Uncertainty-guided Boundary Learning for Imbalanced Social Event Detection

Jiaqian Ren, Hao Peng, Lei Jiang, Zhiwei Liu, Jia Wu, Zhengtao Yu, Philip S. Yu, *Fellow, IEEE*

Abstract—Real-world social events typically exhibit a severe class-imbalance distribution, which makes the trained detection model encounter a serious generalization challenge. Most studies solve this problem from the frequency perspective and emphasize the representation or classifier learning for tail classes. While in our observation, compared to the rarity of classes, the calibrated uncertainty estimated from well-trained evidential deep learning networks better reflects model performance. To this end, we propose a novel uncertainty-guided class imbalance learning framework - UCL$_{SED}$, and its variant - UCL-EC$_{SED}$, for imbalanced social event detection tasks. We aim to improve the overall model performance by enhancing model generalization to those uncertain classes. Considering performance degradation usually comes from misclassifying samples as their confusing neighboring classes, we focus on boundary learning in latent space and classifier learning with high-quality uncertainty estimation. First, we design a novel uncertainty-guided contrastive learning loss, namely UCL and its variant - UCL-EC, to manipulate distinguishable representation distribution for imbalanced data. During training, they force all classes, especially uncertain ones, to adaptively adjust a clear separable boundary in the feature space. Second, to obtain more robust and accurate class uncertainty, we combine the results of multi-view evidential classifiers via the Dempster-Shafer theory under the supervision of an additional calibration method. We conduct experiments on three severely imbalanced social event datasets including Events2012_100, Events2018_100, and CrisisLexT_7. Our model significantly improves social event representation and classification tasks in almost all classes, especially those uncertain ones.

Index Terms—Social event detection, evidential deep learning, demperster-shafer theory, imbalanced data

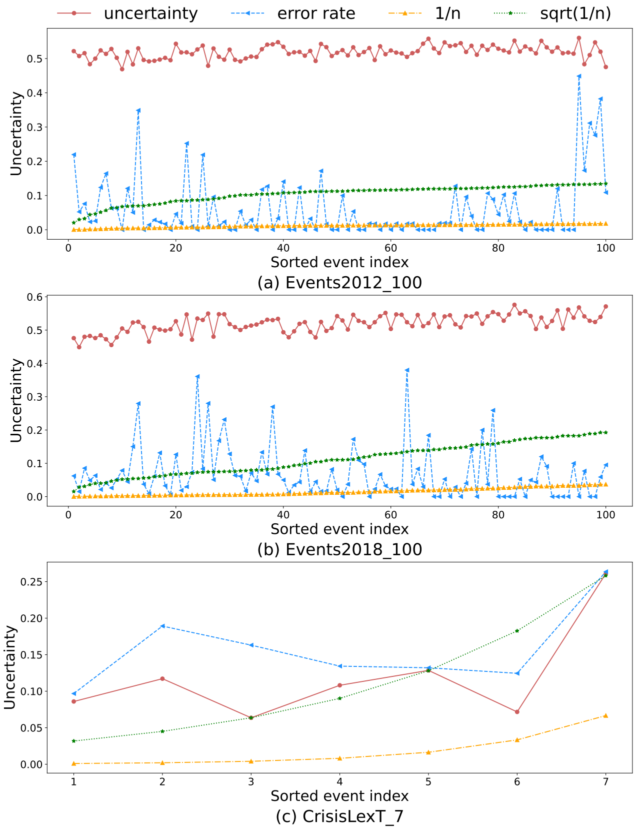
1 Introduction

Social event detection (SED) aims to correctly categorize the numerous social messages to detect the occurrences of events. Due to its wide application, recent years have witnessed lots of research on the detection methods [1], [2], [3]. However, few works investigate the severe data distribution imbalance problem in SED. Events have varying recognition difficulty levels because of the following two reasons. First, in the real-world scenario, event data typically exhibit a long-tail distribution with few head-dominant event classes and many low-frequent tail classes. Lacking sufficient training samples, the trained model's detection abilities for most events are data-sensitive, which means they are easily affected by per-class sample qualities. Second, some events may share semantically similar contexts with other events. This semantic-level overlapping issue also increases the complexity of event detection.

Early approaches mainly focus on learning a balanced classifier to tackle the data imbalance issue. Two common strategies are: 1) data re-sampling [4], [5], [6], whose core idea is to increase the numbers of samples in the tail classes or decrease the samples of those head classes; and 2) loss reweighting [7], [8], [9], [10], in which weights assigning to tail classes are larger. Though unbiased classifiers can be obtained by applying both aforementioned strategies, some recent works [11], [12], [13], [14] argue that they are unable to explicitly control the latent representation space, and therefore, are sub-optimal. Inspired by the intuition that qualitative features improve the classification, a recent line of work [15], [16], [17], [18], [19] focuses on learning more separable representations for imbalanced data. For example, authors in [17] design a hybrid framework with a supervised contrastive learning branch for representation regularization and a classifier branch for bias eliminating. Later, some approaches [18], [19] modify the original contrastive learning loss under the guidance of class frequency to further improve representation learning for imbalanced data. For example, BCL [19] incorporates class-averaging and class-complement strategies to strengthen tail classes. DRO-LT [20] learns high-quality representations based on distributional robustness optimization. However, as the difficulty level of event recognition is also affected by class overlapping, the class frequency may be an insufficient indicator to model performance. Our previous work [21] shows that evidential uncertainty estimated from well-trained EDL neural networks highly correlates with performance error. This correlation is shown in Fig. 1. Compared to class frequency, the predicted uncertainty better indicates model generalization capacity. Therefore, in this paper, we explore a new direction toward learning balanced and separable representations under the guidance of uncertainty. The focus

- Jiaqian Ren and Lei Jiang are with the Institute of Information Engineering, Chinese Academy of Sciences, and the School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China. E-mail: {renjiaqian, jianglei}@iie.ac.cn;
- Hao Peng is with the School of Cyber Science and Technology, Beihang University, Beijing 1000191, China. E-mail: penghao@buaa.edu.cn;
- Zhiwei Liu is with Salesforce AI Research, CA 94301, USA. E-mail: zhiweiliu@salesforce.com;
- Jia Wu is with the School of Computing, Macquarie University, Sydney, Australia. E-mail: jia.wu@mq.edu.au;
- Zhengtao Yu is with the Faculty of Information Engineering and Automation, and Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China. E-mail: ztyu@hotmail.com;
- Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA. E-mail: psyu@uic.edu.

Manuscript received November 2022, major revised May 2023, major revised August 2023, accepted October 2023. (Corresponding author: Hao Peng.)



(a) Events2012_100

(b) Events2018_100

(c) CrisisLexT_7

Fig. 1: Statistics of the estimated per-class uncertainty, per-class error rate, the inverse of per-class number $1/n$ and the sqrt of $1/n$ on the training sets of Events2012_100, Events2018_100, and CrisisLexT_7, respectively.

shifts from tail classes to uncertain classes. We aim to improve the overall model performance by improving learning for those uncertain ones. To clarify, we also define uncertain event classes as events that are difficult to recognize due to the lack of labeled data or excessive noise from semantically similar events. Our goal is to propose a unified framework that combines the learning of uncertainty in the classification head with the uncertainty-guided boundary adjustment in the representation head.

To achieve this goal, in this work, we propose a novel uncertainty-guided class imbalance learning framework, namely UCL$_{SED}$, as well as its variant - UCL-EC$_{SED}$, for imbalanced social event detection tasks. It in essence works as a representation regularization technique guided by class-dependent uncertainty. The core of our framework is to make adaptive and automatic boundary adjustments in the latent space. The blurry boundary gets separated by designing a loss that assigns larger margins in the latent space for those more uncertain classes. Meanwhile, given that our assumption roots in high-quality uncertainty estimation, we also emphasize robust and accurate evidential classifier learning. Specifically, we design a novel uncertainty-guided contrastive learning loss (UCL and UCL-EC) to manipulate distinguishable representation distribution. To ensure robustness, in the classification head, the final uncertainty is calculated by combining multi-view results via Dempster's rule [22]. To ensure accuracy, an additional calibration method is utilized to prevent uncertain true predictions and certain false predictions.

We conduct extensive experiments on three imbalanced event datasets to evaluate our model. Experimental results show our method achieves great results in all classes, especially in those hard (uncertain) classes. This demonstrates the superiority of our model. The code of this work is publicly available at GitHub[1].

2 Related Work

Social Event Detection. Social event detection is a long-standing and challenging task. First, distinct from the formal texts that appeared on other occasions, text contents in social networks are often restricted to be pretty short and contain many informal expressions [23]. These characteristics of social texts make the information extracted from original semantic text mining technologies far from satisfactory. Second, to depict an event, short social texts usually contain rich social network attributes [24], such as hashtags, timestamps, users, mentions, retweets, and so on. It is difficult to incorporate these heterogeneous attributes effectively. According to the utilized information, social event detection approaches can be roughly divided into three categories: content-based methods [25], [26], [27], [28], [29], [30], attribute-based methods [31], [32], [33] and content-attribute-combining methods [34], [35], [36], [37], [38], [39], [23], [24], [21]. As for content-based methods, a series of works make detection by analyzing text semantics. This type of method typically builds on text representation models such as Bag-of-words model [40], Word2Vec [41] and Bert [42] or topic models like LDA [43] to represent social texts. Because the text contents are short, which makes the captured information insufficient, some leverage multi-task technologies to extend the original knowledge. For example, authors in [28] utilize Deep Neural Networks to jointly make event detection and summarization. Some even incorporate information from external knowledge bases. As for the line of attribute-based methods, many studies make event detection by using important social attributes, such as hashtags [32], mention [44], [45], retweet [46] and so on. This kind of work ignores the text semantics and thus is also insufficient. To grasp more comprehensive information, there is a trend toward content-attribute-combining methods. They propose to integrate the content and multiple attributes with fusion or graph models. Due to their powerful expressiveness for graph data, in recent years, Graph Neural Networks have attracted lots of attention in the social event detection domain [47], [37], [38], [39], [23], [24], [48], [21], [47]. These GNN-based detection methods build heterogeneous information networks to represent social event data. Various attributes in social networks effectively complement each other and play an independent role in text semantic propagation and aggregation. For example, KPGNN [38] utilizes users, keywords, and entity attributes to construct an event message graph and then leverages inductive Graph Attention Networks (GAT) to learn message representations. PP-GCN [37] utilizes multiple attributes by designing sophisticated meta-paths and then uses Graph Convolutional Networks (GCN) to obtain representations. Later,

[1] https://github.com/RingBDStack/UCL_SED

some works leverage multi-view learning strategies to further strengthen the feature learning process. MVGAN [24] learns message features from both the semantic and temporal views, then proposes an attention mechanism to fuse them together. ETGNN [21] instead learns representations from *co-hashtag*, *co-entity* and *co-user* views. Dempster-Shafer Theory extracts shared beliefs to resist noise and make the final decision more robust. Our work builds on ETGNN and makes a modification in the representation learning process to adapt to imbalanced data. Besides, we also add an uncertainty calibration method to ensure the accuracy of the estimated uncertainty.

