UniTor at PFB: Constrained Agentic Prompting and Self-Consistency for Financial Multi-Choice QA

Federico Borazio[1], Seyed Alireza Mousavian Anaraki[1], Shahid Iqbal Rai[1], Danilo Croce[1] and Roberto Basili[1]

[1]*Department of Enterprise Engineering, University of Rome Tor Vergata*
Via del Politecnico 1, 00133, Rome, Italy

Abstract

General-purpose Large Language Models (LLMs) often struggle to combine domain specificity with robust cross-lingual generalization, particularly in high-stakes domains such as finance. In this paper, we present our contribution to the PFB: Prometeia Financial Benchmark, tackling Financial Multi-Choice Question Answering (MCQA) in English, Italian, and Turkish. We introduce a constrained agentic workflow that improves reliability exclusively through inference-time control, without fine-tuning or external tools. Our system implements a hierarchical pipeline based on semantic routing to specialized reasoning archetypes (Analyst, Fact-Checker, Researcher), parallel stochastic sampling to expose reasoning instability, and a deterministic algorithmic consensus mechanism that promotes answers stable under resampling while suppressing hallucinations. Extensive evaluation on the blind test set shows that this architecture consistently closes the cross-lingual performance gap, enabling a mid-sized open model (GPT-OSS-20B) to reach state-of-the-art accuracy ($\approx$88%) across all three languages. Remarkably, our approach matches or slightly outperforms much larger proprietary models, while remaining robust in low-resource settings such as Turkish. Ablation results demonstrate that stochastic self-consistency is essential for handling structural complexity, particularly tabular and boolean reasoning, and that controlled aggregation provides a scalable, training-free alternative to model scaling and domain-specific fine-tuning.

Keywords

Large Language Models, Agentic Workflows, Ensemble Inference, Financial NLP

1. Introduction

In recent years, Natural Language Processing has witnessed an unprecedented paradigm shift driven by the advent of Large Language Models (LLMs). While general-purpose models have demonstrated impressive performance across a wide range of tasks, from logical reasoning [1] to creative generation [2], their effectiveness in highly specialized domains, such as finance, remains an open question [3, 4, 5]. The financial domain presents unique challenges that go beyond standard linguistic comprehension. It requires the ability to interpret complex technical terminology, grasp subtle regulatory nuances, and perform rigorous numerical reasoning over structured data [6]. While recent literature has begun to explore the adaptation of LLMs to financial language, the vast majority of studies focus on English, leaving other languages, such as Italian, significantly underrepresented. This creates a critical gap: although multilingual models promise universality, it is unclear whether they can compete with specialized baselines [7] in vertical sectors where local market specificities play a crucial role.

In this context, the present work outlines our contribution to the **PFB: Prometeia Financial Benchmark**[1], a shared task organized within EVALITA 2026 [8] specifically focused on Financial Multi-Choice Question Answering (MCQA). This benchmark serves as a stress test for model capabilities in a constrained, real-world scenario, offering a high-quality multilingual dataset (Italian, English,

[1]https://www.prometeia.com/it/about-us/insights/article/prometeia-financial-benchmark-benchmarking-language-models-in-the-financial-domain-26892170

Turkish) annotated by complexity. The task requires systems to select the correct option among a fixed set of candidates for each question. Each instance consists of a natural language question, five answer options (A–E), and a single correct label, and is evaluated using accuracy-based metrics stratified by difficulty level. Unlike generic QA datasets, the PFB task challenges models with heterogeneous reasoning requirements. Financial questions exhibit strong structural heterogeneity, ranging from conceptual queries requiring domain knowledge, to boolean verification tasks and complex tabular reasoning scenarios demanding numerical extraction and calculation.

The design choices of the proposed architecture are motivated by two empirical observations about LLM behavior in specialized MCQA tasks. First, financial questions exhibit strong structural heterogeneity, and no single prompting strategy is uniformly effective across conceptual, boolean, and tabular reasoning. Second, LLM errors in such settings are often *unstable*: hallucinated or incorrect answers tend to vary across stochastic generations, whereas correct solutions are more likely to reappear consistently. Based on this observation and inspired by [9], our approach treats stochastic sampling not as a source of noise, but as a mechanism to probe the stability of candidate answers. By generating multiple reasoning variants for the same query and aggregating them deterministically, the system implicitly favors answers that are robust under resampling, while suppressing spurious or idiosyncratic generations. The iterative refinement step further enforces this principle by constraining the search space when consensus is weak, effectively amplifying stable reasoning patterns and mitigating hallucinations.

We evaluate this architecture across a spectrum of models, from lightweight open-source solutions (e.g., LLaMA 3.1 8B) to large-scale proprietary engines (e.g., DeepSeek v3.1). This comparative analysis allows us to investigate the trade-offs between model size, inference efficiency, and cross-lingual transfer in the financial domain.

Our study is driven by the following core research questions:

- **RQ1:** Can general-purpose LLMs achieve domain-competent performance in specialized financial tasks relying solely on structured reasoning across LLM ensembles?
- **RQ2:** How does the structural complexity of the question (e.g., tabular vs. conceptual, easy vs. hard) impact performance across different languages?
- **RQ3:** Can reasoning patterns and financial knowledge effectively favor the transfer from an English-centric pre-training to the target Italian and Turkish contexts?
- **RQ4:** Do stochastic sampling and hybrid aggregation strategies significantly improve reliability compared to standard greedy decoding?

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on LLMs for reasoning, financial domain adaptation, and agentic workflows. Section 3 details our proposed system architecture, describing the semantic routing mechanism, the specialized reasoning modules, and the aggregation strategy. Section 4 presents the experimental setup and discusses the results, providing a granular analysis of model performance across languages and complexity levels. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related Work

Large language models (LLMs) have achieved significant advances in QA across multiple domains [10, 11]. In the educational context, elimination-based reasoning with LLMs has been applied to MCQA [12], incorporating structured prompting and intermediate decision steps with chain-of-thought (CoT) reasoning to eliminate incorrect options sequentially. In the medical domain, approaches such as reasoning with LLMs [13] and collaboration among multiple LLMs [14] have been explored to enhance accuracy and consistency. These efforts include ensemble reasoning strategies that refine intermediate steps and reduce inconsistencies, demonstrating the potential of LLMs to perform complex, multi-step reasoning in specialized question-answering tasks.

Despite these advances, increasing the size of the model alone does not substantially improve performance in complex reasoning tasks. CoT prompting has therefore been proposed, guiding the

model through intermediate reasoning steps to solve tasks sequentially. Several enhancements have further improved CoT performance [15]. A simple yet effective method is self-consistency [16], which generates multiple reasoning paths and determines the final answer via majority voting [17]. Additional approaches include calibrating reasoning paths with a verifier [18] and using dynamic voting strategies for more efficient reasoning in LLMs [15].

In financial MCQA, the type of questions and the format of the data are particularly important. Tables and structured data are central to many financial tasks, such as analyzing balance sheets or performance reports. While traditional LLMs excel at text-based reasoning, they struggle to interpret data presented in image or table formats. Recent multimodal LLMs have begun to bridge this gap by processing both text and images; however, their performance on table images remains limited and lacks domain specificity. To address this, [19] introduces FinTab-LLaVA, a finance-specific multimodal LLM designed for table understanding using FinTMD.

Intelligent AI agents are becoming a core component of autonomous, goal-oriented systems that can perceive their environment, reason over information, learn from data, and execute actions. In the financial domain in particular, artificial intelligence is fundamentally transforming financial infrastructures and services. Agent-based AI approaches are now widely applied in areas such as algorithmic trading, fraud and anomaly detection, credit scoring and risk evaluation, automated investment advisory services, and regulatory technology (RegTech) for compliance and monitoring [20].

