HolisticaQuant: A Practice-Oriented Intelligent Investment Research System for Fintech

Authors: Wang Xinzhu, Hu Wanting, Peng Chuwen, Wang Xinzhu

Date: 2025-11-26T00:00:00+00:00

Abstract

With the advancement of financial technology, fintech courses in higher education are confronted with challenges including an overemphasis on theory, insufficient practice, and limited teaching tools, making it difficult for students to master end-to-end investment research operational capabilities. This paper proposes the HolisticaQuant system based on LangGraph, which realizes the semi-automation of the complete investment research workflow from data acquisition to model prediction and report generation through a three-dimensional architecture of "frontend interaction layer—intelligent orchestration kernel—multi-tool invocation system". The system incorporates an Agentic Memory mechanism to achieve context-aware and data-driven intelligent decision-making, while simultaneously enhancing student engagement and operational experience through streaming interaction and scenario-based frontend design. This platform can support both teaching and hands-on practical training, addressing the disconnect between theory and practice faced by students, and providing implementable practical tools and methodologies for fintech education.

Full Text

Preamble

HolisticaQuant: A Practice-Oriented Intelligent Investment Research System for FinTech

Chuwen Peng,[1] Wanting Hu,[2] and Xinzhu Wang[3], *
Shanghai Jiao Tong University, China
wangxinzhu3141@163.com

Project GitHub: https://github.com/xinzhuwang-wxz/HolisticaQuant

---

Abstract

As financial technology advances, university FinTech curricula face mounting challenges: an overemphasis on theory, insufficient practical training, and limited pedagogical tools, leaving students ill-equipped to master end-to-end investment research operations. This paper proposes HolisticaQuant, a LangGraph-based system that semi-automates the complete investment research pipeline —from data acquisition to model prediction and report generation—through a three-dimensional architecture comprising a frontend interaction layer, an intelligent orchestration kernel, and a multi-tool invocation framework. The system integrates an Agentic Memory mechanism to enable context-aware, data-driven intelligent decision-making, while stream-based interaction and scenario-based frontend design enhance student engagement and operational experience. The platform supports both pedagogical instruction and hands-on training, bridging the gap between theory and practice while providing a deployable practical toolkit and methodology for FinTech education.

I. Overview of FinTech Development

China' s financial technology sector has entered a period of rapid iteration, with policy frameworks consistently emphasizing digitalization, traceability, and risk control. The People' s Bank of China and relevant regulatory bodies have explicitly incorporated FinTech into top-level design through the 14th Five-Year Plan and phased policies for 2022–2025, encouraging digital transformation, inclusive finance, and deep integration between finance and technology [1]. Concurrently, regulators have imposed clear requirements on model risk, data governance, and explainability. In 2024–2025, the China Securities Regulatory Commission and industry authorities further encouraged securities firms, funds, and insurance companies to adopt digital and AI technologies to enhance investment research and risk management capabilities, while mandating standardized information disclosure, data standardization, and implementation of supporting policies for technology-driven finance. Regulatory sandboxes in local jurisdictions and Hong Kong continue to expand, providing gray-scale piloting opportunities for compliant and controllable FinTech education and training products [2].

From a capital and market ecosystem perspective, investment in AI and finance continues to grow [3], shifting focus from standalone tools or algorithms to "scenario-based, deployable workflows" and educational training platforms. The upstream data and model ecosystem remains dominated by major providers such as Wind, iFinD (同花顺), and East Money, while downstream securities firms, asset managers, and investment advisors increasingly demand traceable reports, model compliance, and integrated trading interfaces. This creates a window for embedded FinTech products emphasizing "compliance + templating + traceability." Against this backdrop, prioritizing pilot deployments in universities or institutions, interfacing with regulatory sandboxes, and planning sustainable monetization pathways will facilitate rapid validation of product models and commercial value creation.

