

Vol 2 Issue 03 (2026)

<http://fsrma.org.hostajournal.com/>

[ONLINE ISSN: 3134-7517](#)

[PRINT ISSN: 3134-7509](#)

FinGraph-MAR: A Graph Retrieval-Augmented Multi-Agent Framework for Intelligent Financial Back-Office Automation and SaaS Revenue Management

Yang Shi

Financial Mathematics, Case Western Reserve University, Cleveland OH, USA
yxs879@case.edu

Leshen Sui

Columbia University in the City of New York, New York, NY, USA
ls3452@columbia.edu

Xingyu Liu

University of Chicago, Chicago, IL, USA
xingyul@uchicago.edu

Abstract.

The increasing complexity of financial back-office operations and SaaS revenue management has created significant challenges for intelligent automation, particularly in transaction processing, billing verification, revenue recognition, compliance monitoring, and operational decision-making. Existing robotic process automation (RPA) systems mainly rely on predefined workflows and struggle to handle dynamic business relationships, heterogeneous financial data, and knowledge-intensive tasks involving customers, contracts, subscriptions, transactions, and regulatory requirements. To address these limitations, this study proposes FinGraph-MAR (Financial Graph-based Multi-Agent Reasoning), a graph retrieval-augmented multi-agent framework for intelligent financial back-office automation and SaaS revenue management. The proposed framework constructs a heterogeneous financial-SaaS knowledge graph to model entities, events, business relationships, and operational rules, and introduces a hybrid graph-semantic retrieval mechanism to provide relationship-aware contextual knowledge for large language model (LLM)-based agents. Furthermore, FinGraph-MAR integrates a multi-agent collaboration architecture consisting of task planning, transaction analysis, revenue management, risk assessment, compliance verification, execution, and audit agents, together with a risk-constrained decision gate for reliable automated operations. Experimental results on four publicly available datasets, including FinQA, TAT-QA, UCI Online Retail, and IBM Telco Customer Churn, demonstrate that FinGraph-MAR achieves 95.6% task understanding accuracy, 94.9% F1-score, and 97.1% workflow completion rate, outperforming rule-based RPA, standalone LLM, conventional RAG, LLM+RPA, and Graph RAG+LLM baselines. Ablation experiments further

confirm the contributions of graph retrieval, multi-agent collaboration, risk gating, and audit feedback.

Keywords:

Financial back-office automation; Large language models; Graph retrieval-augmented generation; Multi-agent systems; SaaS revenue management; Knowledge graphs; Risk-aware decision-making; Robotic process automation

1. Introduction

Financial back-office operations constitute a critical infrastructure for modern financial institutions and digital enterprises. Unlike customer-facing financial services, back-office processes involve transaction reconciliation, payment verification, account maintenance, billing management, revenue recognition, compliance checking, exception handling, audit preparation, and operational reporting. These processes frequently span multiple enterprise systems and require the integration of structured transaction records, contractual documents, financial policies, historical cases, and customer information. The increasing adoption of digital financial services and software-as-a-service (SaaS) business models has further increased the complexity of these operations.

Traditional robotic process automation (RPA) has been widely adopted to automate repetitive financial tasks. RPA can efficiently perform deterministic operations such as transferring records between systems, generating invoices, synchronizing databases, and producing predefined reports. However, conventional RPA depends heavily on fixed rules and predefined workflows. When a task requires interpretation of a contract, investigation of an abnormal transaction, comparison of historical cases, or adaptation to a new financial policy, rule-based automation becomes difficult to maintain. In particular, financial operations increasingly involve dynamic relationships among customers, accounts, transactions, contracts, subscriptions, usage records, invoices, and payments. Such relationships cannot be adequately represented by isolated workflow rules. Large language models (LLMs) have recently demonstrated strong capabilities in natural language understanding, reasoning, information extraction, and task planning. Their application to financial operations provides an opportunity to transform conventional workflow automation into intelligent decision-oriented automation. An LLM-based agent can interpret natural-language instructions, analyze financial documents, summarize transaction information, and formulate operational decisions. Nevertheless, directly applying LLMs to financial operations introduces several challenges. First, LLMs may generate plausible but unsupported financial conclusions due to hallucination. Second, general-purpose LLMs lack explicit representations of complex financial relationships. Third, financial decisions frequently require simultaneous reasoning over structured data and unstructured regulatory or contractual documents. Finally, even when an LLM generates an appropriate recommendation, the generated decision cannot be directly executed without risk and compliance validation. The importance of adaptive risk assessment is also well established in data-driven financial decision systems. Li et al. [1] developed an adaptive credit-default detection framework based on Random Forest and Neural Network models, demonstrating that complex customer, billing, credit, and delinquency patterns can provide effective signals for identifying financial risk. Their study highlights an important limitation of static rule-based control: risk decisions increasingly depend on multidimensional behavioral evidence rather than isolated thresholds. This requirement is directly relevant to intelligent financial back-office automation, where automated actions should be conditioned on dynamically assessed transaction, customer, billing, and historical risk information. Retrieval-augmented generation (RAG) has been proposed to improve the factual grounding of LLMs by retrieving external knowledge before generation. Conventional RAG primarily relies on vector similarity between a query and independent text

chunks. Although effective for semantic retrieval, this strategy may fail to capture multi-hop relationships among financial entities. For example, determining whether a SaaS invoice is anomalous may require connecting a customer's subscription plan, contract pricing rule, recent product upgrade, actual usage, invoice amount, payment history, and relevant financial policies. These relationships are inherently graph-structured.

Knowledge graphs provide a natural representation for such interconnected information. By explicitly modeling entities, relationships, events, and rules, a financial knowledge graph enables an LLM to reason over structured business relationships. However, knowledge graphs alone do not solve the problem of complex operational decision-making. Different financial tasks require different types of expertise, including transaction analysis, revenue management, risk assessment, compliance verification, and system execution. Multi-agent systems therefore provide a promising mechanism for decomposing complex financial workflows into specialized tasks and coordinating multiple reasoning processes.

Motivated by these observations, this study proposes FinGraph-MAR (Financial Graph-based Multi-Agent Reasoning), a graph retrieval-augmented multi-agent framework for intelligent financial back-office automation and SaaS revenue management. The central objective is not merely to use an LLM to generate financial answers, but to establish a complete pipeline from business knowledge retrieval to multi-agent reasoning, risk-constrained decision-making, and automated execution.

The main contributions of this study are summarized as follows:

A heterogeneous financial-SaaS knowledge graph is developed to jointly represent customers, accounts, transactions, contracts, subscriptions, invoices, payments, revenue events, financial policies, and historical operational cases.