The advent of LLMs has further driven the development of intelligent agents, where the LLM functions as the primary reasoning component or "brain". These models offer strong capabilities in natural language understanding, text generation, reasoning, and knowledge retrieval. Agent architectures typically integrate LLMs with external tools, memory components, and additional processing modules [21]. Such LLM-based agents have also been applied to financial information retrieval and to the augmentation of knowledge-intensive work in finance [22, 23].

While agent capabilities provide an important foundation, the architectural organization of agents is equally critical. In the literature, agent-based systems are commonly structured as multi-agent systems (MAS), hierarchical agents, or hybrid architectures. MAS comprises multiple autonomous agents interacting within a shared environment, supporting collaborative or competitive problem solving through coordination and distributed decision making [24]. Hierarchical architectures address long-horizon tasks by decomposing them into multi-level goals [25], while hybrid systems integrate diverse AI techniques, such as LLM reasoning, retrieval, external tools, and specialized models, within a unified framework to leverage complementary strengths [26]. In the context of financial QA, FinAgentBench introduces the first large-scale benchmark for evaluating retrieval with multi-step reasoning in finance, a setting referred to as agentic retrieval [27].

While these approaches demonstrate the effectiveness of structured prompting, self-consistency, and agent-based reasoning, most prior work focuses either on open-ended QA, retrieval-augmented settings, or domain-specific fine-tuning. In contrast, the Financial MCQA setting considered in this work imposes a strict closed-set decision constraint, where systems must select a single option without access to external tools or additional supervision. Moreover, existing agentic frameworks typically emphasize autonomy, tool usage, or long-horizon planning, whereas our approach adopts a deliberately constrained agentic design tailored to inference-time robustness. By combining task-aware routing, stochastic reasoning, and deterministic aggregation, our work bridges ensemble-based self-consistency methods and agentic workflows in a unified architecture specifically designed to mitigate hallucinations in multilingual financial MCQA.

3. Description of the System

Financial MCQA exposes two recurrent limitations of LLMs. First, financial questions exhibit strong structural heterogeneity: conceptual queries, boolean verification tasks, and tabular reasoning problems require fundamentally different reasoning strategies. Second, model predictions in this setting are often unstable: early reasoning errors or hallucinations in an early inference pass can disproportionately

affect the final answer. As a consequence, a monolithic prompting strategy and single-pass inference are insufficient to ensure robust performance.

To address these issues, we propose a modular inference architecture that decomposes reasoning into specialized stages and consolidates multiple stochastic hypotheses into a single, stable decision. We refer to this design as a **constrained agentic workflow**. The term *agentic* reflects the use of role-based specific stages and explicit orchestration of reasoning behaviors, while *constrained* emphasizes that the system deliberately avoids autonomous tool usage, open-ended planning, memory accumulation, or unbounded interaction loops. All components operate within a fixed and fully controlled inference graph, ensuring reproducibility, stability, and low latency.

Within this framework, each input question is first analyzed and routed to a question-specific reasoning role. The selected role is then instantiated multiple times under stochastic sampling to generate diverse reasoning hypotheses. Rather than treating redundancy as a source of noise, the system exploits it to probe the stochastic stability of candidate answers: correct solutions tend to recur across independent generations, while hallucinated or brittle answers are less consistent. A deterministic aggregation stage subsequently consolidates these hypotheses, favoring answers that are robust under resampling and suppressing spurious outputs.

3.1. Overall Architecture

The proposed system follows a three-stage orchestration pipeline, illustrated in Figure 1, composed of *semantic routing*, *specialized reasoning*, and *hybrid decision aggregation*.
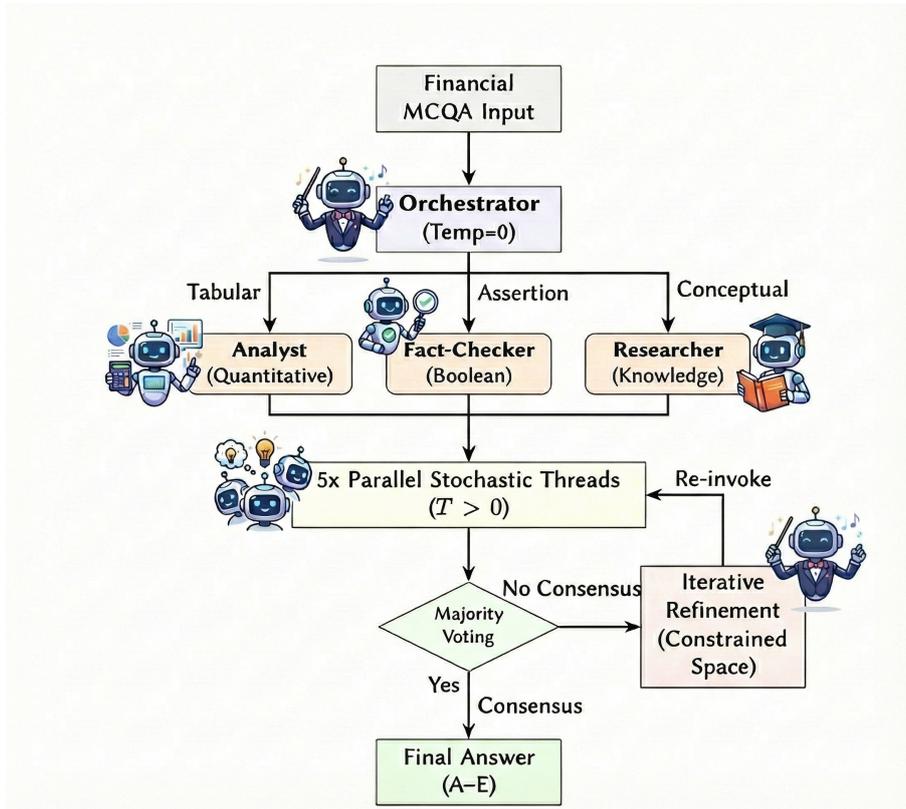
Figure 1: Architectural overview of the proposed constrained agentic workflow. Inference is decomposed into a deterministic routing stage, followed by parallel stochastic reasoning and a controlled aggregation mechanism. The Orchestrator assigns each query to a specialized reasoning archetype. Multiple stochastic generations probe the stability of candidate answers, while hybrid aggregation consolidates robust solutions and suppresses unstable or hallucinated outputs.

The pipeline begins with the **Semantic Routing** stage, implemented by a lightweight LLM-based *Orchestrator*. Its role is to prevent a one-size-fits-all reasoning strategy from being applied to structurally

different questions. Given an input query, the Orchestrator deterministically classifies it into one of three reasoning archetypes: conceptual (*Researcher*), boolean verification (*Fact-Checker*), or tabular reasoning (*Analyst*). Determinism at this stage is critical, as routing errors would systematically propagate to all subsequent reasoning threads.

Following routing, the system enters the **Specialized Reasoning** phase. Instead of producing a single solution, the selected reasoning role is instantiated in multiple parallel inference threads. Each thread operates with non-zero temperature ($T = 0.7$), inducing variability in intermediate reasoning steps while preserving the same task-specific prompt. This design simulates a panel of independent experts analyzing the same problem and enables the system to explore alternative reasoning paths.

