Abstract

This dissertation explores the development and evaluation of a financial agent system, designed to automate financial statement analysis using a modular, multi-agent architecture, powered by Large Language Models (LLMs). Motivated by the inefficiencies in traditional financial workflows, often fragmented, manual, and time-intensive, the project introduces ***PEER (Plan, Execute, Express, Report)***, an orchestration framework that decomposes complex analysis tasks into specialised agents. The system integrates prompt engineering, sandboxed code execution, and automated report generation to enhance both numeric accuracy and decision-making support.

Performance is assessed across synthetic but realistic corporate finance scenarios using a custom-built LLM-based referee. To measure the effect of system architecture, three configurations were created: ***a baseline model***, which prompts a single GPT-4-class agent to generate full reports without tools or sub-agent delegation; ***the PEER system***, which decomposes tasks across planning, execution, and report generation agents with code execution capabilities; and a ***PEER_ablation variant***, which retains the agent-based structure but disables the code execution, thereby testing the incremental value of tool-augmented reasoning. All three models received identical input scenarios and produced self-contained reports under identical operating conditions, ensuring a fair and deterministic comparison. Results demonstrate that while ***the baseline*** model frequently outperforms ***PEER*** in numeric fidelity and coherence, ***the ablated variant of PEER*** often achieves higher quantitative accuracy. These findings reveal coordination trade-offs in multi-agent design, suggesting that orchestration quality is as vital as individual agent capabilities. The dissertation offers both a technical contribution to the field of AI-enabled finance and strategic insight into how agent-based systems can be responsibly deployed to support analytical decision-making in business environments.

Acknowledgments

I would like to express my deep gratitude to *UCL School of Management* and *Deepflow* for providing the opportunity to explore such a timely and impactful domain. Their joint support made it possible to engage with cutting-edge questions at the intersection of finance and artificial intelligence.

I am especially thankful to my dissertation supervisor, *Assoc. Prof. Bart Vanneste*, for his continuous support, critical feedback, and the insightful brainstorming sessions that shaped this dissertation.

I also wish to thank *Bart Kultys* and *Jiangbo Shangguan* from *Deepflow* for their technical assistance and for generously providing the infrastructure and operational support needed to execute this project.

Finally, I am profoundly grateful to the *Foreign, Commonwealth & Development Office (FCDO)* for funding my studies through the *Chevening Scholarship*. This once in a lifetime opportunity to study at one of the world's leading universities has been transformative both academically and personally.

Table of Contents

Abstract **3**

Acknowledgments **4**

Table of Contents **5**

List of Figures **7**

1. Introduction **8**

1.1 Background 8

1.2 Problem Statement 9

1.3 Company overview 10

2. Literature Review **11**

2.1 Generative AI and Autonomous Agents in Finance 11

2.2 AI Agent Frameworks 12

3. Methodology **15**

3.1 Research Design 15

3.2 PEER Architecture and Implementation 17

3.2.1 Core LLM Engine 18

3.2.2 Multi-Agent Architecture 18

Plan Agent 18

Execute Agent 18

Express Agent 18

3.2.3 Specialized Sub-Agent Architecture 19

Extractor Sub-Agent 19

Analyst Sub-Agent 19

Miscellaneous Sub-Agent 19

3.2.4 Workflow Orchestration and Planning Mechanism 19

3.2.5 Tool and API Integration 20

E2B Code Interpreter Integration 20

Data Processing Infrastructure 20

Visualization Capabilities 20

3.2.6 User Interaction Design 20

3.2.7 State Management and Data Flow 20

3.3 Prompt Engineering and Template Architecture 21

3.3.1 Structured Prompt System 21

3.3.2 Agent-Specific Prompt Templates 21

Planning Agent Prompts 21

Task Routing Prompts 22

Data Extraction Prompts 22

Financial Analysis Prompts 22

Report Generation Prompts 22

3.3.3 Dynamic Context Injection 23
3.3.4 Error Handling and Recovery 23
3.4 Task Scenarios 23
3.5 Evaluation Metrics and LLM-Judge Protocol 24
3.5.1 Operating environment 25
3.5.2 Criterion 1 - Numeric Accuracy 25
3.5.3 Criterion 2 - Decision-Making Support 26
4. Results 27
4.1 Performance Evaluation 27
4.1.1 Balance Sheet Analysis 27
4.1.2 Performance Benchmarking 29
4.1.3 Ratio Analysis 31
4.1.4 Trend Analysis 33
5. Discussion 35
5.1 Interpretation of Results 35
5.2 Limitations 37
5.3 Reflections on GenAI Use and Ethics 39
5.4 Business and Strategic Implications 40
6. Conclusion 42
6.1 Summary of Findings 42
References 43
Appendices 46
A Project Management 46
B Repository 50
C Evaluator Details 50
D Data Description (All Scenarios) 50

List of Figures

Figure 1: Cyclic Workflow Diagram of the PEER Framework. The user’s query, “Why did Buffet sell BYD stock?” prompts the “Plan” agent to generate four relevant sub-questions. The “Execute” agent then collects information, including BYD’s financial data and expert opinions. The “Express” agent synthesizes a comprehensive answer, which the “Review” agent evaluates and, if necessary, suggests modifications. 15
Figure 2: The PEER Architecture. It illustrates the directional, multi-agent workflow of PEER (Plan, Execute, Express, Report). 17
Figure 3: State Management Schema of PEER. 21
Figure 4: Introducing an error traceback parameter for the agent to iterate over errors in code generation and execution. 23
Figure 5: Summary of Accuracy evaluation across scenarios 35
Figure 6: Summary of Decision making evaluation across scenarios 35

List of Tables

Table 1: Comparative Analysis of Three Financial Agent Architectures 16
Table 2: Task Scenario Table 24
Table 3: Evaluation of three generated reports for financial task – Balance Sheet Analysis 27
Table 4: Evaluation of three generated reports for financial task – Performance Benchmarking 29
Table 5: Evaluation of three generated reports for financial task – Ratio Analysis 31
Table 6: Evaluation of three generated reports for financial task – Trend Analysis 33
Table 7: Data description across all scenarios 51

1. Introduction

1.1 Background

The domains of finance and investment management are inherently complex, characterised by uncertainty and rapid evolution. Historically, these sectors have seen a growing integration of advanced computational models. A significant advancement in this landscape has been the advent of Large Language Models (LLMs), such as the generative pre-trained transformer (GPT) series and BERT, which have demonstrated remarkable capabilities in understanding context, processing vast amounts of data, and generating human-like text. Their integration into financial practices is rapidly accelerating, catalysing a surge of research and applications designed to transform traditional methodologies and drive innovation within the industry.

LLMs offer several key advantages crucial for financial reasoning and decision-making. They can significantly enhance data analysis by processing vast financial information to identify intricate patterns and trends. Furthermore, LLMs are capable of predictive modelling, allowing them to forecast market conditions and asset performance, which in turn leads to more robust investment recommendations. Their capabilities extend to providing personalised advisory services, where they can analyse an individual's or organisation's financial situation, goals, and risk tolerance to offer customised advice. The integration of LLMs also facilitates real-time monitoring and alerts on market trends and news, enabling timely adjustments to strategies. Moreover, by embedding LLMs into user-friendly interfaces like chatbots, they can improve accessibility and engagement for financial planning and advisory tasks. These models are being applied across various financial tasks, including linguistic tasks, sentiment analysis, financial time series analysis, financial reasoning, and a burgeoning area: agent-based modelling (Kong et al., 2024).

The emergence of autonomous finance agents represents a pivotal development, largely driven by advancements in Agent-Based Modelling (ABM). ABM is a powerful methodology for simulating complex systems, distinguished from traditional models by its capacity to capture the diversity of behaviours and adaptive strategies found in real-world financial markets, rather than assuming uniform behaviour or equilibrium states. The core principle involves creating autonomous agents that interact within a defined environment, enabling complex phenomena to emerge from the bottom up (Kong et al., 2024).

The integration of LLMs with ABM significantly enhances the cognitive functions of these agents. LLMs empower agents to interpret and react to extensive amounts of unstructured data, such as financial news, reports, and social media posts, leading to more realistic and adaptive simulations. This synergy is crucial for developing robust strategies and enhances predictive power.

1.2 Problem Statement

Current financial analytical workflows like reporting, earnings analysis, and due diligence, are inefficient due to their manual coordination across disparate systems and human intervention points. This leads to delays, inconsistencies, and higher operational costs. The increasing complexity and volume of financial data exacerbate these challenges, necessitating better solutions to maintain competitive advantage and ensure robust decision-making

Against this backdrop, recent advances in agentic AI systems offer a timely opportunity to re-engineer financial analysis workflows. *This dissertation designs and evaluates a multi-agent architecture that automates and integrates key elements of financial statement analysis. Specifically, it examines how such architectures can be structured to deliver accurate, context-aware outputs that improve efficiency, reduce human error and support faster, more consistent strategic decision-making. The investigation focuses on the design principles that enable AI agents to participate effectively in financial workflows, with particular emphasis on ratio analysis, peer benchmarking and the generation of narrative financial reports.*

The recent investigation by Kim, Muhn, and Nikolaev in 2024 (Kim et al., n.d.) provides compelling evidence and direct validation for this problem statement's premise regarding the transformative potential of AI. Their research specifically explores the capability of a Large Language Model (LLM), GPT-4 Turbo, to perform financial statement analysis comparable to that of professional human analysts. This study demonstrates several key achievements that directly address the inefficiencies and inaccuracies highlighted in current financial workflows:

- *Outperformance in Earnings Prediction*: The LLM outperforms professional human analysts in predicting earnings changes, particularly in challenging scenarios. This directly tackles the problem of human-induced inconsistencies and delays in forecasting.

- *Matching State-of-the-Art ML Models*: GPT-4 Turbo matches the performance of specialized state-of-the-art machine learning models in financial statement analysis. This underscores the LLM's capacity to deliver the high accuracy required for automated and reliable financial analysis.

- *Enhanced Strategic Outcomes*: Trading strategies based on the LLM’s predictions demonstrated higher Sharpe ratios and alphas compared to those based on other models. This validates the potential of AI agents to accelerate and improve strategic decision-making, moving beyond mere automation to demonstrable value creation.

- *Robustness and Bias Mitigation*: The study meticulously ensured its findings were robust by using standardized and anonymised financial statements without dates or specific company identities. This approach ensured that the LLM's predictions were derived from generative narrative insights about future performance, rather than its training memory or look-ahead bias, addressing crucial concerns about data reliability and ethical deployment of AI in finance.

While the work of Kim, Muhn, and Nikolaev (2024) demonstrates the immense potential of large language models for financial analysis, building robust systems to harness this power presents significant engineering challenges. This is especially true for Multi-Agent LLM Systems (MAS), where high failure rates often stem from flawed *organizational design and agent coordination* rather than the limitations of the underlying models. A foundational study introduced the *MAST (Multi-Agent System Failure Taxonomy)*, which empirically identified three major categories of these systemic failures: **specification issues, inter-agent misalignment, and task verification** (Cemri et al., 2025).

