Detecting Spam Movie Review under Coordinated Attack with Multi-View Explicit and Implicit Relations Semantics Fusion

Yicheng Cai, Haizhou Wang, *Member, IEEE,* Hao Cao, Wenxian Wang, Lei Zhang, and Xingshu Chen

***Abstract*—Spam reviews have long polluted review systems, undermining their industries. Detecting spam movie reviews faces some brand-new challenges compared to traditional spam detection. These include coordinated spamming attacks during premieres or at advance screenings. However, most of existing studies only use inherent relations among reviews, movies, and users, they do not fully exploit explicit and implicit relations between reviews in coordinated spamming attacks. To address these novel challenges, we propose a spam movie review detection method based on mining explicit and implicit relation semantics and fusing multi-view semantics. To the best of our knowledge, we are the first to enhance spam movie review detection by exploiting both explicit and implicit relations between reviews in coordinated spamming attacks. First, we build an explicit relation movie-review graph with movie synopses and high-quality external reviews. We extract movie factual knowledge embeddings using a Heterogeneous Graph Transformer (HGT) network. Next, we input the factual knowledge embeddings with corresponding review embeddings into a contrastive network to get review credibility features. Additionally, we build an implicit relation graph between reviews using metadata and semantic similarities. We extract relation-enhanced review semantics via another HGT network. Finally, we fuse the three review semantic features through an attention layer before making classification. Experiments show our method achieves higher performance and robustness over state-of-the-art methods.**

***Index Terms*—Spam movie review detection, multi-view implicit relation, coordinated spamming attack, factual movie knowledge, heterogeneous graph transformer.**

I. Introduction

REVIEW systems are an important way for e-commerce platforms to gather user feedback and understand user experiences with their products. At the same time, reviews, as an important part of review systems, greatly affect purchasing decisions of new consumers for a product or service based on their authenticity and quality [1]. Therefore, spam reviews (i.e., reviews that are false, irrelevant, or meaningless) have received widespread research attention as they could pollute and undermine review systems [2], [3]. Among all Online to Offline (O2O) e-commerce businesses, the movie industry is especially vulnerable to review spam. The issue stems from fierce competition and the Matthew effect, where success breeds more success [4]. Consequently, studying spam movie reviews detection methods is of great significance.

Although research on spam movie reviews detection [4]–[6] is currently limited, broader spam review detection work can provide insights that are likely applicable to the problem of identifying spam movie reviews as well. The current spam review detection methods fall into three categories: review-centric [7], [8], user-based [9] or target-based [10], and relation-based [11]–[13]. However, previous methods only used data from one closed review system. They did not utilize external review system data. Nilizadeh et al. [14] broke this single-system mindset. They identified inconsistent rating trends across platforms in a time window as attacks indicators. Then, they detected spam reviews in those windows with trained models. However, they did not fully explore correlations between systems. Nor did they fully leverage the correlation between the two review systems. Our research investigates these limitations in detail.

Compared to other spam review detection scenarios, spam movie review detection presents several unique challenges: **First**, movie reviews are more subjective, longer, and contain background details like plots, actors, and effects [5]. This complexity makes most of the pre-trained model-based methods [7], [8] perform badly as they did not optimize the representations for terms in the movie domain. **Second**, attackers can mislead viewers with seemingly professional but unsubstantiated advance screening reviews. However, current methods [9]–[13], [15]–[18] fail to cope with such advanced spamming technique that mimics the authentic reviews. **Finally**, in coordinated spamming attacks, the relations between reviews are complex, covert, and uncertain, leading to the failure of existing methods [11]–[13], [15]–[18]. These relation-based methods failed to model the complex implicit relations between reviews, particularly the uncertain consistency and inconsistency of content and attributes in these reviews.

To address the aforementioned challenges, we proposed a spam movie review detection method using explicit and implicit relation semantics with multi-view fusion. **For the first challenge**, we use movie domain corpus to fine-tune the pre-trained language model used in review semantic features extraction. We further fine-tune the layer norm parameters of the model when training it on the downstream task. The fine-tuning process improves movie terminology representation and alleviates semantic drift in the pre-trained language model (see Section III-C2). **For the second challenge**, we construct an explicit movie review graph with movie synopses and

(Corresponding author: Haizhou Wang.)

Yicheng Cai, Haizhou Wang, Hao Cao, Wenxian Wang, and Xingshu Chen are with the School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China (e-mail: cyc21csri@stu.scu.edu.cn; whzh.nc@scu.edu.cn; caohao1@stu.scu.edu.cn; catean@scu.edu.cn; chenxsh@scu.edu.cn).

Lei Zhang is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: zhanglei@scu.edu.cn).

professional reviews from both target and external review systems. We use an HGT network to extract movie fact embeddings from movie nodes of the graph. Afterwards, we pass the extracted embeddings and review semantic embeddings through a contrastive network to obtain the review semantic credibility features. The features help better detect seemingly professional but unsubstantiated spam reviews (see Section III-D1). **For the third challenge**, we model four typical spamming techniques used in coordinated attacks and accordingly construct a multi-view implicit review relation graph based on the content and metadata of reviews. We then extract the graph embedding features using another HGT network. The features effectively uncover the implicit consistent and inconsistent relations between reviews in coordinated spamming attacks (see Section III-E1).

The contributions of our work are summarized as follows:

- **We propose a novel spam movie review detection framework which is the first to jointly incorporate explicit and implicit relations between reviews in coordinated spamming attacks to enhance spam detection.** It first employs explicit relation movie-review graphs to integrate external knowledge on movie facts into review representations. Then it uses multi-view implicit relation review graphs to reveal semantics in relationships between reviews. Experiments show that our method outperforms state-of-the-art methods greatly in performance by an average of 8.5% F1-Score.
- **We utilize the data from an external movie review system to overcome the limitations of detecting spam movie reviews in a single closed movie review system.** Specifically, we design a *Split-Node* operation (see Section III-D1) to fully utilize long-review information from external review system. Besides, we extract review credibility features using contrastive networks to fully utilize explicit external knowledge in a explicit relation movie-review graph.
- **We construct a multi-view implicit relation review graph to uncover consistency and inconsistency implicit relationships in coordinated spamming attacks.** Specifically, we first find out all the review pairs that are similar in content. Then, we divide these reviews into different groups based on score and publishing time of spam movie reviews. Finally, we follow specific rules to build four types of relations within or between these review groups. The experimental results show that this method can effectively reveal the coordinated spamming behaviors and improve spam movie review detection performance.
- **We perform a comprehensive study on the *robustness* and *influencing factors* of the proposed method.** For *robustness*, we have done six experiments in terms of noise in dataset, data sample distribution, data scale, time shifts, external dataset source, and advanced spamming techniques. The results show that our model is robust on real world scenarios. For *influencing factors*, we fully explore the hyper-parameters and components that affect the model performance. The experiments reveal that some hyper-parameters and components are significant to a good detection performance of the proposed method.

II. Related Work

A. Spam Review Detection

Spam review research can be categorized based on modeling objects into review-centric methods [7], [8], [19]–[22], user-based methods [9] or target-based methods [10], [14], and relation-based methods [11]–[13], [15]–[18].

1) Review-Centric Approaches: Such methods model the review itself as the object. Most review-centric methods [7], [19], [21], [22] utilize review text content and metadata to extract review textual features [7], [19], [21], [22] and user behavior features [7], [8], [22]. Some methods [7], [8], [20] also embed review texts into semantic feature representations. Jian et al. [7] extracted features about review text, platform review and user-related metadata. Li et al. [22] extracted N-gram features of reviews and twelve user-centric features from the review dataset.

Review-centric methods mostly require tedious feature engineering. Some features are platform-specific, and limited to these application/dataset-specific features, the methods have poorer generalization capabilities.

2) Account-Based or Target-Based Approaches: Account-based or target-based methods [9], [10], [14] achieve spam review detection by modeling user behavior representations [9] or modeling how coordinated user behaviors cause changes in attributes of attack targets (e.g., product ratings) [10], [14]. Tang et al. [9] designed a GAN network that can generate long-term user behavior features based on new users' three common characteristics, thus enhancing new commenters' behavior feature representations and significantly improving model performance on cold-start spam review detection tasks. Wang et al. [10] extracted suspicion degree features from three-dimension time-series consisting of numbers of reviews, average review ratings, and overall review information entropy through anomaly detection algorithms, effectively improving the detection of coordinated spam reviews.

In general, account-based or target-based methods make assumptions about user behaviors and attack scenarios. This imposes limitations on the scenarios they can handle, since they may miss new or unusual attack vectors that don't match expected behaviors and adversaries can try to evade detection by going outside predicted norms.

3) Relation-Based or Graph-Based Approaches: Relation-based methods [11]–[13], [15]–[18] view the review system as a whole, comprehensively utilizing information of entities (reviews, users, products) in the system and relationships between them to detect spam reviews. Zhang et al. [11] extracted user-item subgraphs and review subgraphs from the complete review system graph according to predefined meta-paths and fuse node representations of users, items, and reviews extracted from meta-path-based subgraphs respectively, enhancing review node representations. Zhu et al. [12] automatically extracted basic attribute features of nodes and relations in the review graph, and cross-generated complex interaction features from these basic features, saving human efforts in feature engineering and improving interpretability. Xu et al. [13] aggregated the semantic embeddings of users, products, and reviews, and enhanced the representation of reviews by the similarity relations between review attributes. However, they did not account for the implicit complex relationships among reviews in the context of coordinated attacks, where reviews may share similar content but have vastly different ratings.

In summary, relation-based methods mainly use graphs that depict explicit relations between entities and attributes, but they overlook the implicit relations that exist between reviews during coordinated spamming attacks. To address this gap, our method incorporates both explicit relations across review systems and implicit relations that reveal the consistency and inconsistency of review content and attributes. This approach enables us to achieve more comprehensive review representations tailored to coordinated spamming attack scenarios.

B. Spam Movie Review Detection

There is currently very little research on spam movie review detection [4]–[6], two [5], [6] of which are our previous works. Gao et al. [4] proposed the first spam movie review detection model. Based on Wasserstein GAN, they concatenated user historical interest preference features as control conditions with semantic embeddings of spam reviews, so that the features of samples generated by the trained generator can better match the real data distribution. However, their method incorporates too much expert experience and relies heavily on the dataset, resulting in poor generalization capabilities. Our previous work [5] proposed a method to fuse movie knowledge with the review semantic embeddings and user behavior features, proving the effectiveness of their method on a self-constructed dataset. Another previous work of us [6] constructed a user interaction graph using relationships between users commenting on the same movie, with manually extracted user features as initial node embeddings in the graph. Then MLP and graph attention networks (GAT) are used to uncover advanced individual user behavior and collaborative inter-user features. Hence, the model can handle the spam review detection scenario during the spam review outbreak period around a movie's release.

However, spammers work collaboratively, yet no existing method has simultaneously considered mining explicit and implicit relation semantics between reviews in coordinated spamming attacks. Our method first constructs a explicit relation movie review-graph to incorporate expert knowledge from an external review system. Then it mines implicit consistency and inconsistency between reviews in terms of review content, scores, and timestamps. The detection performance of model improves significantly by exploiting these explicit and implicit relations. To the best of our knowledge, we are the first to reveal the effectiveness of explicit and implicit relations semantics between reviews on coordinated movie spamming detection scenarios.

