Towards Explainable Transaction Risk Analysis With Dual Graph Retrieval Augmented Generation

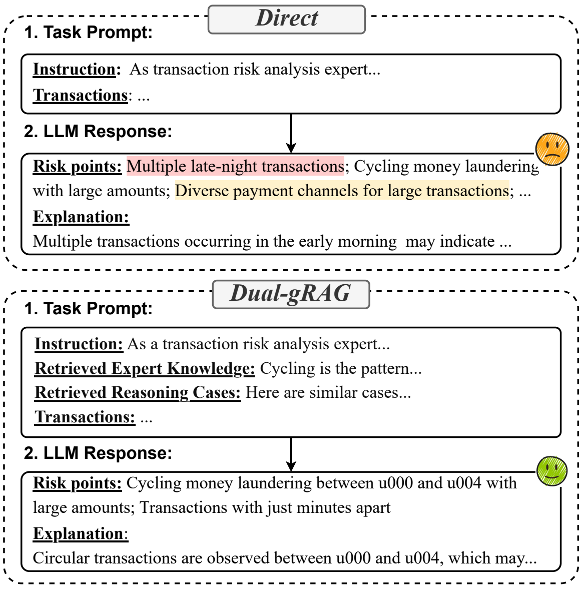
Anonymous ACL submission

Abstract

Explainable transaction risk analysis is a challenge for traditional deep learning models, which only predict suspicious transactions without explanations. Current explainable methods rely on hand-crafted rules and lack the ability to automatically generate language-based explanations. Large Language Models (LLMs) offer promise due to their reasoning and text generation abilities but struggle with domain knowledge and hallucinations, making risk analysis difficult. Specifically, LLMs face: **(1) insufficient adaptation to transaction data analysis**, and **(2) ineffective knowledge retrieval methods** that ignore the rich graph structure of transaction data. To address these issues, we propose the **Dual G**raph **R**etrieval-**A**ugmented **G**eneration (**Dual-gRAG**) framework, which utilizes dual retrieval: expert knowledge and reasoning case retrieval. Expert knowledge compensates for domain gaps, while reasoning case retrieval provides step-wise analysis guidance. We incorporate both graph-structured features and semantic features into the retrieval process to enhance the effectiveness of the retrieval. Extensive experiments show that Dual-gRAG improves LLMs' risk analysis capabilities, achieving a 50% increase in different metrics.

1 Introduction

In the financial domain, transaction risk analysis, particularly anti-money laundering (AML), is a critical billion-dollar challenge (Altman et al., 2023). Traditional rule-based approaches, while offering clear and understandable decision-making processes, face significant drawbacks. They require extensive human labor and are increasingly inadequate due to the evolving techniques of money laundering and the growing volume of data (Labib et al., 2020a). Moreover, these rule-based systems can be easily bypassed by new or unknown laundering patterns (Chen and Tsourakakis, 2022). As a result, deep learning technologies have gained recognition in AML detection. However, these models are often viewed as "black boxes," as they lack transparency and fail to provide explanations for their outcomes. This inability to generate natural language explanations significantly limits their usability and practical value, as users must rely on additional human effort to investigate and confirm the reasons behind the suspicion of a transaction (Kute et al., 2021).



Instruction: As transaction risk analysis expert...
Transactions: ...

Risk points: Multiple late-night transactions; Cycling money laundering with large amounts; Diverse payment channels for large transactions; ...
Explanation:
Multiple transactions occurring in the early morning may indicate ...

Instruction: As a transaction risk analysis expert...
Retrieved Expert Knowledge: Cycling is the pattern...
Retrieved Reasoning Cases: Here are similar cases...
Transactions: ...

Risk points: Cycling money laundering between u000 and u004 with large amounts; Transactions with just minutes apart
Explanation:
Circular transactions are observed between u000 and u004, which may...

Figure 1: Dual-gRAG enables LLMs to perform explainable transaction risk analysis more effectively. In contrast, when LLMs analyze transaction risk directly, the results may contain incorrect conclusions and irrelevant findings.

Currently, researchers are striving to provide interpretability for deep learning methods. For instance, knowledge distillation techniques are used to extract knowledge from black-box models into transparent models, i.e., surrogate models, as a form of post-hoc model explanation (Che et al., 2015; Tan et al., 2017). However, these methods generally fail to handle sequence data, i.e., transaction data. Other approaches (Zhang et al., 2022) employ Monte Carlo Tree Search (MCTS) (Browne et al., 2012) to generate a set of statistics from the outputs of black-box models, subsequently composed into logical rules with neural logic network. Although such approach can distill sequence models into rule-based systems, it can only produce explanations based on pre-defined rules rather than intuitive language explanations.

Motivated by the growing capabilities of large language models (LLMs) in text generation, reasoning, and real-world applications (Shang and Huang, 2024), LLMs are increasingly seen as a promising tool for explainable AML solutions. One key advantage of LLMs in risk analysis is their ability to reason about new, previously unseen money laundering patterns, which can compensate for the limitations of traditional rule-based models. Their reasoning capabilities can capture emerging risks that rule-based systems may miss. Additionally, LLMs excel in text generation, enabling them to provide natural language explanations for flagged transactions, thus addressing the practical limitations of deep learning models in offering transparency. Recent studies (Zhao et al., 2023) have demonstrated the strong performance of LLMs in natural language tasks, showcasing basic analytical and reasoning abilities. Preliminary research has also explored LLMs in financial tasks like sentiment analysis and stock prediction (Koa et al., 2024; Yang et al., 2023). However, their use specifically in AML remains limited.

Despite the remarkable performance of LLMs in language processing, they encounter significant challenges in tackling transaction risk analysis (example illustrated in Figure 1): **(1) Deficiency in transaction data analysis and reasoning.** LLMs are primarily designed for general natural language tasks, not structured transaction data (Zhao et al., 2023). This makes them ill-equipped to analyze and reason about laundering risk behaviors in transaction data. **(2) Unsuitable knowledge retrieval methods.** A challenge in using LLMs for risk analysis is their lack of domain-specific knowledge. While the RAG (Lewis et al., 2020) algorithm was proposed to address this, many RAG methods rely on text-based similarity for retrieval (Gao et al., 2023). These methods are inadequate for transaction data, which not only contains semantic features but also has a strong graph structure (Li et al., 2023), making them unsuitable for risk analysis.

To address the aforementioned issues, we propose a novel framework, Dual Graph Retrieval Augmented Generation (**Dual-gRAG**). This framework features a dual retrieval approach from two augmented knowledge bases: an expert knowledge base and a reasoning case library. The expert knowledge base compensates for the lack of domain-specific knowledge, while the case library provides a step-by-step reasoning process for effective guidance. The retrieved reasoning cases serve as reference examples for LLMs, providing stepwise guidance to steer LLMs learn how to analyze transaction data and achieve accurate results. Relevant expert knowledge and similar representative cases are combined with input transaction records to enhance the reasoning capability of LLMs to generate reliable risk analysis. Specifically, we devise graph retriever that incorporate the semantics of transaction data as well as the structural patterns of transactions. By integrating the semantic and graph structure features of transaction data, we enhance the effectiveness of retrieval augmentation, thereby further improving LLMs' analytical capabilities. We conduct extensive experiments to demonstrate Dual-gRAG's effectiveness, showing an average improvement of 50% for all LLMs.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first work leveraging LLMs for textual explainable transaction risk analysis. Unlike existing methods that only predict risk probability, our approach can generate textual analysis, providing a more applicable solution.

- We introduce a novel Dual-gRAG framework featuring dual graph retrievers with semantic and structure features of transaction data, addressing LLMs' inherent limitations.

- Extensive experiments show that our Dual-gRAG framework improves risk analysis performance by over 50% in precision and coverage across various LLMs.