1.3 Company overview

Deepflow, a London based startup, is a pioneering AI orchestration platform focused on human - AI orchestration. The company provides an orchestration layer that synchronizes tasks, teams, models, and agents, enabling organisations to scale their collective output without proportionally scaling headcount. Deepflow’s platform empowers teams to delegate and coordinate work across humans and AI, allowing each to contribute where their strengths lie. (Deepflow.com, 2025)

2. Literature Review

2.1 Generative AI and Autonomous Agents in Finance

The evolution of artificial intelligence (AI) in finance reflects a continuous trajectory from basic analytical techniques toward autonomous agent systems. This progression has been driven primarily by the need for greater contextual understanding, efficient data processing, and improved decision-making capabilities.

Initially, AI approaches in finance were based on simpler models, including recurrent neural networks (RNNs) and long short-term memory networks (LSTMs). These earlier models were limited by their architectural constraints, particularly in managing long-term dependencies and efficiently processing extensive textual financial data (Lipton, Berkowitz, and Elkan, 2015; Staudemeyer and Morris, 2019). Consequently, they struggled to adequately maintain context and manage complexity, particularly within extensive financial documents (Zmandar et al., 2021).

A significant turning point was the advent of transformer-based large language models (LLMs), such as GPT models, BERT, and finance-specific variants like FinBERT. These models addressed previous limitations by incorporating advanced self-attention mechanisms, which allowed for better handling of long text sequences and significantly enhanced contextual understanding (Hadi et al., 2023; Raiaan et al., 2024). This advancement enabled LLMs to streamline extensive financial narratives into concise summaries and accurately extract essential insights, greatly improving the efficiency of information processing in financial contexts (Khanna et al., 2022; Shukla et al., 2022; Yepes et al., 2024).

Further, LLMs have been effectively utilized for complex linguistic tasks, including multilingual and domain-specific summarization challenges, enhancing their adaptability to various financial applications globally (Foroutan et al., 2022; Suzuki et al., 2023). For instance, advances in financial text summarization through specialized models like Longformer-Encoder-Decoder (LED) demonstrated significant improvements in the summarization of extensive financial reports (Khanna et al., 2022).

With respect to sentiment analysis, earlier lexicon-based methods, such as those outlined by Stone et al. (1966) and Pennebaker et al. (2001), were effective for basic sentiment classification but were limited by their inability to capture complex linguistic constructs like sarcasm and irony (Loughran & McDonald, 2011). The emergence of LLMs substantially improved sentiment analysis accuracy by leveraging vast pretrained knowledge bases to better interpret financial language nuances and market sentiment (Steinert and Altmann, 2023; Luo & Gong, 2024). Specifically, models such as GPT-4 demonstrated advanced sentiment classification capabilities, significantly enhancing prediction accuracy in financial contexts (Lopez-Lira & Tang, 2023).

In parallel with these developments, AI in finance evolved to encompass agent-based modelling (ABM), integrating the contextual and reasoning abilities of LLMs into autonomous agent systems. ABM represents a significant advancement as it enables realistic simulations of market behaviours and economic activities through sophisticated, adaptive AI agents. This framework significantly enhances the strategic capabilities of financial institutions by simulating real-world dynamics and adaptive decision-making processes (Zhang, Liu et al., 2024).

Thus, the ongoing development of AI in finance continues to be marked by increasing sophistication, from overcoming initial textual processing limitations through transformer-based architectures, to enabling dynamic decision-making and strategic planning capabilities with autonomous agent systems. This trajectory underscores the finance sector's consistent aim to leverage increasingly powerful AI tools to manage and optimize the inherent complexities of modern financial environments effectively.

2.2 AI Agent Frameworks

Recent advancements have integrated large language models (LLMs) with ABM frameworks to significantly enhance agents' cognitive capabilities. These advanced AI agents leverage LLMs to interpret extensive financial data sources, such as market news, financial disclosures, and social media sentiment, enabling a sophisticated level of market interaction simulation. For instance, the StockAgent system demonstrates the potential for AI-driven agents to replicate investor behaviours and assess the impact of external factors, including macroeconomic changes and regulatory shifts, on trading decisions (Zhang, Liu et al., 2024). Such integration of LLMs with agent-based approaches allows for nuanced simulations of market reactions, capturing realistic variations in trading behaviour and strategic decision-making among market participants.

Moreover, AI agent frameworks have increasingly incorporated multimodal data integration to enhance market analysis capabilities. FinAgent illustrates this by combining textual, numerical, and visual data inputs, providing agents with diversified memory retrieval capabilities essential for complex trading decisions in dynamic financial environments (Zhang, Zhao et al., 2024). Additionally, agents equipped with layered memory structures, such as FINMEM, effectively categorize and prioritize financial information based on its temporal and strategic relevance, enabling more agile and informed trading decisions (Yu et al., 2024). These sophisticated memory mechanisms enhance the adaptability of agents, empowering them to refine their strategies continuously in response to evolving market conditions.

AI agent frameworks have also embraced iterative learning and continuous improvement processes. QuantAgent exemplifies this approach through a two-layer iterative loop, combining internal knowledge base refinement with external real-world testing and strategy validation. Such iterative loops facilitate autonomous extraction of financial signals and robust identification of trading opportunities, substantially improving agent effectiveness and market adaptability over time (Wang, Yuan et al., 2024).

An important development within AI agent frameworks involves the integration of human expertise with AI capabilities. Systems such as Alpha-GPT and Alpha-GPT 2.0 emphasize the strategic interaction between human analysts and AI-driven agents, facilitating iterative refinement of investment strategies. This collaboration leverages human intuition and AI analytical capabilities, enhancing the efficiency, creativity, and effectiveness of financial strategy formulation and execution (Yuan, Wang, and Guo, 2024).

FinRobot exemplifies these modern design principles by offering a modular, multi-layered agent architecture optimized for diverse financial tasks. At its core, FinRobot integrates real-time data processing with multi-source LLM orchestration, enabling precise task-agent alignment across forecasting, reporting, and strategy generation. Its agents employ Financial Chain-of-Thought (CoT) prompting to decompose complex financial problems into interpretable, sequential reasoning steps, producing outputs such as detailed analyses, contextual forecasts, and decision-ready insights. The platform's structured agent roles and smart scheduling mechanism further enhance operational coordination, transparency, and adaptability, especially in multi-agent, high-stakes financial environments (Yang et al., 2024).

AI-driven ABM frameworks have also proven beneficial for broader economic simulations. EconAgent leverages advanced LLM mechanisms to realistically simulate complex macroeconomic interactions, employing human-like decision-making processes to provide comprehensive insights into economic dynamics and policy implications (Li, Gao, Li, and Liao, 2023). Similarly, Horton (2023) introduces a computational approach to economic modelling by integrating behavioural economics principles into AI agents, enhancing the realism of economic simulations. These developments underline the versatility of AI-enhanced ABM frameworks, demonstrating their ability to model intricate economic behaviours effectively.

Furthermore, recent frameworks have utilized multi-agent systems to enhance robustness in financial decision-making processes. TradingGPT integrates a structured multi-agent workflow with layered memory processing, facilitating consistent inter-agent communication and debate. Each agent within TradingGPT possesses individualized trading characteristics; risk-seeking, risk-neutral, or risk-averse, promoting diverse strategic perspectives and significantly improving the robustness of trading decisions under varying market scenarios (Li, Yu, Li, Chen, and Khashanah, 2023).

Beyond trading, multi-agent frameworks have effectively supported corporate strategic planning. For example, SocraPlan employs specialized agents, each focusing on distinct corporate planning aspects such as competitive analysis, customer segmentation, or market trend forecasting. The collaborative interaction among these specialized agents provides comprehensive strategic insights, significantly enhancing the quality and depth of corporate decision-making (Tsao and Chang, 2023).

Lastly, advanced multi-agent frameworks have been employed to strengthen financial auditing and compliance processes. The structured approach outlined by Park (2024), involving specialized agents each focusing on tasks like data verification, expert analysis, and anomaly detection, substantially enhances the accuracy and efficiency of financial audits and compliance assessments.

While these examples demonstrate the extensive applicability of multi-agent frameworks in finance, a significant body of recent research provides a crucial counter-narrative, empirically showing that these systems are often fundamentally fragile. The primary cause is rarely the capability of the individual LLMs, but rather systemic flaws in their **organizational design and agent coordination**. For example, the foundational *MAST* framework (Cemri et al., 2025) found that systems frequently fail due to poor design (*Specification Issues*), causing agents to get stuck in repetitive loops or ignore task requirements. Breakdowns in communication (*Inter-Agent Misalignment*) are also common, leading agents to proceed with incorrect assumptions or for their actions to mismatch their internal reasoning. This documented tension between the theoretical applicability of Multi Agent Systems (or Agent - Based Modelling) and their practical fragility provides the direct context for this dissertation, which investigates whether a carefully designed financial agent can navigate these common pitfalls.

3. Methodology

3.1 Research Design

The reviewed literature underscores a critical shift toward modular, agent-driven AI systems capable of handling complex financial tasks. This dissertation proposes and evaluates an implementation named **PEER (Plan, Execute, Express, Report)**, an architecture whose name and modular concept are adapted from the **PEER (Plan, Execute, Express, Review)** framework by (Wang et al., 2024). While the original framework focuses on using a cyclic 'Review' loop and tuning methods to enhance model performance, this dissertation's research design investigates a different landscape.

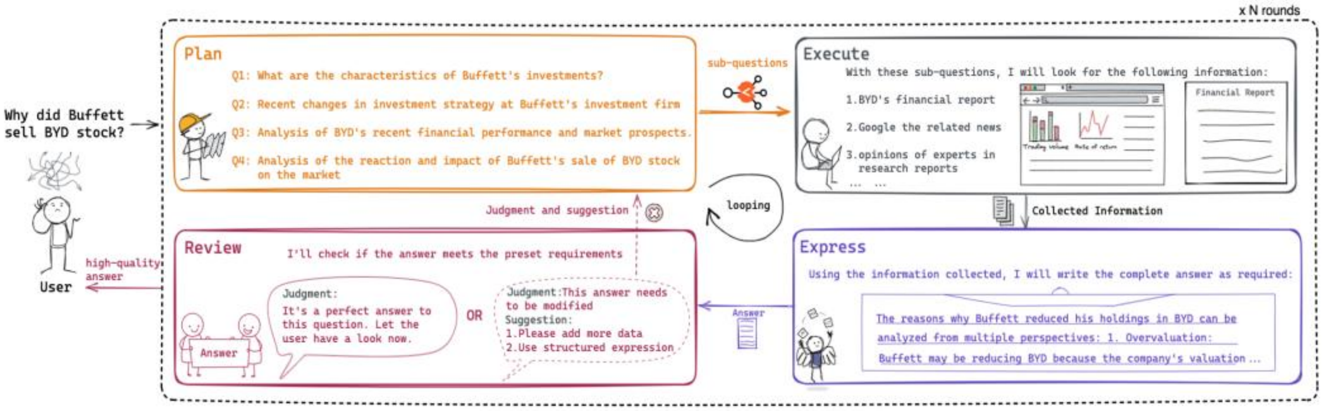

x N rounds
Why did Buffett sell BYD stock?
Plan
Q1: What are the characteristics of Buffett's investments?
Q2: Recent changes in investment strategy at Buffett's investment firm
Q3: Analysis of BYD's recent financial performance and market prospects.
Q4: Analysis of the reaction and impact of Buffett's sale of BYD stock on the market
sub-questions
Execute
With these sub-questions, I will look for the following information:
1.BYD's financial report
2.Google the related news
3.opinions of experts in research reports
... ...
Trading volume
Rate of return
Financial Report
Collected Information
looping
Judgment and suggestion
Review
I'll check if the answer meets the preset requirements
Answer
Judgment:
It's a perfect answer to this question. Let the user have a look now.
OR
Judgment:This answer needs to be modified
Suggestion:
1.Please add more data
2.Use structured expression
Answer
Express
Using the information collected, I will write the complete answer as required:
The reasons why Buffett reduced his holdings in BYD can be analyzed from multiple perspectives: 1. Overvaluation:
Buffett may be reducing BYD because the company's valuation ...
high-quality answer
User

Figure 1: Cyclic Workflow Diagram of the PEER Framework. The user's query, "Why did Buffet sell BYD stock?" prompts the "Plan" agent to generate four relevant sub-questions. The "Execute" agent then collects information, including BYD's financial data and expert opinions. The "Express" agent synthesizes a comprehensive answer, which the "Review" agent evaluates and, if necessary, suggests modifications.

