

Detecting LLM-Generated Spam Reviews by Integrating Language Model Embeddings and Graph Neural Network

Xin Liu*

liuxin19@tsinghua.org.cn
 Tsinghua University
 Beijing, China
 liuxin4@supcon.com
 SUPCON
 Hangzhou, China

Rongwu Xu*

xrw22@mails.tsinghua.edu.cn
 Tsinghua University
 Beijing, China

Xinyi Jia

jiaxy21@mails.tsinghua.edu.cn
 Tsinghua University
 Beijing, China

Jason Liao

jliao8@student.ubc.ca
 University of British Columbia
 Vancouver, British Columbia, Canada

Jiao Sun

jiaosun.thu@gmail.com
 Google DeepMind
 Mountain View, California, USA

Ling Huang†

linghuang@fintec.ai
 AHI Fintech
 Beijing, China

Wei Xu†

weixu@tsinghua.edu.cn
 Tsinghua University
 Beijing, China

Abstract

The rise of large language models (LLMs) has enabled the generation of highly persuasive spam reviews that closely mimic human writing. These reviews pose significant challenges for existing detection systems and threaten the credibility of online platforms. In this work, we first create *three realistic LLM-generated spam review datasets* using three distinct LLMs, each guided by product metadata and genuine reference reviews. Evaluations by GPT-4.1 confirm the high persuasion and deceptive potential of these reviews.

To address this threat, we propose **FraudSquad**, a *hybrid detection model* that integrates text embeddings from a pre-trained language model with a gated graph transformer for spam node classification. FraudSquad captures both semantic and behavioral signals without relying on complex feature engineering or massive training resources. Experiments show that FraudSquad outperforms state-of-the-art baselines by up to 44.22% in precision and 43.01% in recall on three LLM-generated datasets, while also achieving promising results on two human-written spam datasets. Furthermore, FraudSquad maintains a modest model size and requires minimal labeled training data, making it a practical solution for real-world applications. Our contributions include new

synthetic datasets, a practical detection framework, and empirical evidence highlighting the urgency of adapting spam detection to the LLM era. Our code and datasets are available at: <https://anonymous.4open.science/r/FraudSquad-5389/>.

CCS Concepts

- Security and privacy → Intrusion/anomaly detection and malware mitigation.

Keywords

spam review detection, large language model, graph transformer

ACM Reference Format:

Xin Liu, Rongwu Xu, Xinyi Jia, Jason Liao, Jiao Sun, Ling Huang, and Wei Xu. 2025. Detecting LLM-Generated Spam Reviews by Integrating Language Model Embeddings and Graph Neural Network. In . ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/nmmnnnn.nmmnnnn>

1 Introduction

Online reviews play a pivotal role in shaping consumer decision-making and influencing business reputations [9]. However, the proliferation of spam reviews, i.e., deceptive content designed to mislead consumers, has emerged as a significant challenge [1, 18]. Fraudsters often control multiple accounts to post coordinated spam reviews, manipulating a target's reputation for profit [2, 3]. These fake reviews not only distort consumer trust but also erode the credibility of platforms like Amazon and Yelp. The economic impact is substantial, with spam reviews estimated to cost consumers and businesses around \$152 billion annually [4]. Detecting and curbing spam reviews is therefore essential for protecting consumer interests and ensuring the integrity of online review ecosystems.

However, detecting spam reviews remains challenging in two key aspects. *First*, recent advances in large language models (LLMs) [55], such as ChatGPT [29], Llama [12, 38, 39], and DeepSeek [5],

*Both authors contributed equally to this research.

†Both authors are corresponding authors.



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Conference'17, Washington, DC, USA

© 2025 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM

<https://doi.org/10.1145/nmmnnnn.nmmnnnn>

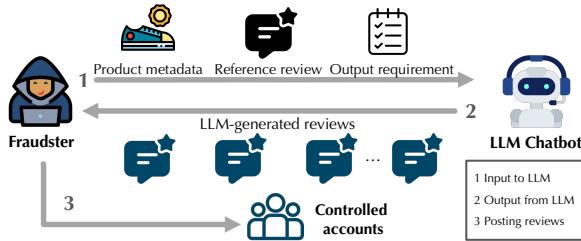


Figure 1: Workflow of LLM-generated review spamming. There are three steps. Step 1: Provide the LLM-based chatbot with the product metadata, genuine reference review texts, and specific output requirements. Step 2: Obtain the review texts generated by the LLM based on the input. Step 3: Use controlled accounts to post the generated reviews.

have made it easier to generate sophisticated, deceptive content [11, 43], heightening the urgency of effective detection. In particular, fraudsters can now exploit publicly available information to craft realistic fake reviews using LLMs [47]. As illustrated in Figure 1, a fraudster may input product metadata, genuine reference reviews, and specific output instructions into an LLM-based chatbot. The generated reviews are then posted through controlled accounts, making them appear authentic and difficult to detect.

Following the fraudster's workflow illustrated in Figure 1, we first constructed three LLM-generated spam review datasets to fill the lack of publicly available data. The generation pipeline was applied to three distinct LLMs using real-world review data from Amazon. Each LLM was prompted with product metadata, reference review texts, and specific output requirements to simulate realistic spam generation scenarios and generated 2,500 spam reviews. We employed GPT-4.1 to evaluate the quality of LLM-generated reviews. The assessment revealed that these reviews are *highly persuasive to potential consumers and closely resemble genuine human-written content*. Notably, the evaluation scores of the generated reviews surpassed those of genuine human-written reviews in being persuasive, detailed, convincing and influential on a five-point Likert scale. Furthermore, the generated reviews well met the output requirements specified by the fraudster, underscoring the urgency for developing accurate detection methods to mitigate the potential impact of such sophisticated spam content.

Second, existing spam/fraud detection methods often overlook the rich linguistic features embedded in review texts, which can provide crucial insights for identifying sophisticated spam reviews generated from LLMs. While graph-based approaches [16, 37], including recent advancements in Graph Neural Networks (GNNs) [53], have demonstrated effectiveness in capturing complex interactions within review graphs, they predominantly rely on *engineered features* derived from review contents to distinguish spamming reviews from fraudsters and non-spamming reviews from genuine users. This dependence limits their capacity to detect nuanced LLM-generated spam reviews that closely mimic authentic human writing.

To address the new challenge of detecting LLM-generated spam reviews, we propose a novel hybrid detection framework, namely **FraudSquad**, which integrates language model-enhanced node

Table 1: Prompt for generating the review texts using LLMs. Inputs related to product metadata, reference review texts, and output requirements are included in [].

I need your help to write reviews for a product [product name] on Amazon in the category of [product category]. The official description of the product given by the store is as follows: [product official description]. Besides, I will give you a set of review of this product for reference: [reference review texts].

Now, please output [review number] [positive/negative] reviews. Each review contains no more than [max word] words. Please write diversified reviews as if they were written by different customers, for example, with different lengths and styles. Start with another paragraph for each review and begin with Review 1. 2. 3., etc.

embeddings with graph neural networks. FraudSquad introduces two key innovations: first, it enriches node representations using text embeddings from a pre-trained language model; second, it employs gated graph transformers to capture relationships among review nodes within a constructed review graph for spam classification. This design allows FraudSquad to leverage both the linguistic content of review texts and user behavioral patterns—specifically, actions such as rating a product at a given time. By carefully selecting a lightweight language model for text embeddings, FraudSquad remains both efficient and effective, achieving high detection accuracy without relying on complex feature engineering [8, 26, 45] or extensive training resources.