Finally, the **Hybrid Aggregation** stage consolidates the outputs of the parallel reasoning threads into a single prediction. Majority voting is used to select the most stable answer, effectively filtering out isolated reasoning errors. When consensus is weak, an iterative refinement mechanism is triggered to constrain the answer space and force convergence. Through this process, stochastic exploration is progressively narrowed into a deterministic decision, improving robustness without introducing open-ended inference loops.

In the remainder of this section, we describe each stage of the workflow in detail, clarifying its role, design rationale, and interaction with the other components.

Routing and Task Decomposition.   The first stage of the pipeline performs *semantic routing*, whose purpose is to associate each input question with a reasoning strategy that is appropriate to its dominant structure. The routing scheme is not derived from an external taxonomy, but is introduced by the authors as a pragmatic abstraction based on an empirical inspection of the PFB development set. Through this analysis, we observed that the vast majority of questions could be naturally grouped according to their primary reasoning requirement. We therefore define three *reasoning archetypes*: (i) *Analyst*, for questions involving numerical reasoning or explicit value extraction from tables or structured financial statements; (ii) *Fact-Checker*, for questions presenting multiple assertions that must be individually validated as true or false; and (iii) *Researcher*, for conceptual or theoretical questions primarily relying on domain knowledge. These archetypes are not intended as exhaustive or universally valid categories, but as a lightweight operational decomposition tailored to the structure of the benchmark. Routing is implemented by a lightweight LLM-based *Orchestrator* operating deterministically ($T = 0$), ensuring stable and reproducible classification decisions. On the PFB development set, this heuristic routing achieved near-perfect agreement with the manually inspected assignment, reflecting the strong regularity of question structures in the benchmark rather than an intrinsic generalization guarantee. To mitigate potential misclassification in less regular or unseen cases, a conservative fallback strategy is employed. When the Orchestrator fails to identify a clear dominant archetype, the system defaults to a generic inference configuration using a neutral prompt that does not enforce any role-specific reasoning constraints. This routing component is implemented through a dedicated prompt template; the full prompt specifications are reported in Appendix A.1.

Specialized Reasoning Modules.   After routing, the selected reasoning archetype is instantiated multiple times to generate a set of independent hypotheses. Each archetype (*Analyst*, *Fact-Checker*, *Researcher*) is implemented through a *distinct, role-specific prompt* that encodes a different reasoning strategy (quantitative extraction, boolean verification, or conceptual synthesis, respectively). Within a given archetype, all parallel instances share the same prompt and inference configuration, but operate under stochastic sampling ($T = 0.7$). As a result, the system produces multiple alternative reasoning traces corresponding to the *same reasoning strategy*, rather than mixing heterogeneous behaviors. This design can be interpreted as a panel of independent virtual experts operating under the same methodological guidelines (i.e., the specific prompt constraints), each providing a separate judgment on the same question. Correct answers tend to emerge consistently across repeated samples, while hallucinated or fragile conclusions are less stable. By explicitly repeating the same role-conditioned reasoning process, the system enables downstream aggregation to assess answer robustness under

resampling, rather than relying on a single reasoning trace. All reasoning modules terminate generation via EOS detection and are constrained to output a single option label (A–E), ensuring consistent parsing and reliable aggregation. The full prompt templates defining each reasoning archetype are reported in Appendix A.1.

Aggregation Policies. The final decision is obtained through a deterministic aggregation of the $N = 5$ stochastic reasoning outputs produced for each question. Each reasoning instance returns a single option label (A–E), and these labels are interpreted as independent *votes*. We define *consensus* as the emergence of a dominant option that is selected by a strict majority of the instances. Accordingly, the system applies **Majority Voting**: the option receiving the highest number of votes is selected as the final answer. This mechanism operationalizes self-consistency by favoring answers that remain stable across resampled reasoning paths, while suppressing isolated hallucinations that occur sporadically in individual generations. A special handling is introduced for the option *"None of the above"* (Option E). Empirical inspection revealed a systematic *negative refusal bias*, whereby models tend to select E as a fallback under uncertainty. To prevent spurious refusals, Option E is accepted as a valid consensus *only* if it is unanimously supported (5 out of 5 votes). If this condition is not met, votes for E are discarded, and the majority decision is recomputed over the remaining options. In cases where majority voting does not yield a clear consensus, specifically, when the vote margin between the two most frequent options is ≤ 1 or a perfect tie occurs, the system activates a deterministic **Iterative Refinement** procedure. During refinement, the reasoning modules are re-invoked with a constrained version of the original prompt, in which low-support options are explicitly masked (e.g., replaced with *"do not consider this option"*). This forces the model to re-evaluate the question within a reduced answer space, amplifying stable reasoning patterns. The refinement loop is bounded to a maximum of two additional rounds to guarantee termination. If no consensus is reached after the final iteration, the system deterministically selects the option with the highest cumulative vote count, breaking residual ties via lexicographical order.

System Details. In line with the objectives of the shared task, which aims to assess the behavior of both large and small language models in the financial domain, we evaluate a deliberately diverse set of LLMs differing in scale, openness, and architectural design. This selection is intended to analyze how model capacity and accessibility interact with our inference-time architecture, rather than to optimize performance through model-specific tuning. Specifically, the evaluated models allow us to compare reasoning robustness, cross-lingual transfer, and inference efficiency under a unified and controlled experimental setup.

LLaMA 3.1 8B [28] is included as a lightweight open-source baseline, reflecting the shared task's emphasis on evaluating compact models alongside larger ones.[2] Its limited parameter budget provides a lower-bound reference for assessing how much of the observed performance gains can be attributed to architectural orchestration rather than raw model scale. GPT-OSS-20B [29] serves as the primary open-source reference model. It is a reasoning-oriented Mixture-of-Experts architecture with 21B total parameters (approximately 3.6B active per inference), striking a balance between expressive capacity and computational efficiency.[3] This model is particularly well-suited to our setting, as it enables extensive stochastic sampling and parallel inference without prohibitive latency.[4] DeepSeek v3.1 [30] is included as a large-scale proprietary upper bound, featuring a Mixture-of-Experts architecture with 671B total parameters and approximately 37B active per inference.[5] Its strong performance on complex reasoning tasks makes it a suitable reference to assess whether inference-time orchestration can narrow the gap between mid-sized open models and state-of-the-art proprietary systems. Inference for DeepSeek is

---

[2]https://huggingface.co/meta-llama/Llama-3.1-8B
[3]https://huggingface.co/openai/gpt-oss-20b
[4]In our setup, the `reasoning_effort` parameter is set to `low` for the Orchestrator and `medium` for the specialized reasoning modules.
[5]https://www.deepseek.com

performed via the Ollama Cloud managed service, enabling access to datacenter-scale hardware without local resource constraints.[6]

All models are evaluated *as-is*, without any additional fine-tuning or domain adaptation. Consequently, all performance differences observed in the experiments are attributable solely to the proposed system architecture, prompt specialization, and deterministic aggregation strategies. To ensure comparability across languages and isolate cross-lingual reasoning effects, the same English prompts are used for English, Italian, and Turkish inputs. This design choice reflects the fact that most large-scale pre-training and reasoning alignment occur in English, and allows us to explicitly measure how well structured reasoning patterns transfer to lower-resource languages. Overall, this experimental setup enables a controlled investigation of how model scale and openness interact with constrained agentic prompting and self-consistency in multilingual financial MCQA. The orchestration logic is implemented via a lightweight custom controller, without relying on external agentic frameworks, in order to retain full control over inference flow and reproducibility.

4. Results & Discussion

This section presents a comprehensive evaluation of the proposed framework across English, Italian, and Turkish. The analysis is structured to address the research questions introduced in Section 1, covering architectural effectiveness and self-consistency (RQ1, RQ4), cross-lingual transfer (RQ3), and sensitivity to question structure and difficulty (RQ2).

