Fin-Agent-QG: A Collaborative Multi-Agent Framework for Higher-Order Cognitive and Difficulty-Aware Financial Exam Question Generation

Xuan Yao[1], Yi Zhou[1], Xiaoyu Qu[1], Lulin Lyu[1], Zefan Zhang[1], Zac Wong[2], Ke-Wei Huang[1]

[1]Asian Institute of Digital Finance, National Univeristy of Singapore
[2]School of Computing, National University of Singapore
yaoxuan@nus.edu.sg, {e1350671, e1561354, e0902038}@u.nus.edu, zefan002@e.ntu.edu.sg, {zac_wong, dishkw}@nus.edu.sg

Abstract

Automated Question Generation (AQG) plays a crucial role in enabling scalable assessment and personalized learning, serving as a cornerstone of modern Intelligent Tutoring Systems (ITS). However, existing AQG systems, particularly those built upon Large Language Models (LLMs), face a significant cognitive ceiling in generating high-stakes, exam-ready content for specialized domains like finance (e.g., CFA Exams). They often produce linguistically fluent yet cognitively shallow items that fall short of expert-level assessment standards. Such standards demand precise alignment between question difficulty and learning objectives, rigorous quantitative reasoning, and the construction of non-trivial distractor options that reflect common misconceptions.

To address these challenges, we propose Fin-Agent-QG, a cognitive-collaborative multi-agent framework that emulates the workflow of professional financial exam committees. Fin-Agent-QG coordinates specialized agents for different cognitive phases of question construction: a Teacher–Critic loop for conceptual and difficulty alignment, a Quantitative Analyst Agent utilizing Code-as-Chain-of-Thought for computational reasoning, and a Knowledge Graph Construction Agent with a Case Strategist for multi-concept integration via a Financial Knowledge Graph.

Empirical evaluations by both LLM metrics and human experts demonstrate that Fin-Agent-QG produces exam-ready financial questions with greater cognitive depth, accuracy, and conceptual coherence than traditional AQG systems. This work bridges the gap between automated generation and expert-level financial assessment design.

1 Introduction

Automated Question Generation (AQG) plays a central role in intelligent education systems, enabling scalable assessment and personalized learning. In professional finance certification programs such as the Chartered Financial Analyst (CFA) and Financial Risk Manager (FRM), question quality directly determines the validity of learning assessment. These examinations assess advanced competencies in investment management and financial risk management, requiring precise difficulty control, rigorous quantitative reasoning, and integration across multiple financial concepts. However, existing AQG systems remain inadequate for meeting the demands of such professional examinations (Kurdi et al. 2019).

Early work on Automated Question Generation primarily formulated the task as question generation given a textual context and an optional known answer. Most of the approaches relied on sequence-to-sequence neural models that rewrote input sentences as questions. This research mainly focused on general reading comprehension settings and was evaluated on datasets such as SQuAD and NewsQA (Zhou et al. 2017). Subsequent studies introduced mechanisms for controlling question attributes such as relevance and difficulty, particularly in educational and reading comprehension settings. However, these approaches are typically grounded in text-centric contexts and short-answer supervision, and do not explicitly model the domain knowledge structures or quantitative reasoning processes required in professional financial assessments. As a result, their applicability to high-stakes, calculation-intensive examinations remains limited.

The emergence of LLMs—such as GPT-4, Claude, and Gemini—has shifted AQG from supervised learning to instruction-following generation. LLMs are capable of consistently generating clear and coherent questions (Biancini, Ferrato, and Limongelli 2024), and demonstrate impressive generalization across domains. However, LLMs struggle to produce cognitively demanding, exam-ready content. Direct prompting often generates questions that are factually correct but cognitively shallow, making them ineffective for evaluating the analytical and reasoning skills of students (Scaria, Chenna, and Subramani 2024).

Recent studies have explored self-refinement and multi-agent paradigms, where multiple LLMs collaboratively generate, critique, and revise assessment content. These frameworks attempt to emulate the cognitive workflow of professional exam committees, which decompose the question design into structured phases: defining learning objectives, determining Bloom-level targets, selecting contexts, constructing distractors, and conducting final alignment reviews (Chen et al. 2023). By distributing cognitive responsibilities across specialized agents, such approaches aim to evaluate and optimize item quality from complementary pedagogical perspectives.

While this strategy improves overall coherence and quality assurance, it does not fully resolve the cognitive ceiling inherent in LLM-based question generation. According to Bloom's taxonomy, cognitive skills can be hierarchically categorized into six levels: Remembering, Understand-

ing, Applying, Analyzing, Evaluating, and Creating (Bloom et al. 1956). Empirical evidence shows that LLMs excel at generating low Bloom's-level concept questions but struggle with high-cognitive questions (Scaria, Dharani Chenna, and Subramani 2024). In finance exams that require numerical computation or multi-concept case analysis, even the strongest LLMs exhibit cognitive ceiling effects. Overcoming this limitation requires tools and specialized agents with enhanced reasoning capabilities.

