A Syntactic Feature Interaction and Cross-table Relation Search Model for Threat Intelligence Triple Extraction

Long Chen[1], Chong Zhao[2,*], Jingjun Deng[2], Dongming Zuo[2], Jianjiang Zheng[3]
[1]*School of Electronic and Information Engineering, China West Normal University, Nanchong, China*
[2]*School of Computer Science, China West Normal University, Nanchong, China*
[3]*Information Technology Center, Chongqing Vocational Institute of Tourism, Chongqing, China*
chenlongxihua@stu.cwnu.edu.cn, zhaochong@cwnu.edu.cn, djing0974@gmail.com,
xiaomingtongxue@stu.cwnu.edu.cn, tsgzjj@cqvit.edu.cn

***Abstract*—Joint relational triple extraction models based on table filling are the key methods for Cyber Threat Intelligence (CTI) analysis. However, traditional table models typically encode the subject and object of CTI independently and excessively focus on the local features of a single table, resulting in overlooking subject-object semantic interaction and the global associations of relations and entities. To address these issues, we propose a threat intelligence triple extraction model based on Syntactic Feature interaction and Cross-table Search strategy (SyCro). First, we design a Syntactic-aware Graph Convolutional Network (SGCN), which calculates a syntactic feature matrix by integrating syntactic dependency, thereby establishing syntactic associations between subjects and objects. Second, we construct a table feature for each relation of triple and design a cross-table relation search strategy, which utilizes FourierKAN-Attention to capture the interaction relationships between single tables, thereby extracting global relation features. Finally, we integrate the syntactic feature matrix into the global relation features and iteratively apply the above process multiple times to obtain the final table features, thereby further exploring the global associations of relations and entities. Experimental results demonstrate that SyCro significantly outperforms existing models, achieving SOTA results on two intelligence datasets. Our code is available at https://github.com/xiaochen-xihua/SyCro.**

***Index Terms*—Threat intelligence, Extraction of relational triplets, Table models**

I. Introduction

In recent years, triple extraction has aimed to extract structured information from massive unstructured texts automatically and plays a vital role in the analysis of threat intelligence [1]–[3].

Currently, triplet extraction research is mainly categorized into three types. The first type is pipeline-based methods, which typically process in two stages: first, named entity recognition (NER) is performed, followed by relation classification on the identified entity pairs [4]. For example, Wang et al. [5] propose an enhanced representation and binary tagging framework (ERBTF). The model utilizes a binary entity tagger to identify subject and object entities, effectively constructing relational triples. However, errors in entity recognition can propagate to the relation classification stage, limiting overall model performance. The second type is joint extraction methods, which combine entity recognition and relation extraction into a unified framework, preventing the error accumulation typical of pipeline-based methods [6], [7]. For instance, Wang et al. [8] proposed the TPLinker model, which integrates entity recognition and relation extraction into a single sequence labeling task. This effectively addresses the error propagation issue in traditional pipeline methods. Besides, Shang et al. [9] propose a span-based multivariate information-aware embedding network (SMIEN) to enhance the joint extraction of entities and relations by introducing a perceptual embedding module. The third type is generative methods, which treat the triplet extraction task as a sequence generation problem, enhancing the flexibility and generalization capability of the approach [10], [11]. For example, Zeng et al. [12] proposed GenRE, which introduces a copy mechanism to achieve multitask learning for entities and relations, effectively addressing the diversity requirements of triplet extraction. However, this type of method requires higher training costs. Therefore, among these three methods, joint extraction based on table filling is currently the mainstream method. [13].

However, compared to general domain applications, threat intelligence texts often contain complex syntactic structures and multi-level nested relations. Traditional table models typically encode the subject and object independently and focus on the local features of tables, leading to overlooking subject-object semantic interaction and the global associations of relations and entities (details as in Section 2). These challenges result in low efficiency in extracting threat intelligence triplets. To address these issues, this paper proposes a threat intelligence triple extraction model based on syntactic feature interaction and cross-table relation search. SyCro aims to achieve more accurate CTI triplet extraction by utilizing the subject-object syntactic feature interaction and cross-table relation search strategy. This research extracts triplets from unstructured threat texts that reflect the early attack trend of the network to a certain extent, thereby providing a foundational

*Corresponding author

basis for subsequent threat intelligence research. The main contributions of our work are listed below.

1) This paper proposes a syntax-aware graph convolutional network. By incorporating an attention mechanism that dynamically assigns weights to different syntactic types, the network effectively extracts syntactic information from subject-object dependency trees.

2) This paper proposes a cross-table relation search strategy. Specifically, the SyCro first employs a FourierKAN-Attention mechanism to capture global relational features across tables. Subsequently, a cross-attention mechanism is applied to extract global association features between entities and relations. This approach effectively extracts triples from the multi-level nested structures in CTI.