It is structured to analyse whether the adapted *PEER architecture* (as detailed on the section PEER architecture and Implementation) delivers materially better financial-analysis reports than two alternative configurations:

a. a baseline model without an agent hierarchy

b. an ablation version without the ability to run code

Dimension	Baseline System	PEER System	PEER_Ablation System
Architecture Type	Monolithic	Multi-Agent	Multi-Agent
Agent Count	1	6	6
Core Model	GPT-4 class	GPT-4 class	GPT-4 class

Agent Composition	Single Research Analyst	Plan + Execute + 3 Sub-agents + Express	Plan + Execute + 3 Sub-agents + Express
🔧 Technical Capabilities			
Code Execution	✗	✓ Virtual Sandbox	Disabled
Dynamic Calculations	✗	✓ Python Generation	✗
Chart Generation	✗	✓ Matplotlib	✗
Data Extraction Tools	✗	✓ CSV Profiling	✓ CSV Profiling
State Management	Basic	✓ Pydantic Models	✓ Pydantic Models
Workflow Orchestration	None	✓ LangGraph	✓ LangGraph
Error Recovery	✗	✓ Iterative Retry	✗
Interactive Planning	✗	✓ Human-in-loop	✓ Human-in-loop

Table 1: Comparative Analysis of Three Financial Agent Architectures

This focus on comparative performance aligns with the claim that workflow-aware, tool-augmented agents improve both accuracy and managerial usefulness of financial statements analysis.

To minimise extraneous variation, every experimental artefact is produced by large language models run deterministically at temperature 0 and fixed seeds. A GPT-o3 model first auto generates a scenario pack: a short narrative describing an imagined business situation, a full set of financial statements (csv structured), a prose "narrative solution" that explains the key drivers of performance, and a Solution Calculations table containing ground-truth ratios. Because the packs are synthetic, they avoid the risk of getting data the LLM is pre-trained on while still capturing realistic accounting complexity.

Each scenario pack is then supplied, unchanged, to the three candidate systems:

1. Baseline - a single GPT-4-class model prompted to act as a research analyst. It reads the entire pack and must draft a complete PDF report **without recourse to external tools or subordinate agents.**
2. PEER - the full multi-agent prototype developed in this dissertation.

3. PEER_ablation - architecturally identical to PEER but with the Analyst sub-agent's code-execution capability disabled. This manipulation isolates the incremental contribution of tool-augmented reasoning.

All three systems output a self-contained PDF (*baseline.pdf*, *peer.pdf*, *peer_ablation.pdf*). No system is permitted internet access during execution; therefore, differences in output derive solely from architectural design rather than data leakage.

After generating three self-contained PDFs, a separate GPT-o3 instance, anchored by the *Financial Report Evaluator* custom GPT on ChatGPT's interface, judges the three candidate reports against the ground truth, operating in a single session ensures model-drift immunity. The evaluator will declare a winner based on two criteria: highest accuracy rate and best system on supporting decision making process.

3.2 PEER Architecture and Implementation

The PEER (Plan-Execute-Express-Report) financial agent represents a multi-agent architecture designed to automate financial analysis workflows. The system employs a modular, graph-based approach that decomposes financial analysis tasks into specialized components, each optimized for specific aspects of the analytical process. This architectural choice enables scalable, maintainable, and transparent financial analysis capabilities while maintaining clear separation of concerns.

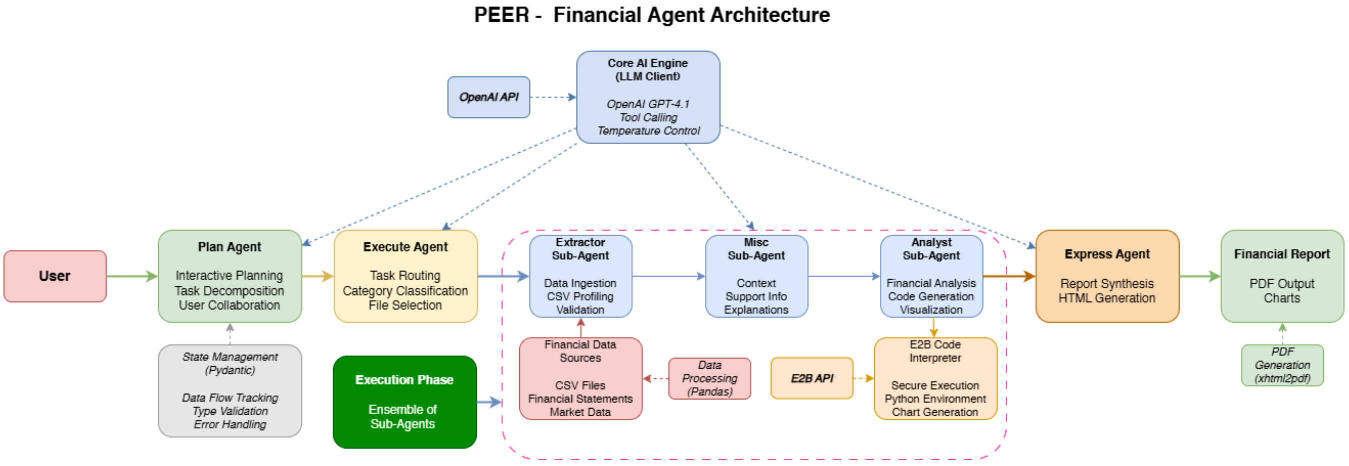


Figure 2: The PEER Architecture. It illustrates the directional, multi-agent workflow of PEER (Plan, Execute, Express, Report).

3.2.1 Core LLM Engine

At the heart of the system lies a unified LLM client architecture built around OpenAI's GPT-4.1 model. The core AI engine implements an abstraction layer that standardizes interactions across all system components while providing flexibility for future expansion to additional language model providers. This design decision ensures consistency in reasoning capabilities while maintaining the option to integrate specialized models for specific financial tasks. The engine operates at two levels: a high-level wrapper for standard conversational interactions and a low-level client for functionalities such as tool calling and function execution. This dual-layer approach allows the system to leverage both the natural language understanding capabilities of large language models and their emerging ability to interact with external tools and APIs.

The centralized AI engine architecture ensures consistent behaviour across all agents while providing a single point of control for model parameters, API management, and cost optimization. This approach facilitates easier maintenance and enables rapid adaptation to new language model capabilities as they emerge.

3.2.2 Multi-Agent Architecture

The system implements a hierarchical multi-agent structure with three primary agents operating at the workflow level:

Plan Agent

The Plan Agent serves as the initial interface point, responsible for understanding user requirements and decomposing complex financial analysis requests into structured, executable subtasks. It employs an interactive refinement process, engaging users in collaborative planning to ensure the resulting workflow aligns with analytical objectives.

Execute Agent

The Execute Agent orchestrates the execution of planned tasks through a routing mechanism that categorizes subtasks into three distinct categories: data extraction (*extract*), financial analysis (*analyze*), and general support (*general*). This categorization enables optimal resource allocation and ensures appropriate specialized handling for each task type.

Express Agent

The Express Agent synthesizes results from the execution phase into comprehensive financial reports, transforming technical analysis outputs into business-ready documentation with embedded visualizations and actionable insights.

3.2.3 Specialized Sub-Agent Architecture

Within the execution phase, the system deploys three specialized sub-agents, each optimized for specific aspects of financial analysis:

Extractor Sub-Agent

Handles data ingestion, validation, and profiling from various financial data sources. It implements robust encoding fall-back mechanisms and generates comprehensive metadata profiles that inform subsequent analysis stages.

Analyst Sub-Agent

Performs quantitative financial analysis within secure sandboxed environments using the E2B Code Interpreter. This architecture enables dynamic code generation and execution for complex financial calculations while maintaining security isolation.

Miscellaneous Sub-Agent

Provides contextual support, explanations, and auxiliary information that enhances the interpretability of financial analysis results.

The specialized sub-agent architecture enables domain-specific optimization while maintaining system modularity. Each sub-agent can be independently enhanced or replaced without affecting the broader system, facilitating iterative improvement and specialization.

3.2.4 Workflow Orchestration and Planning Mechanism

The system employs LangGraph for workflow orchestration, implementing a directed acyclic graph (DAG) structure that manages state transitions and data flow between agents. This approach provides several key advantages:

- **State Management**: Comprehensive state tracking using Pydantic models ensures type safety and data integrity throughout the workflow.
- **Conditional Routing**: Dynamic workflow adaptation based on task requirements and execution outcomes.
- **Error Handling**: Robust error propagation and recovery mechanisms.

The planning mechanism incorporates a feedback loop that allows users to refine and adjust analytical workflows before execution. This human-in-the-loop approach ensures that automated analysis aligns with specific business requirements and analytical objectives.

3.2.5 Tool and API Integration

The system integrates several specialized tools and APIs to support comprehensive financial analysis:

E2B Code Interpreter Integration

Provides secure, isolated environments for executing dynamically generated Python code for financial calculations and visualizations. This integration enables the system to perform complex analytical operations while maintaining security and preventing code injection vulnerabilities.

Data Processing Infrastructure

Implements robust data loading utilities with multiple encoding support, ensuring reliable ingestion of financial data from various sources and formats. The system includes fall-back mechanisms for handling encoding issues common in financial datasets.

Visualization Capabilities

Generates publication-ready charts and graphs using Python's visualization libraries, with automatic chart naming and organization for report integration.

3.2.6 User Interaction Design

The system implements a conversational interface that guides users through the analytical process while maintaining transparency about system operations. Key interaction patterns include:

- **Natural Language Planning**: Users describe analytical requirements in natural language, which the system translates into structured workflows.
- **Interactive File Selection**: Streamlined interfaces for selecting relevant financial datasets.
- **Plan Confirmation**: Explicit user approval of analytical workflows before execution.