To bridge this gap, we propose Fin-Agent-QG, a cognitive–collaborative multi-agent framework that orchestrates multiple specialized agents to enhance the reasoning and knowledge integration required for higher-order cognitive question generation. The framework integrates multiple specialized agents to generate high-quality financial case problems: (1) a Teacher–Critic loop iteratively refines conceptual relevance, difficulty, and distractor design; (2) a Quantitative Analyst Agent models correct solution paths and common error patterns using Code-as-Chain-of-Thought reasoning to ensure deterministic calculations and realistic distractors; and (3) a Knowledge Graph Construction Agent with a Case Strategist dynamically builds and navigates a Financial Knowledge Graph (FKG), integrating cross-topic concepts to generate coherent and contextually rich case scenarios.

Through this collaboration, Fin-Agent-QG synthesizes linguistically fluent, cognitively deep, and pedagogically aligned financial exam questions. Empirical evaluations by both LLM metrics and human experts show that our framework significantly improves the quality and cognitive fidelity of the question, addressing long-standing limitations in AQG for professional financial examinations. Our key contributions are as follows:

- We identify and empirically characterize the cognitive ceiling in LLM-based AQG systems applied to financial examinations.
- We introduce Fin-Agent-QG, a novel multi-agent architecture that decomposes question generation into distinct cognitive sub-processes, integrating specialized agents for numerical reasoning and contextual case study problems.
- We demonstrate, through quantitative and qualitative evaluations, that Fin-Agent-QG produces expert-level exam items with superior reasoning depth, accuracy, and conceptual integration.

2 Related Work

2.1 Neural Question Generation

Pioneering work in Neural Question Generation (NQG) primarily utilized sequence-to-sequence (Seq2Seq) architectures, which map an input text to an interrogative sequence. These models, often based on Recurrent Neural Networks (RNNs) with attention mechanisms, were trained to minimize cross-entropy loss on large-scale datasets where context-question pairs were available (Du, Shao, and Cardie 2017). While effective at capturing syntactic transformations, this paradigm treated question generation as a monolithic text-to-text task, lacking explicit control over semantic or pedagogical properties. To introduce a degree of control, subsequent methods incorporated answer information, for example, by using pointers or special tokens to guide the decoder in generating questions relevant to a specific answer span(Song et al. 2018). Despite these enhancements, the core limitation remained: the models' performance was fundamentally tied to the quality and scope of the supervised training data, which typically consisted of general-domain corpora like SQuAD. As surveyed by Pan et al. (2019), this dependency on datasets lacking domain-specific nuances and complex reasoning patterns made NQG methods ill-suited for specialized fields such as finance (Pan et al. 2019).

2.2 LLM-based Question Generation

The advent of Large Language Models (LLMs) instigated a paradigm shift from supervised fine-tuning to in-context learning via the following of instructions (Ouyang et al. 2022). Initial applications demonstrated that models such as GPT-2 could perform zero-shot question generation. This was achieved by leveraging the pre-trained knowledge of the models to formulate relevant questions without task-specific training (Radford et al. 2019). However, this direct prompting approach offers limited control over the cognitive complexity and pedagogical quality of the output. To address this, more advanced techniques have been proposed. One prominent method is iterative self-refinement, in which an LLM is prompted to critique and revise its own generated questions based on a set of predefined quality criteria (Madaan et al. 2023). While this improves linguistic quality, it does not inherently impose a structured, pedagogically-grounded generation process. A more structured approach involves multi-agent frameworks, which decompose the QG task into distinct functional roles. For example, systems such as AgentFarm simulate a collaborative workflow with "generator", "evaluator", and "refiner" agents, each governed by specific instructions to collectively enhance the quality of the questions(Chen et al. 2023). Our work builds upon this multi-agent concept but specializes the agents for the distinct reasoning challenges in finance.

2.3 Enhancing LLM Reasoning

To overcome the inherent limitations of LLMs in tasks that require high-level cognition as categorized by Bloom's Taxonomy (Bloom et al. 1956), research has focused on augmenting them with structured reasoning tools. For quantitative tasks, which are prevalent in finance, the Program-Aided Language Models (PAL) approach has proven effective (Gao et al. 2023). This technique prompts the LLM to generate code (e.g., Python script) as an intermediate reasoning step. By offloading the computational logic to a deterministic code interpreter, PAL ensures numerical accuracy and makes the reasoning process transparent and verifiable. For tasks requiring complex conceptual integration, such as financial case studies, another line of research integrates LLMs with Knowledge Graphs (KGs). Recent surveys have shown that KGs provide structured and symbolic representations of entities and their relationships. When augmenting an LLM's generation process with a KG, it effectively grounds the

generation process, mitigating hallucinations and enabling multi-hop reasoning across disparate concepts (Pan et al. 2024). Fin-Agent-QG synergistically combines both code-based reasoning and KG-guided context generation, creating specialized agents that address these two distinct modes of high-cognitive financial reasoning.

