In contrast, focusing on hard samples accelerates training but can cause overfitting and poor generalization (Loshchilov and Hutter 2016). Additionally, some heuristic batch selection methods have proven effective in single-label datasets. *Ada-Boundary* (Song et al. 2020b) focuses on moderately challenging samples near the decision boundary to optimize learning progress. *Active Bias* (Chang, Learned-Miller, and McCallum 2017) uses uncertainty-based sampling, prioritizing uncertain samples for the next batch. It maintains a history queue storing all previous predictions and measures uncertainty by computing the prediction variance. *Recency Bias* (Song et al. 2020a) also measures the variance of recent predictions within a fixed-sized sliding window, eliminating the impact of outdated predictions on the uncertainty estimation. For multi-label data, *Balance* (Hand, Castillo, and Chellappa 2018) adjusts batches to match desired label distributions, balancing over- and under-represented labels by sampling and weighting instances. *Hard Imbalance* (Zhou et al. 2024) prioritizes samples with high losses and imbalanced labels based on cross-entropy loss and label imbalance. In Table 1, we summarize several SOTA batch selection methods in single-label or multi-label data.

Proposed Method

In this section, we first compute the uncertainty for each label from both the current epoch and the recent historical window perspectives. Next, we derive the sample uncertainty by considering the dynamic uncertainty-based label correlation. Finally, we select samples for the next batch based on their uncertainty-based weights.

Problem Formulation

Let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)|_{i=1}^n\}$ be a multi-label dataset containing n instances, where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iq}] \in \{0,1\}^q$ are the feature and label vectors of i-th instance, respectively. Let $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ be the label set, $y_{ij} = 1$ indicates i-th instance relevant to l_j and $y_{ij} = 0$ otherwise. Formally, multi-label classifiers aim to learn from dataset $\mathcal{D}$ a function $f(\cdot) : \mathbb{R}^d \to \{0,1\}^q$ that maps the input features to output labels. For training deep multi-label learning models, selecting a mini-batch $\mathcal{B} = \{(\mathbf{x}_i, \mathbf{y}_i)|_{i=1}^b\} \in \mathcal{D}$ is necessary to update their parameters (weights) due to efficiency and machine memory constraints.

Label Uncertainty

For a sample $\mathbf{x}_i$, the probability of label l_j given by the model at epoch t is defined as $\hat{y}_{ij}^t = P(y_{ij} = 1|\mathbf{x}_i, \theta_t)$, where $\hat{y}_{ij}^t \in [0,1]$ with larger values indicating $\mathbf{x}_i$ is more likely relevant to label l_j, θ_t are the parameters of the deep learning model at epoch t. We use entropy to measure the uncertainty of each label prediction at the current epoch:

$$e_{ij}^t = -\left(\hat{y}_{ij}^t \log_2 \hat{y}_{ij}^t + (1 - \hat{y}_{ij}^t) \log_2 (1 - \hat{y}_{ij}^t)\right) \quad (1)$$

The value of e_{ij}^t is in the range of [0,1], with the maximum value obtained when $\hat{y}_{ij} = 1/2$. A larger e_{ij}^t indicates lower confidence (higher uncertainty) of the prediction for l_j label at epoch t.

Merely considering the uncertainty at the current epoch is not sufficient. As shown in Figure 2, the uncertainty obtained by Eq. (1) is the same in three different cases. However, by tracing through several historical windows, we observe that three cases exhibit different historical predictive trends. Inspired by (Song et al. 2020a), we consider the uncertainty based on historical window prediction to explain the finer details that Eq .(1) cannot distinguish. Let $H_{ij}^t = \{\hat{y}_{ij}^{t-T+1}, \hat{y}_{ij}^{t-T+2}, \ldots, \hat{y}_{ij}^t\}$ be a prediction history queue corresponding to a sliding window of size T at epoch t. There are two traditional measures to evaluate the uncertainty of H_{ij}^t, namely prediction variance (Chang, Learned-Miller, and McCallum 2017):

$$std(H_{ij}^t) = \sqrt{var(H_{ij}^t) + \frac{var(H_{ij}^t)^2}{|H_{ij}^t| - 1}} \quad (2)$$

where $var(H_{ij}^t)$ is the prediction variance estimated by his-

tory H_{ij}^t, and entropy-based uncertainty (Song et al. 2020a):