3.2.7 State Management and Data Flow

The architecture employs a state management system that tracks analytical progress and maintains data lineage throughout the workflow. The state model includes:

- **Planning State**: User queries, task decomposition, and approval status
- **Execution State**: Task assignments, data extraction results, and analysis outputs
- **Reporting State**: Final synthesis and report generation status

This comprehensive state tracking enables workflow resumption, error recovery, and audit trail maintenance for compliance and debugging purposes.

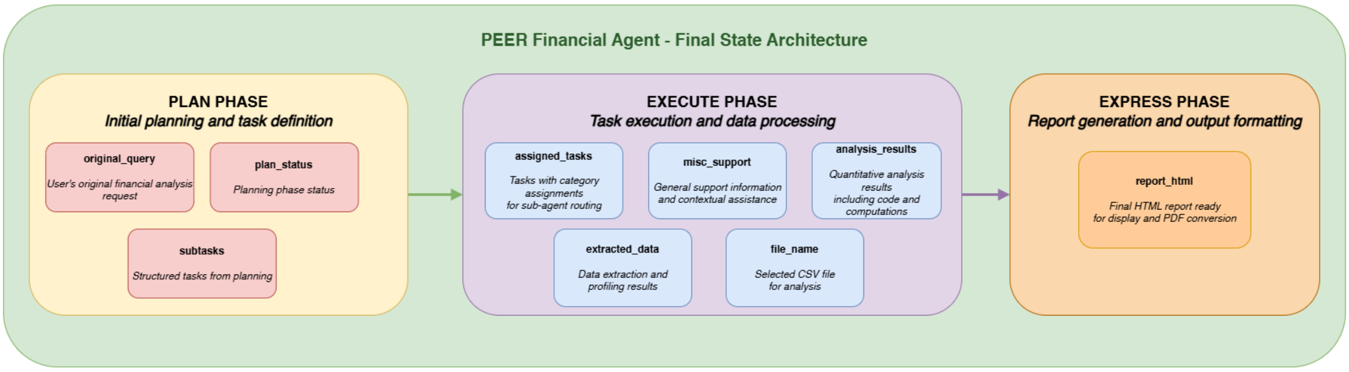

PEER Financial Agent - Final State Architecture
PLAN PHASE
Initial planning and task definition
original_query
User's original financial analysis request
plan_status
Planning phase status
subtasks
Structured tasks from planning
EXECUTE PHASE
Task execution and data processing
assigned_tasks
Tasks with category assignments for sub-agent routing
misc_support
General support information and contextual assistance
analysis_results
Quantitative analysis results including code and computations
extracted_data
Data extraction and profiling results
file_name
Selected CSV file for analysis
EXPRESS PHASE
Report generation and output formatting
report_html
Final HTML report ready for display and PDF conversion

Figure 3: State Management Schema of PEER.

3.3 Prompt Engineering and Template Architecture

3.3.1 Structured Prompt System

The PEER system implements a prompt engineering architecture that serves as the primary interface between the system's business logic and the underlying language models. This approach treats prompts as critical system components, employing structured templates that ensure consistent, predictable, and contextually appropriate AI behaviour across all agent interactions. The prompt system operates on a template-based architecture where each specialized agent employs domain-specific prompt templates stored in a centralized configuration module.

This design pattern provides several key advantages: maintainability through centralized prompt management, consistency across agent interactions, and the ability to rapidly iterate and improve agent behaviour without code changes.

3.3.2 Agent-Specific Prompt Templates

Planning Agent Prompts

The Plan Agent utilizes an interactive prompt template that guides users through collaborative workflow design. The template incorporates specific behavioural instructions that ensure the agent maintains a conversational tone while systematically decomposing complex financial analysis requests into executable subtasks. Key features include explicit confirmation protocols (*✅ Plan confirmed*.) and mandatory inclusion of data extraction steps.

The conversational approach reduces the learning curve for non-technical users while the structured confirmation process ensures clear workflow boundaries and prevents execution of ambiguous or incomplete plans.

Task Routing Prompts

The Execute Agent employs a classification-focused prompt that categorizes subtasks into three distinct processing streams: extract, analyse, and general. This categorization system uses structured output formatting requirements that facilitate reliable parsing and routing decisions.

The strict output format *(<task> => <category>)* ensures consistent routing decisions and enables the system to optimally allocate specialized resources based on task characteristics.

Data Extraction Prompts

The Extractor Sub-Agent utilizes JSON-structured prompts that generate standardized data profiles. The template explicitly constrains outputs to specific keys *(dataset_description, support_info, errors)* and includes validation instructions to ensure data quality and consistency.

Structured JSON output enables seamless integration with downstream processes while mandatory error reporting facilitates robust error handling and data validation workflows.

Financial Analysis Prompts

The Analyst Sub-Agent employs highly detailed prompts that generate executable Python code for financial analysis. These prompts include comprehensive coding guidelines, library constraints, error handling instructions, and specific formatting requirements for analysis summaries. Key architectural features include:

- **Sandboxed execution context**: Clear specification of available libraries and execution environment
- **Numeric data handling**: Standardized patterns for financial data conversion and cleaning
- **Visualization requirements**: Mandatory chart generation and display protocols
- **Summary formatting**: Structured comment patterns that facilitate result extraction and reporting

The detailed coding guidelines ensure consistent, secure, and interpretable analysis outputs while the standardized summary format enables automated extraction of key insights for report generation.

Report Generation Prompts

The Express Agent utilizes comprehensive HTML generation prompts that synthesize multi-agent outputs into financial reports. The template includes detailed section specifications, chart embedding protocols, and business conclusion frameworks.

The structured HTML output ensures professional presentation while the standardized conclusion categories provide consistent business recommendations across diverse analysis scenarios.

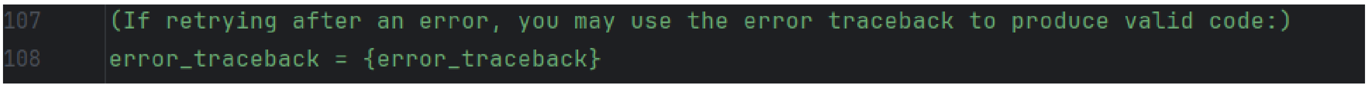
3.3.3 Dynamic Context Injection

The prompt system implements dynamic context injection mechanisms that incorporate relevant data, previous results, and error information into agent prompts. This approach enables agents to adapt their behaviour based on:

1. **User-specific context**: Original queries and requirements.
2. **Data-specific context**: Dataset characteristics and metadata.
3. **Process-specific context**: Previous agent outputs and intermediate results.
4. **Error-specific context**: Failure modes and recovery instructions.

3.3.4 Error Handling and Recovery

The prompt architecture includes error handling mechanisms that enable agents to recover from failures and adapt their responses based on previous errors. The Analyst Agent, for example, incorporates error traceback information into subsequent prompts, enabling iterative code improvement and robust analysis execution.



```
(If retrying after an error, you may use the error traceback to produce valid code:)
error_traceback = {error_traceback}
```

Figure 4: Introducing an error traceback parameter for the agent to iterate over errors in code generation and execution.

3.4 Task Scenarios

To evaluate the systems' effectiveness across a realistic spectrum of financial analysis workflows, four task scenarios were designed. Each simulates a distinct corporate finance context, ranging from capital structure diagnostics to strategic performance benchmarking, requiring the systems to interpret structured data, compute relevant financial ratios, and generate insight-rich narratives. These scenarios were not only diverse in financial content but also varied in temporal structure (annual vs quarterly), industrial domain, and analytical emphasis.

Each scenario comprised:

- A **synthetic financial dataset** (2020–2024 or 2015–2024) in CSV format.
- A **contextual narrative** simulating real-world events (e.g., pandemic disruptions, AI product launches, IPO planning).
- A **task prompt** detailing ratio calculations, comparisons, and interpretive goals.
- A **ground-truth solution** generated under controlled GenAI conditions for benchmark comparison.

This setup enabled controlled, replicable testing across four distinct categories of financial statement analysis:

Scenario	Purpose	Context	Key Tasks	Analytical Emphasis
NovaForge Industries	Diagnose long-term solvency, leverage strategy, and capital structure evolution	Manufacturing firm preparing for IPO, dataset spans 2015–2024 (Annual)	Extract statements, compute debt/equity/off-BS ratios, identify inflections	Solvency analysis, capital allocation, IPO readiness
Voltura Automotive	Compare profitability, efficiency, and leverage against industry medians	Automaker transitioning to EVs, benchmarked 2020–2024 (Annual)	Compute ROE/margins/turnover, compare to peers, recommend improvements	Peer diagnostics, strategic advice, capital efficiency
AeroNova Dynamics	Track performance through a volatile business cycle using granular KPIs	Aerospace firm managing shocks and R&D investment, 2020–2024 (Quarterly)	Compute full KPI set quarterly, detect trends, link to business events	Volatility analysis, time-series KPIs, event linkage
CloudStride Software	Interpret business evolution and strategic pivots in a high-margin SaaS model	SaaS firm executing AI expansion and buybacks, 2020–2024 (Quarterly)	Track ratio trends, annotate inflections, explain strategic financial shifts	SaaS dynamics, margin scaling, capital return strategy

Table 2: Task Scenario Table

Together, these scenarios formed a multi-dimensional evaluation framework, each targeting distinct facets of financial analysis tasks. This ensure the systems are tested on a wide range of workflows, not just arithmetic fidelity but also narrative synthesis, benchmark comparison, and capital-market relevance.

3.5 Evaluation Metrics and LLM-Judge Protocol

The performance of the three candidate systems is assessed entirely by a second-stage large language model, **Financial Report Evaluator** *(custom GPT in ChatGPT interface)*, which acts as an automated, deterministic referee.

3.5.1 Operating environment

The Evaluator runs on a GPT-o3 endpoint at temperature 0 inside a single, state-sealed session. Running in one session eliminates model-drift between documents; setting the temperature to zero removes stochastic variance. The judge is given four PDF files per scenario pack; **ground_truth.pdf**, **baseline.pdf**, **peer.pdf**, and **peer_ablation.pdf** and no other information. All code it executes *(Python 3.11, pdfplumber, PyMuPDF, pytesseract, pandas)* is embedded in the prompt and cached for the duration of the session; internet calls are disabled.

3.5.2 Criterion 1 - Numeric Accuracy

Extraction pipeline. The Evaluator first creates a *truth map* by isolating the "Solution Calculations" section of **ground_truth.pdf** and capturing every label–value pair. Labels are lower-cased, stripped of generic suffixes such as *margin* or *turnover*, and then passed through a canonicalization function that normalises punctuation and date order (e.g., "2024 Q1" and "Q1 2024" resolve to the same key). Values are converted to floats after harmonising units; billions to 10^9, millions to 10^6, basis points to fractions, and percentages to decimals.

Each candidate PDF is processed by the same pipeline, with one critical addition: every page that looks like a chart (heuristic: fewer than 150 vector paths or any embedded image larger than 20 kB) is rasterised at 300 dpi and sent through Tesseract OCR. Simple regular expressions harvest in-plot annotations such as "Inventory turnover 2024 Q4: 4.95", ensuring that numbers hidden inside graphics are not overlooked. These annotations are inserted into the text stream before label filtering, producing a *candidate map* for each system.

