# A ROBUST JOINT-TRAINING GRAPH NEURAL NETWORKS MODEL FOR EVENT DETECTION WITH SYMMETRY AND ASYMMETRY NOISY LABELS

Mingxiang Li[1], Huange Xing[1*], Tengyun Wang[2],

Jiaxuan Dai[1], and Kaiming Xiao[2]

[1]Naval University of Engineering, Wuhan, China
[2]National University of Defense Technology, Changsha, China

***ABSTRACT***

*Events are the core element of information in descriptive corpus. Although many progresses have been made in Event Detection (ED), it is still a challenge in Natural Language Processing (NLP) to detect event information from data with unavoidable noisy labels. A robust Joint-training Graph Convolution Networks (JT-GCN) model is proposed to meet the challenge of ED tasks with noisy labels in this paper. Specifically, we first employ two Graph Convolution Networks with Edge Enhancement (EE-GCN) to make predictions simultaneously. A joint loss combining the detection loss and the contrast loss from two networks is then calculated for training. Meanwhile, a small-loss selection mechanism is introduced to mitigate the impact of mislabeled samples in networks training process. These two networks gradually reach an agreement on the ED tasks as joint-training progresses. Corrupted data with label noise are generated from the benchmark dataset ACE2005. Experiments on ED tasks has been conducted with both symmetry and asymmetry label noise on different level. The experimental results show that the proposed model is robust to the impact of label noise and superior to the state-of-the-art models for ED tasks.*

***KEYWORDS***

*Event Detection, Graph Convolution Networks, Noisy Label, Robustness, Small-loss Selection*

## 1. INTRODUCTION

As one of the core elements of multi-heterogeneous text data, events play a role that cannot be ignored in detection, extraction, and utilization of large-scale open-source knowledge. In order to promote the application of artificial intelligence in the field of open-source information detection, it is urgent to automate and intelligently process relevant events in various corpus [1], e.g., reports, News, and social media texts, etc. Event Detection (ED), an information extraction process to detect trigger words and recognize event types from plain text, plays a crucial role in Natural Language Processing (NLP) [2-4]. Through ED, valuable structured information can be obtained to facilitate various tasks such as automatic text summarization [5], question answering [6], and information retrieval [7], etc.

Although lots of studies on event detection have been proposed [8-11], most of them focus on ED tasks with clean and accurate labels which ignores the challenge of unavoidable noisy labels. As is known, collecting and labeling of large-scale datasets with fully precise annotations is expensive and time-consuming [12]. Manual annotation usually suffers from unavoidable noisy labels, especially for tasks that requires natural language understanding. Moreover, previous research on deep learning has shown the negative impact of label noise on the performance of learning models [13-15]. Unfortunately, few studies have paid attention to robustness of ED

tasks to noisy labels. Without consideration of label noise, previous ED models are vulnerable to data label contamination leading to degraded performance.

In this paper, we proposed a robust Joint-training Graph Convolution Networks (JT-GCN) model to meet the challenge of ED tasks with noisy labels. The model is based on two Graph Convolution Networks (GCN) with edge enhancement [16]. Motivated by previous co-training and joint-training frameworks for Deep Neural Networks (DNN) [12,17], a joint loss is then calculated by two networks' predictions of the same mini-batch data, in which both detection loss and contrast loss between two GCNs are considered. Besides, a small-loss selection mechanism is integrated to the joint-training process so as to mitigate the impact of mislabeled samples in networks. These two networks in JT-GCN gradually reach an agreement on the ED tasks as joint-training progresses. The primary contributions of this work can be summarized in the following points:

1) A novel GCN-based model is proposed to address the challenge of ED tasks with noisy labels.

2) A joint-training framework integrated with small-loss selection mechanism is revised and applied to ED tasks.

3) The performance of proposed model is validated on corrupted data with both symmetry and asymmetry label noise on different level generated from the benchmark dataset ACE2005.

The rest of this paper is structured as follows. Section 2 provides a brief review of related works on event detection, while Section 3 gives preliminaries of key notions related to ED. In Section 4, a robust joint-training graph convolution networks model is proposed for ED tasks with noisy labels. Experimental results of the proposed framework on the original and corrupted ED benchmark dataset ACE2005 are shown and discussed in Section 5. Conclusions are lastly offered in Section 6.

## 2. RELATED WORK

### 2.1. Event Detection

In late 1980s, the concept of event extraction was first proposed in the Message Understanding Conference (MUC) [18]. Following the MUC, the International Evaluation Conference on Automatic Content Extraction (ACE) [19] further promoted the development of ED technology which is then widely used in finance, medical care, law, social media and other fields.