Despite rapid FinTech development, university students and retail investors face significant learning and practice challenges. First, as a highly interdisciplinary field spanning finance, accounting, computer science, artificial intelligence, and blockchain, FinTech' s complexity exceeds the capacity of traditional education systems to cover requisite skills [4]. Current FinTech educational resources remain inadequate, with university curricula misaligned with industry demands. Most existing courses prioritize theoretical instruction, lacking access to real-world data, experimental platforms, and practical training. Moreover, the diverse and fragmented tool stack imposes high learning costs [5], preventing students from developing systematic technical and application capabilities [6, 7]. Compared to mature disciplines like traditional finance or computer science, FinTech education lacks depth and practicality. Universities generally suffer from a shortage of interdisciplinary faculty with combined finance and technology expertise, limiting course coverage of technical details, system operations, and actual business processes [8]. This faculty mismatch further prevents curriculum design from centering on real-world scenarios, weakening students' systematic mastery of core FinTech competencies. Simultaneously, many universities underinvest in experimental environments and infrastructure, lacking FinTech platforms for simulated trading, risk control testing, or automated investment research training, which denies students opportunities for applied training in industry-standard environments [7]. Additionally, the highly fragmented tool and data landscape, coupled with the inherent complexity of quantitative factor systems, machine learning models, and risk management methods, raises the barrier for students and retail investors seeking to build systematic quantitative capabilities [9]. Empirical research further demonstrates that even university students who frequently use basic FinTech tools like digital wallets exhibit only moderate financial literacy and FinTech understanding, revealing a clear gap between "tool usage" and "competency development" [5]. Pedagogically, particularly in developing countries, FinTech courses, faculty, and experimental platforms remain insufficient, preventing students from practicing in authentic contexts [7]. Overall, interdisciplinary complexity, inadequate practical training, and fragmented technical environments render FinTech learning and application considerably challenging.

At the industry level, immature university-industry collaboration systems prevent universities from accessing enterprise-grade data, system interfaces, or real business cases, denying students opportunities to engage in practical projects [10]. This structural disconnect between education and industry produces FinTech graduates who "know but cannot apply" —lacking project-based practical experience and insufficient understanding of FinTech systems, toolchains, and industry workflows [11]. For retail investors, this skills gap is even more pronounced; lacking systematic knowledge structures and access to real financial data and investment research tools, they struggle to develop deep understanding of FinTech products, quantitative strategies, and risk management.

As artificial intelligence becomes widely applied in financial investment, the applicability and robustness of its models in extreme market conditions has emerged as a critical concern. In highly uncertain financial environments, "black swan" events—characterized by low probability but extreme impact—occur frequently. Under such conditions, AI models trained on historical data often fail to predict future price movements, producing outputs that tend toward sycophantic 迎合, vague conclusions, and poor explainability, thereby reducing investment advice accuracy and increasing risk exposure. According to Nassim Nicholas Taleb' s black swan theory, many extreme events in financial systems (low-probability, high-impact) cannot be properly predicted by traditional models [12, 13]. While reinforcement learning algorithms perform well in normal market conditions, their performance degrades significantly during black swan events like market crashes. AI models often exhibit "overconfidence" and "insufficient risk perception" [14, 15]. When models lack mechanisms to model true risk structures, their predictive outputs may appear reasonable but deviate substantially from actual market movements during extreme events, potentially maintaining optimistic biases even as volatility intensifies [14]. This demonstrates that models relying solely on historical data cannot cope with sudden black swan events. Financial market complexity may cause AI training to suffer from "overfitting to historical patterns," generating misleading overconfident predictions. When models perform strongly on training data, their ensemble predictions may reduce robustness in real contexts and cause systematic misjudgments during rare events [16], consistent with observed failures of investment AI during black swan events.

The FinTech sector also faces challenges in information and monitoring mechanisms. Research indicates significant "data gaps" in FinTech-related data, with monitoring and reporting mechanisms remaining underdeveloped [17]. Despite rapid FinTech technology and application expansion, information coverage, application scenarios, and channel integration remain in early stages [18]. Consequently, no channel comprehensively and timely covers FinTech news and industry dynamics, posing challenges for students, researchers, and practitioners. Overall, high learning barriers, scarce practice opportunities, and fragmented tools and data disconnect learning from career development, necessitating systematic, practice-oriented solutions. Building a practice platform oriented toward real-world scenarios with project-based learning mechanisms that connects universities and industry has become a critical trend for improving FinTech education quality.