$$ent(H_{ij}^t) = -\sum_{c\in\{0,1\}} P(y_{ij}=c|\mathbf{x}_i)\log_2 P(y_{ij}=c|\mathbf{x}_i),$$
$$P(y_{ij}=c|\mathbf{x}_i) = \frac{\sum_{\widetilde{y}\in H_{ij}^t}[\![\widetilde{y}=c]\!]}{T} \quad (3)$$

where $\widetilde{y}\in\{0,1\}$ is the binary prediction based on its predicting probability $\hat{y}$, $[\![\cdot]\!]$ is an indicator function that returns 1 if the input is true and 0 otherwise. As shown in Figure 2, when considering the recent five predictions (i.e., T=5), for the first two cases, if we use Eq. (2) or Eq. (3) to measure uncertainty, the uncertainty in these two cases will be the same. However, in case 1, the model's predictions exhibit greater volatility (indicating higher uncertainty), whereas case 2 can be seen as a model prediction with a certain trend. Therefore, to capture prediction fluctuations within the window at a finer granularity, we decided to extract information from the differences in queue H_{ij}^t. In detail, we define d_{ij}^t as the mean of the absolute differences between adjacent predictions within H_{ij}^t, denoting the uncertainty of the historical period:

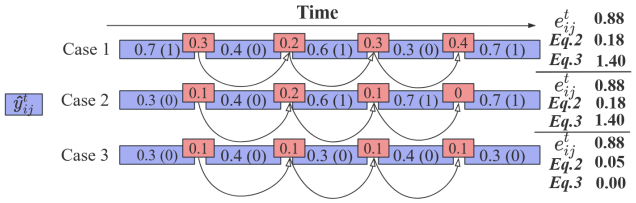


Figure 2: The different uncertainty measurement methods across three cases for historical predictions of the label.

$$d_{ij}^t = \frac{1}{T-1}\sum_{h=1}^{T-1}\left|\hat{y}_{ij}^{t-h+1} - \hat{y}_{ij}^{t-h}\right| \quad (4)$$

where the values of d_{ij}^t range between 0 and 1. The larger the value of d_{ij}^t, the greater the uncertainty in the historical period.

Based on Eq. (1) and Eq. (4), we define the uncertainty u_{ij}^t of x_i regarding l_j at epoch t that combines uncertainties of the current prediction and recent variant trend:

$$u_{ij}^t = \lambda_1 d_{ij}^t + (1-\lambda_1)e_{ij}^t \quad (5)$$

where λ_1 is trade-off parameters determining the importance of two factors. u_{ij}^t of all training instances and labels compose the uncertainty matrix $\mathbf{U}^t \in \mathbb{R}^{n\times q}$ [2]

Label Correlation Guided Sample Uncertainty

Considering each sample, assuming each label is completely independent, we can directly sum the uncertainties for every label, i.e., $\sum_{j=1}^q u_{ij}^t$, as a measure of the sample's uncertainty. However, a key characteristic of multi-label data is the existence of label correlations. During model training, the uncertainty of each label varies and changes dynamically. Ideally, this uncertainty should gradually approach zero. Furthermore, we hypothesize that there is a correlation between the uncertainties of different labels. This correlation may arise due to shared underlying factors that affect multiple labels simultaneously. For instance, in a multi-label classification problem, certain features might influence several labels, leading to simultaneous high uncertainty when these features are ambiguous or conflicting. This can occur when different labels share common subspaces or dependencies, where uncertainty in one label can imply uncertainty in others. Understanding and quantifying this correlation can provide valuable insights into the overall uncertainty of samples. For example, if we observe that high uncertainty in one label often coincides with high uncertainty in others, we can infer that these samples are inherently more challenging and may require more attention during training or evaluation.