4.1. Experimental Setup

We adopt a strict separation between system calibration and final evaluation, using only the official PFB datasets. All architectural choices, prompt designs, and routing heuristics are defined on the Development Set, while all reported results are obtained on the blind Test Set.

Data.   The Development Set consists of 500 instances per language. For this subset, question archetypes (*Researcher*, *Fact-Checker*, *Analyst*) are manually identified and used exclusively to validate the routing strategy. Question difficulty is *not* used during system design or calibration. Difficulty labels are employed only *a posteriori* for stratified result analysis. The development data include 177 conceptual, 148 assertion-based, and 175 tabular questions. Final evaluation is conducted on the official blind Test Set, comprising 1,001 unlabelled instances per language. The test set preserves the structural heterogeneity of the task, with 303 conceptual, 371 assertion-based, and 327 tabular questions. Difficulty annotations (166 easy, 771 medium, 64 hard) are used exclusively for reporting and analysis, and do not influence inference or aggregation decisions.

Evaluation Protocol and Baselines.   We evaluate the proposed framework through a multi-faceted analysis that considers overall accuracy, cross-lingual transfer, and robustness with respect to question structure and difficulty. All results are reported on the blind test set and are stratified by language, difficulty level, and question archetype when relevant. To isolate the contribution of each architectural component, we compare the full system against a set of progressively enriched baselines. These configurations allow us to disentangle the effects of routing, modular prompting, stochastic sampling, and aggregation, and to quantify the benefits of each design choice.

We consider the following inference configurations:

- **Monolithic Zero-Shot (Baseline).** A single-pass inference using a generic instruction-following prompt, without explicit reasoning steps or task decomposition. This setting represents the lower bound and measures the model's raw financial QA capability.

- **Integrated Chain-of-Thought (Single-Pass).** A stronger baseline in which the model is prompted to internally identify the question type and produce a step-by-step solution within a single inference pass. This configuration evaluates whether in-context reasoning alone can replace explicit architectural modularization.
- **Orchestrated Greedy Inference (No Voting).** The full routing and role-specific prompting architecture is activated, but inference is performed deterministically ($N = 1, T = 0$). This setting isolates the impact of semantic routing and specialization independently of stochastic effects.
- **Stochastic Ensemble (No Iterative Refinement).** Multiple parallel reasoning instances are generated ($N = 5, T = 0.7$) and aggregated via majority voting, while disabling the iterative refinement loop. Comparing this configuration to greedy inference quantifies the contribution of diversity and self-consistency.
- **Full System.** The complete architecture includes semantic routing, stochastic ensemble reasoning, majority voting, and iterative refinement. This configuration represents the final system submitted to the shared task.

Unless otherwise stated, all configurations share the same base prompts and inference constraints, ensuring that observed performance differences are attributable solely to architectural components rather than prompt content.

Table 1

Accuracy (%) on the PFB English Test Set. Comparison of models and inference configurations under the proposed pipeline. Scores are reported as overall accuracy (*Total*) and stratified by the official difficulty tiers (*Easy, Medium, Hard*); when available, results are also broken down by question typology (*Conceptual, Boolean, Tabular*). An asterisk (*) marks the configuration submitted to the shared task. The *Top-Competitor System* row reports the official leaderboard reference and is provided only with aggregate and difficulty-level scores; typology-specific results are not available (indicated by "–").

Model	Setting	Total	Difficulty Level			Question Type		
			Easy	Medium	Hard	Conceptual	Boolean	Tabular
LLaMA 3.1 8B	Baseline	45.15	46.38	45.27	40.62	65.02	31.27	42.51
GPT-OSS-20B	Baseline	66.13	77.11	64.33	59.38	54.32	76.64	69.72
	Single-Pass	77.62	84.94	76.52	71.88	73.93	71.16	88.38
	No Voting	78.52	82.53	77.30	82.81	81.06	68.82	87.20
	No Refinement	88.41	**88.55**	88.98	81.25	85.76	87.37	**92.05**
	Full System	87.80	88.10	87.73	87.88	**86.93**	85.33	90.81
DeepSeek v3.1 671B*	No Refinement	88.41	87.35	88.46	90.62	84.49	88.65	91.77
	Full System*	**89.41**	87.95	89.62	**90.62**	86.09	**90.00**	91.74
Top-Competitor System		89.00	84.00	**90.00**	87.50	-	-	-

4.2. Performance Analysis

Performance Analysis: English Language. Table 1 reports the results on the English test set, which represents a high-resource and linguistically stable setting. This scenario provides a controlled environment to assess the effectiveness of the proposed inference-time architecture and the contribution of stochastic self-consistency mechanisms, before moving to cross-lingual evaluation.

We begin by analyzing the progressive impact of architectural components on **GPT-OSS-20B**, which serves as the main ablation backbone. The *Baseline* configuration achieves 66.13% accuracy, already substantially outperforming the lightweight LLaMA 3.1 8B baseline (45.15%), but still exhibiting clear weaknesses on structurally complex questions, particularly conceptual and tabular ones. Introducing explicit reasoning via the *Single-Pass* integrated CoT yields a marked improvement (+11.49%), confirming that financial MCQA strongly benefits from structured intermediate reasoning. However, the most

significant gain arises when activating parallel stochastic sampling with majority voting. The *No Refinement* configuration reaches 88.41% accuracy (+10.79% over greedy inference), demonstrating that aggregating multiple reasoning paths effectively suppresses unstable or hallucinatory generations. This effect is especially pronounced for tabular reasoning, where accuracy increases from 88.38% to 92.05%.

The introduction of the iterative refinement mechanism (*Full System*) produces a more nuanced effect. While overall accuracy slightly decreases for GPT-OSS-20B (87.80%), refinement substantially improves performance on hard instances (from 81.25% to 87.88%). This suggests that refinement acts as a targeted intervention for ambiguous or high-complexity cases, at the cost of occasional over-correction on simpler questions. As such, refinement is not uniformly beneficial but proves critical when cognitive load increases.

A breakdown by question typology highlights the role of specialization. Tabular questions consistently achieve the highest accuracy across models, validating the effectiveness of the Analyst prompting strategy based on explicit value extraction and calculation. Boolean verification remains the most challenging category, reflecting the intrinsic difficulty of tracking multiple truth-value dependencies within a single question. Notably, for the full configurations, performance remains stable across difficulty levels, and in some cases improves on the Hard tier, indicating that the architecture does not degrade under increased reasoning complexity.

Regarding model scale, DeepSeek v3.1 is evaluated only under the two most relevant configurations (*No Refinement* and *Full System*). Due to its high computational cost, extensive ablation is impractical; instead, we focus on assessing whether the proposed inference-time orchestration can further enhance an already strong proprietary model. The *Full System* achieves the best overall accuracy (89.41%) and represents our official submission. In contrast, LLaMA 3.1 8B is evaluated exclusively in the baseline setting, as its limited capacity does not support reliable stochastic ensembling, and additional architectural complexity yields marginal returns relative to computational overhead.

Finally, comparison with the official competitive system shows that our approach is particularly effective on edge cases. This indicates that inference-time architectural control can rival, and in some cases surpass, highly specialized baselines, offering a competitive alternative to resource-intensive domain adaptation strategies.

Table 2
Accuracy (%) on the PFB Italian Test Set. Overall accuracy and stratified results by difficulty and question type. The asterisk (*) indicates the official submission. Question-type scores are not available for the Top-Competitor System.

