# Event Pattern-Instance Graph: A Multi-Round Role Representation Learning Strategy for Document-Level Event Argument Extraction

**Qizhi Wan[1,2], Tao Liu[1], Changxuan Wan[1,2,*], Rong Hu[1,2,*], Keli Xiao[3], Yuxin Shuai[1]**
[1]School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics
[2]Jiangxi Key Lab of Data and Knowledge Engineering
[3]College of Business, Stony Brook University
wanqizhi@jxufe.edu.cn, lt119517@163.com, wanchangxuan@263.net

## Abstract

For document-level event argument extraction, existing role-based span selection strategies suffer from several limitations: (1) ignoring interrelations among arguments within an event instance; (2) relying on pre-trained language models to capture role semantics at either the event pattern or document, without leveraging pattern-instance associations. To address these limitations, this paper proposes a multi-round role representation learning strategy. First, we construct an event pattern-instance graph (EPIG) to comprehensively capture the role semantics embedded in various direct and indirect associations, including those among roles within event patterns, arguments within event instances, and the alignments between patterns and instances. Second, to enhance the learning of role node representation in the graph, we optimize the update mechanisms for both node and edge representations in the EPIG graph. By leveraging the graph attention network, we iteratively update the representations of role nodes and role edges. The role representations learned from the EPIG are then integrated into the original role representations, further enriching their semantic information. Finally, a role representation memory module and a multi-round learning strategy is proposed to retain and refine role representations learned from previously analyzed documents. This memory mechanism enhances the prediction performance in subsequent rounds of span selection. Extensive experiments on three datasets verify the effectiveness of the model.

## 1 Introduction

Document-level event argument extraction (EAE) aims to identify event-related arguments and their roles within a document (Wan et al., 2024a; Hu et al., 2025). In recent years, scholars leverage pre-trained language models (PLMs) and prompt tuning to predict span boundaries directly. A key challenge is to capture semantic dependencies between sentences or events to accurately identify arguments. For instance, Ma et al. (2022) enhanced argument extraction performance by constructing event-type-specific joint prompts and leveraging PLMs' cross-attention to dynamically fuse context with structured prompts, generating context-sensitive role representations while capturing implicit role semantics within the same event type.

Subsequent works enriched role semantics through diverse strategies. Nguyen et al. (2023) fused document-contextualized representations via event type-document graphs and soft prompts. Li et al. (2023) established dual dependency graphs (internal role interactions and external correlations with similar events) to strengthen relational reasoning. For multi-event scenarios, He et al. (2023) employed correlation-aware decoder inputs to capture cross-event associations implicitly, and Liu et al. (2024) concatenated full event-type prompts with documents, using attention weights to inject event context into role representations. Contrastingly, Zhang et al. (2024) adopted hyper-spherical prototypes, optimizing span selection through prototype matching and distance-based loss computation. These approaches collectively advance argument extraction by systematically modeling semantic dependencies across roles, events, and an document.

Despite the impressive results achieved, at the pattern level, they do not fully utilize information such as event types and roles and learn role representation based on graph mechanism. At the instance level, only capturing semantics from the document directly or indirectly updates the role representation, or uses the prediction span to optimize the loss function, which indirectly affects the span selection and does not directly affect the span

* Corresponding Author.



*Findings of the Association for Computational Linguistics: ACL 2025*, pages 1865–1877

selection by updating the role representation.

To address these limitations, this paper constructs an event pattern-instance graph (EPIG) that comprehensively captures the role semantics embedded in various direct and indirect associations, including those among roles within event patterns, arguments within event instances, and the alignments between patterns and instances. For better understanding, Figure 1 illustrates an example of the EPIG in this paper.

Given a specific document segment at the top of Figure 1, the event type *conflict.attack.bombing* is known. This allows for the explicit identification of all associated roles, including *attacker*, *target*, *instrument*, and *place*. Consequently, edges can be established between the event type and its corresponding roles, forming the event pattern level association graph shown in the bottom right part of Figure 1. Additionally, by constructing edges between trigger and predicted span nodes, where the edge type corresponds to the role played by the span, we obtain the event instance level association graph shown in the top right part of Figure 1.

At the initial stage of graph construction, all edges and nodes have well-defined content, except for the span nodes, which remain temporarily empty. Once the predicted spans are obtained, we populate the span nodes with the predicted information. Finally, we connect the span nodes in the event instance graph to their corresponding role nodes in the event pattern graph, while also linking the trigger to the event type, thereby forming a complete event pattern-instance graph. This dual-layer structure not only captures the structural semantics among spans within an event but also fully leverages the associations between event types, triggers, and roles, enabling a richer representation of role semantics.

In summary, the EPIG is a heterogeneous graph centered around roles, with the aim of updating role node representations and role edge representations. To better encode the EPIG, we refine the node and edge update strategies in the graph neural network, drawing inspiration from the graph update mechanisms proposed by Cui et al. (2020) and Wan et al. (2024c). By leveraging multiple iterative interactions between nodes and edges, our approach enables a deeper integration of structural information within the graph.

Given that spans in the EPIG are obtained through prediction, we introduce a multi-round iterative prediction strategy to improve the accuracy of predicted spans. This strategy allows the model to incorporate more relevant information, continuously updating and refining span predictions to progressively approach the correct results.

Furthermore, we design a memory unit for each role representation. During training, the memory unit is continuously updated with role representation knowledge learned from previous training samples. This mechanism enhances the model's generalization ability, enabling it to better handle new and unseen event instances.

To model the above ideas, this paper proposes a Document-level Event Extraction (EPIG-EAE) model based on the Event Pattern-Instance Graph, which aims to learn role representations through multiple interactions from role associations of event patterns, span associations of event instances, and pattern-instance matching associations. The main contributions of this paper are as follows.

- We design a dual-layer Event Pattern-Instance Graph (EPIG) to capture role semantics via pattern-level role associations, instance-level span connections, and cross-level role-span mappings, enabling the extraction of latent role semantics from the graph.

- We enhance the update mechanism for node and edge representations in the graph with Graph Attention Networks (GAT), effectively captures structural semantics among nodes and enabling iterative semantic fusion between node and edge representations.

- We design a role representation memory unit to store and integrate role representations learned from each document. In addition, we propose a multi-round iterative span prediction strategy, where role representations are continuously optimized in each iteration, leading to progressively improved span selection.

- We validate model effectiveness on three datasets and analyze the impacts of graph structures, update mechanisms, and iteration rounds on extraction performance. The experimental code can be accessed at `https://github.com/jc-noss/EPIGEAE`.