3 Methodology

This section details the architecture of Fin-Agent-QG, a cognitive-collaborative framework designed to emulate the rigorous, multi-stage workflow of human financial exam committees. We first formalize the problem of cognitively-grounded question generation and then present the design of our specialized generation agents, each tailored to a distinct category of assessment items. The overall system architecture is illustrated in Figure 1.

3.1 Problem Formulation

The primary objective is to generate a **pedagogically robust assessment item** Q—consisting of a question stem q, a correct answer a_{correct}, and a set of plausible distractors $\{a_{\text{distractor},i}\}_{i=1}^{k}$ (Biancini, Ferrato, and Limongelli 2024). The generation process is conditioned on a specific financial concept c and a set of pedagogical constraints $\mathcal{P} = \{b, t\}$, where b denotes the target cognitive depth on *Bloom's Taxonomy* (e.g., *Apply*, *Analyze*) and t specifies the question type (e.g., *conceptual*, *computational*, *case-based*).

The core challenge is to accurately model the conditional probability distribution $p(Q \mid c, \mathcal{P})$, ensuring that the generated Q is factually correct and pedagogically effective. Existing approaches using monolithic LLMs typically treat this as a single-pass text generation task. Such methods often fail to fully capture the structure and constraints within $\mathcal{P}$, resulting in assessments that are cognitively shallow and/or flawed. Our framework overcomes this limitation by decomposing the task into specialized, structured generation processes, each designed to maintain alignment with the desired cognitive and pedagogical objectives.

3.2 Conceptual Question Generator

For **Conceptual Question Generation**, the workflow is organized as a two-agent iterative dialog between a *Teacher* and a *Critic*. The Teacher functions as a professional financial content generator: given a concept c and a Bloom's Taxonomy level b, it produces a single, grounded multiple-choice question (MCQ) including *question*, *options*, *correctAnswer*, *hints*, *explanation*. Teacher outputs must: (1) be explicitly anchored to retrieved concept material or few-shot examples; (2) provide exactly three mutually exclusive options with an exact string match for *correctAnswer*; (3) include 1–3 progressive, scaffolded hints; and (4) supply option-level explanations.

The Critic acts as an objective evaluator that inspects grounding, Bloom alignment, clarity, option quality, hint progression and explanation completeness, and then issues a keep/revise decision together with concise, actionable feedback and integer scores for relevance, importance, clarity, difficulty matching and answerability. During each feedback round, the Teacher revises the MCQ according to the Critic's guidance, enabling iterative improvement of the factual fidelity and pedagogical fit for the target concept c and Bloom level b.

Algorithm 1: Iterative Refinement Process for Conceptual Question Generation

Input: Financial concept c, pedagogical constraints $\mathcal{P}$
Output: Pedagogically grounded MCQ Q^*

Initialize Teacher Agent T and Critic Agent C.
Retrieve domain-relevant materials $\mathcal{D}_c$ and few-shot examples $\mathcal{E}_c$.
$t \leftarrow 0$.
repeat
  $Q_t \leftarrow T(c, b, \mathcal{P}, \mathcal{D}_c, \mathcal{E}_c)$ {Teacher generates grounded MCQ including *question*, *options*, *correctAnswer*, *hints* and *explanation*}
  $F_t \leftarrow C(Q_t)$ {Critic evaluates along grounding, clarity, Bloom alignment, and pedagogical quality}
  if F_t.decision = "keep" **then**
    $Q^* \leftarrow Q_t$
    break
  else
    $T \leftarrow T$.revise(F_t) {Teacher refines question using structured feedback}
  end if
  $t \leftarrow t + 1$
until $t \geq T_{\max}$ or convergence criterion met
return Q^*

Formally, the generation process can be viewed as an iterative refinement of a conditional distribution:

$$Q^{(t+1)} = f_{\text{teacher}}\left(c, \mathcal{P}, \text{feedback}_{\text{critic}}^{(t)}\right),$$

where the teacher updates the question Q at round $t+1$ based on critic feedback $\text{feedback}_{\text{critic}}^{(t)}$, and $\mathcal{P} = \{b, t\}$ denotes the pedagogical constraints. Experimentally, we varied the number of interaction rounds $T \in [2, 5]$ to examine the convergence behavior of the refinement process. We formalized the interaction between the teacher and critic agents as an iterative refinement process, as shown in Algorithm 1. This process repeats for T rounds, allowing the model to converge towards a higher-quality, pedagogically aligned assessment items.