Model	Setting	Total	Difficulty Level			Question Type		
			Easy	Medium	Hard	Conceptual	Boolean	Tabular
LLaMA 3.1 8B	Baseline	40.66	37.95	41.37	39.06	65.79	27.57	32.11
GPT-OSS-20B*	Baseline	54.35	68.67	52.14	43.75	64.47	41.89	59.02
	Single-Pass	72.93	74.10	73.80	59.38	75.66	64.86	79.51
	No Voting	75.62	80.12	75.10	70.31	78.29	67.92	81.90
	No Refinement	86.01	86.75	85.99	84.38	82.57	85.95	89.30
	Full System*	87.71	**88.55**	87.81	84.38	84.16	**88.95**	**89.60**
DeepSeek v3.1 671B	No Refinement	83.80	84.52	84.07	78.79	83.05	83.11	85.14
	Full System	83.92	84.34	83.92	82.81	**85.81**	83.51	82.62
Top-Competitor System		**91.00**	86.00	**92.00**	**92.00**	-	-	-

Performance Analysis: Italian Language. Table 2 reports the results on the Italian test set. As a mid-resource language in the financial domain, Italian provides a critical benchmark for evaluating cross-lingual transfer and for assessing whether the proposed inference-time architecture can compensate for linguistic and domain mismatches without model retraining. The same models were investigated for English are evaluated here; however, for lightweight models such as LLaMA 3.1 8B, we restrict the

analysis to the monolithic baseline, as preliminary experiments showed that more complex orchestration does not yield stable gains under severe capacity and language constraints.

A direct comparison between English and Italian highlights a pronounced language gap in raw model performance. For GPT-OSS-20B, the monolithic baseline drops from 66.13% in English to 54.35% in Italian, confirming the weaker alignment of general-purpose models with Italian financial language. Crucially, this gap is almost entirely eliminated once the proposed architecture is activated. With the Full System, GPT-OSS-20B reaches 87.71% accuracy, closely matching its English counterpart (87.80%). This result demonstrates that the structured reasoning enforced by semantic routing, role-specific prompting, and self-consistent aggregation transfers effectively across languages, allowing logical scaffolding to compensate for reduced parametric familiarity with Italian.

The Italian setting also provides insight into the interaction between model scale and architectural optimization. While the small LLaMA 3.1 8B model remains below a usable threshold (40.66%), confirming that limited capacity struggles with the combined burden of language and domain complexity, a counterintuitive pattern emerges when comparing larger models. In the Full System configuration, GPT-OSS-20B (87.71%) outperforms the substantially larger DeepSeek v3.1 (83.92%). This suggests that, in non-English settings, effective instruction-following and alignment with structured prompting may outweigh raw parameter count. Consistent with the English results, stochastic ensembling plays a decisive role: transitioning from single-pass greedy inference (No Voting, 75.62%) to parallel stochastic sampling (No Refinement, 86.01%) yields a substantial performance jump, confirming the robustness of the voting mechanism across languages.

A breakdown by question typology reveals a language-specific effect of the Iterative Refinement mechanism. Unlike English, where refinement shows mixed trade-offs, Italian delivers a clear net gain. In particular, Boolean questions benefit markedly, with accuracy increasing from 85.95% to 88.95%. This improvement can be attributed to the syntactic and semantic complexity of Italian financial assertions, where nested clauses and negations increase the risk of brittle reasoning. By masking low-support options and forcing re-evaluation in a constrained space, refinement mitigates linguistic ambiguity and stabilizes logical consistency.

For the Italian track, the submitted system corresponds to the GPT-OSS-20B Full configuration. Compared to the Competitive System, our approach achieves higher accuracy on Easy instances (88% vs. 86%), indicating strong reliability on foundational financial knowledge. However, a gap remains on Medium and Hard questions, where competitors achieve higher scores. This contrast underscores the central trade-off of our approach: while language-specific training can better capture nuanced, culture-dependent financial knowledge, the proposed inference-only architecture attains competitive performance (nearly 88% overall accuracy) without modifying model parameters, highlighting its portability, efficiency, and practical appeal in multilingual settings.

Performance Analysis: Turkish Language   Table 3 reports results on the Turkish test set. As a low-resource language for financial NLP, characterized by rich agglutinative morphology and limited domain-specific pre-training data, Turkish represents the most challenging scenario for evaluating cross-lingual generalization and inference stability. The baseline results highlight the severity of this setting. LLaMA 3.1 8B performs close to random guessing (30.87%), confirming that small general-purpose models are unable to cope with the combined linguistic and domain complexity. Even GPT-OSS-20B exhibits a substantial degradation in its monolithic baseline (48.85%), compared to English and Italian. However, once the proposed inference-time architecture is activated, performance improves dramatically. The `Full System` configuration reaches 88.21% accuracy, yielding a gain of nearly +40 percentage points over the baseline. This result provides the strongest evidence that structured reasoning and controlled orchestration can effectively compensate for limited language-specific pre-training, enabling robust cross-lingual transfer even in highly challenging conditions. The contribution of stochastic aggregation is particularly pronounced in Turkish. Moving from deterministic greedy inference (`No Voting`, 78.22%) to the stochastic ensemble (`No Refinement`, 87.01%) yields one of the largest improvements observed across all languages. This confirms that, in high-entropy settings where

individual generations are more prone to local inconsistencies or hallucinations, majority voting acts as an effective stabilizer by filtering idiosyncratic errors across parallel reasoning paths. The effect is especially visible on *Hard* questions, where accuracy drops to 78.12% without refinement but is restored to 87.50% in the `Full System`. This suggests that, for difficult instances in low-resource languages, the additional constrained re-evaluation step is essential to suppress spurious alternatives and force convergence.

Table 3
Accuracy (%) on the PFB Turkish Test Set. Overall accuracy and results stratified by difficulty level and question typology. The asterisk (*) denotes the official submission. Scores by question type are not available for the Top-Competitor System.

Model	Setting	Total	Difficulty Level			Question Type		
			Easy	Medium	Hard	Conceptual	Boolean	Tabular
LLaMA 3.1 8B	Baseline	30.87	30.72	30.87	31.25	49.50	19.41	26.61
GPT-OSS-20B*	Baseline	48.85	71.08	45.27	34.38	62.54	30.03	58.72
	Single-Pass	72.23	78.92	71.98	57.81	72.94	65.77	78.90
	No Voting	78.22	80.72	77.43	81.25	78.89	72.75	83.74
	No Refinement	87.01	87.95	87.55	78.12	83.44	87.50	89.85
	Full System*	**88.21**	**89.16**	**88.07**	**87.50**	83.33	**89.97**	**90.80**
DeepSeek v3.1 671B	No Refinement	82.42	85.54	82.23	76.56	**83.73**	77.31	87.16
	Full System	80.60	79.76	80.94	78.79	80.57	76.00	84.57
Top-Competitor System		88.00	88.00	88.00	**87.50**	-	-	-

A comparison across model scales further reinforces the central role of architectural design. Despite its substantially smaller size, GPT-OSS-20B with the proposed workflow outperforms the much larger DeepSeek v3.1 model (88.21% vs. 80.60%). Moreover, DeepSeek's performance degrades when the full constraint-based prompting is applied, indicating a potential mismatch between its internal safety or reasoning mechanisms and the strict inference control imposed by our pipeline. These results suggest that, in specialized multilingual financial QA, an inference architecture aligned with the model's reasoning style can be more impactful than sheer parameter count.