II. HolisticaQuant System Design and Core Modules

Addressing these pain points, this study proposes HolisticaQuant—an AI-powered FinTech education and technology solution for financial institutions and universities. The system embeds modularly into digital banking or university experimental platforms to construct an "AI FinTech Learning Lab," providing intelligent education, investment research, and decision-making training services. Through an embedded AI portal design and natural language-driven learning and analytical tools, users can conduct FinTech learning, practical operations, and data analysis within existing digital banking apps or teaching systems. HolisticaQuant' s core value lies in lowering FinTech learning barriers, delivering actionable intelligent learning and practical experiences, and establishing replicable talent cultivation and innovation pipelines for universities and financial institutions. The product design balances educational, research, and quantitative trading practice needs, satisfying both academic training requirements and enterprise digital transformation and talent incubation objectives, forming a university-industry-research collaboration platform.

HolisticaQuant's design logic derives from both policy and market environments and from pain points in university education and financial institution practices. The system achieves closed-loop experiences in education, research, and decision-making through modular embedding in digital banking apps or university experimental platforms. On one hand, it fills gaps in university FinTech talent cultivation and practical operations, promoting university-industry-research synergy; on the other, it provides quantifiable and reviewable investment research and decision support for financial institutions, creating commercial value for embedded services and forming a data moat to ensure long-term competitive advantage. In this context, HolisticaQuant decomposes into three complementary core modules, each addressing specific user group needs while collaboratively building a complete learning-analysis-decision chain.

In HolisticaQuant' s design, three modules serve as complementary core scenarios to meet different user group needs while achieving the system' s overall value 闭环. The Learning Studio module focuses on FinTech talent cultivation and practical training, providing students and institutions with an immersive AI learning environment. The Research Lab module targets investment researchers, improving research efficiency and achieving risk quantification through end-to-end intelligent data analysis and strategy generation. The QA Engine module 下沉 professional investment analysis capabilities to end users, generating personalized investment strategies and risk warnings through natural language interaction. These three scenarios address the need for university-industry-research integration while embodying practice-oriented and embedded design concepts, providing HolisticaQuant with a complete education, analysis, and decision-making 闭环.

The following sections elaborate on each module, presenting its functional positioning, scenario experience, and potential commercial value to fully demonstrate HolisticaQuant' s system design and application prospects in FinTech education and intelligent investment research practice.

A. Learning Studio—AI FinTech Talent Incubation Factory

The Learning Studio module aims to create a high-quality talent cultivation and practical training platform for the FinTech domain. The system integrates educational content, interactive experiments, and intelligent assessment to construct an immersive AI learning environment. Users access the module through a financial experimental platform, where an AI mentor dynamically decomposes tasks based on learning objectives, generating 3D visual experimental models, risk simulations, and competency radar assessments. For instance, when a student proposes a FinTech design task, the system guides them to understand core principles, identify key features, and complete system-level architecture modeling and validation. Learning Studio supports modular integration, seamlessly embedding into university curricula and in-service training systems for financial institutions to form an "Education as a Service (EaaS)" model. Its core value lies not only in providing a continuous talent supply pipeline but also in establishing stable commercialization pathways through subscription and competency assessment services. The overall design follows the "education closed-loop + AI-accompanied practice" concept, constructing a replicable and scalable FinTech talent incubation system.

In a practical Learning Studio scenario, a graduate student logs into the platform and selects a learning case on blockchain payments and digital currency pilots. The left side of the interface presents multiple task options, while the right timeline displays real-time AI decomposition and guidance. The AI first introduces pilot objectives and key data—for example, pilot implementations in several key cities covering numerous registered users and generating large transaction volumes. Subsequently, the AI guides the student to apply the growth rate formula:

$$\text{Growth Rate} = \frac{\text{Post-Pilot Revenue} - \text{Pre-Pilot Revenue}}{\text{Pre-Pilot Revenue}}$$

to calculate the magnitude of change in the platform' s digital business and analyze the pilot' s potential impact on operational efficiency and user behavior. Throughout the process, the AI mentor dynamically explains each step's logic in the learning timeline, generating real-time analysis reports. Upon task completion, the system automatically generates a learning summary and competency radar chart, displaying the student' s improvement trajectory across dimensions such as data analysis, financial understanding, and system design.

