applied sciences

MDPI

Article

A Class-Incremental Learning Method for Interactive Event Detection via Interaction, Contrast and Distillation

Jiashun Duan and Xin Zhang *

National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410073, China; duanjiashun18@nudt.edu.cn
* Correspondence: shinezhang_nudt@163.com

Abstract: Event detection is a crucial task in information extraction. Existing research primarily focuses on machine automatic detection tasks, which often perform poorly in certain practical applications. To address this, an interactive event-detection mode of "machine recommendation-human review–machine incremental learning" was proposed. In this mode, we study a few-shot continual class-incremental learning scenario, where the challenge is to learn new-class events with limited samples while preserving memory of old class events. To tackle these challenges, we propose a class-incremental learning method for interactive event detection via **I**nteraction, **C**ontrast and **D**istillation (ICD). We design a replay strategy based on representative and confusable samples to retain the most valuable samples under limited conditions; we introduce semantic-boundary-smoothness contrastive learning for effective learning of new-class events with few samples; and we employ hierarchical distillation to mitigate catastrophic forgetting. These methods complement each other and show strong performance. Experimental results demonstrate that, in the 5-shot 5-round class incremental-learning settings on two Chinese event-detection datasets ACE and DuEE, our method achieves final recall rates of 71.48% and 90.39%, respectively, improving by 6.86% and 3.90% over the best baseline methods.

Keywords: event detection; information extraction; human-in-the-loop; incremental learning; few-shot learning; contrastive learning

Citation: Duan, J.; Zhang, X. A Class-Incremental Learning Method for Interactive Event Detection via Interaction, Contrast and Distillation. *Appl. Sci.* **2024**, *14*, 8788. https://doi.org/10.3390/app14198788

Academic Editor: Fabrizio Marozzo

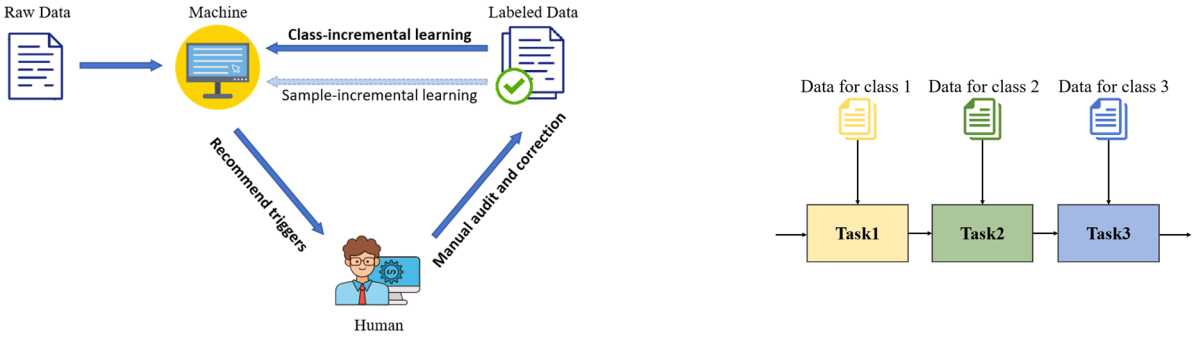
Received: 14 September 2024
Revised: 25 September 2024
Accepted: 27 September 2024
Published: 29 September 2024

1. Introduction

Contemporary research in event detection focuses primarily on machine-based automatic detection. However, in practical applications, many scenarios still require manual review and correction of machine-detected results to ensure high accuracy. To enhance the efficiency of manual review, our previous research [1] introduced the event-trigger recommendation task and developed a recommendation model. This model aims to provide multiple candidate triggers for potential events, targeting high recall to reduce the time spent on manual verification and correction. Nevertheless, as the real world evolves and areas of interest shift, new types of events or more complex situations may emerge, leading to misidentifications by the model and necessitating additional manual corrections. These instances highlight the model's weaknesses, and using them for incremental training can help improve the model's performance.

To address this, we propose a human-in-the-loop interactive event-detection mode: "Machine recommends event triggers→Manual review and correction→Machine incremental learning". As shown in Figure 1a, the process involves the following steps: first, the machine processes unlabeled data and recommends candidate triggers and their event types. Next, human reviewers audit and correct these recommendations to generate labeled data. Finally, the machine uses these labeled data for incremental learning to improve recommendation performance, creating a positive feedback loop. During incremental training, there are two main scenarios: class-incremental learning and sample-incremental learning.

Class-incremental learning refers to the model learning new event types, as illustrated in Figure 1b; sample-incremental learning involves the arrival of original class samples in batches. This paper focuses on the scenario of class-incremental learning.



(**a**) Interactive event-detection mode (**b**) Class-incremental learning

Figure 1. The concept of interactive event-detection mode and the class-incremental learning.

In practical applications, new events typically emerge gradually, and initially only have a few samples. To enable the model to identify new events early on, class-incremental learning must be performed with limited samples, which presents the challenge of avoiding overfitting while retaining previously learned classes. Furthermore, in traditional class-incremental learning [2–5], new training data are not obtained through manual review and correction, whereas in the interactive event-detection mode, new training data are derived from the model's incorrect predictions and corrected by humans, including the model's errors and manual correction feedback. Effectively learning from such feedback is a key issue to be addressed in interactive event-detection modes.

To address these issues, we propose a class-incremental learning method for interactive event-detection mode. To tackle the challenges of few-shot class-incremental learning, we introduce semantic-boundary smoothing-based contrastive learning, which assigns soft labels and designs contrastive loss to provide the model with richer gradient information, improving its ability to represent new classes. To learn from human feedback, we treat new-class samples as corrected samples, retaining their outputs from the old model, and then select representative and confusable samples to construct a replay set. To mitigate catastrophic forgetting, we employ a hierarchical distillation to preserve the model's previous knowledge.

Our contributions can be summarized as follows:

1. We introduce an interactive event-detection mode: "Machine recommends event triggers→Manual review and correction→Machine incremental learning". This model better aligns with the operational mode required for high-precision applications and provides a reference for subsequent research.
2. We designed the ICD method for class-incremental learning in interactive event-detection tasks. This method constructs a replay set based on representative and confusable samples, designs semantic-boundary smoothing-based contrastive learning to improve performance with few samples, and employs hierarchical distillation to preserve existing knowledge.
3. Experimental results demonstrate that our method significantly outperforms other baseline methods in class-incremental learning tasks within interactive detection. Specifically, under the 2-way 5-shot and 2-way 10-shot conditions of ACE and DuEE, our method improves upon the best baseline methods by 6.86%, 4.53%, 3.90%, and 3.00%, respectively.

2. Related Work

2.1. Event Detection

Deep learning-based event detection leverages neural networks to extract semantic features. The advent of pre-trained models like BERT [6] has introduced the "pre-training + fine-tuning" paradigm, enhancing model performance. Methods are generally categorized into span-based classification, sequence labeling, and pointer networks.

2.1.1. Span-Based Classification

This method directly classifies text segments to determine their event types, sometimes restricting the segment length to 1, degenerating into character-level classification. The DMCNN model proposed by Chen et al. [7] uses convolutional neural networks and dynamic multi-pooling layers to extract semantic features from characters. Wang et al. [8,9] enhance word representations by incorporating event type hints, while Guan et al. [10] improve trigger-word prediction accuracy through trigger-argument interaction information. Guzman-Nateras et al. [11] and Yue et al. [12] investigated few-shot learning scenarios, and Liu et al. [13] studied partially annotated sample scenarios. Xu et al. [14] used trigger-argument interaction attention to obtain enhanced segment representations, showing excellent performance in event-detection tasks.