Experimental results show that our method FraudSquad can accurately detect these LLM-generated spam reviews, outperforming state-of-the-art fraud detectors [8, 26, 45] by up to 44.22% in precision and 43.01% in recall. FraudSquad achieves overall metric scores of 89.45%-99.98% with only 1% annotated labels at training time. In addition, we find FraudSquad is also significantly more effective on two human-written spam review datasets. The ablation studies verify that advanced text embedding and graph structure are indispensable for accurate detection, saving the labor of maintaining engineered features without requiring massive training resources.

On the whole, our contributions are threefold:

- (1) We are the first to comprehensively study the **problem** of detecting LLM-generated spam reviews.
- (2) We synthesize three realistic **datasets** of LLM-generated spam reviews and evaluate the quality of the generated texts from multiple perspectives.
- (3) We propose a detection **model** FraudSquad that integrates language model embeddings and gated graph transformers, which achieves state-of-the-art detection performance without feature engineering or massive training resources.

2 Synthesizing LLM-Generated Spam Review Datasets

Since there are no public datasets specifically designed to test the detection of LLM-generated spam reviews, we create three datasets in this work. Each dataset contains the spam reviews generated from a distinct LLM. Basically, we simulate a **fraudster** by generating spam review texts using an **LLM** and letting **controlled accounts** post these reviews, as shown in Figure 1. It is important to note that we do not consider all LLM-generated reviews as spam by default.

Instead, what we define as spam is the *coordinated and repeated posting of such reviews by fake or controlled accounts across multiple products, typically with the intent of manipulation or profit*. In this process, LLMs are one *component* used to facilitate spamming.

Notation of review. We consider a review as a specific tuple that contains the user, review text, rating star, product, and timestamp.

2.1 Generating Spam Review Texts

Task description and the generation pipeline. Suppose a fraudster wants to post spam reviews for a specific product. The goal is to generate *review texts* that are both highly relevant to the target product and hard to distinguish from human-written ones. To this end, the fraudster could use an LLM-based chat assistant, providing it with detailed information about the product. This may include product metadata and reference review texts, both of which can easily be collected from e-commerce platforms like Amazon. In this work, we simulate such behavior by assuming the fraudster provides the following inputs to the LLM:

- *Product metadata*: Includes the product’s name, category, and the official description provided by the seller.
- *Reference review texts*: A set of genuine user review texts the product has received, which may be either positive or negative in sentiment.
- *Output requirements*: Specifies the desired sentiment, output number, maximum word count, diversity in content length and style, and formatting requirements (e.g., each review should appear as a separate paragraph).

The generation pipeline follows the prompt in Table 1. The fraudster sends a user message requesting to write reviews and providing the inputs marked within $\boxed{\cdot}$. The fraudster receives the assistant message from LLM with the generated spam review texts.

Generation setups. We apply this generation pipeline to the Amazon dataset, which is built upon the large-scale Amazon Review Dataset [17]. Specifically, we select reviews from the year 2022 across eight product categories: Baby Products, Video Games, Software, Musical Instruments, Appliances, All Beauty, Health & Personal Care, and Digital Music. In total, the derived Amazon dataset contains 7,617 products and 86,758 reviews.

We focus on generating *positive review texts* for *low-performing products*, i.e., those with the lowest average star ratings and the fewest reviews. Since over 75% of products in the Amazon dataset have an average rating above 4.3 on a five-star scale, it is reasonable to assume that products falling below this threshold may seek to improve their reputation. We randomly select 500 such products and generate five positive reviews for each, with a maximum length of 100 words per review. For context, each generation is guided by the first genuine review of the target product, used as a reference.

LLM selection. Three LLMs are leveraged for spam review text generation: (1) Qwen2-72B-Instruct [48] (Qwen2); (2) Llama3-8B-Instruct¹ (Llama3); and (3) DeepSeek-R1-Distill-Qwen-32B (Qwen-DSR1)². All models are open-source and implemented using the Ollama³ framework. This process results in three separate datasets

of LLM-generated spam review *texts*, which we later use to build the three final spam review datasets (see Section 2.3 for details).

2.2 Evaluating LLM-Generated Spam Review Texts

Persuasiveness and human-likeness. First, we aim to assess whether LLMs can generate spam review texts that are *highly persuasive to potential customers* and *closely resemble those written by real users*. To this end, we employ the advanced GPT-4.1 [30] model as an automatic evaluator [13] to rate the LLM-generated reviews across *five distinct dimensions*: (1) whether the review is clearly *positive*; (2) whether it is *detailed*; (3) whether it is *convincing*; (4) whether it appears to be written by a typical *human*; and (5) whether it is *influential* from a potential customer’s perspective. Each review is rated on a Likert scale from 0 to 5, where a higher score indicates better performance on the respective criterion. The full evaluation prompt is shown in Table 6. For comparison, we also randomly sample 2,500 five-star human-written reviews from the Amazon dataset and evaluate them using the same procedure. Figure 2 shows the evaluation results from GPT-4.1. All LLMs achieve consistently high scores (above 4) across all evaluation dimensions, with particularly strong performance in the positive and influential aspects. Notably, LLM-generated reviews outperform human-written reviews in all areas except for the human-like aspect, which shows slightly diverged results. Two of the three LLMs perform on par with or better than human-written reviews, while Qwen-DSR1 scores slightly lower. Overall, these results suggest that *LLM-generated reviews can easily be mistaken for genuine human-written ones and may significantly influence potential customers, potentially by spreading misleading information*.

Fulfilling output requirements. Second, we aim to evaluate whether the LLM-generated review texts meet the output requirements specified in the fraudster’s prompts. Table 2 summarizes the generation statistics. The number of outputted reviews is extremely close to the required number, with Llama3 and Qwen-DSR1 matching it exactly. This indicates that all LLMs follow the output format very reliably, enabling automatic extraction of review texts with a high success rate (over 99.5%). Besides, the word limit requirement is also well satisfied. In particular, Llama3 consistently produces reviews with fewer than 100 words, adhering strictly to the maximum length constraint. To assess the diversity of reviews generated for the same product, we calculate the average pairwise BLEU scores [33]. Lower scores indicate greater diversity. As shown in Table 2, all LLMs achieve low BLEU scores, suggesting that their outputs are varied and not repetitive. Additionally, a *manual check* of 100 randomly selected reviews confirms that the texts differ significantly in wording and highlight various aspects of the product. For qualitative examples, refer to Appendix B.

2.3 Creating the Final LLM-Generated Spam Review Datasets

To build the three final spam review datasets, *not just the review texts*, we simulate how fraudsters might inject fake reviews into a real review platform. Specifically, we assume that fraudsters take over some existing user accounts from the original Amazon dataset.

¹<https://github.com/meta-llama/llama3>

²<https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B>

³<https://github.com/ollama/ollama>

Table 2: Performance statistics of spam review text generation using LLMs demonstrate that the output requirements are well met. (1) The LLMs closely follow the specified output format, producing the required number of reviews with high accuracy. (2) The length of the generated reviews is consistently close to the maximum limit of 100 words. (3) The average pairwise BLEU scores among reviews generated for the same product are low, indicating a high degree of diversity in content and style.