Finally, in the competitive setting, the submitted configuration (GPT-OSS-20B Full System) achieves state-of-the-art performance. It matches or slightly exceeds the average scores of top competitors across all difficulty tiers, reaching 89% on Easy, 88% on Medium, and 87.5% on Hard questions. Notably, this level of performance is obtained without any language-specific fine-tuning or additional data, relying exclusively on inference-time control. In a low-resource scenario such as Turkish, these results demonstrate that structured orchestration and self-consistent aggregation provide a viable and data-efficient alternative to fine-tuning for achieving competitive financial reasoning performance.

Discussion Across Languages and Research Questions. Taken together, the experimental results across English, Italian, and Turkish provide a coherent answer to the four research questions introduced in Section 1. First, results consistently show that inference-time architectural design is sufficient to reach domain, competent performance in financial MCQA (**RQ1**), especially as no model fine-tuning is applied. Across all languages, routing and role-specialized prompting outperform monolithic inference, confirming that explicitly aligning reasoning strategies with question structure is more effective than relying on implicit in-context classification. Second, stochastic self-consistency proves to be a central stabilizing mechanism (**RQ4**). Majority voting over parallel reasoning traces yields substantial gains in all settings, with particularly strong effects for structurally complex questions and in lower-resource languages. Sampling diversity, therefore, is not causing noise, but works as a practical mechanism to foster answer robustness. The iterative refinement step further strengthens this effect by intervening only when consensus is weak, and is especially beneficial for hard instances and linguistically ambiguous cases. Third, cross-lingual transfer is strongly supported by the evidence (**RQ3**). Although raw model

performance degrades when moving from English to Italian and Turkish, the full system largely closes this gap. Once enforced through controlled orchestration, structured reasoning patterns transfer effectively across languages, allowing mid-sized models to approach, or match, performance observed in large-scale resource settings. Finally, analysis across question types and difficulty levels highlights the role of structural complexity (**RQ2**). The system remains robust as difficulty increases, particularly for tabular and hard questions, and benefits from refinement precisely when reasoning ambiguity is highest. Overall, these results indicate that architectural specialization and controlled aggregation matter more than surface form in determining performance stability.

In summary, the experimental evidence demonstrates that constrained agentic prompting and deterministic self-consistency provide a robust, portable, and training-free alternative to model scaling and domain-specific fine-tuning for multilingual financial MCQA.

For a detailed error analysis with representative examples, including cases highlighting the effects of different inference strategies and multilingual reasoning, see Appendix A.2.

Computational Efficiency and Latency Analysis   A key advantage of the proposed architecture is its reliance on inference-time optimization combined with efficient model selection. To quantify this, we measured the end-to-end wall-clock time required to process the entire test set (1,001 instances) leveraging the vLLM library[7].

Benchmarks were conducted on a single NVIDIA A100 GPU (80GB). To simulate a realistic **multi-tenant production environment**, the GPU resources were shared with concurrent active processes. Consequently, the inference engine was strictly capped at a **50GB VRAM budget**. The backbone model, **GPT-OSS-20B**, utilizes a sparse *Mixture-of-Experts* (MoE) architecture. Crucially, despite its 20B total parameter count, the model activates only ∼**4B parameters per token** during inference.

This architectural sparsity, synergistic with vLLM's PagedAttention [31], enabled the full agentic workflow (comprising Routing, 5x Stochastic Sampling, and Aggregation) to achieve a total execution time of **929 seconds**, corresponding to an average latency of ∼**0.92 seconds per query**, fully accounting for the additional refinement rounds.

This result highlights the viability of the approach: by leveraging MoE sparsity, the system delivers the reasoning depth of a large-scale model with the throughput characteristic of a lightweight 4B model, maintaining sub-second latency even under shared resource constraints.

5. Conclusions

This work explored whether general-purpose LLMs can be reliably applied to multilingual Financial MCQA through inference-time control alone. Across English, Italian, and Turkish, we showed that a carefully designed *constrained agentic workflow* is sufficient to achieve strong and stable performance without any form of fine-tuning or parameter updates. The experimental results demonstrate that architectural control at inference time is a powerful alternative to model scaling and language-specific training. By combining semantic routing, role-specialized prompting, and self-consistent aggregation, the proposed system systematically outperforms monolithic prompting strategies and remains robust under increasing structural and linguistic input complexity. In particular, the architecture proves especially effective on tabular and boolean questions, where naive single-pass inference is most prone to instability. A key outcome of this study is the strong cross-lingual behavior of the proposed approach. While raw model performance degrades significantly when moving away from English, the full system largely eliminates this gap. In Italian and Turkish, the constrained agentic pipeline enables a mid-sized open model (GPT-OSS-20B) to reach or surpass the performance of substantially larger proprietary systems. Notably, on Turkish, a low-resource and morphologically complex language, the proposed system matches the best reported results across all difficulty levels, effectively positioning it as one of the strongest systems in the shared task. These findings suggest that, for multilingual financial

---

[7]https://github.com/vllm-project/vllm

reasoning, structured orchestration and controlled self-consistency can act as a practical and cost-efficient substitute for domain-specific fine-tuning. The resulting system is portable, reproducible, and entirely training-free, making it particularly attractive in settings where labeled data or computational resources are limited.

As future work, we plan to explore alternative aggregation strategies. In particular, we aim to replace the current rule-based voting and refinement mechanism with a dedicated adjudication agent (*Judge*), exposed either to the final answers alone or to their associated reasoning traces. This direction may further improve robustness on edge cases while preserving the inference-only nature of the framework.

Acknowledgments

Declaration on Generative AI

Parts of the writing and editing process for this manuscript were supported by the use of generative AI tools, specifically OpenAI's ChatGPT and Google's Gemini. These tools were employed to enhance clarity and correct spelling. All AI-assisted content was carefully reviewed and validated by the authors to ensure accuracy, originality, and compliance with ethical and scientific standards. The authors bear full responsibility for the final content.

References

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[2] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, Y. Zhang, Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL: https://arxiv.org/abs/2303.12712. arXiv:2303.12712.

[3] Y. Li, S. Wang, H. Ding, H. Chen, Large language models in finance: A survey, in: Proceedings of the fourth ACM international conference on AI in finance, 2023, pp. 374–382.

[4] H. Yang, X.-Y. Liu, C. D. Wang, Fingpt: Open-source financial large language models, 2025. URL: https://arxiv.org/abs/2306.06031. arXiv:2306.06031.

[5] F. Borazio, D. Croce, R. Basili, Adapting llms for domain-specific retrieval: A case study in nuclear safety, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 116–122.

[6] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, G. Mann, Bloomberggpt: A large language model for finance, 2023. URL: https://arxiv.org/abs/2303.17564. arXiv:2303.17564.

[7] G. Attanasio, P. Basile, F. Borazio, D. Croce, M. Francis, J. Gili, E. Musacchio, M. Nissim, V. Patti, M. Rinaldi, D. Scalena, CALAMITA: Challenge the abilities of LAnguage models in ITAlian, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 1054–1063. URL: https://aclanthology.org/2024.clicit-1.116/.

[8] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[9] K. Tian, E. Mitchell, H. Yao, C. Manning, C. Finn, Fine-tuning language models for factuality, in: NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following, 2023.

[10] J. Robinson, C. M. Rytting, D. Wingate, Leveraging large language models for multiple choice question answering, 2023. URL: https://arxiv.org/abs/2210.12353. arXiv:2210.12353.

[11] F. Borazio, A. Shcherbakov, D. Croce, R. Basili, UniTor at BioASQ 2025: Modular Biomedical QA with Synthetic Snippets and Multiple Task Answer Generation, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025.