First, a discrete distribution is formed using each column of $\mathbf{U}$ (denoted as $\mathbf{u}_{\cdot j}$) by placing the u_{ij} in τ bins of width $1/\tau$. For two labels l_a and l_b, mutual information C_{ab} is defined as:

$$C_{ab} = \sum_{u_{\cdot a}^\tau\in\mathbf{u}_{\cdot a}}\sum_{u_{\cdot b}^\tau\in\mathbf{u}_{\cdot b}} p(u_{\cdot a}^\tau, u_{\cdot b}^\tau)\log\left(\frac{p(u_{\cdot a}^\tau, u_{\cdot b}^\tau)}{p(u_{\cdot a}^\tau)p(u_{\cdot b}^\tau)}\right) \quad (6)$$

where $p(u_{\cdot a}^\tau, u_{\cdot b}^\tau)$ represents the joint probability distribution of $u_{\cdot a}^\tau$ and $u_{\cdot b}^\tau$, where $u_{\cdot a}^\tau$ denotes the bin in which the value u_{ij} from the $\mathbf{u}_{\cdot j}$, with τ bins in total. Similarly, $p(u_{\cdot a}^\tau)$ and $p(u_{\cdot b}^\tau)$ are the marginal probability distributions of the values in these bins for the labels l_a and l_b, respectively [3]. The larger the mutual information C_{ab}, the stronger the dependency between the two labels, indicating more shared information between them. Based on Eq. (6), we obtain the positive definite symmetric matrix $\mathbf{C}$, where the diagonal elements are defined as 1. By combining label correlation with uncertainty, we re-obtain an uncertainty matrix $\bar{\mathbf{U}}$:

$$\bar{\mathbf{U}} = \mathbf{U}\cdot\mathbf{C} \quad (7)$$

Finally, we define the uncertainty weight vector $\mathbf{w} = [w_1, w_2, \ldots, w_n] \in \mathbb{R}^n$. For the i-th sample, the uncertainty weight w_i is defined as:

$$w_i = \sum_{j=1}^q u_{ij} \quad (8)$$

For all samples, the $\mathbf{w}$ is normalized to the range [0, 1]. An example demonstrating the importance of incorporating $\mathbf{C}$ in sample uncertainty is provided in Appendix A.2.

Selection Probability

Motivated by (Song et al. 2020a,b), we exponentially decay the sampling probability of the i-th sample based on its uncertainty weight w_i. In detail, we utilize a quantization method to reduce sampling probabilities, with the quantization index derived from a simple quantizer $Q(z)$ as follows:

$$Q(z) = \lfloor(1-z)/\Delta\rfloor \quad (9)$$

[2] In the following text, $\mathbf{U}$ without the superscript t indicates the current value for readability.

[3] The detailed calculation process can be found in Appendix A.

Algorithm 1: Training by Uncertain Batch Selection

Input: training set $\mathcal{D}$, *epochs*, batch size: b, initial select presure: s_0, warm period:γ, Model: Θ

Initialize the $\mathbf{U}$,$\mathbf{C}$,P;
for $t = 1$ *to epochs* **do**
  if $t > \gamma$ **then**
    $s_t \leftarrow$ Decay Pressure(s_0,t);
    Update $\mathbf{C}$ by Eq. (6);
    for $i = 1$ *to* n **do**
      $w_i \leftarrow$ Compute uncertain weight;
      $P(x_i) \leftarrow$ Compute Prob(s_t,w_i)
  Model training;
  for $i = 1 \leftarrow$ *n/b* **do**
    if $t < \gamma$ **then**
      $\mathcal{B}$=$\left\{(x_i, Y_i)|_{i=1}^{b}\right\} \leftarrow$ Random selection
    else
      $\mathcal{B}$=$\left\{(x_i, Y_i)|_{i=1}^{b}\right\} \leftarrow P(x_i)$
    Forward;
    Update $\mathbf{U}$ by Eq. (1) (4) (5) (6) (7);
    Calculate loss and Backward;
    Optimize Θ

where Δ is the quantization step size, defined as $1/n$, with n representing the total number of samples. This ensures that the quantization index is bounded by n. A crucial component of our approach is the selection pressure s_t, which controls the distribution of sampling probabilities over time. The sampling probability $P(x_i|\mathcal{D}, w_i, n, s_t)$ is then defined as:

$$P(x_i|\mathcal{D}, w_i, n, s_t) = \frac{1/\exp(\log(s_t)/n)^{Q(w_i)}}{\sum_{i=1}^{n} 1/\exp(\log(s_t)/n)^{Q(w_i)}} \quad (10)$$

The higher the uncertainty, the smaller the quantization index. Therefore, a higher selection probability is assigned for more uncertain samples by Eq. (10). For s_t, to mitigate overfitting caused by using only a portion of the training data, we gradually increase the number of training samples as training progresses. This is achieved by exponentially decaying the selection pressure s_t using

$$s_t = s_0 \left(\exp\left(\log(1/s_0)/(t_{end} - t_{start})\right)\right)^{t_{now} - t_{start}} \quad (11)$$

At each epoch t_{now} from t_{start} to t_{end}, the selection pressure s_t exponentially decreases from s_0 to 1. Because this technique gradually reduces the sampling probability gap between the most and the least uncertain samples, more diverse samples are selected for the next mini-batch at a later epoch. When the selection pressure s_t becomes 1, the most and the least uncertain samples are sampled more uniformly. The convergence guarantee of Algorithm 1 is discussed in Appendix B.