III. Methodology

A. Problem Definition

1) Observations on Spam Movie Review Characteristics and Attack Scenarios: In this paper, we integrate professional movie reviews from external systems into the closed target system. Besides, we mine the implicit relations between movie reviews in coordinated spamming attacks to facilitate spam detection. Our method is primarily based on the following two observations:

Observation 1: Companies may hire legal users or even experts to attack a specific review system - hype their works (products) or maliciously slander competitors' works (products). These users tend to work collaboratively in a coordinated spamming attack, which can be revealed by hidden relations between reviews.

The Matthew effect applies in the movie industry - the rich get richer and the poor get poorer. Due to this, legitimately influencing the initial batch of reviews achieves better results than using fake accounts ("water armies") to manipulate overall ratings after release. To this end, movie production companies often hire legal users and experts to post reviews of their upcoming films on target platforms. These carefully crafted reviews promote their own works while criticizing competitors' films being released within the same time period. By generating excitement for their movies and dismissing rivals, they influence the first viewers. Moreover, these spammers tend to work collaboratively. For instance, their spam reviews may manifest higher consistency in content than non-spam reviews. Despite the similarity in content, the publishing time of these spam reviews may vary greatly. *Therefore, implicit consistency and inconsistency patterns exist across review metadata like scores, timing, and semantics. Uncovering these relationships can reveal coordinated spam attacks.*

Observation 2: Public opinions on major aspects of movie works (e.g., special effects, acting, plot, etc.) should be broadly consistent. Non-spam reviews have higher consistency with professional high-quality reviews compared to spam reviews.

Influenced by mainstream social values and objective facts about the works, general public praise or criticism of various aspects of a movie should be similar within the same cultural context, without extremely conflicting subjective viewpoints. Professional high-quality reviews often contain many objective factual descriptions of movies and conform to overall public cognition standards. Non-spam reviews generally have more information than spam reviews and contain more factual information. Moreover, it is common to find people scan through reviews across multiple review systems before a final decision on whether to watch a movie or not. *Therefore, professional movie reviews from external sources contain factual knowledge about movies that can help identify unsubstantiated claims in potential spam reviews.*

2) Definition of Spam Movie Review Detection in This Work: Based on the above observations, our task is defined as follows:

$$\begin{aligned}&f(r_d,(R_D,M_D,R_E))\rightarrow\{spam,non{-}spam\},\\&r_d\in R_D,\Phi(r_d)\rightarrow m_d,(R_D,M_D)\subseteq(R,M);\\&\Phi(r_e)\rightarrow m_d,r_e\in R_E,m_d\in M_D\end{aligned}$$

where (R,M) is the target review system, (R_D,M_D) is the constructed dataset, $r_d\in R_D$ and $m_d\in M_D$ are the reviews

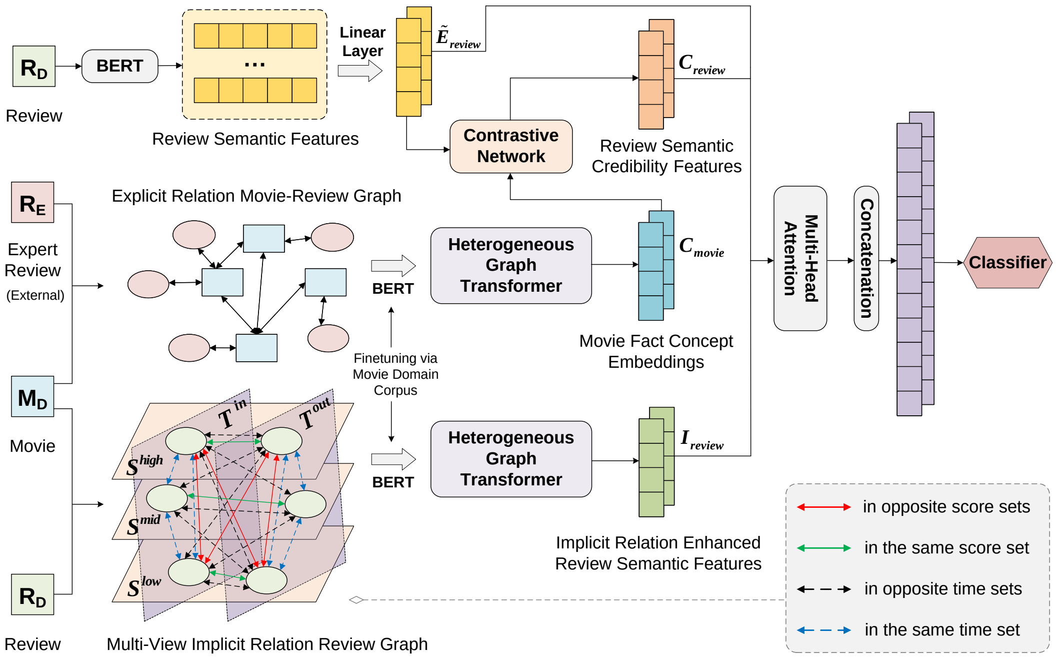


Fig. 1. The proposed spam movie review detection framework enhanced by explicit and implicit relation semantics between movies and reviews.

and movies in the dataset, respectively. The $r_e \in R_E$ represents professional reviews from the external system. The Φ is a mapping function that maps a review to its corresponding movie. For example, $r_d \rightarrow m_d$ means review r_d belongs to the movie m_d.

B. Overview of the Proposed Method

This paper proposes a spam movie review detection method enhanced by explicit and implicit relations with multi-view semantic fusion, as shown in Fig. 1. For convenience, we summarize the main notations of our method in Table I.

- *Text Model:* First, we extract the semantic embedding features of reviews (E_{review}). For the pre-trained language model used in the model, we fine-tune it using all the long review texts from the external system. We further fine-tune the layer norm parameters of the model when training it on the downstream task.
- *Explicit Relation Graph Model:* Second, we construct an explicit movie-review relation graph using movie reviews and synopses from external review systems, and extract movie factual embeddings using an HGT network. The movie factual embeddings are combined with the review semantic embeddings through a contrastive network to extract review semantic credibility features (C_{review}).
- *Implicit Relation Graph Model:* Third, we construct a multi-view implicit relation graph between reviews using review metadata and semantic similarities, and extract implicit relation-enhanced review semantic features (I_{review}) via another HGT network.
- *Classifier:* Finally, the three obtained review semantic features are fused through a multi-head self-attention layer that extracts interactive features, and the fused features are classified to obtain detection results.

TABLE I
DESCRIPTION OF MAIN NOTATIONS

Notation	**Description**
R_D	The movie review dataset of targeted review system
R_E	The movie review dataset of external review system
M_D	The movie dataset of targeted review system
E_{review}	The review semantic features
C_{movie}	The movie fact concept embeddings
C_{review}	The review semantic credibility features
I_{review}	The implicit relation enhanced review semantic features
S^{high}	The reviews whose scores are higher than their movie scores
S^{low}	The reviews whose scores are lower than their movie scores
S^{mid}	The reviews whose scores are similar to their movie scores
T^{in}	The reviews whose timestamps fall in certain time window
T^{out}	The reviews whose timestamps fall out certain time window
Φ	The mapping function that maps a review to its movie
Ψ	The mapping function that maps a movie to its genre types
φ	The score set mapping function for review
χ	The time set mapping function for review
f_{cmp}	The function of contrastive network
Split-Node	The operator for split node operation
δ_{text}	The parameter of movie review text truncation length
ν_{review}	The parameter of reviews number for each movie in ERG
γ_{linear}	The parameter of linear layers output dimension
σ	The threshold of text similarity between reviews in IRG
ρ	The threshold of score difference between reviews in IRG
ω	The threshold of time window between reviews in IRG
α_{hgt}	The parameter of head number in attention layer of HGT
λ_{hgt}	The parameter of HGT layer number

C. Text Model

1) Language Encoder: We use a character-level Chinese BERT (Bidirectional Encoder Representations from Transformers) pre-trained model [23] as the language encoder for review semantic features extraction. The reasons for our choice are as follows: First, it is a common practice to use pre-trained language models (PTM) as the backbone for downstream natural language processing tasks [24]. Second, we experimented with diverse popular choices of model architectures for PTMs, as shown in Section IV-F, the results showed that the Transformer-based BERT model is the most suitable for the spam movie review detection task. Third, since the reviews in our dataset are in Chinese and due to the characteristics of Chinese, we chose character-level tokenization for input review texts.

Our adaption of the PTM to spam movie review detection includes two steps: a) Fine-tuning; b) Training.

a) Fine-Tuning on Movie Domain Corpus: Since the training corpus for the pre-trained BERT model does not contain movie domain knowledge, to get better representations of movie terminology (including plots, special effects, actors, etc.), we first fine-tune the pre-trained model on an appropriate amount of high-quality movie reviews from external systems, as shown in Table II. Following the practice of Tay et al. [25], we set the learning rate to 5e-5 and the sequence length to 512, and pad texts shorter than the target length. We train on the corpus for 20 epochs, saving the model at the end of each epoch. We choose the model with the highest accuracy for all the semantic embedding extraction tasks in our method, including the ERG and the IRG node embedding initialization.

b) Training on Downstream Task: Following the practice of Lu et al. [26], when applying it to spam movie review detection, we add a linear layer to the end of the language encoder, i.e., the BERT model. During the training, we freeze all the parameters of the BERT except for layer norm parameters. This approach prevents issues such as training times increasing quadratically and the model becoming prone to overfitting, which occurs when all model parameters are involved in training. Moreover, it allows the model to adapt extensively to downstream tasks compared to freezing all the parameters.

2) Review Semantic Features Extraction: Given the significant variation in review lengths, we must set an appropriate fixed length, δ_{text}, to balance embedding extraction time and the quality of the embeddings, as shown in Subsection IV-E1. Then, we truncate any reviews longer than the specified input length. For reviews shorter than the input length δ_{text}, we pad them to the full length using special [PAD] tokens. Finally, we input the preprocessed review texts into the language encoder to obtain the semantic embedding representations for the review texts E_{review}.

D. Explicit Relation Graph Model

Based on **Observation 2** in Subsection III-A1, we aim to model the consistency between the content of movie reviews and factual knowledge about the movie. Movie factual knowledge includes the views and descriptions of the main aspects of the movie by the public and the filmmakers. Specifically, we extract the semantic embeddings of reviews, denoted as E_{review}, to represent their content. To represent factual knowledge about the movie, we aggregate semantic embeddings from professional reviews and synopses, denoted as C_{movie}. Then, we use a contrastive network, denoted as f_{cmp}, to measure the semantic consistency between E_{review} and C_{movie} and obtain the semantic credibility features of reviews, denoted as C_{review}. Hence, our modeling objective can be formalized as $f_{cmp}(E_{review}, C_{movie}) \rightarrow C_{review}$.