B. Research Lab—End-to-End Intelligent Investment Research Decision Engine

The Research Lab module positions itself as an intelligent investment research experimental platform, focusing on digitizing and automating traditional complex research processes to achieve a closed loop from information collection to strategy generation. The system processes multidimensional data including financial reports, news, patents, and supply chains, automatically constructing competitive matrices and backtesting models to generate analysis reports and SWOT assessments. Users need only input research questions—for example, "Analyze the development potential of a new energy battery company after a technological breakthrough" —for the system to rapidly complete data crawling, organization, and analysis. The module emphasizes data-driven analytical logic and risk quantification methods, replacing manual collection, cleaning, and modeling through automated workflows to improve research efficiency and reduce operational complexity. From a capital narrative perspective, Research Lab demonstrates not only technical differentiation and data processing capabilities but also provides foundational support for subscription services and value-added products across multi-tiered research institutions.

In a practical Research Lab scenario, a graduate student logs into the platform and selects a valuation case for a new energy battery company. The left side of the interface provides multiple strategy options, while the right side displays the AI' s real-time analysis process. The AI first outlines macro industry trends—for example, that the global new energy sector is undergoing structural adjustment with overall penetration increasing but growth rhythms varying across segments. Subsequently, the AI guides the student to analyze individual stock financial and operational indicators, such as quarterly revenue and net profit changes, gross margin trends, and R&D investment directions, while prompting attention to upstream/downstream supply chains and market competition conditions. In the micro-analysis phase, AI-generated data reports cover dimensions including corporate production capacity, market share, technological progress, and potential risk points, assisting students in understanding the company' s position in the industrial chain and key operational factors. Risk analysis covers industry competition, demand fluctuations, and raw material price changes, while also alerting to potential technological iteration and customer concentration risks.

C. QA Engine—Decision-Level Financial AI Assistant

The QA Engine module targets retail and institutional users, aiming to 下沉 professional analytical capabilities to end users and achieve a closed loop from information event capture to personalized analysis and strategy generation. Users can pose questions in natural language—for example, "Market trends after a commodity price breaks a key threshold" —and the system instantly integrates global news, market fluctuations, and quantitative indicators to generate understandable analysis reports and potential risk warnings, supporting simulated analysis experiences. The module' s core lies in transforming complex research logic into easily understandable and operable user experiences, making AI a 24/7 analytical assistance tool. Under capital narrative logic, QA Engine constructs a closed-loop path of "event—analysis—strategy—decision," demonstrating the value of data integration and algorithmic reasoning while providing a sustainable design foundation for premium subscriptions and value-added services.

In a practical QA Engine scenario, a user accesses the system and selects a corporate valuation analysis task. The left side of the interface provides a question input box and typical example questions, while the right side displays the AI' s analysis process in real time, including conclusions, logic chains, and data sources. When the user inputs a question such as "Is the valuation level of a new energy company reasonable," the system first integrates macro industry data to explain overall market trends, competitive landscape, and changes in key financial indicators. Subsequently, the AI conducts micro-analysis on the individual stock, including comparisons of rolling P/E ratios, P/B ratios, and other key financial ratios against industry averages or historical levels. During analysis, the system dynamically generates summary reports to help users understand the company' s position in the industrial chain and potential operational risks. The entire learning process focuses on data analysis capability and logical reasoning training.

In summary, HolisticaQuant constructs a complete university-industry-research closed loop through its three core modules—Learning Studio, Research Lab, and QA Engine: from university talent cultivation and practical training to automated analysis of professional investment research processes, and finally to intelligent decision assistance for end users, achieving a continuous chain of education, analysis, and decision-making. The system demonstrates exemplary value in university-industry-research integration. Through modular embedding into digital banking apps or university experimental platforms, the system not only optimizes user experience and operational efficiency but also ensures traceability in data processing and analysis, forming a potential data and algorithm moat. The overall design satisfies both university education and financial institution practice needs while laying the foundation for future commercial monetization models, establishing a solid systematic basis for HolisticaQuant' s deployment and expansion in FinTech education and intelligent investment research.

III. System Architecture

A. System Architecture Overview

HolisticaQuant' s overall system architecture is illustrated in Figure 1 [Figure 1: see original paper], employing a "three-layer coupled architecture" comprising a frontend interaction layer, an intelligent orchestration kernel, and a multi-tool invocation framework. The frontend is responsible for scenario-based interaction and information visualization. The intelligent orchestration kernel, built on LangGraph [19], enables dynamic scheduling of multi-agent workflows. The tool invocation framework provides unified interfaces to access external data sources and computational tools. The system implements asynchronous frontend-backend communication through dual channels of REST API [20] and WebSocket [21], while supporting context-aware intelligent decision-making and generating traceable strategy reports.