3.3 Calculation Question Generator

In the Calculation Question Generator, we adopt a schema- and execution-constrained design, in which reasoning chains are formalized as executable code to ensure high numerical determinism in financial calculation problems. This approach not only mitigates common drift or errors in arithmetic reasoning by large language models but also ensures that each computational step is traceable and verifiable. Each problem includes: (i) a fully specified natural language question stem, such as cash flow discounting, interest rate conversion, or portfolio weight calculation; (ii) four fenced Python code blocks, where Code 1 implements the cor-

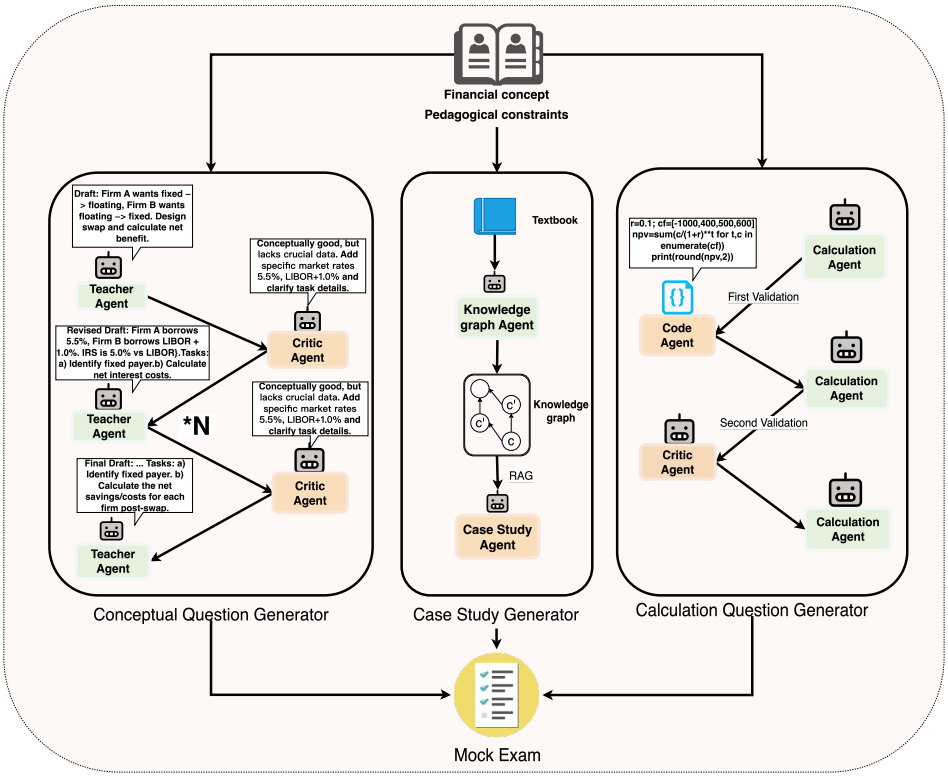

Figure 1: Overall architecture of the proposed Fin-Agent-QG framework, illustrating the collaborative workflow among specialized agents.

rect solution and reflects the reasoning chain, and Codes 2–4 instantiate typical error patterns (e.g., misapplied compounding/discounting, percentage unit confusion, or incorrect weight summation); and (iii) optional hints and explanation sections. With this design, each option's numerical value can be directly obtained via code execution, preserving transparency and verifiability.

The generation pipeline, reflecting programmatic reasoning principles, consists of four main stages:

1. **Retrieval-augmented conditioning**: Relevant passages are fetched from a financial concept store to provide consistent terminology, numeric ranges, and regulatory or ethical context, reducing drift. For example, cash flow discounting problems retrieve discount rate definitions, while interest rate conversion problems fetch nominal vs. effective rate definitions.
2. **Controlled prompting**: Bloom parameters control reasoning depth and computational complexity, while prompts enforce question structure and output constraints. Each Python block becomes an executable reasoning chain—for instance, portfolio allocation problems demonstrate full computation or simulate common calculation errors.
3. **Structured decoding and execution**: Outputs are mapped to a schema-constrained representation, with each code block executed independently to produce deterministic numerical options. Externalizing reasoning as code ensures precision and traceability.
4. **Validation and normalization**: Option counts and labels are enforced, answer–explanation alignment checked, and numeric plausibility verified (e.g., discount ranges, portfolio weights summing to 1). Minor issues are corrected deterministically; major violations trigger constrained regeneration.