LLM	Outputted/required	Max. #words	Avg. #words	Avg. pairwise BLEU
Llama3	2488/2500	94	56.5 ± 7.4	0.05 ± 0.03
Qwen2	2500/2500	133	54.5 ± 8.7	0.03 ± 0.02
Qwen-DSR1	2500/2500	102	60.5 ± 10.0	0.10 ± 0.04

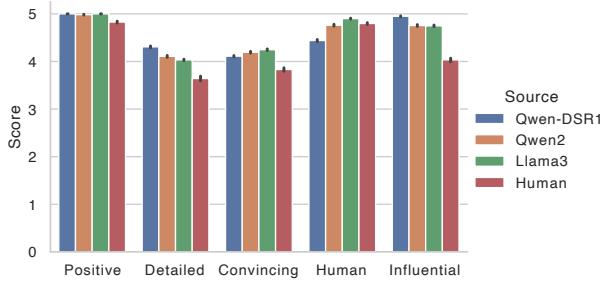


Figure 2: Evaluation results of LLM-generated and human-written review texts by GPT-4.1, rated on a Likert scale. The results show that LLM-generated reviews outperform human-written ones in terms of being positive, detailed, convincing, and influential. Moreover, they are often perceived as highly human-like in style and tone.

These accounts had previously posted normal reviews, but are now used to post the LLM-generated spam reviews, each user posting only 2 generated spam reviews. This setup creates a more realistic and challenging detection scenario [7].

We select compromised users based on how active they are—the more reviews a user has written, the more likely they are to be chosen. Each compromised user posts two fake five-star reviews for a target product. The posting time for these reviews is randomly chosen within five days of the product’s first real review, at any hour of the day.

In this way, we create three LLM-generated spam review datasets: Amazon-Llama3, Amazon-Qwen2, and Amazon-Qwen-DSR1. All three datasets contain the same genuine reviews in the Amazon dataset but different LLM-generated spam reviews.

3 FraudSquad: A Novel Approach for Spam Review Detection

In this section, we present a novel approach FraudSquad for spam review detection, including the problem formulation and the model architecture in detail.

3.1 Problem Formulation

To detect spam reviews by considering *both the review texts and fraudsters’ behavior*, we frame the problem as a node classification task on a review graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, T)$. In this graph, the node set $\mathcal{V} = v_1, \dots, v_N$ represents N individual reviews, and the edge

set \mathcal{E} captures relationships between these review nodes. Each review node $v_i \in \mathcal{V}$ is associated with a text T_i , which consists of multiple tokens (T_{i1}, T_{i2}, \dots) . Some of the nodes have known labels $y_i \in \{0, 1\}$, where 0 indicates a normal review and 1 indicates a spam (fraudulent) review. Typically, only a small portion of nodes are labeled for training, while most remain unlabeled. The goal is to predict which of the unlabeled nodes are spam, using both the graph structure and the content of the reviews.

3.2 The FraudSquad Detector

Our detection model, FraudSquad, consists of *four main components*: review graph construction, language model (LM)-enhanced node embeddings, gated graph transformers, and MLP layers for classification. The overall architecture of FraudSquad is shown in Figure 3. Compared to prior work, FraudSquad introduces *two key innovations*. First, while state-of-the-art fraud detection methods, such as CARE-GNN [7], PC-GNN [26], DGA-GNN [8], and GTAN [45], typically rely on feature engineering for node representation, FraudSquad leverages advanced language models to *directly process raw review texts*, enabling richer semantic understanding. Second, FraudSquad adopts a more capable architecture by using *gated graph transformers*. In contrast, prior approaches use traditional message-passing and aggregation mechanisms (e.g., CARE-GNN, PC-GNN, and DGA-GNN) or graph attention networks tailored for temporal graphs (e.g., GTAN). In the following paragraphs, we describe the design and components of FraudSquad in more detail.

Review graph construction. The review graph is built to *capture fraudster behavior*, specifically the act of users posting reviews on products, services, or other items. In typical review platforms like Amazon, we construct edges between review nodes based on three types of relationships, following established practices [7, 26]: (1) connecting pairs of reviews written by the same *user*; (2) connecting reviews of the same *product* that share the same *star rating*; (3) connecting reviews of the same *product* posted within the same *month*. This framework can be adapted to other scenarios as well. For example, in question-answering platforms where answers serve as reviews of the product mentioned in the question, the graph can be constructed using the following relationships: (1) connecting QA pairs under the same *question*; (2) connecting QA pairs where the *questions* are asked by the same user within the same *month*; (3) connecting QA pairs where the *answers* are given by the same user within the same *month*.

LM-enhanced node embeddings. The LM-enhanced node embeddings are designed to *capture the semantic features of review texts*. We construct the initial embeddings for each node in the

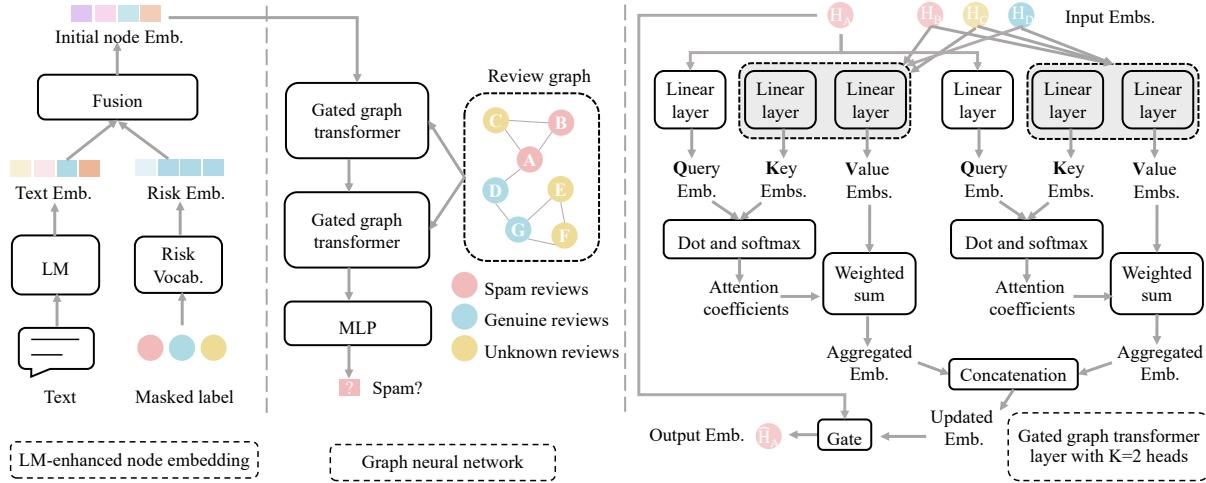


Figure 3: The overall architecture of FraudSquad by integrating LM-enhanced node embedding and graph neural network on the constructed review graph, where the review nodes are connected by various relations. The graph neural network in FraudSquad employs multi-headed gated graph transformer layers, which are also illustrated in this figure.

review graph as follows. First, each review text T_i is passed through a pre-trained language model (LM) with frozen weights to obtain its text embedding X_i in a latent space.