## 2 Event Pattern-Instance Graph

This section details the motivation and procedure for constructing the event pattern-instance graph .

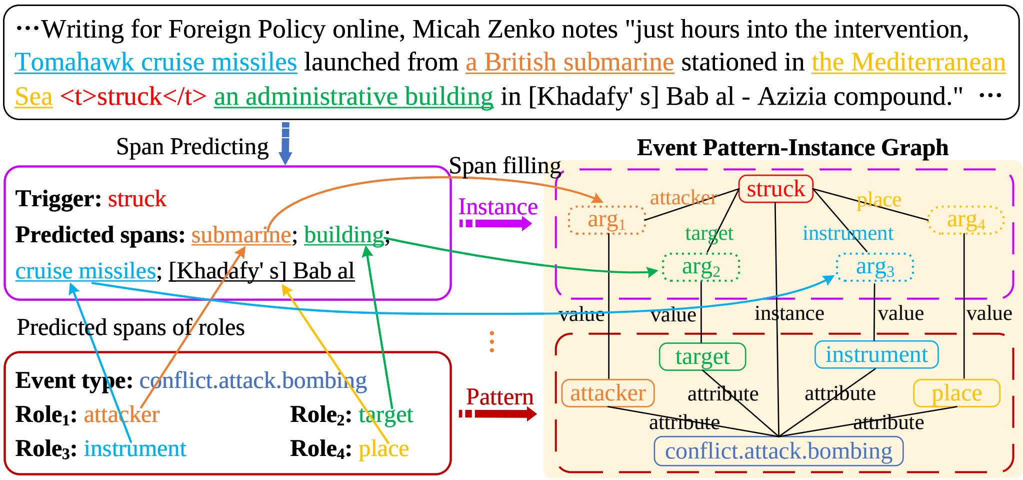



Figure 1: Example of the event pattern-instance graph. The top is a gold document, where red indicates the trigger and the other colors with underlines indicate the gold arguments corresponding to different roles. The purple box on the left denotes the event instance information, containing the trigger and the prediction span. In this case, the black underlined content indicates that the prediction span does not match the gold span. The red box is the event pattern information. On the right side is the event pattern-instance graph constructed based on event patterns and event instances. The dotted dashed boxes are span nodes (filled using predicted spans), the other boxes are nodes of other elements related to the event, and role edges and role nodes match their corresponding predicted span colors.

**Event Instance Graph.** Traditional methods treat arguments as independent units, ignoring the interrelationships among arguments within event instances. In this paper, an event instance refers to a predicted event triggered by a given trigger, consisting of the trigger and predicted spans for each role. By linking trigger node to predicted span nodes, we construct event instance level connections with edge types corresponding to the roles assigned to the spans.

This approach effectively captures the state characteristics of the event triggered by the trigger, while also encoding both the direct semantic associations between the trigger and its corresponding event arguments (spans) and the indirect semantic associations among arguments (spans) within the same event. Considering that initial predicted spans may contain errors, directly establishing connections between spans could introduce semantic interference. Thus, we only establish edges between trigger and spans, simplifying the structure and enhancing robustness.

**Event Pattern Graph.** Traditional methods at the pattern level do not fully utilize event types and roles to model relationships among roles. Thus, we construct a pattern graph to capture both direct semantic associations (between event type and their roles) and indirect associations (among roles within the same event type).

By establishing edges between event type nodes and corresponding role nodes, it characterizes the role features of the event types, helping the model understand how specific events conform to general event patterns. Additionally, considering semantic differences among roles under the same event type, not directly linking role nodes can reduce direct semantic interference between different roles while still conveying indirect associations through the event types.

**Matching Associations Between Subgraphs.** Given that existing work does not combine pattern level and instance level information to enrich role representations semantically, this paper links these two levels to capture matching associations between patterns and instances. First, we establish edges between event type nodes and trigger nodes; then, we connect role nodes to their corresponding span nodes. Given the bidirectional flow of information and lack of clear directionality among nodes, undirected edges are used within the graph. Detailed construction procedure of EPIG is provided in Appendix A.

## 3 Methodology

This section introduces the framework of our EPIG-EAE model, as illustrated in Figure 2. Following previous studies (Ma et al., 2022; Zhang et al., 2024), this work adopts a role-based span prediction strategy, where the task is to predict the start and end positions of spans for each role associ-

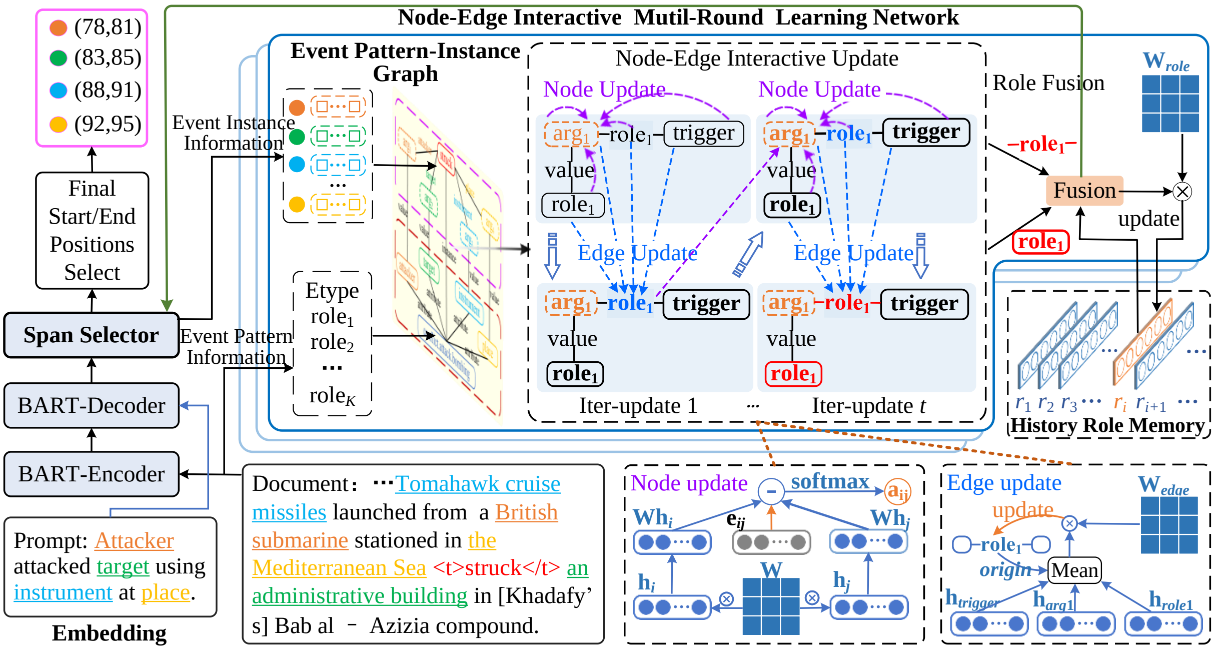


Figure 2: Overall architecture of EPIG-EAE. The middle dashed line highlights the interaction and update between nodes and edges. Thick blue arrows represent the update process, including node and edge revisions; bold $\text{arg}_1$, $\text{role}_i$, and trigger denote updated nodes and edges. For clarity, only the update of the $\text{arg}_1$ node is illustrated.

