Detecting Continuously Evolving Scam Calls under Limited Annotation: A LLM-Augmented Expert Rule Framework

Haoyu Ma[1], Qinliang Su[1,2*], Minhua Huang[3], Wu Kai[3],
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[3] China Mobile Internet Company Ltd.
{mahy28@mail2,suqliang@mail}.sysu.edu.cn
{huangminhua,wukai6}@cmic.chinamobile.com

Abstract

The increasing prevalence of scam calls, particularly on online platforms for recruitment, ride-hailing, and delivery services, has become a significant social and economic issue. Traditional approaches to scam call detection rely on labeled data and assume a static distribution of scam narratives. However, scammers continuously evolve their tactics, making these methods less effective. In this paper, we propose a novel approach leveraging large language models (LLMs) to detect continuously evolving scam calls. By abstracting scam and normal call rules based on expert knowledge, we develop a hierarchical few-shot prompting framework. This framework consists of a discrimination module to identify scam characteristics, a reflection module to reduce false positives by comparing with normal call features, and a summary step to synthesize the final detection results. Our method is evaluated on real-world and synthesized datasets, demonstrating superior performance in detecting evolving scam calls with minimal labeled data. Furthermore, we show that the framework is highly adaptable to new scam detection scenarios, requiring only modifications to the expert rules.

1 Introduction

With the development of network and communication technologies, voice calls have become an indispensable part of our daily lives. While bringing great convenience, they are also exploited by malicious actors to conduct scam activities, causing severe economic and social problems.(Alkhalil et al., 2021; Rao et al., 2021) Beyond conventional inter-personal calls, voice calls are also widely used by many online platforms specialized for recruiting, car-hailing, food/package delivery etc., under which relevant parties are often required to use platform-embedded calls to communicate with each other for better service assurance. However, due to the exposure of phone numbers, a surge of scam activities are observed on these platforms, *e.g.*, defrauding job seekers or e-commerce shoppers by pretending to be a recruiter or package deliverer. Hence, given the huge volume of daily calls, it is of great importance to detect scam calls automatically on these platforms.

To detect scam calls, Xu et al. (2022); Hong et al. (2023); Jiang (2024) propose to first convert calls into texts and then train a text classifier to discriminate between normal and scam calls. Despite promising results have been reported, these methods are generally built on two assumptions: 1) sufficient annotation of scam and normal calls; 2) training and testing datasets coming from the same distribution. But in practice, due to the high labeling cost and low occurrence rate of scam calls, assuming a large number of labeled scam calls is often unrealistic. More seriously, scammers do not stick with the same deceptive narrative over time, but instead will change them from time to time to enhance their credibility. Tabel 1 show two examples of a type of scams, in which scammers first pretend to provide opportunities for job seekers to switch to a well-paid industry and then persuade them to accept the associated job training scheme. As seen in the table, scammers in Period 1 frequently mention "*XXX Software Technology Company*" and "*software testing*", but later shift to narratives of providing opportunities in *live-streaming marketing* in Period 2 when live-streaming gains popularity. When the classifier is trained on data from Period 1, coupling with the availability of only a small number of labeled examples (*e.g.*, tens or hundreds), it easily renders the classifier to wrongly believe that "*Software*" is highly relevant to scams, making it hard to detect the scams in the narrative from Period 2. Obviously, this kind of methods fall short in recognizing continuously evolving deceptive narratives, even if the core deceptive trick is unchanged.

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5047–5068
November 4-9, 2025 ©2025 Association for Computational Linguistics

Example 1 (Period 1)	我们是xxx软件技术公司的，我们这边是做软件测试的，你之前有了解过这个行业么？没做过也没关系，我们会提供一个培训服务，我们这儿将来也会是一个重要行业，赚的也很多。 Trans: We are from XXX software technology company and we are engaged in software testing. Have you ever known about this industry? It doesn't matter if you haven't done it before. We will provide training services. This will be an important industry in the future and earn a lot of money.
Example 2 (Period 2)	我们是XXX公司，您这块做过直播带货的工作么，想要尝试一下么？我们前期会有老员工带带你，教你商品介绍、推销话术之类的，对对对相当于一个培训的过程。 Trans: We are XXX Company. Have you worked in live-streaming marketing before, or would you like to give it a try? We will have experienced employees guide you, teaching you product introductions, sales pitches, and so on. Yes, it's essentially a training process.

Table 1: Two example scam call texts from different periods, with the type "inviting individuals to transition to a new industry and offering training opportunities."

To have the methods better adapt to continuously evolving scenarios, a simple way is to couple with domain adaptation methods (Du et al., 2020; Wu and Shi, 2022; Zhang et al., 2023; Rostami et al., 2023) to learn domain-invariant features. But due to the significance of changes in scam narratives as well as the limited annotations, as seen in our experiments, these methods are largely ineffective, too. On the other hand, there also exist some methods that first predefine a list of keywords relevant to scams and then use the keyword-matching method to detect scams (Bajaj et al., 2019; Zhao et al., 2018). But due to the existence of countless ways to express the same meaning, these methods often miss to detect lots of scam calls. Recently, with the rise of LLMs, Jiang (2024) explored the ways of utilizing GPT-3.5 or GPT-4 to detect scam calls by simply providing the model with several normal and scam call demonstrations. However, it primarily leverages LLMs' few-shot classification ability, rarely taking advantages of LLMs' powerful generalization and reasoning capability.

Inspired by our human beings recognizing scam calls by checking whether the calls comply with a set of scam rules, in this paper, we propose to use LLMs to replace human beings' role in the process. To this end, we first abstract a set of judgment rules, which are expressed in natural language, based on experts' prior knowledge on scams for each type of scam calls. The set of rules are required to capture the main characteristics of scam calls. In addition to the scam rules, to promote the distinguishability, we also abstract a set of rules for normal rules. To better make use of these rules, we decompose the scam detection task into a series of reasoning steps, leading to a hierarchical few-shot prompting detection framework. Specifically, we first design a discrimination module to judge whether a call shows some characteristics of a specific type of scam calls by instructing LLMs to refer to the scam rules. Then, due to the high similarity between scam and normal calls, to reduce the false positive rate, we further propose a reflection module, which essentially asks LLMs to refer to normal call rules. Finally, a summary step is added to output the structured comprehension results. We experiment with our method on a real-world scam call dataset from an online recruitment platform and a synthesized scam call dataset, with the results showing significantly better performance in detecting continuously evolving scam calls with few annotations. We also demonstrate that our framework can be easily transferred to a new scam detection scenario by simply changing the expert rules. We publicize the synthesized dataset, which is modified from real calls, to foster the development of this valuable task.

2 Related Work

Telecommunication Scam Detection Telecom fraud detection faces several challenges (Bolton and Hand, 2002; Kou et al., 2004). Some existing methods (Wang et al., 2019; Hu et al., 2023; Liu et al., 2021; Hu et al., 2022) use graph neural networks (GNNs) to identify suspicious scammers or abnormal attributes in mobile social networks. However, our task involves textual call data rather than graph-structured data. For textual data detection, Hong et al. (2023) trained an LSTM model, while Xu et al. (2022) proposed a BiLSTM-Attention model to enhance feature extraction by focusing on scam-related keywords. Jiang (2024) fine-tuned a BERT model (Kenton and Toutanova, 2019) for scam detection. Additionally, Oyeyemi and Ojo (2024) integrated BERT with Naive Bayes for SMS spam detection, and Songailaitė et al. (2023) fine-tuned a BERT-based model (Liu, 2019) for phishing detection. Other methods (Bajaj et al., 2019; Zhao et al., 2018) rely on predefined keyword lists or scam patterns, restricting the model's focus. Recent work has explored LLMs for scam call detection, with (Jiang, 2024) investigating their effectiveness in identifying scam patterns, and (Shen et al., 2025) developing real-time detection systems using LLM-based approaches.

Prompting Method Large language models have demonstrated strong reasoning abilities, improving logical reasoning without requiring parameter updates (Brown et al., 2020; Kojima et al., 2022; Liu et al., 2023; Mishra et al., 2021). Chain of thought prompting method improves the model's perfor-

Types	Key Features
1	offering many job positions without details about the roles
2	encouraging people to switch to a new industry, often associated with deceptive training courses
3	offering free meals and accommodation for positions that typically do not provide such benefits
4	offering jobs close to the recruits' locations, regardless of where they are
5	claiming the company has many branches across the country

Table 2: Several common types of scams on online recruitment platforms

mance on complex reasoning tasks, such as arithmetic and commonsense reasoning tasks, by guiding the model to produce intermediate reasoning steps (Wei et al., 2022). Additionally, decomposing tasks into simpler subtasks has proven effective for complex problems (Khot et al., 2022; Zhou et al., 2022; Wang et al., 2023; Patel et al., 2022; Press et al., 2023). In our work, we apply a similar approach to scam call detection, breaking the task into subtasks and progressively incorporating expert rules to guide the model through different reasoning stages.