Nguyen and Grishman [20] proposed an event detection model using Convolution Neural Networks (CNN) in 2015, which is applicable to cases with one-word trigger words and event elements. Further research has been conducted by introducing Recurrent Neural Networks (RNN) to the joint extraction process of trigger words and event elements [21], and it achieves better extraction results than the CNN model. Liu et al. [22] exploited argument information explicitly for event detection via supervised attention mechanisms, and investigated the impact of different attention strategies. Recently, more attentions are paid to the utilization of Graph Neural Networks (GNN) in NLP tasks and brought encouraging performance improvements [2,23]. By converting text sequences into graph-structured data to incorporate rich semantic information, Nguyen and Grishman [2] first leveraged GCN to conduct ED tasks and achieved remarkable model performance. The state-of-the-art (SOTA) GCN-based model for ED was proposed by Cui et al., in which node updating and edge updating modules were introduced to learn the embedding vectors for the edges in the syntactic dependency graph [16].

### 2.2. Robust Deep Learning

The robustness of deep learning models is a key aspect of artificial intelligence. Plenty of studies have shown the vulnerability of deep learning models towards data with noisy labels [24], adversary poisoning [25], and imbalance sampling [26]. Especially, various of approaches have been proposed to address the challenge of robust learning with noisy labels in image classification tasks [24,27], including specific robust designs on regularization [28], loss function [29], sample selection [30], and architecture [31].

In the field of NLP, the research on model robustness has also gradually received attention. Different from adversarial samples generation research for computer vision applications, Papernot et al. [32] first studied how to design adversarial text sequences for RNN processing sequential data, and inspired subsequent explorations of robust deep learning in NLP. To overcome the weakness of ED models in learning generalization knowledge, a Delta-learning approach was proposed to distill discrimination and generalization knowledge by effectively decoupling, incrementally learning and adaptively fusing event representation [33]. Liu et al. [34] proposed a training paradigm called context-selective mask generalization so as to improve the ED model robustness against adversarial attacks. Unfortunately, little attention has been paid to the challenge of robust learning with noisy labels in ED or more general NLP tasks, while manual annotation usually suffers more from unavoidable noisy labels due to the complexity of natural language understanding.

## 3. PRELIMINARIES

We first clarify some concepts and notions related to ED tasks.

**Related Concepts to ED:** 1) Entity: one or a group of semantically classified objects, including people, organizations, places, time, etc. 2) Event Mention: a phrase or sentence describing an event that contains event trigger words and their corresponding arguments. 3) Event Trigger: the trigger word is the core of event identification, usually a verb or a noun. 4) Event Role: an event role corresponds to a predefined field of the event table. 5) Event Argument: an event argument is an entity that plays a predefined event role, usually referring to the time, place, and participants of the event.

**ED Tasks with Noisy Labels:** To illustrate the ED tasks with noisy labels, we give two examples as shown in Figure 1. In corpus sample 1 (S1), an event detector aims to figure out the event trigger ***fired*** and recognize the event type ***EndPosition*** in this context. As an event mention, S1 includes three arguments, i.e., two participants (*the airline*, *pilot*) and the cause (*fault in work*). When there are label noise in data, as shown in S2, the event type is incorrectly identified as ***Attack***, not as true type ***Accident*** due to uncertainty of manual annotation. The existing of noisy labels in dataset has been proven to be baneful to deep learning [12].

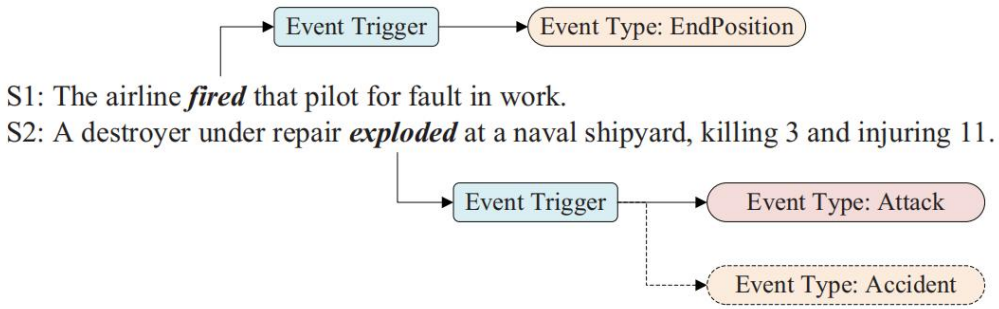


Figure 1. Descriptive examples of event detection with noisy labels

# 4. THE PROPOSED MODEL

In this section, we propose the JT-GCN model to meet the challenge of ED tasks with noisy labels. The core of JT-GCN model lies in three aspects: 1) the construction of simultaneous learning framework for two networks; 2) the design of join loss and joint training procedure as well; 3) the implementation of small-loss selection for noise effect mitigation.