1) Explicit Relation Movie-Review Graph Construction: To obtain factual knowledge about the movies, we construct an explicit relation movie-review graph (ERG) using high-quality professional reviews from the external review system and movie synopses from the target review system, where $m \in M_D$ and $r_e \in R_E$ represent movie nodes and review nodes, respectively. According to **Observation 2**, high-quality professional reviews from external systems are unlikely to be spam. These reviews should have abundant objective facts about the movies. Additionally, movie synopses from review systems provide basic plot facts. There is minimal risk of tampering with these synopses. Therefore, the information sources used to construct our ERG are reliable and unpolluted.

a) External Review Dataset for ERG Construction: To build the ERG, we first construct the external review dataset from credible sources. For each movie in the target review dataset, we gather professional reviews externally from the most authoritative Chinese movie review website. Specifically, we collect the top 30 popular reviews from the most authoritative movie review websites in China. Popularity of reviews is determined by the total number of thumb-ups and comments. Reviews with the top popularity rankings are generally considered to be representative and high-quality [27]. We then select long reviews (those exceeding 100 characters) and remove reviews exceeding 5,000 characters, as our data analysis indicated that such lengthy reviews are outliers.

b) Design of ERG Schema: Next, we define the basic subgraph patterns for the ERG. As shown in Fig. 2(a), we define two types of relationships. First, there exists an inherent bidirectional edge relationship between review nodes r_e and their corresponding movie nodes m (reviews and reviewed movies); second, descriptions of plots and special effects used have certain similarities for movies of the same type τ, and understanding the same movie type from a human perspective helps form basic factual cognition about that movie type. Hence, bidirectional edges exist between movie nodes m of the same type. We define a mapping function Ψ to map a movie to its corresponding type set, i.e., $\Psi(m) \rightarrow \tau$. In addition, to ensure movie factual information is fully retained in movie nodes, movie nodes should have self-loops.

c) Design of Split-Node Operator: In this work, we use a pre-trained language model to extract semantic embeddings of ERG nodes as initial representations. However, the average length of long reviews exceeds 2,000 characters in Chinese, clearly beyond the input scope of most language models. For this issue, we tried simple truncation (taking the first 510 characters of the text, which is the maximum input of a BERT model). We also considered the practice from [28] - concatenating the first 384 and last 128 characters. However, experiments show that these means are not ideal for retaining information in the review nodes. This is because that factual information about the movies can appear anywhere in the reviews, not necessarily at the beginning, middle, or end, but scattered sporadically. Therefore, simple truncation will inevitably lead to losing important factual information about the movies.

To address the above issues, we design the *Split-Node* operator to retain as much semantic information from the

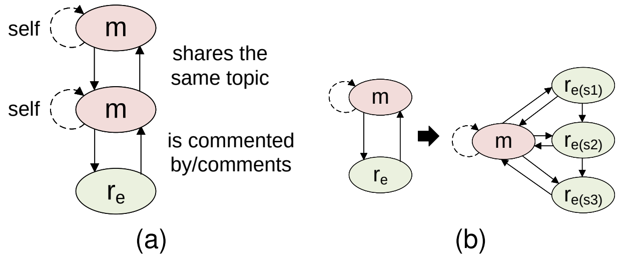


Fig. 2. Illustration of *Split-Node* operator: (a) Subgraph schema; (b) Split node operation.

review texts as possible while not significantly increasing model training overheads. Specifically, as shown in Fig. 2(b), our *Split-Node* operator consists of two steps:

Segmentation: We split lengthy review nodes r_e into multiple review segment nodes $r_{e(s_i)}$ within input scope of the language model, i.e., $r_e = \sum r_{e(s_i)}, i \in N$, where N denotes the natural number set.

Concatenation: First, for every review segment node $r_{e(s_i)}$, we add bidirectional edges between the movie node m and the review segment node $r_{e(s_i)}$. Second, for every review segment node $r_{e(s_i)}, i \geq 2$, we add unidirectional edge from the last review segment node $r_{e(s_{i-1})}$ to the $r_{e(s_i)}$.

The idea of the *Split-Node* operator is inspired by the Hierarchical BERT [29], which adds RNN (Recurrent Neural Network) extensions to the BERT, enabling it to handle long document inputs through simple segmentation. Compared to the Hierarchical BERT, our *Split-Node* operator, when paired with the GNNs (Graph Neural Network), makes it more flexible to handle lengthy diverse information in the structure of graph, with linear growth in computational complexity. The graph ablation experiments showed that this simple mechanism significantly improves the performance of the model, as shown in Section IV-C.

d) Implementation of the Split-Node Operator: For the segmentation step, we set the segmentation length of review nodes to the maximum input document length of the BERT, which is the language model used in the embedding initialization of the graph nodes. In this way, we could minimize the disruptions of semantic continuity of the reviews, with a minimal loss of contextual information critical for spam movie review detection. Specifically, we set the segmentation length to 510 characters, as the reviews are in Chinese and spaces of two tokens are left for the [CLS] and the [SEP] tokens.

For the concatenation step, we tried replacing unidirectional edges between the review segmentation nodes $r_{e(s_i)}$ with bidirectional edges. However, the experimental results in Section IV-C showed that it can lead to a decrease in the model performance.

e) Implementation of the ERG: In this work, we use the HGT to extract and fuse the semantic features of nodes in the ERG. To fully exploit the dynamic weighting mechanism of the HGT for different types of nodes and edges [30], we tried defining heterogeneous edges and nodes in the ERG construction. Nevertheless, the experimental results showed that adding heterogeneity to the edges of the ERG leads to a decrease in model performance, as shown in Section IV-C. Therefore, we implement the ERG with heterogeneous nodes and homogeneous edges. The process of ERG construction is described in Algorithm 1.

Algorithm 1 The ERG Construction Algorithm

```
Input: Target review dataset set $R_D$, Movie dataset $M_D$,
    External review dataset $R_E$, Mapping function $\Phi$, $\Psi$
    Parameter $\nu_{review}$ for ERG.
Output: ERG.
Procedure:
for $m$ in $M_D$ do
    add movie $m$ as a node in ERG;
    for $n$ in ERG; $\tau_m$ in $\Psi(m)$; $\tau_n$ in $\Psi(n)$ do
        if $\tau_m = \tau_n$ and $m$ is not connected with $n$ then
            add bidirectional edge to $m$ and $n$;
        end if
    end for
end for
for $n$ in ERG; $r$ in $\nu_{review}$ do
    sample a review r from $R_E$;
    if $\Phi(r)$ is movie $n$ then
        add review $r$ as a node in ERG;
        add bidirectional edge to $r$ and $n$;
        $Split\text{-}Node(r)$;
    end if
end for
```

2) Movie Fact Concept Embedding Extraction: To fully exploit the inherent connections between movie factual semantics and expertise knowledge in our constructed heterogeneous information network ERG, we build a HGT network [30], allowing important movie facts and knowledge to be fully aggregated and fused into movie nodes. Finally, we obtain movie factual concept embeddings - movie node representations incorporate aggregated basic facts and conceptual information about the movies, denoted as C_{movie}.

The choice of applying the HGT to the embedding extraction lies in that the HGT not only provides dynamic weights for different node and edge types, but also assigns different weights to neighbors during aggregation. This could optimize the movie fact concept embeddings to enhance the sensitivity to spam indicators of our method. Moreover, we compared the HGT with other homogeneous [31], [32] and heterogeneous [33] GNN alternatives, as shown in Section IV-F, the results showed that the HGT is the most suitable for our approach.

3) Contrastive Network and Review Semantic Credibility Feature Extraction: To capture semantic consistency between review text embeddings and corresponding movie factual concept embeddings, we design a contrastive network based on the work of Hu et al. [34]:

$$f_{cmp}(x, y) = W_s[x - y, x \odot y] \tag{1}$$

where $W_s \in R^{(N \times 2N)}$ is a transformation matrix and $\odot$ denotes the element-wise multiplication of two matrices. This lightweight module improves the performance of our model significantly against sophisticated spamming techniques that mimic authentic reviews, as shown in Subsection IV-D6.

We pass the review text embedding $E_{review} \in R^{M}$ through the linear layer $W_e \in R^{(M \times N)}$ to obtain a scaled vector $\widetilde{E}_{review} \in R^{N}$. Then, we input this vector together with the matching movie factual concept embedding C_{movie} into the f_{cmp} function to generate the review semantic credibility feature $C_{review} \in R^{N}$.

E. Implicit Relation Graph Model

Based on **Observation 1** in Section III-A1, we aim to model the consistent and inconsistent hidden patterns that could potentially reveal coordinated spamming attacks. First, in the scenario of coordinated spamming attacks, the content of spam reviews tends to be related. Thus, we mainly focus on spam reviews that reach a certain level of semantic similarity. Second, spam reviewers often post inconsistent reviews to manipulate movie ratings. They might give high ratings to the movie but describe negative aspects in their reviews, and vice versa. Additionally, spammers also copy existing reviews to increase or decrease the ratings. *The edge type-1 and type-2 are constructed accordingly.* Third, groups of spammers collaborate to write similar content or copy-paste reviews in short time periods, which will confuse potential viewers. *The edge type-3 and type-4 are constructed accordingly.*

1) Multi-View Implicit Relation Review Graph Construction: To model the attack patterns mentioned above, we construct a multi-view implicit relation review graph (IRG). First, for each movie, we compute text similarity scores between each two review text embeddings: $\textit{text-sim}(r_i, r_j) = \cos(e_i, e_j)$. If the similarity exceeds a given threshold σ, the two reviews are considered to have similar content. Next, we divide these content-similar reviews into groups based on two metadata - scores and timestamps. The specific grouping method is:

a) Based on Scores: Reviews of certain movies with $\textit{text-sim}(r_i, r_j) > \sigma$ are divided into three groups such that reviews in the same group have similar scores, while reviews from different groups have different scores. The set mapping function φ is defined as:

$$\varphi(r) = \begin{cases} r \in S^{high}, \frac{score_r - score_m}{score_m} > \rho \\ r \in S^{low}, \frac{score_r - score_m}{score_m} < -\rho \\ r \in S^{mid}, otherwise \end{cases} \tag{2}$$

where r is a review of a movie m; S^{high}, S^{low}, S^{mid} represents review set containing reviews whose scores are higher than, lower than, or close to their movie scores, respectively; the ρ is threshold of score difference.

b) Based on Timestamps: Reviews of certain movies with $\textit{text-sim}(r_i, r_j) > \sigma$ are divided into two groups such that reviews in one group are written before a certain time point, while reviews in the other group are written after the time point. We define the time window between the time point and the movie release time as ω (e.g., 7 days). The set mapping function χ is defined as:

$$\chi(r) = \begin{cases} r \in T^{in}, d(time_r - time_m) \leq \omega \\ r \in T^{out}, d(time_r - time_m) > \omega \end{cases} \tag{3}$$

where T^{in} and T^{out} represents review set containing reviews whose timestamps fall in or outside certain time window ω, respectively; $time_r$ is the published timestamp of a review r; $time_m$ is the released timestamp of a movie; r is a review of a movie m; $d()$ calculates the time difference in days.