II. Problem Analysis

As mentioned before, traditional triplet extraction models face weak semantic correlation between subjects and objects and an excessive focus on local features of entity relations. The detailed description is shown below.

First, traditional table-based extraction models present the relation between the subject and object using product operations or simple feature concatenation. Therefore, these models overlook the syntactic interaction between the subject and object entities, which often prevents them from fully understanding the complex semantic relationships. As shown in Fig. 1, the correct triplet is (attackers, attack, users). In this case, when the traditional model processes the relation between the entities "proxy servers" and "users," the lack of syntactic information about the verb-object relation between "attackers" and "proxy servers" leads to the incorrect extraction of the triplet (proxy servers, attack, users).

Second, traditional table-based extraction models often rely on semantic features of the relative positional relations between tokens in single tables, thereby overlooking the association between single tables. However, in CTI texts, due to complex sentence structures, nested relations, and implicit cross-paragraph context links, relying solely on local features makes it difficult to capture global semantic cues. As shown in Fig. 1, the correct triplet is (attackers, target, account credentials). When the traditional model queries the triplet related to the entity "users," it detects the purpose-indicating word "to" in the sentence, leading to the incorrect extraction of the triplet (attackers, target, communication). Moreover, the sentence contains entity-relation nested information, such as the triplet (attackers, attack, users) in table of "target" and the triplet (users, process, account credentials) in table of "process", but the traditional table models fail to integrate and process this valuable information. Consequently, the lack of modeling global entity semantic consistency and multi-level path information significantly limits the model's performance.

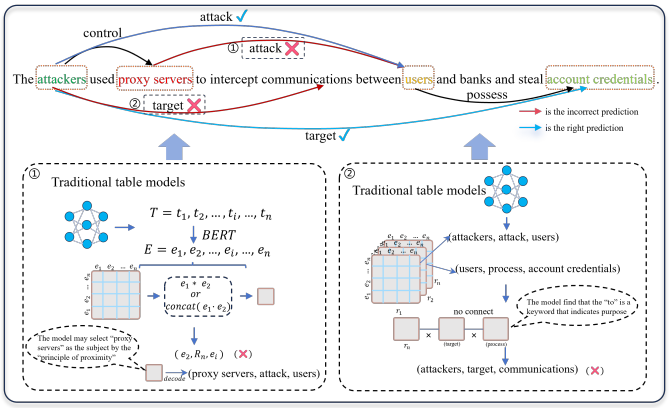

Fig. 1. Example analysis

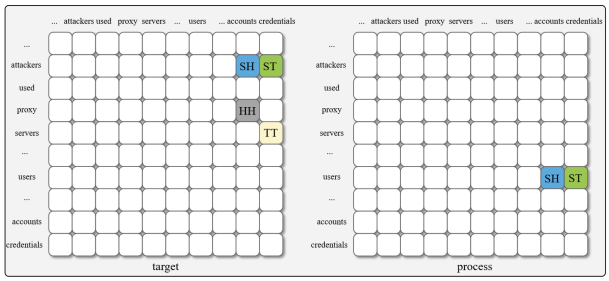

Fig. 2. Examples of table filling and decoding strategy

III. Proposed Method

A. Table Filling Strategy

Given a sentence $S = S_1 S_2 \dots S_n$ and a relation set $R = r_1, r_2, \dots, r_j$, we assign an $n \times n$ table $Table_r$ to each relation r_i.

Next, we define the table label set as follows:

$$L = \{"HH", "TT", "SS", "ST", "SH", "TS", "HS"\} \quad (1)$$

We assign the correct label to each word pair in the table, and for the word pair indexed by the i-th row and j-th column, it is represented as (s_i, s_j) and its label is represents as l. In addition, for the triplet (A, R, B), A represents the subject, B represents the object, and R denotes the relation. The head entity is represented by the first word token in either A or B, while the tail entity is represented by the last word token in either A or B.

The specific meanings of these labels are as follows: H denotes the head entity in the triplet, T represents the tail entity in the triplet, and S indicates that the entity in the triplet consists of a single word token. In Fig. 2, the start and end tags of the subject and object in the triple (proxy servers, target, account credentials) are (proxy, account) and (servers, credentials), respectively. Therefore, the HH label is assigned to (proxy, account) of the relation target table, and the TT label is assigned to (servers, credentials). In addition, for triple (attackers, target, account credentials), the start and end tags of the subject are both attackers, while the start and end tags of

the object entity are 'account' and 'credentials', respectively. Therefore, the SH label is assigned to (attackers, account), and the ST label is assigned to (attackers, credentials).