The descriptive framework of JT-GCN model is shown in Figure 2. In each iteration of training, a mini-batch of data with noisy labels is first passed into two networks, i.e., EE-GCNs with $\Theta_1$ and $\Theta_2$ for forward propagation simultaneously. Two predictions $p_1$ and $p_2$ are then obtained separately. In order to make the prediction results of the two networks eventually converge to an agreement, a joint-training procedure is introduced by designing joint loss of two predictions. Furthermore, a small-loss selection mechanism is integrated to the joint-training process in which instances with large loss are considered high-probability mislabeled and excluded from the back propagation process. In this way, the joint-training procedure mitigates the impact of mislabeled samples on the way to consensus of two networks.

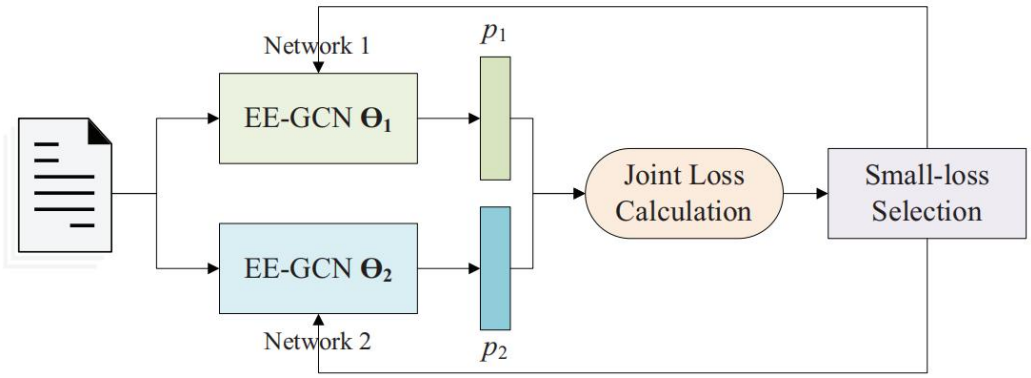


Figure 2. The schematic of JT-GCN model

## 4.1. Network Construction

The proposed JT-GCN model is based on two Graph Convolution Networks (GCN) with edge enhancement [16]. In each EE-GCN, there are three components to construct the ED architecture, i.e., *the input layer, the GCN layer, and the classification layer*.

**Input Layer:**For a given input sequence $S = (\omega_1, \omega_2, \ldots, \omega_n)$, each $\omega_i$ is vectorized to $x_i = [\omega_i, e_i] \in \boldsymbol{R}^{(d_\omega + d_e)}$, where $\omega_i$ and $e_i$ denote the word embedding vector and entity type embedding vector with $d_\omega$ and $d_e$ dimensions respectively. Then, a BiLSTM layer is applied to obtain the contextual information of each words based on the embedding vector $[x_1, x_2, \ldots, x_n] \in \boldsymbol{R}^{n \times (d_\omega + d_e)}$, resulting contextualized word representations. To utilize the syntactic dependency parsing in each sequence $S$, a syntactic dependency graph in form of adjacency matrix is generated by taking words as nodes and dependencies as edges. To illustrate, the syntactic dependency parsing of sample S1 is given in Figure 3.

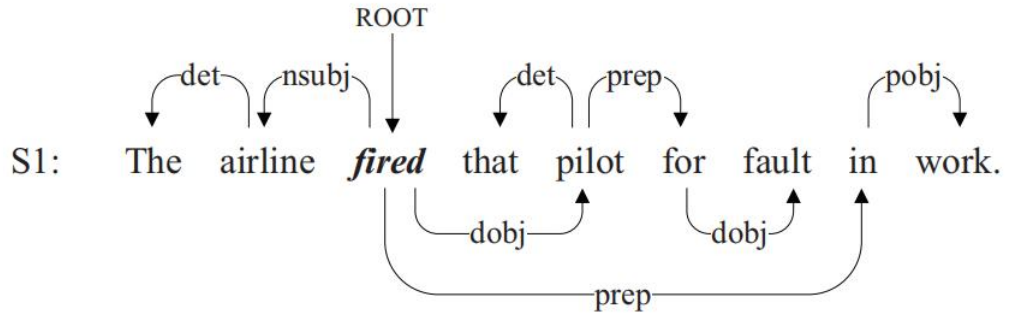


Figure 3. The syntactic dependency parsing of sample S1

**GCN Layer:**In order to utilizing the useful linguistic knowledge implicit in dependency parsing information, an edge representation tensor $\boldsymbol{E} = [e_{i,j,k}] \in \boldsymbol{R}^{n\times n\times p}$ is introduced by Cui et al.[16], where $e_{i,j} \in \boldsymbol{R}^{p}$ is the vector representation of the corresponding edge in syntactic dependency graph. Denote by $\boldsymbol{H} = (h_1, h_2, \ldots, h_n) \in \boldsymbol{R}^{n\times d_l}$ the node representation tensor, where $d_g$ is the dimension of each node's (word's) representation. Then two modules are adopted at each layer $l$ of EE-GCN to update $\boldsymbol{H}$ and $\boldsymbol{E}$ mutually through information aggregation:

$$(\boldsymbol{H}^l, \boldsymbol{E}^l) = EE - GCN(\boldsymbol{H}^{l-1}, \boldsymbol{E}^{l-1}). \tag{1}$$