2.1.2. Sequence Labeling

This method annotates text sequences through labeling schemes to detect triggers. The JRNN model proposed by Nguyen et al. [15] uses two recurrent neural networks to learn sentence representations and decodes them using conditional random fields. Wei et al. [16] enhance sentence semantic representation through dialogue and adopt the B-I-O sequence labeling strategy. Zheng et al. [17] extended the B-I-O label scheme to accommodate complex scenarios, but there are label ambiguity issues. Tian et al. [18] used bidirectional long short-term memory networks to detect events in the financial domain and proposed a new sequence labeling scheme to address the multi-event problem. Cao et al. [19] designed a word-pair labeling scheme in the OneEE model, modeling the event extraction task as a word-pair relationship identification and jointly decoding event triggers and arguments. Li et al. [20] combined sequence labeling with text-generation tasks, attempting to inject external knowledge through prompts.

2.1.3. Pointer Networks

This method uses pointer networks to indicate text segments, addressing overlapping and nested issues. The PLMEE model proposed by Yang et al. [21] uses multi-layer pointer networks to determine the start and end positions of event triggers. Du et al. [22], Chen et al. [23], and Li et al. [24] modeled event detection as a question-answering task, using head and tail pointers to locate triggers. Machine reading comprehension (MRC) models also follow this approach.

2.1.4. Others

Some approaches model event detection as a sentence-level classification task. Wan et al. [25] used a bidirectional attention mechanism to capture the overall semantics of a sentence, but did not identify the specific location of the trigger. Yan et al. [26] leveraged an external knowledge graph to create a type hierarchy, enhancing the semantic representation of event types. They also divided the input text into word-level and context-level features. However, their method only determines the presence of an event type without identifying the trigger, which limits its effectiveness for downstream tasks.

2.2. Incremental Learning

To address the problem of catastrophic forgetting in machine learning and enable incremental learning capabilities in models, researchers have explored various approaches.

These approaches can be broadly categorized into three main types, based on how historical information is preserved: regularization, replay, and parameter isolation.

Regularization: Li and Hoiem's Learning without Forgetting (LwF) [27] (2016) is a classic method based on regularization. It uses knowledge distillation to ensure that the outputs of the new model stay as close as possible to those of the old model. Kirkpatrick et al.'s EWC [28] (2016) algorithm is based on a Bayesian framework and introduces additional parameter-related regularization loss, constraining parameters according to their importance.

Replay: Rebuffi and Sylvestre-Alvise et al.'s iCaRL [29] (2016) is a classical incremental-learning model based on replay methods. They assumed that samples closer to the class-feature centers are more representative, and selected a subset of these representative old data to constrain the model updates, thus preserving the features learned from old data. Lopez-Paz et al.'s Gradient Episodic Memory (GEM) [30] (2017) stores some of the previous data in an "Episodic Memory" and uses gradient updates to constrain the direction of new task updates without altering the parameters of old tasks. Replay-based methods have been shown to be the most promising for NLP tasks [2,3].

Parameter Isolation: Mallya et al.'s PackNet [31] (2018) is a typical parameter-isolation method. Whenever a new task arrives, it prunes redundant model space to leave room for new task parameters and allocates space for each task. The parameters of old tasks are fixed to train new data, and once completed, the model is pruned to remove less important parameters, followed by incremental training and the allocation of new parameters. Serra et al.'s HAT [32] (2018) uses attention mechanisms to mask certain parameters in the model, preserving historical task information without affecting the current task. For each task, a hard attention mask is learned, allowing for optimal allocation of model space for different tasks.

The aforementioned research primarily focuses on computer vision tasks, especially image classification.

2.3. Class-Incremental Learning in Event Detection

Since 2020, class-incremental learning tasks have gradually gained attention in the field of event detection. The main challenge lies in learning new tasks while avoiding the forgetting of previously learned tasks. To address this challenge, Cao et al. [33] and colleagues proposed an innovative Knowledge-Preserving Incremental Heterogeneous Graph Neural Network (KPGNN) specifically designed for incremental social-event detection. The KPGNN facilitates effective data utilization by modeling complex social information into a unified social graph, and explores the strong expressive capability of Graph Neural Networks (GNNs) in knowledge extraction. Cao et al. [2] introduced a replay–distillation method to retain knowledge through memory sets and previous models. In addition to replay and distillation, Yu et al. [3] used initialization methods to transfer knowledge, while Liu et al. [4] employed soft-prompt learning techniques to preserve previous knowledge. Wei et al. [34] incorporated knowledge-transfer modules between each new class to maintain the feature space of old class events. Liu et al. [5] independently trained classification nodes for each new-class event and combined all learned classifiers during inference to mitigate classifier drift.

While the methods of these works perform well in class-incremental learning, they perform poorly in few-shot cases, which are more common in interactive event detection.

3. Problem Definition

The task definition for machine recommendation of triggers in interactive event detection is detailed in prior work [1]. The class-incremental learning task involves incrementally learning new event types from a small number of labeled data reviewed manually. Given a set of tasks $T = \{T_1, T_2, \ldots, T_n\}$, each task T_i has training, validation, and test sets denoted as $T_i = \left\{D_i^{train}, D_i^{valid}, D_i^{test}\right\}$, where $D_i = \left\{\left(X_i^j, E_i^j, Y_i^j\right)\right\}_{j=1}^{m}$, m represents the number of

event types in each task. X_i^j, E_i^j and Y_i^j are the sentence-level text, triggers, and corresponding event types, respectively. The first task is considered the base task, denoted as T_{base}, and contains sufficient training data. The remaining tasks $\{T_2, \ldots, T_n\}$ are class-incremental tasks, denoted as T_{incre}, and each has only a small number of training samples (set to 5 or 10 in our study). For any two tasks, T_i and T_j, the event types they include should not overlap, i.e., $T_i \cap T_j = \varnothing$. During task t t, we validate and evaluate the model on all previously learned event types, i.e., $C_t^{test} = D_t^{test} \cup C_{t-1}^{test}$.

4. Method

Our proposed incremental learning method for interactive event-detection ICD, as illustrated in Figure 2, includes an event-trigger recommendation backbone model, a representative- and confusable-sample replay module, a semantic-boundary-smoothing contrastive learning module, and a hierarchical-distillation module.