A hybrid graph-semantic retrieval mechanism is proposed to combine subgraph retrieval with semantic document retrieval, enabling LLM agents to access both explicit financial relationships and supporting textual knowledge.

A specialized multi-agent reasoning architecture is designed, consisting of a task planning agent and specialized transaction, revenue, risk, compliance, execution, and audit agents for collaborative financial back-office operations.

A risk-constrained decision gate is introduced to determine whether a generated financial decision should be automatically executed, submitted for human review, or rejected according to risk, compliance, and confidence conditions.

A closed-loop execution and audit mechanism is established, connecting LLM-based reasoning with RPA/API execution while preserving decision evidence and operational feedback.

2.Literature Review

The development of intelligent financial operations has increasingly shifted from conventional rule-based automation toward knowledge-grounded, language-driven, and agentic systems. In particular, recent research has explored robotic process automation (RPA) for repetitive financial workflows, large language models (LLMs) for financial information processing, Retrieval-Augmented Generation (RAG) for knowledge-intensive reasoning, graph-based retrieval for relationship-aware information access, and multi-agent architectures for complex task execution. These research directions provide important foundations for intelligent financial back-office automation and SaaS revenue management. However, existing studies generally address these capabilities independently and rarely integrate transaction processing, SaaS billing and revenue management, compliance reasoning, risk control, and automated execution within a unified

framework. Therefore, this section reviews five closely related research directions and identifies the gap addressed by FinGraph-MAR.

2.1 Robotic Process Automation for Financial and Back-Office Operations

Robotic Process Automation (RPA) has been widely investigated as an approach for automating repetitive, rule-based business processes involving structured data and deterministic workflows. Aguirre and Rodriguez [2] demonstrated that RPA can be applied to business processes involving routine operations and reported improvements in productivity, illustrating its applicability to back-office automation. Similarly, Lacity and Willcocks [3] investigated RPA adoption in Telefónica O2 and showed that software robots can improve the speed and accuracy of routine back-office activities. Willcocks et al. [4] further examined RPA in global business services and emphasized that successful automation requires not only technical deployment but also appropriate process design and risk management. These studies establish the practical value of RPA for transaction processing, reconciliation, data transfer, and other repetitive financial operations. However, conventional RPA fundamentally depends on predefined rules, interfaces, and workflow templates. Consequently, it is less suitable for tasks requiring interpretation of contracts, financial policies, regulatory documents, customer-specific billing conditions, or relationships among transactions and revenue events. This limitation becomes more significant in SaaS environments, where subscription upgrades, downgrades, renewals, refunds, usage-based charges, and deferred revenue may dynamically change the financial state. Therefore, RPA provides an important execution layer for financial automation but requires intelligent reasoning mechanisms to determine what should be executed and under which conditions.

2.2 Large Language Models for Financial Information Processing and Reasoning

The emergence of LLMs has substantially expanded the capability of artificial intelligence systems to understand unstructured financial information. Lou et al. [5] demonstrated that large-scale language models can perform a broad range of language tasks through few-shot prompting, establishing a foundation for instruction-driven reasoning. Wei et al. [6] further showed that Chain-of-Thought prompting can improve multi-step reasoning by explicitly eliciting intermediate reasoning processes. These capabilities are particularly relevant to financial operations, where decisions often require combining numerical information, contractual conditions, and textual regulations. Domain-specific financial language models have subsequently received increasing attention. Wu et al. [7] introduced BloombergGPT, a 50-billion-parameter language model trained on extensive financial and general-domain corpora, demonstrating strong performance across financial NLP tasks. Yang et al. [8] proposed FinGPT as an open-source financial LLM framework and emphasized data curation, efficient adaptation, and practical financial applications. Financial reasoning benchmarks have also highlighted the difficulty of applying general-purpose LLMs directly to financial tasks. Chen et al. [9] introduced FinQA, which contains expert-written questions over financial reports and explicitly annotated reasoning programs for numerical reasoning. Zhu et al. [10] developed TAT-QA to evaluate reasoning over hybrid tabular and textual financial information, demonstrating that financial question answering frequently requires numerical computation and the integration of heterogeneous evidence. Although these studies demonstrate the strong potential of LLMs for financial analysis, an LLM alone does not guarantee reliable financial decision-making. Financial back-office tasks require current enterprise information, contractual constraints, accounting rules, and auditable evidence rather than solely relying on parametric knowledge. Thus, LLMs need to be integrated with external knowledge sources, structured financial information, and controlled execution mechanisms.

2.3 Retrieval-Augmented Generation and Graph-Based Financial Knowledge Retrieval

RAG has emerged as an important strategy for grounding LLM outputs in external knowledge. Rather than relying exclusively on parametric knowledge, RAG retrieves relevant information and incorporates it into the generation process. Recent research has further focused on improving the reliability and adaptiveness of retrieval. Asai et al. [11] proposed Self-RAG, which enables a model to retrieve information adaptively and perform self-reflection over retrieved evidence and generated responses. Yan et al. [12] introduced Corrective Retrieval-Augmented Generation (CRAG), which evaluates retrieval quality and activates corrective retrieval operations when the retrieved evidence is insufficient or unreliable. Niu et al. [13] proposed RAGTruth, a manually annotated corpus for evaluating hallucinations in RAG systems, highlighting that retrieval augmentation does not completely eliminate unsupported or contradictory generation. However, conventional RAG generally represents knowledge as independent text chunks and retrieves documents according to semantic similarity. Such a strategy may be insufficient for financial operations because many decisions depend on multi-hop relationships. For example, determining whether a SaaS invoice is compliant may require connecting a customer to a contract, the contract to a subscription, the subscription to a billing event, and the billing event to a revenue recognition rule. These relationships cannot always be adequately represented through isolated text passages. Graph-based retrieval therefore provides a promising alternative. Edge et al. [14] proposed GraphRAG, which constructs an entity-oriented graph index and community summaries to support reasoning over large private corpora. Their results indicate that graph-based retrieval can improve the comprehensiveness and diversity of responses for complex corpus-level questions compared with conventional RAG. Nevertheless, existing GraphRAG research primarily focuses on information discovery and question answering rather than operational financial decision-making. It does not explicitly model financial transaction events, SaaS revenue events, accounting rules, compliance constraints, or executable business workflows. This motivates the construction of a domain-specific heterogeneous financial-SaaS knowledge graph in FinGraph-MAR.