[12] Q. Zhao, M. Zhang, Elimination-based reasoning with llm for multiple-choice educational question answering, Journal of King Saud University Computer and Information Sciences 37 (2025) 1–17.

[13] M. M. Lucas, J. Yang, J. K. Pomeroy, C. C. Yang, Reasoning with large language models for medical question answering, Journal of the American Medical Informatics Association 31 (2024) 1964–1975.

[14] K. Shang, C.-H. Chang, C. C. Yang, Collaboration among multiple large language models for medical question answering, 2025. URL: https://arxiv.org/abs/2505.16648. arXiv:2505.16648.

[15] M. Xue, D. Liu, W. Lei, X. Ren, B. Yang, J. Xie, Y. Zhang, D. Peng, J. Lv, Dynamic voting for efficient reasoning in large language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 3085–3104. URL: https://aclanthology.org/2023.findings-emnlp.2 03/. doi:10.18653/v1/2023.findings-emnlp.203.

[16] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, 2023. URL: https://arxiv.org/abs/2203.1 1171. arXiv:2203.11171.

[17] Z. Wu, L. Sheng, Y. Xia, Y. Zhang, Y. Chen, A. Zhang, Personalized recommendation agents with self-consistency, in: Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 2978–2982.

[18] Y. Li, Z. Lin, S. Zhang, Q. Fu, B. Chen, J.-G. Lou, W. Chen, Making large language models better reasoners with step-aware verifier, 2023. URL: https://arxiv.org/abs/2206.02336. arXiv:2206.02336.

[19] H. Park, J. Lee, H. Oh, Fintab-llava: Finance domain-specific table understanding multimodal llm using fintmd, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2025, pp. 235–246.

[20] M. Rizinski, D. Trajanov, Ai agents in finance and fintech: A scientific review of agent-based systems, applications, and future horizons, Computers, Materials, & Continua 86 (2026) 1.

[21] H. Yang, B. Zhang, N. Wang, C. Guo, X. Zhang, L. Lin, J. Wang, T. Zhou, M. Guan, R. Zhang, C. D. Wang, Finrobot: An open-source ai agent platform for financial applications using large language models, 2024. URL: https://arxiv.org/abs/2405.14767. arXiv:2405.14767.

[22] J. Li, J. Zhang, H. Li, Y. Shen, An agent framework for real-time financial information searching with large language models, 2024. URL: https://arxiv.org/abs/2502.15684. arXiv:2502.15684.

[23] S. Ganesh, L. Ardon, D. Borrajo, D. Garg, U. M. Sehwag, A. L. Narayanan, G. Canonaco, M. M. Veloso, Generative ai agents for knowledge work augmentation in finance, Annual Review of Control, Robotics, and Autonomous Systems 8 (2025) 189–210.

[24] H. Du, S. Thudumu, R. Vasa, K. Mouzakis, A survey on context-aware multi-agent systems: Techniques, challenges and future directions, 2025. URL: https://arxiv.org/abs/2402.01968. arXiv:2402.01968.

[25] S. Pateria, B. Subagdja, A.-h. Tan, C. Quek, Hierarchical reinforcement learning: A comprehensive survey, ACM Computing Surveys (CSUR) 54 (2021) 1–35.

[26] I. Okpala, A. Golgoon, A. R. Kannan, Agentic ai systems applied to tasks in financial services: Modeling and model risk management crews, 2025. URL: https://arxiv.org/abs/2502.05439. arXiv:2502.05439.

[27] C. Choi, J. Kwon, A. Lopez-Lira, C. Kim, M. Kim, J. Hwang, J. Ha, H. Choi, S. Yun, Y. Kim, et al., Finagentbench: A benchmark dataset for agentic retrieval in financial question answering, in: Proceedings of the 6th ACM International Conference on AI in Finance, 2025, pp. 632–637.

[28] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.2

1783. arXiv:2407.21783.

[29] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao, et al., gpt-oss-120b & gpt-oss-20b model card, 2025. URL: https://arxiv.org/abs/2508.10925. arXiv:2508.10925.

[30] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al., Deepseek-v3 technical report, 2025. URL: https://arxiv.org/abs/2412.19437. arXiv:2412.19437.

[31] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the 29th symposium on operating systems principles, 2023, pp. 611–626.

A. Appendix

A.1. Prompt Design and Examples

To operationalize the specialized reasoning capabilities described in Section 3, the architecture relies on a set of strictly formatted prompts. Each prompt is engineered to induce a specific role (Orchestrator, Analyst, Fact-Checker, Researcher) and enforces rigid output constraints (e.g., single-letter parsing). These structural guarantees are essential to enable deterministic question processing and robust aggregation during the voting phase. Representative examples of these prompts are provided below.

Prompt 1: Routing (Orchestrator Module).   This prompt activates the Orchestrator, whose role is to classify incoming queries into one of three specialized LLM modules. It ensures that each question is routed to the most suitable reasoning module, **Analyst**, **Fact-Checker**, or **Researcher**, based on its content and structure. By enforcing strict output constraints, the Orchestrator guarantees consistent downstream processing and prevents ambiguity in module selection.

> ### Prompt 1: Routing (Orchestrator Module)
>
> You are the Orchestrator of a financial analysis team.
> Your ONLY job is to analyze the user input and classify it into one of three categories:
>
> - **ANALYST**: If the input contains a Markdown Table, financial statements, or raw data requiring calculation.
> - **FACT_CHECKER**: If the input contains a numbered list of statements (e.g., "1. ... 2. ...") and asks to identify which are True/False.
> - **RESEARCHER**: For standard questions requiring general knowledge, retrieval of financial theory, or conceptual definitions.
>
> **Output Constraint:** Output ONLY the category name: ANALYST, FACT_CHECKER, or RESEARCHER. Do not write anything else.

Prompt 2: Numerical and Table-Based Reasoning (Analyst Module)   When financial tables, statements, or explicit numerical constraints are present, this prompt triggers the **Analyst** module. The module extracts required values, performs step-by-step arithmetic verification, and resolves calculation-heavy queries. This structured reasoning approach addresses the combined challenges of table structure, numerical alignment, and domain-specific terminology.

> ### Prompt 2: Numerical and Table-Based Reasoning (Analyst Module)
>
> You are a Quantitative Financial Analyst.
> You are provided with financial data (tables, balance sheets, or numerical values) in the prompt.
> **Strategy:**
>
> 1. Extract the specific values required from the provided text or table.
> 2. Perform the necessary arithmetic calculations step-by-step.
> 3. Compare your result with the options and identify the correct one.
>
> **Final Answer:** Write ONLY the uppercase letter of the chosen option (e.g., A, B, C). Do not include any additional text.

Prompt 3: True/False Verification (Fact-Checker Module)   Designed for questions containing multiple independent statements, this prompt engages the **Fact-Checker** module. The system decom-

poses the query, evaluates each statement individually as true or false, and then synthesizes the results to select the correct answer. This strategy ensures high fidelity in Boolean verification tasks.

Prompt 3: True/False Verification (Fact-Checker Module)

You are a Financial Fact-Checker.
The user provided a list of numbered statements. Your goal is to identify which are FALSE (or TRUE, as requested).
Strategy:

1. Decompose: Isolate each numbered statement.
2. Verify each statement independently using your internal knowledge.
3. Explicitly label each statement as TRUE or FALSE.
4. Match: Select the multiple-choice option that corresponds to your findings.

Final Answer: Write ONLY the uppercase letter of the chosen option (e.g., A, B, C). Do not include any additional text.