B. Frontend Design: Scenario-Based Experience and Streamed Interaction

The frontend employs React 18 [22] + Vite [23] + TypeScript [24] as core frameworks, emphasizing modular development and maintainability. Visual design is driven by TailwindCSS [25], achieving thematic consistency through a unified visual token system, and incorporating glass morphism and noise layering effects to create a lightweight yet layered interactive style.

For dynamic interaction, Framer Motion [26] is introduced to implement page fade-ins and key visual animations, enhancing the rhythm and narrative quality of information presentation. In research and Q&A scenarios, the system adopts a WebSocket streaming rendering mechanism, presenting output content in segments of approximately 500 characters with typewriter-style playback and stage progress bars to achieve a "reasoning visualization" user experience. This approach not only enhances immersion but also improves the comprehensibility and process-awareness of complex reasoning results.

The overall design philosophy is "research scenario-centric." Users can complete question posing, model analysis, and result retrospection within a unified logical space, forming a closed-loop operational experience.

C. Backend Design: Adapted to Financial Education Research and Data Needs

The backend centers on FastAPI [27] and LangGraph. FastAPI handles communication and interface management, supporting both REST and WebSocket channels. REST serves static queries and scenario management with average response times under 500 ms; WebSocket serves complex reasoning streams and state event transmission, enabling real-time progress tracking and intermediate result interpretability analysis. The system defines unified message formats through Pydantic [28], enabling direct frontend rendering and reducing parsing complexity.

1. Agent Layering System agents are divided into five functional modules [29]: - **Plan Agent**: Parses user intent, identifies research scenarios, and extracts target metrics; - **Data Agent**: Invokes external data interfaces and internal tools to execute multi-round data collection and sufficiency assessment; - **Strategy Agent**: Generates strategies, recommendations, and risk warnings based on data analysis and historical insights; - **Learning Workshop Agent**: Outputs training tasks and scripts for pedagogical scenarios; - **Q&A Agent**: Handles Q&A tasks, generates natural language narration, and integrates citations.

All agents manage inputs and outputs through a unified state object (AgentState), avoiding prompt pollution and ensuring inter-module 衔接 is visualizable, debuggable, and replayable.

2. Core Execution Main Link LangGraph constructs the system' s core execution link. After user requests are parsed by the Plan Agent, the Data Agent automatically aggregates multi-source information, including active data and passive market information, and decides whether to conduct additional collection based on data sufficiency and iteration thresholds, achieving adaptive information refinement. Once information preparation is complete, tasks are routed to the Strategy Agent, Learning Agent, or Assistant Agent for content generation. If data support is found insufficient, the system callbacks the Data Agent for supplementary collection, enabling self-correction of the generation link. The entire workflow operates around the unified AgentState, with intermediate states, errors, and tool calls tracked and recorded in real time, achieving interpretable, auditable, and replayable intelligent decision-making.

D. Core Modules and Operating Mechanisms

1. Configuration Management System The system employs a three-level configuration logic of "defaults → configuration files → environment variables," supporting dynamic overrides through hierarchical naming (e.g., LLM___{PROVIDER}). Key features include: - **Multi-model management**: Supports multiple LLM APIs for flexible selection and model evaluation; - **Data source prioritization and fallback mechanisms**: Establishes primary and fallback suppliers for market, fundamental, and news data, with automatic switching upon interface timeout; - **Agent parameter injection**: Maximum iteration counts, reflection toggles, and quality thresholds are controllable via environment variables; - **Centralized RAG configuration**: Including vector storage toggles, insight limits, forgetting cycles, and extraction strategies.

2. Memory System The Agentic RAG [30, 31] memory system serves as the knowledge foundation, comprising FinancialReasoningEngine and FinancialInsightMemory. The execution flow is: strategy generation → Reasoning Engine extracts insights → writes to memory bank → queries retrieve relevant insights by keywords, ticker, and intent for injection into context for model reference. The system supports forgetting mechanisms and degradation strategies, automatically switching to keyword matching mode when vector databases are unavailable.