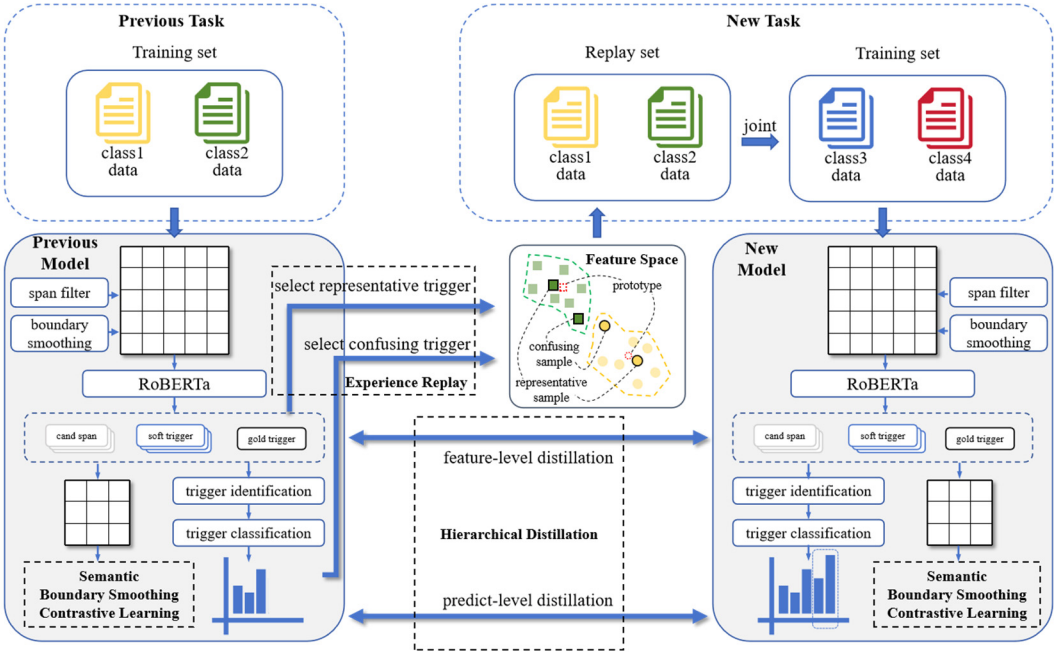


Figure 2. The structure of ICD, which includes a recommendation model, experience replay, contrastive learning, and hierarchical-distillation modules. When learning a new task, it uses the previous model to select replay samples, applies hierarchical-knowledge distillation, and then employs semantic-boundary-smoothing contrastive learning for the new model.

4.1. Event-Trigger Recommendation Backbone

Consider the training data (X, E, Y), where $X = \{x_1, x_2, \ldots, x_l\}$ is the sentence-level text, $E = \{e_1, e_2, \ldots, e_n\}$ is the set of event triggers, and $Y = \{y_1, y_2, \ldots, y_n\}$ is the event type corresponding to E. Using the pre-trained language model RoBERTa as the encoder, the text X is input into the pre-trained model to obtain contextual representations $W = \{w_1, w_2, \ldots, w_l\}$, where $w_i \in \mathbb{R}^d$ denotes the word vector of the i-th word, and d is the vector dimension. The model treats spans as the smallest unit and generates a candidate span set $S = \{s_i | i = 1, 2, \ldots, N_1\}$ through length- and rule-based filtering. For each candidate span $s_i = \{x_{st_i}, \ldots, x_{ed_i}\} = (st_i, ed_i)$, we concatenate the word vectors of its head and tail and a length, embedding $\mathrm{h_i^{length}} \in \mathbb{R}^{\mathrm{d_w}}$ to obtain its representation $h_i = \left[w_{st_i}; w_{ed_i}; h_i^{length}\right]$. This representation is then fed into a binary classifier composed of a multi-layer perceptron

MLP^1 and a sigmoid activation function to compute and output the confidence score $p_i^{trigger}$ for s_i, which is a potential trigger.

$$p_i^{trigger} = \text{Sigmoid}\left(\text{MLP}^1(h_i)\right), \tag{1}$$

The multi-class classifier, composed of a multi-layer perceptron MLP^2 and a softmax activation function, calculates and outputs the confidence vector $\boldsymbol{p}_i^{event\_type} \in \mathbb{R}^{n+1}$ for span s_i, indicating the probability of belonging to each of the predefined event types, where n represents the total number of event types.

$$\boldsymbol{p}_i^{event\_type} = \text{softmax}\left(\text{MLP}^2(h_i)\right), \tag{2}$$

Building on our previous work [1], a semantic-boundary-smoothing approach is employed, which assigns soft labels $\widetilde{y}_i \in [0, 1]$ to s_i based on the following three aspects: overlap with the corresponding gold trigger e_i, semantic integrity, and part-of-speech information. Specifically, for the overlap score f_1, as shown in Formula (3), we use Intersection over Union(IoU) to measure it, where $|s_i \cap e_i|$ represents the length of characters in the intersection of the two spans, and $|s_i \cup e_i|$ represents the length of characters in their union. In addition, we use pkuseg [35] to provide segmentation and part-of-speech information for the text. For semantic integrity score f_2, as shown in Formula (4), if the span s_i is in the segmentation results, it is considered to have complete semantics, scoring 1; otherwise, it scores 0.5. For part-of-speech information f_3, as shown inf Formula (5), if the part-of-speech combination of s_i has appeared in the training set, it is deemed reasonable, scoring 1; otherwise, it scores 0.5.

Combining these three aspects, the soft label $\widetilde{y}_i$ of s_i is calculated as Formula (6):

$$f_1(s_i, e_i) = \text{IoU}(s_i, e_i) = \frac{|s_i \cap e_i|}{|s_i \cup e_i|} \tag{3}$$

$$f_2(s_i) = \begin{cases} 1, & s_i \text{ is in segmentation results} \\ 0.5, & \text{otherwise} \end{cases} \tag{4}$$

$$f_3(s_i) = \begin{cases} 1, & \text{the part of speech of } s_i \text{ is reasonable} \\ 0.5, & \text{otherwise} \end{cases} \tag{5}$$

$$\widetilde{y}_i = f_1(s_i, t_i) \cdot f_2(s_i) \cdot f_3(s_i), \tag{6}$$

The trigger-identification loss is calculated using a weighted adaptive-focal loss, while the trigger classification loss is computed using multi-class cross-entropy, as follows:

$$\begin{aligned} L_{TI} = -\sum_{i}^{N_1} & \mathbf{1}_{[\widetilde{y}_i \geq \alpha]} w_i \left(1 - p_i^{trigger}\right)^{\gamma} \log p_i^{trigger} \\ & + \mathbf{1}_{[\widetilde{y}_i < \alpha]} w_i p_i^{trigger^{\gamma}} \log\left(1 - p_i^{trigger}\right), \end{aligned} \tag{7}$$

$$L_{TC} = -\sum_{i}^{N_2} \boldsymbol{y}_i^{event\_type} \log \boldsymbol{p}_i^{event\_type}, \tag{8}$$

where α is the threshold for trigger identification, γ is the focusing parameter of focal loss, and $\boldsymbol{y}_i^{event\_type} \in \mathbb{R}^{n+1}$ is the one-hot label for the event type of s_i. If $\widetilde{y}_i \geq \alpha$, s_i is considered to have the same event type as e_i; otherwise, it is treated as the "other" category.

In terms of model inference, we obtain the candidate trigger set $S' = \left\{s'_1, \ldots, s'_i, \ldots, s'_{N_2}\right\} \subseteq S$ based on the trigger-identification threshold α and compute the joint confidence $\boldsymbol{p}_i^{joint} \in \mathbb{R}^{n+1}$:

$$\boldsymbol{p}_i^{joint} = 2\left(\boldsymbol{p}_i^{trigger} \circ \boldsymbol{p}_i^{event\_type}\right) / \left(\boldsymbol{p}_i^{trigger} + \boldsymbol{p}_i^{event\_type}\right), \tag{9}$$

Based on the trigger confidence $p_i^{trigger}$ and spatial distribution, potential events in the input text are identified. Then, candidate triggers and their event types for the event are recommended according to the joint confidence, as shown in Algorithm 1.