Denote by Pool the mean-pooling operation to compress information from all channels, and σ the ReLU activation function. By aggregating the information from its neighbors nodes through adjacent tensor, the update operation for $\boldsymbol{H}^l$ at each layer $l \in [1, L]$ is:

$$\boldsymbol{H}^l = \sigma(Pool(\boldsymbol{H}_1^l, \boldsymbol{H}_2^l, \ldots, \boldsymbol{H}_p^l)). \tag{2}$$

Specifically, for each channel $k \in [1, p]$ the aggregation of $\boldsymbol{H}_k^l$ is:

$$\boldsymbol{H}_k^l = \boldsymbol{E}_{:,:,k}^{l-1}\boldsymbol{H}_k^{l-1}\boldsymbol{W}_N, k \in [1, p], \tag{3}$$

where $\boldsymbol{W}_N \in \boldsymbol{R}^{d_g\times d_g}$ is the parameter to be learned. According to the node context, the edge representation of each edge in layer $l$ can be updated as follows:

$$e_{i,j}^l = \boldsymbol{W}_E[e_{i,j}^{l-1} \oplus h_i^l \oplus h_j^l], i, j \in [1, n], \tag{4}$$

where $\boldsymbol{W}_E \in \boldsymbol{R}^{(2*d_g+p)\times p}$ denotes a learnable transformation matrix and $\oplus$ represents the join operation.

**Classification Layer:**After obtaining the final representation of each word (node) $h_i^L$, a fully-connected network with softmax function is then adopted to compute probability distribution over all event types ($t \in \mathcal{T}$) as follows:

$$p(t|h_i^L) = softmax(\boldsymbol{W}_C h_i^L + b_C), \tag{5}$$

where $\boldsymbol{W}_C$ and $b_C$ are a learnable mapping matrix and a bias term. Since there are two EE-GCNs, we can obtain two classification results $\boldsymbol{P}_1$ and $\boldsymbol{P}_2 \in [0,1]^{N_s\times n_i}$ at each mini-batch training process from EE-GCN1 with parameters $\Theta_1$ and EE-GCN2 with $\Theta_2$. Event label with the largest probability is then selected as the final classification result $p_1$ and $p_2$.

## 4.2. Joint Loss Design

For each network $k$, the bias loss function is used to enhance the influence of event labels during training:

$$l_k(\theta) = -\sum_{i=1}^{\boldsymbol{N_s}} \sum_{j=1}^{n_i} log\ p_k\left(y_j^t \middle| s_i, \theta\right) \cdot I(O) + \alpha\ log\ p_k\left(y_j^t \middle| s_i, \theta\right) \cdot (1 - I(O)), \forall k = 1,2. \quad (6)$$

where $\boldsymbol{N_s}$ is the number of sentences, $n_i$ is the number of words in sentence $s_i$ and $y_j^t$ is the ground-truth word's label of event $t$; $I(O)$ equals 1 if the event type of the word is `O', otherwise 0; $\alpha$ is the bias weight large than 1.

Motivated by previous co-training and joint-training frameworks for Deep Neural Networks (DNN) [12,17], a joint loss is then calculated by two networks' predictions of the same mini-batch data, in which both detection loss and contrast loss between two EE-GCNs are considered as follows:

$$L(\theta) = (1 - \lambda)L_d(\theta) + \lambda L_C(\theta), \quad (7)$$

where $\lambda \in [0,1]$ is a parameter of co-regularization. Denote by $L_d$ the conventional detection loss:

$$\mathrm{L_d}(\theta) = \mathrm{l_1}(\theta) + \mathrm{l_2}(\theta). \quad (8)$$

$L_C$ denotes the contrast loss between two EE-GCNs.We used the Kullback-Leibler (KL) Divergence to measure the difference between $p_1$ and $p_2$:

$$\mathrm{L_c}(\theta) = \mathrm{D_{BKL}}(\mathrm{p_1} \parallel \mathrm{p_2}) + \mathrm{D_{BKL}}(\mathrm{p_2} \parallel \mathrm{p_1}), \quad (9)$$

where

$$D_{BKL}(p_1 \parallel p_2) = \sum_{i=1}^{N_s} \sum_{j=1}^{n_i} p_1\left(y_j^{t1} \middle| s_i, \theta\right)\ log \frac{p_1\left(y_j^{t1} \middle| s_i, \theta\right)}{p_2\left(y_j^{t2} \middle| s_i, \theta\right)}, \quad (10)$$

$$\mathrm{D_{BKL}}(\mathrm{p_2} \parallel \mathrm{p_1}) = \sum_{\mathrm{i}=1}^{\mathrm{N_s}} \sum_{\mathrm{j}=1}^{\mathrm{n_i}} \mathrm{p_2}\left(\mathrm{y_j^{t2}} \middle| \mathrm{s_i}, \theta\right)\ \log \frac{\mathrm{p_2}\left(\mathrm{y_j^{t2}} \middle| \mathrm{s_i}, \theta\right)}{\mathrm{p_1}\left(\mathrm{y_j^{t1}} \middle| \mathrm{s_i}, \theta\right)}, \quad (11)$$