3 Problem Formulation

To mimic the task of detecting scam calls under continuously evolving scenarios, we suppose the training and testing call datasets $\mathcal{D}_{Train}$ and $\mathcal{D}_{Test}$ come from two different distributions. In practice, the two datasets could be obtained by collecting calls at two different time periods (*e.g.*, separated by several months). Both datasets contain a proportion of scam and normal calls, with the number of normal calls generally much larger than that of scam calls. The scam calls generally include several types, with the number denoted as K. For instance, scam calls collected from an online recruitment platform for low-end jobs include scam types: 1) persuading to accept job training by pretending to provide opportunities to switch to a new well-paid industry; 2) providing jobs with locations at any places as you want, etc. More types are shown in Table 2, with more examples of these scam types provided in Appendix A.3. The goal of this paper is to develop a framework that is capable of detecting scam calls from $\mathcal{D}_{Test}$ by only making use of $\mathcal{D}_{Train}$ and the prior expert knowledge.

4 Methodology

To accurately detect continuously evolving scam calls, we find that it is necessary to use large LLMs with huge parameters, partially because of their more powerful understanding and reasoning abilities, instead of using some simple and small LLMs. But due to the large volumes of daily calls, if large LLMs are employed to handle every call, the computational cost would be extremely expensive. To alleviate this issue, we note that most of the calls can be easily judged as normal with some relatively simple models. Thus, to reduce the computation cost, we propose to use some light-weight models to first filter out some certainly normal calls, and then only employ large LLMs to handle the difficult-to-judge calls, as seen in the overall model diagram in Fig. 1.

4.1 Prompting for Pre-Selection

For the pre-selection module, it is not necessary to require it achieving high detection accuracy, but should ensure it not to filter out lots of scam calls. There are many ways to realize it. In this paper, we simply use a small LLM to realize it, coupled with prompts designed to have relatively a more relaxed meaning. Specifically, we propose to assign each scam type a short name that loosely and broadly summarizes its key characteristics, serving as a manually defined abstraction based on expert knowledge rather than a precise scam indicator. For example, for scam type 1 in Table 2, we simply designate its name as "offering many job positions". Then, we simply prompt a small LLM to decide whether the call contains content relevant to the type name, as seen in the left of Fig. 1. Here, the relaxed type name enables the model to find calls that may be only loosely relevant to the scam, reducing the possibility of filtering out scam calls. In addition, to increase the detection accuracy, instead of detecting all scam types simultaneously, we propose to pre-select each type of scam calls separately, allowing the model to focus more on the specific characteristics of each type. In this way, we can obtain K small sub-datasets $\mathcal{D}_k$ for $k = 1, 2, \cdots, K$ from $\mathcal{D}_{Test}$, with sub-dataset $\mathcal{D}_k$ containing calls possibly relevant to the k-th scam type. Obviously, the datasets $\mathcal{D}_k$ will be much smaller than $\mathcal{D}_{Test}$ because lots of easy-to-judge normal calls have already been filtered out.

4.2 Prompting for Rule-Guided Detection

After the dataset size is significantly reduced, we then input the suspicious call into large LLMs for more precise and fine-grained analyses to determine whether it is a scam call. To better make use

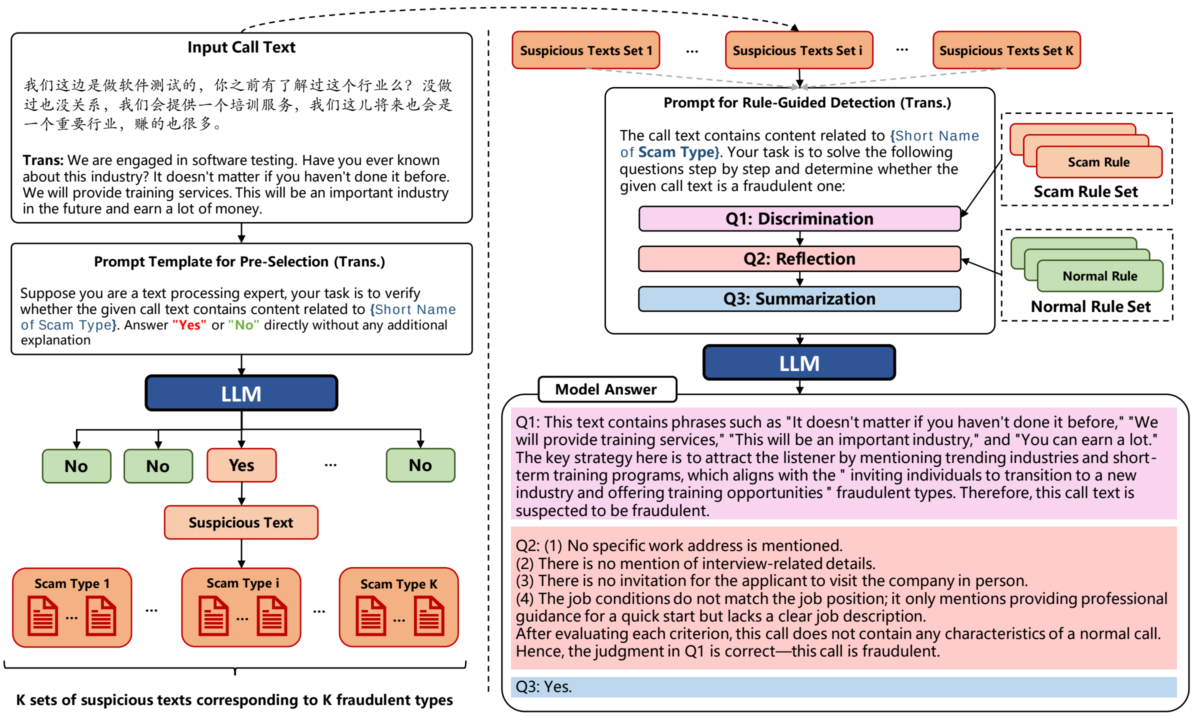


Figure 1: The diagram shows the detection process for a sample input. On the left, the pre-selection phase filters texts by scam type and stores suspicious content into K sets. On the right, the rule-guided detection phase further filters each set through three reasoning steps: discrimination, reflection, and summarization, to produce the final result.

of the detection rules on scam and normal calls, we propose to decompose the scam detection process into a series of reasoning steps. Specifically, we first design a discrimination module to judge whether a call shows the characteristics of a specific type of scam by instructing LLMs to consulting the scam rules. Then, to address the high similarity issue between scam and normal calls, we further propose a reflection module by asking LLMs to refer to normal call rules.

Discrimination Different from the pre-selection, the goal of this module is to determine whether a call is a scam or not as accurate as possible by making use of the powerful understanding and reasoning ability of large LLMs. To this end, for each scam type k, we first manually abstract a set of scam rules from experts' prior knowledge on scam, based on domain experience and generalizable patterns observed from representative cases, as

$$R_k^+ = \{r_{k1}^+, r_{k2}^+, ...\}\,, \tag{1}$$

where r_{ki}^+ denotes the i-th characteristic (*i.e.*, rule) that a scam call of k-th type may possess, and the rule r_{ki}^+ is expressed in natural language. In addition to the manually-crafted rules, we also supplement them by instructing LLMs to generate specific expressions of the rules with similar meanings, helping the model better understand the abstract rules, as shown in Fig. 2. The rule set R_k^+ can be expanded to accommodate newly emerging characteristics of scams, with its natural language formulation allowing seamless integration into the prompt and enhancing the model's adaptability to dynamic changing environments. As shown in the prompt below, we use the short name in pre-selection stage and its scam rule set to instruct LLMs to analyze whether the call is a scam or not. In the prompt, we also instruct the model to output the reasoning process, being consistent with the chain-of-thought method that emphasizes the importance of step-by-step reasoning. Please refer to Appendix B.2 for the concrete prompt form.

Prompt for discrimination

> Q1: Texts that use {Short Name of Scam Type} for fraud often exhibit the characteristic: {Scam Rule}. Based on this rule, analyze the text step by step to determine if it shows signs of fraud.

Reflection Because of the ambiguity between scam and normal calls, it is hard to conclude a call must be fraudulent if it contains some scam

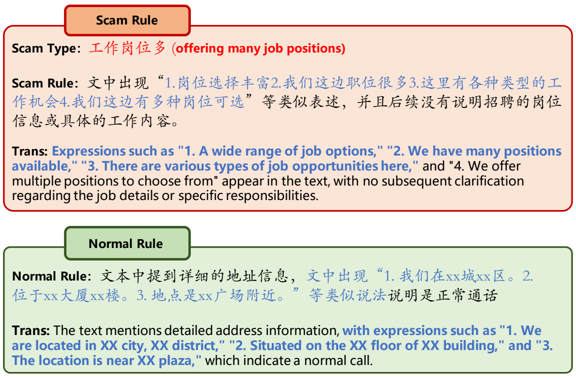


Figure 2: Example of a scam rule and a normal rule. The text in blue highlights the generated specific expressions by LLM.