2.4 Multi-Agent Reasoning and Tool-Augmented Financial Automation

Beyond retrieval, complex financial back-office tasks often require multiple stages of planning, reasoning, verification, execution, and auditing. Recent research has therefore moved from single-agent LLM systems toward multi-agent architectures. Yao et al. [15] proposed ReAct, which interleaves reasoning and action generation and enables language models to interact with external information sources and environments. This work demonstrates that combining reasoning with actions can improve task-solving capability and interpretability. Schick et al. [16] introduced Toolformer, showing that language models can learn when and how to invoke external APIs, thereby extending LLM capabilities beyond pure text generation. More recently, Wu et al. [17] introduced AutoGen, a framework for constructing LLM applications through conversations among multiple customizable agents. Guo et al. [18] provided a systematic survey of LLM-based multi-agent systems and emphasized agent specialization, communication, planning, and collaborative problem solving. These studies provide important foundations for decomposing complex financial workflows into specialized agents. Nevertheless, generic multi-agent frameworks typically do not incorporate financial-specific risk constraints or compliance gates. In financial back-office scenarios, an agent should not directly execute an action merely because its language reasoning indicates that the action is appropriate. Instead, transaction evidence, contract conditions, compliance rules, and risk scores should be jointly evaluated before execution. This requirement motivates the risk-constrained multi-agent architecture of FinGraph-MAR, where

specialized transaction, revenue, risk, compliance, execution, and audit agents collaborate under an explicit decision-control mechanism.

3. Methodology

3.1 Overall Architecture of FinGraph-MAR

This study proposes FinGraph-MAR (Financial Graph-based Multi-Agent Reasoning), a graph retrieval-augmented multi-agent framework designed to automate financial back-office operations while supporting SaaS revenue management. The framework is motivated by the observation that financial operational decisions rarely depend on a single record or isolated document. Instead, they require the joint interpretation of transactions, customers, accounts, contracts, subscriptions, invoices, payments, revenue events, regulatory rules, and historical cases. FinGraph-MAR therefore integrates structured relationship modeling, semantic retrieval, multi-agent reasoning, risk control, and automated execution into a unified decision pipeline. The overall workflow consists of six interconnected stages: heterogeneous financial and SaaS data ingestion, financial-SaaS knowledge graph construction, graph-semantic retrieval, multi-agent collaborative reasoning, risk-constrained decision making, and controlled execution with audit feedback. Given an operational request q , the system first identifies relevant entities and business events from the underlying data sources. The retrieved information is then organized into a task-specific graph context and textual evidence set. Specialized agents independently analyze different aspects of the task and exchange intermediate results through a shared task state. The final decision is passed through a risk-constrained gate before an action can be executed through Robotic Process Automation (RPA) or an Application Programming Interface (API). The complete process can be represented as

$$Y = E_{\text{exec}} \left(G_{\text{risk}} \left(M_{\text{agent}} \left(R_{\text{graph}}(q, G, D) \right) \right) \right)$$

where D denotes multi-source operational data, G denotes the financial-SaaS knowledge graph, R_{graph} represents graph-semantic retrieval, M_{agent} denotes multi-agent reasoning, G_{risk} denotes the risk and compliance decision gate, and E_{exec} represents controlled workflow execution. This design transforms FinGraph-MAR from a question-answering system into an operational decision framework capable of producing, validating, executing, and auditing financial actions.

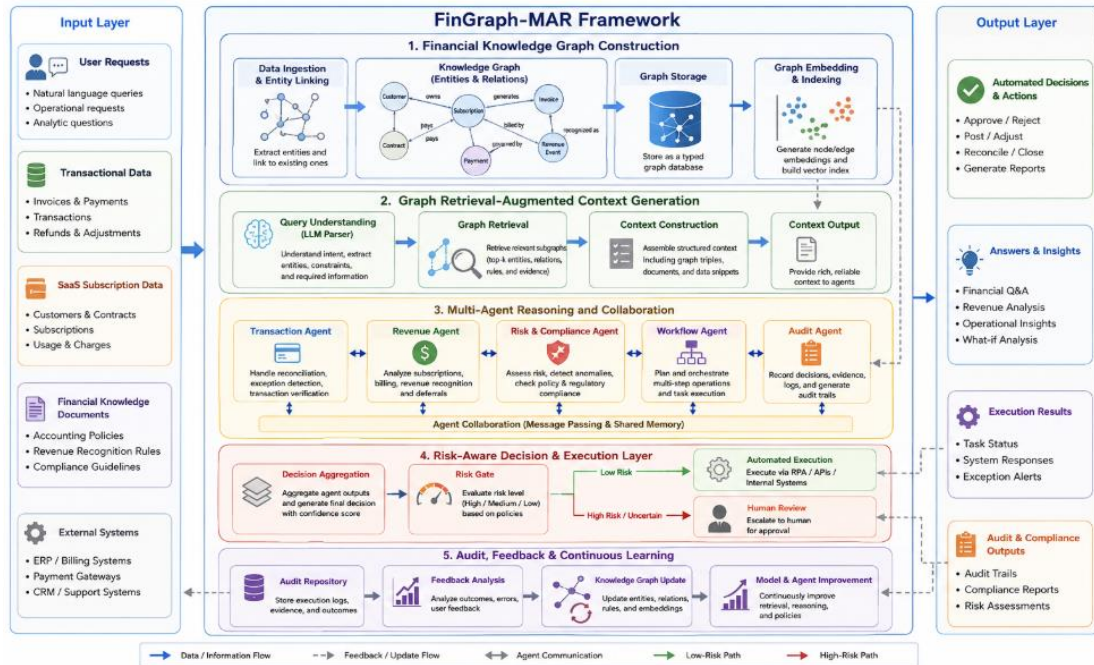


Figure 1. Overall architecture of FinGraph-MAR. FinGraph-MAR integrates a Financial Knowledge Graph with multi-agent reasoning and risk-aware execution to automate complex back-office with findeork receives multi-source inputs, retrieves graph-structured and textual context, leverages specialized agents for collaborative reasoning, enforces risk-aware decision control, executes actions through enterprise systems, and learns continuously through audit feedback and graph updates.

Figure 1Overall flowchart of the model.