Long-tail Recognition. Real-world data such as events, images and objects usually follow a long-tail distribution, which makes the trained model generalize badly. Early solutions tend to solve this problem from two perspectives: (1) modifying the training data through some data re-sampling strategies [4], [5], [6], [49], [50], [51], [52], [53] and (2) modifying the loss function with re-weighting strategy [7], [8], [9], [10], [54] or margin modification strategy [55], [56], [57]. Re-sampling approaches mainly consist of three popular techniques, including over-sampling tail classes by sample copying [49], [50], generating augmented samples to supplement tail classes [6], [51], [52] and under-sampling head classes by discarding part of data [53]. Despite the good results, these three techniques face tail-class overfitting, expensive cost, and model generalization problems, respectively [49], [4]. Loss re-weighting approaches tailor the loss function based on up-weighting the samples in tail classes and down-weighting those in head classes. For instance, works such as [58], [7] re-weight the loss functions by the inverse of class frequencies. In the work [9], a held-out evaluation set is utilized to optimize the weights to samples. Authors in [54] leverage the difficulty level of sample prediction, measured by the confidence score gap, to rescale the cross-entropy loss. Also, there is an alternative strategy to manipulate the classification loss margins. Some works [55], [57] proposed encouraging larger logit margins for rare classes and decreasing margins for head classes.

All the aforementioned methods focus on learning a balanced classifier. While recently, some works [11], [12], [14] explore decoupling the original classification learning into two separate stages including representation learning and classifier learning. According to their observation, the learned representation with a balanced classifier (trained by re-sampling and re-weighting methods) is sub-optimal. Based on this observation, more and more works turn to learn better representations from imbalanced data to improve classification performance [59], [17], [19], [20], [18]. Inspired by the great promise of contrastive learning [60], [61] in obtaining distinguishable representations, researchers have investigated the potential of leveraging contrastive learning loss to manipulate further performance gain. SSP [59] leverages self-supervised and semi-supervised contrastive learning to boost long-tailed learning tasks. Authors in [17] design a hybrid framework with a supervised contrastive learning branch for better representation and a classifier branch for bias elimination. Sharing a similar framework, the work [19] introduces a novel BCL loss in the representation learning branch to deal with the domination of head classes. Based on distributional robustness optimization, DRO-LT [20] explicitly seeks to improve the quality of representations for tail classes. Authors in [18] instead propose targeted supervised contrastive learning with a set of pre-defined feature distributions. Most of these works modify the contrastive learning loss under frequency guidance. However, according to our observation, frequency is a worse indicator of model performance than uncertainty. Our work leverages uncertainty to adaptively adjust class boundary learning in the latent space. A similar work to ours is [56], which links class imbalance problems with Bayesian uncertainty estimates. However, it indirectly models uncertainty through network weights, which is inefficient. Besides, it utilizes uncertainty to adjust logit margins instead of representation margins.

3 Preliminary

3.1 Classification Task

The aim of the classification task is to learn a complicated mapping function from an input space $\mathcal{X}$ to a target space $\mathcal{Y} = \{1, 2, ..., C\}$. Generally, the mapping function is composed of two parts: an encoder model f which maps the input to a latent space $\mathcal{Z} \in \mathbb{R}^h$, and a classifier g which maps the latent space $\mathcal{Z}$ to the target space $\mathcal{Y}$. In this work, we leverage a modified contrastive loss to adjust the latent space $\mathcal{Z}$. We tend to improve the final classification performance by making the learned representations more distinguishable.

3.2 Temporal-aware GNN encoder

Graph neural networks are proposed for representation learning on graph data [62], [63]. For each node on the graph, a GNN encoder iteratively updates its representation by combining information from its one-hop neighbors. In this way, the learned representation contains graph structural and node attribute information and is more comprehensive. Typically, a GNN encoder layer comprises two types of operation: feature transformation operation and feature aggregation operation. Suppose the node representation of index i in the $(l-1)-th$ layer is denoted as $\mathbf{h}_i^{l-1}$, its updated representation in the next layer is computed as follows:

$$\boldsymbol{h}_i^{(l)} \leftarrow \sigma\left(\underset{\forall j \in \mathcal{N}(i)}{\text{Aggregator}}\left(\text{ Transformation }\left(\boldsymbol{h}_j^{(l-1)}\right)\right)\right), \quad (1)$$

where $\mathcal{N}(i)$ represents the set of neighbor indices of node i. Aggregator and Transformation are designed differently in different GNN models. Since temporal information is important in indicating events, we adopt the temporal-aware GNN aggregator in our previous work [21] to incorporate temporal information into graph representation learning. As for the transformation operation, we use the simple linear trainable transformation. The specific layer-wise propagation becomes:

$$\mathbf{h}_i^l \leftarrow \sigma\left(\sum_{j \in \mathcal{N}(i)} a_{ij} \mathbf{W} \mathbf{h}_j^{l-1}\right). \quad (2)$$

Here $\mathbf{W}$ denotes the transformation matrix learned during training. $\sigma(\cdot)$ is an activation function. Attention weight

a_{ij} measures the temporal approximation between message from node i and message from node j and is computed as follows:

$$a_{ij} = \frac{e^{-fc(h_i^l)\cdot|t_j - t_i|}}{\sum_{j\in\mathcal{N}(i)} e^{-fc(h_i^l)\cdot|t_j - t_i|}}, \tag{3}$$

where t_i and t_j are the publishing time of message i and message j. $|t_j - t_i|$ is the corresponding time interval that can be counted in days, hours, minutes, etc. Here in this paper, we counted in days. $fc(\cdot)$ represents a fully connection layer. More details can be seen in [21].

3.3 Contrastive Learning

Contrastive learning imposes geometric constraints on the sample representations to regulate the model. It follows a simple principle of pulling the samples from the same class together and pushing the samples from different classes apart. We here introduce some variants of supervised contrastive learning loss which will facilitate the understanding of our later modification.

Supervised contrastive loss [61]. Supervised contrastive loss (SCL) utilizes label information to find samples within the same class as the positive ones. Formally, in a batch B, for an instance $\mathbf{x}_i$ whose representation learnt by encoder f is $\mathbf{z}_i$, supervised contrastive loss is written as:

$$\mathcal{L}_{SCL}(\mathbf{z}_i) = -\frac{1}{|\{x_i^+\}|}\sum_{j\in\{x_i^+\}} \log \frac{\exp\left(S(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\sum_{k\in B\backslash\{i\}} \exp\left(S(\mathbf{z}_i, \mathbf{z}_k)/\tau\right)}, \tag{4}$$

where $\{x_i^+\}$ denotes a subset of B that contains all samples within the same class as x_i. $|\{x_i^+\}|$ is the number of all the positive samples in the batch B. τ is a temperature parameter. S denotes a similarity metric function where cosine similarity is often selected. Because cosine similarity removes the effect of feature length and emphasizes angle information, which further facilitates linear classification thus making the training process more stable. To sum up, SCL maximizes agreement between the anchor and all positive samples by contrasting against samples from other classes. While simple and effective, this loss faces memory issues [61].

Prototypical supervised contrastive loss [17]. Prototypical supervised contrastive (PSC) loss replaces specific positive and negative samples with prototypes to tackle the memory issue. In PSC, each sample is pulled close to the prototype of its class and pushed away from prototypes of other classes. Formally, suppose there is a sample x_i (representation is z_i) whose label is y_i, the PSC loss function can be expressed as follows:

$$\mathcal{L}_{PSC}\left(\mathbf{z}_i\right) = -\log \frac{\exp\left(S(\mathbf{z}_i \cdot \mathbf{p}_{y_i})/\tau\right)}{\sum_{c=1}^{\mathcal{C}} \exp\left(S(\mathbf{z}_i \cdot \mathbf{p}_c)/\tau\right)}, \tag{5}$$

where $\mathbf{p}_{y_i}$ is the prototype representation of class y_i that sample x_i belongs to. The prototype representations are learned during training.

4 The Proposed Model

4.1 Overall Framework

We start with an overview of our uncertainty-guided class imbalance learning framework UCL$_{SED}$ and UCL-EC$_{SED}$ and provide the details in subsequent sections.

The overall framework is demonstrated in Fig. 2, built on our previous work - ETGNN [21]. The aim of this paper is to enhance imbalanced social event detection by learning better representations such that the class boundaries are well separated. With the observation that evidential uncertainty well reflects model performance and the assumption that model performance is highly correlated with the representation distribution, the key idea becomes using the predicted evidential uncertainty from the classifier to monitor and adjust the representation distribution status during the training process.

The detailed training process of UCL$_{SED}$ is depicted in Algo. 1. Following [21], we first construct three view-specific message graphs (*co-hashtag*, *co-entity* and *co-user*) by simply connecting messages sharing the same corresponding element together. Then we utilize a temporal-aware GNN encoder (also introduced in Sec. 3.2) to obtain the message representations. The next steps are this paper's two key modules, which will be roughly introduced in the next paragraph and detailed in the subsequent sections. Note that UCL-EC$_{SED}$ is similar to UCL$_{SED}$ but different in acquiring the prototypes. Considering the page limit, we didn't show the detailed algorithm of UCL-EC$_{SED}$. But readers can refer to Sec. 4.2.2 to see the concrete difference.

Two key modules in the framework are the representation adjustment module and the multi-view classifier module. In the representation adjustment module, to ensure class separability, we propose a novel uncertainty-guided contrastive learning loss (i.e. UCL and UCL-EC) to assign larger inter-class margins in the latent space for those uncertain classes. In the multi-view classifier module, we use Dempster-Shafer theory to combine the results from the three single views. Meanwhile, considering uncertainty plays an important role in representation adjustment and classification making, an additional calibration constraint is added to get better and more robust uncertainty estimations and class predictions. Generally, in our work, representation, and classification learning are combined closely and mutually promote each other. On one hand, the result from classification learning works as an effective indicator to reflect the current representation distribution status. The representations become more distinguishable by setting larger margins for more uncertain classes. On the other hand, in line with the property of intra-class compactness and inter-class separability, the adjusted representation further facilitates classification learning.

4.2 Representation Adjustment Module

4.2.1 *Uncertainty-Guided Contrastive Learning Loss (UCL)*

In this section, we introduce UCL in detail, an extension of prototypical supervised contrastive loss. The key difference lies in the setting of the class margin.