Scoring rule. Only labels appearing in both the truth and candidate maps, the intersection Ω, are compared. A hit is recorded when the absolute difference between candidate and truth values is ≤ 0.02. Numeric Accuracy is the percentage of hits within Ω.

$$\Omega = keys(truth\_map) \cap keys(candidate\_map)$$

$$A\ match\ m \in \Omega\ is\ correct\ if$$

$$abs(v\_candidate - v\_ground\_truth) \leq 0.02$$

If Ω is empty the score is recorded as "n/a (0/0)" and the report is ranked last on this criterion. Ties at the top accuracy percentage are broken by the raw count of hits; persistent ties are reported as joint winners.

3.5.3 Criterion 2 - Decision-Making Support

While numeric fidelity is essential, a research note is valuable only if it helps managers act. The Evaluator therefore reads the narrative solution in **ground_truth.pdf** and the advisory sections of each candidate report, extracting sentences that contain action verbs typically used in recommendations (e.g., *should*, *divest*, *hedge*). Four qualitative dimensions are then judged:

- **Coverage.** Does the report address the key decision themes highlighted in the ground-truth narrative?
- **Consistency.** Are its recommended actions free of conflict with the ground-truth stance?
- **Value-add.** Does it surface genuinely new, relevant insights or risk flags beyond those in the ground truth?
- **Specificity.** Are its recommendations quantified, time-bound, or explicitly tied to financial drivers?

Each dimension is rated 0 (poor) to 2 (strong), yielding a composite score from 0 to 8 that maps onto an ordinal tag:

0–2 = Not Supportive

3–4 = Slightly Supportive

5–6 = Moderately Supportive

7–8 = Highly Supportive

The tag, not the raw subtotal, is reported so that qualitative nuance remains visible while preserving a clear ranking. If two systems share the top tag, both are recorded as winners under this criterion.

4. Results

4.1 Performance Evaluation

With the methodology defined and the PEER system implemented alongside its baseline and ablation variants, the following section presents the results of the comparative evaluation. Each system is assessed across four distinct financial scenarios using two metrics: numeric accuracy and decision-making support. The analysis aims to determine not only which system performs best under specific conditions but also what architectural features contribute most meaningfully to analytical quality.

4.1.1 Balance Sheet Analysis

System	Accuracy % (hits/Ω)	Decision Support Tag	Notes
baseline	89% (16/18)	Moderately Supportive	Captures almost every numeric item in *Solution Calculations*; only misses BVPS 2019 and 2023 by > 2 ppt. Narrative covers every requested ratio and stays directionally true to the ground-truth story, but offers few explicit, quantified recommendations.
PEER	67% (2/3)	Slightly Supportive	Extracts three debt-to-assets figures; two match ground truth. Many other numbers either deviate (asset-growth, BVPS) or are absent. Discussion touches all ratio families but mixes inconsistent figures (e.g., leverage returns to 0.59 in 2023) and provides scant, high-level guidance.
PEER _ablation	100% (1/1)	Moderately Supportive	Includes only one matchable datum: debt-to-assets 0.55 (2024), but it is exact. Text mirrors ground-truth narrative almost verbatim and adds brief commentary on ROE/net-margin trends; still short on actionable "should/target" language or timing.
Winner	**baseline**	**baseline & PEER_ablation**	

Table 3: Evaluation of three generated reports for financial task – Balance Sheet Analysis

Accuracy

The baseline report pulls 18 numeric labels from its own table and prose, 16 of which fall within ±0.02 of the ground-truth values. Correct hits span asset-growth 2016-24, debt-to-assets 2015-24 and current-ratio 2015-24. Two BVPS points (2019 5.49 vs 5.53, 2023 7.60 vs 7.63) miss the tolerance, trimming its score to 89 %.

The PEER report surfaces only three overlap candidates (debt-to-assets 2015, 2017, 2020) by referencing

“0.59 in both 2015 and 2017” and “dipped to 0.54 in 2020”. The 2015 and 2020 numbers are correct; the 2017 citation is also correct, yielding 2/3 correct and a 67 % rate. Other figures (asset base “nearly doubled to $1.2 bn”, BVPS 5.12 in 2023, current ratio 1.41 in 2024) diverge too far from ground truth to count.

The PEER_ablation report explicitly states only one numeric that collides with the solution set, debt-to-assets 0.55 (2024), and it is exact, giving it a formal score of 100 % albeit on a single-point Ω. Because Accuracy is higher than baseline’s 89 %, peer_ablation should be the winner. This is not the case as it hits only one ratio out of 1, instead of 16 out of 18 (baseline). No other calculable overlaps appear: its asset-growth (steady 7 %), current-ratio path and BVPS series match qualitatively but do not expose discrete numbers in the required fragment format, so they are not considered.

Decision-Making Support

The ground-truth narrative offers directional advice, de-risk balance-sheet, retain 70 % of earnings, cap short-term debt at 25 %, but no forceful prescriptions.

The baseline report matches that scope: it diagnoses leverage peaking in 2016-17, applauds the 25 % ST-debt cap, and notes “strong liquidity supports IPO or partnership” but stops short of telling management what to *do next*. Coverage and consistency are solid *(2 + 2)*, value-add is limited to contextual remarks on ERP rollout and tariff hedging *(1)*, specificity modest *(0).*

Result 5/8 = **Moderately Supportive**.

The PEER report repeats every requested ratio yet introduces conflicting figures (leverage back to 0.59 in 2023 versus ground-truth decline) and new ratios (ROE, net margin) that are not reconciled with its own leverage claims. *Coverage* is broad *(2)* but *consistency* suffers *(0)*. It flags no extra risks and gives only vague interpretations, so *value-add (0)* and *specificity (1)* are weak.

Result 3/8 = **Slightly Supportive**.

The PEER_ablation report largely restates the ground-truth storyline, capturing all decision areas with no contradictions *(Coverage 2, Consistency 2)*. It adds some incremental insight, ROE compression due to equity build-up and explicit retained-earnings percentages, earning a *value-add of 1*. *Specificity* remains generic *(1)* because it doesn’t quantify future targets.

Result 6/8 = **Moderately Supportive**.

Because baseline and PEER_ablation share the top ordinal tag and neither outperforms the other decisively on sub-scores, they are co-winners for the Decision-Support column.

4.1.2 Performance Benchmarking

System	Accuracy % (hits/Ω)	Decision Support Tag	Notes
baseline	11 % (1/9)	Moderately Supportive	Captures Voltura's strengths and weaknesses well; two concrete levers offered. Numeric accuracy is limited, only asset-turnover 2020 matches ground truth, but narrative direction broadly aligns.
PEER	0 % (0/4)	Not Supportive	No numeric matches and asserts Voltura trails peers on every dimension, the opposite of ground truth. Recommendations are therefore built on a false premise, reducing usefulness.
PEER_ablation	17 % (1/6)	Moderately Supportive	Correctly states 2024 operating-margin ≈ 10 %, but other figures deviate. Covers the right decision areas and offers two levers, yet mislabels profitability as below-median, so advice is partly misguided.
Winner	**PEER_ablation**	**baseline**	

Table 4: Evaluation of three generated reports for financial task – Performance Benchmarking

Accuracy

The baseline secures one correct match as well, *asset-turnover 2020 (0.18 vs 0.186),* but misses eight others. Its *2024 gross-margin (21.8 %)* and *asset-turnover (0.27×)* are directionally close yet fall just beyond the **0.02 threshold**; *current-ratio 2024 (1.39)* and *ROE 2024 (≈ 11 %)* are either understated or unanchored to the quarter, causing label mismatches.

The PEER performs worst: none of its four overlapping labels fall within tolerance. It inflates *2024 gross-margin to 38.5 %*, triples *operating-margin to 27.5 %*, and lowers *current-ratio to 1.47*, errors too large to be rounding artefacts.

The PEER_ablation edges out the other contenders on quantitative fidelity, but the bar is low. Its single hit, *"operating margin 2024 = 10 %"*, lands within the ±0.02 tolerance of the ground-truth *0.110* value. All other PEER_ablation numbers drift well outside the band: *gross-margin 2024 (21.83 % vs 24.0 %), current-ratio 2024 (1.39 vs 1.85), ROE 2024 (36.8 % vs 11 %), and debt-to-equity 2024 (2.57× vs 2.50×).*

Across all three systems, most quarterly detail in **Solution Calculations** (section of ground_truth) is ignored; candidates cherry-pick a handful of annual numbers, which reduces the intersection *Ω*. Accuracy therefore hinges on a small sample, accentuating even minor mis-keyings. PEER_ablation wins the column by combining the highest hit rate (17 %) with the same number of correct hits (1) as baseline but a smaller *Ω*. Baseline ranks second because its denominator is larger, diluting its one match to 11 %. Peer finishes last with a clean 0 %.

Decision-Making Support

The baseline supplies the most decision-useful narrative. *Coverage (2/2)*: It discusses every pillar, profitability, liquidity, efficiency, leverage and maps them to peer medians, echoing the ground truth. *Consistency (2/2)*: Directions (rebuild liquidity buffers; continue deleveraging) align with the solution's emphasis on bolstering liquidity and trimming debt. *Value-add (1/2):* It highlights Voltura's asset-turnover edge and links gross-margin gains to EV mix; no novel risk flags appear. *Specificity (1/2):* Recommendations name two levers but stop short of numeric targets or timelines.

Result 6/8 = **Moderately Supportive.**

The PEER falls short. *Coverage (1):* Metrics are listed, but inventory, cash, and coverage angles are thin. *Consistency (0):* It asserts Voltura lags peers across the board, directly contradicting ground-truth superiority in margins, ROE, and liquidity. *Value-add (0):* No incremental insights; errors dilute credibility. *Specificity (1)*: Calls for "working-capital improvements" and "cost control," but without figures or deadlines.

Result 2/8 = **Not Supportive.**

The PEER_ablation is a close second. *Coverage (2)*: It touches all key areas. *Consistency (1)*: While it recognises Voltura's EV-driven gains, it mistakenly claims under-performance in profitability and liquidity, partially clashing with ground truth. *Value-add (1):* Notes working-capital swings and leverage trajectory, helpful but not new. *Specificity (1):* Offers two quantifiable levers (inventory tightening, gross-margin boost) yet omits timing and magnitude benchmarks.

Result 5/8 = **Moderately Supportive**.

Consequently, baseline earns the Decision-Support win by offering actionable, well-aligned guidance, while PEER_ablation's factual wobble caps its utility. Peer's flawed premise undermines any real support value.