Difficulty is controlled via parameter ranges, computational steps, and distractor types. *Apply*-level problems focus on single-formula calculations, while *Analyze*-level problems involve multi-step pipelines (rate conversion $\rightarrow$ cash flow transformation $\rightarrow$ discounting) with contrastive distractors targeting common misconceptions.

This design produces financial calculation problems that are structurally sound, cognitively calibrated, and numerically deterministic. Encoding reasoning chains as executable code preserves educational value while ensuring reliability and verifiability.

3.4 Case Study Generator

Conditioned on a target concept c and constraint set $\mathcal{P} = \{b,\ t = \text{case-based}\}$, the case study module generates a scenario-driven assessment Q^{case} under explicit structural and cognitive controls. The output is a structured object $Q^{\text{case}} = \big(S, \{(q_i, \mathcal{A}_i, a_{\text{correct},i}, E_i)\}_{i=1}^{4}\big)$, where S is a self-contained scenario, q_i are four scenario-based questions, $\mathcal{A}_i = \{a_{i,A}, a_{i,B}, a_{i,C}, a_{i,D}\}$ are option sets with canoni-

cal A/B/C/D labeling, $a_{\text{correct},i} \in \{A, B, C, D\}$ are correct answers, and E_i are concise evidence-based explanations.

We adopt a knowledge-graph-driven approach to select the most coherent and relevant set of concepts for each case study. Initially, textbook content is parsed into a structured Financial Knowledge Graph (FKG) containing entities and relationships, a process performed by a specialized **Knowledge Graph Construction Agent**. Building upon this graph, a **Case Strategist Agent** dynamically traverses the graph to combine multiple related knowledge nodes, synthesizing a coherent, integrated scenario that aligns with realistic financial contexts.

Based on the resulting subgraph, agents generate a self-contained scenario and a series of related conceptual and computational questions, forming the core of the scenario-driven assessment. This design ensures that the generated case study is conceptually grounded, cognitively structured, and faithfully reflects the domain knowledge encoded in the FKG.

Compared to simple concepts-based generation, this knowledge-graph-driven approach ensures that case study items are coherent and diverse, reflecting both pedagogical and assessment objectives in financial education.

4 Experiments

4.1 Data

To provide domain-grounded knowledge for our multi-agent question generation framework, we construct a comprehensive corpus based on official Chartered Financial Analyst (CFA) materials. Specifically, we parse a total of twenty textbooks covering both Level I and Level II CFA curricula (ten volumes each). The textbooks serve as the primary knowledge base from which financial concepts and their contextual content are derived.

We leverage the hierarchical bookmark structure embedded in the PDF documents to extract the title organization of each textbook. Each textbook is represented as a hierarchical tree, where the first-level nodes correspond to *chapters*, the second-level nodes to *subsections*, and, in some cases, the third-level nodes to *subsubsections*. Formally, this structure can be expressed as a directory tree $\mathcal{T} = \{\text{chapter} \rightarrow \text{subsection} \rightarrow \text{subsubsection}\}$. We define the set of *concepts* $\mathcal{C}$ as the set of all leaf nodes in $\mathcal{T}$. In other words, if a subsection contains subsubsections, each subsubsection is treated as an independent concept; otherwise, the subsection itself becomes a concept. This design ensures that each concept represents the smallest pedagogically meaningful unit in the textbook hierarchy.

For each concept $c \in \mathcal{C}$, we parse the PDF metadata to obtain its starting and ending page numbers, and extract the corresponding text span as its content segment. The resulting parsing yields 1,875 concepts for Level I and 936 concepts for Level II. These content segments serve as the domain knowledge foundation fed into the Fin-Agent-QG framework, enabling the Teacher and Critic agents to operate over grounded, contextually relevant materials rather than unstructured text.

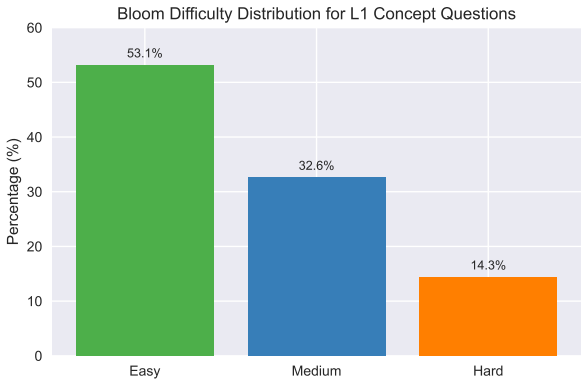

Figure 2: Bloom difficulty distribution for Level I concept questions.