B. Model details

As shown in Fig. 3, the framework contains two main components: A. the syntactic feature interaction module and B. the cross-table relation search module. First, given an input sentence, the SyCro applies a Syntactic-aware Graph Convolutional Network (SGCN) to compute a syntactic feature matrix by integrating dependency parsing, thereby modeling subject–object grammatical interactions (see Section B.1 for details). Second, for each candidate relation in the sentence, a relational table is constructed by SyCro and processed by the cross-table relation search module, which employs FourierKAN-Attention to capture global relational features across multiple tables (see Section B.2 for details). Finally, the syntactic feature matrix and global relational features are iteratively fused to update and refine the relational table representations, and the final table features are decoded to generate the extracted triplets.

1) Syntactic Feature Interaction Module: In order to solve the problem of insufficient subject-object grammatical interaction, this paper proposes a syntactic feature interaction strategy based on a syntactic-aware graph convolutional network. SyCro incorporates an attention mechanism within the GCN [14] to dynamically assign different weights to each syntactic type, effectively extracting syntactic information from the subject-object dependency tree and enhancing both structural and semantic connections between entities. The method involves four steps: (1) extracting the syntactic dependency matrix, (2) constructing the syntactic attention adjacency matrix, (3) constructing type-enhanced sequence embeddings, and (4) fusing syntactic features.

(1) Extracting the syntactic dependency matrix

For the input text sequence $T = \{t_1, t_2, \ldots, t_n\}$, we first use the syntactic analyzer SpaCy [1] to perform dependency syntactic analysis on it and obtains the syntactic dependency graph $G = (V, E)$, as shown in the formula Eq (2).

$$T \xrightarrow{\text{SpaCy}} G = (V, E) \tag{2}$$

where the node set V represents all words, the edge set $E \subseteq V$ represents the dependency relation between words, and each edge $(i, j) \in E$ corresponds to a dependency label $sd_{ij} \in R$, which represents the dependency type between words t_i and t_j. In addition, in order to enable SGCN to handle directed graphs, each dependency type is connected with "in" and "out", which represent forward and backward relations, respectively.

(2) Constructing the syntactic attention adjacency matrix

After getting the syntactic dependency matrix G, as shown in formula Eq (3), the initial syntactic attention adjacency matrix $\hat{G}$ is calculated through the gated attention mechanism. Then the adjacency matrix G is element-wise multiplied with the identity matrix of the dependency matrix $\hat{G}$, and the result is passed through a Softmax function to obtain the final type-attention adjacency matrix M. The specific formula is as follows:

$$\begin{aligned} e_i &= \text{BERT}(t_i) \\ \hat{G} &= score_{i,j} = \sigma(W_g^T \tanh(e_i W_i + e_j W_j + sd_{i,j} W_{sd} + b)) \\ M &= P_{i,j} = \frac{a_{i,j} \cdot \exp(\text{score}_{i,j})}{\sum_{j=1}^{n} a_{i,j} \cdot \exp(\text{score}_{i,j})} \end{aligned} \tag{3}$$

where W_i, W_j, W_{sd}, W_g are all trainable parameter weights, b is a learnable bias vector, e_i represents the word embedding corresponding to the input text t_i, $a_{i,j}$ represents the value corresponding to each edge $(i, j) \in E$ in the unit matrix, $\tanh$ is the activation function, and σ is the sigmoid function.

(3) Constructing Type-enhanced Sequence Embeddings

While getting the syntactic dependency matrix, the syntactic dependency vector is fused with the BERT output word embedding to obtain the enhanced sequence embedding. The formula is as follows:

$$\hat{e}_j = \tanh\left([e_j; sd_{ij}]\, W_{\text{enh}} + b_{\text{enh}}\right) \tag{4}$$

where $W_{\text{enh}} \in \mathbb{R}$ and $b_{\text{enh}} \in \mathbb{R}$ are all trainable parameters.

(4) Fusing Syntactic Features

After completing the above steps, the syntactic attention adjacency matrix vector and the type-enhanced sequence embedding vector are simultaneously input into the GCN, resulting in the subject-object interaction syntactic feature embedding E. The formula is as follows:

$$E_j = relu\left(\sum_{j=1}^{n} P_{ij}\left(\hat{e}_j W_{\text{en}} + b_{\text{en}}\right)\right) \tag{5}$$

where $relu$ is the activation function, W_{en} and b_{en} are learnable parameter embeddings.

2) Cross-table Relation Search Module: To effectively handle multi-level nested relations in tables. This paper proposes a cross-table relation search strategy that models the semantic structure of triples in a table from a global perspective. As shown in Fig. 2, after obtaining the enhanced embeddings that capture subject-object interactions, the relation table features for each relation type are generated. Then, a maximum pooling operation is performed on each relation table feature to extract the maximum relation feature of each table. Next, the process proceeds to the FourierKAN-Attention module, which further captures global relation features within the table. A cross-attention mechanism is employed to achieve fine-grained alignment and fusion between the subject-object interaction embeddings and the global relation features. Finally, SyCro iteratively applies the above process to extract global features from threat intelligence effectively. Specifically, the strategy involves four steps: (1) Subject and object features extraction, (2) Global relation features extraction, (3) Table features refinement, and (4) Table features decoding.