ated with a given event. The EPIG-EAE comprises six component: **Embedding layers**, initializing embeddings for documents and roles in prompts; **Preliminary Span Selection**, generating preliminary predicted spans for each role to fill the content for the span nodes in the EPIG; **Event Pattern-Instance Graph**, constructing the EPIG graph and integrating the preliminary predicted spans into corresponding span nodes; **Node-Edge Interaction-based Multi-round Learning Network**, implementing a multi-round learning strategy for iteratively interactively updating representations of nodes and edges in the EPIG; **Historical Role Memory**, designing a memory unit for each role to store fused representations learned across all documents; **Argument Extraction**, obtaining the spans of roles according to the learned role representations.

### 3.1 Embedding Layers

Given a document $d$, we utilize the pretrained language model BART to obtain the document representation $\mathbf{H}_d$. For role representation, the document representation $\mathbf{H}_d^{\text{En}}$ output by the BART encoder is jointly fed into the BART decoder with the prompt, yielding the prompt representation $\mathbf{H}_P$:

$$\mathbf{H}_d = \text{Encoder-Decoder}(d), \tag{1}$$

$$\mathbf{H}_P = \text{Decoder}(P, \mathbf{H}_d^{\text{En}}), \tag{2}$$

where $P$ denotes the prompt corresponding to the given event type.

### 3.2 Preliminary Span Selection

Based on $\mathbf{H}_P$, we first obtain the representation $\mathbf{h}_{r_k}$ of the $k$-th role $r_k$. Then, preliminary predicted spans are derived from $\mathbf{h}_{r_k}$. To enhance span prediction performance, a multi-round role representation learning strategy is employed to iteratively optimize role representations. The updated role representation $\mathbf{h}_{r_k}^u$ after the $u$-th learning round is used to generate new predicted spans.

$$\Phi_{\text{start}}^{r_k} = \mathbf{h}_{r_k}^u \mathbf{W}_{\text{start}}, \quad \Phi_{\text{end}}^{r_k} = \mathbf{h}_{r_k}^u \mathbf{W}_{\text{end}} \tag{3}$$

$$p_{\text{start}}^{r_k} = \text{softmax}\left(\Phi_{\text{start}}^{r_k} \mathbf{H}_d\right) \tag{4}$$

$$p_{\text{end}}^{r_k} = \text{softmax}\left(\Phi_{\text{end}}^{r_k} \mathbf{H}_d\right) \tag{5}$$

$\mathbf{W}_{\text{start}}$ and $\mathbf{W}_{\text{end}}$ are learnable parameters, $p_{\text{start}}^{r_k}$ and $p_{\text{end}}^{r_k}$ represent the probability distributions of the start and end positions for the predicted span of role $r_k$ across the entire context, and $u \in U$ denotes the number of learning rounds. When $u = 1$, $\mathbf{h}_{r_k}^u$ is $\mathbf{h}_{r_k}$; $u > 1$, $\mathbf{h}_{r_k}^u$ is computed via Equation (11).

### 3.3 Event Pattern-Instance Graph

Following the strategy described in Section 2, we construct the structure of the event pattern-instance graph. This module populates the corresponding span nodes with the predicted spans. In addition to the span nodes, other nodes and edges in the graph are initialized with the BERT to obtain their embeddings. The representation of the span node for role $r_k$ at the $u$-th prediction round is:

$$\mathbf{h}^{u}_{\text{arg}_k} = \text{Mean}\left(\sum_{j=\text{start}_{r_k}}^{\text{end}_{r_k}} \mathbf{h}^{u}_{j}\right), \tag{6}$$

where $\text{Mean}(\cdot)$ denotes the average operation, $\text{start}_{r_k}$ and $\text{end}_{r_k}$ represent the start and end positions of the predicted span for role $r_k$, and $\mathbf{h}^u_j$ is the representation of the $j$-th token within the span.

### 3.4 Node-Edge Interactive Multi-round Learning Network

Inspired by the graph update strategies proposed by Cui et al. (2020) and Wan et al. (2024c), this paper optimizes the mechanisms for node and edge updates within the graph. Cui et al. (2020) aggregating node representations via graph convolutional networks while updating edge representations through learnable weight matrices. However, their approach neglects graph-based edge representation updates. In contrast, Wan et al. (2024c) transforms dependency edges into nodes and updates cross-graph node interactions using graph neural networks, explicitly learning edge representations.

Our EPIG diverges by uniquely integrating role nodes and edges to enhance semantic role representations (see Figure 2). For node updates via Graph Attention Networks, edges are treated as virtual nodes during propagation (e.g., the $\text{arg}_1$ update in Figure 2). Specifically, edge representations are concatenated with neighboring node representations. The attention coefficient between the $i$-th and $j$-th nodes and the updated representation of the $i$-th node are formalized in Equations (7)∼(9):

$$\alpha^{*}_{ij} = \text{LeakyReLU}\left(\mathbf{V}^{\mathrm{T}}[\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j \parallel \mathbf{h}_q]\right), \tag{7}$$

$$\alpha_{ij} = \text{dropout}\left(\frac{\exp(\alpha^{*}_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(\alpha^{*}_{ik})}\right), \tag{8}$$

$$\mathbf{h}^{(l)}_{i} = \sigma\Big(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in\mathcal{C}_i}\alpha^{m(l)}_{ij}\mathbf{W}^{m(l)}\mathbf{h}^{(l-1)}_{j}\Big), \tag{9}$$

where $\mathbf{V}$ is the attention vector, $\text{LeakyReLU}(\cdot)$ denotes the activation function, $\mathbf{h}_q$ represents the edge embedding between the $i$-th and $j$-th nodes, The computation involves Softmax normalization over the neighborhood $\mathcal{N}_i$ to obtain probabilistic attention weights $\alpha_{ij}$, followed by dropout regularization to prevent overfitting, $M$ is the number of attention heads, $\mathcal{C}_i$ is the neighbor set of the $i$-th node, and $l \in L$ indexes the GAT layer.