To facilitate understanding, we rewrite PSC loss as follows:

$$\begin{aligned}\mathcal{L}_{PSC}\left(\mathbf{z}_i\right) &= \log\left[1 + \sum_{c=1,c\neq y_i}^{\mathcal{C}} e^{\Delta_{y_i c} + S(\mathbf{z}_i,\mathbf{p}_c) - S(\mathbf{z}_i,\mathbf{p}_{y_i})}\right], \\ \Delta_{y_i c} &= 0.\end{aligned} \tag{6}$$

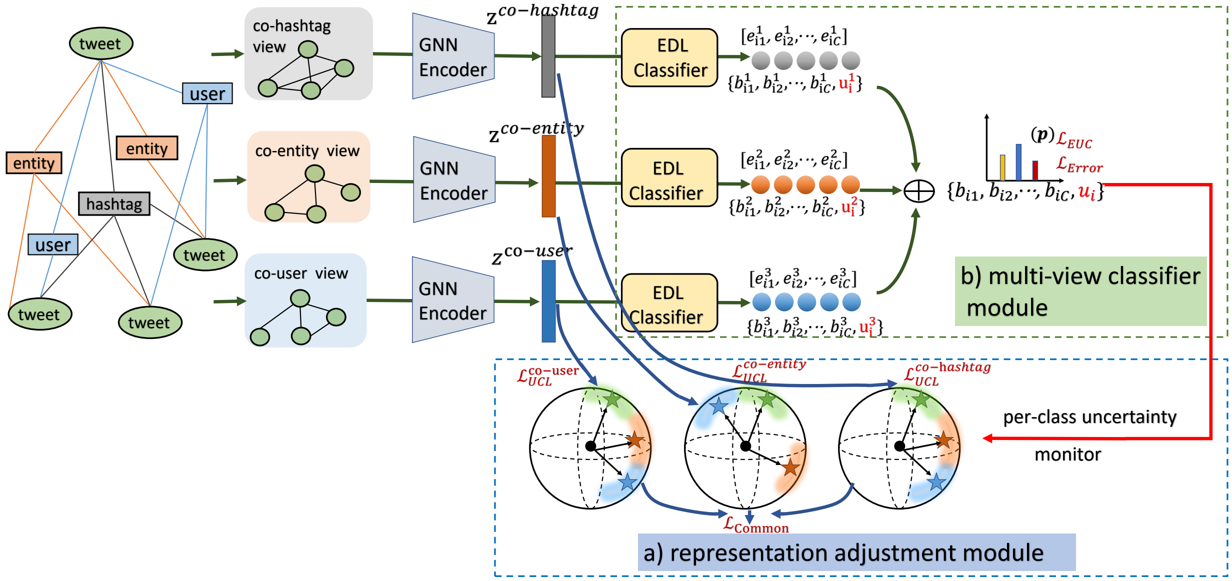

tweet
user
entity
entity
hashtag
user
tweet
tweet
tweet
co-hashtag view
co-entity view
co-user view
GNN Encoder
GNN Encoder
GNN Encoder
$z^{co-hashtag}$
$z^{co-entity}$
$z^{co-user}$
EDL Classifier
EDL Classifier
EDL Classifier
$[e_{i1}^1, e_{i2}^1, \cdots, e_{iC}^1]$
$\{b_{i1}^1, b_{i2}^1, \cdots, b_{iC}^1, u_i^1\}$
$[e_{i1}^2, e_{i2}^2, \cdots, e_{iC}^2]$
$\{b_{i1}^2, b_{i2}^2, \cdots, b_{iC}^2, u_i^2\}$
$[e_{i1}^3, e_{i2}^3, \cdots, e_{iC}^3]$
$\{b_{i1}^3, b_{i2}^3, \cdots, b_{iC}^3, u_i^3\}$
$\oplus$
$\{b_{i1}, b_{i2}, \cdots, b_{iC}, u_i\}$
$(\boldsymbol{p})$ $\mathcal{L}_{EUC}$
$\mathcal{L}_{Error}$
b) multi-view classifier module
$\mathcal{L}_{UCL}^{co-user}$
$\mathcal{L}_{UCL}^{co-entity}$
$\mathcal{L}_{UCL}^{co-hashtag}$
per-class uncertainty
monitor
$\mathcal{L}_{Common}$
a) representation adjustment module

Fig. 2: The architecture of the proposed uncertainty-guided class imbalance learning framework (UCL$_{SED}$ and UCL-EC$_{SED}$). The whole framework contains two modules: a) representation adjustment module, in which larger margins are assigned to more uncertain classes to ensure class separability; b) multi-view classifier module, in which multi-view results are combined via Dempster-Shafer theory with an additional calibration method to ensure robustness and accuracy.

For simplicity, we remove the temperature parameter τ. With the parameter that controls class margin being set to constant zero, PSC loss has a weak capacity to learn separable representation distribution for imbalanced data. While in practice, events are diverse and complex and have highly imbalanced frequencies. All these properties make different events face varying difficulty levels to be well-represented and correctly classified. Concretely, if an event is hard to be distinguished by a classification model (i.e., the classification model is uncertain about its prediction), there is a high probability that this event faces a blurry boundary in the latent space and thus is easy to be misclassified. Therefore, an adaptive and flexible regularization loss that sets proper class-dependent margins is highly needed for better representation distribution for imbalanced event data. For events that are hard to distinguish, intuitively, we should enlarge the margins of other events towards them. In other words, the modified loss should push the distribution of other events away from them to prevent misclassification. To achieve that, we modify the original PSC loss with a tunable margin controlled by uncertainty, turning into UCL as follows:

$$\mathcal{L}_{UCL}\left(\mathbf{z}_i\right) = \log\left[1 + \sum_{c=1, c\neq y_i}^{\mathcal{C}} e^{\beta u_{y_i} + S(\mathbf{z}_i, \mathbf{p}_c) - S(\mathbf{z}_i, \mathbf{p}_{y_i})}\right], \tag{7}$$

where u_{y_i} represents the uncertainty value of class y_i and works as an effective indicator to reflect its current representation distribution status. β is a positive hyperparameter. The computation of u_{y_i} will be introduced in Sec. 4.3.

Comparison with the original PSC loss. The modified UCL added with an additional positive value encourages a larger margin between prototypes. Therefore, it simultaneously enhances the intra-class compactness and inter-class discrepancy. Moreover, this loss can be considered as a soft approximation to $\max(0, \beta u_{y_i} + S(\mathbf{z}_i, \mathbf{p}_c) - S(\mathbf{z}_i, \mathbf{p}_{y_i})), c = \arg\max_{c\neq y_i}(S(\mathbf{z}_i, \mathbf{p}_c))$. With the per-class margin being set to be positively related to class uncertainty, this loss is class-dependent. Meanwhile, the existence of the extra penalty βu_{y_i} forces even larger margins between $S(\mathbf{z}_i, \mathbf{p}_{y_i})$ and $S(\mathbf{z}_i, \mathbf{p}_c)$. Thus, events with large uncertainty are pushed more away from other classes to avoid class overlapping. In this way, the representation boundary of each class can be adjusted automatically and properly.

4.2.2 *Uncertainty-Guided Contrastive Learning Loss With Estimated Centroids (UCL-EC)*

A variant of UCL with estimated centroids (UCL-EC). As seen in Algo. 1, the prototypes in the UCL are learned during training and updated in each batch. Considering samples in different batches vary a lot, the learned prototypes may fluctuate greatly in different batches. This training instability problem is particularly severe for imbalanced data, where most classes are minority classes and are likely to be sampled in different batches. We replaced the prototypes with the global class centroids calculated beyond batch data to stabilize the training process. However, calculating class centroids in the full dataset costs time and computation resources. To make a trade-off between training stability and resource consumption, instead of updating centroids after the training of each batch, we estimate the centroids at the beginning of every epoch and keep them fixed in memory for the duration of the whole epoch.

4.3 Multi-View Classifier Module

4.3.1 *Single-View Uncertainty From EDL*

There are two kinds of uncertainty: epistemic uncertainty, also called model uncertainty, results from limited knowledge; aleatoric uncertainty, also named data uncertainty, is the noise inherent from class overlap. For instance, samples distributed in the blurry class boundary have high aleatoric

Algorithm 1: Uncertainty-guided class imbalance learning framework (UCL$_{SED}$) for imbalanced social event detection.

```
Input: Imbalanced event dataset $\mathcal{X}$ with corresponding
       labels $\mathcal{Y} = \{1, 2, ..., C\}$, maximal epoch number
       $E$, views $v \in \{$co-hashtag, co-entity, co-user$\}$, the
       number of GNN layers $L$, and the number of
       mini-batches $B$
Output: Parameters of the GNN encoder model $f(\theta)$,
        view-specific prototypes $p_c^v, c \in \{1, 2, ..., C\}$,
        parameters of the view-specific classifiers $g^v(\theta)$
for $v \in \{$co-hashtag, co-entity, co-user$\}$ do
    construct view-specific message graph $G^v$ as [21]
Initialize the parameters $f(\theta)$ and $g^v(\theta)$, initialize the
  view-specific prototypes $p_c^v, c \in \{1, 2, ..., C\}$, initialize
  the per-class uncertainty $[u_1, u_2, ..., u_C]$ as
  $[1 - \epsilon, 1 - \epsilon, ..., 1 - \epsilon]$, where $\epsilon$ is a small value.
for $e = 1, 2, ..., E$ do
    for $b = 1, 2, ..., B$ do
        Sample a mini-batch of messages $\{m_b\}$
        for $v \in \{$co-hashtag, co-entity, co-user$\}$ do
            for $l = 1, 2, ..., L$ do
                Obtain $h_i^{v(l)}, i\forall\{m_b\}$ via Eq. 2
            $z_i^v \leftarrow h_i^{v(L)}, i\forall\{m_b\}$
            Calculate $\mathcal{L}_{UCL}^v$ via Eq. 7
            $[e_{i1}^v, e_{i2}^v, ..., e_{iC}^v] \leftarrow$ EDL classifier $g^v$
            Obtain $[b_{i1}^v, b_{i2}^v, ..., b_{iC}^v, u_i^v]$ via Eq. 8
        Obtain $[b_{i1}, b_{i2}, ..., b_{iC}, u_i]$ via Eq. 9
        Calculate $\mathcal{L}_{Error}$ via Eq. 14
        Calculate $\mathcal{L}_{EUC}$ via Eq. 11
        Calculate $\mathcal{L}_{Common}$ via Eq. 13
        Calculate $\mathcal{L}_{Total}$ via Eq. 12
        Update $f(\theta)$, $g^v(\theta)$ and prototypes $p_c^v$
    Update per-class uncertainty $[u_1, u_2, ..., u_C]$
```

uncertainty. This work uses the measured class aleatoric uncertainty to monitor the class representation status. Intuitively, we assume a class with high aleatoric uncertainty is not well represented and should be emphasized in the representation adjustment module. In recent years, evidential deep learning (EDL) has been proposed to estimate aleatoric uncertainties by directly estimating parameters of the predictive posterior based on the output of the deep neural networks [64].

EDL for single-view event detection. Under the framework of Subjective Logic and Dempster-Shafer theory [65], EDL provides a principled way to jointly model high-order probabilities for a prediction and model uncertainty for the overall decision. Specifically, it assumes a Dirichlet distribution as the conjugate before the Multinomial distribution to represent the density of class probability assignment. The belief mass assignment to each event class and the overall uncertainty mass are determined over the Dirichlet distribution, and the Dirichlet parameters are induced from the collected evidence learned by the neural network.

Formally, for each single view v, where $v \in \{$*co-hashtag, co-entity, co-user*$\}$, suppose there are C mutually exclusive events, the Dirichlet distribution of the $i-th$ sample $\alpha_{\mathbf{i}}{}^v = [\alpha_{i1}^v, \alpha_{i2}^v, ..., \alpha_{iC}^v]$ is induced from the evidence $\mathbf{e_i}{}^v = [e_{i1}^v, e_{i2}^v, ..., e_{iC}^v]$ collected from the data with the relation $\alpha_{ic}^v = e_{ic}^v + 1$, $c \in \{1, 2, ..., C\}$. The belief mass assignment to each event, as well as the overall uncertainty mass, is computed as follows:

$$b_{ic}^v = \frac{e_{ic}^v}{S_i^v}, u_i^v = \frac{C}{S_i^v}, \tag{8}$$

where $S_i^v = \sum_{c=1}^C e_{ic}^v + 1 = \sum_{c=1}^C \alpha_{ic}^v$ is referred to as the Dirichlet strength. Obviously, more evidence ensures less uncertainty.