Experiments and Analysis

Experiment Setup

Datasets The characteristics of these datasets are detailed in Table 2, including *Card*, the mean labels per instance associated, and *Dens*, the ratio of *Card* to the overall label count.

name	n	d	q	*Card*	*Dens*	domain
scene	2407	294	6	1.07	0.18	images
yeast	2417	103	14	4.24	0.30	biology
Corel5k	5000	499	374	3.52	0.01	images
rcv1subset1	6000	944	101	2.88	0.03	text
rcv1subset2	6000	944	101	2.63	0.03	text
rcv1subset3	6000	944	101	2.61	0.03	text
yahoo-Arts	7484	2314	25	1.67	0.07	text
yahoo-Business	11214	2192	28	1.47	0.06	text
bibtex	7395	1836	159	2.40	0.02	text
tmc2007	28596	490	22	2.15	0.10	text
enron	1702	1001	53	3.38	0.06	text
cal500	502	68	174	26.04	0.15	music
LLOG-F	1460	1004	75	15.93	0.21	text

Table 2: Multi-label Datasets.

Comparison method We compare the proposed method with the following baselines: *Random*, *Balance* (Hand, Castillo, and Chellappa 2018), *Active* (Chang, Learned-Miller, and McCallum 2017), *Recent* (Song et al. 2020a) and *Hard* (Zhou et al. 2024). Among them, *Active* and *Recent* are originally designed for single-label scenarios. To adapt them for multi-label data, we calculate the uncertainty of a sample by summing the uncertainty of each label. Details of the batch selection methods and parameter settings can be found in the related work and Appendix Section C.

Evaluation Metrics To evaluate the effectiveness of the batch selection method in multi-label classification, we use three common metrics: Macro-AUC, Ranking Loss, and Hamming Loss. Please refer to (Zhang and Zhou 2013) for detailed definitions of these metrics.

Base Classifier and Implementation Details We use three multi-label deep models as base classifiers, namely *MPVAE* (Bai, Kong, and Gomes 2021), *CLIF* (Hang and Zhang 2021), and *DELA* (Hang and Zhang 2022). We configure each model precisely according to the parameter specifications, encompassing layer sizes, activation functions, and other intricate details, outlined in the corresponding original research papers and source codes. In terms of optimization, we utilize the Adam optimizer with a batch size of 128, a weight decay of 1e-4, and momentum values of 0.999 and 0.9. For the hyperparameter settings, we fix λ_1 at 0.5, set the selection pressure s_t initially to 100, and use a sliding window size T of 5. During the first 5 epochs, we employ a warm-up phase to initialize the historical predictions for each instance's label. For experiments, we adopt stratified five-fold cross-validation (Sechidis, Tsoumakas, and Vlahavas 2011) to evaluate the aforementioned models. In each fold, we document the test set results achieved at the epoch that yields the best performance on the validation set. All experiments in this work are conducted on a machine with NVIDIA A5000 GPU and Intel Xeon i9-10900 processor. Our code is publicly available on GitHub repository https://github.com/CquptZA/Uncertainty_Batch.