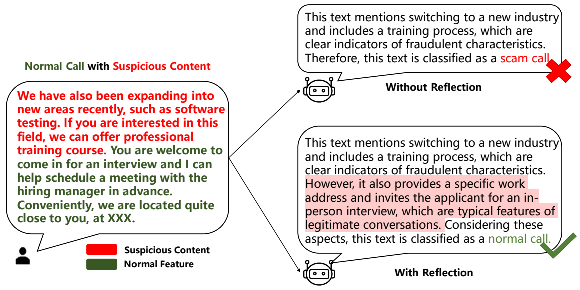


Figure 3: Illustration of the indistinguishable positives and negatives problem, where a normal call text containing suspicious content (red text) and normal features (green text). The right side highlights the difference with and without the reflection phase, with pink background areas indicating the model's reflection process.

characteristics. As a result, it is found that some calls filtered from the discrimination module are still normal calls, resulting in false positive. For instances, a call may contain contents of persuading someone to accept job training by pretending to provide opportunities to switch to a new well-paid industry, which is a suspicious scam characteristic, but it may also provide the detailed address information, which makes it more likely to be a normal call, as illustrated in Fig. 3. Thus, to increase the detection accuracy, we further propose to abstract a set of normal call rules from experts' knowledge, and then instruct LLMs to check whether the calls filtered by the discrimination module comply with them. If the compliance is confirmed, the calls, despite recognized by the the discrimination module as scam, can still be viewed as normal, as illustrated in Fig. 3. Specifically, we denote the set of normal call rules as

$$R^{-} = \{r_1^{-}, r_2^{-}, ...\} . \tag{2}$$

Similar to R_k^+, the rule set can be expanded as more normal characteristics are discovered. It is observed that these rules rarely appear together in a single call, thus we instruct LLMs to compare the calls' content with each rule one-by-one, as shown in the prompt below. Please refer to Appendix B.2 for the concrete prompt form.

Prompt for reflection

> Q2: Normal calls often exhibit the following characteristic: {Normal Rule}. If Q1 determines that the text shows signs of fraud, compare the text with the normal features one by one to check if it contains any normal characteristics.

Summarization To obtain a formatted answer for easier performance analysis or subsequent processing, we instruct the LLM to extract the final answer from the output generated in the discrimination and reflection steps.

In Rule-Guided Detection, we filter each $\mathcal{D}_k$ corresponding to k-th scam type, and then aggregate the filtered scam texts from the K scam types to obtain the final result. During inference, to help the model distinguish scam from normal calls, we select one example scam text and one example normal text for each scam type from the $\mathcal{D}_{train}$, and manually write the reasoning process as few-shot examples.

5 Experiment

5.1 Experimental Setups

Datasets and Metrics To evaluate the performance of our method, we conducted experiments on two Chinese call text datasets. First, we used a large-scale real-world call text dataset, **RealScam-Call** (Real-World Scam Call Dataset), which was collected and de-identified by a large online recruiting platform. However, due to privacy concerns, this dataset can not be publicly released. To foster further research in this field, we constructed and publicly released a synthesized call text dataset **SynthScamCall** (Synthetic Scam Call Dataset)[1], which is generated by using LLMs to rewrite real-world call texts. Although the real-world dataset cannot be shared, our synthetic dataset preserves the key characteristics of real-world calls, including the imbalance between normal and scam call texts, and consists of two subsets used for training and

[1]The dataset is openly accessible at: https://github.com/WsgDcb/SynthCallScam_Dataset

Paradigm	Model	Method	SynthScamCall			RealScamCall		
			Precision	Recall	F1-score	Precision	Recall	F1-score
Classification	RoBERTa	Finetuned-BERT	0.111	0.446	0.178	0.050	0.005	0.009
		BERT-AT	0.138	0.466	0.212	0.034	0.167	0.056
		BERT-AT+ELS	0.139	0.532	0.221	0.033	0.205	0.057
		AdSPT	0.145	0.501	0.225	0.045	0.296	0.078
Few-shot	Qwen-long	SP	0.581	0.813	0.677	0.073	0.487	0.127
		PS	0.519	**0.928**	0.665	0.056	0.450	0.100
		CoT	0.654	0.872	0.695	0.062	0.497	0.110
		CoT+RE2	0.577	0.825	0.679	0.063	0.468	0.111
	Deepseek-v3	SP	0.437	0.847	0.577	0.055	0.439	0.098
		PS	0.609	0.896	0.725	0.066	0.529	0.118
		CoT	0.509	0.914	0.654	0.071	0.570	0.127
		CoT+RE2	0.602	0.892	0.719	0.071	0.568	0.126
	Deepseek-R1	SP	0.501	0.825	0.623	0.057	0.451	0.102
		PS	0.597	0.883	0.713	0.067	0.545	0.120
		CoT	0.566	0.896	0.694	0.081	0.592	0.143
		CoT+RE2	0.605	0.914	0.728	0.083	0.581	0.146
Few-shot + Prior Rule	Qwen-long	Ours	**0.824**	0.881	**0.852**	0.347	0.710	0.466
	Deepseek-v3	Ours	0.709	0.901	0.794	**0.358**	0.707	0.475
	Deepseek-R1	Ours	0.739	0.901	0.812	0.357	**0.712**	**0.476**

Table 3: Performance comparison among traditional pre-trained language model-based methods, traditional prompting methods, and our proposed method on the SynthScamCall and RealScamCall datasets. **SP** stands for Standard Prompting and **PS** stands for Plan-and-Solve Prompting. The best result is marked in bold.

testing, respectively. Each subset represents data collected at a different period and is constructed by modifying attributes irrelevant to scam (e.g., industry names) to simulate the continuously evolving scam tactics, thereby reflecting the data distribution shift between data collected at different periods in real scenarios. Both datasets contain the same 5 scam types. More details of the used datasets and their construction are shown in Appendix A.2. For evaluation metrics, we use precision, recall, and F1-score to test the model's performance on the two datasets.

Baselines The baselines for this experiment are divided into two categories: i) traditional methods based on pre-trained language models, and ii) prompting methods leveraging large language models. For traditional methods based on pre-trained language models, we tested the text classification method Finetuned BERT and domain adaptation methods: BERT-AT (Du et al., 2020), BERT-AT+ELS (Zhang et al., 2023), and AdSPT (Wu and Shi, 2022). For prompting methods, we tested the standard prompt (Brown et al., 2020) where the in-context demonstration includes only the sample and answer. Additionally, we tested chain of thought prompting (Wei et al., 2022), plan-and-solve prompting (Wang et al., 2023) and Re-Reading Prompting (CoT+RE2) (Xu et al., 2024) under few-shot setting, both of which have been proven to enhance the LLM's ability to solve complex problems.

Implementation Details For traditional methods based on pre-trained language models, we select the Chinese Whole Word Masking RoBERTa models pre-trained by TencentPretrain (Apache-2.0 License) (Zhao et al., 2023) and UER-py (Apache-2.0 License) (Zhao et al., 2019) as the base model or encoder and we train the model using D_{train} and evaluate it D_{test}. For more training details (detailed methods, hyperparameter etc.) refer to Appendix B.1. For prompting methods, we used the Qwen-long, Deepseek-v3 (MIT License) (Liu et al., 2024) and Deepseek-R1(MIT License) (Guo et al., 2025) APIs for both baseline and rule-guided detection, while ERNIE-speed API was called for pre-selection in our method. The specific API parameter settings are provided in the Appendix B.1. For the number of examples, we employ a 2-shot setting for each scam type in our method, consisting of one normal example and one scam example, all selected from D_{train}, with a total of 5 scam types. To ensure fairness, other prompting methods are tested with 10-shot using the same examples. To prevent the model from labeling a large number of normal calls as scam, which leads to inflated recall, we set a maximum limit on the number of calls that can be identified as scam. More details of the implementation of the prompting methods are shown in the Appendix B.2. The experimental results are the averages from five random runs.

5.2 Experimental Results

The experiment results of pre-trained language model-based methods, traditional prompting methods, and our method on the SynthScamCall and RealScamCall datasets are shown in Table 3. First, prompting methods consistently outperforms traditional pre-trained models on both datasets. In the more challenging RealScamCall dataset, where all methods experience a performance drop, prompting methods still achieve better results. This suggests that traditional methods struggle to learn key scam-related features from limited annotations, while LLM-based prompting maintains more stable performance through its inherent generalization ability. Second, comparing our method with other prompting methods, our method outperforms other prompting methods on both datasets. Especially on the RealScamCall dataset, the low precision of other methods indicates their difficulty in accurately identifying scam calls, leading to frequent misclassifications. By introducing a scam rule set, we help the model better identify scam call features, while the normal rule set prevents misclassification of indistinguishable normal calls. This improves precision by nearly 0.3, helping the model handle data distribution shifts and indistinguishable cases more effectively. Third, we validated our method on Deepseek-v3 and Deepseek-R1, where it also showed strong performance, demonstrating its stability across different models. However, as a reasoning model, R1 requires significantly longer inference time compared to DeepSeek-v3 and Qwen-long, without delivering noticeable performance gains, suggesting that it may be less suitable for this task.

Furthermore, to validate that our method can quickly adapt to a new scam detection scenario, we collected call data from a new scenario that includes a new scam type, characterized by "using fake advertising to entice the caller to purchase a product or participate in an event." We tested our

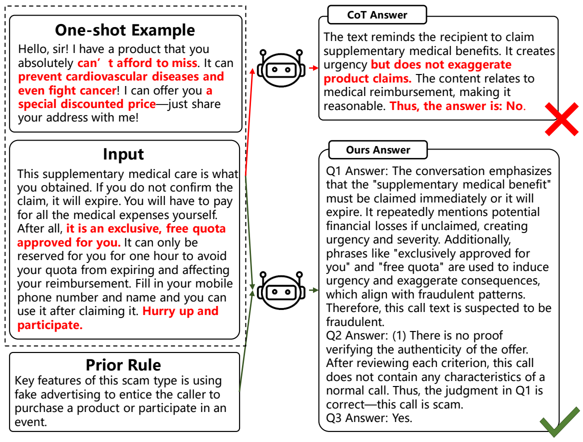


Figure 4: Case study comparing our method with CoT prompt. While CoT gets an incorrect answer, our method gets a right answer with the help of prior rule.