2 Related Work

2.1 Retrieval Augmented Generation

RAG refers to a methodology that enhances model capability by integrating a retriever mechanism that accesses an external knowledge base (Lewis et al., 2020). When a query is submitted, the retriever identifies relevant documents from the knowledge base, which are then combined with the

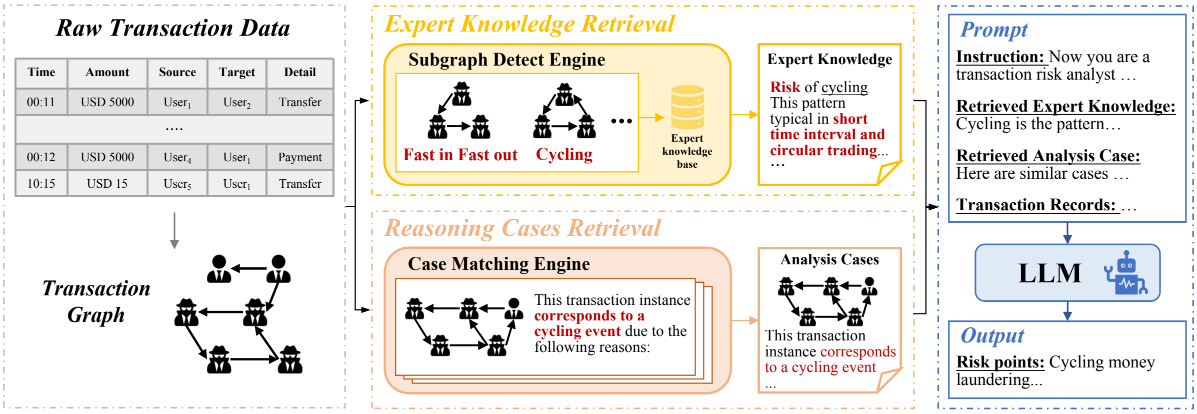

Raw Transaction Data
Time | Amount | Source | Target | Detail
00:11 | USD 5000 | User$_1$ | User$_2$ | Transfer
....
00:12 | USD 5000 | User$_4$ | User$_1$ | Payment
10:15 | USD 15 | User$_5$ | User$_1$ | Transfer
Transaction Graph
Expert Knowledge Retrieval
Subgraph Detect Engine
Fast in Fast out
Cycling
Expert knowledge base
Expert Knowledge
Risk of cycling
This pattern typical in short time interval and circular trading...
…
Reasoning Cases Retrieval
Case Matching Engine
This transaction instance corresponds to a cycling event due to the following reasons:
Analysis Cases
This transaction instance corresponds to a cycling event
...
Prompt
Instruction: Now you are a transaction risk analyst …
Retrieved Expert Knowledge: Cycling is the pattern…
Retrieved Analysis Case: Here are similar cases …
Transaction Records: …
LLM
Output
Risk points: Cycling money laundering...

Figure 2: Overview of Dual-gRAG. It utilizes dual retrieval mechanisms, incorporating both relevant expert knowledge and similar reasoning cases to enable explainable prediction.

query to form the input for the model (Lewis et al., 2020; Borgeaud et al., 2021; Wang et al., 2023). Subsequent advancements in this paradigm have introduced iterative retrieval (Shao et al.; Jiang et al., 2023), self-reflective retrieval (Asai et al., 2023), and the integration of RAG with fine-tuning methodologies. Initially confined to NLP tasks, RAG has expanded to multi-modal settings, incorporating external knowledge such as code (Hayati et al., 2018; Liu et al., 2023) and images (Yasunaga et al., 2023; Xie et al., 2023; Chen et al., 2022). Some have even explored graph-based RAG, using knowledge graphs (KGs) as external base for retrival(Edge et al., 2024), or aligning query embeddings with code graph embeddings for retrieval (Du et al., 2024).

However, these methods are tailored for textual queries, which are transformed into text embeddings for retrieval (Gao et al., 2023). This makes them unsuitable for risk analysis, where the input containing both textual and graph features, which our proposed method addresses by integrating a graph retriever.

2.2 Transaction Risk Analysis

Rule-based detection methods remain the mainstream choice in practice due to their clarity and ease of understanding (Oeben et al., 2019). However, they are easily bypassed by criminals, prompting research into supervised machine learning methods like support vector machines (SVM) (Raiter, 2021) and decision trees (DT) (Jullum et al., 2020). While these methods can detect some risks, they struggle to capture new money laundering patterns (Labib et al., 2020b).

Deep learning methods have been explored to address this limitation. Paule et al. (Ebberth et al., 2016) use Auto-Encoders for anomaly detection, while Han et al. (Han et al., 2018) apply deep learning to enhance AML monitoring. Researchers also leverage the graph structure of transaction data for risk analysis, with Alarab et al. (Alarab et al., 2020) and Weber et al. (Pareja et al., 2019) using Graph Convolutional Networks (GCN) to detect illicit transactions.

In parallel, efforts have been made to develop interpretable risk analysis methods. Che et al. (Che et al., 2015) employ knowledge distillation to explain black-box models, but struggle with sequential data. Zhang et al. (Zhang et al., 2022) extracting logical rules from model outputs using MCTS. However, these methods still rely on manually defined rules, lacking intuitive textual explanations.

In summary, while rule-based and deep learning models offer clarity or deeper insights, they fall short in providing both reliability and interpretability. Current explainable models only produce logical rules, not user-friendly explanations, limiting their practical application.

3 Methodology

In this section, we first formally define the task of explainable risk analysis. Then, we present our framework (shown in Figure 2). The Dual-gRAG framework consists of two main components: **Expert Knowledge Retrieval**, which retrieves relevant expert knowledge, and **Reasoning Cases Retrieval**, which retrieves step-by-step analysis cases similar to the input transaction data. By utilizing two retrievers to acquire domain knowledge and step-wise analytical processes for risk analysis, we enhance the large model's risk analysis capabilities through chain-of-thought prompting.

3.1 Task Description

In the field of risk analysis, it is essential to analyze extensive transaction data to identify potential risk points. Traditional methods are limited to binary classification, indicating whether a transaction record is suspected of risk, i.e., True or False, without pinpointing the specific risk points.

The goal of this paper is to enable LLMs to analyze transaction data and uncover the hidden risk points. Formally, let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ represent the set of transaction records for a user within a specific timespan, where each transaction record T_i contains attributes such as time and amount (see example in Figure 6). The input to the model is $\mathcal{T}$, and the output is a transaction risk analysis report $\mathcal{R} = \{'risk\ points' : P_1, P_2, \ldots, P_m;\ \ 'explanation' : E\}$, where each phrase P_i corresponds to an estimated risk point, E corresponds to an intuitive textual explanation of the result. The detailed example refers to Figure 1.

3.2 Retrieval-augmented Framework

Our framework features a dual-retrieval approach, integrating both domain knowledge and reasoning cases. This significantly differs from existing RAG frameworks (Lewis et al., 2020) that primarily address knowledge deficiencies. By incorporating step-by-step analysis details, the retrieved reasoning cases enhance LLMs' ability to evaluate transaction risks effectively. Moreover, both retrievers are graph-based, considering the semantic and structural features of transaction records, further distinguishing our method from current algorithms that rely solely on the semantic features of queries.

Since transaction data is inherently structured, we start by converting the input transaction data $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ into a transaction graph $G(\mathcal{V}, \mathcal{E})$, as shown in the left part of Figure 2. Each transaction record $T_i \in \mathcal{T}$ represents an interaction between a source user u_i and a target user u_j. We define $\mathcal{V}$ as the set of all users involved, and $\mathcal{E} = \{(u_i, u_j)\}$ denotes the set of interactions between these users.

3.2.1 Expert Knowledge Retrieval

To build a high-quality knowledge base, we collect many analysis reports from experts and synthesize these into an expert knowledge base. Unlike other NLP tasks (such as QA), the knowledge in transaction risk analysis often exists in the form of "*Rules*". Therefore, the expert knowledge base $\mathcal{G}_{\text{know}}$ we constructed can also be referred to as a rule-base, with specific examples provided in the Appendix B.1. Given that the knowledge is predominantly in the form of "*Rules*", our expert knowledge retrieval also adopts a detection engine trigger mechanism similar to that used in rule-based models, as illustrated in Figure 3.