(1) Subject and object features extraction

First, we feed the enhanced subject-object interaction vector into two separate feed-forward networks to generate the

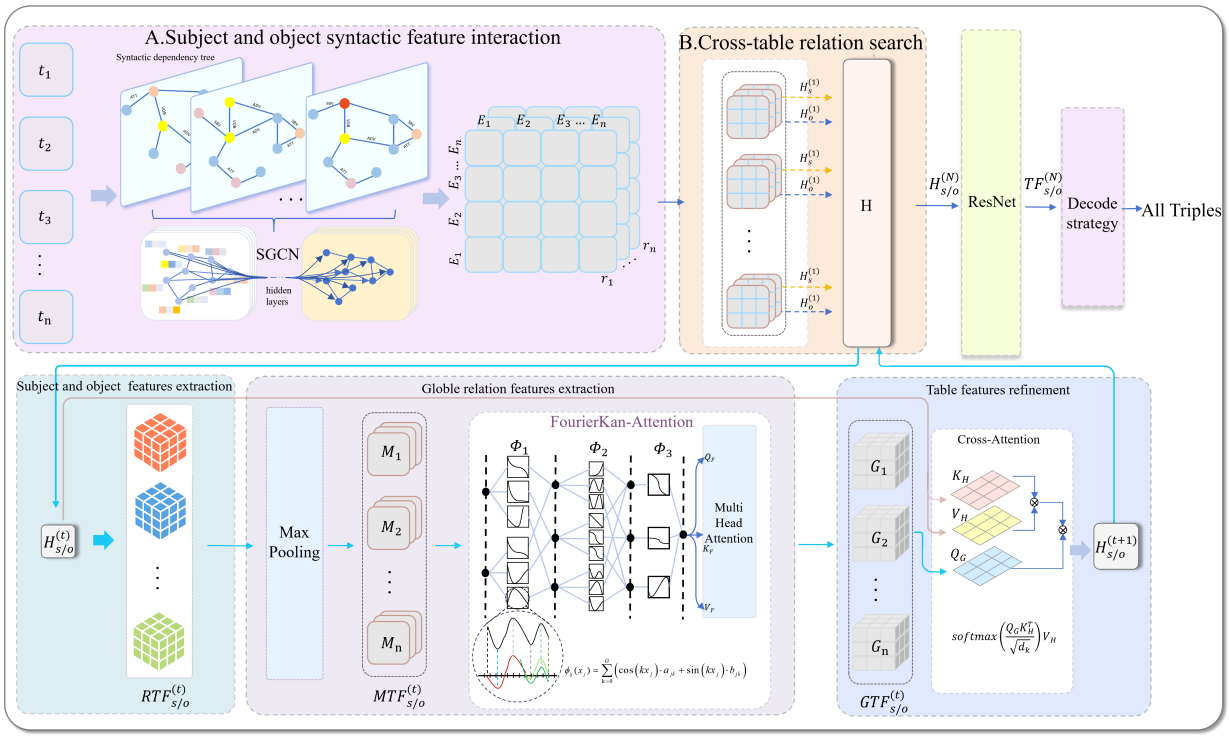

Fig. 3. SyCro model

features associated with the subject $H_s^{(1)}$ and the object $H_o^{(1)}$, respectively. The specific formulation is given as follows.

$$H_s^{(1)} = W_1 E_i + b_1; H_o^{(1)} = W_2 E_i + b_2 \tag{6}$$

where $W_{1/2} \in \mathbb{R}$ and $b_{1/2} \in \mathbb{R}$ are all trainable and weight and bias vectors, respectively. $H_{s/o}^{(1)}$ represents the subject-object feature embedding generated by the first iteration.

Besides, in order to obtain the table features of each relation, we use the Hadamard Product operation and the full connection operation to learn the relation table features $RTF^{(t)}(i,j)$ between subject and object. The formula is as follows.

$$RTF^{(t)}(i,j) = W_3 \,\mathrm{ReLU}\left(H_{s,i}^{(t)} \circ H_{o,j}^{(t)}\right) + b_3 \tag{7}$$

where $\circ$ represents the Hadamard product operation, $\mathrm{ReLU}(\cdot)$ is the rectified linear unit activation function, W_3, b_3 are the trainable weight and bias vectors respectively.

(2) Global relation features extraction

After getting the relation table features $RTF^{(t)}(i,j)$, we calculate the maximum relation feature embedding $MTF_{s/o}^{(t)}$ of each table through maximum pooling and full connection operations. The formula is as follows.

$$\begin{aligned} MTF_s^{(t)} &= W_4 \,\mathrm{maxpool}\left(RTF^{(t)}\right) + b_4 \\ MTF_o^{(t)} &= W_5 \,\mathrm{maxpool}\left(RTF^{(t)}\right) + b_5 \end{aligned} \tag{8}$$

where $W_4, W_5 \in \mathbb{R}$, and $b_4, b_5 \in \mathbb{R}$ are the trainable weight and bias vectors, respectively, and $\mathrm{maxpool}(\cdot)$ represents the maximum pooling operation.