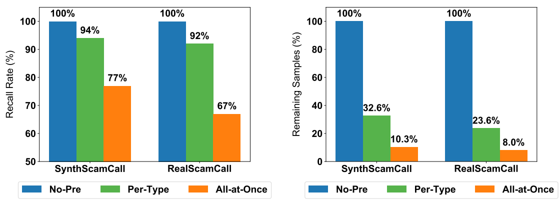
Setup	Method	SynthScamCall	RealScamCall
Few-shot +Prior Rules	SP	0.806	0.264
	PS	0.792	0.180
	CoT	0.783	0.232
	CoT+RE2	0.775	0.246
	Ours	**0.852**	**0.466**

Table 4: Performance of different prompting methods on experiment datasets with prior rules. The best result is marked in bold.

framework and CoT on this scenario with only one scam example of new type. As shown in the Fig. 4, CoT prompting learned the scam pattern as simply "fake product advertising," leading to an incorrect judgment. In contrast, our method, guided by rules, focused better on the core features of the scam type and made the correct judgment.

5.3 Ablation Study

Impact of Prior Rule To validate the effectiveness of our prior rules in assisting LLMs with scam call detection, we compare the performance of other prompting methods using the Qwen-long, incorporating the summarized prior rules as part of the input. The prompt template design is shown in Appendix B.2, and the experimental results are presented in Table 4. Comparing Table 3 and Table 4, the results show that prompting with prior rules outperforms methods without them, highlighting the importance of prior rules in guiding the model to focus on scam-related content. Additionally, compared to other methods with prior rules, our approach achieves better performance, demonstrating that our prompt design more effectively leverages prior rules to enhance scam call detection.



(a) Scam Call Recall (↑) (b) Remaining Samples (↓)

Figure 5: Performance of pre-selection on the SynthScamCall and RealScamCall datasets: scam call recall and remaining samples. **No-Pre** is a reference strategy without pre-selection. **Per-Type** refers to filtering each scam type separately, **All-at-Once** refers to filtering all scam types simultaneously.

Pre-S	Disc	Relf	SynthScamCall	RealScamCall
×	✓	✓	0.779	0.325
✓	×	✓	0.819	0.398
✓	✓	×	0.803	0.384
✓	✓	✓	**0.852**	**0.466**

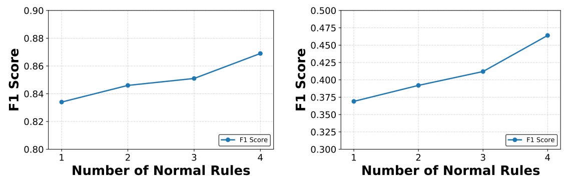
Table 5: Result of ablation experiment, where Pre-S indicates whether the pre-selection stage is included, Disc represents whether the discrimination reasoning step is included, and Relf denotes whether the reflection stage is included, with the results reported as F1-scores.

Impact of Pre-Selection As shown in Fig. 5, we compare two strategies in the pre-selection stage: **Per-Type**, which filters each scam type separately, and **All-at-Once**, which applies a unified filtering process to all types. While All-at-Once reduces the workload, it fails to capture distinct features of each scam type, resulting in lower recall. In contrast, the Per-Type strategy better preserves scam-type distinctions and achieves higher recall, making it our default choice. Using the Per-Type strategy, we observe that the pre-selection stage significantly reduces the number of texts requiring detection, with a consistently high recall above 90% as the dataset size increases. To further validate its effectiveness, we remove the pre-selection stage and apply Rule-Guided Detection to the full dataset. As shown in the first row of Table 5, performance drops notably, suggesting that early filtering of non-scam calls allows the model to focus on more relevant samples and make more accurate predictions.

Reasoning Step Ablation We conduct ablation experiments to evaluate the impact of each reasoning step in rule-guided detection. As shown in Table 5, removing any step leads to a performance decline, confirming the contribution of each stage. The results demonstrate that progressively incorporating prior rules enhances the model's ability to distinguish scam calls, effectively capturing key scam patterns and improving overall detection performance.

Method	SynthScamCall		RealScamCall	
	Period1	Period2	Period1	Period2
Finetuned BERT	0.759	0.178	0.927	0.006
BERT-AT	0.852	0.212	0.896	0.056
BERT-AT+ELS	0.891	0.221	0.890	0.057
AdSPT	0.902	0.225	0.913	0.078

Table 6: Performance degradation of traditional PLM-based and domain-adaptation methods under distribution shift



(a) SynthScamCall (b) RealScamCall

Figure 6: Impact of different number of normal rules

5.4 Further Analysis

Failure of Traditional Methods To assess the impact of data distribution shift, we compared the performance of traditional text classification methods and domain adaptation methods on test data collected during the same period as D_{Train} (Period 1) and on D_{Test} (Period 2). The Table 6 shows that all methods achieve strong performance on data collected during the same period as the training set, but their effectiveness degrades sharply on data collected after some time, underscoring the impact of data distribution shift and the difficulty traditional methods face in adapting to it.

Impact of Number of Rules To simulate real-world scenarios where the rule set evolves over time, we examine how gradually adding refined normal rules affects detection performance. As shown in the Fig. 6, as the number of rules increases and the descriptions of call characteristics become more comprehensive, the detection performance steadily improves. This show that in practice, as our understanding of call characteristics deepens, we can enhance the model's ability to understand these characteristics and improve detection by continuously expanding the prior rule set. The impact of scam rule quantity is detailed in Appendix C.4.

Short Name Source	SynthScamCall	RealScamCall
LLM Summary	0.842	0.453
Manually Rewriten	0.851	0.465
Names Used in Our Method	0.852	0.466

Table 7: Performance (F1-score) comparison of our method using short names obtained through different methods

Robustness to Short Name Selection To verify that our method does not heavily rely on specific human-designed short names, we conducted experiments using alternative short name sources, including manually rewritten names and LLM-generated names. As shown in Table 7, performance remains stable in all variants, with only marginal differences. This confirms that the short name serves merely as a coarse and broad abstraction to assist in filtering, without strongly affecting the final detection results.

6 Conclusion

This paper identifies the data distribution shift caused by the evolving nature of scam calls and introduces a framework using large language models (LLMs) to detect these evolving scams. By leveraging expert-defined scam and normal call rules in a hierarchical few-shot prompting approach, our method ensures stable performance. Experimental results show that our approach outperforms existing methods, providing an effective solution to the detection problem. The framework's adaptability to new scam scenarios through simple rule modifications enhances its real-world applicability. We also release the synthesized dataset to support future research.

Limitations

Our method detects scam calls using LLMs through prompting, but it faces limitations such as long inference times and high computational costs. Given the scale of daily call traffic, it is not efficient enough for large-scale deployment. In the preselection stage, we used unfine-tuned open-source models, resulting in suboptimal performance. Due to time and data constraints, fine-tuning has not yet been explored, but it is a potential improvement. Future work will focus on optimizing LLM usage, reducing processing time and costs, and exploring the use of LLMs to assist in training smaller models for detection.

Ethics Consideration

The primary goal of our work is to detect scam calls. Regarding call data, we ensure that its content does not pose financial or life-threatening risks to individuals or organizations. For our method, all prompts utilized in this research do not pose any threat to the safety or well-being of others, and we are committed to conducting our research in an ethical and responsible manner.

The real-world dataset used in our study was collected and shared by a large online recruiting platform, which clearly informs users during registration that calls made through its virtual number system may be recorded for quality and security purposes. Its Privacy Policy permits the use of de-identified data for academic research serving the public interest without requiring additional consent.

All shared data were de-identified by the platform through automated and manual filtering to remove personal or sensitive information. The scam and non-scam labels were assigned by trained staff using a multi-stage quality assurance process. In addition, we conducted multiple rounds of manual review to ensure that there is no privacy leakage. Data will not be publicly released. We believe that this work, conducted under strict ethical safeguards, can contribute to mitigating the societal threat posed by scam calls.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62276280), Guangzhou Science and Technology Planning Project (No. 2024A04J9967).

References

Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. 2021. Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3:563060.

Nikesh Bajaj, Tracy Goodluck Constance, Marvin Rajwadi, Julie Wall, Mansour Moniri, Cornelius Glackin, Nigel Cannings, Chris Woodruff, and James Laird. 2019. Fraud detection in telephone conversations for financial services using linguistic features. *arXiv preprint arXiv:1912.04748*.

Richard J Bolton and David J Hand. 2002. Statistical fraud detection: A review. *Statistical science*, 17(3):235–255.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 4019–4028.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Brendan Hong, Tee Connie, and Michael Kah Ong Goh. 2023. Scam calls detection using machine learning approaches. In *2023 11th International Conference on Information and Communication Technology (ICoICT)*, pages 442–447. IEEE.

Xinxin Hu, Haotian Chen, Hongchang Chen, Shuxin Liu, Xing Li, Shibo Zhang, Yahui Wang, and Xiangyang Xue. 2023. Cost-sensitive gnn-based imbalanced learning for mobile social network fraud detection. *IEEE Transactions on Computational Social Systems*.

Xinxin Hu, Haotian Chen, Hongchang Chen, Shibo Zhang, Shuxin Liu, and Xing Li. 2022. Telecom fraud detection via imbalanced graph learning. In *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*, pages 1312–1317. IEEE.