## 4.3. Small-loss Selection

Besides, a small-loss selection mechanism is integrated to the joint-training process so as to mitigate the impact of mislabeled samples in networks. As we know that two EE-GCNs have different learning abilities based on different initial conditions. If the joint loss of a sample is small, it is more likely that this sample has a true label [12,33]. Hence, the small-loss selection mechanism is introduced as follows:

$$\tilde{S}_n = arg \min_{S_n^{'}} L(S_n^{'})\ s.t.\ |S_n^{'}| \geq R(T)|S_n|, \quad (12)$$

$$R(T) = 1 - min\left\{\frac{T}{T_K}\tau, \tau\right\}. \quad (13)$$

where $S_n$ and $\tilde{S}_n$ denote the initial mini-batch training set and the selected set. $R(T)$ is a the ratio of small-loss samples at iteration $T$, and $\tau \in [0,1)$ is a parameter depending on the rate of noisy labels. Iteration parameter $T_K$ is set to decide the speed at which $R(t)$ drops from 1 to $\tau$ during the training process.

## 5. EXPERIMENTS

In this section, we test and discuss the performance of proposed JT-GCN model on the original and corrupted ED benchmark dataset ACE2005.

### 5.1. Experimental Settings

**Dataset:**We use ACE2005 as the clean dataset which is a standard supervised dataset for ED. The toolkit CoreNLP[1] is used for dependency parsing. The corrupted datasets are manually transformed by the label transition matrix $Q$, where $Q_{i,j} = Pr(\tilde{y} = j|y = i)$ given that noisy $\tilde{y}$ is flipped from clean $y$. Two representative structures of $Q$ are used in this paper: 1) Symmetry flipping [36]; 2)Asymmetry flipping [37] where labeling-mistakes are only made within very similar classes. Illustrative examples of label transition matrix are shown in Figure 4.

Symmetric Noise 0.3

| | | | |
|---|---|---|---|
| 70% | 10% | 10% | 10% |
| 10% | 70% | 10% | 10% |
| 10% | 10% | 70% | 10% |
| 10% | 10% | 10% | 70% |

Asymmetric Noise 0.3

| | | | |
|---|---|---|---|
| 70% | 0% | 0% | 30% |
| 0% | 100% | 0% | 0% |
| 0% | 30% | 70% | 0% |
| 0% | 0% | 0% | 100% |

Figure 4. Examples of label transition matrix $Q$ (taking 4 classes and noise ratio 0.3 as an example

**Measurement:** Precision $(P)$, Recall $(R)$, $F_1 - \text{score}$ $(F_1)$ are used as metrics to evaluate the performance of ED models.

**Baselines:** We compare JT-GCN with the state-of-the-art models MOGANED [3] and EE-GCN[16].

**Hyper-parameters:** Hyper-parameters used in experiments of JT-GCN are listed in Table 1. All codes are implemented by PyTorch with default parameters on NVIDIA GeForce GTX 1050Ti.

### 5.2. Results Discussion

We test the proposed JT-GCN and existing EE-GCN model on ACE2005 dataset so as to evaluate their robustness performance to label noise in training and validating dataset.

Table 1. Hyper-parameters of JT-GCN.

| **Hyper-parameters** | **Values** |
|---|---|
| Dimension of word vectors ($d_\omega$) | 100 |
| Dimension of entity types vectors ($d_e$) | 50 |

| | |
|---|---|
| Dimension of edge labels vectors ($p$) | 50 |
| Dimension of Bi-LSTM ($d_l/2$) | 100 |
| Dimension of GCN ($d_g$) | 150 |
| Layers of GCN ($L$) | 2 |
| Learning rate | 0.001 |
| Optimizer | Adam |
| Bias weight of loss function ($\alpha$) | 5 |
| Batch size | 30 |
| Epoch | 100 |
| Maximum text length | 50 |
| Parameter of co-regularization ($\lambda$) | 0.5 |
| Iteration parameter ($T_K$) | 10 |