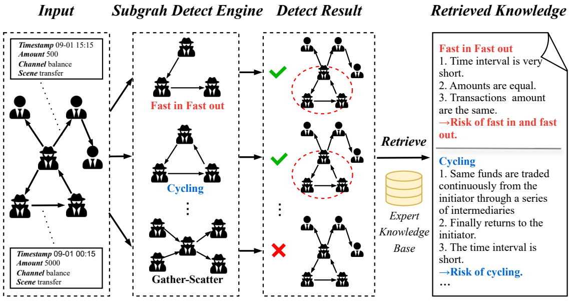


Figure 3: Illustration of Expert Knowledge Retrieval. Expert knowledge is encoded into graphs representing patterns. A detection engine determines if the input graph matches these predefined patterns.

Give an input transaction graph G derived from $\mathcal{T}$, expert knowledge retrieval follows a two-step process: first, subgraph matching is performed to detect if the input matches a known pattern graph $G_{\text{pattern}} \in \mathcal{G}_{\text{know}}$, then the attributes of the nodes and edges in the triggered pattern are examined for further verification. This approach ensures both structural and semantic accuracy by first matching the graph structure and then verifying feature similarity, such as interaction time and transaction amount.

Specifically, for "*Cycling*" subgraph matching, we first adopt Louvain algorithm (Blondel et al., 2008) to derive a set of communities subgraphs of input as $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ where each $S_i(\mathcal{V}_i, \mathcal{E}_i) \subseteq G$ denotes a subgraph and m is the total number of subgraphs. Then, we perform a one-by-one check between S_i and G_{pattern} to determine whether these two graphs are isomorphic, i.e., there exists a mapping $f : \mathcal{V}_i \rightarrow \mathcal{V}_{\text{pattern}}$, ensuring that $\forall (u, v) \in \mathcal{E}_i, (f(u), f(v)) \in \mathcal{E}_{\text{pattern}}$. This process yields a set of triggered patterns $\mathcal{G}' \subseteq \mathcal{G}_{\text{know}}$, representing the retrieved expert knowledge. As shown in Figure 3, parts of the input transaction graph match the *Fast-in-Fast-out* and *Cycling* patterns. We textualize these retrieved patterns to en-

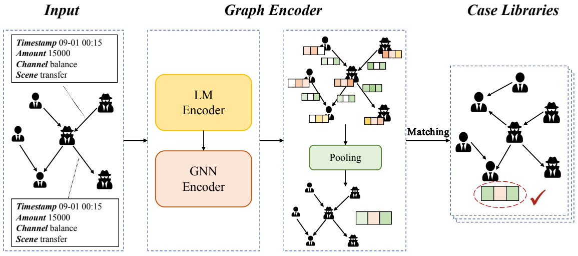


Figure 4: Illustration of Reasoning Cases Retrieval. We use an LM+GNN architecture to encode case and input graphs, retrieving the most similar cases based on embedding similarity.

sure LLMs can comprehend.

3.2.2 Reasoning Cases Retrieval

Similar to the expert knowledge, we construct a set of reasoning cases $\mathcal{G}_{\text{case}} = \{G_{c_1}, \dots, G_{c_M}\}$, where each case G_{c_i} represents a transaction graph instance with a detailed textual analysis report t_{c_i}. These cases, collected from industrial platforms and verified by experts, ensure a diverse and high-quality set of reasoning processes. For detailed examples please refer to Appendix B.2. Unlike expert knowledge patterns with fixed structures, these cases exhibit diverse structures, making traditional graph searching algorithms inapplicable. Thus, we use neural retrieval techniques, encoding both input and case graphs into latent embeddings and matching via similarity, as shown in Figure 4.

Specifically, for the input graph G, we first leverage a pre-trained Language Model to generate initial node embeddings $\mathbf{X}$ for any $u \in \mathcal{V}$. As we can convert record T_i into plain text, an LM encoder can be leveraged to generate T_i's representations as $\mathbf{t}_i = \text{LM}(T_i)$, then a node's initial embedding is computed as mean-pooling of related transactional edges as $\mathbf{x}_u = \text{Mean-Pooling}(\{\mathbf{t}_i | u \in \text{source/target users of } T_i\})$. Then, the GNN further captures the structural patterns of this transaction graph, leading to refined node embedding matrix as $\mathbf{H} = \text{GNN}(\mathbf{X}, \mathcal{E})$. We further use a Read-Out function, i.e., Sum-Pooling, to obtain this transcation graph's embedding as $\mathbf{h}_G = \text{Sum-Pooling}(\mathbf{H})$. As all case graphs in the case library can be encoded in the same way, the case library can be represented as $\mathbf{H}_{\text{case}} = \{\mathbf{h}_{G_{c_1}}, \dots, \mathbf{h}_{G_{c_M}}\}$ where M refers to the number of available cases and $\mathbf{h}_{G_{c_i}}$ denotes the representation of the i-th case graph G_{c_i}. We select the top-K most similar cases from the case library to consisitue the retrival results, as $\mathcal{C}' = \text{Top-K}_{G_{c_i} \in \mathcal{G}_{\text{case}}} \text{Similarity}(\mathbf{h}_G, \mathbf{h}_{G_{c_i}})$.

This method ensures retrieval of the most contextually relevant cases by considering **both the semantic and structural features** of the transaction records. Note that such retrieval can be **training-free**[1] as it leverages a pre-trained LM and a no-parameter GNN, i.e., LightGCN (He et al., 2020). The retrieved reasoning cases are incorporated into the prompts as few-shot-COT examples for the LLMs, steer them to perform chain-of-thought reasoning, thus enhancing their inferential capabilities (Wei et al., 2022).

3.2.3 Input Construction

After completing the dual-retrieval process, we combine three key elements to create comprehensive inputs for LLMs: (1) Expert knowledge serves as the foundational background for model inference. (2) Reasoning cases act as few-shot learning examples to assist the model in analysis. (3) The input transaction data is converted into plain text for the model to analyze. For detailed prompt template and examples, please refer to Appendix C.

4 Experiments

To facilitate explainable transaction risk analysis, we construct a new transaction benchmark dataset: **RA-bench**. Distinct from other anti-money laundering datasets, our dataset not only includes comprehensive labels but also annotates the specific money laundering patterns involved in anomalous transactions. Detailed information about the dataset can be found in Appendix A. We evaluate the performance of Dual-gRAG on the RA-Bench dataset. Our work aims to answer the following three research questions:

- **RQ1:** How much improvement can the Dual-gRAG framework bring to LLMs?
- **RQ2:** How do the two proposed retrieval components contribute to the effectiveness of Dual-gRAG?
- **RQ3:** How does the proposed graph retrieval method perform against other retrieval methods?

Additionally, we conduct a practical application validation in various data scenarios.

4.1 Experimental Settings

4.1.1 Baselines

To demonstrate the effectiveness of the Dual-gRAG framework, we employ several traditional machine

[1] Further discussion please refer to Appendix D

learning models and classic large language models as our baseline.

Traditional Machine Learning Baselines:

- **LightGCN+SVM**: This model uses LightGCN to capture the structural information of the transaction graph, followed by pooling node embeddings for graph-level representation, which is then classified by an SVM.
- **LightGCN+MLP**: This model replaces the SVM with an MLP to explore deep learning performance in this task.
- **GCN+MLP**: Here, a trainable GCN replaces LightGCN, and the GCN and MLP are co-trained in an end-to-end framework for classification.

For all methods, transaction data is converted into a transaction graph, and node features are extracted by language model as described in Section 3.2.2.

LLM Backbones:

- **Llama3-8B-Instruct** (AI@Meta, 2024), a powerful large language model introduced by Meta.
- **Qwen2.5-14B-Instruct** (Bai et al., 2023), a transformer-based decoder-only language model.

For all LLMs, we provide the same prompts to eliminate performance deviations caused by prompt differences.