Liming Jiang. 2024. Detecting scams using large language models. *arXiv preprint arXiv:2402.03147*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Yufeng Kou, Chang-Tien Lu, Sirirat Sirwongwattana, and Yo-Ping Huang. 2004. Survey of fraud detection techniques. In *IEEE international conference on networking, sensing and control, 2004*, volume 2, pages 749–754. IEEE.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

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*, pages 3168–3177.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

Dare Azeez Oyeyemi and Adebola K Ojo. 2024. Sms spam detection and classification to combat abuse in telephone networks using natural language processing. *arXiv preprint arXiv:2406.06578*.

Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Is a question decomposition unit all we need? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4553–4569.

Letian Peng, Yuwei Zhang, and Jingbo Shang. 2024. Controllable data augmentation for few-shot text mining with chain-of-thought attribute manipulation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1–16.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Sanjeev Rao, Anil Kumar Verma, and Tarunpreet Bhatia. 2021. A review on social spam detection: Challenges, open issues, and future directions. *Expert Systems with Applications*, 186:115742.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

T-YLPG Ross and GKHP Dollár. 2017. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988.

Mohammad Rostami, Digbalay Bose, Shrikanth Narayanan, and Aram Galstyan. 2023. Domain adaptation for sentiment analysis using robust internal representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11484–11498.

Zitong Shen, Sineng Yan, Youqian Zhang, Xiapu Luo, Grace Ngai, and Eugene Yujun Fu. 2025. " it warned me just at the right moment": Exploring llm-based real-time detection of phone scams. *arXiv preprint arXiv:2502.03964*.

Milita Songailaitė, Eglė Kankevičiūtė, Bohdan Zhyhun, and Justina Mandravickaitė. 2023. Bert-based models for phishing detection. In *CEUR Workshop proceedings: IVUS 2023: Proceedings of the 28th international conference on Information Society and University Studies, Kaunas, Lithuania, May 12, 2023.*, volume 3575, pages 34–44. Aachen: CEUR-WS.

Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE international conference on data mining (ICDM)*, pages 598–607. IEEE.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Hui Wu and Xiaodong Shi. 2022. Adversarial soft prompt tuning for cross-domain sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2438–2447.

HK Xu, TT Jiang, X Li, and 1 others. 2022. Bilstm network fraud phone recognition based on attention mechanism. *Computer Systems and Applications*, 31(3):326–332.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15549–15575.

YiFan Zhang, Xue Wang, Jian Liang, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. 2023. Free lunch for domain adversarial training: Environment label smoothing. *arXiv preprint arXiv:2302.00194*.

Qianqian Zhao, Kai Chen, Tongxin Li, Yi Yang, and XiaoFeng Wang. 2018. Detecting telecommunication fraud by understanding the contents of a call. *Cybersecurity*, 1:1–12.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP 2019*, page 241.

Zhe Zhao, Yudong Li, Cheng Hou, Jing Zhao, and 1 others. 2023. Tencentpretrain: A scalable and flexible toolkit for pre-training models of different modalities. *ACL 2023*, page 217.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Dataset Details

A.1 Information of Datasets

We first obtain call texts from two different time periods to create a real-world call dataset, **RealScamCall** (**Real**-World **Scam Call** Dataset). We use the call data collected in May as $\mathcal{D}_{Train}$ and the data collected in July as $\mathcal{D}_{Test}$. Subsequent observations and experimental validation reveal significant content differences between these two datasets, indicating that $\mathcal{D}_{Train}$ and $\mathcal{D}_{Test}$ come from different data distributions. The dataset was collected and de-identified by a large online recruiting platform that records calls for quality and safety, with user consent and a privacy policy permitting academic use. All data were anonymized through automated and manual filtering to ensure the absence of personal or sensitive information.

Additionally, to facilitate scam call detection research in addressing the data distribution shift problem and to enable continuous exploration of robust methods, we constructed a synthetic dataset **SynthScamCall** (**Synth**etic **Scam Call** Dataset), a publicly available dataset generated using LLMs. For $\mathcal{D}_{Train}$, we use the LLM to generate normal call data that maintains a similar style to the existing unlabeled data but with different content. A small subset of these calls is then randomly selected, and Scam content are inserted to create synthetic scam call texts. The inserted Scam segments are generated by the model based on real scam call texts. For $\mathcal{D}_{Test}$, we adopt the method proposed in (Peng et al., 2024) for modifying the attributes of call texts. We use the model to alter background information in call texts that is unrelated to fraud (e.g., industry names). Meanwhile, for scam call texts, we ask model to keep the core Scam content unchanged. Through this way, we aim to make the content in $\mathcal{D}_{Test}$ different from $\mathcal{D}_{Train}$, thereby replicating the data distribution shift problem. Furthermore, to address data privacy concerns, we manually anonymized the dataset by replacing sensitive information such as names, addresses, and company names with meaningless placeholders.

A.2 Statistic of Datasets

In this section, we present the specific details of the real dataset RealScamCall and the publicly available dataset SynthScamCall, which is artificially constructed through the LLM. First, we show the statistics of the used datasets as shown in the Table 8.

Dataset	Source	D_{Train}		D_{Test}	
		# Fraud	# Call	# Fraud	# Call
SynthScamCall	synthetic	592	6926	223	2293
RealScamCall	real-world	1141	9888	845	65813

Table 8: Statistics of the used datasets. **#Fraud** denotes the number of scam call text and **#Call** denotes the number of call text.

A.3 Example of Scam Call Text

The names of scam types are quite abstract. In the main text, we have presented example texts for some scam types to help understand the corresponding Scam scripts. In this section, we provide a more detailed presentation. As shown in Table 8, we supplement the previous examples by showcasing example texts for all scam types in the used dataset.

B Experiment details

B.1 Training of PLM-based Methods

For finetuned BERT, we choose the Chinese Whole Word Masking RoBERTa-Large models pretrained by TencentPretrain (Zhao et al., 2023) (Apache-2.0 License) and UER-py (Apache-2.0 License) (Zhao et al., 2019) as the base model. We adopt a hard prompt tuning approach. The prompt template is set as: "*Is the following text a scam call? [text_A] Answer: [ANS]*", where the input text replaces *[text_A]*. The probability of "Yes" appearing in *[ANS]* represents the likelihood that the input text is a scam call, while the probability of "No" represents the likelihood that it is a normal call. To address the issue of class imbalance, we employ Focal Loss(Ross and Dollár, 2017), setting the parameters $\gamma = 2$ and $\alpha = 0.9$. We use the Adamw optimizer (Loshchilov, 2017) with a learning rate of 1×10^{-5} to update model parameters. The training epoch is 5 and batch size is 32.

For domain adaptation methods, we choose the Chinese Whole Word Masking RoBERTa-Large models pretrained by TencentPretrain as encoder. During training, we also use the AdamW optimizer with a learning rate of 5×10^{-6}. The training is conducted for 10 epochs with a batch size of 24.

All experiments are conducted using PyTorch on a single NVIDIA RTX 3090 (24GB) GPU.

B.2 Details of Prompting Methods

We call the Qwen-long, Deepseek-v3 and Deepseek-R1 APIs from Bailian Model Studio for baseline methods and rule-guided detection, and the ERNIE-Speed-128k API from ModelBuilder for pre-selection. For Qwen-long, the parameters are set as follows: temperature $t = 1.0$ and $top\_p = 0.8$. For Deepseek-v3, the parameters are: temperature $t = 0.7$, $top\_p = 0.6$, and $presence\_penalty = 0.95$). For ERNIE-Speed, the parameters are: temperature $t = 0.95$ and $top\_p = 0.7$.

For example selection, we select two examples for each scam type from $\mathcal{D}_{Train}$, consisting of one scam example and one normal example. Except for standard prompting, the reasoning process for examples in other prompting methods is manually written. Specifically, for SynthScamCall, we set the maximum number of scam samples to twice the number of scams in the dataset, while for the larger RealScamCall dataset, we set the limit to five times the number of scam samples. Traditional pretraining methods can control the number of detected scam calls by setting a probability threshold, while prompting methods, without probability outputs, must randomly select calls if detections exceed the maximum allowed.

The detailed prompt template design of other prompting methods used in our experiments is shown in Table 15. As discussed in Section 5.3, to validate the effectiveness of the proposed prior rules, we incorporated the same prior rules into traditional prompting methods for guidance. The detailed prompt design with prior rules is shown in Table 16. For our method, the Scam Rule Set used in the experiment is shown in Table 13, and the Normal Rule Set used is shown in Table 14. The detailed prompt template design of our method is shown in Table 17.

C Additional Experiment Result

C.1 Visualization of Data Distribution Shift

To intuitively demonstrate the presence of data distribution shift caused by changes in call content over different time periods, we extracted features from call transcripts collected at various times using Sentence-BERT(Reimers and Gurevych, 2019). We then applied PCA for the reduction of dimensionality and visualized the results in Fig. 7. The visualization clearly reveals the existence of data distribution shifts across different periods.

C.2 Impact of the Number of Demonstrations

We evaluate the impact of example quantity by comparing 1-shot, 2-shot, and 4-shot settings. As shown in Fig. 8, 2-shot significantly outperforms 1-shot by enabling comparison between normal and scam calls. While 4-shot includes more examples, its improvement is marginal and does not justify the added cost and longer context. This suggests that 2-shot strikes a good balance between effectiveness and efficiency, and is thus adopted in our experiments.