Furthermore, existing studies [15], [16] have demonstrated that, compared with conventional MLP-based attention networks, the FourierKAN-based attention networks is more effective at capturing structured feature patterns and offers improved interpretability. Therefore, we use the FourierKAN-Attention network to capture global relation features $GTF_{s/o}^{(t)}$ between different tables. The formula is shown below.

$$GTF_{s/o}^{(t)} = \text{FourierKAN-Attention}(MTF_{s/o}^{(t)}) \tag{9}$$

(3) Table features refinement

After getting the global relation features $GTF_{s/o}^{(t)}$, we use $GTF_{s/o}^{(t)}$ as Query, the subject and object feature embeddings $H_{s/o}^{(t)}$ as Key and Value, and calculates the response weights of the subject and object entities to the relation through the cross-attention mechanism. Finally, SyCro outputs the global feature embedding. The specific formula is as follows.

$$H_{s/o}^{(t+1)} = crossAttention(GTF_{s/o}^{(t)}, H_{s/o}^{(t)}, H_{s/o}^{(t)}) \tag{10}$$

(4) Table features decoding

After continuously iterating the above three processes, the global feature embedding $H_{s/o}^{N}$ of the last iteration is used as the input of decoding and decoded according to the Algorithm 1 decoding strategy. Finally, all triples in sentence T are generated. The specific steps are as follows.

First, the feature embedding of the subject and the feature embedding of the object are concatenated to produce a combined table feature, as shown in Eq. (11)

$$TF^{(N)}(i,j) = W_6 \left(H_{s,j}^{(N)} \circ H_{o,j}^{(N)}\right) + b_6 \tag{11}$$

where $W_6 \in \mathbb{R}$, $b_6 \in \mathbb{R}$ are the trainable weight and bias vectors, respectively.

Then, according to the combined table feature $TF^{(N)}(i,j)$, the entity relation triples RT are generated through the decoding Algorithm 1. The formula is as follows:

$$\begin{aligned} \widehat{table_r}(i,j) &= W_7\, TF^{(N)}(i,j) + b_7 \\ table_r(i,j) &= \arg\max\left(\text{softmax}\left(\widehat{table_r}(i,j)\right)\right) \\ table_r(i,j) &\xrightarrow{\text{Algorithm 1}} RT \end{aligned} \tag{12}$$

where $W_7 \in \mathbb{R}$, $b_7 \in \mathbb{R}$ are the trainable weight and bias vectors, respectively, and $\arg\max(\cdot)$ is the output with maximum probability, $\text{softmax}(\cdot)$ is the softmax activation function.

Algorithm 1 Table Decoding Strategy

Input: The relation set $R = \{r_1, r_2, \ldots, r_m\}$, the sentence $S = \{s_1, s_2, \ldots, s_n\}$, and all $table_r(i,j) \in \mathbb{R}^{m \times n}$ for each relation $r \in R$.
Output: The predicted triplet set: RT
Define three temporary triple sets A, B, and initialize $A, B \leftarrow \emptyset, \emptyset$.
for each $r \in R$ **do**
    Define two temporary sets W^H and W^T which consist of token pairs with "H" and "T" tags in $table_r(i,j)$
    Define a temporary set W^{SS}, which is the set of token pairs labeled "SS" as in $table_r(i,j)$
    for each $(s_i, s_j) \in W^H$ **do**
        for each $(s_k, s_m) \in W^T$ **do**
            if $i \le k, j \le m$ and token $(s_i, s_j), (s_k, s_m)$ are closest in the table S **then**
                Add $(s_{i\ldots k}, r, s_{j\ldots m})$ to A
            end if
        end for
    end for
    for each $(s_i, s_j) \in W^{SS}$ **do**
        if token s_i is closest in the sentence S **then**
            Add (s_i, r, s_j) to B
        end if
    end for
end for
Return $RT \leftarrow (A \cup B)$

Moreover, the loss function of the SyCro model is defined as follows.

$$L = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{|R|} y_{(i,j),r} - \log P_{(i,j),r}\left(y_{(i,j),r} = table(i,j)\right) \tag{13}$$

where y represents the true value, P represents the predicted label probability, i and j represent the index values of the two-dimensional table, r represents the index value of the relation, and n and R represent the length of sentence S and the number of relations.