In addition to the textbooks, we collect five sets of mock examination papers for both Level I and Level II. The original mock questions contain the fields *question_text*, *options*, *answer*, and *topic*. Based on these, we further annotate *concept* and *bloom difficulty* to enrich the dataset. These mock questions are used as few-shot examples to calibrate difficulty estimation, align generated questions with real exam styles, and support the evaluation of question quality under authentic testing conditions. The detailed statistics of these materials are summarized in Table 1. Figure 2 shows the Bloom difficulty distribution for Level I concept questions.

4.2 Evaluation Metrics

Evaluating Automatic Question Generation (AQG) systems presents unique challenges, particularly due to the lack of standardized benchmark datasets and the inherent subjectivity of human assessments. While traditional Natural Question Generation (NQG) metrics, such as BLEU and ROUGE, provide automated evaluation based on n-gram overlap and text similarity, these measures primarily capture surface-level textual correspondence. They fail to account for pedagogical qualities critical for high-quality professional examination items, including coherence, conceptual relevance, and precise alignment with pre-specified difficulty levels.

Recent studies have highlighted these limitations and proposed alternative evaluation frameworks, such as G-Eval, which leverage large language models (e.g., GPT-4) as reference-free evaluators for tasks like summarization, demonstrating higher alignment with human judgments (Liu et al. 2023). However, some literature also indicates that LLM-based evaluation methods may still exhibit bias and do not always perfectly correlate with expert human scoring across broader domains (Stureborg, Alikaniotis, and Suhara 2024).

In this study, our goal is to generate high-quality professional test items that require pedagogical rigor; therefore, traditional metrics such as BLEU are insufficient. Following established practices, we adopt a prompt-engineering approach to evaluate the quality of each generated item across multiple dimensions, assigning score from 1 (severely

Level	Question Count	Type Distribution	Major Topics (Top 3)	Top Concepts (Top 3)
L1	900	Calculation: 178 Concept: 722 Case: 0	ETH: 135 FSA: 110 FI: 110	Screening for Potential Equity Investments (51) Projecting Future Financial Performance (20) Aggregate Demand and Aggregate Supply (17)
L2	110	Calculation: 0 Concept: 0 Case: 110	ETH: 15 FSA: 15 FI: 14	–
Total	**1010**	Calculation: 178 Concept: 722 Case: 110	ETH: 150 FSA: 125 FI: 124	–

Table 1: Summary of CFA question dataset used for domain knowledge and evaluation.

flawed) to 10 (excellent, nearly flawless) based on the following criteria (Karbasi et al. 2025).

- **Relevance:** Measures whether the question aligns with the provided content and specified topic.
- **Importance:** Assesses whether the question tests a key concept emphasized in the source material.
- **Clarity:** Evaluates if the question is unambiguous, concise, and easy to understand.
- **Difficulty Matching:** Determines whether the question appropriately corresponds to the target Bloom's taxonomy level.
- **Answerability:** Checks if the learner can answer the question using only the information provided in the question and options.

To examine the alignment between our scoring approach and expert judgment, we additionally asked finance professionals to rate each item along the same core dimensions. For numerical questions, experts provided supplementary scores for *Numerical Soundness* and *Distractor Plausibility*, while for case study questions, additional ratings were collected for *Case Coherence* and *Case Diversity* to assess whether the scenarios are coherent, contextually integrated and consistent with real-world financial practice.

4.3 Experimental Design

To evaluate the effectiveness of the proposed Fin-Agent-QG framework, we construct a controlled experiment using a curated set of mock questions that reflect the typical composition of a CFA-style examination. Specifically, we select 50 concept-based questions, 25 numerical (calculation) questions, and 25 case study questions as the ground truth, maintaining a ratio that closely mirrors real-world CFA exam distributions. For each concept question, we extract the associated concepts, relevant content, and Bloom difficulty levels as agent inputs. This setup allows direct comparison with both the ground truth and the authentic testing environment. The total number of generated questions across all experiments includes 200 concept questions (3 baselines + our method, each generating 50), 50 numerical questions (for one ablation study), and 50 case study questions (for another ablation study).

The core experiments include the following:

Experiment 1: Concept Question Generation. This experiment compares Fin-Agent-QG with three baselines to assess overall question quality. Baseline 1 (Zero-Shot) employs GPT with only the teacher agent prompt and content, without few-shot examples. Baseline 2 (Few-Shot) uses GPT with the teacher agent prompt and content, incorporating few-shot examples to guide generation. Baseline 3 (Single Expert Agent) adopts a single complex prompt that embeds both the teacher and critic agent logic within one agent. Each method, along with Fin-Agent-QG, generates corresponding concept questions, and results are compared with the original human-authored items. Evaluation includes both automated and human assessments. For automated evaluation, an LLM serves as the evaluator, scoring each question on five dimensions—Relevance, Clarity, Importance, Difficulty Matching, and Answerability—on a scale from 1 to 10. For human evaluation, one to two finance experts rate a subset of 30 questions using the same criteria, enabling agreement analysis between expert and LLM evaluations.