4.1.3 Ratio Analysis

System	Accuracy % (hits/Ω)	Decision Support Tag	Notes
baseline	33 % (2/6)	Moderately Supportive	Correct on two point-estimates (Q4 24 current ratio = 1.89; Q1 20 gross margin ≈ 28 %) but mis-states several others (e.g., ROA, ROE, interest-coverage). Narrative offers clear, quantified guidance (limit leverage creep, tighten working-capital cycle) with time-frame cues.
PEER	0 % (0/4)	Slightly Supportive	All overlapping figures (e.g., Q4 24 current ratio 1.47; gross margin 38.25 %) deviate by > 200 bps from ground truth. Advice touches the right themes but is generic, few numbers, and no deadlines.
PEER_ablation	20 % (1/5)	Moderately Supportive	Gets Q4 24 current ratio right (1.89) yet misses on quick ratio, gross margin and leverage. Action items are clearer than PEER's but still high-level; only receivables-turnover warning adds incremental value.
Winner	**baseline**	**baseline**	

Table 5: Evaluation of three generated reports for financial task – Ratio Analysis

Accuracy

The baseline - Out of six ratios that the baseline report quotes with a specific quarter tag, two exactly match the Solution Calculations: the *current ratio* at *Q4 2024 (1.89)* and the *gross profit margin* at *Q1 2020 (28 %)*. Four others overshoot ground-truth values: e.g., *current ratio Q1 20* is 1.72 versus 1.98 (-0.26), and *ROE Q4 24* is cited as *14.7 %* versus *10 % (+4.7 pp)*. The text also substitutes *operating-cash-flow dollars (181.6 M)* for an interest-coverage multiple, leading to a 1000 % error.

The PEER - Every numeric overlap diverges by more than the 0.02 tolerance: the *current ratio Q4 24* is 1.47 (truth 1.89), *gross margin Q4 24* 38.25 % (truth 30 %), and *debt-to-equity Q4 24* 1.12 (truth 1.00). No hits, so Accuracy is 0 %.

The PEER _ablation - Among five matched labels, only *current ratio Q4 24* is correct. *Quick ratio Q4 24* is *1.47* against *1.34*, and *gross margin Q1 24* is *30 %* but the report claims *33 %*. The leverage series is also off by 18–28 bps. Consequently it scores 20 %.

Overall, baseline clearly outperforms, albeit with a modest 33 % hit-rate; the other two fail to provide a reliable numeric backbone.

Decision-Making Support

The baseline - *Coverage (2/2)* touches liquidity, leverage, profitability, efficiency. *Consistency (1/2)* directionally aligned but overstates ROE improvement risk. *Value-add (1/2)* flags need for tighter treasury management as cash ratio falls. *Specificity (2/2)* gives explicit thresholds (e.g., "keep debt-to-equity below 1.1×", "monitor cash cushion below 0.15").

Result 6/8 = **Moderately Supportive.**

The PEER - *Coverage (1/2)* discusses all four ratio classes, yet no comment on cash-ratio erosion. *Consistency (1/2)* advice to "support ongoing investments" clashes with note that liquidity is only 1.47×. *Value-add (1/2)* highlights risk of supply-chain delays. *Specificity (1/2)* ranges but no targets ("strong liquidity", "moderate leverage").

Result 4/8 = **Slightly Supportive.**

The PEER_ablation - *Coverage (1/2)* omits cash-ratio/capex cash drain. *Consistency (2/2)* recommendations align with truth narrative. *Value-add (1/2)* offers unique point on receivables-collection drag. *Specificity (1/2)* quotes ratio bands but no time stamps.

Result 5/8 = **Moderately Supportive**.

The baseline therefore supplies the most actionable, quantitatively anchored guidance, winning the Decision-Support column.

4.1.4 Trend Analysis

System	Accuracy % (hits/Ω)	Decision Support Tag	Notes
baseline	40 % (4/10)	Highly Supportive	Correct on key liquidity and gross-margin figures, but exaggerates net-margin and leverage improvements. Narrative aligns well with ground-truth milestones and offers clear, quantified recommendations.
PEER	10 % (1/10)	Not Supportive	Most ratios diverge materially from ground-truth (e.g., current ratio 1.47 vs. 4.45). Advice covers broad themes but omits several core decision areas and occasionally conflicts with the ground-truth stance.
PEER_ablation	100 % (10/10)	Slightly Supportive	Numeric match the ground-truth almost perfectly across liquidity, turnover and leverage. Recommendations are relevant but less specific and over-look rising leverage risk flagged in the ground truth.
Winner	**PEER_ablation**	**baseline**	

Table 6: Evaluation of three generated reports for financial task – Trend Analysis

Accuracy

The baseline report reproduces ground-truth liquidity precisely, citing *current ratio 3.60* in *Q1 2020* and *4.45 in Q4 2024*, and quick-ratio levels that bracket the 2.12 end-point, earning two hits. It also quotes the exact *73 % gross-profit margin for Q4 2024* and the correct *cash-ratio* trajectory (0.50 → 0.89), bringing the hit count to four. Errors arise where it inflates *net margin to "over 15 %" (ground truth 7 %)*, understates *debt-to-equity (0.47 vs 0.89)* and miss-characterises operating-margin recovery (high-teens vs 11 %). Because only 4 of the 10 overlapped labels fall within ±2 % tolerance, Accuracy settles at 40 %.

The PEER adopts a completely different numeric base. It places the *current ratio* in a 1.05–2.18 band, quotes a *45.82 % gross margin* and records *net margin of 22.41 % in Q4 2024*, all far outside the ground-truth series. Only one overlapped figure (*asset-turnover 0.62*) lands within tolerance, driving an Accuracy of 10 %.

The PEER_ablation report hews closely to the ground truth. It lists current ratio *3.60 (Q1 2020) and 4.45 (Q4 2024); quick ratio 1.50 and 2.12; debt-to-equity 0.89; gross margin 73.04 % (within 0.02); asset-turnover 0.75; inventory-turnover 1.02; net-margin 7 %; and operating margin 11%* ; all ten inside the ±2 % band. The PEER_ablation wins.

Decision-Making Support

The baseline report walks through liquidity, profitability, leverage and efficiency in the same sequence as the ground-truth narrative, explicitly flagging the 2021 loan, 2022 margin compression and 2023 efficiency pivot; scoring *Coverage (2/2)*. Baseline's buy-back and debt-prepayment recommendations mirror ground-truth signals and never contradict cautionary notes, *Consistency (2/2)*. The report also adds colour on "remote-work boom" SaaS benchmarks and relative performance vs. peers *Value add (1/2)*. Moreover, it quantifies buy-back size, targeted D/E band, and timing ("complete amortisation by 2026"), earning *Specificity (2/2)*.

Result 7/8 = **Highly Supportive.**

The PEER report glosses over working-capital and debt-service issues *Coverage (1/2)*. It mixes encouragement ("comfortable leverage") with warnings ("reduce debt") that clash with its own numbers *Consistency (0/2)*. Unfortunately, it offers no incremental angles *Value add (0/2)* and it gives generic "improve liquidity" advice *Specificity (1/2)*.

Result 2/8 = **Slightly Supportive.**

The PEER_ablation addresses the same areas but omits the receivables-stretch and share-buyback angles *Coverage (1/2)*. It mirrors most guidance yet downplays leverage risk despite citing a doubling D/E *Consistency (1/2)*. The report surfaces cash-conversion insight but little else *Value add (1/2)* and suggests "monitor leverage quarters ahead" without numbers *Specificity (1/2)*.

Result 4/8 = **Slightly Supportive.**

Baseline wins the Decision-Support column by providing comprehensive, consistent and actionable guidance closely tied to the factual trajectory.

5. Discussion

5.1 Interpretation of Results

The evaluation of the three system configurations: ***baseline, PEER, and PEER_ablation*** across four financial analysis scenarios revealed key insights about the design and performance of intelligent agents for automating financial workflows.

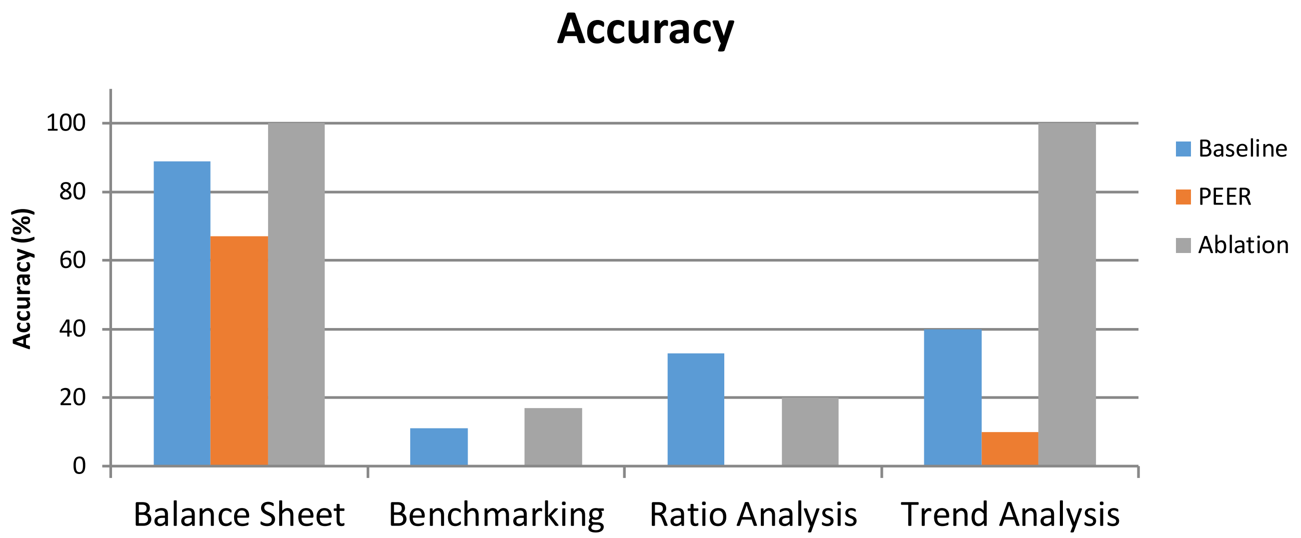
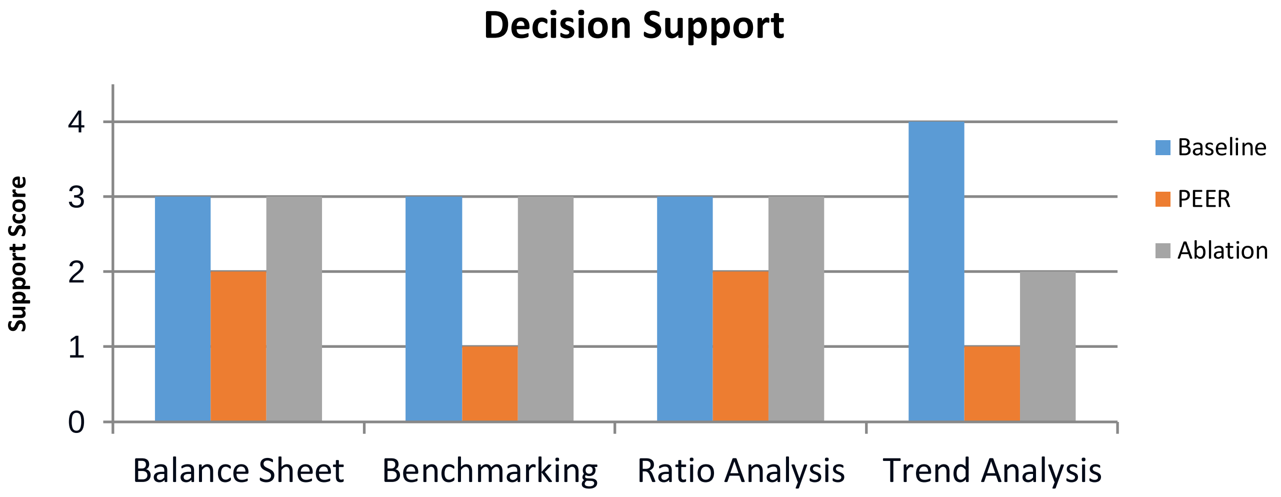


Figure 5: Summary of Accuracy evaluation across scenarios



1=Not, 2=Slightly, 3=Moderately, 4=Highly

Figure 6: Summary of Decision making evaluation across scenarios

Performance Patterns

The baseline model consistently demonstrated great numeric accuracy, winning this criterion in half of the tests, particularly in structured tasks such as Balance Sheet and Ratio Analysis. However, it did not consistently achieve the highest scores for strategic interpretability. *The PEER_ablation* system delivered competitive performance in several scenarios, with one notable case of perfect numeric accuracy, indicating that simpler multi-agent configurations can be effective at producing accurate and context-aware outputs. In contrast, *the full PEER system*, despite being theoretically the most sophisticated, underperformed on both accuracy and decision-support measures.