After updating node representations, we fuse edge representations by integrating information from role nodes, trigger node, updated span nodes, and the edges themselves. This ensures comprehensive consideration of semantic relationships. The updated representation of the $k$-th role edge at the $u$-th round and $l$-th layer, $\mathbf{h}^{u(l)}_{e_k}$, is defined as:

$$\mathbf{h}^{u(l)}_{e_k} = \text{Mean}\left(\mathbf{h}^{u(l-1)}_{e_k}, \mathbf{h}^{u(l)}_{\text{arg}_k}, \mathbf{h}^{u(l)}_{\text{tri}}, \mathbf{h}^{u(l)}_{r_k}\right)\mathbf{W}_e, \tag{10}$$

where $\mathbf{W}_e$ denotes the edge weight matrix, $\mathbf{h}^{u(l)}_{\text{arg}_k}$ and $\mathbf{h}^{u(l)}_{r_k}$ represent the span and role node representation of the $k$-th role at the $l$-th layer of the $u$-th round, respectively. $\mathbf{h}^{u(l)}_{\text{tri}}$ refers to the trigger node representation at the $l$-th layer of the $u$-th round. Through average fusion, all relevant semantic features are integrated into the new edge representations. In contrast, Cui et al. (2020) used a fixed transformation matrix and connected node representations, which cannot fully capture context-specific information.

During the updating process from layer 1 to layer $L$, node representations and edge representations influence each other, continuously generating the latest role node representations and role edge representations. At the $U$-th round and the $L$-th layer, we fuse the role node representation with the role edge representation to obtain the final role representation $\mathbf{h}^{\mathrm{g}}_{r_k}$ output by the multi-round learning network, as shown in Equation (11).

$$\mathbf{h}^{\mathrm{g}}_{r_k} = \text{Mean}\left(\mathbf{h}^{U(L)}_{r_k}, \mathbf{h}^{U(L)}_{e_k}\right) \tag{11}$$

### 3.5 Historical Role Memory

To retain the learning content of each role across the corpus, we designed a memory unit for each role that integrates and updates during training, reflecting the latest learning outcomes. The model fuses the current historical role representation, the role representation from the multi-round learning network, and the role representation in the prompt to form a richer role representation. This aids in generalizing to unseen roles or event types by providing additional contextual information.

For role $r_k$, assuming the representation in the memory unit is initially empty ($\mathbf{h}^{\mathrm{m}}_{r_k}$), this paper fuses the current (at the $u$-th round) historical role representation $\mathbf{h}^{\mathrm{m}(u)}_{r_k}$, the role representation output by the multi-round learning network at the $u$-th round $\mathbf{h}^{\mathrm{g}(u)}_{r_k}$, and the role representation in the prompt $\mathbf{h}_{r_k}$ to obtain the fused role representation. The fused role representation at the $u$-th round is:

$$\mathbf{h}'^{u}_{r_k} = \text{Mean}(\mathbf{h}^{\text{m}(u)}_{r_k}, \mathbf{h}^{\text{g}(u)}_{r_k}, \mathbf{W}_r\mathbf{h}_{r_k}), \quad (12)$$

where $\mathbf{W}_r$ is the historical role learning matrix.

### 3.6 Argument Extraction

After $U$ rounds of learning, we obtain the final role representation $\mathbf{h}^u_r$ enriched with semantic information. Using the span selection strategy described in Section 3.2, we predict the start and end positions of argument spans. Following Ma et al. (2022), we adopt the bipartite matching loss for training, detailed formula in our model is formalized as shown in Equation (13):

$$\mathcal{L} = \sum_{i=1}^{D} \sum_{(s,e)} \left( \log p^{r_k}_{\text{start}} s_{r_k} + \log p^{Rr_k}_{\text{end}} e_{r_k} \right), \quad (13)$$

where $D$ denotes the number of training documents, $(s, e)$ represents the start-end position pairs for the golden argument span distribution set of the $i$-th document. $s_{r_k}$ and $e_{r_k}$ are the golden start and end positions of the $k$-th role.

## 4 Experiments and Results

### 4.1 Experimental Setup

This paper evaluated the model on three used widely datasets, including RAMS (Ebner et al., 2020), WikiEvents (Li et al., 2021) and OEE-CFC (Wan et al., 2024b). Detailed statistics are listed in Appendix B.1. Evaluation metrics refer to Appendix B.2 for details of implementation. To comprehensively evaluate our model, we compared it with many recent advanced baselines, and the details are reported in Appendix B.3.

### 4.2 Overall Performance

As shown in Table 1, our model achieves superior Arg-C scores on the RAMS and WIKIEVENTS using both Base and Large scale PLMs. Additionally, results in Table 2 demonstrate that our model surpasses baselines in Span and Head metrics on the Chinese OEE-CFC dataset, validating its effectiveness. The primary reasons for this are analyzed as follows:

**Construction of EPIG.** Building EPIG allows the model to capture structural semantics between event types and roles at the pattern level, trigger and arguments at the instance level. This structure integrates matching relationships between concrete instances and abstract patterns, helping identify subtle differences in complex events. This approach provides a more comprehensive understanding of events by combining multiple levels of information. Experimental investigations on the graph structure in Table 3 validate the effectiveness of EPIG.

**Interactive Update Mechanism.** Dynamically adjusting node and edge representations can better reflects event semantic changes, The model optimizes its understanding of role relationships during each update, capturing complex interactions more precisely and enhancing role correlation modeling. Experimental results on different update mechanisms of graphs in Table 4 prove its effectiveness.

**Role Memory Fusion.** If similar events or roles were processed during training, historical role representation can help identify anomalies in current predictions, improving adaptability. By integrating static role semantics from original prompts, dynamically updated role semantics from graph updates, and historical role semantics, the model enriches each role's representation, capturing relational structures among roles more effectively. The role ablation results in Table 5 validate the importance of various role fusions.

**Multi-round Role Learning.** This method allows the model to self-correct and optimize based on previous round results when handling long-text event argument extraction tasks. This progressive optimization process facilitates the model gradually approach accurate span boundaries, using updated learning outcomes to refine predictions, ultimately achieving higher accuracy. Figure 5 demonstrates the effects of different rounds of role learning, verifying the significance of multi-round role learning.