Algorithm 1 Candidate recommendation algorithm considering multiple events

Input: candidate trigger set $S' = \left\{s'_1, \ldots, s'_i, \ldots, s'_{N_2}\right\}$,
candidate trigger confidence tuple set $P = \left\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i, \ldots, \boldsymbol{p}_{N_2}\right\}$,
where $\boldsymbol{p}_i = \left(p_i^{trigger}, \boldsymbol{p}_i^{event\_type}, \boldsymbol{p}_i^{joint}\right)$
Output: candidate triggers and event type for all potential events O

$O \leftarrow \{\}$ # Initialize the model output
Arrange the S' in descending order according to trigger confidence $p_i^{trigger}$
$Q \leftarrow \{\}$ # The set of candidate triggers representing potential events
for s'_i in S' **do**
  if s'_i does not overlap with any $q_j \in Q$ **do**
    $Q \leftarrow Q \cup \left\{s'_i\right\}$
  end if
end for
for q_j in Q **do**
  $I \leftarrow \{\}$ # Candidate triggers and their confidence tuple for one potential event
  for s'_i in S' **do**
    if s'_i overlaps with q_j **do**
      $I \leftarrow I \cup \left\{\left(s'_i, \boldsymbol{p}_i\right)\right\}$
    end if
  end for
  $O \leftarrow O \cup \left\{TopK\left(I, p_{i,j}^{joint}\right)\right\}$ # For each potential event, the maximum K joint confidences fidences are found, and their corresponding tuples, like (candidate trigger, event type, joint confidence), are output as recommendations
end for

4.2. Experience Replay

Experience replay is an effective approach to mitigate catastrophic forgetting in class-incremental learning [2,3]. In the class-incremental learning process for interactive event detection, due to the limited number of new-class-event samples, replay can only select from these limited samples. To address more extreme cases, this study restricts the storage of old class events to a maximum of 2 samples per class. To extract the most useful information from these limited samples, we employ a selection strategy based on representative and confusable samples. The selection process for event type j is as follows.

4.2.1. Representative Sample

Each event is treated as a sample, and the average representation of triggers for each event type is used as the prototype. The sample closest to this prototype is selected as the representative sample. For the event type j, the event prototype is calculated as follows:

$$\mu^j = \frac{1}{\left|\boldsymbol{E}^j\right|} \sum_{e_i^j \in \boldsymbol{E}^j} h_i^j, \tag{10}$$

where $\boldsymbol{E}^j$ represents the set of triggers for event type j in the training set, and h_i^j denotes the representation of the trigger e_i^j that belongs to event type j. The event prototype μ^j is considered a point in the feature space, and the Euclidean distances between each event trigger and the prototype in the feature space are computed. The event trigger closest to the prototype is selected as the representative trigger for event type j, and the data associated with this trigger are added to the replay set.

4.2.2. Confusable Sample

To enhance the model's weaknesses in specific categories and features, we consider replaying samples that are easily confused, thereby improving the model's classification ability and generalization performance. Specifically, let $\boldsymbol{p}_i^j$ be the probability distribution of the event trigger e_i^j. The degree of confusion of a trigger is measured by the difference $\Delta_i^j = p_{max} - p_{second}$, where the maximum probability is $p_{max} = \max\left(\boldsymbol{p}_i^j\right)$ and the second-largest probability is $p_{second} = \max\left(\boldsymbol{p}_i^j / \{p_{max}\}\right)$. The trigger with the largest difference is selected, and the data associated with it are added to the replay set.

4.3. Semantic-Boundary-Smoothing Contrastive Learning

In few-shot class-incremental learning, the scarcity of new-class samples often leads to overfitting on the new classes. To address this issue, we use soft labels constructed through semantic-boundary smoothing in Section 4.1 to create positive and negative example sets for triggers. By designing a contrastive loss, we provide the model with richer gradient information, thereby enhancing its ability to represent new classes.

For the boundary-smoothing area surrounding the gold trigger e_i, we categorize it into positive- and negative-example sets based on the soft labels $\widetilde{y}_i$ and the trigger identification threshold α. A contrastive learning loss is designed for the span around the gold trigger, which aids in improving the model's ability to recommend the correct answers from the surrounding span. The contrastive loss for the semantic-boundary-smoothing area is computed as follows:

$$L_{con} = \sum_{(X,E,Y)\in T_t} \sum_{e_i \in E} \frac{1}{|\boldsymbol{e}_i^+| + |\boldsymbol{e}_i^-|} \sum_{s_j \in \boldsymbol{e}_i^+} \sum_{s_k \in \boldsymbol{e}_i^+} -\log \frac{exp(h_j, h_k)}{\sum_{s_p \in \boldsymbol{e}_i^-} exp(h_j, h_p)} \tag{11}$$

4.4. Hierarchical Distillation

In class-incremental learning, knowledge distillation has proven to be an effective method for preventing the model from excessively favoring new-class knowledge, helping to avoid catastrophic forgetting [2]. Specifically, we employ a hierarchical-distillation method. For task T_t, we use the model from task T_{t-1} as the teacher model to perform both feature-level and prediction-level distillation.

4.4.1. Feature-Level Distillation

Let the features of the triggers output by the new-task and old-task models be $\widetilde{h}_i'$ and h_i', respectively. After normalizing these features, the similarity between them is measured using cosine similarity. The feature-level distillation loss is calculated as follows:

$$L_{fd} = \sum_{(X,E,Y)\in T_t} 1 - S(\tilde{h}_{i'}', h_i') \tag{12}$$

If the features extracted by the current model are similar to those of the previous model, then the encoder can effectively preserve previous knowledge.

4.4.2. Predict-Level Distillation

Let the new task T_t include c new event types, while the previously learned tasks include m event types. The probability distributions of the trigger e_i belonging to different event types are obtained from the multi-class classifier of the previous and current model, resulting in $\widetilde{\boldsymbol{p}}^{event\_type} = [\widetilde{p}_1, \widetilde{p}_2, \ldots, \widetilde{p}_m]$ and $\boldsymbol{p}^{event\_type} = [p_1, p_2, \ldots, p_m, p_{m+1}, \ldots, p_{m+c}]$, respectively. The feature-level distillation loss is computed as follows:

$$L_{pd} = \sum_{(X,E,Y)\in T_t} \sum_{i=1}^{m} \widetilde{v}_i \log(v_i) \tag{13}$$

$$\widetilde{v}_i = \frac{exp(\widetilde{p}_i/\tau)}{\sum_{j=1}^{m} exp(\widetilde{p}_j/\tau)}, \qquad v_i = \frac{exp(p_i/\tau)}{\sum_{j=1}^{m} exp(p_j/\tau)} \tag{14}$$

4.5. Model Training

At the T_{base} task, the base model is trained using sufficient data, with L_{TI} and L_{TC} as the training objectives. During the T_{inc} phase, the training data consist of the new-class training set and an old-class replay set. In addition to the fundamental L_{TI} and L_{TC}, we also employ contrastive loss L_{con} to enhance learning of new-class events, and hierarchical-distillation losses L_{fd} and L_{pd} to prevent forgetting of old-class events. Each loss function is assigned a weight λ_i, where $i\epsilon\{TI, TC, con, fd, pd\}$.