4.1.2 Evaluation Metrics

We use GPT-4[2] to evaluate the results, mitigating the influence of human emotional bias (Koa et al., 2024). The evaluation process consists of two parts: (1) comparing the risk points identified by the model with those in the ground truth labels[3], and (2) assessing the quality of the explanations generated by the model.

To provide a multi-dimensional view of the LLM's risk analysis capabilities, we employ three evaluation metrics. Additionally, we introduce another metric specifically for machine learning baselines to compare the performance of these methods with that of LLMs in the risk analysis task. The specific evaluation metrics are as follows:

- **Coverage:** Evaluate how many of the risk points identified by the model overlap with the true risk points.
- **Precision:** Evaluate how many of the risk points identified by the model are actually true risk points.
- **ROUGE:** Evaluate the quality of the explanations generated by the model, we use ROUGE-L as the evaluation metric.
- **Accuracy:** Evaluate whether deep learning models or LLMs can correctly identify whether the transaction record graph indicates potential risks.

4.1.3 Implementation Details

For reasoning case retrieval, we employ m3e (Wang Yuxin, 2023) (an open-source massive mixed embedding model) as our language model encoder, top-K is 1. Furthermore, to investigate the improvements that RAG combined with fine-tuning can bring to explainable risk analysis tasks, we utilize the Dual-gRAG framework to construct an instruction fine-tuning dataset. Subsequently, we use LoRA to fine-tune the Qwen2.5-14B on two A100 GPUs with the learning rate of 1e-4 to develop an LLM specifically focused on risk analysis task: **RA-GPT** (Risk Analysis GPT).

4.2 Performance Comparison(RQ1)

For overall performance comparison, we provide both quantitative results (Table 1) and qualitative results (Figure 1).

4.2.1 Quantitative Results

In this section, we quantitatively evaluate the effectiveness of the Dual-gRAG framework in explainable risk analysis task. Table 1 presents the results across four different metrics relevant to the risk analysis task.

Across the first two evaluation metrics, we observe that all LLMs perform better under the Dual-gRAG framework compared to directly analyzing the input data, have nearly a 50% improvement in both metrics. This demonstrates that the Dual-gRAG framework can reduce the generation of irrelevant or incorrect analysis results by LLMs, enhancing the coverage and precision of their outputs. Regarding the "ROUGE" score, which evaluates the quality of the generated explanations, we observe notable improvements in the Dual-gRAG framework as well. The "ROUGE" scores for all models show nearly a twofold improvement, indicating that the Dual-gRAG framework significantly enhances the quality of the model's analysis and

[2] All experimental data has been strictly anonymized, ensuring no risk of privacy leakage.

[3] For detailed evaluation method, please refer to Appendix E

Table 1: Overall Performance Comparison (RQ1). "Direct" refers to LLM's direct inference (example illustrated in Figure 1), and "Dual-gRAG" denotes the integration with our framework, which can consistently bring significant improvement on all metrics.

Model	**Coverage %**		**Precision %**		**ROUGE %**	
	Direct	Dual-gRAG	Direct	Dual-gRAG	Direct	Dual-gRAG
Qwen2.5-14B	51.64	**53.28**	45.32	**72.63**	23.67	**34.92**
Llama3-8B	44.88	**57.79**	48.67	**55.08**	27.54	**32.52**
RA-GPT	47.95	**62.09**	44.32	**66.45**	25.41	**42.14**

Model	SVM$_{LightGCN}$	MLP$_{LightGCN}$	GCN	Qwen2.5-14B	RA-GPT
Accuracy	0.65	0.70	0.72	0.93	0.97

explanations. This leads to more accurate and concise explanations, thereby improving the overall practicality of the model.

We observe that RA-GPT outperforms its base model Qwen2.5-14B, suggesting that fine-tuning can enhance the model's performance in risk analysis tasks. In the "Direct" case, RA-GPT also achieves a higher "ROUGE" score than Qwen2.5, indicating that fine-tuning can also improve the model's reasoning ability. Under the Dual-gRAG framework, the "ROUGE" score of RA-GPT further increases, and the improvement is greater than that from fine-tuning alone, demonstrating that the Dual-gRAG framework provides a more significant performance boost compared to fine-tuning.

Additionally, since machine learning models are non-language models and cannot generate textual analysis results, we only compare the money laundering pattern classification accuracy. We observe that, in the context of this paper, the performance of LLMs significantly outperforms other deep learning models. This could be because, under the Dual-gRAG framework, LLMs can explicitly learn the knowledge of different money laundering patterns, and then use their reasoning abilities, guided by examples, to analyze the potential money laundering issues in the transaction graphs, thereby improving the accuracy of the analysis.

4.2.2 Case Study

In addition to quantitative metrics, we explore how the Dual-gRAG framework enhances the model's performance in risk analysis. For this, we select an example from Qwen2.5's output to demonstrate the improvement, as shown in Figure 1.

From Figure 1, we see that while the LLM correctly identifies risk points in the transaction data, it also generates incorrect (red) and irrelevant (orange) content. With the Dual-gRAG framework, the LLM's analysis becomes more precise, likely due to the retrieved cases that helps the model focus on the provided knowledge. The framework not only improves accuracy but also identifies specific money laundering users, such as the cycle between u000 and u004 (see Figure 1). The step-by-step reasoning rationale in the cases guides the LLM to analyze potential risks thoroughly, resulting in more detailed outcomes.

Table 2: Ablation study of Dual-gRAG with Qwen2.5-14B serving as the base LLM.

Method	**Coverage (%)**	**Precision (%)**	**ROUGE (%)**
Prompt$_{Direct}$	51.64	45.32	23.67
Prompt$_{Know}$	46.31	**74.34**	33.36
Prompt$_{Case}$	**55.12**	68.62	33.69
Prompt$_{Dual}$	53.28	72.63	**34.92**

In contrast to traditional machine learning models with binary outputs, our results provide detailed explanations for suspected transaction risks, offering high interpretability. Staff can easily verify the identified risks during the validation phase.

4.3 Ablation Study (RQ2)

To assess the effectiveness of the Dual-gRAG framework, we conduct an ablation experiments to validate the contribution of each retrieval component. The results are shown in Table 2.

We test different prompting methods, incorporating expert knowledge and reasoning cases, to evaluate performance improvements. The Qwen2.5-14B model was used in frozen mode. The four prompting methods were: Prompt$_{Direct}$ (direct risk analysis), Prompt$_{Know}$ (expert knowledge only), Prompt$_{Case}$ (reasoning cases only), and

Prompt$_{Dual}$.

From Table 2, we observe that Prompt$_{Case}$ performs well. While Prompt$_{Know}$ shows good precision, it has lower coverage than Prompt$_{Direct}$, likely because expert knowledge reduces incorrect risk points, improving precision. However, using only expert knowledge limits the model's inherent knowledge, leading to reduced diversity in the identified risk points, reducing coverage.

Prompt$_{Case}$, by providing detailed case guidance, allows the LLM to infer more potential risk points. However, due to the lack of corresponding expert knowledge, the precision may decrease. In contrast, Prompt$_{Dual}$, through the dual retrieval method, achieves a good balance between coverage and precision.

4.4 Retrieval Method Comparison (RQ3)

In this section, we examine the effectiveness of the graph retriever within the Dual-gRAG framework. Due to the low semantic similarity between the input transaction data and expert knowledge, text similarity cannot be used for expert knowledge retrieval. Thus, we use simple semantic information (i.e., text similarity between transaction in the input data and case library) for reasoning case retrieval. Given that both the input and knowledge base in our task scenario exhibit graph-structured featurese also use the GRAG algorithm as a baseline graph retrieval method (details on baseline selection are in Appendix F). The results are shown in Table 3.

From Table 3, we see that LLM performance with Simple-RAG decreases across all metrics compared to Dual-gRAG. This indicates that relying solely on text similarity reduces retrieval quality and impacts risk analysis capabilities. Additionally, we observe that the performance of GRAG is suboptimal, likely because the GRAG method does not adapt well to our task scenario. Its pruning operation based on semantic information may lead to the loss of key details (e.g., the removal of a transaction related to a crime), which results in degraded retrieval performance.