4.3.2 Multi-View Uncertainty Via DST

After getting evidence-based single-view opinions, to ensure a more robust final result, we combine them together via Dempster-Shafer theory. The combination rule, also known as Dempster's rule, strongly emphasizes the agreement between multiple views and extracts their common shared beliefs as the final judgment. Specifically, for the $i-th$ sample, we need to combine three independent sets of mass assignments $M_i^v = \{\{b_{ic}^v\}, u_i^v\}$, where $v \in \{$*co-hashtag, co-entity, co-user*$\}$. Here we utilize $\oplus$ to denote the dempster's combination rule in combining two independent views. The detailed calculation is as follows:

$$\begin{aligned} &M_i = M_i^{v_1} \oplus M_i^{v_2}, \\ &b_{ic} = \frac{1}{1 - T_i}\left(b_{ic}^{v_1} b_{ic}^{v_2} + b_{ic}^{v_1} u_i^{v_2} + b_{ic}^{v_2} u_i^{v_1}\right), u_i = \frac{1}{1 - T_i} u_i^{v_1} u_i^{v_2}, \\ &T_i = \sum_{j \neq k} b_{ij}^{v_1} b_{ik}^{v_2}. \end{aligned} \tag{9}$$

The combination rule can be further extended to the multi-view case. As the case in this paper, we have three sets of masses (i.e., masses learned under the three single views: *co-hashtag*, *co-entity* and *co-user*). The final results can be obtained sequentially as follows:

$$M_i = M_i^{v_1} \oplus M_i^{v_2} \oplus M_i^{v_3}, \tag{10}$$

where v_1, v_2 and v_3 correspond to *co-hashtag*, *co-entity* and *co-user* views, respectively.

The above procedure describes the calculation of the uncertainty of each sample in detail. As for the uncertainty value of each class, we use the average uncertainty values of all the training samples within that class as its class uncertainty.

4.3.3 Uncertainty Calibration Method

Though the uncertainty can be modelled directly with EDL and DST, it may not be well calibrated [66]. Considering the important role the estimated uncertainty plays, we need it to be as accurate as possible to reflect the status of representation learning. We adopt an uncertainty calibration method to build the correct relationship between accuracy and uncertainty. This is also inspired by previous calibration studies [67], [68], which point out that a well-calibrated model should be confident when its prediction is accurate and be uncertain when its prediction is inaccurate.

Generally, there are four possible outputs: (1) Accurate and Certain (AC), (2) Accurate and Uncertain (AU), (3) Inaccurate and Certain (IC), and (4) Inaccurate and Uncertain (IU). We propose a calibration method encouraging the multi-view classifier to output more AC and IU samples. y_i denotes the ground-truth label of sample i while $\hat{y}_i$ denotes the prediction of model. α_i is the obtained Dirichlet parameter. $\tilde{\alpha}_i = \mathbf{y}_i + (1 - \mathbf{y}_i) \odot \alpha_i$ is the Dirichlet parameters after removal of the correct evidence for the true class. Specifically, when the model makes a firm and accurate prediction ($\hat{y}_i = y_i$ and $\max(\mathbf{p}_i) \rightarrow 1$), we force it to give a relatively low uncertainty by increasing the total evidence strength ($S_i \rightarrow \infty$). When the model predicts falsely, we force it to give a high uncertainty by making misleading evidence shrink to zero ($\tilde{\alpha}_i \rightarrow \mathbf{1}$):

$$\begin{aligned} \mathcal{L}_{EUC}(\mathbf{p}_i) = \lambda_e(- & \sum_{i \in \{\hat{y}_i = y_i\}} \max(\mathbf{p}_i) \log\left(1 - C/S_i\right) \\ & + \sum_{i \in \{\hat{y}_i \neq y_i\}} \mathrm{KL}\left[D\left(\mathbf{p}_i \mid \tilde{\alpha}_i\right) \| D\left(\mathbf{p}_i \mid \mathbf{1}\right)\right]), \\ & \lambda_e = \min(1.0, e/25), \end{aligned} \tag{11}$$

where C denotes the total number of event classes. S_i represents the total Dirichlet strength and C/S_i is the uncertainty of sample i. The KL term represents the Kullback-Leibler divergence between the wrong and uniform evidence distribution. $D(\cdot|\cdot)$ denotes the multinomial opinions formed by the Dirichlet parameter. The first term encourages AC outputs by ensuring more collected evidence, while the second term tries to give IU outputs by removing all wrong evidence $\tilde{\alpha}_i$. Meanwhile, considering that the learned evidence in the early epochs tends to be inaccurate, we also adopt an annealing coefficient λ_e to dynamically adjust the weight of calibration loss. e denotes the index of the current epoch. In the initial epoch, the class uncertainty used in the UCL loss is set to $1 - \epsilon$, where ϵ is a very small value.

Class Uncertainty: Similar to the calculation of estimated class centroids, to make a trade-off between training stability and resource consumption, we update the class uncertainty in the full dataset every epoch and keep them fixed in memory for the duration of the whole epoch. The class uncertainty is the average uncertainty value of all the training samples within that class.

4.4 Optimization Objective

The optimization objective includes loss from the representation adjustment module and loss from the multi-view classifier module, termed as:

$$\begin{aligned} \mathcal{L}_{Total} = \mathcal{L}_{Error} + \lambda_1 \mathcal{L}_{EUC} + \lambda_2 \mathcal{L}_{UCL}^{v} + \lambda_3 \mathcal{L}_{Common} \\ v \in \{\textit{co-hashtag}, \textit{co-entity}, \textit{co-user}\}, \end{aligned} \tag{12}$$

where λ_1, λ_2 and λ_3 are hyper-parameters. The latter two terms are in the representation module. Specifically, $\mathcal{L}_{UCL}^{v}$ is the proposed uncertainty-guided contrastive learning loss which aims to regularize representations in each view. $\mathcal{L}_{Common}$ is designed to tackle the deficiency of Dempster's rule in handling high-conflict data by ensuring the multi-view commonality. Here we denote the normalized embeddings over a batch of training samples from a specific view as $\mathbf{H}_{nor}^{v}, v \in \{\textit{co-hashtag}, \textit{co-entity}, \textit{co-user}\}$. The similarity of nodes $\mathbf{Sim}^{v}$ is computed as $\mathbf{H}_{nor}^{v} \cdot (\mathbf{H}_{nor}^{v})^{T}$. $\mathcal{L}_{Common}$ gives the following constraint:

$$\begin{aligned} \mathcal{L}_{Common} = & \left\| \mathbf{Sim}^{co-hashtag} - \mathbf{Sim}^{co-entity} \right\|_F^2 \\ & + \left\| \mathbf{Sim}^{co-hashtag} - \mathbf{Sim}^{co-user} \right\|_F^2 \\ & + \left\| \mathbf{Sim}^{co-entity} - \mathbf{Sim}^{co-user} \right\|_F^2 . \end{aligned} \tag{13}$$

The former two terms $\mathcal{L}_{Error}$ and $\mathcal{L}_{EUC}$ are from the multi-view classifier module. Specifically, $\mathcal{L}_{EUC}$ is proposed to calibrate the calculated multi-view uncertainty. $\mathcal{L}_{Error}$ denotes the prediction error loss, integral to the classical cross-entropy loss function over the learned Dirichlet distribution.

$$\mathcal{L}_{Error} = \sum_{i} \int \left[\sum_{j=1}^{C} -y_{ij} \log\left(p_{ij}\right) \right] \frac{1}{B\left(\alpha_i\right)} \prod_{j=1}^{C} p_{ij}^{\alpha_{ij}-1} d\mathbf{p}_i, \tag{14}$$

where $\mathbf{y}_i$ is the true class distribution. $\mathbf{p}_i$ is the class assignment probabilities on a simplex and $B(\cdot)$ is the multinomial beta function.

4.5 Time complexity analysis

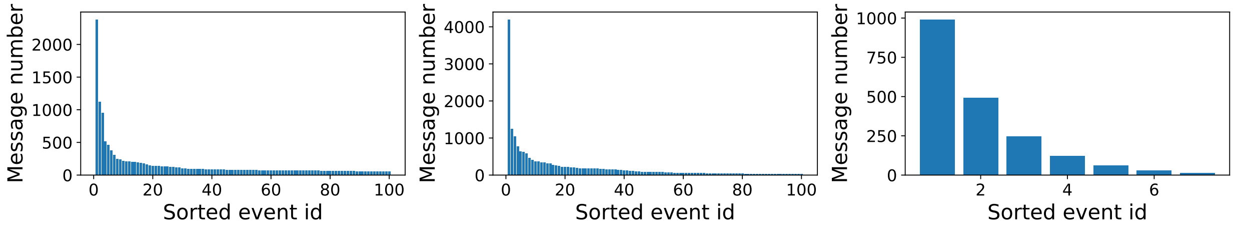
The total time complexity of UCL$_{SED}$ is about $O(\sum_{v \in V} N_e^v)$, where V represents the set of views. N_e^v denotes the total number of edges under the specific view v. That means the time complexity is approximately linear with the multi-view graph size. Specifically, as node features are low-dimensional and $N_e^v \gg N$, the propagation of GNN encoder under all the views (Algorithm 1 lines 8-9) takes $O(|V|Ndd' + \sum_{v \in V} N_e^v d') = O(\sum_{v \in V} N_e^v)$, where $|V|$ denotes the total number of views. N is the total number of messages. d and d' are the input and output dimensions of the propagation layer. The time complexity of UCL loss under all the views (Algorithm 1 line 11) can be roughly estimated as $O(|V|NCd')$, where C denotes the total number of classes. As for the EDL neural network (Algorithm 1 line 12), its time complexity under all the views is about $O(|V|Nd'C)$. Additionally, it takes $O(|V|NC)$ to calculate the view-specific uncertainty (Algorithm 1 line 13), $O(|V-1|N(C+1)^2)$ to multi-view uncertainty (Algorithm 1 line 14), $O(NC)$ to $\mathcal{L}_{EUC}$ and $\mathcal{L}_{Error}$ (Algorithm 1 line 15 and line 16), and $O(|V|(|V|-1)\sum_{b=1}^{B} |m_b|^2 d')$ to $\mathcal{L}_{Common}$ (Algorithm 1 line 17), where $|m_b|$ denotes the batch size and B is the number of mini-batches. Similar to UCL$_{SED}$, the total time complexity of UCL-EC$_{SED}$ is also about $O(\sum_{v \in V} N_e^v)$. The only difference lies in the additional $O(Nd')$ taken to calculate those prototypes, which can be neglected.