Then we connect each pair of reviews (r_i, r_j) in the review set with an edge according to certain rules. Specifically:

- Add edge type-1 to (r_i, r_j) if $\textit{text-sim}(r_i, r_j) > \sigma$ and r_i, r_j belong to the same score sets (S^{high} or S^{low} or S^{mid}). The edge type-1 means the reviews are similar both in content and scores.
- Add edge type-2 to (r_i, r_j) if $\textit{text-sim}(r_i, r_j) > \sigma$ and r_i, r_j belong to opposite score sets (S^{high} and S^{low}), respectively. The edge type-2 means the reviews are similar in content but differ a lot in scores.
- Add edge type-3 to (r_i, r_j) if $\textit{text-sim}(r_i, r_j) > \sigma$ and r_i, r_j belong to the same time sets (T^{in} or T^{out}). The edge type-3 means the reviews are similar both in content and published time.
- Add edge type-4 to (r_i, r_j) if $\textit{text-sim}(r_i, r_j) > \sigma$ and r_i, r_j belong to opposite time sets (T^{in} and T^{out}), respectively. The edge type-4 means the reviews are similar in content but fall into different time windows of published time.

The IRG construction procedure is shown in Algorithm 2. We apply grid search method [35] to determine the thresholds for IRG. We select the set of thresholds that best optimize the performance of the model according to the trade-off between detection accuracy and computational efficiency. The detailed results are presented in Section IV-E.

Algorithm 2 The IRG Construction Algorithm

Input: Target review dataset R_D, Movie dataset M_D,
Mapping function φ, χ, Threshold σ, ρ, ω for IRG.
Output: IRG.
Procedure:

```
for $r_i$ in $R_D$; $r_j$ in $R_D$ do
  if $\textit{text-sim}(r_i, r_j) > \alpha$ then
    if $\varphi(r_i) = \varphi(r_j)$ then
      add bidirectional edge type-1 to $r_i$ and $r_j$;
    end if
    if ($\varphi(r_i) = S^{high}$ and $\varphi(r_j) = S^{low}$) or ($\varphi(r_i) = S^{low}$ and $\varphi(r_j) = S^{high}$) then
      add bidirectional edge type-2 to $r_i$ and $r_j$;
    end if
    if $\chi(r_i) = \chi(r_j)$ then
      add bidirectional edge type-3 to $r_i$ and $r_j$;
    end if
    if ($\chi(r_i) = T^{in}$ and $\chi(r_j) = T^{out}$) or ($\chi(r_i) = T^{in}$ and $\chi(r_j) = T^{out}$) then
      add bidirectional edge type-4 to $r_i$ and $r_j$;
    end if
  end if
end for
```

2) Implicit Relation Enhanced Review Semantic Feature Extraction: In this way, we finally obtain the IRG based on the review set. Furthermore, we feed the constructed IRG into an HGT network to mine the implicit relation semantics among the reviews. Consequently, we obtain implicit relation enhanced review semantic features of each review, denoted as I_{review}. Note that we choose the HGT to extract embeddings because it can not only efficiently process web-scale data but also share parameters among meta relations with different occurrences. The experimental results in Section IV-F showed that this is crucial for uncovering hidden relationships between reviews in coordinated spamming attacks.

F. Classifier

1) Feature Fusion: Through the previous steps, we have obtained the review text embedding features E_{review}, review semantic credibility feature vectors C_{review}, and implicit relation enhanced review semantic features I_{review}. To better capture potential interactive information between the three feature types, we first use linear layers to scale the features to the same dimensional space. Then, we apply multi-head self-attention, denoted as $MultiHead$, to fuse the features.

2) Classification: Finally, we use a two-layer MLP to reduce dimensions of the fused features, which are then fed into a $Sigmoid$ function to calculate the probability of a review being spam. The entire feature fusion and classification process is defined as:

$$\hat{y} = Sigmoid(MLP(MultiHead(E_{review}, C_{review}, I_{review}))) \tag{4}$$

where $\hat{y}$ is the possibility that the review is a spam review.

G. Training

Our model training contains two stages: pre-training and end-to-end training.

In the pre-training stage, we fine-tune the character-level Chinese BERT pre-trained model on the Douban high-quality long review dataset, which contains 456 movies with a total of 19,666 reviews, as shown in Table II. Before feeding to the BERT model, we preprocess the reviews into a uniform length according to Section III-C1.

In the end-to-end training stage, we first use the fine-tuned BERT model to extract semantic embeddings for all nodes in the ERG and IRG as initial representations. For each batch of samples from the training set, we extract review semantic embeddings E_{review} using a BERT module and extract the corresponding movie factual concept representations C_{movie} from the ERG using an HGT module. We then use a contrastive network to measure the consistency between E_{review} and C_{movie} to obtain the review semantic credibility feature C_{review}. Moreover, for each batch of samples, we extract their implicit-relation enhanced semantic features I_{review}; specifically, we sample their one-hop neighbors [36] in IRG and propagate and aggregate the information by another HGT module in the sampled subgraph. In this way, we reduce the training time without undermining the performance of the model. Furthermore, we fuse the obtained E_{review}, C_{review}, and I_{review} features and calculate their probability as spam movie reviews using Eq. (4). Finally, we use binary cross-entropy loss to optimize the entire model, defined as:

$$L = -\sum_{r \in R_D} [y_r \cdot \log \hat{y}_r + (1 - y_r) \cdot \log(1 - \hat{y}_r)] \tag{5}$$

where r is a review of review dataset R_D; y_r is the label of review r; $\hat{y}_r$ is the possibility that r is a spam review.

The overall process of the proposed method is described in Algorithm 3.

Algorithm 3 The Overall Process of the Proposed Method

```
Input: Target review dataset set $R_D$, Movie dataset $M_D$,
    External review dataset $R_E$, Pre-trained $\mathrm{BERT}_{\mathrm{Chinese}}$,
    Untrained model $D_\theta$.
Output: Fine-tuned $\mathrm{BERT}_{\mathrm{movie}}$, Trained model $D_{spam}$.
Procedure:
% Pre-training;
fine-tuning $\mathrm{BERT}_{\mathrm{Chinese}}$ on movie domain corpus $R_E$;
% ERG Construction;
run Algorithm 1;
% IRG Construction;
run Algorithm 2;
% Training;
use $\mathrm{BERT}_{\mathrm{movie}}$ to extract semantic embeddings for all
    nodes in the ERG and IRG as initial vectors;
while not $convergence$ do
    random sample $batch_{review}$ from $R_{D\text{-}TRAIN}$;
    extract semantic features $E_{review}$, $C_{review}$, and $I_{review}$
        corresponding to $batch_{review}$, respectively;
    use Eq. (4) to fuse the features and calculate $\hat{y}$ for each
        sample in $batch_{review}$;
    update parameters $\theta$ of model $D_\theta$ by Equation 5.
end while
```

IV. Results and Evaluation

A. Experiment Setup

1) Hardware and Software Environments: We conducted our experiments on a workstation with two Intel(R) Xeon(R) Gold 6130 CPU, a RAM of 128G, and four NVIDIA Tesla V100 PCIe GPUs with GRAM of 32G. The software settings are Python v3.8.17, Pytorch v1.12.1, Torch-Geometric v2.3.1, Torch-Sparse v0.6.8, CUDA v10.2, and cuDNN v7.6.5.

2) Experimental Dataset: We randomly selected 19,000 spam reviews and 19,000 non-spam reviews from our previous publicly available M-Dataset[1] [5] to form a balanced dataset for experiments, denoted M-Dataset-38k. In Table II, we summarized the statistical details of the target datasets and the newly constructed external datasets, i.e., Douban Dataset[2] and Mtime Dataset[3]. In all experiments, we split the M-Dataset-38k with 60% of the total for training, 20% for validation, and 20% for testing. The training, validation, and testing sets for experiments are also balanced datasets, with an equal number of spam and non-spam reviews.

[1] The M-Dataset is collected from the Maoyan platform, which is one of the most popular online movie ticket sales and movie reviews platform in China. See https://www.maoyan.com/ for Maoyan.

[2] The Douban Dataset is collected from the Douban Movie platform, which is one of the most authoritative and widely used movie review databases in China. See https://movie.douban.com/ for Douban Movie.

[3] The Mtime Dataset is collected from the Mtime platform, which is one of the most authoritative and popular movie review databases in China. See http://www.mtime.com/ for Mtime.

TABLE II
STATISTICS OF DATASETS

Target Dataset	# Reviews	# Spam	# Non-spam	Length	Rating	# Movie	# Genre	Movie Released Year
M-Dataset	65,696	20,092	45,604	1-2,084	0-10	457	40	2017-2021
M-Dataset-38k	38,000	19,000	19,000	6-2,084				

External Dataset	# Reviews	Avg. Len	Min. Len	25% Len	50% Len	75% Len	Max. Len	# Movie
Douban Dataset	19,666	962	100	238	562	1,338	4,999	456
Mtime Dataset	6,904	790	100	100	488	1,244	4,955	411

3) Baselines and Training Settings: We select eleven models for comparison, including four popular deep-learning models for text classification, four state-of-the-art spam review detection models, and three state-of-the-art spam movie review detection models. We use the stochastic gradient descent (SGD) algorithm to optimize the models, with a learning rate of 6e-3 and the weight decay set to 0. Moreover, We train the models for 200 epochs with a batch size of 32 for each iteration and choose the epoch with the best F1-Score performance in the validation set for testing.

As for our proposed model, we use the same training settings as the baselines. Moreover, we conducted parameter sensitivity experiments in section IV-E to explore the hyperparameters' effects on the performance of our model. Specifically, we analyzed the trade-off between detection accuracy and detection time and obtained a set of optimal hyperparameters for our model. Therefore, in other experiments, we set the hyperparameters of our model to the optimal set obtained from section IV-E, as follows:

- We set the text truncation length δ_{text} to 64 for reviews in the dataset.
- In the ERG construction process, we set the long review node number per movie node ν_{review} to 3 and the text split length σ_{text} to 512 for the text content of all nodes.
- In the IRG construction process, we set the text similarity threshold σ to 0.4, the score similarity threshold ρ to 0.25, and the time similarity threshold ω to 7.
- For the HGT network, we set the attention head number α_{hgt} to 2 and the layer of HGT convolutions λ_{hgt} to 2. Before feeding three groups of features (i.e., E_{review}, C_{review}, I_{review}) to the multi-head self-attention feature fusion layer, we transform the dimension of these features to the same by using three linear layers. The output dimension of the three linear layers is defined as γ_{linear}, which is set to 128.
- Finally, we set the number of heads in the multi-head self-attention layer β_{head} to 8 and the dropout probability for hidden layers $\theta_{dropout}$ to 0.4.

4) Evaluation Method: We select four well-known metrics in computer science for evaluating the performance of models [37], i.e., **Accuracy**, **Recall**, **Precision**, and **F1-Score**. We run each experiment five times independently and record the averaged results.

B. Validity of Method

Table III shows that our method significantly outperforms other methods on our dataset, improving the average F1-Score by 8.5% over baselines. It demonstrates the efficacy of utilizing external reviews and fusing explicit/implicit relations for detection.

Compared to the second-best model [5], our model improves recall by 3.7% and precision by 3.6%. The result indicates that our model can enhance identification of spam while reducing misclassifying genuine reviews. The major performance gains stem from optimized graph construction and graph neural networks. These strengthen uncovering complex correlations between internal and external review systems. Specifically, the explicit relation graph integrates factual movie knowledge into reviews. The implicit relation graph reveals consistency patterns and anomalies. Together, these key innovations boost the detection of even sophisticated spam attacks.