## 5 Analysis and Discussion

### 5.1 Ablation of Edges in EPIG

This section aims to investigate the impact of edges constructed in the event pattern-instance graph (EPIG) on model performance and identify which types of edges are most critical for capturing semantic relationships within events. Figure 3 (a) illustrates four ablation examples targeting EPIG edges. Specifically, $\text{eg}_1$–$\text{eg}_4$ correspond to removing edges between: 1) trigger nodes and predicted span nodes, 2) event type nodes and role nodes, 3) trigger nodes and event type nodes, 4) span nodes and their corresponding role nodes.

| Model | PLM | RAMS | | WIKIEVENTS | |
|---|---|---|---|---|---|
| | | Arg-I | Arg-C | Arg-I | Arg-C |
| TSAR(2022)* | RoBERTa-l | - | 51.2 | 71.1 | 65.8 |
| PAIE(2022) | BART-b | 54.7 | 49.5 | 68.9 | 63.4 |
| | BART-l | 56.8 | 52.2 | 70.5 | 65.3 |
| TabEAE(2023) | RoBERTa-l | 57.3 | 52.7 | 71.4 | 66.5 |
| SCPRG(2023)* | BERT-b | 53.9 | 48.9 | 70.1 | 65.8 |
| | Roberta-l | 56.7 | 52.3 | 71.3 | 66.4 |
| SPEAE(2023) | BART-b | 56.0 | 51.1 | 70.6 | 66.2 |
| | BART-l | 58.0 | 53.3 | 71.9 | 66.1 |
| EACE(2023) | BART-b | **58.4** | 50.1 | 71.1 | 66.2 |
| DEEIA(2024) | RoBERTa-l | 58.0 | 53.4 | 71.8 | 67.0 |
| HMPEAE(2024) | RoBERTa-l | <u>**58.6**</u> | 53.7 | 72.1 | 66.6 |
| EPIG-EAE(Ours) | BART-b | 56.7 | **52.1** | **72.0** | **66.3** |
| | BART-l | 58.1 | <u>**53.7**</u> | <u>**72.2**</u> | <u>**67.2**</u> |

Table 1: Overall results on RAMS and WIKIEVENTS. $*$ indicates the results from Liu et al. (2024), Other results are taken from the original paper. Bold denotes the highest scores by base-scale PLMs, and underlined indicates the best results by large-scale PLMs.

| Model | Span | | Head | |
|---|---|---|---|---|
| | Arg-I | Arg-C | Arg-I | Arg-C |
| PAIE(2022)♠ | 63.22 | 58.88 | 76.17 | 69.75 |
| EPIG-EAE(Ours) | **63.49** | **60.34** | **76.22** | **71.50** |

Table 2: Overall results on OEE-CFC using the BART-base. ♠ indicates that we rerun the code on the dataset.

As shown in Table 3, removing trigger-span edges (line 2) results in the most significant performance drop (Arg-C: 50.85%), as these edges inherently encode role semantics that directly affect subsequent edge-node interaction updates. Eliminating other edges reduces Arg-C by 0.31–1.25 percentage points, demonstrating that all edges in EPIG contribute to enriching role semantics.

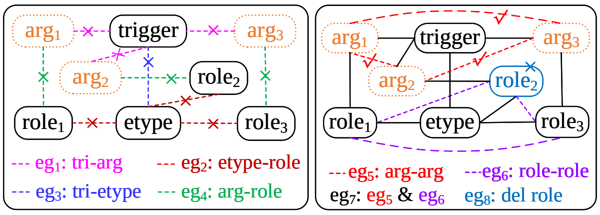
### 5.2 Other Graph Structure

To further explore the impact of alternative graph structures on model performance, this study constructs four graph variants, see Figure 3 (b). $eg_5$ and $eg_6$ add edges between span nodes and role nodes, respectively; $eg_7$ connects span and role nodes; $eg_8$ changes the edge-linking strategy between span and role nodes by retaining only one role node linked to multiple span nodes when arguments share the same role under an event type.

Based on these structures, Table 3 (line 6-9) presents experimental results. Adjusted graph construction methods underperform EPIG, with Arg-C decreasing by 0.48–1.66 percentage points. The first three edge-addition variants introduce excessive noise and redundant information, disrupting the original graph balance and hindering meaningful relationship identification. The edge-pruned variant ($eg_8$) fails to capture semantic differences between arguments of the same role (similar to role prototypes in Zhang et al. (2024)), leading to performance decline.

### 5.3 Graph Update Mechanism

This section analyzes the effects of different node/edge representation update mechanisms in EPIG, as shown in Table 4. When excluding role nodes during edge updates (line 2), Arg-C drops to 51.70%, indicating that role nodes propagate event-type-specific knowledge and contextual semantics during edge updates, which are irreplaceable by trigger nodes, span nodes, or edges alone. Removing role edges during node updates (line 3) reduces Arg-C by 1.19 percentage points, demonstrating that the node update mechanism captures rich semantic associations and enhances understanding of complex node interactions through adaptive relationship weighting. Disabling edge updates entirely (line 4) causes performance droping, indicating that edge updates contribute to modeling direct or latent relationships among trigger, spans, and roles.




(a) Edge ablation (b) Other graph structure

Figure 3: Edge ablation and other graph structures.

| Ablation and Other Graph | Arg-I | Arg-C |
|---|---|---|
| **EPIG** | **56.75** | **52.10** |
| w/o $eg_1$: tri-arg edge | 55.66 | 50.85 |
| w/o $eg_2$: etype-role edge | 56.10 | 51.06 |
| w/o $eg_3$: tri-etype edge | 56.33 | 51.79 |
| w/o $eg_4$: arg-role edge | 55.93 | 51.13 |
| w $eg_5$: arg-arg edge | 55.76 | 50.91 |
| w $eg_6$: role-role edge | 56.13 | 51.31 |
| w $eg_7$: $eg_5$ & $eg_6$ | 55.15 | 50.44 |
| w/o $eg_8$: del the same role | 56.46 | 51.62 |

Table 3: Performance of edge ablation and other graph structures on RAMS dataset.