Next, we derive a trainable risk embedding Z_i based on the node labels to enable label propagation alongside feature propagation in the graph neural network. As per Shi et al. [36], integrating label and feature (i.e., text embeddings) propagation within the same GNN framework improves performance. The risk embedding vocabulary includes three classes, *normal*, *fraud*, and *unknown*, and the embedding dimensionality is set to match that of the text embedding. To avoid label leakage, in each training batch, the labels of the training nodes are all masked as *unknown*, so that these nodes can only aggregate the risk embeddings from their neighbors.

Finally, we combine the text embedding and the risk embedding using trainable weights $\beta_1, \beta_2, \beta_3$. The resulting initial node embedding H_i is computed as:

$$H_i = X_i + \text{PReLU}(X_i\beta_1 + Z_i\beta_2)\beta_3, \quad (1)$$

where PReLU [15] is a non-linear activation function.

Gated graph transformers. We input the initial node embeddings H_i into gated graph transformer layers with multi-headed attention.

The graph transformer architecture [10] uses three fundamental embedding vectors for each node v_i : Query (Q), Key (K), and Value (V), computed as follows for each attention head $s = 1, \dots, S$:

$$\begin{aligned} Q_{is} &= H_i W_{query}^s, \\ V_{is} &= H_i W_{value}^s, \\ K_{is} &= H_i W_{key}^s, \end{aligned} \quad (2)$$

where $W_{query}^s, W_{value}^s, W_{key}^s$ are learnable weights for the s -th attention head. For a given node v_i , we denote its set of neighbors as N_i . The attention coefficient between v_i and a neighbor $v_j \in N_i$ is calculated by measuring the similarity between their Q and K

embeddings:

$$\alpha_{ij}^s = \frac{\exp(Q_{is}^\top K_{js})}{\sum_{v_j \in N_i} \exp(Q_{is}^\top K_{js})}. \quad (3)$$

These attention weights determine how much influence each neighboring node's Value embedding contributes to the updated representation of v_i :

$$\tilde{H}_i^s = \sum_{v_j \in N_i} \alpha_{ij}^s V_{js}. \quad (4)$$

We then concatenate the outputs from all S attention heads to obtain the aggregated representation:

$$\tilde{H}_i = \text{Concat}(\tilde{H}_i^1, \dots, \tilde{H}_i^S). \quad (5)$$

Finally, we compute a shortcut projection from the linearly transformed input of the layer O_i using a gate mechanism following Xiang et al. [45]:

$$\begin{aligned} O_i &= H_i \beta_3, \\ \text{gate}_i &= \text{Sigmoid}(\text{Concat}(O_i, \tilde{H}_i, O_i - \tilde{H}_i) \beta_4), \\ \hat{H}_i &= \text{gate}_i O_i + (1 - \text{gate}_i) \tilde{H}_i. \end{aligned} \quad (6)$$

The resulting \hat{H}_i serves as the final output of one gated graph transformer layer.

MLP for classification. After passing through $L = 2$ gated graph transformer layers, each node embedding is fed into a multi-layer perceptron (MLP) to produce a probability score indicating whether the review is fraudulent (spam).

The entire FraudSquad model is trained using the labeled nodes with a binary cross-entropy loss. Optimization is performed using the Adam optimizer [19].

4 Experiments

To validate the effectiveness of our proposed method, we conduct comprehensive experiments on all three LLM-generated and two

Table 3: Statistics of datasets selected for experiments. The first three datasets are synthesized and contain LLM-generated spam reviews. The last two are publicly available datasets containing human-written spam reviews.

Dataset	Nodes	Edges	Spam nodes
Amazon-Llama3	89,186	4,139,448	2.8%
Amazon-Qwen2	89,192	4,140,166	2.8%
Amazon-Qwen-DSR1	89,197	4,138,569	2.8%
Yelp	5,854	141,123	13.3%
ChineseQA	133,317	66,272,741	34.2%

additional human-written spam review datasets. The following section outlines our experimental setup, main results, ablation studies, and a discussion on feature engineering.

4.1 Experiment Setup

Datasets. We evaluate our model on the datasets listed in Table 3. In addition to the three LLM-generated spam review datasets synthesized in Section 2, we also include two human-written spam review datasets. The Yelp dataset [34] is a public collection of Yelp hotel reviews from Chicago, with labels provided by Yelp indicating whether each review is filtered (spam) or recommended (normal). The ChineseQA dataset [27] is sourced from a Chinese community question-answering platform in 2015. It contains both genuine QA pairs and manipulated content created by large-scale crowdsourcing campaigns, often used for brand promotion or misinformation. These manipulated entries are treated as spam for detection purposes. All five datasets include ground-truth labels for evaluation.

Baselines. To demonstrate the effectiveness of our proposed hybrid spam review detection model FraudSquad, we compare it against several strong baselines. Baselines (1)–(3) are general-purpose classification models, while baselines (4)–(7) are state-of-the-art GNN-based fraud detection methods: (1) *MLP*: A multi-layer perceptron with two hidden layers that takes numerical features as input. (2) *RNN*: A recurrent neural network that processes the raw review texts as input. (3) *GAT* [40]: A graph attention network that performs node classification using node features and graph structure. (4) *CARE-GNN* [7]: A graph neural network designed to handle fraud camouflage by enhancing the aggregation process. (5) *PC-GNN* [26]: A GNN that addresses class imbalance issues commonly found in fraud detection tasks. (6) *GTAN* [45]: A gated temporal attention network developed for fraud detection, particularly in domains like credit card transactions. (7) *DGA-GNN* [8]: A dynamic grouping aggregation GNN tailored for fraud detection scenarios.

For models that require numerical node features (e.g., MLP and the GNN-based baselines), we use the engineered features provided in the original works [7, 34]. All baseline models, as well as FraudSquad, are implemented in PyTorch.

Training setups. We adopt a *minimally supervised setting* in our experiments, reflecting the real-world challenge of obtaining labeled data for the spam review detection task [49]. The dataset splits of the training-validation-testing are set as 1%-9%-90% for all datasets except ChineseQA. Given the larger scale of the ChineseQA dataset,

we use an even more challenging split: 0.1%-9.9%-90%. Besides, all methods, including FraudSquad run on one GPU with 48 GB of memory, showing a resource-constrained training environment.

For FraudSquad, we use BERT-base-uncased [6] as the language model to generate text embeddings. The gated graph transformer layers are configured with a hidden dimension of 100 and 3 attention heads. Training is run for up to 50 epochs. The gated graph transformer architecture is implemented using the Deep Graph Library (DGL) [42].

Evaluation metrics. We evaluate model performance using *precision*, *recall*, and *AUC* (area under the ROC curve). In specific, after the detection model assigns a probability score to each node indicating the likelihood of being spam, we identify the top-ranked nodes as predicted spam and compute precision and recall accordingly. In practice, fraud detection systems often require manual verification to avoid removing genuine content. Therefore, it is important to control the number of flagged candidates. To reflect this, we set the top-ranked prediction ratios based on the approximate spam prevalence in each dataset: 3% for all LLM-generated review spam datasets, 15% for Yelp, and 30% for ChineseQA.