4.5 Practical Application Validation

Besides the RA-Bench dataset, we also collect a risk analysis validation dataset, "Real-world", from real-world scenarios to validate the generalization ability of the Dual-gRAG framework. In addition, we adopt the "IT-AML"(Altman et al., 2023) dataset proposed by IBM as one of the validation datasets.

Table 3: Comparison of Graph-Retrieval with Qwen2.5-14B serving as the base LLM.

Method	**Coverage** (%)	**Precision** (%)	**ROUGE** (%)
Direct	51.64	45.32	23.67
Simple-RAG	46.93	60.58	21.19
GRAG	45.70	57.77	22.36
Dual-gRAG	**53.28**	**72.63**	**34.92**

Table 4: Generalization validation of the Dual-gRAG framework on different datasets with RA-GPT as the base LLM.

Dataset	**Method**	**Coverage** (%)	**Precision** (%)	**ROUGE** (%)
RA-Bench	Direct	47.95	44.32	25.41
	Dual-gRAG	**62.09**	**66.45**	**42.14**
IT-AML	Direct	**34.01**	36.56	22.32
	Dual-gRAG	26.64	**41.94**	**25.42**
Real-world	Direct	40.38	30.43	17.13
	Dual-gRAG	**44.23**	**41.07**	**21.91**

The generalization study results are shown in Table 4. The Dual-gRAG framework improves the performance of the LLM across different datasets, indicating the strong generalization ability of the Dual-gRAG framework, which is not limited to a single dataset. Furthermore, we can observe that RA-GPT performs similarly on both the RA-Bench and Real-world datasets, suggesting that the generated RA-Bench dataset is highly similar to real-world data. The results from RA-Bench can therefore be easily applied to real-world scenarios, demonstrating a high degree of practicality.

5 Conclusion

In this study, we explore the task of explainable transaction risk analysis, which was difficult to address before the advent of generative models. We identify two key challenges for this task: (1) Deficiency in transaction data analysis and reasoning, and (2) Incompatibility of existing RAG methods with risk analysis scenarios. To tackle these issues, we propose the Dual-gRAG framework, which employs dual-retrieval thought and graph retrieval techniques to provide LLMs with the expert knowledge and reasoning cases necessary for risk analysis. Extensive experiments demonstrate that Dual-gRAG significantly enhances the risk analysis capabilities of LLMs across the three evaluation metrics, with both retrieval components contributing to the model's performance improvement. Furthermore, we show through experiments that the synthetic dataset, RA-Bench, closely resembles real-world scenarios, further enhancing the practicality of the Dual-gRAG framework in real-world applications.

6 Limitations

This paper aims to enable LLMs to perform risk analysis on transaction data through dual retrieval, thereby effectively identifying illicit trading activities. The main limitation of this study is the difficulty in collecting a large amount of real illicit transaction data, which prevents us from deploying Dual-gRAG on a large scale in real-world scenarios for validation. Instead, we generate a synthetic dataset, RA-Bench, from real data to validate our results. In the future, we plan to collect more real-world data to further validate the effectiveness of the framework and apply it to practical scenarios to enhance the framework's practicality.

7 Ethics Statement

The "Real-world" data collected in this study has been strictly anonymized, and the proposed "RA-Bench" dataset is also synthesized from the anonymized "Real-world" data. Therefore, conducting experiments with these two datasets on models like GPT-4 poses no risk of privacy leakage. While the experiments in this paper demonstrate that LLMs perform excellently in interpretable financial risk analysis tasks, we emphasize that all the analysis results are purely academic. Although the conclusions derived from the "Real-world" dataset are consistent with those obtained from the synthetic dataset, further research and validation are necessary before deploying LLMs directly into real-world financial risk analysis scenarios.

References

AI@Meta. 2024. Llama 3 model card.

Ismail Alarab, Simant Prakoonwit, and Mohamed Ikbal Nacer. 2020. Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*.

Erik Altman, Jovan Blanuša, and Luc Von Niederhäusern et al. 2023. Realistic synthetic financial transactions for anti-money laundering models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *ArXiv*, abs/2310.11511.

Jinze Bai, Shuai Bai, and Yunfei Chu et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, and Trevor Cai et al. 2021. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*.

Cameron Browne, Edward Jack Powley, Daniel Whitehouse, Simon M. M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez Liebana, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4:1–43.

Zhengping Che, S. Purushotham, Robinder G. Khemani, and Yan Liu. 2015. Distilling knowledge from deep networks with applications to healthcare domain. *ArXiv*, abs/1512.03542.

Tianyi Chen and Charalampos Tsourakakis. 2022. Antibenford subgraphs: Unsupervised anomaly detection in financial networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. Association for Computing Machinery.

Wenhu Chen, Hexiang Hu, and Xi Chen et al. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Conference on Empirical Methods in Natural Language Processing*.

Kounianhua Du, Renting Rui, Huacan Chai, Lingyue Fu, Wei Xia, Yasheng Wang, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Codegrag: Extracting composed syntax graphs for retrieval augmented cross-lingual code generation. *ArXiv*, abs/2405.02355.

LPaula Ebberth, Ladeira Marcelo, NCarvalho Rommel, and Marzagao Thiago. 2016. Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *ArXiv*, abs/2404.16130.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Jingguang Han, Utsab Barman, Jeremiah Hayes, Jinhua Du, Edward Burgin, and Dadong Wan. 2018. Nextgen aml: Distributed deep learning based language technologies to augment anti money laundering investigation. In *Annual Meeting of the Association for Computational Linguistics*.

Shirley Anugrah Hayati, Raphaël Olivier, Pravalika Avvaru, Pengcheng Yin, Anthony Tomasic, and Graham Neubig. 2018. Retrieval-based neural code generation. *ArXiv*, abs/1808.10025.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhengbao Jiang, Frank Xu, and Luyu et al. Gao. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Martin Jullum, Anders Løland, Ragnar Bang Huseby, Geir Ånonsen, and Johannes Lorentzen. 2020. Detecting money laundering transactions with machine learning. *Journal of Money Laundering Control*, 23(1).

Kelvin J.L. Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. Learning to generate explainable stock predictions using self-reflective large language models. *Proceedings of the ACM on Web Conference 2024*.

Dattatray Vishnu Kute, Biswajeet Pradhan, Nagesh Shukla, and Abdullah Alamri. 2021. Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review. *IEEE Access*, 9:82300–82317.

Nevine Makram Labib, Mohammed Abo Rizka, and Amr Ehab Muhammed Shokry. 2020a. Survey of machine learning approaches of anti-money laundering techniques to counter terrorism finance. In *Internet of Things—Applications and Future: Proceedings of ITAF 2019*, pages 73–87. Springer.

Nevine Makram Labib, Mohammed Abo Rizka, and Amr Ehab Muhammed Shokry. 2020b. Survey of machine learning approaches of anti-money laundering techniques to counter terrorism finance.

Patrick Lewis, Ethan Perez, and Piktus et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xujia Li, Yuan Li, Xueying Mo, Hebing Xiao, Yanyan Shen, and Lei Chen. 2023. Diga: Guided diffusion model for graph recovery in anti-money laundering. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Mingwei Liu, Tianyong Yang, Yiling Lou, Xueying Du, Ying Wang, and Xin Peng. 2023. Codegen4libs: A two-stage approach for library-oriented code generation. *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 434–445.

Marvin Oeben, JeroenP. Goudsmit, and Elena Marchiori. 2019. Prerequisites and ai challenges for model-based anti-money laundering.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E. Leisersen. 2019. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *ArXiv*, abs/1902.10191.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *ArXiv*, abs/2408.08921.

Omri Raiter. 2021. Applying supervised machine learning algorithms for fraud detection in anti-money laundering. *Journal of Modern Issues in Business Research*, 1(1):14–26.

Wenbo Shang and Xin Huang. 2024. A survey of large language models on generative graph analytics: Query, learning, and applications. *Preprint*, arXiv:2404.14809.

Zhihong Shao, Yeyun Gong, and Yelong et al. Shen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.