TABLE I
STATISTICS OF THE DATASETS

Category	Hacker			SG-CTI		
	Train	*Dev*	*Test*	*Train*	*Dev*	*Test*
Normal	4982	522	520	4570	672	559
EPO	1129	85	109	2886	468	269
ALL	6111	607	629	7456	1040	828
Relation		22			24	

IV. EXPERIMENTS

A. Experimental Settings

1) Datasets: This paper evaluates the SyCro model using the Hacker [17] and SG-CTI datasets.

The Hacker dataset, developed by Luo et al. for cyber threat intelligence relation extraction, originally contained 2,153 documents with over 9,000 sentence instances and 22 relation types; after cleaning, it yielded 3,587 sentences and 6,726 triples, split into training, validation, and test sets (8:1:1). Statistics of the Hacker dataset is provided in Table I.

Due to the lack of publicly available datasets in the threat intelligence domain, we constructed a dataset for joint extraction of threat intelligence relational triples. We collected approximately 2,000 documents from open-source platforms (OpenCVE, ATT&CK, CAPEC), cybersecurity companies, hacker forums, blogs, and GitHub. After data cleaning, four graduate students annotated the data using the BRAT [4] platform following prior training. To ensure annotation quality, we evaluated inter-annotator agreement (IAA) on a randomly sampled subset using Fleiss' Kappa, achieving scores of 0.82 for entity annotation and 0.76 for relation annotation, indicating substantial agreement. Disagreements were resolved through discussion and final adjudication by a senior researcher. The final corpus contains 4,983 annotated sentences. Following the STIX 2.1 standard, we defined 22 entity types and 23 relation types, resulting in 9,501 entity–relation triples, which were split into training, validation, and test sets at an 8:1:1 ratio. Detailed statistics are reported in Table I.

2) Implementation details: Experimental settings are described as follows. All experiments were conducted on a machine equipped with an NVIDIA RTX 4090 GPU (24 GB) and running Windows 11, with Python 3.10 as the development environment. We employed the Adam optimizer to update model parameters, with a learning rate of 3e-5. The model was trained for 50 epochs. To ensure robustness, each experiment was repeated 10 times, and the average results were reported. For text representation, we adopted the pre-trained BERT-BASE-CASED model, which produces 768-dimensional embeddings. The number of iterative layers in the proposed model was set to 4 for both datasets.

3) Baselines: In this section, the SyCro model is compared with several baseline models as follows.

CasRel [18] (2019, Wei et al.) is a cascade binary tagging framework that extracts relational triples from unstructured

text, effectively handling overlapping triples by mapping subjects to objects.

TPLinker [8] (2020, Cheng et al.) is a joint extraction model using a handshaking tagging scheme to link token pairs, reducing error accumulation in overlapping relations.

GRTE [19] (2021, Ren et al.) refines table-filling features via a generate–mine–fuse mechanism, enhancing global feature extraction.

UniRel [20] (2022, Tang et al.) concatenates entities and relations into sequences and uses an Interaction Map with self-attention to unify entity-relation interactions.

SPAN4RE [21] (2023, Sui et al.) employs a non-autoregressive decoding framework with multi-head attention and bipartite graph matching loss, improving robustness for complex relation extraction.

SOIRP [22] (2024, Dai et al.) is a joint relational triple extraction model that addresses overlapping triples and entity nesting via two novel components: a Subject-Object Interaction (SOI) module for capturing interactions, and a Reasoning Path Extraction (RPE) module for path inference

SMIEN [9] (2024, Shang et al.) targets threat intelligence, using span-based multivariate embeddings and a type-aware GCN to strengthen entity-relation representation under high semantic similarity.

B. Main Experimental Results

Table II compares the proposed SyCro model with baseline methods on the HACKER and SG-CTI datasets. Among all baselines, CasRel performs the worst, with an F1-score of only 25%, mainly due to its cascade framework, which focuses on local entity–relation extraction and fails to capture global relational features.

In contrast, SyCro consistently outperforms the second-best model, achieving an average F1-score improvement of 2.0% across both datasets. On the HACKER dataset, SyCro further surpasses SMIEN by 4.13% in Precision. Although SMIEN enhances syntactic feature interactions, it lacks explicit modeling of global associations among subjects, objects, and relations.

On the SG-CTI dataset, SyCro exceeds the second-best GRTE model by 1.89% in F1-score. While GRTE leverages Transformer-based global feature extraction, it overlooks subject–object syntactic interactions in complex CTI texts. In contrast, SyCro integrates a syntactic-aware graph convolutional network to model dependency structures and employs FourierKAN-Attention to capture long-range dependencies, leading to more effective structural and semantic representation.

C. Detailed Results

In this section, we will demonstrate the effectiveness of our model from the following three aspects.