5 Experiments

5.1 Experimental Setup

5.1.1 Datasets and evaluation metric

We conduct experiments on three imbalanced social event datasets: Events2012_100, Events2018_100, and CrisisLexT_7. The former two datasets are sampled from Events2012 [69] and Events2018 [70], respectively. Considering that events in the original Events2012 and Events2018



(a) Events2012_100 (b) Events2018_100 (c) CrisisLexT_7

Fig. 3: Detailed dataset statistics. (a), (b) and (c) show the number of messages per event on Events2012-100, Events2018-100, and CrisisLexT-7, respectively.

datasets are with different frequencies, to reconstruct datasets following long-tail distribution, we select 100 events and reorder them based on the number of tweets each event contains. The final Events2012_100 consists of 15019 tweets relating to 100 events. With maximally 2377 tweets and minimally 55 tweets per event, the imbalance ratio is about 43. Similarly, the final Events2018_100 contains a total number of 19944 tweets. With maximally 4189 tweets and minimally 27 tweets per event, its imbalance ratio is about 155. CrisisLexT_7 is sampled from CrisisLexT-26 [71], a balanced dataset containing 26 crisis events in multiple languages. Here we only select crisis events in English. Meanwhile, to ensure the long-tail distribution of the reconstructed CrisisLexT_7 dataset, we calculate the number of each event based on the exponential function: $n_i = n_{max}\gamma^i$, where i is the event class index. γ is set to 0.5 in our experiments. With maximally 989 tweets and minimally 15 tweets per event, the imbalance ratio of CrisisLexT_7 is about 66. More details are shown in Fig. 3.

The statistics mentioned above depict the datasets for training. Generally, an imbalanced training set, balanced validation, and test sets should be provided to obtain a more fair and accurate model evaluation for the long-tail recognition task. Thus, for the validation and test sets in our experiments, we select additional 20 and 30 tweets for each event, respectively. As for evaluation metrics, we simply choose the two commonly used metrics in classification tasks: Accuracy (ACC) and F1 value (F1).

5.1.2 Our proposed algorithm

According to how we learn and update the class prototypes, our uncertainty-guided class imbalance learning framework has the following variants: (1) UCL$_{SED}$, which automatically learns the parameters of class prototypes and updates them every batch; (2) UCL-EC$_{SED}$, which uses the global class centroids as class prototypes and updates them every epoch. Please refer to Sec. 4.2.2 for more details.

5.1.3 Baselines

To verify the effectiveness of our proposed UCL$_{SED}$ and UCL-EC$_{SED}$ in detecting events from severely imbalanced datasets, we compare our methods with state-of-the-art techniques in the social event detection domain. Furthermore, to demonstrate the superiority of the proposed UCL loss in learning distinguishable representations for imbalanced datasets, we also compare our method with existing benchmark methods in long-tail recognition tasks. Overall, the selected baselines are listed in the following two groups.

TABLE 1: Dataset details under different views.

View	Num of correct edges/Num of all edges		
	Events2012_100	Events2018_100	CrisisLexT_7
co-hashtag	0.7355	0.8572	0.8778
co-entity	0.1976	0.6026	0.9257
co-user	0.8847	0.7030	0.8707
all	0.2323	0.7234	0.9121

Social event detection methods: The selected social event detection baselines are pre-trained language models: (1) Word2Vec [41] - we leverage the pre-trained word embeddings to get the message vectors and adopt a two-layer neural network to classify them; (2) BERT [42] - we finetune it on our datasets and make the final classification. Topic models: (3) TwitterLDA [71] obtains message representations by learning topic and word distributions. GNN-based models: (4) PP-GCN [37], which first builds a weighted adjacent matrix by measuring event similarity, then leverages a graph convolutional network trained by pair-wise sampling to obtain discriminate message representations; (5) KPGNN [38], which connects messages sharing common elements, then uses a multi-head graph attention network to learn message representations; (6) MVGAN [24], which learns message representations from both semantic and temporal views and uses an attention mechanism to fuse them; (7) ETGNN [21], which learns message vectors from *co-hashtag*, *co-entity* and *co-user* views and uses Dempster–Shafer theory to combine them.

Long-tail recognition methods: We also compare our methods with several long-tail recognition methods. Note that (1) CE (i.e., Cross-Entropy) is the vanilla baseline, using the cross-entropy loss to train our multi-view framework. Other baselines include loss manipulation methods: (2) CB+Focal [7] (i.e., Class-Balanced Focal loss), which combines a re-weighting scheme with the original focal loss by assigning weights to different classes based on their sample numbers; (3) LDAM loss [57], which enforces class-dependent margins based on class frequencies; long-tail representation improvement methods: (4) Hybrid-PSC [17], which is a hybrid framework with a supervised contrastive learning branch for representation regularization and a classifier branch for bias elimination; (5) BCL [19], which further improves the original supervised contrastive learn-

TABLE 2: Comparison with social event detection methods.

Methods	Events2012_100		Events2018_100		CrisisLexT_7	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
TwitterLDA [25]	9.37±.44	8.27±.49	6.90±.51	4.83±.60	31.90±.53	22.59±.58
Word2Vec [41]	74.67±.56	74.89±.59	35.17±.42	33.89±.41	44.29±.54	37.80±.57
BERT [42]	79.11±.38	79.28±.46	56.39±.57	54.07±.61	69.43±.70	66.29±.67
PP-GCN [37]	63.33±.34	54.62±.25	70.00±.39	50.99±.46	73.33±.53	68.71±.44
KPGNN [38]	73.33±.26	59.08±.33	76.67±.38	61.90±.41	75.67±.52	71.11±.60
MVGNN [24]	81.83±.29	82.14±.31	69.93±.42	68.17±.47	71.90±.49	70.55±.57
ETGNN [21]	86.45±.23	86.56±.27	60.43±.31	60.12±.33	76.67±.42	74.30±.50
UCL$_{SED}$	92.21±.30	92.01±.35	78.16±.37	78.77±.41	80.81±.60	80.67±.62
UCL-EC$_{SED}$	**93.18±.20**	**93.27±.29**	**78.91±.25**	**79.11±.29**	**84.33±.41**	**83.96±53**

TABLE 3: Comparison with long-tail recognition methods.

Methods	Events2012_100		Events2018_100		CrisisLexT_7	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
CE	85.77±.26	85.88±.30	73.73±.35	74.00±.38	74.76±.43	72.31±.49
CB+Focal [7]	87.67±.40	86.84±.43	75.57±.37	75.23±.46	75.24±.53	74.95±.66
LDAM [57]	89.83±.29	89.95±.24	77.30±.36	78.14±.44	76.84±.49	76.61±.42
Hybrid-PSC [17]	88.57±.26	88.65±.33	76.87±.39	76.60±.47	78.10±.43	76.20±.52
BCL [19]	90.83±.34	90.98±.37	77.33±.24	78.16±.26	81.90±.47	81.32±.45
DRO-LT [20]	89.43±.28	89.42±.25	77.07±.34	77.52±.40	78.86±.45	78.30±.53
TSC [18]	90.33±.24	90.80±.26	77.40±.35	78.33±.36	81.43±.42	80.58±.48
UCL$_{SED}$	92.21±.30	92.01±.35	78.16±.37	78.77±.41	80.81±.60	80.67±.62
+CB	92.83±.35	92.80±.38	78.70±.45	78.97±.50	81.57±.68	81.25±.74
UCL-EC$_{SED}$	93.18±.20	93.27±.29	78.91±.25	79.11±.29	84.33±.41	83.96±.53
+CB	**93.67±.31**	**93.66±.36**	**79.33±.42**	**79.27±.41**	**84.60±.51**	**84.24±.58**

ing by considering class averaging and class complement; (6) DRO-LT [20], which builds on distributional robustness optimization and explicitly seeks to improve the quality of representations for tail classes; (7) TSC [18], which uses predefined features to guide representation learning.

5.1.4 Experimental Settings and Implementations

The proposed UCL$_{SED}$ framework combines the backbone model ETGNN [21] with an additional representation adjustment module and a multi-view classifier module with an improved uncertainty calibration method. We set the batch size to 1500, the layer of temporal-aware GNN to 2, and the dimensions of the first and second GNN layers to 256. As for the representation adjustment module, we set β in the UCL and UCL-EC to 0.1. Each EDL classifier is designed as a two-layer neural network with an activation layer ReLU in the multi-view classifier module. The hidden layer dimension of EDL is 128. As for those hyper-parameters in the total optimization objective function, we set λ_1, which controls the intensities of uncertainty calibration, to 1, λ_2, which controls UCL to 0.1, and λ_3 which ensures multi-view commonality to 0.5. The framework is trained using Adam optimizer with the learning rate 0.001. The maximal training epoch is 100. Experiments are implemented in Python 3.8 and Pytorch 1.9 and conducted on $8\times$GeForce RTX 3090 GPU. To avoid the one-time occasionality, in comparison experiments, we perform 10 tests for all models and record the mean and standard deviation values.

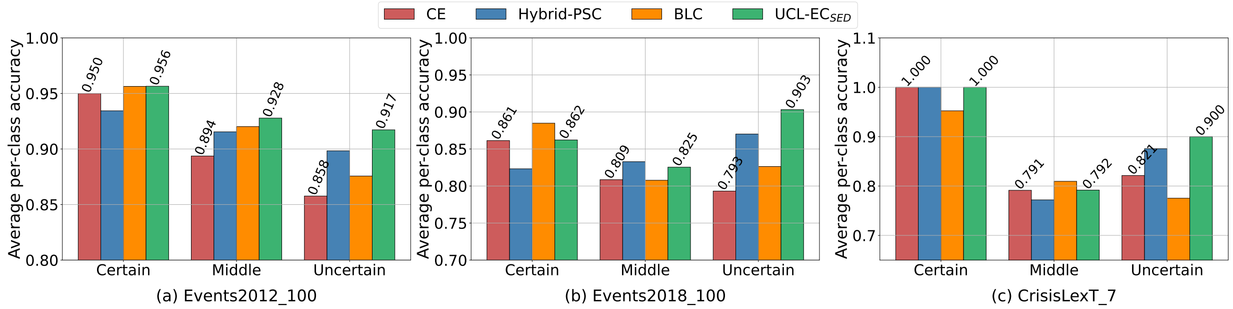
5.2 Results and Comparisons

5.2.1 Comparison with social event detection methods

We first compare our approaches with 7 competitive social event detection methods and report results in Table 2. By carefully analyzing the results, we have the following observations: (1) Our proposed model (i.e., UCL-EC$_{SED}$) achieves state-of-the-art performance on all three imbalanced datasets. On Events2018_100, UCL-EC$_{SED}$ even surpasses ETGNN by about 19%. Because our models capture the view-specific reliability better by adding the additional uncertainty calibration method and therefore, make up for the shortcomings of ETGNN in handling data whose most views are noisy. (2) We also noticed that the results of those GNN-based baselines vary greatly on different datasets and ETGNN has difficulty in handling datasets whose views are noisy. For example, on Events2012_100, ETGNN achieves a remarkable accuracy improvement of 13.12% compared with KPGNN. While on Events2018_100, KPGNN has a relative improvement of 16.24% compared with ETGNN. The connection qualities of the constructed social graphs determine this. Here we depict the detailed connection qualities in Table 1. Note that edges under the view "*all*" are the union of edges under the above three single views. In KPGNN, information is propagated and aggregated over a homogeneous message graph constructed under the "*all*" view. While in ETGNN, representations of three single views are learned independently, and the obtained view-specific results are further combined via Dempster-Shafer theory to get the final decision. As shown in Table 1, on Events2012_100, the connection quality of the *co-entity* view is quite low, which also leads to the low quality of the "*all*" view. Therefore, KPGNN performs badly. Meanwhile, considering most views (i.e., *co-hashtag* and *co-user*) are of relatively high quality, ETGNN can still obtain trusted results by combining multi-view results via Dempster-Shafer theory. However, on Events2018_100, the connection qualities under most views (i.e., *co-entity* and *co-user*) are not that good. Meanwhile, the estimated uncertainty from ETGNN is not well calibrated. Therefore, ETGNN performs poorly. To sum up, with the help of the UCL loss, our models are aware of the per-class representation status during training and make timely adjustments to their boundaries. More analysis of the UCL and UCL-EC losses will be made later.