The DSRHA model [20] detects spam reviews using document semantic embeddings. It captures sentence semantics through n-gram convolutions across word vectors. Then a BiLSTM aggregates sentence representations into a document embedding. They use English word2vec models to initialize word vectors. To adapt DSRHA to our dataset, we tokenize reviews with Jieba[4] and extract Chinese word2vec embeddings. However, performance is unsatisfactory at just 72.31% F1-score. Most reviews are short, rendering the sentence-document conversion ineffective. The sentence representation also does not transfer well from English.

This issue is evidenced by fastText [39] - character-level fastText performs much better than word-level, with 70.80% versus 60.13% F1-score. The sentence method struggles in Chinese contexts. One more to add, WSUN [13] is a GCN-based (Graph Convolutional Network) model which uses fastText to encode the reviews. Therefore, despite the complicated model architecture, it only achieves a detection performance of 73.36% F1-Score. Note that as the dataset used in this paper contains no tweeting history of users, we only use review embeddings without aggregation of user embeddings as initial embeddings of graph modules in the WSUN model.

CBRNN [42] faces similar language mismatch issues on our dataset. CBRNN is a model comprising two CNN layers and a BiGRU for movie review sentiment analysis tasks. We similarly tokenize review texts using Jieba and then obtain word embeddings from Chinese word2vec models as its input.

In contrast, TextCNN [38], Att-BLSTM [41], and DPCNN [40] are basic text classification models with structures similar to CBRNN and DSRHA. Nevertheless, as we input text embeddings from fine-tuned Chinese BERT models into these classifiers, their performance are generally better than the specialized spam review detection models. This highlights

[4] Jieba is one of the best Chinese text segmentation Python modules. See https://github.com/fxsjy/jieba/.

TABLE III
PERFORMANCE OF ALL CLASSIFIERS

Category	Method	Accuracy	F1-Score	Spam			Non-Spam		
				Precision	Recall	F1-Score	Precision	Recall	F1-Score
Baseline	TextCNN [38]	0.7924	0.7923	0.7952	0.7880	0.7916	0.7896	0.7967	0.7931
	fastText [39]	0.7080	0.7080	0.7012	0.7101	0.7056	0.7148	0.7060	0.7104
	DPCNN [40]	0.7598	0.7597	0.7452	0.7871	0.7656	0.7761	0.7326	0.7537
	Att-BLSTM [41]	0.8059	0.8059	0.8094	0.8031	0.8062	0.8023	0.8087	0.8055
Spam Review	CBRNN [42]	0.7519	0.7518	0.7595	0.7378	0.7485	0.7446	0.7659	0.7551
	DSRHA [20]	0.7231	0.7231	0.7313	0.7138	0.7224	0.7152	0.7326	0.7238
	BAM [7]	0.7640	0.7640	0.7846	0.7469	0.7653	0.7442	0.7822	0.7627
	WSUN [13]	0.7336	0.7336	0.7051	0.7723	0.7372	0.7658	0.6974	0.7300
Spam Movie Review	addCGAN [4]	0.7608	0.7607	0.7695	0.7596	0.7645	0.7519	0.7621	0.7570
	GAIM [6]	0.7883	0.7881	0.7945	0.7937	0.7941	0.7816	0.7825	0.7820
	GCMK [5]	0.8119	0.8118	0.8033	0.8305	0.8167	0.8213	0.7930	0.8069
	Proposed	**0.8483**	**0.8483**	**0.8581**	**0.8344**	**0.8461**	**0.8390**	**0.8621**	**0.8504**

the limitations of domain-specific expertise versus general language models.

The BAM [7] model achieves 76.40% F1-Score for detection performance. It extracts review embeddings using BERT and concatenates these embeddings with thirty manually extracted features. The aggregated features are then input into an Attention layer and MLP module for fusion before spam review detection. Since most manually extracted features used in their work depend on the dataset, we do not extract them for the dataset used in this paper.

The addCGAN [4] model uses GANs for unsupervised spam movie review detection. It utilizes review-user interest consistency. However, its semantic embedding relies heavily on observed vocabulary. The model also needs Douban-specific user features, especially average actor/director ratings. To adapt addCGAN, we replace its semantic extraction with fine-tuned BERT. But our dataset lacks 3 of 6 user features, including the critical ones. So performance is low at an F1-Score of 76.45%.

GAIM [6] is another state-of-the-art spam movie review detection model, which achieves an F1-Score of 78.81%. The model uses TextCNN to extract review embeddings. These are concatenated with manually extracted user activity features. The combined features are then fed into an MLP module and GAT network separately for further fusion. However, the manual features depend on the dataset. Since we use a different dataset in this paper, we do not extract these features.

Since BAM [7], WSUN [13], addCGAN [4], and GAIM [6] are dataset-dependent, we exclude them from further experiments.

C. Ablation Study

We conducted two ablation experiments to demonstrate the efficacy of the components in our proposed framework.

1) Model Ablation: In these experiments, we ablated components in the framework except the MLP. The experimental results are summarized in Table IV. We ablated the implicit review graph model, the explicit movie-review graph model, and the text model. Removal of these model structures would lead to the removal of corresponding features and significant decreases in model detection accuracy. Removing features I_{review}, C_{review}, and E_{review} resulted in F1-Score losses of 2.26%, 1.96%, and 1.98%. We can conclude that different features contribute differently to the performance of the model, with the I_{review} feature making the largest contribution.

Ablations of other model structures also undermine the performance of the model. Removing the contrastive network led to an over 1.2% decrease in Recall with almost no change in Precision. It demonstrates that significant semantic differences exist between the spam reviews and the professional reviews from the external system. Additionally, the ablation of the self-attention layer resulted in a decrease of 0.55% F1-Score, this indicates that implicit high-level interactions exist between the feature groups, which help detect spam effectively.

2) Graph Ablation: To effectively fuse information from movie synopses and high-quality reviews in the external review system, we defined the ERG comprising different meta-relations - $\langle movie, to, movie\rangle$, $\langle movie, to, review\rangle$, $\langle review, to, movie\rangle$ and $\langle review, to, review\rangle$. Although we introduced node heterogeneity in the above definitions, we did not consider edge heterogeneity. Hence, we explored the impact on the model performance of introducing edge heterogeneity into the graph structure. For example, we defined the 'to' relation in $\langle movie, to, movie\rangle$ as 'same topic with', which resulted in a notable decrease of the model performance (F1-Score dropped by 2.5%), even worse than defining a completely homogeneous graph. Moreover, the Precision for detecting spam reviews changed little, but a significant decrease was seen in the Recall (4.4%). This suggests that overemphasizing the fusion of factual information from similar-topic movies introduces excessive noise into the extracted movie factual concept knowledge. Similarly, introducing meta-relations like $\langle movie, commented\ by, review\rangle$, $\langle review, comments, movie\rangle$ and $\langle review, prior\ to, review\rangle$ and their combinations led to varying decreases in the model performance, indicating that while defining some heterogeneity is crucial for capturing essential movie factual knowledge, excessive definition can be detrimental.

Moreover, we explored removing the *Split-Node* operation during graph construction and external review nodes. Directly removing all external reviews performs slightly better (0.5% higher precision) than adding them without split nodes.

TABLE IV
PERFORMANCE OF ALL CLASSIFIERS IN THE ABLATION STUDY

Ablation Study	Ablation Settings	F1-Score	Accuracy	Recall	Precision
-	**full model**	**0.8481**	**0.8465**	**0.8565**	**0.8398**
Model Ablation	remove implicit review graph model (i.e., remove I_{reivew} feature)*	0.8255	0.8276	0.8226	0.8284
	remove explicit movie-review graph model (i.e., remove C_{reivew} feature)*	0.8285	0.8287	0.8301	0.8270
	remove text model (i.e., remove E_{review} and replace C_{reivew} with C_{movie} feature)*	0.8184	0.8211	0.8068	0.8304
	remove contrastive network (i.e., replace C_{reivew} with C_{movie} feature)*	0.8382	0.8342	0.8444	0.8393
	remove self-attention layer in classifier (i.e., remove feature fusion mechanism)*	0.8426	0.8411	0.8495	0.8358
Graph Ablation	add heterogeneity to relations in ERG	0.8237	0.8261	0.8127	0.8351
	remove *Split-Node* operator in ERG	0.8322	0.8328	0.8268	0.8376
	remove external review nodes in ERG	0.8327	0.8343	0.8231	0.8425
	replace unidirectional edges of review nodes with bidirectional edges in *Split-Node*	0.8456	0.8456	0.8457	0.8459
	remove inconsistency relations in IRG (type-2 and type-4)	0.8320	0.8320	0.8321	0.8320

* Since our model is an end-to-end model [43], [44], features are also extracted in end-to-end manners. Therefore, as the model structure is ablated, the features are also ablated.

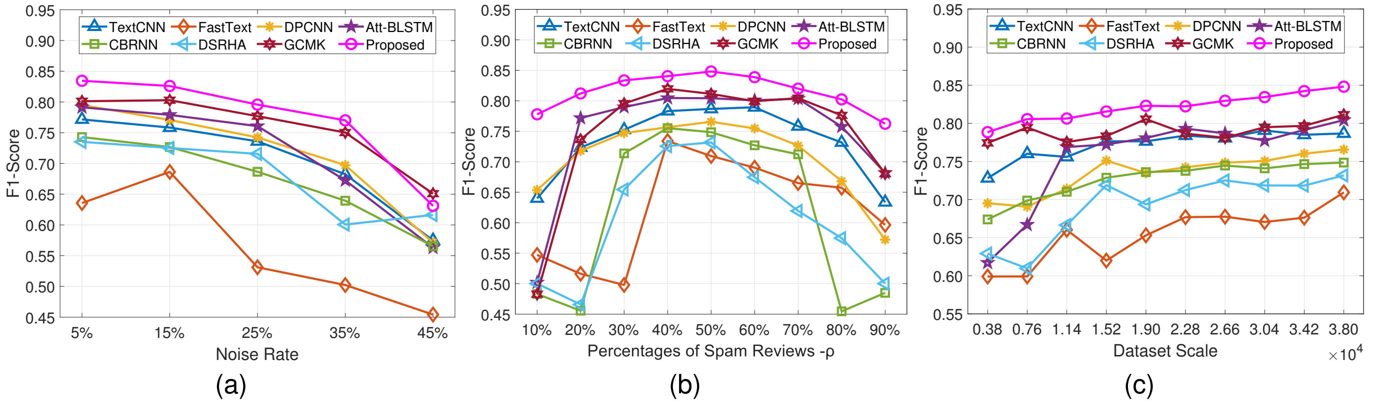


Fig. 3. Performance of all classifiers in model robustness study. (a) Robustness to data noise. (b) Robustness to data distribution. (c) Robustness to data scale.