Dataset	*CLIF*						*DELA*					
	Random	*Balance*	*Active*	*Recent*	*Hard*	*Ours*	*Random*	*Balance*	*Active*	*Recent*	*Hard*	*Ours*
scene	0.9418(6)	0.9442(4)	0.9437(5)	0.9446(3)	0.9454(2)	**0.9476(1)**	0.9405(6)	0.9432(5)	0.9458(3)	**0.9476(1)**	0.9449(4)	0.9465(2)
yeast	0.7107(5)	0.7077(6)	0.7164(4)	0.7195(2)	0.7191(3)	**0.7222(1)**	0.7006(4)	0.6972(6)	0.6996(5)	0.7036(3)	0.7121(2)	**0.7132(1)**
Corel5k	0.7664(3)	0.7625(6)	0.7650(4)	0.7648(5)	0.7683(2)	**0.7689(1)**	0.7626(2)	0.7583(6)	0.7594(5)	0.7618(4)	0.7621(3)	**0.7664(1)**
rcv1subset1	0.9221(6)	0.9262(5)	0.9281(4)	0.9296(3)	0.9307(2)	**0.9324(1)**	0.9179(5)	0.9169(6)	0.9187(4)	0.9190(3)	0.9194(2)	**0.9204(1)**
rcv1subset2	0.9279(6)	0.9316(5)	0.9326(4)	0.9338(2)	0.9329(3)	**0.9345(1)**	0.9203(6)	0.9220(2)	0.9213(5)	0.9216(4)	0.9220(2)	**0.9226(1)**
rcv1subset3	0.9268(6)	0.9277(5)	0.9282(3)	**0.9308(1)**	0.9279(4)	0.9306(2)	0.9175(6)	0.9178(5)	0.9185(3)	0.9189(2)	0.9183(4)	**0.9192(1)**
yahoo-Business1	0.7801(6)	0.7926(3)	0.7862(5)	0.7884(4)	0.7940(2)	**0.8077(1)**	0.7979(5)	0.7972(6)	0.8026(3)	0.8039(2)	0.8010(4)	**0.8086(1)**
yahoo-Arts1	0.7580(4)	0.7591(3)	0.7598(2)	**0.7599(1)**	0.7535(6)	0.7562(5)	0.7407(5)	0.7395(6)	0.7422(4)	0.7460(2)	0.7444(3)	**0.7480(1)**
bibtex	0.9013(2)	0.8975(6)	0.8996(4)	0.8982(5)	0.9011(3)	**0.9059(1)**	**0.9062(1)**	0.8974(6)	0.9046(5)	0.9052(4)	0.9057(3)	0.9060(2)
tmc2007	0.9048(5)	0.9053(3)	0.9051(4)	0.9046(6)	0.9059(2)	**0.9062(1)**	0.9121(4)	0.9087(6)	0.9145(3)	0.9162(2)	0.9120(5)	**0.9179(1)**
enron	0.7700(6)	0.7744(5)	0.7753(4)	**0.7782(1)**	0.7763(3)	0.7764(2)	0.7727(6)	0.7732(5)	0.7748(3)	**0.7806(1)**	0.7748(3)	0.7754(2)
cal500	0.5901(4)	0.5949(2)	0.5843(6)	0.5885(5)	0.5932(3)	**0.6054(1)**	0.5933(2)	0.5761(6)	0.5872(5)	0.5914(4)	0.5927(3)	**0.5954(1)**
LLOG-F	0.7659(6)	0.7686(4)	0.7682(5)	0.7716(2)	0.7703(3)	**0.7721(1)**	0.7909(6)	0.7911(5)	0.7916(4)	**0.7935(1)**	0.7930(2)	0.7924(3)
Avg (Rank)	5.00	4.38	4.15	3.08	2.92	**1.46**	4.46	5.38	4.00	2.54	3.08	**1.38**

Table 3: The Macro-AUC results of batch selection methods under different models.

Experimental Results and Analysis

Results Table 3 presents the average Macro-AUC comparison of six different batch selection methods within *CLIF*, and *DELA*. The detailed results under *MPVAE* model and in other metrics, along with the Wilcoxon signed-ranks test are shown in Appendix Section D. Our batch selection methods consistently achieve the best performance in most datasets. This advantage is particularly evident in Corel5k, rcv1subset1, cal500, and bibtex datasets with larger scales, high label dimensions, or suffering significant imbalance issues. Additionally, our batch selection method performs exceptionally well across different models, demonstrating their versatility and robustness. *Hard* is usually the runner-up, indicating that selecting samples based on their relevance to the learning objectives and prioritizing more informative and challenging samples is beneficial for the classifier. *Active* batch selection, which considers uncertainty over the entire cumulative history and directly accumulates the uncertainty of all labels as the sample uncertainty, outperforms the baseline in most datasets. Similarly, *Recent* batch selection considers uncertainty within the latest sliding windows and directly accumulates the uncertainty of all labels as the sample uncertainty. While it generally outperforms the baseline across most datasets, it underperforms on datasets with many labels. Although experiments on smaller datasets (such as yeast and enron) indicate that the *Balance* method outperforms the baseline, the effectiveness of *Balance* has not been sufficiently demonstrated on the majority of datasets.