1) Ablation experiments: To evaluate the contribution of each component in the SyCro model, this paper conducted an ablation study by incrementally retaining only a single module and observing the performance impact. As shown in

TABLE II
COMPARISON OF METHODS ON HACKER AND SG-CTI DATASETS

Baselines	**Hacker**			**SG-CTI**		
	P(%)	**R(%)**	**F1(%)**	**P(%)**	**R(%)**	**F1(%)**
CasRel [18]	33.84	19.82	25.00	88.74	88.47	88.21
TPLinker [8]	40.97	33.78	37.03	89.54	86.71	88.10
GRTE [19]	45.74	35.58	40.02	90.19	86.74	88.43
UniRel [20]	35.86	39.42	37.56	87.24	86.43	86.83
SPAN4RE [21]	36.18	34.42	35.27	89.84	87.04	88.42
SMIEN [9]	45.98	**43.55**	44.73	-	-	-
SOIRP [22]	37.76	36.46	37.10	89.90	83.34	86.50
SyCro model	**50.11**	42.09	**45.74**	**91.89**	**88.75**	**90.32**

TABLE III
COMPARISON OF METHODS ON HACKER AND SG-CTI DATASETS

Models	**Hacker**			**SG-CTI**		
	P(%)	**R(%)**	**F1(%)**	**P(%)**	**R(%)**	**F1(%)**
SyCro	50.11	42.09	45.74	91.89	88.75	90.32
SyCro(c1)	47.06	39.87	43.62	91.57	86.97	89.21
SyCro(c2)	49.04	39.56	43.79	91.29	85.77	89.09

Table III, the F1-scores on both the Hacker and SyCro datasets consistently decline across all ablation settings, indicating that each module plays a vital role in enhancing the overall model performance.

Specifically, in SyCro(c1), the cross-table relation search module is removed, retaining only the syntactic feature interaction module. The F1-score drops to 43.62%, lower than the full model's 45.74%. This highlights the importance of cross-table relation search module in capturing semantic associations and modeling entity-relation interactions in threat intelligence. In the SyCro(c2) model, where the syntactic feature interaction module is removed, and only the cross-table relation search module is retained, the F1-score is 43.79%. Although this model achieves slightly higher Precision, the performance gap with the full model suggests that global feature extraction alone cannot compensate for the absence of subject-object interaction.

2) Analysis of normal and overlapping triplets: To assess the SyCro model's ability to extract overlapping relational triples, this paper examines its performance on both standard and overlapping triples. Here, an overlapping triple is defined as a relational triple that shares at least one entity with another triple in the same sentence.

As shown in Fig. 4, SyCro outperforms all three baseline models not only in extracting standard triples, but also in handling overlapping triples more effectively. Research shows that the subject-object interaction module of SyCro integrates the syntactic features of the subject and object in the intelligence text through the SGCN network. This enhances structural interaction and improves the accuracy of triple extraction.

3) Analysis of computational efficiency: Table IV compares the computational efficiency of SyCro with state-of-the-art models. All experiments were conducted under the same conditions. Inference time is reported as the average milliseconds

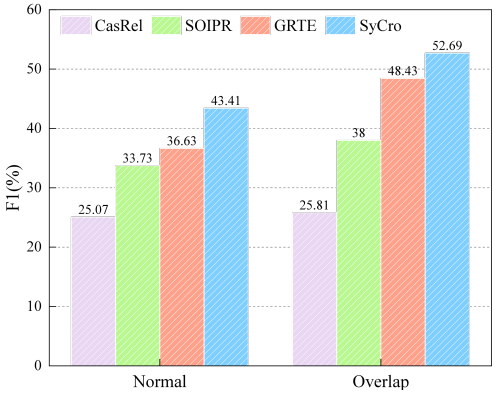

Fig. 4. Comparison of the extraction efficiency of the baseline and SyCro models for normal and overlapping triples on the Hacker test set

TABLE IV
COMPUTATIONAL EFFICIENCY OF METHODS ON HACKER AND SG-CTI DATASETS

Models	Hacker			SG-CTI		
	Params	Prop	Time	Params	Prop	Time
CasRel_BERT	323,265,168	99.73%	45.3	323,274,396	99.96%	58.5
GRTE_BERT	119,350,960	90.14%	11.9	122,098,008	93.21%	12.7
SOIRP_BERT	123,774,598	95.78%	37.7	123,783,826	96.73%	41.3
SyCro_BERT	134,879,799	94.72%	25.4	139,045,447	98.74%	32.6

per sample.

The results show that SyCro introduces a modest increase in the number of parameters compared to GRTE, mainly due to the incorporation of the FourierKAN-Attention. Despite this, SyCro maintains competitive inference efficiency. This can be attributed to two factors: (1) its one-stage extraction architecture supports efficient batch processing, in contrast to cascade frameworks such as CasRel; and (2) its lightweight cross-table retrieval mechanism avoids the computational overhead associated with Transformer-based interaction models, such as SOIRP. Notably, for BERT-based encoders, the pretrained backbone dominates both parameter count and inference time, making other components negligible in comparison.