This shows the first 510 characters of quality reviews have relatively little factual movie information. Simply truncating them brings useless information, weakening factual knowledge embeddings. Our proposed *Split-Node* approach effectively addresses this issue despite its simplicity. We also tried replacing unidirectional edges between review segmentation nodes with bidirectional edges. However, although the convergence speed of the model rises slightly, it leads to the model performance decreasing by around 0.3% in F1-Score. Additionally, removing the external review nodes in the ERG decreased F1-Score by 1.5%. Removing inconsistency relations in the IRG reduced F1-Score by 1.6%. This indicates these graph structures are important for detecting spam movie reviews.

D. Robustness of the Proposed Method

To investigate model robustness in real-world applications, we designed six robustness evaluation experiments.

1) Robustness Against Data Noise: In real-world deployment scenarios, data engineers try to improve data quality through repeated cleaning and annotation, but some mislabeled noisy data inevitably exists in datasets. A successful model should be able to learn the primary feature distribution of correct data samples and resist interference from a small portion of noisy samples. We trained models on datasets with 5% to 45% label errors at 10% intervals, recording their F1-Scores. Fig. 3(a) shows that when the data noise ratio is 5%, our model achieves around 84% performance, close to 84.81% on the noise-free dataset, which demonstrates good robustness for real deployment. As noise increases, the performance of our model drops much less than that of other models until the noise ratio reaches 35%, proving the robustness of our model against noise. We can also see some models exhibit abnormal performance increases as data noise grows, indicating they learned incorrect sample feature distribution patterns. In contrast, our model does not demonstrate such anomalies and keeps learning the correct patterns.

2) Robustness to Data Distribution: For most classification tasks, the ratio of positive samples often varies and is typically small in real-world scenarios. Therefore, it is essential to test model robustness by adjusting the positive sample ratio within datasets. Specifically, we maintained a constant dataset size of 22,000 and randomly sampled specific quantities of positive (spam reviews) and negative (non-spam reviews) samples from the M-Dataset. The ratios of positive samples ranged from 10% to 90%, in increments of 10%

Fig. 3(b) shows that our model achieves optimal performance on the balanced dataset with equal numbers of positive and negative samples. When the ratio of positive examples in the dataset varies between 20% and 80%, the detection accuracy of our model changes much less compared to other

TABLE V
ROBUSTNESS OF THE PROPOSED METHOD AGAINST TIME DRIFT

Model	**2017→2019**			**2017→2020**			**2017→2021**			**2017+18→2019**			**2017+18→2020**			**2017+18→2021**		
	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$	Pre	Rec	$F1$
TextCNN	0.76	0.76	0.76	0.73	0.71	0.71	0.73	0.71	0.70	0.76	0.76	0.76	0.73	0.72	0.71	0.74	0.71	0.70
fastText	0.57	0.58	0.56	0.60	0.60	0.60	0.57	0.56	0.57	0.53	0.53	0.54	0.61	0.60	0.61	0.56	0.56	0.56
DPCNN	0.74	0.73	0.73	0.73	0.73	0.73	0.72	0.71	0.71	0.73	0.73	0.72	0.73	0.73	0.72	0.69	0.68	0.68
Att-BLSTM	0.62	0.61	0.60	0.63	0.63	0.62	0.60	0.60	0.59	0.63	0.60	0.58	0.65	0.63	0.61	0.61	0.60	0.58
CBRNN	0.59	0.59	0.59	0.63	0.63	0.63	0.59	0.59	0.59	0.64	0.64	0.64	0.65	0.65	0.65	0.62	0.62	0.62
DSRHA	0.61	0.60	0.58	0.62	0.62	0.61	0.61	0.60	0.58	0.60	0.59	0.58	0.66	0.65	0.64	0.59	0.59	0.58
GCMK	0.79	0.79	0.79	0.79	0.77	0.77	0.77	0.74	0.73	0.80	0.80	0.80	0.77	0.76	0.76	0.78	0.77	0.76
Proposed	**0.81**	**0.81**	**0.81**	**0.82**	**0.80**	**0.80**	**0.80**	**0.78**	**0.77**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.80**	**0.79**	**0.79**

* Pre, Rec, and $F1$ represent **Precision**, **Recall**, and **F1-Score**, respectively.

models. The results indicate our model's superior adaptability to variations in sample ratios. Moreover, when trained on datasets with 10% and 90% positive sample ratios, our model performs much better than the compared methods.

3) Robustness to Data Scale: Data scarcity is a common challenge in most spam review tasks, including spam movie review detection. Many semi-supervised or unsupervised spam review detection methods have hence been proposed to address reliance on labeled data, with the first spam movie review detection method also being unsupervised. For supervised spam movie review detection, we desire models to have minimal reliance on data scale, achieving decent detection with small amounts of training data. Based on this, we designed experiments to evaluate model robustness against dataset scale. Specifically, we gradually reduced experiment dataset sizes by 10% of the original total volume under fixed dataset balanced ratios and train-test splits. Fig. 3(c) shows that our model's performance steadily declines slightly as data volume decreases. Even at just 10% of the original volume, our model outperforms other models trained on ten times more data. It demonstrates our model's excellent robustness to data scale, making it more capable of handling real-world spam movie review detection than other models.

4) Robustness Against Time Drift: As mentioned, both spam and non-spam reviews see explosive growth during a movie's release week. This mainly determines the overall public sentiment towards the movie and resulting platform ratings on the movie, both of which cap the future box office. Hence, timely detection is much more critical for spam movie reviews than other scenarios. Annotating datasets to train detection models for each new movie release is unacceptable regarding spam review detection timeliness and human cost. In order to test the robustness of our model against time drift, we divided the dataset into five portions from 2017 to 2021. We incrementally added newer yearly data to the earliest 2017 dataset for training. After each training completion, we tested models on future unseen yearly test datasets.

As shown in Table V, overall, our model maintains relatively better performance on unseen test years, with much stronger robustness against time shifts than other models. The model trained solely on the 2017 dataset still achieves an acceptable F1-Score of 77.35% on the 2021 dataset, just around 5% lower F1-Score compared to the "current year" model. (The "current year" here means the model is trained and tested on the 2017 dataset, which achieves an F1-Score of 82.48% on the test dataset.) As newer training data is gradually added, model performance on test datasets of future years improves steadily. Generally, performance degrades more on test datasets of further future years. This may be due to the "concept drift" issue faced by all text classification models. It could also be from new movie knowledge like filming, actors, genres, and online vocabularies. These cause pre-trained model embeddings to diverge from the actual semantic space.

5) Robustness to Diverse External Review Dataset: Since reviews from external review systems are used to construct the ERG graph in our model, the quality of external datasets could affect the performance of the model. To this end, we collected high-quality movie reviews from Douban and Mtime, the two most authoritative and widely used movie review websites in China, and created two distinct external datasets. For each movie, we randomly selected three reviews (see Section IV-A3) from these external datasets to build the ERG.

As shown in Table VI, our method performs similarly on both datasets, with slightly worse performance on the Mtime dataset. This could be due to the absence of 46 movies on the Mtime website, which leads to missing long review nodes for those movies in the ERG graph built from the Mtime dataset.

TABLE VI
ROBUSTNESS OF THE PROPOSED METHOD TO EXTERNAL DATASETS

External Dataset	**F1-Score**	**Accuracy**	**Recall**	**Precision**
Douban Dataset	0.8438	0.8438	0.8437	0.8440
Mtime Dataset	0.8409	0.8409	0.8409	0.8409

6) Robustness Against Advanced Spamming Technique: We also explored the robustness of our proposed method against advanced spamming techniques. Specifically, we consider testing our method against the machine generated reviews by the state-of-the-art large language model GPT-4 [45] in this work. To make the generated reviews as authentic as possible, we prompt the GPT-4 to act as a professional film critic and instruct it to generate reviews based on the synopsis of movies. The prompt is shown as follows: *"You are a seasoned film critic, and you will write a film review in Chinese based on the movies you have watched. Below is the name and synopsis of the movie. They are formatted in 'name$||$synopsis'."*

As shown in Fig. 4, our proposed method achieves an Accuracy of over 65% in identifying synthetic spam reviews.

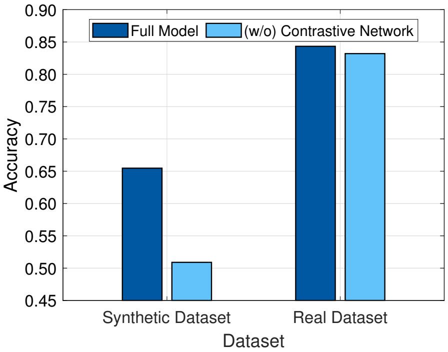


Fig. 4. Performance of the proposed method on synthetic and real datasets. The synthetic dataset consists of 446 GPT-4 generated reviews. The real dataset consists of spam reviews from target review dataset M-Dataset-38k.

The result is remarkable because the spam reviews generated by prompt-tuned GPT-4 closely resemble authentic reviews, making it nearly impossible for humans to differentiate between them. Moreover, we found that the contrastive network in the proposed method plays a crucial role in identifying such advanced spam reviews. Removing this lightweight component reduces the Accuracy of the model by nearly 15%. For comparison, we also tested the full model and the one without the contrastive network on the real-world spam review dataset M-Dataset-38k. We found that the contrastive network also contributes to spam detection by about 1% in Accuracy.

E. Parameter Sensitivity

In this section, we explore the impact of key hyperparameter settings on model detection performance, including model detection accuracy and detection time on the test dataset. In experiments, all the hyperparameters were set according to Section IV-A3, except the investigated hyperparameter.

1) Movie Review Text Truncation Length (δ_{text}) before Semantic Embedding Extraction: Review text lengths vary greatly, which makes it necessary to set an appropriate truncation length for preprocessing. Since BERT embedding extraction time grows approximately quadratically with input sequence length, we try to find a δ_{text} balancing time cost and detection performance.

As shown in Fig. 5(a), the detection accuracy of the model slowly improves as δ_{text} increases from 32 to 96, but sharply drops by around 1% at 128, and remains almost unchanged for larger δ_{text}. During the process, the prediction time of the model increases slowly from less than 10 seconds at $\delta_{text} = 32$ to 10 seconds at $\delta_{text} = 64$, and then accelerates significantly, reaching nearly 70 seconds at $\delta_{text} = 512$. Thus, we consider 64 to be a suitable truncation length δ_{text}. In this setting, the model achieves the highest prediction accuracy with a relatively short prediction time.

2) Number of Long Reviews (ν_{review}) for Each Movie in ERG Construction: In our framework, movie factual knowledge embeddings are extracted from the ERG constructed using high-quality reviews from the external review system and movie synopses from the target review system. To investigate the impact of review quantity, we increment ν_{review} by 3 within [0, 27] and observe performance changes. After removing the *Split-Node* operation from the graph, we also conducted the same experiments.