Analysis To conduct an in-depth analysis of the different batch selections, we plot the convergence curves for five different batching methods across four datasets in Figure 3, and plot the Macro-AUC for each epoch on the validation set of the four datasets in Figure 4. *Balance* determines batch assignments based on the label proportions in the original training set. This can lead to underrepresentation or overrepresentation of certain labels, which may cause overfitting in later stages of training, as observed in the bibtex and yahoo Business1 datasets. The *Active* method focuses on moderately hard samples in the early stages of training, with a loss distribution between random and online batches. However, as the window continues to expand, predictions become outdated, and the proportion of low-loss, easy samples increases in the later stages, which may slow down the convergence. Similarly, the *Recent* method also emphasizes moderately hard samples, with a loss distribution similar to *Active*. However, unlike *Active*, the *Recent* method uses a sliding window to dynamically update its selection of moderately hard samples throughout the training process. We find that traditional uncertainty-based batch selection methods exhibit less than ideal convergence speed and are prone to overfitting in datasets with large sample sizes (e.g., yahoo-Business) or high-dimensional label spaces (e.g., bibtex). This may be because, as sample size or label dimensions increase, identifying and prioritizing uncertain samples becomes more challenging, further complicating the batch selection process. *Hard* prioritizes high-loss samples associated with minority labels, resulting in more informative and challenging samples in each batch. This accelerates the learning process and leads to better generalization, but the *Hard* method shows overfitting on certain datasets, such as yahoo-Business1. This could be due to the frequent selection of difficult samples in the later stages of training. Due to its dynamic consideration of uncertainty using a sliding window, *Our* batch selection typically results in better convergence by providing the model with moderately hard samples and avoiding the frequent selection of certain samples in the later stages through decaying selection pressure.

More detailed empirical analyses, including computational complexity, ablation studies, and parameter sensitivity, are detained in Appendices E, F and G, respectively.

Uncertainty based Label Correlation Drift The concept of uncertainty-based label correlation drift explores how the relationships between labels evolve during model training. Figures 5(a) and 5(b) illustrate the label correlation matrices at the 30th and 70th epochs, showing how the model's perception of label relationships, based on uncertainty, evolves during training. The differences highlighted in Figure 5(c) reveal that these correlations change dynamically, suggest-

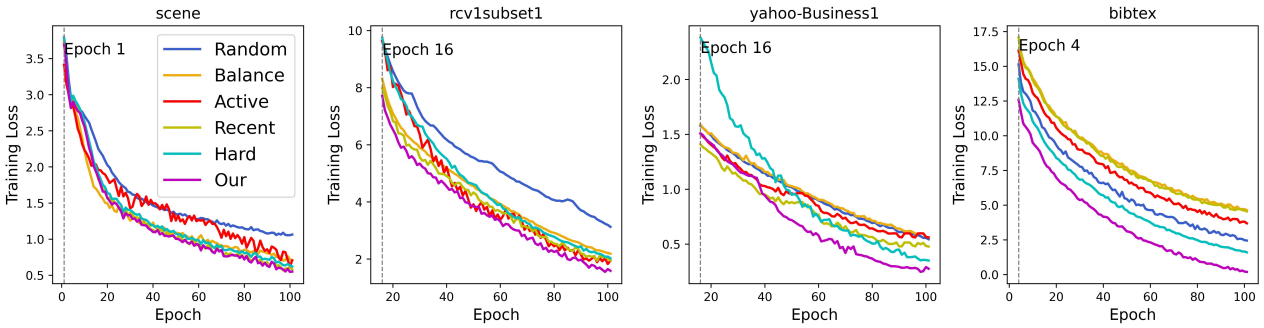


Figure 3: The convergence curves of five batch selection methods using *CLIF*.