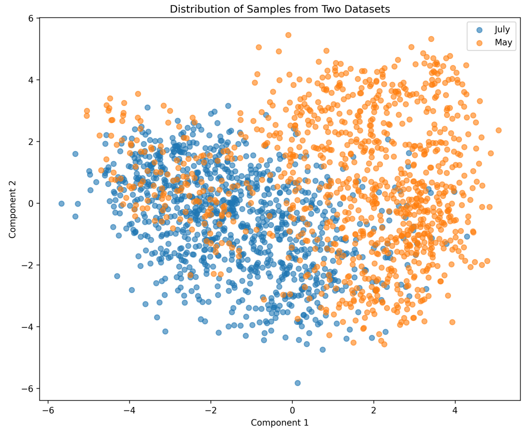
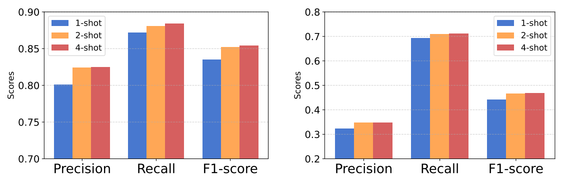


Figure 7: Visualization of Data Collected in Different Time Periods



(a) SynthScamCall (b) RealScamCall

Figure 8: Impact of different number of demonstrations

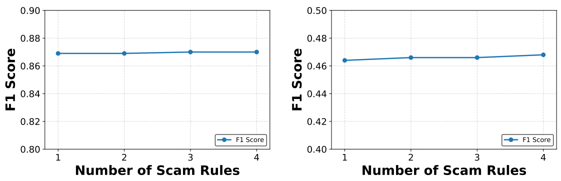
C.3 Impact of Example Selection

Selection Strategy	SynthScamCall			RealScamCall		
	Presicion	Recall	F1-score	Presicion	Recall	F1-score
Complexity-based	0.822	**0.883**	0.851	0.342	**0.732**	**0.466**
Random	**0.824**	0.881	**0.852**	**0.347**	0.710	0.466

Table 9: Comparison of experimental results for two different example selection strategies.

Some studies have suggested that different example selection strategies can improve the performance of prompting methods. In our experiments, we compare random example selection with complexity-based selection. Since our dataset does not include attribution steps, we follow the approach of previous work and use the sample length as a criterion for complexity. Experimental results show that different selection strategies have little impact on performance. A possible reason is that our method primarily relies on prior rules to guide the model's detection, reducing its dependence on specific examples.

C.4 Additional Analysis on Number of Rules



(a) SynthScamCall (b) RealScamCall

Figure 9: Impact of different number of scam rules

We have shown in Section 5.4 that gradually expanding the normal rule set improves detection performance. In this section, we further explore the impact of incrementally adding rules to the scam rule set on experimental performance. As shown in Fig. 9, it is evident that adding more rules does not significantly improve performance. This may be because, in our scenario, scams are already categorized by scam type, and a single feature can effectively capture the characteristics of each type. However, for more complex scam types, where a single rule cannot fully capture the key feature, adding additional rules would likely result in a more noticeable performance improvement.

C.5 Impact of Rule Incorporation

Method	SynthScamCall	RealScamCall
CoT prompt (Prompting everything together)	0.783	0.232
Ours (Step-by-step prior rule integration)	**0.852**	**0.466**

Table 10: Performance of different rule incorporation strategies

To investigate the impact of prompting everything together, we compared two approaches: inputting all prior rules simultaneously and guiding the model to learn their usage through CoT prompts, versus our step-by-step method, which mimics human experts' hierarchical reasoning by gradually introducing prior knowledge to guide the detection process. As shown in the Table 10, the F1 scores indicate that our step-by-step prior rule integration approach more effectively utilizes prior rules, resulting in better performance compared to prompting everything together.

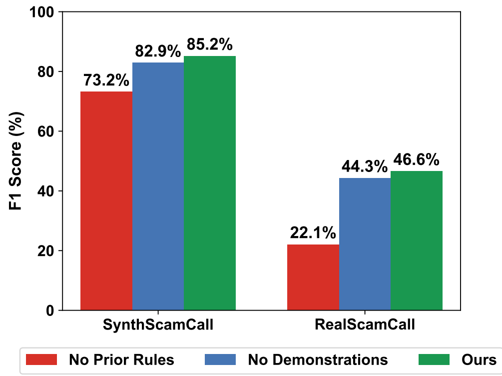


Figure 10: Performance comparison of our method and its ablated variants on SynthScamCall and RealScamCall. **No Prior Rules** removes prior rules, **No Demonstrations** removes in-context examples.

C.6 Effect of Prior Rules vs. Demonstrations

To assess the respective contributions of prior rules and in-context demonstrations, we conduct ablation studies on both components. As shown in Fig. 10, removing demonstrations leads to a moderate performance drop, suggesting that examples help refine the model's understanding but are not essential. In contrast, removing prior rules results in a drastic decline in performance, confirming that our method relies primarily on expert knowledge to guide the detection process. These results highlight the central role of rules in enabling accurate scam detection, with demonstrations playing only a complementary role.

C.7 Analysis of Effectiveness-Cost Trade-off

Method	SynthScamCall		RealScamCall	
	F1-score	Processing Rate	F1-score	Processing Rate
Ours (*w/o* lightweight Pre-selection)	0.861	100%	0.473	100%
Ours (*w/* lightweight Pre-selection)	0.852	32.6%	0.466	23.6%

Table 11: Impact of lightweight pre-selection on performance and LLM processing workload. **#Processing Rate** denotes percentage of texts to be processed by large LLM.

To evaluate how our framework balances detection effectiveness and computational cost, we compare two settings: one using a lightweight model for pre-selection (our framework setting), and another where the large LLM handles all inputs directly, including the pre-selection step. As shown in Tab. 11, incorporating the lightweight model slightly reduces detection performance but significantly lowers the proportion of texts processed by the large LLM, thereby reducing computational overhead. This demonstrates how our design effectively allocates easy cases to the lightweight model while reserving the large LLM for more challenging instances.

D Supplementary Case Study

As shown in Table 18, we conduct a case study to show the effectiveness of our proposed method over the chain-of-thought prompting. The results show that, while both methods identified potentially scam-related content, our method made more accurate judgments with the help of prior rules, whereas the chain-of-thought method gave incorrect answers. A case study demonstrates the importance of the reflection stage, where, as shown in Table 19, our method, guided by normal rules, captured normal features in the conversation and prevented misclassification.

E Computational Budget

As introduced in (Liu et al., 2024), the DeepSeek-v3 model has a total of 671 billion parameters, with 37 billion activated for each token. However, the specific number of parameters for Qwen-long and ERNIE-Speed is unclear. In our experiments, testing a complete detection using the Qwen-long model on SynthScamCall takes approximately 4-6 hours, while testing on RealScamCall takes around 20-24 hours (with parallel processing across different scam types).

Scam Types	Example
工作岗位多 (offering many job positions)	我看到你投了份简历请问你找到工作了吗。我们这边岗位比较多的方便加你个微信把具体的岗位发给你吗 Trans: I saw that you submitted a resume. Have you found a job? We have a lot of positions available. Would it be convenient to add you on WeChat and send you the details of the positions?
拉人培训转行 (inviting individuals to transition to a new industry and offering training opportunities)	我们这边是做软件测试的，你之前有了解过这个行业么？没做过也没关系，我们会提供一个培训服务，我们这儿将来也会是一个重要行业，赚的也很多 Trans: We are engaged in software testing. Have you ever known about this industry? It doesn't matter if you haven't done it before. We will provide training services. This will be an important industry in the future and earn a lot of money.
包吃包住 (free meals and accommodation)	你好找工作吗,我们公司正在招聘高薪接贷包吃包住，你看咱们可以加微信了解一下 Trans: Are you looking for a job? Our company is hiring with a high salary, including accommodation and meals. Would you be interested in adding me on WeChat to learn more?
就近安排 (assign work close to the current location)	在网上看到您的简历您目前在找工作对吗。您考虑做房地产销售嘛工作地点都可以根据你的要求就近安排的 Trans: I saw your resume online. Are you currently looking for a job? Would you consider a position in real estate sales? The work location can be arranged close to your preferences.
全国都有分公司 (have branch offices nationwide)	我在平台上看到你的简历请问你现在在找工作吗那那个我们是全国招聘的对我们全国都有分公司是一家很大的企业不知道你感兴趣么那您加个微信吧我细说跟您说一下 Trans: I saw your resume on the platform. Are you currently looking for a job? We are hiring nationwide and have branch offices across the country. It's a large company. I'm not sure if you're interested, but if you are, please add me on WeChat, and I can tell you more about it.