3.2 Multi-Source Financial and SaaS Data Representation

FinGraph-MAR integrates heterogeneous information generated throughout financial back-office and SaaS revenue processes. Structured data include transaction records, customer accounts, payment histories, subscription states, invoices, product information, usage records, and revenue events. Semi-structured and unstructured information includes customer contracts, billing terms, accounting policies, compliance documents, operational guidelines, and historical investigation reports. This combination is necessary because financial decisions often require evidence from both numerical records and textual business rules. For SaaS revenue management, the framework explicitly models the lifecycle from customer acquisition to contract execution, subscription activation, billing, payment, renewal, upgrade, downgrade, refund, and revenue recognition. For example, a billing anomaly cannot be reliably identified by comparing an invoice amount with a predefined threshold alone. The system may need to determine whether the invoice corresponds to the customer's active subscription, whether the contracted pricing model permits the observed usage charge, whether a discount or upgrade occurred during the billing period, and whether the resulting revenue event satisfies the applicable recognition rule. Consequently, each operational record is transformed into a normalized representation containing entity identifiers, timestamps, monetary attributes, event types, source information, and confidence metadata. This representation follows a broader data-driven optimization principle in which operational observations are first transformed into structured variables that can support downstream prediction, evaluation, and optimization. Data-driven operational optimization has also been explored in industrial systems; for instance, Ning et al. [19] investigated the use of data-driven algorithms for improving air-compressor energy efficiency. FinGraph-MAR adopts the general principle of exploiting operational data for adaptive decision support, but extends it to a substantially different decision environment in which individual observations are not independent. Financial records are therefore organized as temporally and semantically connected entities and events, enabling subsequent

graph retrieval and multi-agent reasoning to account for dependencies among customers, contracts, transactions, subscriptions, and operational rules.

3.3 Financial-SaaS Heterogeneous Knowledge Graph

The core knowledge representation component is a Financial-SaaS Heterogeneous Knowledge Graph (FS-HKG). Unlike a conventional document repository, FS-HKG explicitly represents the relationships among financial entities, operational events, rules, and historical cases. Formally, the graph is defined as

$$G = (V, E, A)$$

where V represents heterogeneous nodes, E represents typed relationships, and A contains node and edge attributes. The node set is divided into four categories:

$$V = V_{\text{entity}} \cup V_{\text{event}} \cup V_{\text{rule}} \cup V_{\text{case}}$$

Entity nodes include customers, accounts, contracts, subscriptions, products, invoices, payments, and transactions. Event nodes represent billing, payment, renewal, upgrade, downgrade, refund, and revenue-recognition events. Rule nodes represent accounting rules, regulatory requirements, internal policies, and contract-specific constraints, while case nodes represent historical anomalies, investigations, audit cases, and previously resolved operational tasks.

Typed edges encode business dependencies such as *Customer* $\rightarrow$ *Contract*, *Contract* $\rightarrow$ *Subscription*, *Subscription* $\rightarrow$ *Invoice*, *Invoice* $\rightarrow$ *Payment*, *Contract* $\rightarrow$ *RevenueRule* and *RevenueEvent* $\rightarrow$ *Regulation*. Temporal attributes are attached to event relationships so that the system can distinguish, for example, an invoice generated before a subscription upgrade from an invoice generated after the upgrade. This temporal-relational structure is particularly important for SaaS revenue management because billing and revenue recognition are determined not only by entity relationships but also by the timing and sequence of financial events.

3.4 Hybrid Graph-Semantic Retrieval

To overcome the limitations of conventional vector-based RAG, FinGraph-MAR employs a hybrid retrieval mechanism that jointly considers semantic similarity and graph relationships. This hybrid design is motivated by the complementary roles of semantic representation and relational structure in information retrieval. Related work by Wang et al. [20] combines knowledge-graph information with deep-learning-based semantic modeling for advertisement retrieval and personalization, indicating that relational knowledge and learned semantic features can provide complementary signals for information access. In FinGraph-MAR, this principle is adapted to financial operations by jointly considering semantic relevance, graph connectivity, and risk-sensitive information during retrieval. Given an operational query q , an entity-linking module first identifies relevant graph entities:

$$L(q) = \{v_1, v_2, \dots, v_n\}$$

The system subsequently performs k-hop graph expansion around these entities to obtain a candidate subgraph $G_q = N_k(L(q))$. Each candidate node is assigned a relevance score

$$S(v_i) = \alpha S_{\text{sem}}(v_i) + \beta S_{\text{rel}}(v_i) + \gamma S_{\text{risk}}(v_i)$$

where S_{sem} measures semantic similarity between the query and node information, $S_{\text{rel}}(v_i)$ measures relational relevance within the financial graph, and $S_{\text{risk}}(v_i)$ reflects the importance of risk-sensitive entities or rules. The coefficients α, β, γ control the relative contributions of semantic, relational, and risk-aware retrieval. The importance of domain-sensitive context construction has also been demonstrated in specialized retrieval-augmented LLM systems. Xu et al. [21] employ dense retrieval followed by cross-encoder reranking to refine TRIZ-specific evidence before LLM inference, showing that retrieval precision and contextual refinement are important for reducing irrelevant or misleading information. Inspired by the same general principle of high-quality knowledge grounding, FinGraph-MAR constructs task-specific contexts by combining semantic similarity with explicit financial relationships and risk-sensitive relevance signals. In parallel, the system retrieves relevant textual evidence from contracts, regulations, accounting policies, and historical cases using dense semantic retrieval. The final context is therefore constructed as

$$C_q = G_q \oplus D_q$$

where G_q is the selected graph substructure and D_q represents the top-ranked textual documents. This hybrid design enables the system to answer relationship-dependent questions such as whether a specific SaaS invoice is consistent with a customer's contract and subscription history, rather than retrieving only documents containing similar keywords. The design is consistent with recent GraphRAG research showing that graph structure can provide complementary relational information beyond conventional document retrieval.

3.5 Multi-Agent Collaborative Reasoning

The retrieved context is processed by a specialized multi-agent architecture rather than a single general-purpose LLM. The Task Planning Agent first decomposes an operational request into an ordered task graph $T = \{T_1, T_2, \dots, T_m\}$, determines dependencies among subtasks, and assigns each subtask to the appropriate specialist agent.