3. Data Tools and Abstraction Layer The data abstraction layer uniformly encapsulates external data interfaces (e.g., AkShare [32], Sina [33], THX News [34]) with caching, degradation, and statistical mechanisms. Data requirements generated during the Plan phase drive the Data phase to invoke corresponding tools, achieving "plan-driven data scheduling." The system also supports combining historical insights with real-time data to form a hybrid "Agentic RAG + real-time stream" data context, providing rich context for strategy generation.

4. Tool and Execution Layer All tools follow a unified abstract interface defining execution parameters, descriptions, and return structures. The system includes built-in tools for calculators, database queries, and web searches. Tool enable/disable is configuration-driven, with failure rate statistics triggering automatic degradation or alternative prompts to ensure long-term operational stability.

E. System Workflow and Execution Mechanisms

HolisticaQuant employs a multi-agent collaborative architecture to achieve full-process automation from task planning and data collection to strategy generation and knowledge explanation. The overall workflow is illustrated in Figure 2 [Figure 2: see original paper]. Through layered agent design, the system decomposes complex financial analysis processes into parallelizable, traceable subtasks, constructing a high-cohesion, low-coupling workflow execution mechanism.

The system operation flow 主要包括以下阶段: 1. **Scenario Initialization Phase**: Users select predefined analysis templates or submit custom tasks through the frontend workbench. The system loads scenario metadata via REST interfaces, automatically configures default parameters and environmental context, and establishes WebSocket real-time communication channels to support streamed result delivery. 2. **Agent Collaborative Execution Phase**: The HolisticaGraph engine dynamically schedules multiple agent types based on task characteristics: planning agents for task parsing and execution path planning; data agents for multi-source data collection and preprocessing; strategy agents for generating investment strategies and risk assessments; and teaching agents for domain knowledge explanation and learning support. All agents share information and context through a shared state object (AgentState), ensuring traceable and debuggable task 流转. 3. **Real-Time Delivery and Visualization Phase**: The system employs dual-channel delivery: streaming channels for real-time push of analysis results (e.g., data slices, strategy summaries, explanatory text), and memory banks for persistent storage of intermediate results and knowledge fragments. The frontend workbench aggregates streamed data in real time, providing progress feedback, result visualization, and interactive operations. 4. **Result Integration and Output Phase**: Agent outputs are aggregated through unified interfaces into structured research reports, risk assessment matrices, and instructional guidance content. Users can export results via interfaces, forming a complete "analysis—explanation—output" closed loop.

The system achieves three key innovations in the above mechanisms: - **Dynamic Task Routing**: Reinforcement learning-based agent scheduling strategies adaptively allocate resources according to task characteristics, improving execution efficiency; - **Cross-Agent Knowledge Sharing**: Distributed memory banks support multi-session knowledge reuse and version management, ensuring analytical consistency; - **Progressive Result Delivery**: Streamed transmission protocols collaborate with frontend rendering engines to support incremental result display, optimizing user waiting experience.

In summary, HolisticaQuant’ s workflow centers on agent collaboration, achieving an automated closed loop from financial problem understanding to strategy generation and teaching feedback. This ensures system scalability and interpretability while providing a unified technical framework for financial education and investment research applications.

F. Infrastructure and Runtime Environment

The system operates flexibly in local development or cloud-native environments. In development mode, Python virtual environments isolate dependencies, while the frontend supports hot reloading and rapid iteration via local npm services. The development toolchain is built on Poetry [35] and Vite to ensure dependency consistency and modular development experience. In cloud-native mode, the system supports one-click deployment to mainstream platforms including Vercel [36] and Railway [37], automatically completing dependency installation, environment variable injection, and horizontal scaling. Core components include data acquisition driven by AkShare and news scraping components, computational logic supported by Calculator toolsets, and storage using a lightweight combination of JSON files and SQLite [38] databases, balancing performance and convenience. At the operations level, the system implements unified monitoring via Prometheus [39], error tracking through Sentry [40], and structured logging pipelines to support observability and traceability, forming a stable and transparent operational closed loop.