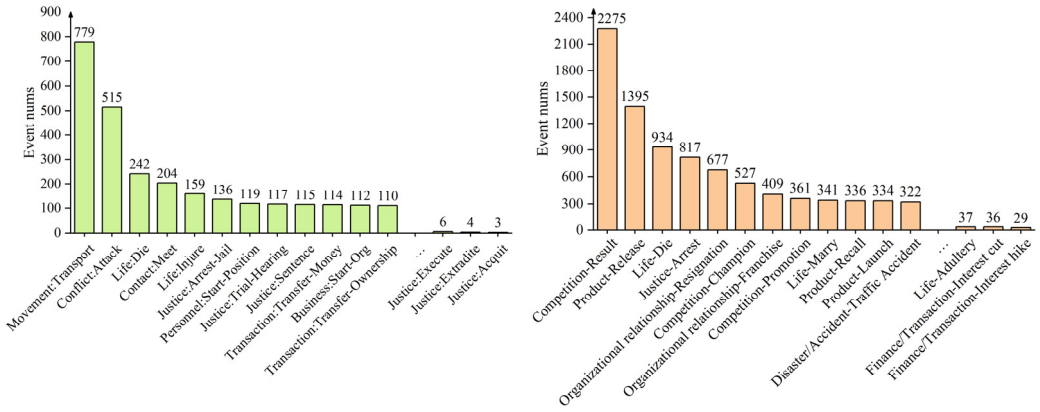
5. Experiments

5.1. Benchmarks and Evaluation Metrics

ACE2005-Chinese, which is a widely used benchmark for event-detection tasks, encompassing a collection of documents from various domains, such as news texts, broadcast dialogues, and blogs. We specifically selected the Chinese portion of this dataset. It defines 9 major event categories and 33 subcategories.

DuEE [36], which is currently the largest Chinese event-detection dataset, was developed jointly by Baidu, the Chinese Computer Federation, the Chinese Information Processing Society, and data resource developers from universities and enterprises. The dataset includes a substantial amount of text from Baidu News, with a relatively consistent annotation style. It defines 9 major event categories and 65 subcategories.

The distribution of event quantities in these two datasets is illustrated in Figure 3.



(**a**) ACE2005-Chinese (**b**) DuEE

Figure 3. The distribution of event counts in the ACE2005-Chinese and DuEE datasets.

We selected the top-10 event types with the highest frequency from both the ACE and DuEE datasets to construct a class-incremental learning setup with 5 sub-tasks. Each sub-task includes 2 event types, and experiments were conducted with 2-way 5-shot and 2-way 10-shot settings. The data partitioning for these two datasets is detailed in Table 1.

Table 1. Class-incremental learning-task data partitioning.

Datasets	Task	Training Set	Validation Set	Test Set
ACE2005	T_{base}	100	30	30
	T_{incre}	5/10	30	30
DuEE2005	T_{base}	70	20	20
	T_{incre}	5/10	20	20

The strict matching standard was adopted; that is, a result is considered correct only if both the boundary of the trigger and the event type are all correct. *Recall@K* on the test set was used to evaluate the model, where *K* denotes the number of candidates.

$$Recall@K = \frac{N_{recall}@K}{N_{gold}}, \tag{15}$$

where N_{gold} denotes the total number of gold triggers and $N_{recall}@K$ denotes the number of recalled gold triggers when *K* candidates are recommended.

It is important to note that our previous research [1] indicated that the overall efficiency of the recommendation model is highest when recommending three candidate items. Therefore, the subsequent experiments will use *Recall*@3 as the evaluation metric. To ensure that the overall efficiency is not adversely affected by excessively low precision, we require that the model's precision for potential event detection must exceed 33% at the time of model saving.

5.2. Implementation Details

Both the ICD method and the baseline methods use the RoBERTa pre-trained model as an encoder to obtain word embeddings for the text sequences. During training, the AdamW optimizer is employed, and to ensure stable model training, learning rate warm-up and decay strategies are also implemented. The experimental environment configuration is detailed in Table 2, while the parameters and their meanings for the ICD method are provided in Table 3.

Table 2. Experimental environment.

Item	Configuration
Operating system	Ubuntu 20.04.3 LTS x86_64
CPU	40 vCPU Intel(R) Xeon(R) Platinum 8457C
GPU	L20(48 GB) × 2
Memory	200 GB
Python	3.8.10
Framework	Pytorch(1.10.0) +Transformers(4.38.2) + Cuda(12.4)

Table 3. Parameter setting.

Parameter	Description	Value	
		ACE2005	DuEE
	Main parameters		
α	trigger-identification threshold	0.2	0.2
β	positive/negative threshold for contrast	0.2	0.2
γ	focusing parameter of modified focal loss	2	2
L	limitation of span length	5	5
	Optimizer parameters		
Learning_rate	initial learning rate	1×10^{-5}	1×10^{-5}
warm_up_steps	warm-up steps	500	500
lr_decay_steps	decay steps	3000	3000
min_lr_rate	minimum learning rate	1×10^{-6}	1×10^{-6}

5.3. Baseline Methods

Joint: This method involves training the model directly with joint data at once, instead of in increments.

Replay all: This method entails training the model using the previous training datasets combined.

Fine-tune: The crux of this method lies in directly fine-tuning the model on the new task.

LwF [27]: This method centers around incorporating a distillation module to match the probabilities of the previous model, preserving prior knowledge.

KCN [2]: This method is a popular continuous event-detection method following the memory replay–knowledge distillation paradigm.

KT [3]: This method generally follows a memory-based paradigm and uses novel initialization methods to transfer knowledge.

EMP [4]: In addition to memory replay, this method introduces prompt learning for each event type to integrate previous knowledge.

ICE-PL&O [5]: This method freeze the encoder and previous classifier, set fixed logits for the "other" class, and trains a new classifier for new task to avoid forgetting.

5.4. Main Results

We conducted five experiments for each incremental learning task using different random seeds, and reported the average performance of our method on the ACE2005-Chinese and DuEE datasets, comparing it with the aforementioned baseline methods. The results are shown in Tables 4 and 5. To visualize the trends more clearly, we also plotted the corresponding line chart, as shown in Figure 4.

Table 4. *Recall*@3 performance of every sub-task on ACE2005-Chinses for comparative experiment.

ACE2005-Chinses										
Methods	**2-Way 5-Shot**					**2-Way 10-Shot**				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Joint	/	/	/	/	64.62	/	/	/	/	67.43
Replay all	93.22	68.87	58.21	62.92	61.73	93.22	72.64	65.17	66.25	64.26
Fine-tune	93.22	64.15	58.21	54.17	48.38	93.22	76.42	70.15	59.17	51.79
LwF	93.22	65.09	59.20	65.42	61.73	93.22	80.19	71.64	70.83	61.01
KCN	93.22	65.09	58.71	67.08	58.12	93.22	80.19	71.14	72.50	64.26
KT	93.22	65.09	64.18	65.42	59.57	93.22	80.19	68.66	75.00	68.59
EMP	90.08	68.12	62.34	68.27	64.62	90.08	82.01	72.75	73.94	68.08
ICE-PL&O	93.22	64.38	59.41	62.76	57.83	93.22	66.23	61.54	63.9	60.56
ICD(Ours)	**93.22**	**80.19**	**73.13**	**74.28**	**71.48**	**93.22**	**83.96**	**78.12**	**75.97**	**73.12**

The best performances are highlighted in bold and the second-best are underlined. The same applies below.

Table 5. *Recall*@3 performance of every sub-task on DuEE for comparative experiment.