Toyotaro Suzumura and Hiroki Kanezashi. 2021. Anti-Money Laundering Datasets: InPlusLab anti-money laundering datadatasets. `http://github.com/IBM/AMLSim/`.

S. Tan, Rich Caruana, Giles Hooker, and Yin Lou. 2017. Distill-and-compare: Auditing black-box models using transparent model distillation. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*.

Boxin Wang, Wei Ping, Lawrence C. McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Instructretro: Instruction tuning post retrieval-augmented pretraining. *ArXiv*, abs/2310.07713.

He sicheng Wang Yuxin, Sun Qingxuan. 2023. M3e: Moka massive mixed embedding model.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Chen-Wei Xie, Siyang Sun, Xiong Xiong, Yun Zheng, Deli Zhao, and Jingren Zhou. 2023. Ra-clip: Retrieval augmented contrastive language-image pretraining. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19265–19274.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *Preprint*, arXiv:2306.06031.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Retrieval-augmented multimodal language modeling. *ArXiv*, abs/2211.12561.

Yao Zhang, Yun Xiong, Yiheng Sun, Caihua Shan, Tian Lu, Hui Song, and Yangyong Zhu. 2022. Rudi: Explaining behavior sequence models by automatic statistics generation and rule distillation. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

Wayne Xin Zhao, Kun Zhou, and Junyi Li et al. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Dataset Construction

To facilitate risk analysis research, many researchers have made their risk analysis (e.g., AML) datasets publicly available (Altman et al., 2023). However, a common limitation of these datasets is that they only indicate whether each transaction is suspected of money laundering, without revealing the reasons for this suspicion. Therefore, such datasets are suitable only for training deep learning models but do not fit the scenario of this paper.

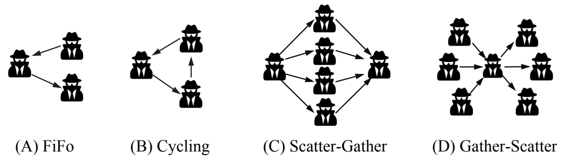


Figure 5: The four most common laundering patterns.

In this work, we construct a new Risk Analysis Benchmark dataset, "RA-Bench", that not only includes comprehensive laundering labels but also specifies the suspected money laundering patterns for each transaction. Given the distinct transaction logic between laundering and normal transactions, we generate them individually. Inspired by Vadim Borisov's GReaT data generation method (Borisov et al., 2023), for normal data, due to its straightforward transaction logic, we utilize LLM to simulate the trading behaviors of individuals in the market, thereby generating normal transaction data. However, the GReaT method generates each record independently, making it incapable of producing sequential data. To address this issue, we ensure that when generating transaction data with LLM, it simultaneously determines the time of the next transaction. This approach enables the LLM to generate serialized data, preserving the temporal characteristics of transaction data and more accurately simulating real-world transactions.

In money laundering cases, criminals often employ specific laundering patterns to carry out their illegal activities. Therefore, we select the four most common patterns (Altman et al., 2023): Fast in Fast out (FiFo), Cycling, Scatter-Gather, and Gather-Scatter (see Figure 5). Based on these four patterns and the distribution of real transaction data, we utilize a simulator adapted from AMLsim (Suzumura and Kanezashi, 2021) to generate risky transaction data and then integrate them into normal transaction data after manual review.

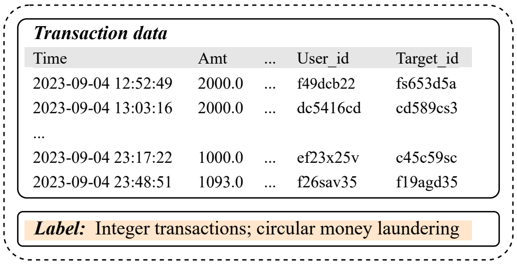
Table 5: Details of RA-Bench dataset.

Statistics of RA-Bench			
# Transactions	709k+	# Illegal transactions	5067
# Fifo groups	686	# Cycling groups	711
# Scatter-gather groups	135	# Gather-scatter groups	129

Example Records from RA-Bench						
Time	Amt	...	User_id	Target_id	Pattern	Group_id
9/1 0:01	500	...	2151e55a		normal	-1
9/1 0:25	3300	...	f49dcb22	8af7060f	loop	2784
9/1 2:38	9000	...	3444f58c	8b6ae7bc	fifo	546

The generated RA-Bench dataset closely resembles the real data distribution and includes comprehensive labels for money laundering activities along with labels for specific laundering patterns. For specific distribution details, please refer to Table 5. In the original RA-Bench dataset, for each money laundering user, we extract their two-hop transaction data and segment it by day, to obtain a more comprehensive daily transaction network that covers all possible illicit transactions for each illegal user. Subsequently, we label each set of extracted transaction data according to the ground truth, resulting in multiple data groups ready for model training and testing. For detailed example of testing data group, please refer to Figure 6.

The reason we construct a synthetic dataset is due to the difficulty in collecting real anti-

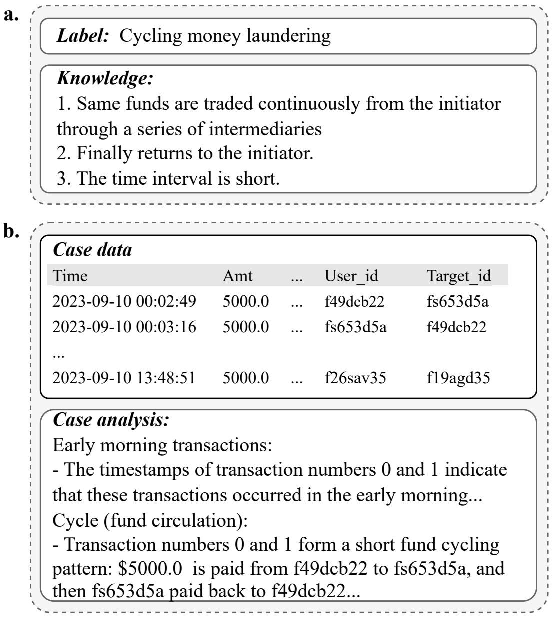


Transaction data

Time	Amt	...	User_id	Target_id
2023-09-04 12:52:49	2000.0	...	f49dcb22	fs653d5a
2023-09-04 13:03:16	2000.0	...	dc5416cd	cd589cs3
...				
2023-09-04 23:17:22	1000.0	...	ef23x25v	c45c59sc
2023-09-04 23:48:51	1093.0	...	f26sav35	f19agd35

Label: Integer transactions; circular money laundering

Figure 6: Example of the extracted data groups.



a.

Label: Cycling money laundering

Knowledge:
1. Same funds are traded continuously from the initiator through a series of intermediaries
2. Finally returns to the initiator.
3. The time interval is short.

b.

Case data

Time	Amt	...	User_id	Target_id
2023-09-10 00:02:49	5000.0	...	f49dcb22	fs653d5a
2023-09-10 00:03:16	5000.0	...	fs653d5a	f49dcb22
...				
2023-09-10 13:48:51	5000.0	...	f26sav35	f19agd35

Case analysis:
Early morning transactions:
- The timestamps of transaction numbers 0 and 1 indicate that these transactions occurred in the early morning...
Cycle (fund circulation):
- Transaction numbers 0 and 1 form a short fund cycling pattern: $5000.0 is paid from f49dcb22 to fs653d5a, and then fs653d5a paid back to f49dcb22...

Figure 7: Example of the two Retrieval Base.

money laundering data, which requires a significant amount of effort to gather and validate the quality of the data. First, suspicious illicit transactions need to be identified from a large amount of transaction data. Then, considerable human resources are required to verify whether these cases are illegal, which is necessary to obtain a valid illicit case. If a large-scale real transaction dataset were to be collected, it would incur incalculable labor costs and raise numerous privacy protection concerns. Therefore, this paper chose to start from a small amount of real transaction data and synthesize a high-quality transaction dataset, "RA-Bench", for research purposes.