IV. Conclusion

HolisticaQuant addresses the high barriers, limited practice opportunities, and fragmented data and tools plaguing FinTech education and investment research practice. Through modular design, the system closes the loop on education, research, and decision-making processes, providing immersive learning and practical experiences for students, in-service financial professionals, and research institutions. The Learning Studio module offers scenario-based AI learning environments and competency radar assessments, enabling users to complete systematic tasks through visual experimental models and dynamic guidance after posing questions. The Research Lab module achieves full-process automation from information collection to strategy analysis, improving research efficiency through multidimensional data integration and risk quantification methods. The QA Engine module transforms complex analytical logic into understandable outputs and potential business insights through natural language interaction. The overall design achieves a complementary closed loop of education, analysis, and decision-making while echoing the practice-oriented integration of university, industry, and research.

The system’ s innovation manifests across multiple dimensions. First, modular embedded design enables HolisticaQuant to flexibly integrate into existing digital financial platforms or teaching systems, realizing an Education as a Service (EaaS) model that seamlessly connects learning with real business environments. Second, AI-accompanied learning and scenario-based practice break traditional theory-practice divides, enabling users to complete tasks in real or simulated business contexts to form learning closed loops. Third, end-to-end research closed loops and automated strategy generation make data integration, analysis, and risk assessment processes highly transparent and traceable, significantly improving efficiency and quantifiable value. The system' s unified intelligent workflow architecture allows the same framework to extend to different scenarios, demonstrating significant reusability and expansion potential. Additionally, the implicit ecosystem layout enables the system to leverage embedded data flows to form a data moat, strengthening long-term competitive advantages while creating commercialization space for future subscription services, competency assessments, and value-added analytical products.

In terms of feasibility, HolisticaQuant rests on solid technical foundations. LangGraph multi-agent workflows and frontend-backend asynchronous communication mechanisms ensure real-time reliability of learning and analytical tasks, while tool invocation interfaces enable integration of existing data sources, reports, and third-party APIs for efficient module collaboration. The current phase employs open-source and free data sources for training and strategy demonstration, with future plans to introduce higher-precision data sources to further enhance analytical credibility. Commercially, the system designs multi-dimensional leverage paths, including entry through educational institutions and industry incubation channels, combined with data resource and service partnerships to form a closed loop of training—trial—commercialization, laying the foundation for capital expansion and long-term revenue growth.

HolisticaQuant' s potential limitations also warrant attention. Data quality and coverage constraints may affect strategy generation and learning accuracy, while model generalization capabilities require validation across different market environments. Demand differences among user groups challenge module adaptability, and business model maturity is subject to market and regulatory factors. Furthermore, the system' s reliance on multi-agent and real-time data stream processing imposes high requirements on technical maintenance and upgrades, necessitating continuous optimization to ensure long-term stable operation, especially under large-scale concurrent usage.

Overall, HolisticaQuant organically combines FinTech education and investment research practice through modular, scenario-based, and closed-loop design, forming an operable and scalable learning and analysis platform. The system balances educational closed loops, intelligent investment research, and potential commercial value while implicitly constructing data and process advantages, providing practical solutions for university talent cultivation and financial institution digital capability enhancement, and demonstrating frontier exploration value in the fields of FinTech education and automated investment research.

V. Acknowledgments

We express sincere gratitude to Caroline and Jacky for their invaluable suggestions and significant contributions throughout this research, which enabled the successful completion of this work.