Table 12: Common scam types in the dataset and their corresponding example texts

Scam Type	Scam Rule
工作岗位多 (offering many job positions)	文中出现“1.岗位选择丰富2.我们这边职位很多3.这里有各种类型的工作机会4.我们这边有多种岗位可选”等类似表述，并且后续没有说明招聘的岗位信息（如外卖员、骑手、客户经理等职业名称信息） If the text mentions statements like ’1. Rich variety of job positions. 2. We have many positions available. 3. There are various types of job opportunities here. 4. We offer multiple positions to choose from,’ without further clarification on specific job titles (such as delivery drivers, riders, customer managers, etc.), it indicates a suspicious call.
拉人培训转行 (inviting individuals to transition to a new industry and offering training opportunities)	文中出现“1.你有没有考虑过转行，比如参加个短期培训就可以上手了。2.现在有些行业挺热门的，通过培训很容易入行，要不要试试看？”等类似表述，重点不在于培训而是通过条件诱惑被招聘人转行。 If the text mentions statements like ’1. Have you considered changing careers, you can easily get started with a short-term training. 2. Some industries are quite popular now, and you can easily get into them with training, want to give it a try?’ The focus here is not on the training itself, but on tempting the recruit to change careers through these conditions, indicating a possible scam.
包吃包住 (free meals and accommodation)	文中出现“1.我们提供包吃包住，待遇不错，快来试试吧2.招聘岗位，包吃包住，欢迎加入我们团队”等类似表述，包吃包住是提供吃和住两方面条件，而不是只提供住宿或者提供车辆等其他常见福利，并且该条件与工作内容不匹配 If the text mentions statements like ’1. We offer free meals and accommodation with good benefits, come try it out! 2. Job positions with free meals and accommodation, welcome to join our team,’ the offer of free meals and accommodation should cover both food and lodging, not just accommodation or other common benefits like transportation. Moreover, if these benefits do not align with the job requirements
就近安排 （assign work close to the current location）	文中出现“1.针对您的住址帮您安排最近的工作单位。2.根据您的位置为您匹配合适的办公地点。”等类似表述，并且一般首先不确定被招聘人的位置，没有具体的工作信息，直接提出可以就近安排 If the text mentions statements like ’1. We will arrange the nearest work location based on your address. 2. We’ll match you to an office nearby according to your location,’ but lacks specific job details, directly offering a local arrangement without confirming the applicant’s location, it suggests a suspicious call.
全国都有分公司 (have branch offices nationwide)	文中出现“1.我们是XX公司的，全国设有多个分公司。2.我们公司在全国各地都有分支机构3.在全国有xx家分公司”等类似表述，注意是全国到处都有而不是具体在某地有分公司 If the text mentions statements like ’1. We are XX company, with multiple branches nationwide. 2. Our company has branches across the country. 3. We have xx branches nationwide,’ and it refers to ’nationwide’ without specifying locations, it suggests a suspicious call.

Table 13: Scam Rule Set

Normal Rule
（1）文本中提到详细的地址信息，文中出现“1. 我们在xx城xx区。2. 位于xx大厦xx楼。3. 地点是xx广场附近。”等类似说法说明是正常通话 (1) If the text mentions detailed address information, such as "1. We are located in XX city, XX district. 2. Located in XX building, XX floor. 3. The location is near XX square," it suggests a normal call. （2）文本中提到邀请面试或邀请线下参观，文中出现“1. 欢迎线下参观了解。2. 诚邀来公司面试。3. 请到现场详细沟通。”等类似说法说明是正常通话 (2) If the text includes an invitation for an interview or an in-person visit, such as "1. You are welcome to visit in person. 2. We sincerely invite you for an interview. 3. Please come to the site for detailed communication," it indicates a normal call. （3）提出的工作条件与工作岗位相匹配（提供住宿、带薪培训实习等常见的招聘条件）,比如“1. 提供包住和补贴。2. 有带薪培训机会。3. 工作条件明确合理。”等类似说法不认为具有诈骗嫌疑 (3) If the job conditions match typical recruitment offerings (e.g., accommodation, paid training internships), such as "1. Accommodation and subsidies provided. 2. Paid training opportunities available. 3. Job conditions are clear and reasonable," it is not considered to have scam characteristics. （4）招聘骑手或送货员提出可以就近安排或包吃包住或全国都有分公司等条件是正常通话，为保安提供包吃包住等条件是正常通话，比如“1. 骑手工作可就近安排。2. 送货员包吃住条件合理。3.我们这边保安是包吃住的”等类似说法不认为具有诈骗嫌疑 (4) If the job offer involves positions like delivery riders or security guards with reasonable conditions, such as "1. Rider positions can be arranged locally. 2. Delivery workers have reasonable accommodation and meal provisions. 3. Security guards are provided with meals and accommodation," it suggests a normal call, and these conditions are not considered scam-related.

Table 14: Normal Rule Set

Method	Prompt Template(*w/o* Prior Rule)
SP	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain Scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example: {Example} 请参考给定的样例文本，判断下面给出的文本是否是一则诈骗通话文本，请直接回答"是"或"否"，答案长度限制在1个字，不需要做额外的解释。 (Trans: Please determine whether the given text is a scam call based on the provided example texts. Answer only with "Yes" or "No", with a length limit of one character, and without any explanation.) Input: {Input}
CoT	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain Scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example: {Example} 请参考给定的样例文本，让我们一步一步进行思考，判断下面给出的文本是否是一则诈骗通话文本，请仿照样例文本一步一步进行思考并得到最终答案。 (Trans: Please determine whether the given text is a scam call based on the provided example texts.Let's think step by step, following the reasoning process of the example texts, and arrive at final conclusion.) Input: {Input}
PS	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain Scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example:{Example} 请参考给定的样例文本，判断下面给出的文本是否是一则诈骗通话文本，让我们首先理解这个问题并给出解决问题的方案，然后执行解决方案，一步一步解决问题，请仿照样例文本的答案格式进行回答。 (Trans: Please determine whether the given text is a scam call based on the provided example texts. Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan, solve the problem step by step. Please follow the answer format of the example texts.) Input: {Input}
CoT+Re	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K},请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example: {Example} 请参考给定的样例文本，让我们一步一步进行思考，判断下面给出的文本是否是一则诈骗通话文本，请仿照样例文本一步一步进行思考并得到最终答案。 (Trans: Please determine whether the given text is a scam call based on the provided example texts. Let's think step by step, following the reasoning process of the example texts, and arrive at final conclusion.) Input: {Input} 再读一遍问题：请参考给定的样例文本，让我们一步一步进行思考，判断下面给出的文本是否是一则诈骗通话文本，请仿照样例文本一步一步进行思考并得到最终答案。 (Trans: Read the problem again: Please determine whether the given text is a scam call based on the provided example texts. Let's think step by step, following the reasoning process of the example texts, and arrive at final conclusion.) Input: {Input}

Table 15: Prompt template of baseline prompting methods. **SP** stands for Standard Prompting and **PS** stands for Plan-and-Solve Prompting.

Method	Prompt Template(*w/* Prior Rule)
SP	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，该类通话往往具有诈骗特征{scam rules}，正常通话往往具有特征{Normal rules}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Such calls usually have {Scam rules}, while normal calls have {Normal rules}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example: {Example} 请参考给定的样例文本和先验规则，判断下面给出的文本是否是一则诈骗通话文本，请直接回答"是"或"否"，答案长度限制在1个字，不需要做额外的解释。 (Trans: Please determine whether the given text is a scam call based on the provided example texts and prior rules. Answer only with "Yes" or "No", with a length limit of one character, and without any explanation.) Input: {Input}
CoT	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，该类通话往往具有诈骗特征{Scam rules}，正常通话往往具有特征{Normal rules}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Such calls usually have {Scam rules}, while normal calls have {Normal rules}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example: {Example} 请参考给定的样例文本和先验规则，让我们一步一步进行思考，判断下面给出的文本是否是一则诈骗通话文本，请仿照样例文本一步一步进行思考并得到最终答案。 (Trans: Please determine whether the given text is a scam call based on the provided example texts and prior rules. Let's think step by step, following the reasoning process of the example texts, and arrive at final conclusion.) Input: {Input}
PS	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，该类通话往往具有诈骗特征{Scam rules}，正常通话往往具有特征{Normal rules}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Such calls usually have {Scam rules}, while normal calls have {Normal rules}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example:{Example} 请参考给定的样例文本和先验规则，判断下面给出的文本是否是一则诈骗通话文本，让我们首先理解这个问题并给出解决问题的方案，然后执行解决方案，一步一步解决问题，请仿照样例文本的答案格式进行回答。 (Trans: Please determine whether the given text is a scam call based on the provided example texts and prior rules. Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan, solve the problem step by step. Please follow the answer format of the example texts.) Input: {Input}
CoT+Re	Q:你是一个诈骗电话检测专家，下列文本为收集的通话文本，其中存在一些希望提到不真实信息来欺诈的文本，类别包括{short name of scam types 1}...{short name of scam type K}，该类通话往往具有诈骗特征{Scam rules}，正常通话往往具有特征{Normal rules}，请你帮忙找出隐藏在大量正常通话中的诈骗通话。 (Trans: You are an expert in scam call detection. The following texts are collected call transcripts, some of which contain scam content aimed at deceivingCommon types of scam calls include {short name of scam type 1} ... {short name of scam type K}. Such calls usually have {Scam rules}, while normal calls have {Normal rules}. Please help identify the few scam calls hidden among the large volumn of normal call texts.) Example: {Example} 请参考给定的样例文本和先验规则，让我们一步一步进行思考，判断下面给出的文本是否是一则诈骗通话文本，请仿照样例文本一步一步进行思考并得到最终答案。 (Trans: Please determine whether the given text is a scam call based on the provided example texts and prior rules. Let's think step by step, following the reasoning process of the example texts, and arrive at final conclusion.) Input: {Input} 再读一遍问题：请参考给定的样例文本和先验规则，让我们一步一步进行思考，判断下面给出的文本是否是一则诈骗通话文本，请仿照样例文本一步一步进行思考并得到最终答案。 (Trans: Read the problem again: Please determine whether the given text is a scam call based on the provided example texts and prior rules. Let's think step by step, following the reasoning process of the example texts, and arrive at final conclusion.) Input: {Input}

Table 16: Prompt template of baseline prompting methods with the guide of prior rules. **SP** stands for Standard Prompting and **PS** stands for Plan-and-Solve Prompting.