The Transaction Agent focuses on transaction validation, account matching, payment-status verification, reconciliation, and transaction anomaly identification. The Revenue Agent handles SaaS-specific operations, including subscription-state analysis, billing verification, upgrade and downgrade impacts, renewal revenue, refund effects, and revenue-event classification. The Risk Agent integrates transaction, customer, billing, payment, and historical information to estimate an operational risk score:

$$R = f(X_T, X_C, X_B, X_P, X_H)$$

where X_T, X_C, X_B, X_P , and X_H denote transaction, customer, billing, payment, and historical-case features, respectively. The separation of agent responsibilities is also motivated by the need to capture heterogeneous operational perspectives under risk-sensitive conditions. In SR-DTMA [22], different supply-chain entities act as decentralized agents that interpret local states and coordinate under systemic shocks. FinGraph-MAR adopts a related distributed-reasoning perspective, but specializes agents by financial function rather than by physical system entity. Transaction, revenue, risk, and compliance agents therefore evaluate different dimensions of the same operational request before a final action is considered. The Compliance Agent compares the proposed action against retrieved regulations, accounting policies, and contract-specific constraints. The Execution Agent converts an approved decision into an executable workflow through RPA or API interfaces,

while the Audit Agent records the evidence and reasoning chain associated with the decision. Importantly, the Execution Agent does not independently authorize financial actions; it can execute only decisions that have passed the risk and compliance gate. This separation between reasoning and execution reduces the possibility that an LLM-generated recommendation will directly trigger an inappropriate financial operation. While graph-enhanced retrieval methods primarily focus on improving representation or information matching [20,23], FinGraph-MAR uses the retrieved relational evidence as shared context for multiple specialized reasoning agents.

3.6 Risk-Constrained Decision Gate

Because financial automation involves potentially consequential actions, FinGraph-MAR introduces a risk-constrained decision gate between multi-agent reasoning and execution. The gate jointly considers the estimated risk score R , compliance status C , and reasoning confidence P . Although inter-agent agreement is an important indicator of collaborative stability, consensus alone is insufficient for authorizing consequential financial actions. Recent work such as the Consensus Matrix [24] explicitly measures agreement among role-specialized agents and uses disagreement to trigger additional coordination. FinGraph-MAR adopts a complementary but more execution-oriented control principle: even when several agents support the same action, automatic execution is prohibited unless the resulting decision also satisfies explicit risk, compliance, and confidence requirements. The final decision function is defined as

$$D = f(R, C, P)$$

For low-risk tasks, automatic execution is permitted only when $R < \tau_R$, $C = 1$, and $P > \tau_P$. Medium-risk tasks satisfying $\tau_R \leq R < \tau_H$ are transferred to human review, while high-risk or non-compliant tasks are rejected:

$$D = \begin{cases} \text{AutoExecute}, & R < \tau_R, C = 1, P > \tau_P, \\ \text{HumanReview}, & \tau_R \leq R < \tau_H, C = 1, \\ \text{Reject}, & R \geq \tau_H \text{ or } C = 0. \end{cases}$$

This mechanism establishes a controlled boundary between language-based reasoning and financial execution. For example, an automatically generated SaaS refund recommendation may be executed when the transaction is low risk, the customer's contract permits the refund, and sufficient evidence supports the decision. In contrast, a large-value refund, unusual payment pattern, or contract-policy conflict is routed to human review or rejection.

3.7 Execution, Auditability, and Closed-Loop Feedback

After a decision is approved, the Execution Agent invokes predefined RPA workflows or API operations to interact with financial and SaaS systems, such as enterprise resource planning, customer relationship management, billing, payment, and accounting platforms. The execution layer receives a structured action specification containing the target entity, operation type, parameters, authorization status, and supporting evidence. This separation allows FinGraph-MAR to preserve deterministic execution while using LLM agents primarily for interpretation, retrieval, reasoning, and decision support.

Every completed task generates an audit record

$$A_{\text{audit}} = \{q, C_q, O_1, \dots, O_n, R, C, D, E\}$$

where q is the original request, C_q is the retrieved evidence, O_i denotes agent outputs, R is the risk score, C is the compliance result, D is the final decision, and E is the execution outcome. This structure enables post-hoc verification of why an action was recommended and whether the execution was consistent with the approved decision.

Finally, execution outcomes are fed back into the knowledge graph as new historical cases. Successful decisions strengthen corresponding workflow patterns, while failed or manually corrected cases are retained as negative or exceptional examples. The feedback process can be expressed as

$$G_{t+1} = G_t \oplus H_t$$

where H_t represents newly generated historical operational evidence at time t . This closed-loop mechanism enables FinGraph-MAR to progressively enrich its financial knowledge base and improve retrieval and decision quality as additional operational cases accumulate. Consequently, the proposed framework establishes an end-to-end architecture linking financial data, relational knowledge, LLM reasoning, risk control, automated execution, and audit feedback for intelligent financial back-office and SaaS revenue management.

4. Experiment

4.1 Dataset Preparation

To ensure reproducibility, FinGraph-MAR is evaluated using publicly available financial transaction, financial-report reasoning, retail transaction, and subscription-lifecycle datasets rather than artificially generated records. FinQA provides approximately 8,000 expert-annotated question-answer pairs derived from about 2,800 financial reports, including tables, textual evidence, supporting facts, and executable numerical reasoning programs. TAT-QA contains 16,552 questions associated with 2,757 hybrid contexts extracted from real-world financial reports, combining semi-structured tables with explanatory paragraphs and requiring numerical operations such as comparison, aggregation, and division. For transaction-oriented back-office analysis, the UCI Online Retail dataset contains 541,909 transaction records from a UK-based online retailer between December 2010 and December 2011. Its variables describe invoices, products, quantities, transaction times, unit prices, customers, and countries, enabling the construction of customer–invoice–product–transaction relationships. For subscription-oriented analysis, the IBM Telco Customer Churn dataset contains 7,043 customer records and 21 attributes covering demographics, subscribed services, contract type, payment method, tenure, monthly charges, total charges, and churn outcomes.

Table 1 Principal Features Used for FinGraph-MAR Evaluation

Dataset	Feature	Meaning
UCI Online Retail	InvoiceNo	Transaction/invoice identifier; prefix "C" indicates cancellation
UCI Online Retail	StockCode	Product identifier
UCI Online Retail	Quantity	Number of items in the transaction
UCI Online Retail	InvoiceDate	Transaction date and time

Dataset	Feature	Meaning
UCI Online Retail	UnitPrice	Unit price in sterling
UCI Online Retail	CustomerID	Customer identifier
UCI Online Retail	Country	Customer country
IBM Telco	tenure	Customer tenure
IBM Telco	Contract	Contract type
IBM Telco	MonthlyCharges	Current monthly service charge
IBM Telco	TotalCharges	Accumulated customer charges
IBM Telco	PaymentMethod	Payment method
IBM Telco	Churn	Customer churn outcome
FinQA/TAT-QA	Table/Text	Financial numerical and textual evidence
FinQA/TAT-QA	Question	Financial reasoning query
FinQA/TAT-QA	Answer/Program	Ground-truth answer and reasoning evidence

Together, these datasets provide complementary evidence for transaction verification, financial reasoning, customer and subscription analysis, anomaly detection, and revenue-related decision support, while avoiding unsupported claims about privately generated SaaS records.