B Retrieval Knowledge Base

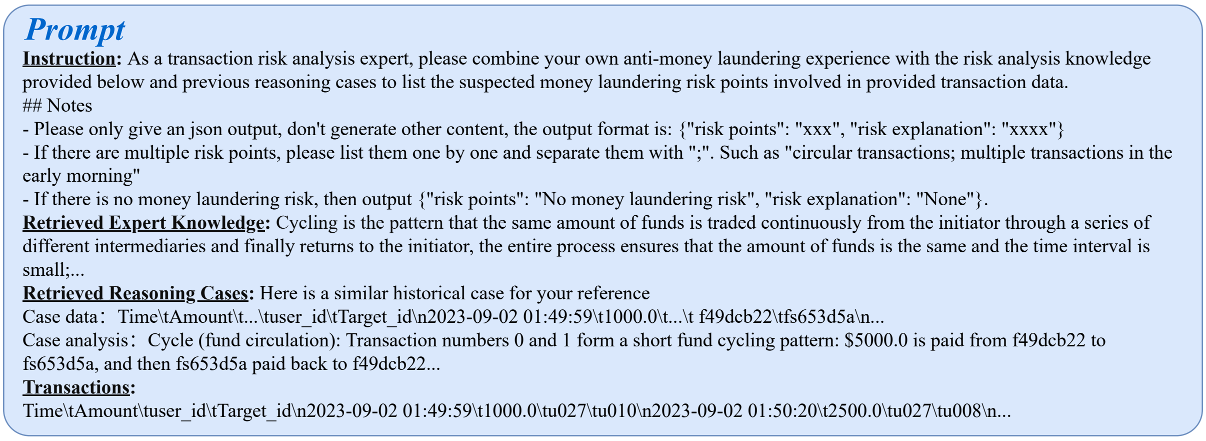
B.1 Expert Knowledge Base

From the collected analysis reports, we extract critical expert knowledge related to risk analysis. This encompasses not only the textual descriptions of the four money laundering patterns previously discussed but also three additional common risk indicators (e.g., multiple transactions during early hours and sudden large transactions). As mentioned earlier, in transaction risk analysis tasks, expert knowledge is often represented in the form of rules. Therefore, our expert knowledge base is more like a rule base, such as the textual descriptions of the four money laundering patterns. However, as we have discussed, a major limitation of rule-based models is their vulnerability to being circumvented by criminals. To address this, we have added three common risk indicators in addition to the four typical money laundering patterns, aiming to combine the reasoning ability of LLMs with these common risk analysis insights to capture potential new risk points, thereby compensating for the limitations of rule-based models. After thorough manual review, we integrate the knowledge associated with these seven risk points into our expert knowledge base. For specific examples of the knowledge base, please refer to Figure 7 (a).

For example, a high-risk transaction pattern is the "*Cycling*" mode, where transaction records form a closed loop. Mathematically, this can be described as: a source user u_1 transfers money to a target user u_2, then u_2 makes a new transaction with u_3, and after several intermediary transactions, a user u_k transfers it back to the initial user u_1. This cycle pattern is represented as a graph G_{cycle} with a node set representing distinct users $\mathcal{V}_{\text{cycle}} = \{v_1, ..., v_k\}$, and edges forming a cycle $\mathcal{E}_{\text{cycle}} = \{(v_1, v_2), \ldots, (v_k, v_1)\}$. Similarly, we can encode a variety of transaction patterns in graph format, resulting in a **graph-formatted expert knowledge base** $\mathcal{G}_{\text{know}} = \{G_{\text{cycle}}, G_{\text{Fast-in-Fast-out}}, G_{\text{Scatter}}, \ldots\}$, where each G_{pattern} represents a graph encoding a distinct pattern of a high-risk laundering mode. Illustrative examples of these patterns are shown in Figure 3.

B.2 Reasoning Case Library

After aggregating a substantial number of historical analysis cases from industrial platforms, we meticulously filter and select representative cases that encompass diverse combinations of risk indicators to ensure the comprehensive diversity of the case library. Subsequently, domain experts rigorously examine these cases to validate the accuracy of the analysis results and processes, further refining the analytical procedures embedded within each case. Through extensive and systematic validation, we


Prompt

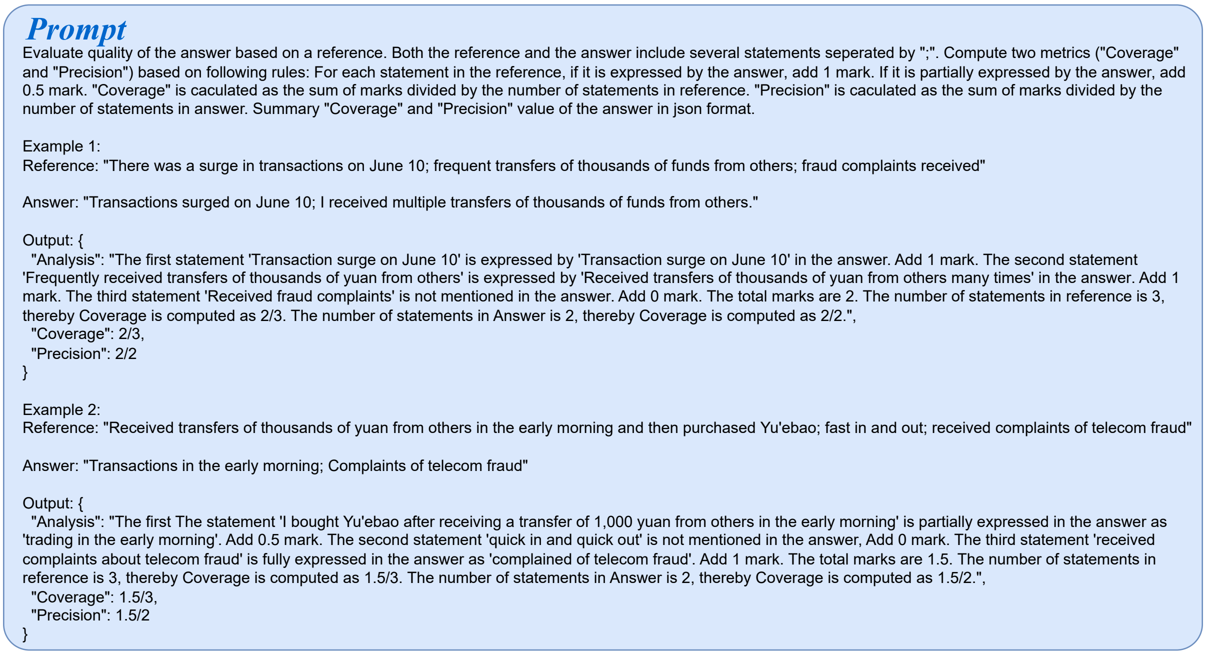
Instruction: As a transaction risk analysis expert, please combine your own anti-money laundering experience with the risk analysis knowledge provided below and previous reasoning cases to list the suspected money laundering risk points involved in provided transaction data.
Notes
- Please only give an json output, don't generate other content, the output format is: {"risk points": "xxx", "risk explanation": "xxxx"}
- If there are multiple risk points, please list them one by one and separate them with ";". Such as "circular transactions; multiple transactions in the early morning"
- If there is no money laundering risk, then output {"risk points": "No money laundering risk", "risk explanation": "None"}.

Retrieved Expert Knowledge: Cycling is the pattern that the same amount of funds is traded continuously from the initiator through a series of different intermediaries and finally returns to the initiator, the entire process ensures that the amount of funds is the same and the time interval is small;...

Retrieved Reasoning Cases: Here is a similar historical case for your reference
Case data：Time\tAmount\t...\tuser_id\tTarget_id\n2023-09-02 01:49:59\t1000.0\t...\t f49dcb22\tfs653d5a\n...
Case analysis：Cycle (fund circulation): Transaction numbers 0 and 1 form a short fund cycling pattern: $5000.0 is paid from f49dcb22 to fs653d5a, and then fs653d5a paid back to f49dcb22...