V. CONCLUSION

This paper proposes SyCro, a relation extraction model for threat intelligence that integrates syntactic interaction and cross-table relation search. Experimental results on the Hacker and SG-CTI datasets demonstrate SyCro's superior performance over existing methods, with ablation studies confirming the contribution of each module.

In future work, we will improve performance on small or noisy datasets through advanced data augmentation.

ACKNOWLEDGMENT

The work was supported by the Sichuan Science and Technology Program (No. SCJJ25RKX173), and the Development of a Low-Code-Based Smart Tourism Teaching Platform (KJQN202304603).

REFERENCES

[1] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang, and Z. Tian, "Cskg4apt: A cybersecurity knowledge graph for advanced persistent threat organization attribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5695–5709, 2022.

[2] K. Wilhoit and J. Opacki, *Operationalizing Threat Intelligence*. Packt Publishing, 2022.

[3] L. Chen, H. Deng, J. Zhang, B. Zheng, and R. Jiang, "Threat intelligence named entity recognition based on segment-level information extraction and similar semantic space construction," *Symmetry*, vol. 17, no. 5, p. 783, 2025.

[4] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "Brat: a web-based tool for nlp-assisted text annotation," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 102–107.

[5] X. Wang, Z. Liu, and J. Liu, "Joint relational triple extraction with enhanced representation and binary tagging framework in cybersecurity," *Computers & Security*, vol. 144, p. 104001, 2024.

[6] Y. Yuan, X. Zhou, S. Pan, Q. Zhu, Z. Song, and L. Guo, "A relation-specific attention network for joint entity and relation extraction," in *International joint conference on artificial intelligence*. International Joint Conference on Artificial Intelligence, 2021.

[7] B. Yu, Z. Zhang, X. Shu, Y. Wang, T. Liu, B. Wang, and S. Li, "Joint extraction of entities and relations based on a novel decomposition strategy," *arXiv preprint arXiv:1909.04273*, 2019.

[8] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "Tplinker: Single-stage joint extraction of entities and relations through token pair linking," *arXiv preprint arXiv:2010.13415*, 2020.

[9] W. Shang, B. Wang, P. Zhu, L. Ding, and S. Wang, "A span-based multivariate information-aware embedding network for joint relational triplet extraction of threat intelligence," *Knowledge-based systems*, no. Jul.8, p. 295, 2024.

[10] Y. Guo, H. Zan, H. Chang, L. Zhou, and K. Zhang, "A bart-based study of entity-relationship extraction for electronic medical records of cardiovascular diseases," in *China Health Information Processing Conference*. Springer, 2023, pp. 82–97.

[11] R. R. Choudhury and S. Dey, "Gpt-3 powered information extraction for building robust knowledge bases," *arXiv preprint arXiv:2408.04641*, 2024.

[12] D. Zeng, H. Zhang, and Q. Liu, "Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 9507–9514.

[13] N. An, L. Hei, Y. Jiang, W. Meng, J. Hu, B. Huang, and F. Ren, "Rtf: region-based table filling method for relational triple extraction," *arXiv preprint arXiv:2404.19154*, 2024.

[14] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[15] A. A. Imran and M. F. Ishmam, "Fourierkan outperforms mlp on text classification head fine-tuning," *arXiv preprint arXiv:2408.08803*, 2024.

[16] J. Xu, Z. Chen, J. Li, S. Yang, W. Wang, X. Hu, and E. C. Ngai, "Fourierkan-gcf: Fourier kolmogorov-arnold network–an effective and efficient feature transformation for graph collaborative filtering," *arXiv preprint arXiv:2406.01034*, vol. 10, 2024.

[17] Y. Luo, S. Ao, N. Luo, C. Su, P. Yang, and Z. Jiang, "Extracting threat intelligence relations using distant supervision and neural networks," in *IFIP International Conference on Digital Forensics*. Springer, 2021, pp. 193–211.

[18] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," *arXiv preprint arXiv:1909.03227*, 2019.

[19] F. Ren, L. Zhang, S. Yin, X. Zhao, S. Liu, B. Li, and Y. Liu, "A novel global feature-oriented relational triple extraction model based on table filling," *arXiv preprint arXiv:2109.06705*, 2021.

[20] W. Tang, B. Xu, Y. Zhao, Z. Mao, Y. Liu, Y. Liao, and H. Xie, "Unirel: Unified representation and interaction for joint relational triple extraction," *arXiv preprint arXiv:2211.09039*, 2022.

[21] D. Sui, X. Zeng, Y. Chen, K. Liu, and J. Zhao, "Joint entity and relation extraction with set prediction networks," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 9, pp. 12 784–12 795, 2023.

[22] Q. Dai, W. Yang, L. Wang, F. Wei, and M. Tuo, "Soirp: Subject-object interaction and reasoning path based joint relational triple extraction by table filling," *Neurocomputing*, vol. 580, p. 127492, 2024.