5.2.2 Comparison with long-tail recognition methods

Note that this work focuses on social event detection in imbalanced data. The proposed UCL method aims to enhance the generalization capacity by regularizing the representation learning. Thus, we also compare our methods with the vanilla CE baseline and six state-of-the-art long-tail recognition methods, especially with those long-tail representation improvement methods. The results are presented in Table 3. Our UCL-EC$_{SED}$ consistently outperforms all the baselines on all three datasets, emphasizing the superiority of enforcing margins in the feature space under the guidance of per-class uncertainty.



(a) Events2012_100

(b) Events2018_100

(c) CrisisLexT_7

Fig. 4: Average per-class accuracy and F1 value of the certain, middle, and uncertain groups.

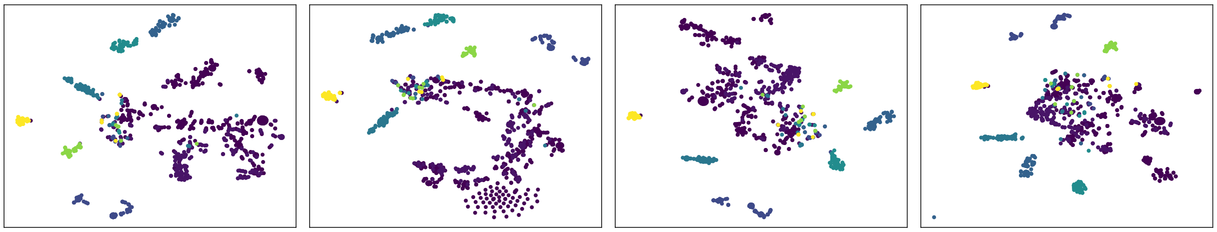
We here analyze the experimental results in detail. As seen from Table 3, CE performs worst among all the baselines. This is due to the limitation of the original cross-entropy loss in handling imbalanced data. To deal with the imbalanced data distribution, CB+Focal manipulates the focal loss by assigning more weights to classes with less effective samples. As a loss re-weighting method, CB+Focal slightly improves the detection results. LDAM is also a loss manipulation method. Instead of modifying per-class weight, it sets different logit margins for different classes based on frequency. LDAM obtains a relatively good performance. This method focuses on improving the output layer of the final classifier while the remaining methods consider improving the output layer of the GNN model. In Table 3, Hybrid-PSC, which applies prototypical supervised contrastive loss to learn distinguishable features, outperforms the CE counterpart. For example, on ACC, Hybrid-PSC outperforms CE by 2.8%, 3.14%, and 3.34% on Events2012_100, Events2018_100, and CrisisLexT_7, respectively. This validates the idea that better representation can help distinguish event classes. We also noticed that other methods that tailor the contrastive learning loss for imbalanced datasets achieve better results than Hybrid-PSC. For example, TSC pre-computes a set of targets uniformly to ensure data balance. However, it is not flexible enough as it has no ability to make proper adjustments for different classes. DRO-LT extends prototypical contrastive learning by introducing distributional robustness. It learns separable per-class representation by pushing and pulling towards a worst-case possible distribution. BCL is also a representation improvement method that applies contrastive learning. It implements class-averaging and class-complement to the original contrastive learning loss to enhance representation learning. Table 3 shows a significant performance boost when comparing the tailored BCL with Hybrid-PSC. DRO-LT and BCL methods are tailored for imbalanced data under frequency guidance. However, as observed from Fig 1, evidential uncertainty is a better indicator of model generalization capacity than class frequency. We, therefore, enforce larger margins for uncertain classes during representation learning. The result that our UCL-EC$_{SED}$ surpasses all the baselines validates the superiority of our uncertainty-guided learning.

For a more fine-grained understanding, we also split all the labels into three groups based on their measured per-class uncertainties and plot the final group results in Fig. 4. Concretely, we divide all uncertainty values into three intervals. Assume the maximal and minimal class uncertainty values are denoted as U_{max} and U_{min}, classes whose uncertainty values are within $[U_{min}, U_{min} + 1/3(U_{max} - U_{min})]$ are split into certain classes. Classes within $[U_{min}+1/3(U_{max}-U_{min}), U_{min}+2/3(U_{max}-U_{min})]$ are middle and the rest are uncertain. Fig. 4 reveals that our UCL-EC$_{SED}$ achieves a large gain on the uncertain group without sacrificing the detection performances on the certain and middle groups. This further demonstrates the robustness of our model. Adding proper uncertainty-guided margins during training makes class representations in all groups more separable. Besides, it is observed that UCL-EC$_{SED}$ performs even better than UCL$_{SED}$. We argue this is due to the gap in sample distribution in different batches.

Remark: unlike most classical long-tail recognition methods that directly act on the classifier (e.g., re-weighting strategies), our work tends to solve the imbalance problem by manipulating the latent feature space. By ensuring the learned representations of minority event classes are well-separated from other event classes, it becomes much easier to recognize them. Our methods work differently from classical methods, which means they are parallel and may complement each other. To validate this opinion, we here further incorporate the re-weighting strategy (i.e., CB in [7]) into the classification Error loss $\mathcal{L}_{Error}$ of our framework and record the results. Encouragingly, the performance gets further improved. For example, UCL-EC$_{SED}$ + CB gets a further 0.49% improvement on ACC on Events2012_100.

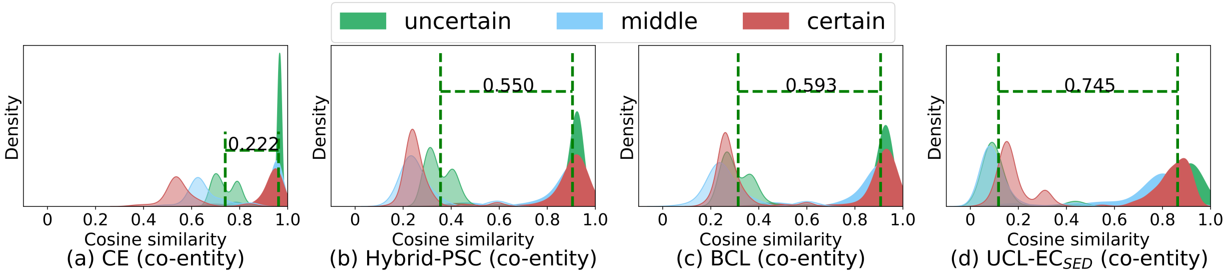
5.3 Representation Visualization

To make a better analysis of representations learned by baselines (CE, Hybrid-PSC, BCL) and our model (UCL-EC$_{SED}$), in Fig. 5, we plot the t-SNE results of eight randomly selected events of *co-entity* view on Events2012_100. Obviously, the boundary learned by our model is less blurry compared to other baselines. Compared to Hybrid-PSC which adopts the original PSC, the class overlapping problem gets well alleviated by our work. To further demonstrate the extent to which our UCL loss helps adjust a clear separable boundary in the latent space, we also visualize the mean intra-class similarity distribution and mean inter-class similarity distribution of samples in certain, middle and uncertain groups on the Events2012_100 dataset. The results are plotted in Fig. 6. Note that the dark area represents intra-class cosine similarities while the light color represents inter-class ones. As can be observed, the representations of UCL-EC$_{SED}$ are the best for all three groups (uncertain,



(a) CE (co-entity) (b) Hybrid-PSC (co-entity) (c) BCL (co-entity) (d) UCL-EC$_{SED}$ (co-entity)

Fig. 5: t-SNE visualization of the learned features of *co-entity* view on Events2012_100. Here we randomly select 8 events which are drawn in different colors.



(a) CE (co-entity) (b) Hybrid-PSC (co-entity) (c) BCL (co-entity) (d) UCL-EC$_{SED}$ (co-entity)

Fig. 6: Visualization of mean intra-class and inter-class cosine similarity distribution of *co-entity* view on Events2012_100. Dark color area indicates intra-class similarities while light color area indicates inter-class ones. Uncertain, middle, and certain classes are plotted separately. Average inter-class cosine similarity and average intra-class similarity of the uncertain group are marked with dashed green vertical lines.

middle, and certain) under all three views. The inter-class similarity of UCL-EC$_{SED}$ gets lower compared to the competitive baseline - BCL, which is attributed to the added uncertainty-guided margin in the UCL loss. By adding a tunable margin, UCL pushes the distribution of other classes away from uncertain classes and, therefore, gets more separable representations. Consistent with the results in Table 3, representations learned by CE are the worst. Their intra-class similarities and inter-class similarities under all three views are closest. Compared to CE, Hybrid-PSC decreases the inter-class similarities significantly, owing to the ability of contrastive learning to push inter-class samples away. Similar to Hybrid-PSC, BCL decreases inter-class similarities, especially for uncertain samples. We argue that may be because tail classes are more likely to be uncertain classes. BCL has a stronger ability to deal with tail classes thanks to its class-averaging and class-compensation strategies. Thus BCL also performs well in uncertain classes.

5.4 Uncertainty Analysis

In this section, we conduct experiments to validate that the estimated uncertainty is highly correlated to the model performance and therefore, is a good indicator to adjust representation distribution. We here visualize the estimated uncertainty of the true and false predictions in the validation set. The results are drawn in Fig. 7. Obviously, on all three datasets, higher uncertainties are usually estimated for those false predictions while lower uncertainties are more likely to belong to those true predictions. This observation implies the effectiveness of using estimated uncertainty to indicate the status of representation learning since the estimated uncertainty is correlated to the prediction performance.

5.5 Hyper-parameter Sensitivity

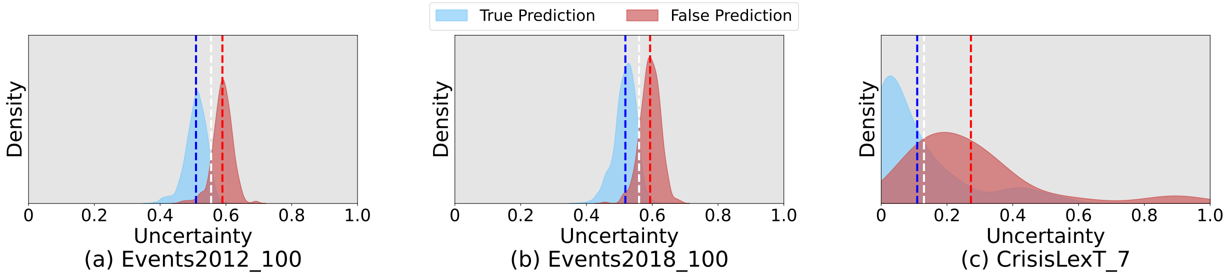
In this section, we study the sensitivities of parameters λ_1, λ_2 and λ_3 in the optimization objective function (i.e., Eq. 12). Due to the page limit, we only plot the results on Events2012_100 in Fig. 8.