Transactions:
Time\tAmount\tuser_id\tTarget_id\n2023-09-02 01:49:59\t1000.0\tu027\tu010\n2023-09-02 01:50:20\t2500.0\tu027\tu008\n...

Figure 8: Example of the prompt we used in this study.


Prompt

Evaluate quality of the answer based on a reference. Both the reference and the answer include several statements seperated by ";". Compute two metrics ("Coverage" and "Precision") based on following rules: For each statement in the reference, if it is expressed by the answer, add 1 mark. If it is partially expressed by the answer, add 0.5 mark. "Coverage" is caculated as the sum of marks divided by the number of statements in reference. "Precision" is caculated as the sum of marks divided by the number of statements in answer. Summary "Coverage" and "Precision" value of the answer in json format.

Example 1:
Reference: "There was a surge in transactions on June 10; frequent transfers of thousands of funds from others; fraud complaints received"

Answer: "Transactions surged on June 10; I received multiple transfers of thousands of funds from others."

Output: {
"Analysis": "The first statement 'Transaction surge on June 10' is expressed by 'Transaction surge on June 10' in the answer. Add 1 mark. The second statement 'Frequently received transfers of thousands of yuan from others' is expressed by 'Received transfers of thousands of yuan from others many times' in the answer. Add 1 mark. The third statement 'Received fraud complaints' is not mentioned in the answer. Add 0 mark. The total marks are 2. The number of statements in reference is 3, thereby Coverage is computed as 2/3. The number of statements in Answer is 2, thereby Coverage is computed as 2/2.",
"Coverage": 2/3,
"Precision": 2/2
}

Example 2:
Reference: "Received transfers of thousands of yuan from others in the early morning and then purchased Yu'ebao; fast in and out; received complaints of telecom fraud"

Answer: "Transactions in the early morning; Complaints of telecom fraud"

Output: {
"Analysis": "The first The statement 'I bought Yu'ebao after receiving a transfer of 1,000 yuan from others in the early morning' is partially expressed in the answer as 'trading in the early morning'. Add 0.5 mark. The second statement 'quick in and quick out' is not mentioned in the answer, Add 0 mark. The third statement 'received complaints about telecom fraud' is fully expressed in the answer as 'complained of telecom fraud'. Add 1 mark. The total marks are 1.5. The number of statements in reference is 3, thereby Coverage is computed as 1.5/3. The number of statements in Answer is 2, thereby Coverage is computed as 1.5/2.",
"Coverage": 1.5/3,
"Precision": 1.5/2
}

Figure 9: The prompt we used for GPT4 evaluation.

guarantee that the case library is not only diverse but also of exceptional quality. For specific examples of the case library, please refer to Figure 7 (b).

C Prompt template

In this paper, the expert knowledge and reasoning cases obtained through the dual-retrieval mechanism play distinct yet complementary roles during model inference. Expert knowledge acts as foundational background information, enriching LLMs with comprehensive domain knowledge that underpins their risk analysis capabilities. Meanwhile, the reasoning cases, which detail step-by-step analytical processes, function as few-shot learning examples, offering LLMs reference material that significantly enhances their inferential capabilities in performing risk analysis tasks. For detailed prompt templates, please refer to Figure 1. Additionally, we provide a simplified example to illustrate our prompt, as shown in Figure 8.

D Choice of GNN-Encoder

In this paper, we adopt the non-parametric LightGCN as our GNN Encoder model to enhance the practicality and generalization of our model. This is because, if we had used a parametric GNN Encoder model (such as the GCN model), changes in the case library or input dataset could require retraining the model for the new dataset and case library. However, with a non-parametric GNN model, we can directly transfer the model to new data scenarios since it does not require retraining.

Furthermore, during the early exploration phase, we also test parametric models. However, after trying multiple GNN models, we conclude that the improvement provided by parametric GNNs is not significant enough to warrant switching to them. Therefore, we decide to use the non-parametric model as the GNN Encoder, further enhancing the generalization and practicality of the Dual-gRAG framework.

E Evaluation Method

We leverage GPT-4 to evaluate the results to mitigate the interference from human emotion, the prompt used for evaluation can be found in Figure 9. The scoring process involves comparing the model's identified risk points against those in the ground truth label, a full match earning 1 point, partial similarity 0.5 points, and no match 0 points. The final score is the total of all risk points in the model's output. A specific example can be seen in Figure 10.

Aimed at presenting the model's risk analysis capabilities, we design two evaluation metrics intended to assess the model's strengths and effectiveness from multiple dimensions. The specific evaluation metrics are as follows.

• Coverage: We aspire for the model's outcomes to encompass all potential risks involved in the given set of transactions as comprehensively as possible. A higher degree of coverage indicates a greater capability of the model in risk analysis. The calculation of coverage involves dividing the total score by the number of risk points identified in expert reports.

• Precision: In addition to aiming for the model to comprehensively cover all risks, we also desire for the model's output to be more precise. Higher precision means the model is less likely to output irrelevant risk points. The calculation method for precision is dividing the total score by the number of risk points identified in the model's results.

F Baseline Selection

To further demonstrate the effectiveness of the Dual-gRAG framework, we also identify comparable graph retrieval algorithms. However, most existing graph RAG algorithms focus on retrieving corresponding content from a graph database based on a textual query (Peng et al., 2024). Many Graph RAG algorithms begin by identifying a node in a graph database (such as a knowledge graph) that corresponds to the entity of the input query, then expanding to paths or subgraphs in the graph. The retrieved content (paths, subgraphs) is then converted into text and provided to the LLM as retrieved knowledge. This "question entity" to "knowledge base node" correspondence does not exist in our task scenario, and thus cannot be applied to our problem. Furthermore, their inputs do not exhibit any graph-structured features, making them incompatible with our task scenario.

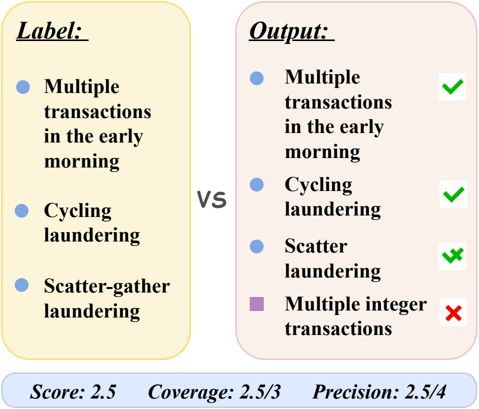

Label:
Multiple transactions in the early morning
Cycling laundering
Scatter-gather laundering
VS
Output:
Multiple transactions in the early morning
Cycling laundering
Scatter laundering
Multiple integer transactions
Score: 2.5
Coverage: 2.5/3
Precision: 2.5/4

Figure 10: Example of scoring for the evaluation metrics.

Therefore, we adapt GRAG as our baseline, as it can convert a subgraph (similar to our transaction records) into graph embeddings, enabling us to perform case retrieval by computing the similarity of embeddings.

G Efficiency Discussion

In the financial transaction risk analysis scenario, an important issue that needs to be addressed is the efficiency of the model. The complexity of our framework is not high. The expert knowledge retrieval module relies on a traditional subgraph matching strategy, with complexity scaling linearly with the number of patterns. For case retrieval, the graph convolution complexity also scales linearly with the number of edges, as we utilize LightGCN, and each graph is relatively small, maintaining high efficiency. Therefore, the overall complexity of our retrieval framework remains $\mathcal{O}(\mathcal{V} + \mathcal{E})$. Therefore, our computational efficiency is also quite high. Additionally, our analysis scenario involves analyzing the transaction data network of a user within a single day, focusing on 1-hop or 2-hop transactions. Therefore, the transaction network to be analyzed is not particularly large, which means that the retrieval and analysis time will not be excessively long. Furthermore, when business staff use LLMs for risk analysis, the LLM can directly inform the staff about the transaction risks associated with a user. The staff only need to perform simple validation, without the need to analyze the specific risk points associated with the user as in the past. This improves the efficiency of transaction risk analysis and reduces the manual labor costs involved.