DuEE										
Methods	**2-Way 5-Shot**					**2-Way 10-Shot**				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Joint	/	/	/	/	84.08	/	/	/	/	87.99
Replay all	98.55	82.61	89.39	89.58	81.68	98.55	84.78	83.84	88.80	84.38
Fine-tune	98.55	86.96	86.87	83.40	75.38	98.55	91.30	86.87	81.08	76.28
LwF	98.55	87.68	89.90	90.73	86.49	98.55	88.04	89.39	91.51	87.09
KCN	98.55	86.96	87.88	90.73	84.38	98.55	89.13	88.38	92.66	86.67
KT	98.55	86.23	88.89	90.73	84.38	98.55	87.68	88.89	91.12	86.49
EMP	97.10	87.68	89.13	91.22	86.49	97.10	89.13	89.39	92.00	89.49
ICE-PL&O	98.55	84.22	86.34	85.19	79.28	98.55	84.26	86.66	85.90	79.37
ICD(Ours)	**98.55**	**92.03**	**92.93**	**93.82**	**90.39**	**98.55**	**94.93**	**93.43**	**94.60**	**92.49**

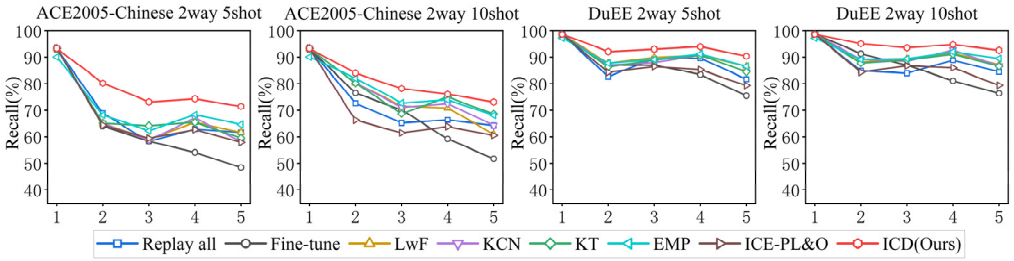


Figure 4. *Recall*@3 performance of every sub-task on ACE2005-Chinses and DuEE for comparative experiment.

From the results, we can observe the following:

- Compared to previous incremental-learning baseline methods, our approach significantly outperforms these methods across all sub-tasks. Specifically, in the ACE 5-shot and 10-shot, and DuEE 5-shot and 10-shot settings, our method improves the final recall rate by 6.86%, 4.53%, 7.69%, and 4.53%, respectively, compared to the best baseline model. This demonstrates the clear effectiveness of our method in handling continuously arriving data.
- Although the Joint method retrains using all available data, it is not always the optimal baseline method. In our setup, the initial task has a sufficient number of samples, while the training samples for subsequent new tasks are limited. This imbalance in sample sizes affects the performance of the Joint method. In contrast, our approach improves upon Joint by 6.86%, 5.69%, 3.90%, and 3.00% in the ACE 5-shot and 10-shot, and DuEE 5-shot and 10-shot settings, respectively. This indicates that our method effectively mitigates the negative impact of sample imbalance in few-shot incremental learning conditions.
- The recall rates in the 10-shot setting are generally higher than in the 5-shot setting, which suggests that increasing the number of samples is an effective way to enhance model performance. This implies that generating more samples can address the challenges posed by few-shot learning. Additionally, our performance on ACE is noticeably better than on DuEE. We believe this is because ACE involves greater variability in the expression patterns of the same event type, making it more challenging, whereas DuEE's triggers have a more consistent style, which is relatively simpler.

5.5. Ablation Study

We conduct ablation experiments on the DuEE dataset under the 5-shot and 10-shot scenarios to investigate the effectiveness of various components of the ICD method. The experimental settings are as follows: "-replay" indicates the absence of old-class data replay, "-replay(representative)" indicates that old-class data is replayed randomly, rather than with representative samples, "-replay(confusable)" indicates that old-class data is replayed randomly instead of with confusable samples, "-replay(repr&conf)" means neither representative nor confusable samples are used, resulting in random replay, "-contrast" refers to the exclusion of semantic-boundary-smoothing contrastive learning, and "-distill" indicates the absence of hierarchical-distillation loss. The results are shown in Table 6.

- **Effectiveness of Replay**: The inclusion of old-class data in replay increases the final recall rate by 8.41% and 5.70%, respectively, demonstrating a significant effect of storing old data on incremental learning. Furthermore, compared to random replay, selecting representative and confusable samples improves the recall rate of the final sub-task by 4.18% and 2.55%, respectively. In addition, the ablation of either component leads to a decrease in the final recall rate, while the simultaneous ablation of both components results in an even further decline. This indicates that each component plays a positive role, and, when used together, they produce additional benefits, highlighting the fact that constructing replay samples from these two perspectives adds more value.

- **Effectiveness of Contrastive Learning**: Our method improves the final recall rate by 5.68% and 4.10%, compared to the removal of contrastive learning. This suggests that, in the context of few-shot incremental learning, using contrastive learning with spans around gold triggers provides additional information to the model, thereby improving performance.
- **Effectiveness of Hierarchical Distillation**: Compared to the exclusion of hierarchical distillation, our method achieves an increase in final recall rate by 1.71% and 2.06%, respectively. This relatively modest improvement indicates that while distillation loss helps retain original knowledge, its impact is somewhat limited.

Table 6. *Recall*@3 performance of every sub-task on ACE2005-Chinses for ablation study.

	DuEE									
Methods	**2-Way 5-Shot**					**2-Way 10-Shot**				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
ICD (Ours)	**98.55**	**92.03**	**92.93**	**93.82**	**90.39**	**98.55**	**94.93**	**93.43**	**94.60**	**92.49**
-replay	98.55	90.66	88.89	84.17	81.98	98.55	89.20	89.93	91.66	86.79
-replay (representative)	98.55	92.03	92.07	92.81	87.24	98.55	94.39	91.16	92.99	91.13
-replay (confusable)	98.55	91.78	90.53	93.54	88.26	98.55	94.93	93.94	93.82	91.59
-replay (repr&conf)	98.55	92.16	89.53	92.54	86.21	98.55	94.93	90.47	91.74	89.94
-contrast	95.65	90.66	91.79	92.17	84.71	95.65	92.95	92.33	92.08	88.39
-distill	98.55	91.20	92.76	93.12	88.68	98.55	94.93	92.42	93.05	90.43

5.6. In-Depth Analysis

5.6.1. Analysis of Old- and New-Class Performance

Figure 5 illustrates the recall rates of our method compared to the knowledge distillation-based KT [3] method under the 5-shot settings on the ACE and DuEE datasets for both new and old event types. As shown in the figure, our method significantly outperforms KT in learning new event types, indicating that our approach is more effective in identifying new classes of events under few-shot conditions. Additionally, our method also shows a slight improvement over KT in handling old tasks, suggesting that the integration of components in our method enhances not only the detection of new events, but also the performance on existing tasks.

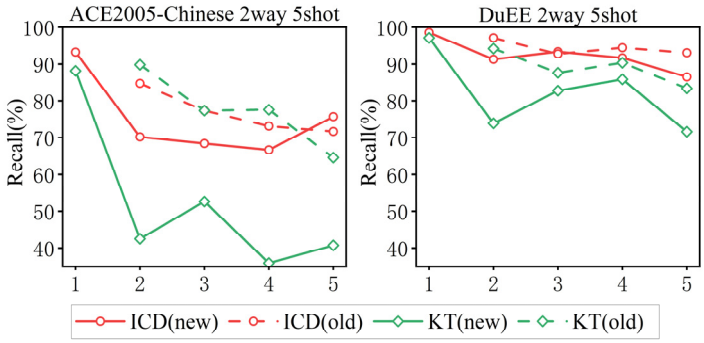


Figure 5. *Recall*@3 performance on old and new tasks in each sub-task on ACE2005-Chinses and DuEE.