### 5.4 Role Fusion Strategy

Table 5 evaluates the impact of role fusion strategies. Removing historical role fusion (line 2) results in minor performance decline, suggesting limited overall influence. Excluding role edge fusion (line 3) reduces Arg-C to 51.18%, demonstrating the significance of relational semantics embedded in role edges for modeling complex interactions. Similarly, omitting role node fusion (line 4) causes significant degradation, as role node fusion ensures comprehensive consideration of role-specific attributes, facilitating precise localization and global knowledge transfer. Removing both role node and edge fusion (line 5) leads to large performance collapse, further validating the necessity of integrated role representation fusion.

### 5.5 Number of Graph Update Iterations

While Section 5.3 examined whether node-edge interaction updates benefit argument extraction, this section evaluates the performance of graph update iteration counts, and the results in Figure 4 and Table 6 indicate that moderate iterations optimize semantic representations of nodes and edges. Both Arg-I and Arg-C achieve their peak performance (56.75% and 52.1%, respectively) at the 2nd iteration, and then declines progressively. Detailed analysis is reported in the Appendix C.1.

| Graph Update Mechanism | Arg-I | Arg-C |
|---|---|---|
| **EPIG-EAE** | **56.75** | **52.10** |
| w/o Role Node in Edge Update | 56.49 | 51.70 |
| w/o Role Edge in Node Updat | 55.82 | 50.91 |
| w/o Edge Update in Graph | 55.94 | 51.33 |

Table 4: Performance of different node and edge update mechanisms in EPIG.

| Role Fusion Strategy | Arg-I | Arg-C |
|---|---|---|
| **EPIG-EAE** | **56.75** | **52.10** |
| w/o Historical Role | 56.67 | 51.81 |
| w/o Role Edge | 56.09 | 51.18 |
| w/o Role Node | 56.18 | 51.36 |
| w/o Role Node & Role Edge | 54.97 | 50.22 |

Table 5: Performance of different role fusion strategies.

### 5.6 Number of Multi-Round Learning

Figure 5 and Table 7 explore the performance of the model with different learning rounds. The results show that the 1st round achieves satisfactory performance, while the 2nd round slightly improve accuracy through a reconstruction strategy. However, performance notably deteriorates after exceeding round 3. For instance, the reconstructed EPIG strategy yields 51.44% in Arg-C at round 3, with a slight rebound to 51.88% at round 4, while the original EPIG-based iteration strategy exhibits a more pronounced decline, reaching 51.07% in Arg-C at round 4. Detailed analysis is provided in the Appendix C.2.

## 6 Related Work

For event extraction, Wan et al. (2021) modeled structured semantics based on syntactic and semantic dependency parsing. Recent work (Wan et al., 2023a, 2024d) has further optimized document-level event association capture through multi-channel hierarchical graph attention networks and Token-Event-Role multi-channel structures, while (Wan et al., 2023b) proposed graph structure connects tokens via $eType\text{-}role_1\text{-}role_2$ edges to jointly model event types and argument roles within a unified framework.

For event argument extraction, traditional classification (Zhang et al., 2020; Yang et al., 2023; Ren

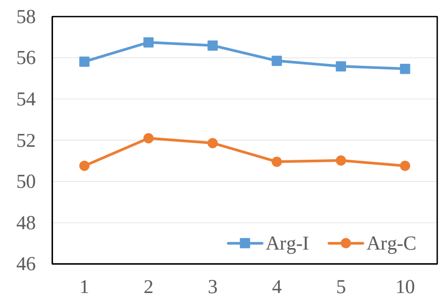
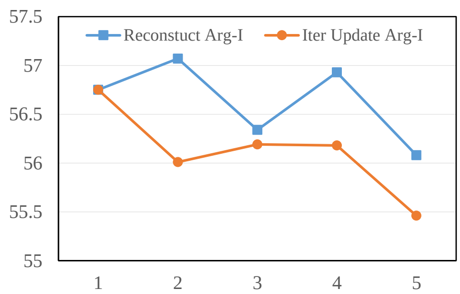


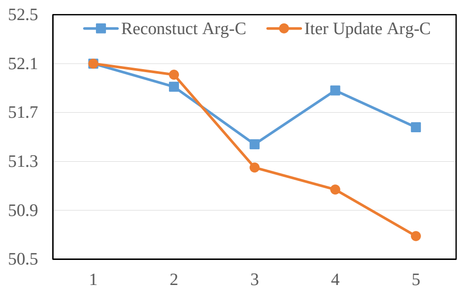

Figure 4: Performance with different update iterations counts of event pattern-instance graph.

| Iter | Arg-I | Arg-C |
|---|---|---|
| 1 | 55.82 | 50.76 |
| **2** | **56.75** | **52.10** |
| 3 | 56.60 | 51.87 |
| 4 | 55.86 | 50.96 |
| 5 | 55.59 | 51.02 |
| 10 | 55.47 | 50.76 |

Table 6: The specific data performance with different update iterations counts of event pattern-instance graph.




(a) Arg-I of two strategies




(b) Arg-C of two strategies

Figure 5: Performance under two strategies with different rounds of role learning.

| Round | Reconstruct graph | | Iter update graph | |
|---|---|---|---|---|
| | Arg-I | Arg-C | Arg-I | Arg-C |
| 1 | 56.75 | **52.10** | **56.75** | **52.10** |
| **2** | **57.07** | 51.91 | 56.01 | 52.01 |
| 3 | 56.34 | 51.44 | 56.19 | 51.25 |
| 4 | 56.93 | 51.88 | 56.18 | 51.07 |
| 5 | 56.08 | 51.58 | 55.46 | 50.69 |

Table 7: The specific data performance under two strategies with different rounds of role learning.

et al., 2023; Liu et al., 2023b; Shuang et al., 2024) and machine reading strategies (Li et al., 2020; Liu et al., 2021; Wei et al., 2021) struggle with large candidate span spaces or neglect semantic relationships between roles. In contrast, span selection strategies (Ma et al., 2022; He et al., 2023; Zhang et al., 2024; Liu et al., 2024; Zhou et al., 2024b; Wang et al., 2025), which utilize specific prompts based on PLMs to obtain role representations, have shown superior performance in argument extraction (Zhou et al., 2024a; Zhang et al., 2023; Liu et al., 2023a).

Based on this, Nguyen et al. (2023) constructed an event type-document graph to capture context semantics in a document. Li et al. (2023) modeled intra-event and inter-event role dependencies, providing internal role dependencies and clues with similar event types. He et al. (2023) designed prompts by creating multi-event correlation lists and used these as inputs of decoders, implicitly capturing semantic associations between roles across multiple event types by PLMs. On this basis, Liu et al. (2024) devised prompts to implicitly capture semantic associations across multiple event types or explicitly use document information for role semantics. Zhang et al. (2024) interpreted roles as multiple prototypes, matching spans with the best prototype and updating prototype representations.