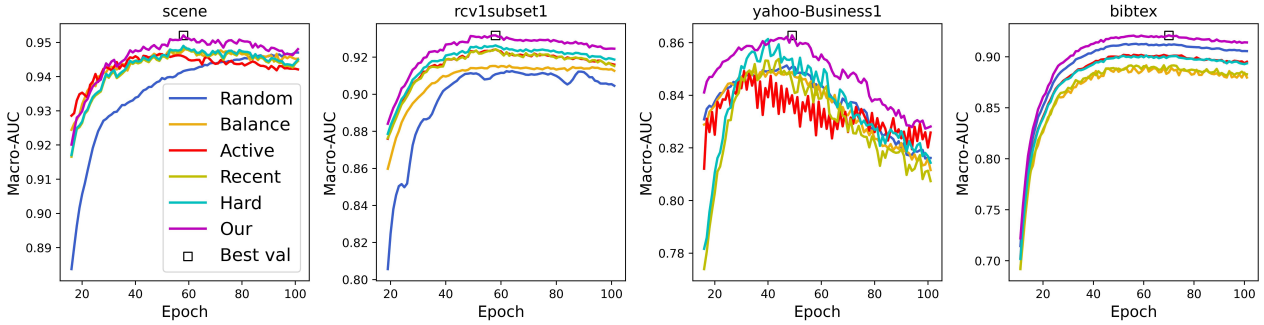
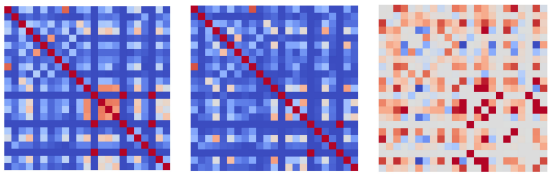


Figure 4: The Macro-AUC on validation set of five batch selection methods using *CLIF*.



(a) 30-th epoch (b) 70-th epoch (c) difference

Figure 5: Visualization of label correlation matrix $\mathbf{C}$ change in *CLIF* with Corel5k dataset.

ing that as the model learns, it adjusts its understanding of these uncertainty-based label relationships.

Conclusion

This paper proposes an uncertainty-based multi-label batch selection method, filling the gap in uncertainty-based multi-label batch selection. We introduce two key components to better adapt to the characteristics of multi-label data. First, for each label, we consider both the confidence of the current prediction and the changes between consecutive predictions within a sliding window, allowing for a more accurate assessment of label uncertainty. Second, we leverage dynamic uncertainty-based label correlations to comprehensively evaluate each sample's uncertainty, prioritizing samples that exhibit synergistic uncertainty across multiple labels during training. Experimental results show that our method greatly enhances model performance, speeds up convergence, and outperforms five other methods across diverse datasets and deep multi-label learning models.

In future work, temperature scaling can be explored to improve the entropy-based uncertainty assessment in deep learning models, and theoretical uncertainty methods based on Bayesian averaging, such as Monte Carlo dropout, can be considered.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62302074) and the Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202300631).

References

Bai, J.; Kong, S.; and Gomes, C. 2021. Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. In *IJCAI*, 4313–4321.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *ICML*, 41–48.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition*, 37(9): 1757–1771.

Chai, Y.; Li, Z.; Liu, J.; Chen, L.; Li, F.; Ji, D.; and Teng, C. 2024. Compositional generalization for multi-label text clas-

sification: a data-augmentation approach. In *AAAI*, 17727–17735.

Chakraborty, S.; Balasubramanian, V.; and Panchanathan, S. 2011. Optimal batch selection for active learning in multi-label classification. In *ACM-MM*, 1413–1416.

Chang, H.-S.; Learned-Miller, E.; and McCallum, A. 2017. Active bias: training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, 1003–1013.

Chen, S.; Wang, R.; Lu, J.; and Wang, X. 2022. Stable matching-based two-way selection in multi-label active learning with imbalanced data. *Information Sciences*, 610: 281–299.

Chen, W.-J.; Shao, Y.-H.; Li, C.-N.; and Deng, N.-Y. 2016. MLTSVM: A novel twin support vector machine to multi-label learning. *Pattern Recognition*, 52: 61–74.

Fan, Y.; Tian, F.; Qin, T.; and Liu, T.-Y. 2016. Neural data filter for bootstrapping stochastic gradient descent. In *ICLR*.

Gupta, R.; Roy, A.; Christensen, C.; Kim, S.; Gerard, S.; Cincebeaux, M.; Divakaran, A.; Grindal, T.; and Shah, M. 2023. Class prototypes based contrastive learning for classifying multi-label and fine-grained educational videos. In *CVPR*, 19923–19933.

Hand, E.; Castillo, C.; and Chellappa, R. 2018. Doing the best we can with what we have: multi-label balancing with selective learning for attribute prediction. In *AAAI*, 6878–6885.