References

[1] People' s Bank of China, *FinTech Development Plan (2022-2025)*, Tech. Rep. (People' s Bank of China, 2022).
[2] China Banking and Insurance Regulatory Commission, *FinTech Regulation and Sandbox Mechanism in Pilot Zones*, Tech. Rep. (CBIRC, 2021).
[3] KPMG China, *2023 China Fintech 50 Report*, Tech. Rep. (KPMG China, 2024).
[4] K. Gai, M. Qiu, and X. Sun, Journal of Network and Computer Applications 103, 262 (2018).
[5] W. Juliyanti, R. Mustafa Zahri, E. Wulan Sari, and A. Nur Aziz, in *Proceedings of the 4th International Conference on Economics and Social Science* (Atlantis Press, 2023).
[6] R. R. Suryono, I. Budi, and B. Purwandari, Information 11, 1 (2020).
[7] A. W. Menberu, Cogent Education 11 (2024), 10.1080/23311975.2023.2294879.
[8] I. Lee and Y. J. Shin, Business Horizons 63, 35 (2020).
[9] J. Ye, H. Zhang, and Q. Li, Finance Research Letters (2024).
[10] P. Gomber, R. J. Kauffman, C. Parker, and B. W. Weber, Journal of Management Information Systems 35, 220 (2017).
[11] G. Cornelli, J. Frost, L. Gambacorta, P. R. Rau, and R. Wardrop, BIS Working Papers (2020).
[12] N. N. Taleb, *The Black Swan: The Impact of the Highly Improbable* (Random House, 2007).
[13] Sustainable Finance Observatory, "All swans are black in the dark," (2017).
[14] S. Hong, Preprint on ResearchGate (2025).
[15] "Navigating black swan events in algorithmic trading: A reinforcement learning perspective," (2024).
[16] Y. Grushka-Cockayne et al., Management Science 63, 1805 (2017).
[17] J. M. Marqués, B. H. Cohen, and M. Demertzis, Journal of Financial Regulation and Compliance 29, 375 (2021).
[18] G. Kou and Y. Lu, Financial Innovation 11 (2025), 10.1186/s40854-024-00668-6.
[19] J. Wang and Z. Duan, "Agent ai with langgraph: A modular framework for enhancing machine translation using large language models," (2024), arXiv:2412.03801 [cs.CL].
[20] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, Ph.D. thesis, University of California, Irvine (2000), doctoral dissertation.
[21] G. L. Muller, "Html5 websocket protocol and its application to distributed

computing," (2014), arXiv:1409.3367 [cs.DC].
[22] T. R. Team, "React v18.0," Blog post, React official site (2022), march 29, 2022 – "Our latest major version includes out-of-the-box improvements⋯" .
[23] E. You and the Vite Team, Journal of Web Engineering X, ⋯(2024), see "Getting Started" on the Vite docs.
[24] M. Corporation, "Typescript," (2024), version x.x –typed JavaScript at scale.
[25] T. Labs, "Tailwind css," (2025), utility-first CSS framework.
[26] I. Framer, "Motion (formerly framer motion)," (2025), production-grade animation library for React/JS.
[27] S. Ramírez, "Fastapi," (2023), modern high-performance Python web framework based on type hints.
[28] N. D, "Building a structured research automation system using pydantic," Analytics Vidhya Blog (2025).
[29] Z. Durante, Q. Huang, N. Wake, R. Gong, J. S. Park, B. Sarkar, R. Taori, Y. Noda, D. Terzopoulos, Y. Choi, K. Ikeuchi, H. Vo, L. Fei-Fei, and J. Gao, "Agent ai: Surveying the horizons of multimodal interaction," (2024), arXiv:2401.03568 [cs.AI].
[30] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, "Agentic retrieval-augmented generation: A survey on agentic rag," (2025), arXiv:2501.09136 [cs.AI].
[31] Q. Zhang, C. Hu, S. Upasani, B. Ma, F. Hong, V. Kamanuru, J. Rainton, C. Wu, M. Ji, H. Li, U. Thakker, J. Zou, and K. Olukotun, "Agentic context engineering: Evolving contexts for self-improving language models," (2025), arXiv:2510.04618 [cs.LG].
[32] A. Community, "Akshare: A free and open-source python financial data interface library," (2025), accessed: 2025, open-source finance data for Chinese markets.
[33] S. Corporation, "Sina finance: Comprehensive financial information platform," Web portal (2024).
[34] N. A. Team, "News api: Search and retrieve live & historical news via rest api," (2025), accessed: 2025, provides structured news data for developers.
[35] P. Developers, "Poetry: Python dependency management and packaging made easy," (2025), open-source tool for dependency management and packaging in Python.
[36] V. Inc., "Vercel: Frontend cloud platform for static and serverless deployments," Web platform (2025), provides serverless hosting and CI/CD for modern web frameworks.
[37] R. Team, "Railway: Cloud deployment platform for developers," Web platform (2025), simplifies cloud app deployment with automatic builds and environment management.
[38] D. Richard Hipp, *SQLite Documentation* (2000), widely used lightweight relational database engine.
[39] B. Brazil and J. Volz, in *Proceedings of the 2015 USENIX Conference on Advanced Topics in Systems and Software Practice* (2015) open-source systems monitoring and alerting toolkit.

[40] I. Functional Software, "Sentry: Real-time error tracking and performance monitoring platform," Web platform (2025), provides real-time error tracking and observability for applications.

Note: Figure translations are in progress. See original paper for figures.