Prompt 4: Conceptual Domain Knowledge (Researcher Module)   This prompt invokes the **Researcher** module to handle conceptual or theoretical finance questions. By leveraging parametric knowledge, the module retrieves relevant definitions, regulatory context, and financial theory to inform qualitative reasoning. This ensures accurate selection of the correct answer based on domain expertise.

Prompt 4: Conceptual Domain Knowledge (Researcher Module)

You are a Financial Academic Researcher.
You are tasked with answering conceptual or theoretical finance questions.
Strategy:

1. Identify the key financial concepts and definitions in the question.
2. Synthesize the relevant domain knowledge required to answer.
3. Select the correct option based on your synthesis.

Final Answer: Write ONLY the uppercase letter of the chosen option (e.g., A, B, C). Do not include any additional text.

A.2. Error Analysis with Representative Examples

To better understand the impact of different prompting, self-consistency, and refinement strategies, we conduct a qualitative error analysis on representative examples where model predictions differ across inference settings. Specifically, we analyze cases in which *Orchestrated Greedy Inference (No Voting) fails while Stochastic Ensemble (No Refinement) succeeds*, cases where *our proposed Full System improves over Stochastic Ensemble through multi-turn refinement*, and a multilingual tabular reasoning case that evaluates the model's ability to jointly handle non-English (Italian) language understanding, numerical extraction, and table structure. Together, these examples provide insight into the mechanisms driving the observed performance gains.

Case 1 (GPT-OSS-20B): Orchestrated Greedy Inference (No Voting) vs. Stochastic Ensemble (No Refinement) on English Conceptual Question.   For this case, we examine a medium-difficulty, conceptual question (ID: Book__219), shown in Table 4. Under the **No Voting** setting, GPT-OSS-20B incorrectly predicts **E** ("None of the above"). This suggests difficulty in validating all candidate options in a single-pass reasoning process. In contrast, the **Stochastic Ensemble** strategy correctly predicts **B** with **80% confidence**, supported by the voting distribution [B, B, B, E, B].

Table 4
English conceptual question corrected by Stochastic Ensemble.

Question (ID: Book__219, Medium, conceptual-type, English)
What is a key factor that leads to the linkage between health insurance and employment in the United States?
A. Government subsidy for health insurance costs
B. The Stabilization Act of 1942, which exempted insurance from wage controls
C. Higher premiums for individual health insurance policies
D. Inability to classify individuals into precise risk groups
E. None of the above
Correct Answer: B

Interpretation. The "None of the above" option requires global verification of all alternatives rather than local semantic plausibility. While the **No Voting** fails to sufficiently reject plausible distractors, **Stochastic Ensemble** enables historically grounded reasoning to dominate through majority voting.

Case 2 (DeepSeek v3.1 671B): Full System vs. Stochastic Ensemble (No Refinement) on English Boolean Question. For this case, we examine a medium-difficulty, Boolean question (ID: Book__1386), shown in Table 5. Using the **Stochastic Ensemble (No Refinement)**, DeepSeek v3.1 671B predicts **B** with **60% confidence**, based on the vote distribution [D, B, D, B, B]. This outcome reflects competing reasoning paths across self-consistency samples, indicating the absence of a stable consensus in the interpretation of the multi-statement logic.

Table 5
English boolean question corrected by the Full System.

Question (ID: Book__1386, Medium, Boolean-type, English)
1. Hedging is a strategy to increase the returns from foreign investments regardless of currency fluctuations.
2. Foreign tourists benefit from a weaker currency when visiting the country with the stronger currency.
3. When a currency appreciates, it can buy more of other currencies than before.
Which of these statements are **FALSE**?
A. 2
B. 1
C. 2, 3
D. 1, 2
E. None of the above
Correct Answer: D

After applying **Full System** with multi-turn refinement, the model correctly predicts **D**, supported by the refined vote distribution [D, D, B, B, D]. In the refinement stage, low-support alternatives are masked, allowing the model to re-evaluate the remaining candidates under a constrained answer space.

Interpretation. The performance improvement from **Stochastic Ensemble (No Refinement)** to **Full System** arises from structured option pruning during refinement. By narrowing the set of plausible answers in the second pass, the model achieves stronger internal agreement and more stable convergence in multi-statement reasoning tasks.

Case 3 (GPT-OSS-20B): Baseline vs. Full System on Italian Tabular Question. To further evaluate the robustness of our approach across *non-English languages* and *tabular reasoning*, we analyze a medium-difficulty, tabular-format question in Italian (ID: FINANCIALS__4501), shown in Table 6.

This example is drawn from the FINANCIALS category and requires precise extraction of numerical values from a multi-column financial statement.

Table 6
Italian tabular question corrected by the Full System according to data derived from Table 7.

Question (ID: FINANCIALS__4501, Medium, Tabular-type, Italian)
Quanto denaro contante era disponibile alla fine del 2023 (USD)?
(Estratto dal Rendiconto Finanziario Consolidato; see Table 7)
A. 50.679.489
B. 36.447.024
C. 39.399.770
D. 69.080.893
E. Nessuna delle precedenti
Correct Answer: A

Under the **GPT-OSS Baseline** setting, the model predicted option **C** that does not correspond to the correct cash position at the end of 2023 in USD, reflecting difficulty in jointly handling table structure, numerical alignment, and Italian-language financial terminology. In contrast, the **Full System** correctly predicts option **A** with **60% confidence**, supported by the voting distribution [C, A, C, A, A].

Interpretation. This example highlights three compounding sources of difficulty: (i) reasoning over a dense financial table, (ii) correct temporal alignment (end of 2023 vs. beginning of the year), and (iii) comprehension of Italian financial terminology such as *"Disponibilità liquide e mezzi equivalenti"*. The **Baseline** model is distracted by numerically salient but contextually incorrect values (e.g., beginning-of-year figures or KRW-denominated columns).

The **Full System** benefits from multi-sample self-consistency and refinement, enabling consistent alignment between the question constraint ("fine del 2023", USD) and the correct table cell. Majority voting suppresses spurious but locally plausible alternatives (e.g., option **C**), while refinement reinforces agreement among samples that correctly track both language and table structure.

Table 7
Estratto del rendiconto finanziario consolidato utilizzato nella domanda FINANCIALS__4501.

Voce	Nota	2024 (KRW)	2023 (KRW)	2024 (USD)	2023 (USD)
Attività di finanziamento					
Aumento netto dei prestiti a breve termine	27	5.871.346	2.145.400	4.307.368	1.573.920
Aumento dei prestiti a lungo termine	27	404.954	354.712	297.084	260.226
Rimborso di obbligazioni e prestiti a lungo termine	27	(1.364.508)	(1.219.579)	(1.001.038)	(894.714)
Dividendi pagati		(10.888.749)	(9.864.474)	(7.988.261)	(7.236.827)
Azioni proprie acquistate		(1.811.775)	–	(1.329.164)	–
Operazioni con azionisti di minoranza		(8.511)	(9.118)	(6.244)	(6.690)
Flussi di cassa netti da attività di finanziamento		**(7.797.243)**	**(8.593.059)**	**(5.720.255)**	**(6.304.085)**
Riclassificazione in attività destinate alla vendita	32	–	(14.153)	–	(10.383)
Effetto delle variazioni dei tassi di cambio		4.821.010	792.785	3.536.815	581.607
Aumento (diminuzione) netto delle disponibilità liquide		**(15.375.314)**	**19.400.183**	**(11.279.719)**	**14.232.465**
Disponibilità liquide e mezzi equivalenti					
Inizio dell'esercizio		69.080.893	49.680.710	50.679.489	36.447.024
Fine dell'anno		53.705.579	69.080.893	39.399.770	**50.679.489**