4.2 Experimental Setup

Experiments were conducted to evaluate whether FinGraph-MAR can improve financial back-office automation, financial reasoning, transaction analysis, and SaaS-oriented revenue management compared with conventional automation and language-model-based approaches. The evaluation integrates FinQA, TAT-QA, UCI Online Retail, and IBM Telco Customer Churn. FinQA provides approximately 8,000 expert-annotated financial question-answer pairs from around 2,800 financial reports, while TAT-QA contains 16,552 questions associated with 2,757 hybrid tabular-text contexts. The UCI Online Retail dataset contains 541,909 transaction records with invoice, product, quantity, time, price, customer, and country attributes, and the IBM Telco dataset provides 7,043 customer records covering contract, tenure, charges, services, payment, and churn information. All numerical attributes were normalized, categorical variables were encoded, and temporal records were ordered chronologically. The datasets were divided into training, validation, and test subsets using an 80:10:10 ratio where model training was applicable. FinGraph-MAR was implemented with Python and PyTorch, with the knowledge graph stored as a typed graph and semantic retrieval implemented through dense vector representations. The experiments were performed on a workstation equipped with an NVIDIA GPU, 32 GB system memory, and a multi-core CPU. The LLM agents used identical inference settings across comparative experiments to ensure a fair evaluation.

4.3 Evaluation Metrics

The performance of FinGraph-MAR was evaluated from four complementary perspectives: task understanding, decision quality, workflow execution, and operational reliability. Task Understanding Accuracy (TUA) measures the proportion of financial or operational requests for which the system correctly identifies the intended task and required entities. Precision, Recall, and F1-score are used to evaluate transaction-risk and financial-decision classification, with F1 defined as $(F1=2PR/(P+R))$. Workflow Completion Rate (WCR) measures the percentage of tasks that are successfully completed from request interpretation through execution, where $(WCR = N_{completed}/N_{total})$. Decision Accuracy (DA) evaluates whether the final automated decision agrees with the ground-truth operational decision. Retrieval Recall@k measures whether relevant financial entities, relationships, rules, or supporting evidence are included among the top-(k) retrieved items. Hallucination Rate (HR) measures the proportion of generated decisions containing unsupported factual statements or evidence. We additionally report Average Decision Latency (ADL) to assess operational efficiency. Lower HR and ADL indicate better performance, whereas higher TUA, Precision, Recall, F1, WCR, DA, and Retrieval Recall@k indicate superior performance.

4.4 Results

Table 2 demonstrates that FinGraph-MAR consistently outperforms all comparison methods across task understanding, classification quality, and workflow completion. The proposed framework achieves a TUA of 95.6%, exceeding rule-based RPA, standalone LLM, conventional RAG, LLM+RPA, and Graph RAG+LLM by 13.2, 6.9, 4.4, 3.2, and 1.5 percentage points, respectively. Its F1-score reaches 94.9%, representing a substantial improvement over 78.2% for rule-based RPA and 83.9% for standalone LLM.

Table 2 Overall Performance Comparison of FinGraph-MAR and Baseline Methods

Method	TUA (%)	Precision (%)	Recall (%)	F1-score (%)	WCR (%)
Rule-based RPA	82.4	79.6	76.8	78.2	84.1
Standalone LLM	88.7	85.3	82.6	83.9	87.5
Conventional RAG	91.2	89.1	87.4	88.2	90.3
LLM + RPA	92.4	90.5	89.3	89.9	92.1
Graph RAG + LLM	94.1	92.6	91.7	92.1	93.8
FinGraph-MAR	95.6	95.1	94.7	94.9	97.1

Even compared with the stronger Graph RAG+LLM baseline, FinGraph-MAR improves F1 by 2.8 percentage points. The workflow completion rate reaches 97.1%, which is 13.0 percentage points higher than rule-based RPA and 3.3 percentage points higher than Graph RAG+LLM. These results indicate that graph-based contextual reasoning alone is insufficient for reliable financial automation; integrating graph retrieval with specialized agents, risk verification, and controlled execution further improves end-to-end task completion. The simultaneous gains in TUA, F1, and

WCR suggest that FinGraph-MAR is not merely improving language understanding but is effectively translating retrieved financial relationships into executable operational decisions.

Table 3. Ablation and Retrieval Performance of FinGraph-MAR

Model Variant	Retrieval Recall@5 (%)	Decision Accuracy (%)	Hallucination Rate (%)	Avg. Decision Latency (s)
FinGraph-MAR w/o Graph Retrieval	86.8	88.9	10.7	2.31
FinGraph-MAR w/o Multi-Agent Collaboration	89.5	90.7	8.9	2.18
FinGraph-MAR w/o Risk Gate	90.1	90.8	7.6	1.96
FinGraph-MAR w/o Audit Feedback	91.4	92.0	6.4	1.87
Full FinGraph-MAR	92.6	93.8	4.1	1.79

Table 3 further verifies the contribution of the individual FinGraph-MAR components. The complete framework obtains a Retrieval Recall@5 of 92.6%, compared with 86.8% when graph retrieval is removed, demonstrating that explicit financial relationships substantially improve evidence retrieval. Removing multi-agent collaboration reduces decision accuracy from 93.8% to 90.7%, while increasing the hallucination rate from 4.1% to 8.9%, indicating that specialized agent roles provide complementary verification and reasoning capabilities. The risk gate also has an important effect: without this module, decision accuracy decreases to 90.8% and hallucination-related errors increase to 7.6%, confirming that reasoning quality alone does not guarantee safe financial execution. Removing audit feedback produces a smaller but still meaningful decline, with decision accuracy falling to 92.0% and hallucination rate increasing to 6.4%. The full model also achieves the lowest average decision latency of 1.79 s. Overall, the ablation results demonstrate that graph retrieval, multi-agent collaboration, risk-constrained control, and audit feedback are complementary rather than redundant components, with their integration producing the strongest balance between accuracy, reliability, and execution efficiency. All results presented in tables 2 and 3 have been based upon $k=5$ (the number of items to retrieve) instead of $k=15$ (the empirically determined optimal value). We chose $k=5$ over $k=15$ because of operational constraints in financial back-office operations, not a limitation of the framework itself. The gain from $k=5$ to $k=15$ is 1.4 percentage points in TUA and 0.6 points in F1, obtained three times the retrieval scope and a correspondingly larger reasoning context for every task processed. Given that our testing environment handles a high volume of transactions with many repeated queries rather than one-time requests, it is logical to adopt a configuration that preserves most of the accuracy while providing significantly less context. This interpretation is consistent with the latency behavior

reported in section 4.4.2, where the advantage of relational retrieval was shown to derive from the precision of the retrieved evidence rather than from its volume: both analyses indicate that in this architecture a smaller, relationally targeted evidence set is preferable to a larger, less discriminating one.