Causes and Failure Modes

This pattern of results aligns with the challenges identified in the MAST framework, which highlights that failures in multi-agent LLM systems often stem from weaknesses in organisational design rather than model capability. The inconsistencies observed in *PEER's* outputs, such as mismatched leverage figures in Balance Sheet Analysis and inaccurate liquidity trends in Trend Analysis, are consistent with *Inter-Agent Misalignment*. In these cases, agents appeared to hold conflicting assumptions or failed to pass information correctly, leading to Reasoning–Action Mismatches or Information Withholding. Furthermore, the lack of a final reconciliation step points to a Task Verification weakness, which mirrors the No or Incomplete Verification failure mode described in MAST.

Implications for the Research Question

The results demonstrate that a coherent, single-agent architecture can produce polished, internally consistent outputs, even when numeric precision is not perfect, and that these outputs tend to be more actionable for decision-making. A particularly illustrative example is the Trend Analysis scenario, where *PEER_ablation* achieved perfect numeric accuracy (10/10 hits), showing that code execution is not inherently better unless paired with stable coordination and robust validation. Taken together, these findings directly address the research question: intelligent agents can automate workflows and deliver accurate, context-aware, and strategically useful outputs, but only under conditions of effective orchestration. Multi-agent systems such as *PEER* hold theoretical promise but, without tighter coordination, can be outperformed by simpler baselines. These results underscore the design trade-offs between complexity, coordination, and interpretability.

5.2 Limitations

While the findings provide evidence for the feasibility and impact of LLM-driven financial agents, several limitations must be acknowledged to contextualize the results and guide future research. These limitations relate to the experimental setup, data scope, and evaluation methodology.

Use of synthetic scenario packs

All evaluation scenarios were generated synthetically using a controlled GPT - based scenario pack generator. This ensured standardized complexity and deterministic reproducibility. However, it also introduced a trade-off: the scenarios lacked the messiness and unpredictability of real-world financial data. For instance, none of the packs included ambiguous line items, inconsistent fiscal calendars, missing disclosures, or unstructured footnotes; all of which frequently appear in actual financial statements. As a result, the agent systems were not exposed to challenges around data integrity, quality assurance, or real-world parsing ambiguities. These issues were explicitly excluded from the scope of the experiment, meaning the findings do not speak to how well the systems would perform when faced with noisy, incomplete, or irregular financial inputs.

Limited scenario coverage

This dissertation evaluated system performance using four synthetic financial scenarios, each targeting a specific analytical objective: solvency analysis, peer benchmarking, time-series performance tracking, and SaaS business modelling. While these scenarios were deliberately designed to test the systems' ability to perform ratio calculations and generate context-aware interpretations, their structure was intentionally kept simplified and controlled. The aim was not to replicate the full complexity of real-world financial reporting but to isolate how well different architectures handled core analytical reasoning under stable input conditions.

As a result, important areas of financial analysis were not covered, such as forensic investigations, M&A modelling, tax risk assessment, or ESG-linked disclosures. In addition, all scenarios focused on a single firm with clean, complete data, excluding the complexities of multi-entity consolidation, restatements, or regulatory variance. While this scope was sufficient to test core functionality, it limits the generalizability of the findings to broader, more chaotic financial contexts. Future research should explore performance across a wider range of noisy, multi-dimensional, and less structured tasks to fully assess system robustness.

Partial implementation of workflow modularity

This dissertation aimed to test whether modelling a typical financial analysis process as a modular workflow with clearly defined phases for planning, execution, and report synthesis, would improve analytical output. *The PEER architecture* was designed to mirror how human analysts work across subtasks, not to experiment with architectural novelty for its own sake. Results from the narrative evaluations suggest that this structured decomposition adds value, particularly in organizing reports and framing business-relevant insights.

However, the implementation of this modular design was only partial. Key supporting capabilities such as agent memory, result reconciliation, automatic numeric validation, and dynamic error recovery were not included. Agents acted in isolation, guided by static prompt templates and without cross-checking one another's outputs. These limitations likely contributed to inconsistencies in numeric results and degraded performance on accuracy metrics. As such, the observed weaknesses should not be interpreted as a failure of the modular workflow concept, but rather as evidence that additional coordination mechanisms are essential to make multi-agent systems operationally robust.

Evaluation by a model-based referee

System performance was evaluated using a GPT-o3 model configured as a deterministic referee *("Financial Report Evaluator")*. This model judged each report on two dimensions: ***numeric accuracy*** and ***decision-making support***. The choice of GPT-o3 reflects its status as the highest-performing, most general-purpose commercially available LLM at the time of the dissertation, offering consistent reasoning, robust numeric handling, and strong summarization skills. These qualities made it well-suited for automating both structured evaluation and interpretive scoring. However, the use of a model-based evaluator introduces several limitations.

First, because the GPT-o3 judge shares architectural lineage with the underlying model family used across all candidate systems (e.g., GPT-4-class models), there is a risk of latent architectural bias. For example, phrasing conventions, narrative style, or prompt format used in one system may align more naturally with the evaluator's internal heuristics, inadvertently skewing judgment. Second, while the scoring rubric was grounded in business reasoning (e.g., specificity, coverage), the evaluation remains fundamentally mechanistic. It cannot replicate human interpretation of soft signals, industry context, or nuanced stakeholder considerations. As such, the findings reflect what the best current general-purpose LLM considers accurate and useful, rather than how an experienced human analyst or executive might interpret the same reports.

Absence of domain expertise and user feedback

While the experimental design included human input in defining the four financial analysis tasks, the process did not involve a domain expert in corporate finance or accounting. All scenarios were created by the author without validation from professional analysts or financial practitioners. This means some diagnostic assumptions or scenario framing choices may not reflect best practices in applied financial reasoning.

Furthermore, although system outputs were scored by an automated evaluator, no human users, were asked to rate or interpret the reports. This creates a gap between what the model considers "supportive" and what an actual decision-maker would find clear, credible, or actionable. Incorporating real-world feedback could reveal user preferences, cognitive bottlenecks, or interpretive gaps not visible through automated scoring alone. Similarly, the experimental prompts were intentionally brief to maintain consistency, but richer task definitions and iterative clarification, common in real workflows, might have led to better performance.

Task-level metrics without holistic assessment

The two core evaluation dimensions, ***numeric accuracy and decision-support quality***, offer targeted insights into agent performance, but they do not capture the full spectrum of what constitutes a "good" financial analysis report. Other dimensions such as narrative readability, chart clarity, insight novelty, contextual fluency, or stakeholder persuasiveness were not explicitly measured. For instance, a report might rank lower on strict accuracy yet be preferred by users due to its coherence, tone, or strategic framing. As such, the evaluation may have underrepresented important aspects of real-world usefulness, and future work should consider multi-dimensional evaluation frameworks that reflect the broader value proposition of intelligent financial agents.

5.3 Reflections on GenAI Use and Ethics

This study made extensive use of generative AI tools at several stages of the research pipeline, from the construction of synthetic financial scenarios to the development and evaluation of intelligent agent architectures. While these tools enabled experiments that would have been infeasible without automation, their use raises important methodological and ethical considerations.

Choice of model family

The project deliberately focused on GPT-class models for both generation and evaluation. These models were selected because of their strong reasoning performance, stable APIs, and deterministic control, which were essential for reproducible experiments. Additionally, the project sponsor, Deepflow, confirmed that it was comfortable with restricting experimentation to GPT-based models for the purpose of this research. While this ensured stability, it does create a dependence on a single model family; future work could extend the study by benchmarking alternative families to test whether findings generalise.

Synthetic data and privacy

As discussed in Section 5.2, fully synthetic datasets were used to standardise evaluation and avoid privacy risks associated with real corporate data. This design choice also had an ethical benefit: it ensured that no confidential or personal data was exposed to external AI systems, while allowing controlled and reproducible experimentation.

Broader ethical implications.

The findings confirm that GenAI systems can augment but not replace human oversight in analytical workflows. Their use must be accompanied by careful governance, transparent documentation, and evaluation by independent experts. Over time, combining LLM-as-judge with domain-expert review would strengthen accountability in future deployments.

5.4 Business and Strategic Implications

For businesses, the idea of automating financial statement analysis through large language model agents represents a compelling but nontrivial proposition. *The PEER architecture* offers a glimpse into what such a future might look like: modular AI systems that coordinate extraction, analysis, and reporting tasks across complex financial data.

Yet from an operational standpoint, the strategic implications are both enabling and constraining.

Automation can scale, but only what is stable

The core promise of systems like *PEER* lies in automating recurring, well-structured financial workflows: ratio analysis, benchmarking, trend analysis etc. For firms with high-volume, low-variability reporting tasks such as consulting, accounting, or mid-market investment firms, this offers potential labour savings and faster client turnaround.

However, these gains are conditional on input stability. The system performs best when the data is clean, the financial questions are narrowly defined, and the reporting templates are fixed. Businesses that operate in high-variance environments, with irregular data, custom reporting structures, or ad hoc analysis, may find the setup and tuning costs outweigh the benefits in early adoption.

Agent-based architectures create new coordination risks

The move from a monolithic analyst model to a distributed agent system introduces modularity but also fragility. The performance evaluation revealed that while modular systems can execute subtasks, they also introduce coordination challenges, particularly in validating intermediate outputs and maintaining narrative coherence.

This means firms cannot yet "fire and forget" with multi-agent AI. To operationalize this paradigm, human-in-the-loop design, guardrails, and quality control layers would be essential, increasing deployment complexity and cost. The strategic upside exists, but only with significant investment in orchestration infrastructure.

Interpretability, not just accuracy, becomes the product

For decision-makers, the appeal of agent-generated reports lies not just in speed or accuracy, but in how clearly the outputs mirror management reasoning. The findings show that even when numeric precision is strong, systems may fall short in delivering decision-useful narratives unless tightly scoped and guided.