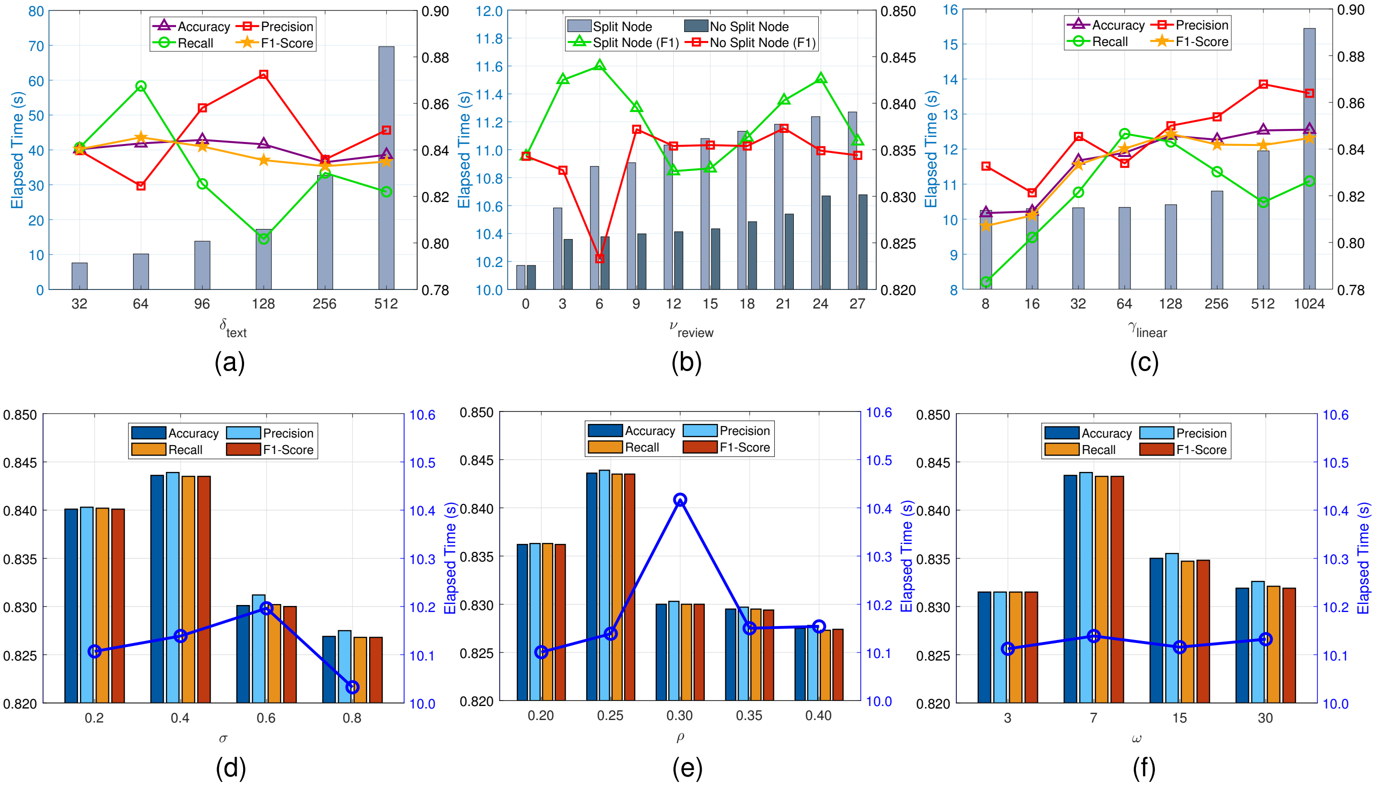
As shown in Fig. 5(b), incorporating the *Split-Node* operation during graph construction significantly improves model detection accuracy, indicating this operation can effectively aggregate movie factual knowledge from different positions of long reviews into movie node embeddings. Moreover, we can see the model already achieves outstanding results when ν_{review} is 3 or 6, meaning we only need to find 3 high-quality external reviews per movie to detect spam reviews accurately. Hence, our model has very low startup dependencies, easily satisfying the timeliness needs of review detection tasks. Furthermore, we can see that the performance of the model fluctuates rather than monotonically increasing as ν_{review} continues growing, likely because subjective detailed content in high-quality reviews is overlearned. Additionally, the detection time of the model is gaining rapidly as the ν_{review} increases. Therefore, we consider 3 to be an appropriate number of long reviews ν_{review}. The model under this setting achieves an F1-Score that is only about 0.1% lower than the best-performing model (with $\nu_{review} = 6$), yet it reduces the detection time cost by approximately 3%.

3) Output Dimension of Linear Layers (γ_{linear}) before Self-Attention Layer: We also explore the impact of the dimensionality of E_{review}, C_{review}, and I_{review} representations. The dimensionality represents the size of the information modeling space. Specifically, we double γ_{linear} from 8 to 1024 in powers of 2 and record the metric changes.

Fig. 5(c) shows that as γ_{linear} grows, the detection accuracy of the model generally improves, indicating larger modeling space allows review text embeddings to capture more fine-grained semantic details. Specifically, as γ_{linear} increases from 8 to 128, the model's detection time remains almost unchanged, while the prediction accuracy improves rapidly. However, when γ_{linear} continues to increase beyond 128, the detection time rises quickly, but the increase in prediction accuracy becomes very slow. Therefore, we consider 128 to be the appropriate dimension for the model's linear layer output γ_{linear}. It achieves a prediction accuracy only 0.3% lower than the best-performing model at 1024, yet it reduces detection time by nearly 33%.

4) Thresholds of Text Similarity (σ), Score Difference (ρ), and Time Window (ω) between Reviews in IRG Construction: We also explore how thresholds of parameters in IRG construction influence the performance of the model.

Fig. 5(d), Fig. 5(e), and Fig. 5(f) show that the impact of the three thresholds on model detection accuracy is more significant than their impact on model computational efficiency. The reason is that a slight deviation between reviews in text, score, and time leads to most reviews being divided into positive sets and negative sets, bringing too much redundant information. In contrast, a significant deviation divides most reviews into neutral sets, insufficiently capturing the consistency and inconsistent relationships between reviews. Therefore, we consider 0.4, 0.25, and 7 to be the appropriate values for thresholds σ, ρ, and ω. Under these settings, the model performs the best and achieves more than 0.5% accuracy than the second-best model, with less than 0.4% increase in detection time.

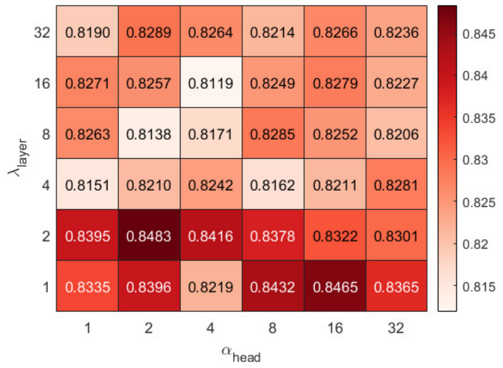


(a) (b) (c)

(d) (e) (f)

Fig. 5. Performance of the proposed method in parameter sensitivity experiments. (a) Performance variance of the proposed method under different movie review text truncation lengths. (b) Performance variance of the proposed method under different numbers of long reviews for each movie in ERG. (c) Performance variance of the proposed method under different output dimension of linear layers before self-attention layer. (d), (e), and (f) depict the performance variance of the proposed method under different thresholds of IRG.

5) Number of Heads in Attention Layers (α_{hgt}) and the Number of Layers (λ_{hgt}) for HGT Network: Finally, we investigated the impact on the performance of the two most important HGT hyperparameters, i.e., the number of attention heads α_{hgt} and layers λ_{hgt}. We cyclically set α_{hgt} and λ_{hgt} as $\{1, 2, 4, 8, 16, 32\}$, recording F1-Scores of the model, with results shown in Fig. 6.



λ_{layer} \ α_{head}	1	2	4	8	16	32
32	0.8190	0.8289	0.8264	0.8214	0.8266	0.8236
16	0.8271	0.8257	0.8119	0.8249	0.8279	0.8227
8	0.8263	0.8138	0.8171	0.8285	0.8252	0.8206
4	0.8151	0.8210	0.8242	0.8162	0.8211	0.8281
2	0.8395	0.8483	0.8416	0.8378	0.8322	0.8301
1	0.8335	0.8396	0.8219	0.8432	0.8465	0.8365

Fig. 6. Performance variance of the proposed method under different number of heads and different number of layers in HGT networks.

The performance of model is best with 1 or 2 HGT layers. It declines as depth of HGT network increases, likely due to over-smoothing in GNNs. Specifically, each movie node learns enough facts. But with more layers, the facts for each movie become obscured by others. This makes the review credibility features less targeted, thus degrading detection performance.

The impact of attention heads is less clear. Generally, more heads lead to better results with fewer layers and heads. This is likely because nodes need diverse signals from different dimensions to get representationally rich embeddings when their receptive fields are smaller.

F. Model Variations

In this section, we explore alternative architectures for two main components in the proposed model.

1) Comparative Analysis of Language Model Alternatives: In our approach, text models are essential, especially for extracting semantic features from reviews. To explore if any language model architecture is more suitable for this task than the Transformer architecture adopted in the proposed method, we replaced and compared PTMs of diverse architectures. For a fair comparison, all the PTMs were pre-trained on the same corpus and used the Whole Word Piece [23] for corpus tokenization. We adopted the optimal training and network configurations in the work of Zhao et al. [46] for pre-training these models. Moreover, we froze all the parameters of the PTMs during the training of the spam review detection model. As shown in Table VII, models using the BERT (Transformer architecture) delivered the best predictive performance, followed by ELMo (BiLSTM architecture), and then models with GatedCNN, LSTM, and GRU architectures.

TABLE VII
PERFORMANCE OF MODEL WITH DIFFERENT LANGUAGE MODELS

PTM Architecture	F1-Score	Accuracy	Recall	Precision
GatedCNN [46]	0.8247	0.8247	0.8248	0.8247
LSTM [46]	0.8081	0.8081	0.8079	0.8084
GRU [46]	0.8026	0.8026	0.8024	0.8028
ELMo [46]	0.8255	0.8259	0.8258	0.8261
BERT [46] (This Paper)*	**0.8328**	**0.8328**	**0.8329**	**0.8328**

* For fair comparison, we frozen all the BERT parameters during training.

2) Comparative Analysis of Graph Neural Network Alternatives: We experimented with various graph neural networks for extracting embeddings from the ERG and IRG graphs. As shown in Table VIII, models using heterogeneous graph neural networks achieve higher predictive accuracy compared to those using homogeneous graph neural networks. This is likely due to assigning different weights to different nodes and relations, which enhances the sensitivity of the model to spam indicators. Moreover, the performance of the model is better with the HGT compared to the HAN. This is likely because heterogeneous nodes and relations with few occurrences in the graphs could benefit from the parameter sharing from the nodes and relations of the majorities.

TABLE VIII
PERFORMANCE OF MODEL WITH DIFFERENT GRAPH NEURAL NETWORKS

GNN Architecture	F1-Score	Accuracy	Recall	Precision
GCN [31] (homo GNN)	0.8282	0.8306	0.8167	0.8401
GAT [32] (homo GNN)	0.8300	0.8334	0.8131	0.8477
HAN [33] (hetero GNN)	0.8348	0.8350	0.8347	0.8355
HGT [30] (This Paper)	**0.8481**	**0.8465**	**0.8565**	**0.8398**

V. DISCUSSION AND FUTURE WORK

Our proposed model, which integrates explicit and implicit relational semantics, shows significant performance gains over baseline methods. These gains are largely attributed to the accurate modeling of coordinated spamming attacks and the rich feature representations obtained from multi-view learning. However, there remain several limitations.