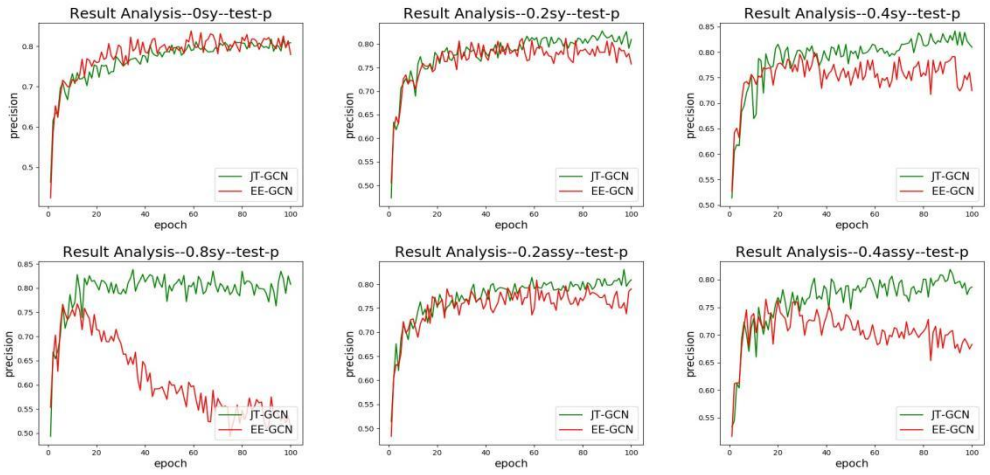


Figure 5. Results on ACE2005 dataset with different label noise ratio. $P$(%) vs. Epochs

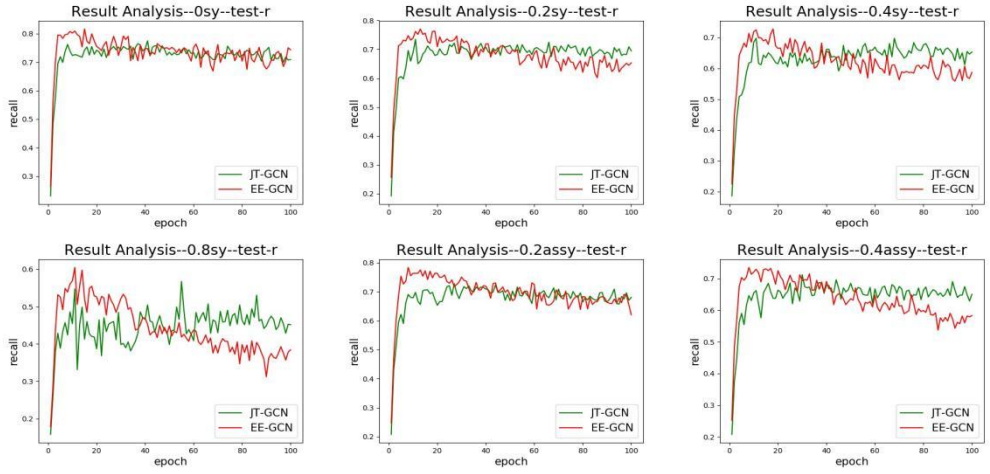


Figure 6. Results on ACE2005 dataset with different label noise ratio. $R$(%) vs. Epochs

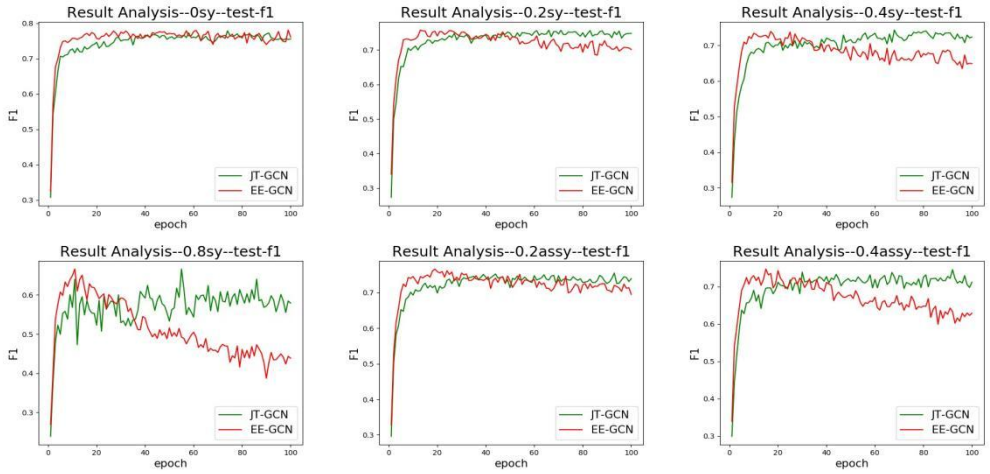


Figure 7. Results on ACE2005 dataset with different label noise ratio. $F_1(\%)$ vs. Epochs

As show in Fig. 5, Fig. 6 and Fig. 7, the label noise ratio of Symmetry flipping is set from 0% to 80% and the label noise ratio of Asymmetry flipping is set from 20% to 40%. The superiority of JT-GCN continues to expand as the level of label contamination increases. The test $F_1$ under EE-GCN model first reaches a high level and then gradually decreases and overfits to mislabeled data. With the label noise ratio improves, the downward tendency is more obvious. Compared with symmetry flipping and asymmetry flipping, the model performs poorly on asymmetric flipping. With the increase of noise ratio, the performance difference between JT-GCN model and EE-GCN model gradually increases. JT-GCN is less likely to overfit to mislabeled data as the increase of training epoch and keeps a stable state. The model can keep relatively stable performance under the same noise ratio regardless of symmetry or asymmetry flipping.