## 7 Conclusion

This paper proposes a multi-round role representation learning strategy for document-level event argument extraction. The goal is to merge abstract event pattern with concrete event instance to better understand internal event structures. First, an event pattern-instance graph is constructed to capture semantics between event types and roles at the pattern level, as well as between trigger and arguments at the instance level. Then, to update the role representations in the graph, we design a node-edge interactive multi-round learning network, including the interaction update mechanisms of nodes and edges, along with a historical role memory strategy and a multi-round learning strategy. Extensive experiments are constructed on three datasets, and the results verify the effectiveness of the model.

## Limitations

For datasets with a single event type and a large number of roles (e.g., OEE-CFC), the "event type" node inherently lacks pattern level relevance. Including all role nodes in the EPIG introduces structural redundancy, which may propagate noise during updates, thereby disrupting the model's focus on critical roles. Furthermore, the model's performance across different datasets is sensitive to hyperparameter settings such as the number of graph update iterations and multi-round learning cycles. While these iterative updates enhance semantic refinement for complex datasets, they can lead to over-smoothing or error accumulation in simpler scenarios. Consequently, fine-tuning hyperparameters for each dataset is necessary.

## Ethics Statement

The data used in our study is sourced exclusively from publicly licensed corpora and does not involve the utilization of personally identifiable information.


## Acknowledgements

This research was partially supported by the Natural Science and Foundation of Jiangxi Province (20242BAB25119 and 20232ACB202008), the National Natural Science Foundation of China (62272205, 62272206, and 62462034), the Science and Technology Project of the Department of Education of Jiangxi Province (GJJ2400411).


## References


Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*, pages 2329–2339.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8057–8077.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can eae models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12542–12556.

Rong Hu, Changxuan Wan, Qizhi Wan, Dexi Liu, and Xiping Liu. 2025. Document-level event extraction based on deep learning: A survey. *Chinese Journal of Computers*, 48(2):381–516.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*, pages 829–838.

Hao Li, Yanan Cao, Yubing Ren, Fang Fang, Lanxue Zhang, Yingjie Li, and Shi Wang. 2023. Intra-event and inter-event dependency-aware graph network for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6362–6372.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725.

Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. 2023a. Document-level event argument extraction with a chain reasoning paradigm. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9570–9583.

Wanlong Liu, Shaohuan Cheng, Dingyi Zeng, and Hong Qu. 2023b. Enhancing document-level event argument extraction with contextual clues and role relevance. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12908–12922.

Wanlong Liu, Li Zhou, Dingyi Zeng, Yichen Xiao, Shaohuan Cheng, Chen Zhang, Grandee Lee, Malu Zhang, and Wenyu Chen. 2024. Beyond single-event extraction: Towards efficient document-level multi-event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9470–9487.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6759–6774.

Chien Nguyen, Hieu Man, and Thien Nguyen. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4352–4361.

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 293–306.

Kai Shuang, Zhouji Zhouji, Wang Qiwei, and Jinyu Guo. 2024. Thinking about how to extract: Energizing llms' emergence capabilities for document-level event argument extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5520–5532.

Qizhi Wan, Changxuan Wan, Rong Hu, and Dexi Liu. 2021. Chinese financial event extraction based on syntactic and semantic dependency parsing. *Chinese Journal of Computer*, 44(3):508–530.

Qizhi Wan, Changxuan Wan, Rong Hu, Dexi Liu, Xiping Liu, and Guoqiong Liao. 2024a. Event extraction based on deep learning: A survey of research issue. *ACTA Automatica Sinica*, 50(11):2079–2101.

Qizhi Wan, Changxuan Wan, Rong Hu, Dexi Liu, Xu Wenwu, Kang Xu, Zou Meihua, Liu Tao, Jie Yang, and Zhenwei Xiong. 2024b. OEE-CFC: A dataset for open event extraction from Chinese financial commentary. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4446–4459.

Qizhi Wan, Changxuan Wan, Keli Xiao, Rong Hu, and Dexi Liu. 2023a. A multi-channel hierarchical graph attention network for open event extraction. *ACM Transactions on Information Systems (TOIS)*, 41(1):1–27.

Qizhi Wan, Changxuan Wan, Keli Xiao, Dexi Liu, Chenliang Li, Bolong Zheng, Xiping Liu, and Rong Hu. 2023b. Joint document-level event extraction via token-token bidirectional event completed graph. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10481–10492.

Qizhi Wan, Changxuan Wan, Keli Xiao, Kun Lu, Chenliang Li, Xiping Liu, and Dexi Liu. 2024c. Dependency structure-enhanced graph attention networks for event detection. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, pages 19098–19106.

Qizhi Wan, Changxuan Wan, Keli Xiao, Hui Xiong, Dexi Liu, Xiping Liu, and Rong Hu. 2024d. Token-event-role structure-based multi-channel document-level event extraction. *ACM Transactions on Information Systems (TOIS)*, 42(4):1–27.

Guanghui Wang, Dexi Liu, Jian-Yun Nie, Qizhi Wan, Rong Hu, Xiping Liu, Wanlong Liu, and Jiaming Liu. 2025. Degap: Dual event-guided adaptive prefixes for templated-based event argument extraction with slot querying. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 7598–7613.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 4672–4682.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream amr-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 5025–5036.

Yuqing Yang, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. An amr-based link prediction approach for document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12876–12889.

Guangjun Zhang, Hu Zhang, Yujie Wang, Ru Li, Hongye Tan, and Jiye Liang. 2024. Hyperspherical multi-prototype with optimal transport for event argument extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9271–9284.

Kaihang Zhang, Kai Shuang, Xinyue Yang, Xuyang Yao, and Jinyu Guo. 2023. What is overlap knowledge in event argument extraction? APE: A cross-datasets transfer learning model for eae. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 393–409.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7479–7485.

Hanzhang Zhou, Junlang Qian, Zijian Feng, Lu Hui, Zixiao Zhu, and Kezhi Mao. 2024a. Llms learn task heuristics from demonstrations: A heuristic-driven prompting strategy for document-level event argument extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11972–11990.

Ji Zhou, Kai Shuang, Qiwei Wang, and Xuyang Yao. 2024b. EACE: A document-level event argument extraction model with argument constraint enhancement. *Information Processing and Management*, 61(1):103559.