First, our model extracts high-dimensional contextual and relational semantic features of reviews through a multi-view approach, yet these features may be further optimized. On one hand, there may exist irrelevant and redundant features among these high-dimensional features. For this issue, a common practice in the domain of cyber security is to apply feature selection [47]–[49]. Therefore, we could optimize the features by feature selection methods, such as genetic algorithms [47] and the calculation of the information gain ratio [49]. On the other hand, the self-attention layer used in the proposed method for feature fusion yields unsatisfactory performance gains. Thus, techniques such as channel-wise feature scaling [44] could be considered for better feature fusion.

Second, optimizing the implicit relation graph model is also a fruitful direction. On one hand, in copy attacks, spammers copy the review content of normal users, resulting in spam and non-spam reviews being exactly the same or extremely similar. This poses challenges for data annotators in accurately identifying spam during dataset construction. Consequently, this can cause incorrect labeling, with actual spam reviews mistakenly labeled as non-spam. This uncertainty can substantially slow down the convergence and reduce the effectiveness of deep learning models. To address this issue, one potential approach is to integrate fuzzy logic units [50] into the design of graph neural networks. On the other hand, we can enhance the feature extraction from multi-view review subgraphs by various graph encoders and fuse the features with the variational gating mechanism [35]. This would allow for more fine-grained embedding representations.

Lastly, another limitation of the proposed hybrid model is its increased complexity. We found through experiments that the language encoder component accounts for the majority of the computational time during model training and inference. In the future, we will further explore more efficient Transformer architectures and investigate the practical applications of our model in processing large-scale, real-time data.

VI. CONCLUSION

This paper proposes a spam movie review detection method using explicit and implicit relations with multi-view semantic fusion. The explicit relation graph incorporates factual movie information into review semantics via external reviews and synopses. The implicit relation graph reveals consistency or inconsistency patterns in review metadata and content. Experiments show our method is robust and effective, improving F1-Score over baselines by 8.5% on average. We thoroughly explore the model in components and hyperparameters.

REFERENCES

[1] Y. Liu, W. Zhou, and H. Chen, “Efficiently Promoting Product Online Outcome: An Iterative Rating Attack Utilizing Product and Market Property,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1444–1457, 2017.

[2] S. He, B. Hollenbeck, G. Overgoor, D. Proserpio, and A. Tosyali, “Detecting Fake-Review Buyers Using Network Structure: Direct Evidence from Amazon,” *Proceedings of the National Academy of Sciences*, vol. 119, no. 47, p. e2211932119, 2022.

[3] H. S. Dutta and T. Chakraborty, “Blackmarket-Driven Collusion Among Retweeters–Analysis, Detection, and Characterization,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1935–1944, 2020.

[4] Y. Gao, M. Gong, Y. Xie, and A. K. Qin, “An Attention-Based Unsupervised Adversarial Model for Movie Review Spam Detection,” *IEEE Transactions on Multimedia*, vol. 23, pp. 784–796, 2021.

[5] H. Cao, H. Li, Y. He, X. Yan, F. Yang, and H. Wang, “GCMK: Detecting Spam Movie Review Based on Graph Convolutional Network Embedding Movie Background Knowledge,” in *Proceedings of the 31st International Conference on Artificial Neural Networks*, Bristol, UK, 2022, pp. 494–505.

[6] L. Zhang, X. Song, X. Zhao, Y. Fang, D. Li, and H. Wang, “GAIM: Graph-Aware Feature Interactional Model for Spam Movie Review Detection,” in *Proceedings of the 26th International Conference on Pattern Recognition*, Montreal, QC, Canada, 2022, pp. 621–628.

[7] Y. Jian, X. Chen, and H. Wang, “Fake Restaurant Review Detection Using Deep Neural Networks with Hybrid Feature Fusion Method,” in *Proceedings of the 27th International Conference on Database Systems for Advanced Applications*, Virtual Event, 2022, pp. 133–148.

[8] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, “ScoreGAN: A Fraud Review Detector Based on Regulated GAN with Data Augmentation,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 280–291, 2022.

[9] X. Tang, T. Qian, and Z. You, "Generating Behavior Features for Cold-Start Spam Review Detection With Adversarial Learning," *Information Sciences*, vol. 526, pp. 274–288, 2020.

[10] N. Wang, J. Yang, X. Kong, and Y. Gao, "A Fake Review Identification Framework Considering the Suspicion Degree of Reviews with Time Burst Characteristics," *Expert Systems with Applications*, vol. 190, p. 116207, 2022.

[11] Z. Zhang, Y. Dong, H. Wu, H. Song, S. Deng, and Y. Chen, "Metapath and Syntax-Aware Heterogeneous Subgraph Neural Networks for Spam Review Detection," *Applied Soft Computing*, vol. 128, p. 109438, 2022.

[12] Y. Zhu, H. Liu, Y. Du, and Z. Wu, "IFSpard: An Information Fusion-Based Framework for Spam Review Detection," in *Proceedings of the 30th Web Conference*, Virtual Event, 2021, pp. 507–517.

[13] Y. Xu *et al.*, "Mining Weak Relations Between Reviews for Opinion Spam Detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 31, pp. 152–162, 2023.

[14] S. Nilizadeh, H. Aghakhani, E. Gustafson, C. Kruegel, and G. Vigna, "Think Outside the Dataset: Finding Fraudulent Reviews Using Cross-Dataset Analysis," in *Proceedings of the 28th World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 3108–3115.

[15] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "DFraud^3: Multi-Component Fraud Detection Free of Cold-Start," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3456–3468, 2021.

[16] K. Burkholder, K. Kwock, Y. Xu, J. Liu, C. Chen, and S. Xie, "Certification and Trade-Off of Multiple Fairness Criteria in Graph-Based Spam Detection," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, Queensland, Australia, 2021, pp. 130–139.

[17] L. He, G. Xu, S. Jameel, X. Wang, and H. Chen, "Graph-Aware Deep Fusion Networks for Online Spam Review Detection," *IEEE Transactions on Computational Social Systems*, pp. 1–9, 2022.

[18] A. Fahfouh, J. Riffi, M. A. Mahraz, A. Yahyaouy, and H. Tairi, "A Contextual Relationship Model for Deceptive Opinion Spam Detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[19] V. Lai, H. Liu, and C. Tan, ""Why Is 'Chicago' deceptive?" Towards Building Model-Driven Tutorials for Humans," in *Proceedings of the 39th CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 2020, pp. 1–13.

[20] Y. Liu, L. Wang, T. Shi, and J. Li, "Detection of Spam Reviews Through a Hierarchical Attention Architecture with N-gram CNN and Bi-LSTM," *Information Systems*, vol. 103, p. 101865, 2022.

[21] Z. Shunxiang, Z. Aoqiang, Z. Guangli, W. Zhongliang, and L. KuanChing, "Building Fake Review Detection Model Based on Sentiment Intensity and PU Learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2023.

[22] S. Li, G. Zhong, Y. Jin, X. Wu, P. Zhu, and Z. Wang, "A Deceptive Reviews Detection Method Based on Multidimensional Feature Construction and Ensemble Feature Selection," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 1, pp. 153–165, 2023.

[23] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-Training with Whole Word Masking for Chinese BERT," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 29, pp. 3504–3514, 2021.

[24] X. Han *et al.*, "Pre-Trained Models: Past, Present and Future," *AI Open*, vol. 2, pp. 225–250, 2021.

[25] Y. Tay *et al.*, "Are Pretrained Convolutions Better than Pretrained Transformers?" in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Online, 2021, pp. 4349–4359.

[26] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Frozen Pretrained Transformers as Universal Computation Engines," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 36, Virtual Event, 2022, pp. 7628–7636.

[27] H.-K. Oh, S.-W. Kim, S. Park, and M. Zhou, "Can You Trust Online Ratings? A Mutual Reinforcement Model for Trustworthy Online Rating Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 12, pp. 1564–1576, 2015.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 23nd Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[29] X. Zhang, F. Wei, and M. Zhou, "HIBERT: Document Level Pre-Training of Hierarchical Bidirectional Transformers for Document Summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 5059–5069.

[30] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous Graph Transformer," in *Proceedings of the 29th Web Conference*, Taipei, Taiwan, 2020, pp. 2704–2710.

[31] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017, pp. 1–14.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada, 2018, pp. 1–12.

[33] X. Wang *et al.*, "Heterogeneous Graph Attention Network," in *Proceedings of the 28th World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 2022–2032.

[34] L. Hu *et al.*, "Compare to the Knowledge: Graph Neural Fake News Detection with External Knowledge," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Virtual Event, 2021, pp. 754–763.

[35] Y. Guo, D. Zhou, X. Ruan, and J. Cao, "Variational Gated Autoencoder-Based Feature Extraction Model for Inferring Disease-miRNA Associations Based on Multiview Features," *Neural Networks*, vol. 165, pp. 491–505, 2023.

[36] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, pp. 1025–1035.

[37] K. Grosse, L. Bieringer, T. R. Besold, B. Biggio, and K. Krombholz, "Machine Learning Security in Industry: A Quantitative Survey," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1749–1762, 2023.

[38] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1746–1751.

[39] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, 2017, pp. 427–431.

[40] R. Johnson and T. Zhang, "Deep Pyramid Convolutional Neural Networks for Text Categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 562–570.

[41] P. Zhou *et al.*, "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 207–212.

[42] S. Soubraylu and R. Rajalakshmi, "Hybrid Convolutional Bidirectional Recurrent Neural Network Based Sentiment Analysis on Movie Reviews," *Computational Intelligence*, vol. 37, no. 2, pp. 735–757, 2021.

[43] J. Shi *et al.*, "Two End-to-End Quantum-Inspired Deep Neural Networks for Text Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 4335–4345, 2023.

[44] Y. Guo, D. Zhou, P. Li, C. Li, and J. Cao, "Context-Aware Poly(A) Signal Prediction Model Via Deep Spatial–Temporal Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2022.

[45] OpenAI *et al.*, "GPT-4 Technical Report," 2024.

[46] Z. Zhao *et al.*, "UER: An Open-Source Toolkit for Pre-training Models," in *Proceedings of the 24th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, 2019, pp. 241–246.

[47] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, and G. Epiphaniou, "An Optimized Fuzzy Deep Learning Model for Data Classification Based on NSGA-II," *Neurocomputing*, vol. 522, pp. 116–128, 2023.

[48] S. Nakhodchi, B. Zolfaghari, A. Yazdinejad, and A. Dehghantanha, "SteelEye: An Application-Layer Attack Detection and Attribution Model in Industrial Control Systems Using Semi-Deep Learning," in *Proceedings of the 18th International Conference on Privacy, Security and Trust*, Auckland, New Zealand, 2021, pp. 1–8.

[49] J. Sakhnini *et al.*, "A Generalizable Deep Neural Network Method for Detecting Attacks in Industrial Cyber-Physical Systems," *IEEE Systems Journal*, vol. 17, no. 4, pp. 5152–5160, 2023.

[50] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, G. Srivastava, and H. Karimipour, "Secure Intelligent Fuzzy Blockchain Framework: Effective Threat Detection in Iot Networks," *Computers in Industry*, vol. 144, p. 103801, 2023.