4.2 Results on Detecting LLM-Generated Spam Reviews

Table 4 presents the detection performance on the three synthetic datasets containing LLM-generated spam reviews. For each metric, the best-performing method is highlighted in **bold**, and the second-best is underlined. Across all metrics and datasets, FraudSquad consistently achieves strong performance, with scores ranging from 89.45% to 99.98%, outperforming all baselines by a large extent. Though other methods, including RNN and DGA-GNN, could have relatively high AUC scores, their precision and recall scores are significantly low, which indicates that ***the top suspicious review nodes they predict are not accurate***. These results demonstrate that FraudSquad can effectively and accurately detect LLM-generated spam reviews. Under a challenging, minimally supervised setting and resource-constrained environment, FraudSquad remains highly effective in countering LLM-driven spam attacks. Notably, the language model used for node embeddings in FraudSquad is smaller than the generative LLMs that produce the spam, leading to a ***lower detection cost than the cost of generation***. This cost advantage helps reduce the economic incentive for carrying out such review spamming attacks.

Moreover, the three LLMs used to generate spam reviews show noticeable differences in their ability to evade detection, particularly against graph-based detectors. Among them, Qwen2 and Qwen-DSR1 exhibit stronger evasion capabilities compared to Llama3. For instance, when detecting spam reviews generated by Qwen2, GNN-based baselines achieve precision and recall scores barely above 50%. In contrast, these same baselines typically achieve over 70% precision and recall when identifying spam generated by the other two LLMs. Notably, FraudSquad records its lowest precision and recall when detecting Qwen-DSR1-generated spam, further highlighting the evasiveness of Qwen-DSR1. These observations suggest that Qwen2 and Qwen-DSR1 are more effective at generating hard-to-detect spam review texts.

Table 4: Detection performance (%) on both LLM-generated and human-written spam reviews. Pre: precision, Rec: recall.

Method	LLM-generated spam review datasets									Human-written spam review datasets					
	Amazon-Qwen2			Amazon-Llama3			Amazon-Qwen-DSR1			Yelp			ChineseQA		
	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec
MLP	79.37	35.26	37.80	69.72	8.64	9.29	92.50	45.93	49.16	61.05	21.27	24.00	77.42	63.58	55.79
RNN	96.41	48.14	56.01	94.07	68.86	57.46	96.64	75.52	58.13	55.78	12.60	2.29	58.53	32.74	5.31
GAT [40]	86.09	43.48	46.62	77.45	21.47	23.08	74.61	23.21	24.84	60.88	21.39	24.14	72.00	57.41	50.37
CARE-GNN [7]	89.69	44.48	47.68	95.63	45.68	49.11	95.11	48.01	51.38	54.80	18.73	21.14	78.78	68.47	60.08
PC-GNN [26]	83.43	31.98	34.28	94.69	42.86	46.07	96.45	51.29	54.89	59.16	22.28	25.14	78.68	66.96	58.76
GTAN [45]	90.00	45.14	48.40	96.91	62.87	67.59	97.95	64.87	69.42	60.25	21.27	24.00	79.81	69.63	61.10
DGA-GNN [8]	89.05	36.41	42.65	93.30	78.23	80.85	71.67	21.77	17.16	56.01	20.69	9.43	66.80	46.70	56.84
FraudSquad	99.98	92.36	99.02	99.94	90.99	97.81	99.93	89.45	95.73	70.32	33.67	38.00	99.43	99.91	87.67

4.3 Results on Detecting Human-Written Spam Reviews

In addition to detecting LLM-generated spam reviews, FraudSquad also achieves the best performance on identifying human-written spam reviews, as shown in the last two datasets in Table 4. However, we observe that detecting spam in the Yelp dataset is significantly more challenging than in the ChineseQA dataset, a trend consistent across all methods. On Yelp, detection metrics rarely exceed 70% and often fall below 20%. In contrast, detection scores on ChineseQA are generally above 60%, with FraudSquad achieving over 90%. One likely reason for this difference is that spam in the ChineseQA dataset exhibits clearer patterns—both in expressive features (e.g., user grades for askers and answerers) and in coherent spamming behaviors [27]. These features, when used as input to simple models like MLPs, already enable reasonable performance (e.g., precision and recall above 50%). Additionally, the ground-truth labels in Yelp are derived from more complex filtering mechanisms, making the spam signals more difficult to learn.

4.4 Ablation Study on LM-Enhanced Node Embeddings

We compare the effectiveness of LM-enhanced node embeddings with traditional feature-engineered node embeddings for spam review detection. We evaluate four language models for text embeddings: BERT-base and BERT-large [6], along with two recent embedding-focused LLMs—stella-400M and stella-1.5B [52], both of which rank highly in the classification track of the Massive Text Embedding Benchmark (MTEB) [28]. Considering ChineseQA is a Chinese dataset, we also include two Chinese embedding models: xiaobu-v2⁴ and Conan-v1 [23]. All models except stella-1.5B are under 1B parameters in size. The detection results using these LMs are shown in Figure 4. Overall, LM-enhanced embeddings significantly outperform feature-engineered embeddings across all four datasets, particularly on the three LLM-generated spam review datasets. For the human-written spam datasets Yelp and ChineseQA, engineered features perform reasonably well in the absence of graph information (i.e., without gated graph transformers), but still fall short of the

best-performing LM-based approaches. Among the models tested, BERT remains a strong performer, striking a good balance between classification accuracy and model size. Notably, for the Chinese-language ChineseQA dataset, when no graph structure is used, the two Chinese embedding models outperform BERT, underscoring the importance of language-specific pretraining corpora.

4.5 Ablation Study on Gated Graph Transformer

While text embeddings produced by language models can be directly fed into a linear classifier with reasonably good accuracy, incorporating user behavior through the constructed review graph provides additional benefits. To evaluate this, we perform an ablation study by comparing detection performance with and without the use of gated graph transformers, as shown in Figure 4. In the baseline setup, only text embeddings are used. The results reveal a clear performance gap across all datasets. The improvement is especially pronounced for the human-written spam datasets, Yelp and ChineseQA, highlighting the importance of leveraging relational information captured in the review graph.

4.6 Discussion: Are Engineered Features Still Necessary?

Finally, we investigate whether the combination of LM-enhanced embeddings and graph neural networks effectively *replaces the need for engineered features* that have historically played a key role in spam review detection. To explore this, we concatenate the engineered fraud features, derived from raw data by prior domain expertise, with the initial node embeddings before feeding them into the gated graph transformer layers. The performance differences are summarized in Table 5.

Results show that adding engineered features has only a *marginal* impact on detection performance, particularly for LLM-generated spam reviews. In many cases, the performance gap (Δ) is negligible or even slightly negative. This suggests that the current architecture already captures the core patterns these features were designed to highlight. That said, engineered features appear slightly more useful in detecting *complex human-written spam*, such as the Yelp dataset. Therefore, we recommend that practitioners assess the relevance of such features based on the specific types of spam

⁴<https://huggingface.co/lier007/xiaobu-embedding-v2>

Table 5: Impact of engineered features (EF) on spam review detection performance. Scores for all metrics are shown in %.