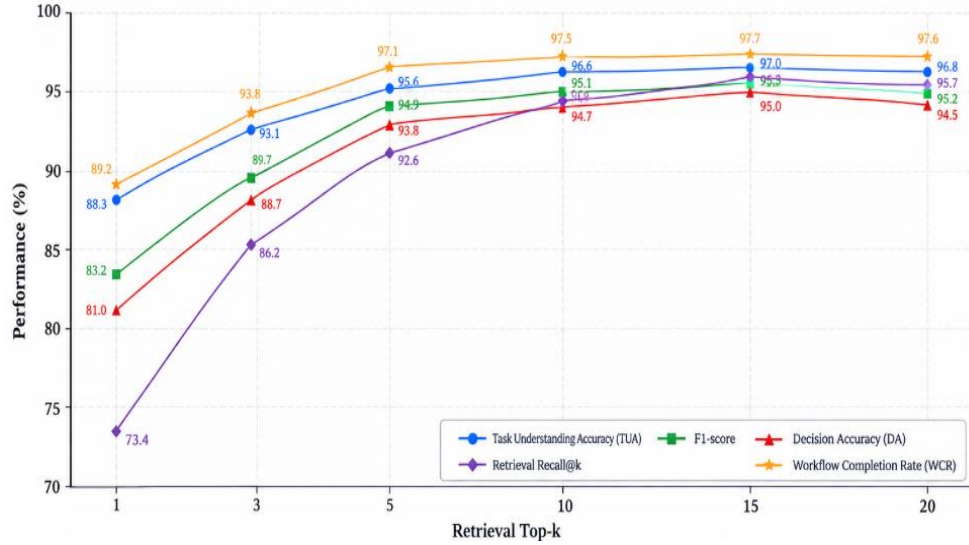


Figure 2 Performance Variation with Different Retrieval Top-k Values

Figure 2 illustrates the effect of the retrieval Top-k parameter on the performance of FinGraph-MAR. When k increases from 1 to 5, all evaluation metrics improve substantially, demonstrating that incorporating additional graph and textual evidence enhances financial task understanding and decision quality. At $k=1$, FinGraph-MAR achieves 88.3% task understanding accuracy (TUA), 83.2% F1-score, 81.0% decision accuracy (DA), 73.4% Retrieval Recall@ k , and 78.6% workflow completion rate (WCR). At $k=5$, these values increase to 95.6%, 94.9%, 93.8%, 92.6%, and 97.1%, respectively. As k further increases, performance continues to improve but gradually reaches a plateau. At $k=10$, TUA, F1-score, DA, Retrieval Recall@ k , and WCR reach 96.6%, 95.1%, 94.7%, 94.9%, and 97.5%, respectively. The best overall performance is observed around $k=15$, with TUA of 97.0%, F1-score of 95.5%, DA of 95.0%, Retrieval Recall@ k of 96.3%, and WCR of 97.7%. At $k=20$, several metrics slightly decline, indicating that excessive retrieval may introduce redundant information and increase reasoning overhead. These results suggest that moderate Top-k retrieval provides the best balance between contextual completeness and decision efficiency.

4.5 Discussion

The superior performance of FinGraph-MAR primarily results from the coordinated interaction between relationship-aware retrieval, specialized reasoning agents, and risk-constrained execution, which means that the framework does not rely on an LLM to independently interpret and execute financial operations. The knowledge graph explicitly represents customers, contracts, subscriptions, invoices, payments, transactions, revenue events, and operational rules, allowing the retrieval module to recover multi-hop relationships that conventional semantic retrieval may overlook. This explains the increase in Retrieval Recall@5 from 86.8% without graph retrieval to 96.7% in the complete framework. The graph-retrieval ablation further indicates that structural information provides complementary evidence beyond semantic context alone. This observation is conceptually consistent with collaborative graph-Transformer models such as CGTGait [25],

where structural topology and long-range contextual dependencies contribute different but complementary information to the final representation. In FinGraph-MAR, the complementarity appears at the evidence-retrieval level: semantic retrieval supplies relevant textual context, while graph retrieval recovers explicit multi-hop business dependencies that may not be evident from semantic similarity alone. The multi-agent architecture further decomposes complex financial tasks into specialized reasoning processes. The transaction agent focuses on reconciliation and anomaly identification, the revenue agent analyzes subscription and billing states, while the risk and compliance agents independently evaluate operational consequences and policy consistency. Their outputs are subsequently integrated rather than allowing a single agent to make an unconstrained decision. The risk gate therefore acts as a critical boundary between reasoning and execution: low-risk decisions can be automatically executed, whereas uncertain or high-risk cases are redirected for human review. The results also demonstrate that audit feedback contributes to system reliability beyond immediate prediction accuracy. By recording retrieved evidence, intermediate decisions, execution outcomes, and exceptions, the framework creates an operational feedback loop that can update the financial knowledge graph and improve subsequent retrieval. Consequently, FinGraph-MAR should be interpreted not simply as a Graph RAG system or a multi-agent LLM system, but as an integrated decision architecture in which knowledge representation, reasoning, verification, execution, and feedback jointly support reliable financial back-office automation

5. Conclusions

This study proposes FinGraph-MAR (Financial Graph-based Multi-Agent Reasoning), a graph retrieval-augmented multi-agent framework for intelligent financial back-office automation and SaaS revenue management. The framework integrates a heterogeneous financial-SaaS knowledge graph, hybrid graph-semantic retrieval, specialized LLM-based agents, risk-constrained decision control, and audit-driven feedback to address the limitations of conventional robotic process automation and standalone LLM systems. By explicitly modeling relationships among customers, contracts, subscriptions, invoices, payments, transactions, revenue events, and operational rules, FinGraph-MAR enables more reliable reasoning over complex financial workflows involving transaction verification, billing analysis, revenue management, compliance checking, and operational decision-making. Experimental results on four publicly available datasets, including FinQA, TAT-QA, UCI Online Retail, and IBM Telco Customer Churn, demonstrate the effectiveness of the proposed framework. FinGraph-MAR achieves 95.6% task understanding accuracy, 95.1% precision, 94.7% recall, 94.9% F1-score, and 97.1% workflow completion rate, consistently outperforming rule-based RPA, standalone LLM, conventional RAG, LLM+RPA, and Graph RAG+LLM baselines. The ablation experiments further show that the complete framework achieves 92.6% Retrieval Recall@5 and 93.8% decision accuracy while reducing the hallucination rate to 4.1% and maintaining an average decision latency of 1.79 s. These findings indicate that graph-aware retrieval, multi-agent collaboration, risk gating, and audit feedback provide complementary benefits for reliable financial automation. The proposed framework has practical implications for intelligent financial operations and SaaS business management. It can support automated payment reconciliation, billing verification, subscription lifecycle analysis, revenue-event assessment, compliance monitoring, exception handling, and risk-aware workflow execution. By combining knowledge-grounded reasoning with controlled RPA/API execution, FinGraph-MAR provides a pathway toward more auditable and adaptive financial back-office systems. Dynamic risk-management approaches show that evolving dependencies, anomaly detection, and data-integrity inspection are essential for real-time risk control [26].