5.5.1 *Analysis of coefficient λ_1.*

The hyper-parameter λ_1 in Eq. 12 controls how accurate the estimated uncertainty is. We vary it from 0.001 to 10. The results are shown in Fig. 8(a). With the increase of λ_1, the performance rises first. Because the uncertainty calibration loss helps in both classification and representation adjustment modules. By forcing wrong evidence to shrink to zero and highlighting those correct parts, it assists the learning of EDL neural networks. Meanwhile, more accurate uncertainty estimation helps adjust representation. However, the result drops rapidly when λ_1 reaches a relatively large value. Because too much emphasis is placed on eliminating wrong evidence in early training. In early epochs, misclassified samples are dominant. A large λ_1 may cause premature convergence to the uniform distribution, thus preventing the model from classifying correctly.

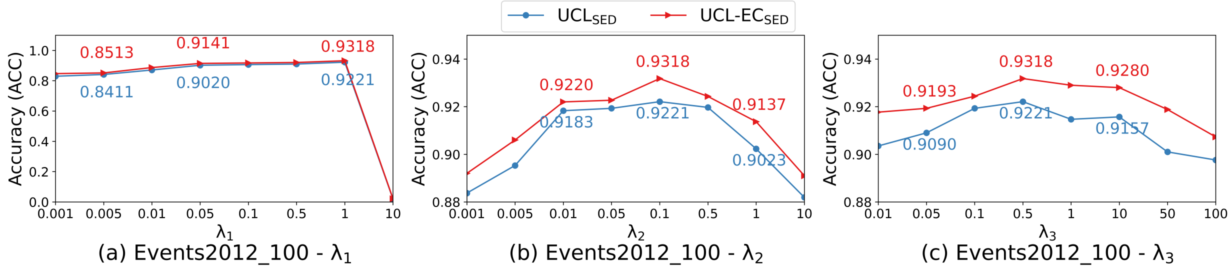
5.5.2 *Analysis of coefficient λ_2.*

The hyper-parameter λ_2 in Eq. 12 controls the impact of the representation adjustment module, which intends for the GNN model to learn separable features. We vary it from 0.001 to 10. The results are shown in Fig. 8(b). Similarly, as λ_2 becomes larger, the accuracy scores increase first and then decrease. Because in this work, instead of measuring the quality of the learned representations, the selected metrics in fact focus on the classification results. Though better representation helps better classification. The overwhelming



(a) Events2012_100 (b) Events2018_100 (c) CrisisLexT_7

Fig. 7: Visualization of uncertainty distribution on Events2012_100, Events2018_100, and CrisisLexT_7.



(a) Events2012_100 - λ_1 (b) Events2012_100 - λ_2 (c) Events2012_100 - λ_3

Fig. 8: Hyper-parameter sensitivity analysis.

emphasis on representation may lead to the under-learning of classifiers.

5.5.3 *Analysis of coefficient* λ_3.

The hyper-parameter λ_3 in Eq. 12 controls the consistency constraint on the three similarity matrices under different views. We vary it from 0.01 to 100. As shown in Fig. 8(c), the performance of our proposed framework is not very sensitive to λ_3 when it is within a reasonable range.

5.6 Ablation Study

In this section, we conduct quantitative ablation studies to analyze the components of our model. We record the results on Event2012_100 in Table 4 for illustration purposes.

5.6.1 *Ablation study on the UCL and UCL-EC loss.*

We argue that our work's proposed uncertainty-guided contrastive learning losses (i.e., UCL and UCL-EC) are expected to learn better features for imbalanced datasets, leading to better detection performance. To verify it, in the two proposed models UCL$_{SED}$ and UCL-EC$_{SED}$, we replace UCL and UCL-EC with their vanilla versions which remove the uncertainty-guided margins. As shown in Table 4, without the uncertainty-guided margin, the results have a drop. To make a more careful comparison, we also add a fixed margin (i.e., the $+m$ strategy) to PSC and PSC-EC and report the results. When adding a fixed margin, the results are slightly better than those of the vanilla version losses but worse than uncertainty-guided ones. This further demonstrates the superiority of UCL and UCL-EC in learning separable representation. What's more, to demonstrate the superiority of our uncertainty-related approach, we also compare our methods with the baseline adopting the $+dm$ strategy but removing the uncertainty-related parts (i.e., the calibration part in the classification head and the uncertainty adjustment part in the representation head) in the whole framework. In this baseline, the per-class error rate of the training set is used to adjust representation. As demonstrated in Table 4, it gets a great result while still worse than our uncertainty approach. Furthermore, in comparison to this baseline, our framework offers enhanced interpretability and reliability. It achieves this by providing uncertainty values during the prediction process.

TABLE 4: Ablation studies of proposed models on Events2012_100. The check mark indicates which losses are applied in the framework. $\mathcal{L}^v_{PSC}$ denotes the original Prototypical Supervised Contrastive loss. $+m$ means adding a fixed margin in the original PSC loss (introduced in Sec. 3.3 and Sec. 4.2.1). Similarly, $+dm$ means adding a dynamic margin controlled by the per-class error rate of the training set in each epoch. The superscript v (i.e., v in $\mathcal{L}^v_{UCL}$, $\mathcal{L}^v_{PSC}$, $\mathcal{L}^v_{UCL-EC}$ and $\mathcal{L}^v_{PSC-EC}$) denotes the three views, $v \in \{$*co-hashtag*, *co-entity*, *co-user*$\}$.

Methods	$\mathcal{L}^v_{UCL}$	$\mathcal{L}^v_{PSC}$	$+m$	$+dm$	$\mathcal{L}_{EUC}$	ACC	F1
UCL$_{SED}$	✓				✓	0.9221	0.9201
UCL$_{SED}$		✓			✓	0.8825	0.8838
UCL$_{SED}$		✓	✓		✓	0.8957	0.8955
UCL$_{SED}$		✓		✓		0.9073	0.9056
UCL$_{SED}$	✓					0.8193	0.8179
	$\mathcal{L}^v_{UCL-EC}$	$\mathcal{L}^v_{PSC-EC}$	$+m$	$+dm$	$\mathcal{L}_{EUC}$	ACC	F1
UCL-EC$_{SED}$	✓				✓	0.9318	0.9327
UCL-EC$_{SED}$		✓			✓	0.8910	0.8900
UCL-EC$_{SED}$		✓	✓		✓	0.8983	0.8975
UCL-EC$_{SED}$		✓		✓		0.9183	0.9145
UCL-EC$_{SED}$	✓					0.8230	0.8300

5.6.2 *Ablation study on the uncertainty calibration method*

The uncertainty calibration method (i.e., $\mathcal{L}_{EUC}$) is quite important to our model. As can be seen in Table 4, if we remove $\mathcal{L}_{EUC}$, the detection results have a significant decrease. For

TABLE 5: Time consumption. The table records the per-epoch running time of model training in seconds.

Dataset		Time		Time
Events2012_100	UCL$_{SED}$	122.28	UCL-EC$_{SED}$	135.43
Events2018_100		109.23		111.77
CrisisLexT_7		3.31		3.35

example, on ACC, the result of UCL$_{SED}$ has a 10.28% drop. We argue that uncertainty is important in the representation adjustment and multi-view classifier modules. We use per-class uncertainty in the representation module to set a margin for each class. In the classifier module, we utilize the view-specific uncertainty of each sample to make the multi-view combination and obtain the final result. Thus, we need the estimated uncertainty to be as accurate as possible, which makes the uncertainty calibration method indispensable.

5.7 Time consumption

We record the time consumption information of UCL$_{SED}$ and UCL-EC$_{SED}$ in Table 5. As can be observed, overall, the per-epoch training time of UCL$_{SED}$ and UCL-EC$_{SED}$ is comparable, with UCL$_{SED}$ taking slightly less time than the latter. This is consistent with the time complexity analysis in Sec. 4.5. Compared to UCL$_{SED}$, in each epoch, UCL-EC$_{SED}$ needs extra calculation to update those prototypes.

6 Conclusion and future work

This paper proposes a novel uncertainty-guided class imbalance learning framework, namely UCL$_{SED}$, and its variant - UCL-EC$_{SED}$, for imbalanced SED tasks. As a label-dependent representation regularization technique, the UCL$_{SED}$ aims to improve the model generalization capability by enhancing representation learning for all classes, especially for those uncertain ones. Specifically, we design a novel uncertainty-guided contrastive learning loss that assigns larger margins for those more uncertain classes to manipulate separable representation boundaries. Meanwhile, we propose a multi-view combination architecture with an additional calibration method to ensure accurate and robust uncertainty estimation. The final detection result is combined via Dempster-Shafer theory under the supervision of uncertainty calibration. Experimental results verify the superiority of our model.

However, there are also some limitations. Both UCL$_{SED}$ and UCL-EC$_{SED}$ learn only one single prototype for each class, which makes them insufficient to handle complicated classes that follow a multimodal distribution. We leave the extension to multiple prototypes as future work.

Acknowledgement

This work is supported by National Key R&D Program of China through grant 2022YFB3104700, NSFC through grants 62322202, U21B2027, 62002007, 61972186 and 62266028, Yunnan Provincial Major Science and Technology Special Plan Projects through grants 202302AD080003, 202202AD080003 and 202303AP140008, General Projects of Basic Research in Yunnan Province through grant 202301AS070047, 202301AT070471, Beijing Natural Science Foundation through grant 4222030, S&T Program of Hebei through grants 20310101D and 21340301D, and the Fundamental Research Funds for the Central Universities. Jia Wu was supported by the Australian Research Council (ARC) Project DP230100899. Philip S. Yu was supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

References

[1] M. Fedoryszak, B. Frederick, V. Rajaram, and C. Zhong, "Real-time event detection on social data streams," in *ACM SIGKDD*, 2019, pp. 2774–2782.

[2] A. Q. Macedo, L. B. Marinho, and R. L. Santos, "Context-aware event recommendation in event-based social networks," in *RecSys*, 2015, pp. 123–130.

[3] K. Li, W. Lu, S. Bhagat, L. V. Lakshmanan, and C. Yu, "On social event organization," in *ACM SIGKDD*, 2014, pp. 1206–1215.

[4] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," *arXiv preprint arXiv:1608.06048*, pp. 1–7, 2016.

[5] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks*, vol. 106, pp. 249–259, 2018.

[6] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen *et al.*, "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *MIPR*. IEEE, 2018, pp. 112–117.

[7] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9268–9277.

[8] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," *NeurIPS*, vol. 30, pp. 7030–7040, 2017.

[9] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *ICML*. PMLR, 2018, pp. 4334–4343.

[10] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, "Equalization loss for long-tailed object recognition," in *CVPR*, 2020, pp. 11 662–11 671.

[11] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *ICLR*, 2019, pp. 1–11.

[12] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *CVPR*, 2020, pp. 9719–9728.