Experiment 2: Ablation Study on Quantitative Analyst Agent. This experiment examines the necessity of the "code-as-chain-of-thought" design for generating numerical questions. We compare two configurations: Fin-Agent-QG with the Quantitative Analyst agent versus Fin-Agent-QG without it, where a conventional teacher agent with numerical question prompts is used instead. Performance is evaluated in terms of Numerical Soundness, Distractor Plausibility, and Overall Average Score. Given the potential limitations of current large language models in reliably judging high-level cognitive tasks, we rely solely on human experts to provide these scores as the definitive gold standard.

Experiment 3: Ablation Study on Case Study Agents. To validate the advantage of the KG-driven Case Synthesis approach over a conventional teacher-driven baseline, we conduct an ablation study on case study question generation. We compare two versions of Fin-Agent-QG: the full model (including the Knowledge Graph agent and Case Strategist) and a reduced variant in which both agents are removed and replaced by a standard Teacher Agent that generates cases solely from a list of concepts. Evaluation introduces two additional dimensions—Case Coherence and Diversity—to assess whether generated cases are logically consistent, contextually integrated, and aligned with realistic financial scenarios; human experts provide scores for these dimensions.

Generation Method	Evaluator	Relevance	Clarity	Importance	Difficulty Matching	Answerability	Overall Average
Baseline 1 (Zero-shot)	Human Expert	8.53	8.47	8.27	7.33	9.13	8.35
Baseline 1 (Zero-shot)	LLM	8.88	8.52	9.02	7.85	9.42	8.74
Baseline 2 (Few-shot)	Human Expert	8.60	8.33	8.40	7.60	9.07	8.40
Baseline 2 (Few-shot)	LLM	9.04	8.29	8.81	7.52	9.21	8.57
Baseline 3 (Single Expert)	Human Expert	8.73	8.67	8.67	7.8	9.27	8.63
Baseline 3 (Single Expert)	LLM	9.50	8.62	9.42	7.94	9.65	9.02
Fin-Agent-QG (Our Method)	Human Expert	**8.93**	**8.93**	**8.87**	**7.93**	**9.33**	**8.80**
Fin-Agent-QG (Our Method)	LLM	**9.52**	**8.71**	**9.52**	**8.17**	**9.77**	**9.14**
Ground Truth	Human Expert	8.60	8.53	8.60	**7.93**	9.13	8.56
Ground Truth	LLM	8.56	7.88	8.33	7.0	8.54	8.06

Table 2: Comparison of different question generation methods. Best LLM scores and best Human Expert scores are in **bold**.

Model Configuration	Evaluator	Numerical Soundness	Distractor Plausibility	Overall Average
Fin-Agent-QG (w/o Quantitative Analyst)	Human Expert	4.90	4.44	4.45
Fin-Agent-QG (Full w/ Quantitative Analyst)	Human Expert	**10.00**	**6.60**	**8.30**
Ground Truth	Human Expert	**10.00**	5.70	7.85

Table 3: Ablation study on the Quantitative Analyst Agent. Best Human Expert scores are in **bold**.

Model Configuration	Evaluator	Case Coherence	Case Diversity	Overall Average
Fin-Agent-QG (w/o KG & Case Strategist)	Human Expert	6.90	6.80	6.85
Fin-Agent-QG (Full w/ KG & Case Strategist)	Human Expert	8.60	7.80	8.20
Ground Truth	Human Expert	**8.70**	**8.90**	**8.80**

Table 4: Ablation study on the Case Study Agents. Best Human Expert scores are in **bold**.

4.4 Results and Discussion

Experiment 1: Concept Question Generation Table 2 shows that Fin-Agent-QG outperforms all baseline models in concept question generation, achieving the highest overall LLM scores and demonstrating superior coherence, pedagogical relevance, and Bloom-aligned difficulty. Baseline comparisons indicate that single expert agents improve over zero- and few-shot teacher-only models, highlighting the importance of self-reflection prompt design. However, in comparison, decomposing the generation into specialized agents and executing multiple rounds of self-refinement provides further improvements in question quality.

For model-generated questions, human and LLM evaluations are generally consistent, reflecting reasonable agreement between automated metrics and expert judgment. Interestingly, LLMs tend to assign lower scores to ground truth questions compared to humans, who rate them higher than zero- and few-shot baselines. This gap likely arises because LLM evaluations focus on surface-level criteria, such as Bloom-level difficulty, without fully capturing deeper knowledge connections and reasoning requirements.