The model shows a robust performance that stops or alleviates the decreasing process. Moreover, we compare the average test performance of both models over the last 10 epochs shown in Table 2. The test performance of EE-GCN is slightly higher than that of JT-GCN when there is no label noise in training and validating dataset. However, it is clear that JT-GCN model dominates EE-GCN model in all flipping rates (20%-80%), different label transition matrix and metrics ($P$, $R$, and $F_1$). When the label noise rate is set as 20% and label transition matrix is symmetry, JT-GCN reports 74.53% in $F_1$ which is higher than that of EE-GCN reporting 70.45%. When the noise rate is 80% (i.e., 80% labels are wrong in training and validating data) and label transition matrix is symmetry, the performance of JT-GCN model outperforms EE-GCN model more than 14%. When the label noise rate is set as 20% and label transition matrix is asymmetry, JT-GCN reports 73.72% in $F_1$ which is higher than that of EE-GCN reporting 71.08%. When the noise rate is 40\% and label transition matrix is asymmetry, the performance of JT-GCN model outperforms EE-GCN model more than 8%. Under the same noise ratio and different transfer matrices, JT-GCN model and EE-GCN model have low difference of $F_1$.

Table 2. Average test performance (%) on ACE2005 over the last 10 epochs.

| Noise Ratio | EE-GCN | | | JT-GCN | | |
|---|---|---|---|---|---|---|
| | *P* | *R* | $F_1$ | *P* | *R* | $F_1$ |
| 0% | **80.56** | 71.72 | **75.85** | 80.21 | **71.78** | 75.74 |
| Symmetry-20% | 78.05 | 64.24 | 70.45 | **81.02** | **69.03** | **74.53** |
| Symmetry-40% | 75.39 | 57.98 | 65.51 | **82.52** | **64.41** | **72.33** |
| Symmetry-80% | 53.40 | 37.23 | 43.86 | **80.26** | **45.46** | **58.03** |
| Asymmetry-20% | 76.44 | 66.49 | 71.08 | **80.61** | **67.94** | **73.72** |
| Asymmetry-40% | 68.60 | 57.10 | 62.32 | **79.37** | **65.39** | **71.69** |

## 6. CONCLUSIONS

In this paper, we proposed a robust Joint-training Graph Convolution Networks (JT-GCN) model to meet the challenge of ED tasks with noisy labels. A joint-training framework integrated with small-loss selection mechanism is revised and applied to ED tasks. The performance of proposed model is validated on corrupted data with symmetry label noise on different level generated from the benchmark dataset ACE2005. The general robustness of GCN models for ED tasks considering more complex corrupted data (e.g., asymmetry label noise, data poisoning attacks) is a potential direction for further study.

## REFERENCES


[1] Ritter, A.; Etzioni, O.; Clark, S. Open domain event extraction from twitter. In Proceedings of the Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 1104-1112.

[2] Nguyen, T.; Grishman, R. Graph convolutional networks with argument-aware pooling for event detection. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2018, Vol. 32.

[3] Yan, H.; Jin, X.; Meng, X.; Guo, J.; Cheng, X. Event detection with multi-order graph convolution and aggregated attention. In Proceedings of the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 5766–5770.

[4] Veyseh, A.P.B.; Lai, V.; Dernoncourt, F.; Nguyen, T.H. Unleash GPT-2 power for event detection. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 6271-6282.

[5] Marujo, L.; Ribeiro, R.; Gershman, A.; de Matos, D.M.; Neto, J.P.; Carbonell, J. Event-based summarization using a centrality-as-relevance model. Knowledge and Information Systems 2017, 50, 945–968.

[6] Zhu, F.; Lei, W.; Wang, C.; Zheng, J.; Poria, S.; Chua, T.S. Retrieving and reading: A comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774 2021.

[7] Kaur, P.; Pannu, H.S.; Malhi, A.K. Comparative analysis on cross-modal information retrieval: A review. Computer Science Review 2021, 39, 100336.

[8] Ahn, D. The stages of event extraction. In Proceedings of the Proceedings of the Workshop on Annotating and Reasoning about Time and Events, 2006, pp. 1–8.

[9] Yang, B.; Mitchell, T. Joint extraction of events and entities within a document context. arXiv preprint arXiv:1609.03632 2016.