Setup	LLM-generated spam review datasets									Human-written spam review datasets					
	Amazon-Qwen2			Amazon-Llama3			Amazon-Qwen-DSR1			Yelp			ChineseQA		
	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec
With EF	99.99	92.82	99.51	99.51	85.67	92.10	99.90	91.99	98.44	68.85	32.41	36.57	99.37	99.83	87.60
Without EF	99.97	92.69	99.38	99.98	91.82	98.71	99.72	90.32	96.67	70.11	31.65	35.71	99.39	99.94	87.70
Δ	0.02	0.13	0.13	-0.47	-6.15	-6.61	0.18	1.67	1.77	-1.26	0.76	0.86	-0.02	-0.11	-0.10

attacks they expect to encounter. In general, we conclude that the integration of LM embeddings and graph-based modeling in FraudSquad largely subsumes the value of engineered features. Nevertheless, FraudSquad is flexible and can be easily *augmented* to incorporate such features when needed.

5 Related Work

Spam review detection has been widely studied, with research spanning from the design of meaningful features to the development of effective detection methods. Additionally, insights from misinformation detection offer valuable perspectives that can enhance spam detection approaches.

Features of spam detection. Numerous studies have explored the characteristics of fraudulent behavior, especially in the context of spam reviews [21, 24]. Rayana and Akoglu [35] provide a comprehensive list of engineered features commonly used for spam detection, forming the basis for many recent methods [7, 26]. There has been increasing interest in enriching feature sets by combining linguistic and behavioral signals, which often improves detection performance over using either alone. Integrating review graphs with metadata, capturing both textual and relational cues, has also been proven to be effective for more accurate and robust spam detection [35, 41]. More recently, Xiang et al. [45] improve detection effectiveness by jointly utilizing numerical and categorical attributes. Duan et al. [8] further addresses the challenge of non-additive attributes by employing decision trees to encode non-additive node attributes into binarized vectors.

Graph-based detection. Graph-based methods leverage the relationships among reviews, users, and products to enhance fraud detection. Early techniques like Graph Convolutional Networks (GCNs) [14, 20] have proven effective in capturing complex interactions within review graphs. Recent advances in Graph Neural Networks (GNNs) [7, 8, 26, 45, 46, 54] address key challenges such as class imbalance and heterophily. For example, spectral analysis has been integrated to enhance detection performance [46, 54]. Methods like CARE-GNN and its enhanced variant RLC-GNN tackle issues of relation and feature camouflage, achieving notable gains in fraud detection accuracy [51]. Additionally, the Pick and Choose Graph Neural Network (PC-GNN) effectively mitigates class imbalance by selectively sampling nodes and edges to construct balanced subgraphs for training [26]. Furthermore, Xiang et al. [45] propose a Gated Temporal Attention Network for fraud transaction detection, which is also applicable to spam review detection.

Linguistic-based detection. These methods focus on analyzing the textual content of reviews to identify deceptive patterns. Common techniques include sentiment analysis, syntactic analysis, and lexical feature extraction. For instance, sentiment and psycholinguistic features have been incorporated to achieve higher detection accuracy, though these models often struggle with sophisticated fraudsters who mimic genuine review characteristics [22, 31]. On the other hand, text embedding produced by language models are proposed to cover a range of tasks [28], including retrieval, summarization, classification and more, which could be useful in detection.

Misinformation detection. Misinformation generated by LLMs has received increasing attention [25, 32]. A notable example is the generation of targeted propaganda that closely imitates the style of legitimate news articles [50]. DECOR [44] is a recent method for fake news detection on the social graph. Wu et al. [43] also addresses the problem of fake news and proposes a framework against style-based attacks from LLMs. Feng et al. [11] is another work to detect bot accounts who utilize LLMs to evade detection in social media that post fake contents. While effective in the news domain, these approaches are specifically designed for that context and are not directly applicable to review-based scenarios.

6 Conclusion

In this work, we address the newly arisen challenge of detecting LLM-generated spam reviews by first synthesizing three realistic datasets that simulate how fraudsters might exploit LLMs to generate deceptive review content. Our analysis shows that these reviews are highly persuasive and closely resemble genuine human-written ones, motivating the urgency for robust detection methods.

To this end, we propose a novel hybrid spam detection approach, FraudSquad, which integrates language model-enhanced node embeddings with gated graph transformers to jointly capture linguistic cues and user behavior patterns. Experimental results demonstrate that FraudSquad is highly effective in detecting both LLM-generated and human-written spam reviews. These findings highlight the importance of combining semantic and structural signals to keep pace with increasingly sophisticated spam tactics.

Acknowledgement

This work started from the project in the Deep Learning course in Tsinghua University. We sincerely thank Professor Xiaolin Hu and Professor Jun Zhu for their support.

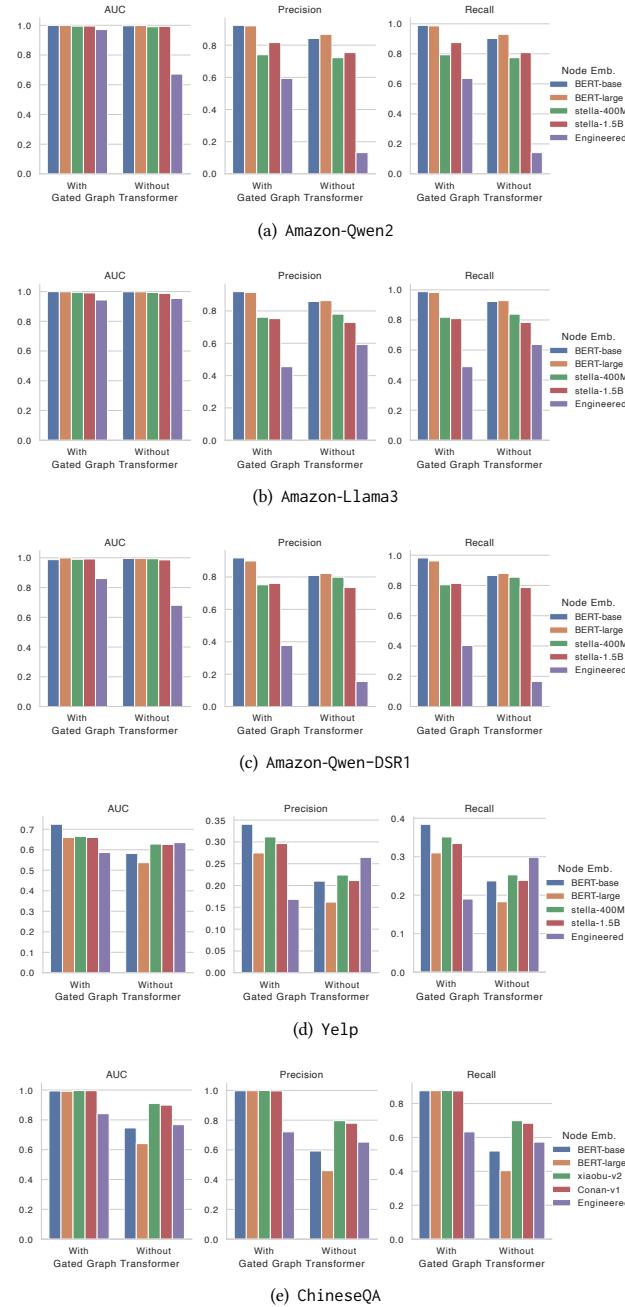


Figure 4: Ablation study on node embeddings (different bars) and gated graph transformer (with vs. without sets of bars).