Experiment 2: Ablation Study on the Quantitative Analyst Agent Table 3 shows the results of an ablation study evaluating the effect of the Quantitative Analyst (QA) agent on question quality. When the QA agent is removed, the overall human expert scores drop significantly. In contrast, the full Fin-Agent-QG system with the QA agent achieves the highest scores, surpassing even the ground truth questions in overall quality.

These results demonstrate that the QA agent plays a crucial role in ensuring numerical correctness and generating plausible distractors. Without this agent, questions may still be linguistically coherent but lack quantitative rigor, highlighting the importance of task-specific specialization within the multi-agent framework.

Experiment 3: Ablation Study on the Case Study Agent Table 4 presents the results of an ablation study evaluating the impact of the Knowledge Graph and Case Strategist Agent, on the quality of generated case study questions. When both KG and Case Strategist are removed, the overall human expert score drops significantly. In contrast, the full Fin-Agent-QG system that incorporates both modules achieves substantially higher scores, reflecting notable improvements in both the logical coherence of cases and the diversity of scenarios.

Unlike the results observed for concept and quantitative questions, the ground truth case studies achieve the highest scores in case study questions. This proves the cognitive ceiling effect for LLM-generated questions. Nonetheless, our multi-agent approach with KG and Case Strategist significantly narrows the gap, demonstrating that structured guidance and task-specific reasoning can markedly enhance the quality and pedagogical value of generated case studies.

5 Conclusion

We present Fin-Agent-QG, a cognitive-collaborative multi-agent framework that overcomes the limitations of existing LLM-based AQG systems in higher-order cognitive financial questions. By coordinating Teacher–Critic, Quantitative Analyst, and Knowledge Graph Construction agents, Fin-Agent-QG generates exam-ready questions with enhanced cognitive depth, computational correctness, and multi-concept integration. Empirical evaluations show significant improvements in accuracy, coherence, and pedagogical quality, bridging the gap between automated generation and expert-level financial assessment.

References

Biancini, G.; Ferrato, A.; and Limongelli, C. 2024. Multiple-Choice Question Generation Using Large Language Models: Methodology and Educator Insights. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '24, 584–590. ACM.

Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; and Krathwohl, D. R. 1956. *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain.* New York: David McKay Company.

Chen, J.; Bai, J.; Chen, S.; Yang, C.; Zhang, R.; Liu, S.; Wang, Z.; and Zhang, R. 2023. AgentFarm: A General Framework for Information-Seeking Agent. *arXiv preprint arXiv:2309.04000*.

Du, X.; Shao, J.; and Cardie, C. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1342–1352. Vancouver, Canada: Association for Computational Linguistics.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. PAL: Program-aided Language Models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, 10764–10799. PMLR.

Karbasi, K.; Hong, K.; Samadi, M. A.; and Pottie, G. 2025. Multi-Agent Collaborative Framework for Math Problem Generation. In *Proceedings of the 18th International Conference on Educational Data Mining (EDM 2025)*, 288–295. Educational Data Mining Society.

Kurdi, G.; Leo, J.; Parsia, B.; Sattler, U.; and Al-Emari, S. 2019. A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education*, 30: 121–204.

Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; and Zhu, C. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, K.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Gupta, M.; Sasno, M.; Shen, J.; Misra, Y.; Lanchantin, J.; Sontag, D.; Singh, S.; Clark, P.; Yazdanbakhsh, A.; and Majumder, B. P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *Advances in Neural Information Processing Systems*, 36: 66164–66188.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Pan, L.; Zhao, J.-M.; Cai, D.; Zhou, B.; Wang, X.; and Zhang, M. 2019. Recent advances on neural question generation. *arXiv preprint arXiv:1905.08949*.

Pan, S.; Chen, L.; Wang, Y.; Zhang, C.; Wang, X.; He, L.; Luo, X.; Chen, X.; Li, Y.; Wang, Y.; Chen, Y.; and Dai, C. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv preprint arXiv:2306.08302*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8).

Scaria, N.; Chenna, S. D.; and Subramani, D. 2024. Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation. *arXiv preprint arXiv:2408.04394*.

Scaria, N.; Dharani Chenna, S.; and Subramani, D. 2024. *Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation*, 165–179. Springer Nature Switzerland. ISBN 9783031642999.

Song, L.; Wang, Z.; Hamza, W.; Zhang, Y.; and Gildea, D. 2018. Leveraging Context Information for Natural Question Generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 564–569. New Orleans, Louisiana: Association for Computational Linguistics.

Stureborg, R.; Alikaniotis, D.; and Suhara, Y. 2024. Large Language Models are Inconsistent and Biased Evaluators. *arXiv preprint arXiv:2405.01724*.

Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; and Zhou, M. 2017. Neural Question Generation from Text: A Preliminary Study. arXiv:1704.01792.