## Appendix

## A Steps for Constructing EPIG

According to the construction strategy outlined in Section 2, the construction of the event pattern-instance graph for a given document $d$ includes the following three steps:

**Step 1: Construction of the Event Pattern Subgraph.** First, generate corresponding nodes for the event type involved in the current event contained in document $d$, with node content revealing the event type, as shown in the bottom right of Figure 1. At the same time, generate corresponding nodes for each role under that event type, with node content describing the role. Then, establish undirected edges between the event type node and its corresponding role nodes, with the edge type labeled as "attribute".

**Step 2: Construction of the Event Instance Subgraph.** First, based on the given event trigger, generate a trigger node. Simultaneously, according to the given event ontology, generate corresponding span nodes with empty content, to be filled after predicting spans. Then, establish edges between the trigger node and corresponding event span nodes, with the edge type being the corresponding role, represented by the words describing the role.

**Step 3: Construction of Matching Associations Between Subgraphs.** Establish "instance"-type edges between event type nodes and trigger nodes, and "value"-type edges between role nodes and their corresponding span nodes.

## B Experiments Setup

### B.1 Datasets

We evaluated the model on three datasets, including RAMS (Ebner et al., 2020), WikiEvents (Li et al., 2021) and OEE-CFC (Wan et al., 2024b). The detailed dataset description and statistical data are shown in Table 8. RAMS includes 139 event types and 65 argument roles; It contains 9,124 documents, and a document contains only one event, totaling 9,124 events. WIKIEVENTS is also an event extraction dataset derived from Wikipedia English articles, including 50 event types and 59 argument roles; There are 246 documents and 3,951 events in total. OEE-CFC is an open document-level event extraction dataset derived from Chinese financial reviews, including 1 event type and 21 argument roles. The dataset has 4,253 documents and 17,469 events.

### B.2 Evaluation Metrics

In the implementation process, this experiment is based on the open source code of Ma et al. (2022), and uses two indicators to evaluate the performance: (1) Argument identification $F1$ (Arg-I). If the boundary of the prediction span matches the boundary of any gold argument, it is considered correct. (2) The prediction span of argument classification $F1$ (Arg-C) is correct only when its boundary matches the boundary of a gold argument and its role is also the role of the gold argument.To comprehensively evaluate our model, we compared it with many recent advanced baselines, and details are reported in Appendix B.3.

### B.3 Baselines

In order to comprehensively evaluate our model, we compared it with more advanced baselines.

(1) **TSAR** (Xu et al., 2022) combined Abstract Meaning Representation (AMR) information to perform document-level event argument extraction.

(2) **PAIE** (Ma et al., 2022) proposes a role-based span selection strategy, which captures the role representation for each event type by designing a specific prompt for that event type.

(3) **SPEAE** (Nguyen et al., 2023) enrich the semantics of roles by capturing semantics from the document through soft prompts.

(4) **SCPRG** (Liu et al., 2023b) utilize the attention weights in PLM to aggregate context information relevant to the candidate spans, and capture the semantics between roles through role interaction encoding, thereby enriching the representation of candidate spans.

(5) **TabEAE** (He et al., 2023) propose a non-autoregressive generation framework to extract arguments for multiple events in a document in parallel, and use PLM to implicitly capture the associative semantics between multiple events within the document, thereby enriching the semantics of role representations.

(6) **EACE** (Zhou et al., 2024b) constructs a role dependency tree that predefines hierarchical dependency relationships among roles for each event type.

(7) **DEEIA** (Liu et al., 2024) use PLM to capture the attention weight matrix between each token in the input text, then explicitly integrate the contextual semantics of events into role representations

| Dataset | RAMS | | | WIKIEVENTS | | | OEE-CFC | | |
|---|---|---|---|---|---|---|---|---|---|
| **Event types** | 139 | | | 50 | | | 1 | | |
| **Role types** | 65 | | | 69 | | | 21 | | |
| **Spilt** | **Train** | **Dev** | **Test** | **Train** | **Dev** | **Test** | **Train** | **Dev** | **Test** |
| **Events** | 7,329 | 924 | 871 | 3,241 | 345 | 365 | 12,024 | 3,621 | 1,824 |
| **Arguments** | 17,026 | 2,188 | 2,023 | 4,542 | 428 | 566 | 30,497 | 9,389 | 4,828 |

Table 8: Data statistics of RAMS, WIKIEVENT, and OEE-CFC

through this weight matrix.

(8) **HMPEAE** (Zhang et al., 2024) leverage a hyperspherical prototype mechanism (where each role has multiple prototypes), match spans with the best prototype, and allow spans to update their matched prototype representations. Calculate the loss based on the distance between spans and their matched prototypes, achieving optimization of the role-based span selection model's performance by capturing semantic information from predicted spans.

## C Detailed Ablation Study

### C.1 Number of Graph Update Iterations

While Section 5.3 examined whether node-edge interaction updates benefit argument extraction, this section investigates the impact of varying graph iteration counts, As shown in Figure 4 and Table 6, the model's performance first improves and then declines as the number of graph update iterations increases. Both Arg-I and Arg-C achieve their peak performance (56.75% and 52.1%, respectively) at the 2nd iteration, then declines. Within an appropriate iteration range, multi-round updates effectively refine node and edge representations, enhancing the model's understanding of intra-event semantics. However, insufficient iterations fail to fully model role dependencies, while excessive iterations induce over-smoothing, eroding node distinctiveness and weakening the model's ability to resolve complex role relationships.

### C.2 Number of Rounds for Mutil-Round Role Learning

This section evaluates model performance with different learning rounds, including EPIG reconstruction and updates based on the original EPIG, as illustrated in Figure 5 and Table 7. The results reveal that the model achieves strong performance with 1 learning round. When increasing to 2 rounds, the reconstruction strategy improves Arg-I. However, both strategies exhibit performance degradation at 3 or more rounds. For instance, reconstructed EPIG yields 51.44% in Arg-C at round 3, with a slight rebound to 51.88% at round 4, yet the overall trend remains downward. The original EPIG-based strategy shows a more pronounced decline, reaching 51.07% in Arg-C at round 4.

In summary, increasing learning rounds introduces overfitting risks, where the model captures noise or specific patterns in training data rather than generalizable rules, thereby weakening generalization on unseen data. Additionally, excessive iterations with original EPIG updates lead to over-smoothing and diminished node representation distinctiveness, reducing the model's ability to discriminate between nodes and capture complex relationships. Furthermore, errors from initial iterations accumulate progressively with additional rounds, ultimately degrading prediction accuracy.