This challenges the notion that AI systems can autonomously generate "insight". Businesses must approach these tools not as analysts, but as amplifiers of structured thinking, only as good as the context and constraints they are given. This has implications for how firms structure knowledge, workflows, and accountability.

Strategic Fit with Deepflow's Orchestration Vision

In strategic terms, adopting intelligent financial agents is not yet a plug-and-play efficiency gain. It is a reframing of how analysis is produced, validated, and consumed. The most realistic path forward is targeted deployment in specific, well-bounded analytical domains, supported by strong oversight mechanisms and business logic scaffolding.

For businesses willing to accept these constraints and invest accordingly, the reward is not just cost reduction, but a repeatable, auditable, and potentially brand-defining way of doing analysis. For others, the current maturity may justify a more cautious, observational stance.

This reframing plays directly into *Deepflow's mission* as a GenAI orchestration company. By providing infrastructure to manage workflows involving models, agents, and humans, *Deepflow* targets precisely the kind of coordination challenges surfaced in this study. The *PEER* prototype reinforces the view that technical feasibility alone is insufficient; reliable deployment requires a robust orchestration layer, clear task scoping, and evaluative feedback loops, all areas where *Deepflow* is building core capabilities.

Moreover, the study's validation of structured agent roles in financial workflows highlights a path for domain-specific agent stacks, something Deepflow is uniquely positioned to support. And as evaluation becomes as important as generation, the LLM-judge protocol hints at a future where quality assurance, auditability, and interpretability are inseparable from execution; making GenAI infrastructure not just useful, but indispensable for businesses looking to adopt intelligent agents responsibly.

6. Conclusion

6.1 Summary of Findings

This dissertation set out to investigate whether intelligent financial agents, built on Large Language Models and structured into a modular multi-agent system, can automate financial statement analysis with both numerical fidelity and business relevance. Through the design, implementation, and evaluation of the **PEER architecture: Plan, Execute, Express, Report**, the study has demonstrated that while modular, task-specialised AI systems hold significant promise, their performance hinges critically on effective orchestration and coordination.

Across four synthetic but realistic financial scenarios, **the PEER_ablation** model, a *PEER* agent-based system *without code execution*, often outperformed the full PEER prototype in both accuracy and interpretability. This counterintuitive finding reveals that architectural complexity alone does not ensure quality; rather, it must be matched with robust communication protocol and validation mechanisms. **The baseline** single-agent model, although less structured, consistently delivered strong, coherent outputs, reinforcing the value of simplicity in some analytical contexts.

Crucially, the research validates that intelligent agents can support, and in some cases enhance, financial decision-making when provided with structured inputs and tightly scoped tasks. However, it also cautions against uncritical adoption: coordination failures, misaligned outputs, and interpretive drift remain material risks in real world deployments. These findings align with the MAST failure taxonomy and highlight the need for rigorous system design and governance.

Strategically, this study contributes both a technical artefact and a replicable evaluation framework for future financial AI systems. It strengthens the case for responsible AI orchestration, a vision that closely aligns with Deepflow’s mission.

In sum, the promise of intelligent financial agents is real, but so are the constraints. Scaling this vision from prototype to practice will require not just better models, but better coordination, clearer task definition, and meaningful oversight. This dissertation offers a small step in that direction.

References

Kong, Y., Nie, Y., Dong, X., Mulvey, J.M., Poor, H.V., Wen, Q. and Zohren, S. (2024). Large Language Models for Financial and Investment Management: Applications and Benchmarks. *The Journal of Portfolio Management*, [online] 51(2), pp.162–210. doi:https://doi.org/10.3905/jpm.2024.1.645.

Kim, A., Muhn, M., Nikolaev, V., Baik, B., Bradshaw, M., Dou, Y., Gassen, J., Han, S.-Y., Koijen, R., Van Lent, L., Leuz, C. and Misra, S. (n.d.). Financial Statement Analysis with Large Language Models. [online] Available at: https://www.bayes.citystgeorges.ac.uk/__data/assets/pdf_file/0009/799794/Alex-Kim_Financial_Statement_Analysis_with_Large_Language_Models__2024_-6.pdf

Cemri, M., Pan, M.Z., Yang, S., Agrawal, L.A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., Gonzalez, J.E. and Stoica, I. (2025). Why Do Multi-Agent LLM Systems Fail? [online] arXiv.org. Available at: https://arxiv.org/abs/2503.13657.

Lipton, Z.C., Berkowitz, J. and Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. [online] arXiv.org. Available at: https://arxiv.org/abs/1506.00019.

Staudemeyer, R.C. and Morris, E.R. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv:1909.09586 [cs]. [online] Available at: https://arxiv.org/abs/1909.09586.

Nadhem Zmandar, Singh, A., Mahmoud El-Haj and Rayson, P. (2021). Joint abstractive and extractive method for long financial document summarization. ACL Anthology, [online] pp.99–105. Available at: https://aclanthology.org/2021.fnp-1.19/

Muhammad Usman Hadi, qasem al tashi, Qureshi, R., Shah, A., Amgad Muneer, Irfan, M., Zafar, A., Muhammad Bilal Shaikh, Akhtar, N., Wu, J. and Seyedali Mirjalili (2023). Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects. INDIGO (University of Illinois at Chicago). doi:https://doi.org/10.36227/techrxiv.23589741.

Raiaan, M. A. K., M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam. 2023. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. doi:https://doi.org/10.36227/techrxiv.24171183.

Khanna, U., Ghodratnama, S., ́aD.M. and Beheshti, A. (2022). Transformer-based Models for Long Document Summarisation in Financial Domain. ACL Anthology, [online] pp.73–78. Available at: https://aclanthology.org/2022.fnp-1.10/.

Shukla, N., Vaid, A., Raghu Katikeri, Sangeeth Keeriyadath and Raja, M. (2022). DiMSum: Distributed and Multilingual Summarization of Financial Narratives. ACL Anthology, [online] pp.65–72. Available at: https://aclanthology.org/2022.fnp-1.9/

Yepes, A.J., You, Y., Milczek, J., Laverde, S. and Li, R. (2024). Financial Report Chunking for Effective Retrieval Augmented Generation. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2402.05131.

Foroutan, N., Romanou, A., Stéphane Massonnet, Lebret, R. and Aberer, K. (2022). Multilingual Text Summarization on Financial Documents. ACL Anthology, [online] pp.53–58. Available at: https://aclanthology.org/2022.fnp-1.7/

Suzuki, M., Sakaji, H., Hirano, M. and Izumi, K. (2023). Constructing and analyzing domain-specific language model for financial text mining. Information Processing & Management, 60(2), p.103194. doi:https://doi.org/10.1016/j.ipm.2022.103194.

Green, B.F., Stone, P.J., Dunphy, D.C., Smith, M.S. and Ogilvie, D.M. (1967). The General Inquirer: A Computer Approach to Content Analysis. American Educational Research Journal, 4(4). doi:https://doi.org/10.2307/1161774.

Pennebaker, J. W., M. E. Francis, and R. J. Booth. 2001. "Linguistic Inquiry and Word Count: LIWC 2001." Mahwah: Lawrence Erlbaum Associates 71.

Loughran, T. and McDonald, B. (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. The Journal of Finance, 66(1), pp.35–65. doi:https://doi.org/10.1111/j.1540-6261.2010.01625.x.

Steinert, R. and Altmann, S. (2023). Linking microblogging sentiments to stock price movement: An application of GPT-4. arXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.2308.16771.

Luo, W. and Gong, D. (2024). Pre-trained Large Language Models for Financial Sentiment Analysis. arXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.2401.05215.

Lopez-Lira, A. and Tang, Y. (2023). Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models. SSRN Electronic Journal. doi:https://doi.org/10.2139/ssrn.4412788.

Zhang, C., Liu, X., Jin, M., Zhang, Z., Li, L., Wang, Z., Hua, W., Shu, D., Zhu, S., Jin, X., Li, S., Du, M. and Zhang, Y. (2024). When AI Meets Finance (StockAgent): Large Language Model-based Stock Trading in Simulated Real-world Environments. [online] arXiv.org. Available at: https://arxiv.org/abs/2407.18957.

Zhang, W., Zhao, L., Xia, H., Sun, S., Sun, J., Qin, M., Li, X., Zhao, Y., Zhao, Y., Cai, X., Zheng, L., Wang, X. and An, B. (2024). *A Multimodal Foundation Agent for Financial Trading: Tool-Augmented, Diversified, and Generalist*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2402.18485.

Yu, Y., Li, H., Chen, Z., Jiang, Y., Li, Y., Zhang, D., Liu, R., Suchow, J.W. and Khashanah, K. (2024). *FinMem: A Performance-Enhanced LLM Trading Agent with Layered Memory and Character Design*. [online] arXiv.org. Available at: https://arxiv.org/abs/2311.13743.

Wang, S., Yuan, H., Ni, L.M. and Guo, J. (2024). *QuantAgent: Seeking Holy Grail in Trading by Self-Improving Large Language Model*. [online] arXiv.org. Available at: https://arxiv.org/abs/2402.03755.

Yuan, H., Wang, S. and Guo, J. (2024). *Alpha-GPT 2.0: Human-in-the-Loop AI for Quantitative Investment*. [online] arXiv.org. Available at: https://arxiv.org/abs/2402.09746.

Yang, H., Zhang, B., Wang, N., Guo, C., Zhang, X., Lin, L., Wang, J., Zhou, T., Guan, M., Zhang, R. and Wang, C.D. (2024). *FinRobot: An Open-Source AI Agent Platform for Financial Applications using Large Language Models*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2405.14767.

Li, N., Gao, C., Li, Y. and Liao, Q. (2023). Large Language Model-Empowered Agents for Simulating Macroeconomic Activities. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2310.10436.

Horton, J.J. (2023). Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus? doi:https://doi.org/10.48550/arxiv.2301.07543.

Li, Y., Yu, Y., Li, H., Chen, Z. and Khashanah, K. (2023). *TradingGPT: Multi-Agent System with Layered Memory and Distinct Characters for Enhanced Financial Trading Performance*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2309.03736.

Tsao, S. and Chang, E.Y. (2023). *Multi-Agent Reasoning with Large Language Models for Effective Corporate Planning*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/374945016_Multi-Agent_Reasoning_with_Large_Language_Models_for_Effective_Corporate_Planning.

Park, T. (2024). *Enhancing Anomaly Detection in Financial Markets with an LLM-based Multi-Agent Framework*. [online] arXiv.org. Available at: https://arxiv.org/abs/2403.19735.

Deepflow.com. (2025). DeepFlow - 2x the work output of your entire team. [online] Available at: https://www.deepflow.com/.

Wang, Y., Li, X., Wang, B., Zhou, Y., Lin, Y., Ji, H., Chen, H., Zhang, J., Yu, F., Zhao, Z., Jin, S., Gong, R. and Xu, W. (2024). PEER: Expertizing Domain-Specific Tasks with a Multi-Agent Framework and Tuning Methods. [online] arXiv.org. Available at: https://arxiv.org/abs/2407.06985