5.6.2. Analysis of Positive/Negative Threshold for Contrast

In the semantic-boundary-smoothing contrastive learning, the setting of threshold β determines the division between positive and negative examples, representing the model's strictness in distinguishing between them and influencing the gradient updates. Previous research [37] indicates that the robustness of the model to sample perturbations is an important aspect of model performance. The threshold for positive and negative samples

determines the distinction between them, representing the model's strictness in differentiating these samples, which can be seen as a form of perturbation in the training samples. Given that this hyper-parameter is closely related to event patterns and dataset annotation styles, selecting an appropriate threshold is crucial.

We tested five different threshold values and reported their recall rates on the final task. The results are shown in Table 7. and it indicate that a threshold of 0.2 yields the best overall performance. A threshold that is too high may result in too few positive examples, leading to limited effectiveness with a small sample size, whereas a threshold that is too low might lead to an excessive number of positive examples, affecting the differentiation between positive and negative cases. Therefore, we selected 0.2 as the final threshold setting.

Table 7. *Recall*@3 performance on the final sub-task with different value of threshold β.

Threshold	ACE		DuEE	
	5-Shot	10-Shot	5-Shot	10-Shot
0.1	89.69	91.89	68.59	71.11
0.2	**90.39**	**92.49**	**71.48**	**73.12**
0.3	89.69	91.29	66.07	71.11
0.4	87.69	86.49	63.90	71.11
0.5	85.59	87.69	62.46	67.37

5.6.3. Evaluation in Extreme Scenarios

To further investigate the performance of the ICD method under extreme conditions, where each new class has only 1 or 2 samples, we conducted experiments with 2-way 1-shot and 2-way 2-shot settings. The results of these experiments are illustrated in Figure 6.

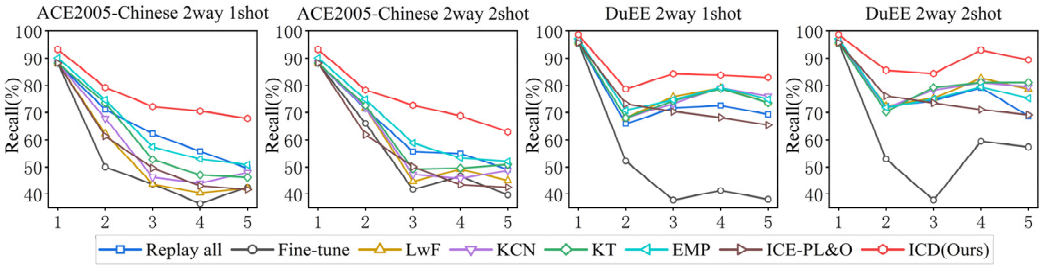


Figure 6. *Recall*@3 performance of every sub-task on ACE2005-Chinses and DuEE for extreme scenarios.

Under extreme conditions with very few samples, we observe a significant decline in the performance of all methods, indicating the substantial challenge posed by such scenarios. However, the ICD method still outperforms other methods under these conditions. We attribute this advantage primarily to the contrastive learning objective constructed based on the semantic-boundary-smoothing method. This approach provides richer and more effective gradient information for model training under limited sample conditions, thereby enhancing the model's performance and stability with sparse data resources.

5.6.4. Analysis of Computational Complexity

The PyTorch framework performs parallel computations using tensors, and discussing time complexity or space complexity in isolation is not practically meaningful. Therefore, we discuss the overall computational complexity from the perspectives of both the backbone model and the incremental learning methods to assess the required computational resources.

In terms of the backbone model, traditional span-based models have a high computational cost, as for a text containing n characters, it is theoretically possible to partition it into

n × (n + 1)/2 spans, leading to a computational complexity of $\mathcal{O}(n^2)$. In contrast, models based on character classification and sequence labeling process each character individually, resulting in a complexity of only $\mathcal{O}(n)$. Our proposed span-based backbone model reduces the computational complexity to $\mathcal{O}(nL)$ by limiting the span length L and punctuation filtering, significantly saving computational resources. Although its complexity is still higher than that of character classification and sequence labeling models, the time difference is acceptable, considering the parallel computing features of PyTorch(1.10.0). It is important to note that this paper focuses on incremental learning methods, and all baseline methods use the same span-based backbone model, so this part of the computational complexity is analyzed only theoretically.

In terms of incremental learning methods, we set the batch size to 4 and epoch to 30, with other parameters consistent with those in Section 5.2. We report the average time per epoch for different baseline methods trained on the ACE2005-Chinese and DuEE datasets, as shown in Figure 7.

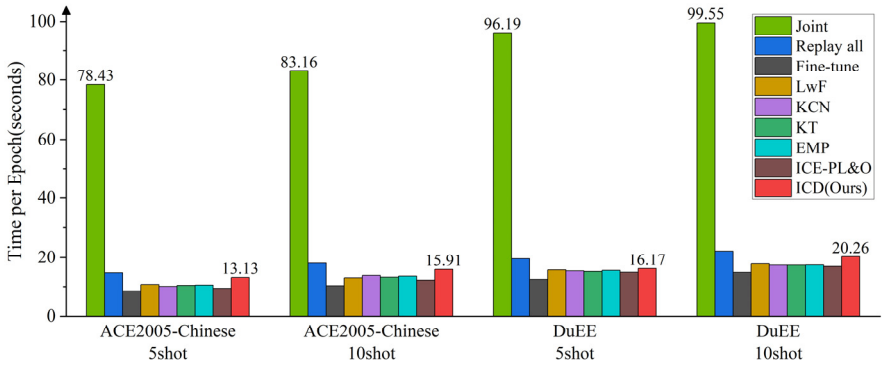


Figure 7. Average time consumption per epoch of ICD and baseline methods under different settings.

From the results, we can see that the Joint method, which trains on all data at once, consumes the most time. Our method, ICD, consumes less time compared to the Replay all method, averaging a reduction of 2.24 s, while slightly increasing time compared to other baseline methods, particularly the baseline method EMP, which shows the best overall performance, with an average increase of 2.1 s. This is analyzed to be due to the additional loss introduced by the semantic-boundary smoothing in the contrastive learning module, which brings extra gradient information but also increases the computational load, resulting in a slight increase in model training time. We believe that achieving a significant improvement in recall with such a small time cost is meaningful.

6. Conclusions

We have introduced an interactive event-detection mode that operates through a cycle of "machine-recommended event triggers→human review and correction→machine incremental learning" to facilitate event detection. To address the challenges of class-incremental learning, we introduced the ICD method. To mitigate forgetting of old-class events, we designed a replay strategy based on representative and confusable samples. Additionally, we applied semantic boundary smoothing techniques to construct positive and negative samples around gold triggers for contrastive learning, thereby addressing the overfitting problem in new-class event learning. Finally, we incorporated hierarchical distillation techniques to alleviate the model's forgetting issues.

Our experimental results demonstrate that our approach excels in class-incremental learning tasks related to interactive event detection. However, when tested in more complex environments, such as the ACE dataset, we observe that the recall rate drops below 90% after several rounds of class-incremental learning. This indicates a significant gap between our model's performance and the high precision requirements needed for effective application.

Future work will focus on addressing these challenges by enhancing the model's ability to handle diverse and complex scenarios, such as the subjunctive mood, multiple negation, and so on. This will involve refining our approach to better manage continuous class-incremental learning, exploring advanced strategies to further reduce forgetting, and optimizing performance in more challenging environments. In addition, we will also focus on another common scenario in incremental learning: sample increment, aiming to investigate how to address issues related to sample distribution shift and missing data. By advancing these, our goal is to enhance the efficiency of the interactive event-detection mode in real-world applications, thereby establishing a positive feedback loop.