[10] Lou, D.; Liao, Z.; Deng, S.; Zhang, N.; Chen, H. MLBiNet: A cross-sentence collective event detection network. arXiv preprint arXiv:2105.09458 2021.

[11] Hanqing, Z.; Kaiming, X.; Lizhen, O.; Mao, W.; Lihua, L.; Hongbin, H. Attention-Based Graph Convolution Networks for Event Detection. In Proceedings of the 2021 7th International Conference on Big Data and Information Analytics (BigDIA). IEEE, 2021, pp. 185–190.

[12] Wei, H.; Feng, L.; Chen, X.; An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13726–13735.

[13] Jiang, L.; Zhou, Z.; Leung, T.; Li, L.J.; Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Proceedings of the International conference on machine learning. PMLR, 2018, pp. 2304-2313.

[14] Kim, Y.; Yim, J.; Yun, J.; Kim, J. Nlnl: Negative learning for noisy labels. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 101–110.

[15] Ding C, Pereira T, Xiao R, Lee RJ, Hu X. Impact of Label Noise on the Learning Based Models for a Binary Classification of Physiological Signal. Sensors (Basel). 2022 Sep 21;22(19):7166. doi: 10.3390/s22197166. PMID: 36236265; PMCID: PMC9572105.

[16] Cui, S.; Yu, B.; Liu, T.; Zhang, Z.; Wang, X.; Shi, J. Edge-enhanced graph convolution networks for event detection with syntactic relation. arXiv preprint arXiv:2002.10757 2020.

[17] Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4320–4328.

[18] Hirschman, L. The evolution of evaluation: Lessons from the message understanding conferences. Computer Speech & Language 1998, 12, 281 ‑ 305.

[19] Doddington, G.R.; Mitchell, A.; Przybocki, M.A.; Ramshaw, L.A.; Strassel, S.M.; Weischedel, R.M. The automatic content extraction (ace) program-tasks, data, and evaluation. In Proceedings of the Lrec. Lisbon, 2004, Vol. 2, pp. 837–840.

[20] Nguyen, T.H.; Grishman, R. Event detection and domain adaptation with convolutional neural networks. In Proceedings of the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2015, pp. 365-371.

[21] Nguyen, T.H.; Cho, K.; Grishman, R. Joint event extraction via recurrent neural networks. In Proceedings of the Proceedingsof the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 300–309.

[22] Liu, S.; Chen, Y.; Liu, K.; Zhao, J. Exploiting argument information to improve event detection via supervised attention mechanisms. In Proceedings of the Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1789–1798.

[23] Lai, V.D.; Nguyen, T.N.; Nguyen, T.H. Event detection: Gate diversity and syntactic importance scoresfor graph convolution neural networks. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); , 2020; pp. 5405-5411.

[24] Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.G. Learning from noisy labels with deep neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems 2022.

[25] Xu, G.; Li, H.; Ren, H.; Yang, K.; Deng, R.H. Data security issues in deep learning: attacks, countermeasures, and opportunities. IEEE Communications Magazine 2019, 57, 116 ‑ 122.

[26] Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. Journal of Big Data 2019, 6, 1–54.

[27] Karimi, D.; Dou, H.; Warfield, S.K.; Gholipour, A. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Medical Image Analysis 2020, 65, 101759.

[28] Hendrycks, D.; Lee, K.; Mazeika, M. Using pre-training can improve model robustness and uncertainty. In Proceedings of the International Conference on Machine Learning. PMLR, 2019, pp. 2712-2721.

[29] Ma, X.; Huang, H.; Wang, Y.; Romano, S.; Erfani, S.; Bailey, J. Normalized loss functions for deep learning with noisy labels. In Proceedings of the International conference on machine learning. PMLR, 2020, pp. 6543-6553.

[30] Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the International Conference on Machine Learning. PMLR, 2019, pp. 7164-7173.

[31] Han, B.; Yao, J.; Niu, G.; Zhou, M.; Tsang, I.; Zhang, Y.; Sugiyama, M. Masking: A new perspective of noisy supervision. Advances in neural information processing systems 2018, 31.

[32] Papernot, N.; McDaniel, P.; Swami, A.; et al. Crafting adversarial input sequences for recurrent neural networks. In Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference; , 2016; pp. 49-54.

[33] Lu, Y.; Lin, H.; Han, X.; et al. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In Proceedings of the Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; , 2019; pp. 4366-4376.

[34] Liu, J.; Chen, Y.; Liu, K.; Jia, Y.; Sheng, Z. How does context matter? On the robustness of event detection with context-selective mask generalization. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 2523-2532.

[35] Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems 2018, 31.

[36] van Rooyen, B.; Menon, A.; Williamson, R.C. Learning with Symmetric Label Noise: The Importance of Being Unhinged. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2015, Vol. 28.

[37] Patrini, G.; Rozza, A.; Menon, A.K.; Nock, R.; Qu, L. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, pp. 2233-2241.