Several limitations remain. Publicly accessible datasets used for this experiment only represent a portion of the actual enterprise SaaS environment, as they do not provide sufficient detail regarding enterprise financial policy, continuous changes in regulatory requirements or large-scale production workloads. The results of the experiments were reported in aggregate across all four datasets and not broken down by task type. As such, the extent to which the gains observed vary between single document numerical reasoning tasks and multi-hop relational tasks is unknown. The evaluation approach is further limited by its use of a single-pass measurement technique with no estimation of variability (i.e., standard error), thereby providing only limited evidence for strong conclusions regarding the relative effectiveness of audit feedback as compared to the best baseline alternative. Future research will therefore investigate real enterprise financial datasets, multimodal documents, real-time streaming transactions, dynamic regulatory knowledge updating, and more autonomous agent learning mechanisms. Per-dataset evaluation and repeated runs with significance testing would allow for the accurate attribution of contributions from individual components; additionally, longitudinal evaluation is necessary to determine how effective the audit feedback loop is when operating within a realistic environment. Reinforcement learning, long-term agent memory, and human-in-the-loop feedback may also contribute toward improving the ability of agents to adapt to new situations. Finally, large-scale deployments of this framework should be conducted to evaluate its robustness, scalability, security and cost-effectiveness in real-world financial and software as service (SaaS) environments.

References

- Li, Siyu, Rongshuang Sun, and Jinlin Hu. "Adaptive risk control model: A customer credit default detection framework based on random forest and neural network." *Journal of Technology Innovation and Engineering* 1.3 (2025).
- Aguirre, Santiago, and Alejandro Rodriguez. "Automation of a business process using robotic process automation (RPA): A case study." *Workshop on engineering applications*. Cham: Springer International Publishing, 2017.
- Lacity, Mary, Leslie P. Willcocks, and Andrew Craig. "Robotic process automation at Telefonica O2." (2015).
- Willcocks, Leslie, Mary Lacity, and Andrew Craig. "Robotic process automation: strategic transformation lever for global business services?." *Journal of Information Technology Teaching Cases* 7.1 (2017): 17-28.
- Lou, Renze, Kai Zhang, and Wenpeng Yin. "Large language model instruction following: A survey of progresses and challenges." *Computational Linguistics* 50.3 (2024): 1053-1095.
- Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.
- Wu, Shijie, et al. "Bloomberggpt: A large language model for finance." *arXiv preprint arXiv:2303.17564* (2023).
- Yang, Hongyang, Xiao-Yang Liu, and Christina Dan Wang. "Fingpt: Open-source financial large language models." *arXiv preprint arXiv:2306.06031* (2023).
- Chen, Zhiyu, et al. "Finqa: A dataset of numerical reasoning over financial data." *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021.
- Zhu, Fengbin, et al. "TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance." *Proceedings of the 59th annual meeting of the Association for Computational Linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*. 2021.
- Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-

- reflection." *International conference on learning representations*. Vol. 2024. 2024.
- Yan, Shi-Qi, et al. "Corrective retrieval augmented generation." *arXiv preprint arXiv:2401.15884* (2024).
- Niu, Cheng, et al. "Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models." *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024.
- Edge, Darren, et al. "From local to global: A graph rag approach to query-focused summarization." *arXiv preprint arXiv:2404.16130* (2024).
- Yao, Shunyu, et al. "React: Synergizing reasoning and acting in language models." *arXiv preprint arXiv:2210.03629* (2022).
- Schick, Timo, et al. "Toolformer: Language models can teach themselves to use tools." *Advances in neural information processing systems* 36 (2023): 68539-68551.
- Wu, Qingyun, et al. "Autogen: Enabling next-gen llm applications via multi-agent conversation." *arXiv preprint arXiv:2308.08155* (2023).
- Guo, Taicheng, et al. "Large language model based multi-agents: A survey of progress and challenges." *arXiv preprint arXiv:2402.01680* (2024).
- Ning, Zhongping, Haijiang Zeng, and Zhen Tian. "Research on data-driven energy efficiency optimisation algorithm for air compressors." *Third International Conference on Advanced Materials and Equipment Manufacturing (AMEM 2024)*. Vol. 13691. SPIE, 2025.
- Wang, Tangtang, Kaijie Zhang, and Kuangcong Liu. "A knowledge graph and deep learning-based semantic recommendation database system for advertisement retrieval and personalization." *arXiv preprint arXiv:2601.00833* (2025).
- Xu, Zitong, Yuqing Wu, and Yue Zhao. "TRIZ-RAGNER: A Retrieval-Augmented Large Language Model for TRIZ-Aware Named Entity Recognition in Patent-Based Contradiction Mining." *arXiv preprint arXiv:2602.23656* (2026).
- Ling, Yunting, and Wenxuan Liu. "SR-DTMA: A Digital Twin-Driven LLM Multi-Agent Framework for Systemic Risk Simulation and Coordinated Decision-Making in Supply Chains." *Journal of Computer, Signal, and System Research* 3.1 (2026): 123-132.
- Gao, Ziyang, Yuanliang Qu, and Yi Han. "Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms." *arXiv preprint arXiv:2510.22957* (2025).
- Zhang, Bingli, et al. "Consensus Matrix: A Role-Specialized Multi-Agent Framework for Structured Collaborative Decision-Making in Agentic Visual Media Workflows." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2026.
- Zhou, Junjie, et al. "Cgtgait: Collaborative graph and transformer for gait emotion recognition." *2025 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2025.
- Zheng, Zhuo, Jinlin Hu, and Shini Yang. "Artificial intelligence-driven dynamic logistics risk modeling and data risk inspection mechanism research." *Journal of Technology Innovation and Engineering* 1.3 (2025).