A Detailed Prompts

Table 6 presents the complete prompt used to evaluate five-star spam review texts, whether generated by LLMs or written by humans. The prompt includes basic information about the target product, which is provided as input to GPT-4.1.

Table 6: Prompts for evaluating spam review texts. Inputs of the prompt are highlighted using [].

Role	Message
system	You are a helpful assistant and know a lot about e-commerce on Amazon, especially about how the reviews influence potential customers.
user	<p>Please first read a review about the product titled [product name] in the category of [product category]: [review to be evaluated]</p> <p>Now, please evaluate the influence of the given review on a potential customer on Amazon in the following five aspects:</p> <ul style="list-style-type: none"> - Will the user feel the review is positive? - Will the user feel the review contains useful details? - Will the user feel the review is convincing? - Will the user feel the review is written by a normal user? - Will the user be more willing to buy the product after reading the review? <p>For each question, please first answer with a rating ranging from 1 (totally no) to 5 (totally yes) and then give a brief reason for the rating.</p>

B Qualitative Examples of LLM-Generated Spam Review Texts

Tables 7, 8, and 9 provide qualitative examples of spam review texts generated by three distinct LLMs—Llama3, Qwen2, and Qwen-DSR1. These reviews target products from various categories within the Amazon dataset.

Table 7: Spam review texts generated by Llama3.

Category	Review text
Musical Instruments	I'm not a professional musician, just a hobbyist who loves to play for fun. I was blown away by the quality of these strings - they brought out the best in my guitar and made it sound like a completely different instrument. The price is very reasonable too.
All Beauty	I'm thrilled with the results I've gotten from Scruples Quickseal Detangling Conditioner! As someone who color-treats their hair regularly, it's amazing to find a product that not only detangles but also helps maintain my color vibrancy. This stuff has become a regular in my beauty routine - can't recommend it enough!
Baby Products	I was blown away by this pet gate! As a busy pet parent, I need solutions that work quickly and easily. This magic gate exceeded my expectations - it's a breeze to set up, sturdy, and does exactly what I need it to do: keep my furry friend safe indoors. Plus, the design is sleek and modern, making it a great addition to my home.
Video Games	What a fantastic product! I was skeptical at first, but the 60-day pre-paid time card has given me the freedom to play without interruption. The in-game content is top-notch, and the community is so engaging. A must-have for any Star Wars fan like myself!
Software	What a treat! M&M'S Adventure has become my go-to distraction when I need a break. The puzzles are cleverly designed, and the visuals are top-notch. I love how the game keeps me on my toes with new levels and characters to discover. It's also great that it's free, with no hidden costs or in-app purchases to worry about. A must-have for any puzzle game fan!

Table 8: Spam review texts generated by Qwen2.

Category	Review text
All Beauty	Absolutely in love! I've never felt my skin so soft after using an exfoliating product. The 100% PURE Cocoa Coffee Body Scrub is a game-changer. It feels like a spa treatment at home. The scent is divine, a warm and comforting aroma of cocoa that's not overpowering. The combination of sea salts and nourishing oils leaves my skin smooth as silk without any irritation. Highly recommend this for a luxurious pampering session!
Video Games	Absolutely thrilled with my purchase! The \$40 Xbox Gift Card was delivered instantly to my inbox, allowing me to download the latest games without any hassle. It's perfect for gamers who prefer digital copies and it's also great when you want to treat yourself or a friend to some new adventures.
Baby Products	As a new mom, finding the right products for my baby has been essential. The KAKIBLIN Pillow is truly a lifesaver! Its soft fabric combined with organic pp cotton ensures that my little angel gets the best comfort possible. Even during long drives, he seems relaxed and happy. The fact that it's machine washable adds to its convenience. A must-have for parents!
Software	Absolutely thrilled with my new Rii Wired Gaming Controller! It's super responsive and the dual-vibration/turbo feature really enhances my gaming experience on PC and Android. Love that it's plug and play, so easy to set up.
Health & Personal Care	I've been using this shaver for a couple of weeks now, and I'm impressed by how well it performs every time. The cordless design adds so much flexibility in my routine. It's also great that the battery holds up really well between charges. A must-have for on-the-go grooming!

Table 9: Spam review texts generated by Qwen-DSR1.

Category	Review text
All Beauty	This perfume exceeded my expectations! The blend of aromatic and woody notes creates a unique and captivating fragrance. It's perfect for everyday use and leaves a subtle, sophisticated trail. The bottle is beautifully designed, and the spray application is smooth. A fantastic addition to my fragrance collection!
Software	I'm thoroughly impressed with Wolf Quest! The high-quality 3D graphics and immersive gameplay make it a standout. I love the unlimited quests and the ability to grow a strong wolf pack. It's a fantastic mix of adventure and strategy. This game has quickly become one of my favorites!
Musical Instruments	This subwoofer is a game-changer for my home studio setup! The bass is deep, punchy, and incredibly clear. The build quality feels solid, and the adjustable crossover control allows me to fine-tune the sound to my liking. It's perfect for both music production and casual listening. Highly recommend this for anyone looking to upgrade their audio experience.
Baby Products	I absolutely love this gate! It's lightweight and easy to install, which was a huge plus for me. The size is perfect for our doorway, and the mesh is sturdy enough to keep our curious puppy contained. It also looks great in our home, blending seamlessly with our decor. A must-have for any pet owner!
Appliances	This gap cover is exactly what I needed to keep my kitchen looking neat and tidy. The silicone material is flexible and non-slip, so it stays in place perfectly. I love that it's easy to clean and can withstand high temperatures. It's a small but incredibly useful product that makes a big difference. I'm so happy with my purchase!

GenAI Usage Disclosure

In this work, generative AI (GenAI) tools were primarily used to create the three datasets of LLM-generated spam reviews, as already described in Section 2. These datasets were synthesized through a structured generation pipeline utilizing large language models.

Part of the experimental codebase was adapted from publicly available implementations [45], while the remaining components were developed independently. We used AI-assisted tools (Cursor more specifically) mainly to implement the computation of average pair-wise BLEU scores in review text evaluation.

The manuscript's core content was written by the authors. AI tools (GPT-4o) were used solely for minor editing and language refinement to improve clarity and fluency.