[13] H.-J. Ye, H.-Y. Chen, D.-C. Zhan, and W.-L. Chao, "Identifying and compensating for feature deviation in imbalanced deep learning," *arXiv preprint arXiv:2001.01385*, pp. 1–15, 2020.

[14] W. Jitkrittum, A. K. Menon, A. S. Rawat, and S. Kumar, "Elm: Embedding and logit margins for long-tail learning," *arXiv preprint arXiv:2204.13208*, pp. 1–24, 2022.

[15] B. Kang, Y. Li, S. Xie, Z. Yuan, and J. Feng, "Exploring balanced feature spaces for representation learning," in *ICLR*, 2020, pp. 1–15.

[16] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*. PMLR, 2020, pp. 9929–9939.

[17] P. Wang, K. Han, X.-S. Wei, L. Zhang, and L. Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *CVPR*, 2021, pp. 943–952.

[18] T. Li, P. Cao, Y. Yuan, L. Fan, Y. Yang, R. S. Feris, P. Indyk, and D. Katabi, "Targeted supervised contrastive learning for long-tailed recognition," in *CVPR*, 2022, pp. 6918–6928.

[19] J. Zhu, Z. Wang, J. Chen, Y.-P. P. Chen, and Y.-G. Jiang, "Balanced contrastive learning for long-tailed visual recognition," in *CVPR*, 2022, pp. 6908–6917.

[20] D. Samuel and G. Chechik, "Distributional robustness loss for long-tail learning," in *ICCV*, 2021, pp. 9495–9504.

[21] J. Ren, L. Jiang, H. Peng, Z. Liu, J. Wu, and Y. Philip S., "Evidential temporal-aware graph-based social event detection via dempster-shafer theory," in *ICWS*. IEEE, 2022, pp. 331–336.

[22] K. Sentz and S. Ferson, "Combination of evidence in dempster-shafer theory," pp. 1–96, 2002.

[23] H. Peng, R. Zhang, S. Li, Y. Cao, S. Pan, and P. S. Yu, "Reinforced, incremental and cross-lingual event detection from social messages," *TPAMI*, pp. 980–998, 2022.

[24] W. Cui, J. Du, D. Wang, F. Kou, and Z. Xue, "Mvgan: Multi-view graph attention network for social event detection," *TIST*, vol. 12, no. 3, pp. 1–24, 2021.

[25] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing twitter and traditional media using topic models," in *ECIR*. Springer, 2011, pp. 338–349.

[26] X. Yan, J. Guo, Y. Lan, J. Xu, and X. Cheng, "A probabilistic model for bursty topic discovery in microblogs," in *AAAI*, 2015, pp. 353–359.

[27] H. Amiri and H. D. III, "Short text representation for detecting churn in microblogs," in *AAAI*, 2016, pp. 2566–2572.

[28] Z. Wang and Y. Zhang, "A neural model for joint event detection and summarization," in *IJCAI*, 2017, pp. 4158–4164.

[29] S. Sahnoun, S. Elloumi, and S. Ben Yahia, "Event detection based on open information extraction and ontology," *Journal of Information and Telecommunication*, vol. 4, no. 3, pp. 383–403, 2020.

[30] K. Morabia, N. L. B. Murthy, A. Malapati, and S. Samant, "Sedtwik: Segmentation-based event detection from tweets using wikipedia," in *NAACL: Student Research Workshop*, 2019, pp. 77–85.

[31] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang, "Topicsketch: Real-time bursty topic detection from twitter," *IEEE TKDE*, vol. 28, no. 8, pp. 2216–2229, 2016.

[32] C. Xing, Y. Wang, J. Liu, Y. Huang, and W.-Y. Ma, "Hashtag-based sub-event discovery using mutually generative lda in twitter," in *AAAI*, vol. 30, no. 1, 2016, pp. 2666–2672.

[33] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang, "Streamcube: Hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream," in *IEEE ICDE*. IEEE, 2015, pp. 1561–1572.

[34] X. Zhou and L. Chen, "Event detection over twitter social media streams," *The VLDB journal*, vol. 23, no. 3, pp. 381–400, 2014.

[35] Y. Liu, H. Peng, J. Li, Y. Song, and X. Li, "Event detection and evolution in multi-lingual social streams," *Frontiers of Computer Science*, vol. 14, no. 5, pp. 1–15, 2020.

[36] Y. Wang, J. Liu, Y. Huang, and X. Feng, "Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs," *IEEE TKDE*, vol. 28, no. 7, pp. 1919–1933, 2016.

[37] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," in *IJCAI*, 2019, pp. 3238–3245.

[38] Y. Cao, H. Peng, J. Wu, Y. Dou, J. Li, and P. S. Yu, "Knowledge-preserving incremental social event detection via heterogeneous gnns," in *WWW*, 2021, pp. 3383–3395.

[39] H. Peng, J. Li, Y. Song, R. Yang, R. Ranjan, P. S. Yu, and L. He, "Streaming social event detection and evolution discovery in heterogeneous information networks," *TKDD*, vol. 15, no. 5, pp. 1–33, 2021.

[40] A. K. Pradhan, H. Mohanty, and R. P. Lal, "Event detection and aspects in twitter: A bow approach," in *ICDCIT*. Springer, 2019, pp. 194–211.

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ICLR*, pp. 1–12, 2013.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL*, pp. 4171–4186, 2018.

[43] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[44] A. Guille and C. Favre, "Mention-anomaly-based event detection and tracking in twitter," in *ASONAM*. IEEE, 2014, pp. 375–382.

[45] A. Guille and F. Cecile, "Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach," *Social Network Analysis and Mining*, vol. 5, no. 1, pp. 1–18, 2015.

[46] X. Chen, X. Zhou, T. Sellis, and X. Li, "Social event detection with retweeting behavior correlation," *Expert Systems with Applications*, vol. 114, pp. 516–523, 2018.

[47] J. Ren, L. Jiang, H. Peng, Y. Cao, J. Wu, P. S. Yu, and L. He, "From known to unknown: Quality-aware self-improving graph neural network for open set social event detection," in *CIKM*, 2022, pp. 1696–1705.

[48] J. Ren, H. Peng, L. Jiang, J. Wu, Y. Tong, L. Wang, X. Bai, B. Wang, and Q. Yang, "Transferring knowledge distillation for multilingual social event detection," *arXiv preprint arXiv:2108.03084*, pp. 1–18, 2021.

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

[50] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *ICIC*. Springer, 2005, pp. 878–887.

[51] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, N. Joshi, M. Meister, and P. Perona, "Synthetic examples improve generalization for rare classes," in *WACV*, 2020, pp. 863–873.

[52] J. Kim, J. Jeong, and J. Shin, "M2m: Imbalanced classification via major-to-minor translation," in *CVPR*, 2020, pp. 13 896–13 905.

[53] C. Drummond and R. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling," *ICML Workshop on Learning from Imbalanced Datasets*, pp. 1–8, 01 2003.

[54] S. Ryou, S.-G. Jeong, and P. Perona, "Anchor loss: Modulating loss scale based on prediction difficulty," in *ICCV*, 2019, pp. 5992–6001.

[55] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *ICLR*, 2020, pp. 1–13.

[56] S. Khan, M. Hayat, S. W. Zamir, J. Shen, and L. Shao, "Striking the right balance with uncertainty," in *CVPR*, 2019, pp. 103–112.

[57] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *NeurIPS*, vol. 32, pp. 1567–1578, 2019.

[58] H. He and E. A. Garcia, "Learning from imbalanced data," *TKDE*, vol. 21, no. 9, pp. 1263–1284, 2009.

[59] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," *NeurIPS*, vol. 33, pp. 19 290–19 301, 2020.

[60] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[61] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *NeurIPS*, vol. 33, pp. 18 661–18 673, 2020.

[62] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017, pp. 1–14.

[63] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, vol. 30, pp. 1025–1035, 2017.

[64] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *NeurIPS*, vol. 31, pp. 3183–3193, 2018.

[65] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," in *Classic works of the Dempster-Shafer theory of belief functions*. Springer, 2008, pp. 57–72.

[66] W. Bao, Q. Yu, and Y. Kong, "Evidential deep learning for open set action recognition," in *ICCV*, 2021, pp. 13 349–13 358.

[67] J. Mukhoti and Y. Gal, "Evaluating bayesian deep learning methods for semantic segmentation," *arXiv preprint arXiv:1811.12709*, pp. 1–13, 2018.

[68] R. Krishnan and O. Tickoo, "Improving model calibration with accuracy versus uncertainty optimization," *NeurIPS*, vol. 33, pp. 18 237–18 248, 2020.

[69] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on twitter," in *KMIS*, 2013, pp. 409–418.

[70] B. Mazoyer, J. Cagé, N. Hervé, and C. Hudelot, "A french corpus for event detection on twitter," in *LREC*, 2020, pp. 6220–6227.

[71] A. Olteanu, S. Vieweg, and C. Castillo, "What to expect when the unexpected happens: Social media communications across crises," in *CSCW*, 2015, pp. 994–1009.

Jiaqian Ren is currently a Ph.D. candidate in Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include social event mining and graph representation learning.

Hao Peng is currently an Associate Professor at the School of Cyber Science and Technology in Beihang University. His current research interests include data mining, machine learning, and deep learning. To date, Dr Peng has published over 100+ research papers in top-tier journals and conferences, including the IEEE TPAMI, TKDE, TC, ACM TOIS, TKDD, and Web Conference. He is the Associate Editor of International Journal of Machine Learning and Cybernetics (IJMLC).

Jia Wu is currently the Research Director for the AI-enabled Processes (AIP) Research Centre and an ARC DECRA Fellow in the School of Computing, Macquarie University, Sydney, Australia. His current research interests include data mining and machine learning. Since 2009, he has published 100+ referred journal and conference papers, including TPAMI, TKDE, TNNLS, TMM, TKDD, NIPS, WWW, and KDD. Dr. Wu was the recipient of SDM'18 Best Paper Award in Data Science Track, IJCNN'17 Best Student Paper Award, and ICDM'14 Best Paper Candidate Award. He is the Associate Editor of the ACM Transactions on Knowledge Discovery from Data (TKDD) and Neural Networks (NN).

Zhengtao Yu received the Ph.D. degree in computer application technology from the Beijing Institute of Technology, Beijing, China, in 2005. He is currently a Professor with the School of Information Engineering and Automation, Kunming University of Science and Technology, China. His current research interests include natural language process, image processing, and machine learning.

Lei Jiang is an associate professor in Institute of Information Engineering, Chinese Academy of Sciences. His current research interest include network security and social computing.

Zhiwei Liu is currently a Research Scientist at Salesforce AI Research. His research interests include graph representation learning, recommender system and natural language understanding. He has published over 30 original research works in top-tier journals and conferences, including ACM TIST, Web Conference, SIGIR, CIKM, WSDM, and EMNLP.

Philip S. Yu is a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois at Chicago. Before joining UIC, he was at the IBM Watson Research Center, where he built a world-renowned data mining and database department. He is a Fellow of the ACM and IEEE. Dr. Yu has published more than 1,200 referred conference and journal papers cited more than 179,000 times with an H-index of 189. Dr. Yu was the Editor-in-Chiefs of ACM Transactions on Knowledge Discovery from Data (2011-2017) and IEEE Transactions on Knowledge and Data Engineering (2001-2004).