Hang, J.; and Zhang, M. 2021. Collaborative learning of label semantics and deep label-specific features for multi-label classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 9860–9871.

Hang, J.-Y.; and Zhang, M.-L. 2022. Dual perspective of label-specific feature learning for multi-label classification. In *ICML*, 8375–8386.

Hang, J.-Y.; Zhang, M.-L.; Feng, Y.; and Song, X. 2022. End-to-end probabilistic label-specific feature learning for multi-label classification. In *AAAI*, 6847–6855.

Huang, Y.; Shen, P.; Tai, Y.; Li, S.; Liu, X.; Li, J.; Huang, F.; and Ji, R. 2020. Improving face recognition from hard samples via distribution distillation loss. In *ECCV*, 138–154.

Jiang, T.; Wang, D.; Sun, L.; Yang, H.; Zhao, Z.; and Zhuang, F. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In *AAAI*, 7987–7994.

Katharopoulos, A.; and Fleuret, F. 2018. Not all samples are created equal: Deep learning with importance sampling. In *ICML*, 2525–2534.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *NeurIPS*, 1189–1197.

Liu, B.; and Tsoumakas, G. 2020. Dealing with class imbalance in classifier chains via random undersampling. *Knowledge-Based Systems*, 192: 105292.

Liu, W.; Wang, H.; Shen, X.; and Tsang, I. W. 2021. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 7955–7974.

Liu, Y.; Yang, X.; Zhou, S.; Liu, X.; Wang, Z.; Liang, K.; Tu, W.; Li, L.; Duan, J.; and Chen, C. 2023. Hard sample aware network for contrastive deep graph clustering. In *AAAI*, 8914–8922.

Loshchilov, I.; and Hutter, F. 2016. Online batch selection for faster training of neural networks. In *ICLR*.

Nguyen, H. D.; Vu, X.-S.; and Le, D.-T. 2021. Modular graph transformer networks for multi-label image classification. In *AAAI*, 9092–9100.

Sechidis, K.; Tsoumakas, G.; and Vlahavas, I. 2011. On the stratification of multi-label data. In *ECML-PKDD*, 145–158.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *CVPR*, 761–769.

Song, H.; Kim, M.; Kim, S.; and Lee, J.-G. 2020a. Carpe diem, seize the samples uncertain "at the Moment" for adaptive batch selection. In *CIKM*, 1385–1394.

Song, H.; Kim, M.; and Lee, J.-G. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 5907–5915.

Song, H.; Kim, S.; Kim, M.; and Lee, J.-G. 2020b. Ada-boundary: accelerating DNN training via adaptive boundary batch selection. *Machine Learning*, 109: 1837–1853.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7): 1079–1089.

Wu, Q.; Tan, M.; Song, H.; Chen, J.; and Ng, M. K. 2016. ML-FOREST: A multi-label tree ensemble method for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(10): 2665–2680.

Yeh, C.-K.; Wu, W.-C.; Ko, W.-J.; and Wang, Y.-C. F. 2017. Learning deep latent space for multi-label classification. In *AAAI*.

You, R.; Guo, Z.; Cui, L.; Long, X.; Bao, Y.; and Wen, S. 2020. Cross-modality attention with semantic graph embedding for multi-label classification. In *AAAI*, 12709–12716.

Zhang, M. L.; Li, Y. K.; Yang, H.; and Liu, X. Y. 2020. Towards class-imbalance aware multi-label learning. *IEEE Transactions on Cybernetics*, 52(6): 1–13.

Zhang, M.-L.; and Zhou, Z.-H. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7): 2038–2048.

Zhang, M.-L.; and Zhou, Z.-H. 2013. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1819–1837.

Zhao, W.; Kong, S.; Bai, J.; Fink, D.; and Gomes, C. 2021. Hot-vae: Learning high-order label correlation for multi-label classification via attention-based variational autoencoders. In *AAAI*, 15016–15024.

Zhou, A.; Liu, B.; Wang, J.; and Tsoumakas, G. 2024. Multi-label adaptive batch selection by highlighting hard and imbalanced samples. In *ECML-PKDD*, 265–281.

Zhou, F.; Huang, S.; and Xing, Y. 2021. Deep semantic dictionary learning for multi-label image classification. In *AAAI*, 3572–3580.