Author Contributions: Conceptualization, X.Z. and J.D.; methodology, J.D. and X.Z.; validation, J.D.; investigation, J.D. and X.Z.; resources, X.Z.; data curation, J.D.; writing—original draft preparation, J.D.; writing—review and editing, X.Z.; visualization, J.D.; supervision, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under the grants of NO. 62102431, the National University of Defense Technology under the grant of NO. ZK21-32, and Science and Technology on Information Systems Engineering Laboratory under the grant of NO. 6142101220209.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study have been cited and referenced [36]. They are available at https://ai.baidu.com/broad/introduction (accessed on 18 August 2024). Other data are available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Duan, J.; Zhang, X. Chinese Event Trigger Recommendation Model for High-Accuracy Applications. *Preprints* **2024**, 2024091129. [CrossRef]
2. Cao, P.; Chen, Y.; Zhao, J.; Wang, T. Incremental Event Detection via Knowledge Consolidation Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing 2020, Online, 16–20 November 2020; pp. 707–717. [CrossRef]
3. Yu, P.; Ji, H.; Natarajan, P. Lifelong Event Detection with Knowledge Transfer. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021.
4. Liu, M.; Chang, S.; Huang, L. Incremental Prompting: Episodic Memory Prompt for Lifelong Event Detection. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2157–2165. [CrossRef]
5. Liu, M.; Huang, L. Teamwork Is Not Always Good: An Empirical Study of Classifier Drift in Class-incremental Information Extraction. *arXiv* **2023**, arXiv:2305.16559.
6. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
7. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. Event Extraction Via Dynamic Multi-Pooling Convolutional Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; Volume P15-1, pp. 167–176.
8. Wang, S.; Yu, M.; Chang, S.; Sun, L.; Huang, L. Query and Extract: Refining Event Extraction as Type-oriented Binary Decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 169–182. [CrossRef]
9. Wang, S.; Yu, M.; Huang, L. The Art of Prompting: Event Detection based on Type Specific Prompts. *arXiv* **2022**, arXiv:2204.07241.
10. Guan, Y.; Chen, J.; Lecue, F.; Pan, J.; Li, J.; Li, R. Trigger-Argument based Explanation for Event Detection. In *Findings of the Association for Computational Linguistics: ACL*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023.
11. Nateras, L.G.; Dernoncourt, F.; Nguyen, T. Hybrid Knowledge Transfer for Improved Cross-Lingual Event Detection via Hierarchical Sample Selection. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, USA, 9–14 July 2023.
12. Yue, Z.; Zeng, H.; Lan, M.; Ji, H.; Wang, D. Zero- and Few-Shot Event Detection via Prompt-Based Meta Learning. *arXiv* **2023**, arXiv:2305.17373.
13. Liu, J.; Sui, D.; Liu, K.; Liu, H.; Zhao, Z. Learning with Partial Annotations for Event Detection. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, USA, 9–14 July 2023.

14. Xu, C.; Zeng, Z.; Duan, J.; Qi, Q.; Zhang, X. Extracting Events Using Spans Enhanced with Trigger-Argument Interaction. In Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering, Fuzhou, China, 17–19 November 2023. [CrossRef]
15. Nguyen, T.H.; Cho, K.; Grishman, R. *Joint Event Extraction via Recurrent Neural Networks*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016.
16. Wei, Y.; Liu, S.; Lv, J.; Xi, X.; Yan, H.; Ye, W.; Mo, T.; Yang, F.; Wan, G. DESED: Dialogue-based Explanation for Sentence-level Event Detection. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2483–2493.
17. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint Extraction Of Entities And Relations Based on a Novel Tagging Scheme. *arXiv* **2017**, arXiv:1706.05075.
18. Tian, C.; Zhao, Y.; Ren, L. A Chinese Event Relation Extraction Model Based on BERT. In Proceedings of the 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 25–28 May 2019.
19. Cao, H.; Li, J.; Su, F.; Li, F.; Fei, H.; Wu, S.; Li, B.; Zhao, L.; Ji, D. OneEE: A One-Stage Framework for Fast Overlapping and Nested Event Extraction. *arXiv* **2022**, arXiv:2209.02693.
20. Li, H.; Mo, T.; Fan, H.; Wang, J.; Wang, J.; Zhang, F.; Li, W. KiPT: Knowledge-injected Prompt Tuning for Event Detection. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 1943–1952.
21. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. *Exploring Pre-Trained Language Models for Event Extraction and Generation*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume P19-1, pp. 5284–5294.
22. Xinya, D.; Claire, C. *Event Extraction by Answering (Almost) Natural Questions*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 671–683.
23. Yunmo, C.; Tongfei, C.; Seth, E.; Benjamin, V.D. Reading the Manual: Event Extraction as Definition Comprehension. *arXiv* **2020**, arXiv:1912.01586.
24. Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; Zhu, Y. *Event Extraction as Multi-Turn Question Answering*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 829–838.
25. Wan, X.; Mao, Y.; Qi, R. Chinese Event Detection without Triggers Based on Dual Attention. *Appl. Sci.* **2023**, *13*, 4523. [CrossRef]
26. Yan, Y.; Liu, Z.; Gao, F.; Gu, J. Type Hierarchy Enhanced Event Detection without Triggers. *Appl. Sci.* **2023**, *13*, 2296. [CrossRef]
27. Li, Z.; Hoiem, D. Learning Without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef] [PubMed]
28. D'Autume, C.D.M.; Ruder, S.; Kong, L.; Yogatama, D. Episodic Memory In Lifelong Language Learning. In Proceedings of the Advances in Neural Information Processing Systems 32 (NIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 13122–13131.
29. Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. Icarl: Incremental Classifier and Representation Learning. In Proceedings of the Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 5533–5542. [CrossRef]
30. Lopez-Paz, D.; Ranzato, M.A. Gradient Episodic Memory For Continual Learning. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; p. 30.
31. Mallya, A.; Lazebnik, S. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7765–7773.
32. Serrà, J.; Surís, D.; Miron, M.; Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In Proceedings of the International Conference on Machine Learning (ICML 2018), Stockholm, Sweden, 10–15 July 2018; Volume 80.
33. Cao, Y.; Peng, H.; Wu, J.; Dou, Y.; Li, J.; Yu, P.S. Knowledge-Preserving Incremental Social Event Detection via Heterogeneous GNNs. In Proceedings of the WWW '21: The Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 3383–3395.
34. Wei, K.; Zhang, Z.; Jin, L.; Guo, Z.; Li, S.; Wang, W.; Lv, J. HEFT: A History-Enhanced Feature Transfer framework for incremental event detection. *Knowl.-Based Syst.* **2022**, *254*, 109601. [CrossRef]
35. Luo, R.; Xu, J.; Zhang, Y.; Ren, X.; Sun, X. PKUSEG: A Toolkit for Multi-Domain Chinese Word Segmentation. *arXiv* **2019**, arXiv:1906.11455.
36. Li, X.; Li, F.; Pan, L.; Chen, Y.; Peng, W.; Wang, Q.; Lyu, Y.; Zhu, Y. DuEE: A Large-Scale Dataset for Chinese Event Extraction in Real-World Scenarios. In *Natural Language Processing and Chinese Computing*; Springer: Cham, Switzerland, 2020.
37. Kwon, H. Adversarial Image Perturbations with Distortions Weighted by Color on Deep Neural Networks. *Multimed. Tools Appl.* **2022**, *82*, 13779–13795. [CrossRef]