Stage	Prompt Template
Pre-Selection	你是一个电信诈骗通话检测工作人员，请检查给定文本中是否包含{short name of scam type}相关内容，只需要回答“是”或“否”即可，答案限制在1个字，不需要进行额外解释。 Trans: You are a telecom fraud call detection worker. Please check whether the given text contains content related to {short name of scam type}. Just answer ’Yes’ or ’No’. No further explanation is required. Input:{Input}
Rule-Guided Detection	为了解决诈骗通话检测问题，通话文本中存在一些希望通过提到{short name of scam type}来进行电话诈骗的文本，一步步回答接下来的问题并检测该文本是否是诈骗通话内容。 Trans: The call text contains content related to {short name of scam type}. Your task is to solve the following questions step by step and determine whether the given call text is a fraudulent one. 问题1：诈骗通话具有如下特征： 诈骗特征: {Scam Rule}请给据给定的诈骗通话特征一步一步判断文本是否有诈骗通话嫌疑? Trans: Q1: Texts that use {short name of scam type } for fraud often exhibit the characteristic: {Scam Rule}. Based on this rule, analyze the text step by step to determine if it shows signs of fraud. 问题2：正常通话具有如下特征： 正常特征: {Normal Rule} 如果问题1判断文本有诈骗通话嫌疑，请你根据正常文本逐条比对文本是否包含正常通话特征? Trans: Q2: Normal calls often exhibit the following characteristic: {Normal Rule}. If Q1 determines that the text shows signs of fraud, compare the text with the normal features one by one to check if it contains any normal characteristics. 问题3：请你综合上面问题的答案给出最终答案，该通话是否是一个诈骗通话，只需要给出是或者否的答案即可，不需要进行额外解释 Trans: Q3: Please provide a final answer based on the answers to the previous questions. Is the call a fraud call? Only answer Yes or No. No further explanation is required. 接下来为了帮你更好的理解问题解决的过程和记住输出答案的格式，给出样例文本： Trans: To help you better understand the process and remember the output format, here are samples: Example: {Example} 请综合上面文本及给出的判断过程，解答提出的问题并分析测试文本是否为一则诈骗通话文本，如果提到请回答是，反之请回答否。 Trans: Please combine the above text and the given judgment process to answer the questions and analyze whether the test text is a fraud call. If it is a fraud call, answer Yes; otherwise, answer No. Input:{Input}

Table 17: Prompt template of our methods

Input	请问你是在找工作是吧哦你目前的话是在哪个区呀哦那你这边需要提供住宿吗哦就是只考虑在这个区域找这种是吧我们这边也是服务行业的要不我们会微信具体聊一下呗如果说我们聊的合适的话你直接去赚点面试就可以了哦你发个定位过来吧我看一下附近离你最近的站点是在哪里 (Trans: Are you looking for a job? Which area are you currently in? Do you need accommodation? So, you're only considering finding a job in this area, right? Our company is also in the service industry. How about we discuss the details on WeChat? If everything sounds good, you can directly go for the interview. Send me your location, and I will check the nearest station to you.) 标签：正常文本(label: Normal)
CoT Answer	该文本提到“我们这边也是服务行业的”，但没有具体描述工作内容，直接提出提供住宿和附近站点，属于通过不真实信息进行欺诈的情况，所以判定为诈骗通话文本，所以答案：是 (Trans: The text mentions, "Our company is also in the service industry," but does not provide specific details about the job. It directly offers accommodation and mentions the nearest station, which aligns with the use of false information to deceive the victim. Therefore, it is judged as a scam call text. So the answer is Yes.)
Ours Method Answer	1. 问题1答案：该文本在询问被招聘人的位置后，提出可以就近安排工作站点，并且提到“离你最近的站点”，这符合<就近安排>特征。但不同于典型诈骗的是，它先询问了对方的位置信息("你目前的话是在哪个区呀")，并且提及了具体行业(服务行业)，所以不完全符合诈骗特征中"没有具体工作信息"和"首先不确定被招聘人位置"的特点。因此初步判断该通话文本诈骗嫌疑不大。 2. 问题2答案：(1)没有明确提到非常详细的地址。 (2)文本中提到“如果说我们聊的合适的话你直接去赚点面试就可以了”，这是邀请面试的表现。 (3)提出的工作条件与岗位相匹配。 (4)涉及骑手或送货员这类职业时，“离你最近的站点”这种就近安排的说法是合理的。 经过逐条判断，该通话包含正常通话特征(2)(3)(4)，因此根据问题2的判断标准，该通话为正常通话。 3. 问题3答案：否 (Trans: Q1 Answer: This text asks the recruit's location, then offers to arrange a nearby work station, mentioning "the nearest station to you," which aligns with the local job arrangement feature. However, unlike typical scams, it first inquires about the recruit's location ("Which area are you currently in?") and specifies the industry (service industry), so it doesn't fully match the scam characteristics of "no specific job details" and "not initially asking for the recruit's location." Therefore, based on this, the scam suspicion for this call is minimal. Q2 Answer: (1) No specific address is mentioned. (2) The text mentions, "If everything sounds good, you can directly go for the interview," which is an invitation for an interview. (3) The proposed job conditions align with the job role. (4) When it involves jobs like delivery drivers or couriers, the phrase "the nearest station to you" is a reasonable local job arrangement. After evaluating the points, this call contains normal call features (2)(3)(4). Therefore, according to the criteria for Question 2, this call is considered a normal call. Q3 Answer: No.）

Table 18: A case study of chain-of-thought prompting and our methods using Qwen-long API on an example of SynthScamCall

Input	你好我这里是XY公司人事招聘的，我看你这边昨天有浏览我的招聘信息是还在找工作吗？我们在这里各个区域都有分部，就是现在的话是在招聘相关工作人员你这边有没有兴趣呢？ZD新区是吗？ZD新区有的呀我们这边有很多分部，你在你是在那边租的房子是吗？哦你那个在*哪个位置呀？M城哦M城我们那里也有分部的我加下你的微信吧。你这边今天有时间吗可以到我们分部来了解一下，我加下你的微信加下你的微信今天没空那明天呢？ (Trans: Hello, this is HR recruitment from XY company. I noticed you browsed our job listing yesterday. Are you still looking for a job? We have branches in all regions. Right now, we are recruiting for related positions. Are you interested? Is it ZD New District? Yes, we have a branch there. We have many branches. Are you renting a place there? Oh, what's your location? M City, right? We also have a branch there. Let me add you on WeChat. Do you have time today? You can come to our branch to learn more. I'll add you on WeChat. How about today? If you're not available today, how about tomorrow?)
Ours Answer (*w/o* Reflection)	1. 问题1答案：测试文本中提到"我们在这里各个区域都有分部"，这与给定的诈骗特征相似，暗示无论对方在哪里都能安排工作。此外，对话还试图通过加微信等方式进一步联系，这种行为也较为可疑。因此该通话文本具有诈骗嫌疑。 2. 问题2答案：是 (Trans: Q1 Answer: The test text mentions "we have branches in all regions," which is similar to the given scam feature , implying that they can arrange work regardless of the location. Furthermore, the conversation attempts to establish further contact through WeChat, which is also suspicious. Therefore, this call text raises concerns of fraud. Q2 Answer: Yes.)
Ours Answer （*w/* Reflection）	1. 问题1答案：该文本中提到"我们在这里各个区域都有分部"，这与诈骗特征中的表述相似，暗示在全国各地设有分支机构。因此，根据问题1的判断标准，这段通话文本具有诈骗嫌疑。 2. 问题2答案：接下来逐条比对是否包含正常通话特征： (1) 文本中有提到较为具体的地点信息："ZD新区是吗"、"M城我们那里也有分部"，虽然不是特别详细的地址，但已经涉及到具体城市和地区。 (2) 文本中有提到邀请线下参观的内容："可以到我们分部来了解一下"。 (3) 虽然没有明确提到工作条件，但从整体语境来看，对方是在进行招聘沟通，并且提到了具体的地区有分部，这符合正常的招聘流程。 (4) 对于骑手、送货员或保安等特定职业的提及，该文本并没有涉及这些职业，但考虑到上述两点，仍然更倾向于认为这是正常通话。 3. 问题3答案：否 (Trans: Q1 Answer: The text mentions "we have branches in all regions," which is similar to the description in the scam feature, implying the presence of branches nationwide. Therefore, based on the criteria for Question 1, this call text raises suspicions of fraud. Q2 Answer: We compare the text against the characteristics of normal calls: (1) The text mentions specific location information: "Is it ZD New District?" and "We also have a branch in M City." Although not a detailed address, it refers to specific cities and regions. (2) The text mentions an invitation to visit in person: "You can visit our branch to learn more." (3) Although no specific job conditions are mentioned, the overall context suggests a recruitment conversation, with references to specific regions having branches, which aligns with normal recruitment procedures. (4) The text does not mention specific jobs such as riders, delivery workers, or security guards, but given the above points, it still leans more towards being a normal call. Q3 Answer: No.)

Table 19: A case study of our method using the Qwen-long API on an example from SynthScamCall, comparing the results with and without the reflection stage.