References

- [1] Giuseppina Andresini, Andrea Iovine, Roberto Gasbarro, Marco Lomolino, Marco Degennaris, and Annalisa Appice. 2022. Review Spam Detection using Multi-View Deep Learning Combining Content and Behavioral Features. In *The 1st Italian Conference on Big Data and Data Science (itaDATA)*.
- [2] Alex Beutel, WanHong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior In Social Networks. In *Proceedings of the 22nd International Conference on World Wide Web*. 119–130.
- [3] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering Large Groups of Active Malicious Accounts in Online Social Networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 477–488.
- [4] Max Chekalov. 2024. *The Economic Toll of Fake Reviews: Market Data and Prevention Strategies*. https://www.99firms.com/blog/the-economic-toll-of-fake-reviews/?utm_source=chatgpt
- [5] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shiron Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhusu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaoqiong Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghai Lu, Shangyan Zhou, Shanhuan Chen, Shengfeng Ye, Shiyu Wang, Shiping Yu, Shunfeng Zhou, Shuteng Pan, S. S. Li, Shuang Zhou, Shaoping Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanxia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xiaojin Shen, Xiaoshu Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaochu Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhanghi Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. [arXiv:2501.12948 \[cs.CL\]](https://arxiv.org/abs/2501.12948) <https://arxiv.org/abs/2501.12948>
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- [7] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020).
- [8] Mingjiang Duan, Tongya Zheng, Yang Gao, Gang Wang, Zunlei Feng, and Xinyu Wang. 2024. DGA-GNN: Dynamic Grouping Aggregation GNN for Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11820–11828.
- [9] Ramadhan Ally Duma, Zhendong Niu, Ally S. Nyamawe, Jude Tchaye-Kondi, and Abdulganiyu Abdu Yusuf. 2023. A Deep Hybrid Model for fake review detection by jointly leveraging review text, overall ratings, and aspect ratings. *Soft Computing* 27 (2023), 6281–6296.
- [10] Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699* (2020).
- [11] Shangbin Feng, Herun Wan, Ningnan Wang, Zhaoxuan Tan, Minnao Luo, and Yulia Tsvetkov. 2024. What does the bot say? Opportunities and risks of large language models in social media bot detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. 3580–3601.
- [12] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikalaidis, Damien Allonsius, Daniel Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thatai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Ilyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jieca Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasudevan Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuemley Chiu, Kunal Bhalla, Kushal Lakhota, Lauren Rantal-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadar, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Celebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronin Polidor, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapherty, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonget, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenying Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Couder, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Ayayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei

- Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swei, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Chen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelen, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madiam Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghatham Murthy, Raghu Nayani, Rahul Mitra, Ranaprabh Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadhi Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Tiberiu Mihaleaescu, Vladimir Ivanov, Wei Li, Wencheng Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaoqian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleiman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuqi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI]. <https://arxiv.org/abs/2407.21783>
- [13] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on ILM-as-a-judge. *arXiv preprint arXiv:2411.15594* (2024).
- [14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. arXiv:1502.01852 [cs.CV]. <https://arxiv.org/abs/1502.01852>
- [16] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. FRAUDAR: Bounding Graph Fraud in the Face of Camouflage. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 895–904.
- [17] Yupeng Hou, Jiacheng Li, Zhanhui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).
- [18] Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 219–230.
- [19] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014). <https://arxiv.org/abs/1412.6980>
- [20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations*.
- [21] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Learning to identify review spam. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three* (Barcelona, Catalonia, Spain (IJCAI'11)). AAAI Press, 2488–2493.
- [22] Fei Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2014. Towards a holistic approach to detect spam reviews in online review platforms. *Proceedings of the 23rd International Conference on World Wide Web* (2014), 459–470.
- [23] Shiyu Li, Yang Tang, Shizhe Chen, and Xi Chen. 2024. Conan-embedding: General Text Embedding with More and Better Negative Samples. arXiv:2408.15710
- [24] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) (CIKM '10). Association for Computing Machinery, New York, NY, USA, 939–948. <https://doi.org/10.1145/1871437.1871557>
- [25] Aiwei Liu, Qiang Sheng, and Xuming Hu. 2024. Preventing and Detecting Misinformation Generated by Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3001–3004.
- [26] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference 2021*. 3168–3177.
- [27] Yuli Liu, Yiqun Liu, Ke Zhou, Min Zhang, and Shaoping Ma. 2017. Detecting Collusive Spamming Activities in Community Question Answering. In *Proceedings of the 26th International Conference on World Wide Web*. 1073–1082.
- [28] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. *arXiv preprint arXiv:2210.07316* (2022). <https://arxiv.org/abs/2210.07316>
- [29] OpenAI. 2023. GPT-4 Technical Report. *OpenAI Report* (2023). <https://cdn.openai.com/papers/gpt-4.pdf>
- [30] OpenAI. 2025. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1/>. Accessed: 2025-05-22.
- [31] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 309–319.
- [32] Yikang Pan, Liangming Pan, Wenhui Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models. *arXiv preprint arXiv:2305.13661* (2023).
- [33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. <https://doi.org/10.3115/1073083.1073135>
- [34] Shebuti Rayana and Leman Akoglu. 2015. Collective Opinion Spam Detection: Bridging Review Networks and metadata. In *Proceeding of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining*.
- [35] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 985–994.
- [36] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*.
- [37] Tian Tian, Jun Zhu, Fen Xia, Xin Zhuang, and Tong Zhang. 2015. Crowd fraud detection in internet advertising. In *Proceedings of the 24th International Conference on World Wide Web*. 1100–1110.
- [38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwala Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esioubi, Jude Fernandes, Jeremy Fu, Wenying Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poultney, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]. <https://arxiv.org/abs/2307.09288>
- [40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the Sixth International Conference on Learning Representations*.
- [41] Jian Wang, Shuhua Feng, Bing Liu, and Yuming Li. 2017. Using a hybrid content-based and behavior-based featuring approach in fake review detection. In *Proceedings of the 2017 International Conference on Information Systems*. 849–861.
- [42] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2020. Deep Graph Library: A Graph-Centric,

- Highly-Performant Package for Graph Neural Networks. arXiv:1909.01315
- [43] Jiaying Wu, Jiafeng Guo, and Bryan Hooi. 2024. Fake News in Sheep's Clothing: Robust Fake News Detection Against LLM-Empowered Style Attacks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3367–3378.
- [44] Jiaying Wu and Bryan Hooi. 2023. DECOR: Degree-Corrected Social Graph Refinement for Fake News Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2582–2593.
- [45] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. 2023. Semi-Supervised Credit Card Fraud Detection via Attribute-Driven Graph Representation. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*. Article 1633, 9 pages.
- [46] Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. 2023. Revisiting Graph-Based Fraud Detection in Sight of Heterophily and Spectrum. *arXiv preprint arXiv:2312.06441* (2023).
- [47] Rongwu Xu, Xiaojian Li, Shuo Chen, and Wei Xu. 2025. Nuclear Deployed: Analyzing Catastrophic Risks in Decision-making of Autonomous LLM Agents. *arXiv:2502.11355 [cs.CL]* <https://arxiv.org/abs/2502.11355>
- [48] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Cheng-peng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671* (2024).
- [49] Hang Yu, Zhengyang Liu, and Xiangfeng Luo. 2024. Barely Supervised Learning for Graph-Based Fraud Detection. *Proceedings of the AAAI Conference on Artificial Intelligence* (Mar. 2024), 16548–16557.
- [50] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Defending against neural fake news. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Vol. 32. Article 812.
- [51] Yufan Zeng and Jiashan Tang. 2021. RLC-GNN: An Improved Deep Architecture for Spatial-Based Graph Neural Network with Application to Fraud Detection. *Applied Sciences* 11, 12 (2021), 5656.
- [52] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025. Jasper and Stella: distillation of SOTA embedding models. *arXiv:2412.19048*
- [53] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. 2020. GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 689–698.
- [54] Yu Zhang, Pang-Ning Tan, and Ying Ding. 2020. Fraud review detection using graph convolutional networks. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM, 2773–2781.
- [55] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* 1, 2 (2023).